

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій
(повне найменування факультету)

Кафедра «Системний аналіз та обчислювальна
математика»
(повне найменування кафедри)

Пояснювальна записка

до дипломної роботи

Бакалавра

(ступінь вищої освіти)

на тему Створення сайту «Запорізького авіаційного
фахового коледжу»

(назва теми)

Виконав: студент 4 курсу, групи КНТ-812сп

Спеціальності 124 «Системний аналіз»

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Інтелектуальні технології та прийняття рішень в
складних системах

ПОПОВ Д. І.

(ПРІЗВИЩЕ та ініціали)

Керівник ДЕНИСЕНКО О.І.

(ПРІЗВИЩЕ та ініціали)

Рецензент НЕЧИПОРЕНКО Н.О.

(ПРІЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій
Кафедра «Системний аналіз та обчислювальна математика»
Ступінь вищої освіти бакалавр
Спеціальність 124 «Системний аналіз»
(код і найменування)
Освітня програма (спеціалізація) Інтелектуальні технології та прийняття рішень в складних системах
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

В. о. завідувача кафедри ТЕРЕЩЕНКО Е.В.

« 16 » червня 2025 року

З А В Д А Н Н Я
НА ДИПЛОМНУ РОБОТУ СТУДЕНТА

Попова Дмитра Івановича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Створення сайту «Запорізького авіаційного фахового коледжу»

керівник проєкту (роботи) к. т. н., доц. Денисенко Олександр Іванович,

(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затвердені наказом закладу вищої освіти від « 16 » травня 2025 року № 265

2. Строк подання студентом роботи 13 червня 2025

3. Вихідні дані до роботи Структури таблиць баз даних, дизайн сайту, статті та документації до сучасних фреймворків, інформація про структурні підрозділи коледжу.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Створити сайт до відповідності до отриманого завдання, використовуючи серверні технології, сервери MySQL, Laravel, Bootstrap.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів Зображення, що відображаються на сторінках, логотип, графічні матеріали до новин подій сайту тощо

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
Розділ 1, Розділ 2	Денисенко О. І. доцент каф. САОМ	22.02.2025	13.06.25
Нормоконтроль	Широкоград Д. В доцент каф. САОМ	22.02.2025	13.06.25

7. Дата видачі завдання «22» лютого 20 25 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів роботи	Примітка
1	Затвердження завдання й теми дипломної роботи	22.02.25	
2	Аналіз літературних та інших джерел за темою дослідження	22.03.25	
3	Вибір та обґрунтування засобів розробки	15.04.25	
4	Проектування архітектури та бази даних	30.04.25	
5	Програмна реалізація та тестування системи	20.05.25	
6	Оформлення пояснювальної записки	1.06.26	
7	Попередній захист дипломного проєкту та отримання рецензій	4.06.25	
8	Захист дипломного проєкту	16.06.25	

Студент

_____ Дмитро ПОПОВ
(підпис) (Ім'я ПРИЗВИЩЕ)

Керівник роботи

_____ Олександр ДЕНИСЕНКО
(підпис) (Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

ПЗ: 85 с., 85 рис., 4 таб., 20 джерел.

Об'єкт дослідження – сайт Запорізького авіаційного фахового коледжу.

Предмет дослідження – створення веб-проєкту з базами даних, використовуючі сучасні інформаційні технології та засоби розробки, серверні технології, фреймворки.

Мета роботи – на базі фреймворків Laravel, Bootstrap, використовуючі серверні технології, створити сайт Запорізького авіаційного фахового коледжу для підвищення інформованості студентів та абітурієнтів про структуру закладу, спеціальності, події, новини.

Під час виконання дипломної роботи було розглянуто існуючі аналоги та виявлено їх недоліки. Проведено варіантний аналіз засобів розробки, спроектовано логічну та фізичну структури сайту, структуру бази даних, розроблено програмні модулі системи. Для управління контентом створено захищену адміністративну частину, яка дозволяє модифікувати структуру сайту, додавати та редагувати новини, циклові комісії, групи, студентів та викладачів, наповнювати та редагувати базу даних.

ВЕБ-САЙТ, БАЗА ДАНИХ, СКРИПТ, HTML, CSS, MYSQL, LARAVEL, BOOTSTRAP, BLADE-шаблони

ЗМІСТ

ЗАВДАННЯ	2
РЕФЕРАТ	4
ВСТУП	7
1 АНАЛІЗ ЗАВДАННЯ ТА ВИЗНАЧЕННЯ МЕТОДУ ЙОГО ВИРІШЕННЯ	9
1.1 Мета, актуальність дипломної роботи	9
1.2 Огляд літератури	12
1.3 Аналіз схожих сайтів, з метою визначення зручності використання їх сайтів.....	13
1.4 Постанова задач.....	16
2 ВИБІР ЗАСОБІВ РОЗРОБКИ ПРОГРАМНОГО ПРОДУКТУ.....	18
2.1 Аналіз особливостей роботи сайту.....	18
2.2 Види, структура та типи багатосторінкових сайтів.....	22
2.3 Огляд та вибір фреймворку для розробки серверної частини веб- сайту	28
2.4 Огляд та вибір фреймворку для розробки клієнтської частини веб- сайту	35
3 ПРОЕКТУВАННЯ АРХІТЕКТУРИ СИСТЕМИ	36
3.1 Проектування логічної та фізичної структури сайту	36
3.2 Проектування інтерфейсу.....	42
3.3 Проектування структури таблиць бази даних.....	50
4 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ.....	54
4.1 Створення проекту та початкові налаштування Laravel.....	54
4.2 Створення основних систем та сторінок веб-сайту.....	59
4.2.1 Створення панелі адміністратора.....	61
4.2.2 Створення системи новин	68
4.2.3 Створення клієнтської частини	76

5 ТЕСТУВАННЯ СИСТЕМИ	86
ВИСНОВКИ.....	88
ПЕРЕЛІК ПОСИЛАНЬ.....	89
ДОДАТОК А. Програмний код	91

ВСТУП

З кожним роком відбувається неминучий розвиток в світі, зокрема в категорії IT- технологій та це відразу підхоплюють компанії, бізнеси тощо. Зараз, зручний сайт, цікаво оформлена соціальна мережа компанії або вбудований Штучний Інтелект (ШІ) – це є норма, без якої неможливо уявити сучасний бізнес або якесь підприємство.

Сьогодні, основою будь-якого бізнесу вважається інформативний, гарно оформлений сайт на якому будь-хто може підкреслити якусь інформацію або знайти потрібний товар чи послугу. В більшості випадках, нове покоління , коли обирає що краще зробити – подзвонити чи зайти на сайт та пошукати там інформацію – обирають друге. Це новий тренд сучасності. Для цього сайт оформлюють не тільки як гарну «Цукерку», чого вистачало раніше для отримання нового дзвінка, а й наповнюють більшої кількості, потрібної, інформації.

Оскільки існує дуже багато мов програмування у різних галузях, вони дуже швидко розвиваються та кожен день з'являється щось нове, це може бути або нова мова, або нове оновлення вже існуючої. Тому, в більшості випадках, для створення сайту та подальшої її підтримки наймають спеціальну людину, яка має навички для цього.

Одним з найзручніших методів ведення сайту – панель адміністрування, вона допомагає редагувати сайт, не заходячи до коду, що є дуже зручно для людини, яка не сильно або взагалі не розбирається в програмуванні. Вона використовується як посередник між базою даних та сайтом, що додає багато переваг та зручностей для використання у різних напрямках, наприклад:

- все робиться на одній сторінці;
- додавати, редагувати або видаляти новини компанії;
- змінювати текст на сторінках;

- в залежності від умов – панель адміністратора може стати повноцінною CRM-системою;
- працювати з інформацією про працівників компанії – додавати нових працівників або видаляти тих, хто звільнився;
- не має потреби заходити до Баз даних сайту для редагування інформації.

Також великою перевагою панелі адміністратора є – гнучкість. При її створенні. На етапі розробки програміст, може, відразу створити все необхідне для зручного керування сайтом, наприклад:

- окремі розділи (Для новин, для працівників);
- якщо десь база даних не відповідає – лінк на саму базу, де вручну щось виправити;
- при необхідності – зберігання звітів, історію дзвінків, або листів.

1 АНАЛІЗ ЗАВДАННЯ ТА ВИЗНАЧЕННЯ МЕТОДУ ЙОГО ВИРІШЕННЯ

1.1 Мета, актуальність дипломної роботи

Основною метою бакалаврської дипломної роботи це – створення сайту «Запорізького авіаційного фахового коледжу». Його розробка допоможе у виконанні ряду задач:

— залучення абітурієнтів для навчання саме у цьому закладі за рахунок оновленого сайту з новим дизайном та основною інформацією для вступу. А також знайомство з нинішніми викладачами;

— інформування поточних студентів про новини, майбутні заходи, розклад занять,, екзаменів тощо;

— інформування викладачів про новини, заходи, розклад занять та НМКД;

— зручне адміністрування для додавання новини, інформації про заходи або викладачів.

Окрім загальних до web-сайту вимог до оформлення сайтів закладів освіти висуваються вимоги відповідно до статті 30 «Прозорість та інформаційна відкритість закладу освіти» Закону України «Про освіту» [1]. Відповідно до статті 30 Заклади освіти, що мають ліцензію на провадження освітньої діяльності, зобов'язані забезпечувати на своїх веб-сайтах (у разі їх відсутності – на веб-сайтах своїх засновників) відкритий доступ до такої інформації та документів:

— статут закладу освіти;

— ліцензії на провадження освітньої діяльності;

- сертифікати про акредитацію освітніх програм, сертифікат про інституційну акредитацію закладу вищої освіти;
- структура та органи управління закладу освіти;
- склад закладу освіти згідно з ліцензійними умовами;
- освітні програми, що реалізуються в закладі освіти, та перелік освітніх компонентів, що передбачені відповідною освітньою програмою;
- ліцензований обсяг та фактична кількість осіб, які навчаються у закладі освіти;
- мова (мови) освітнього процесу;
- матеріально-технічне забезпечення закладу освіти (згідно з ліцензійними умовами);
- наявність гуртожитків та вільних місць у них, розмір плати за проживання;
- результати моніторингу якості освіти;
- річний звіт про діяльність закладу освіти;
- правила прийому до закладу освіти;
- умови доступності закладу освіти для навчання осіб з особливими освітніми потребами;
- розмір плати за навчання, підготовку, перепідготовку, підвищення кваліфікації здобувачів освіти;
- перелік додаткових освітніх та інших послуг, їх вартість, порядок надання та оплати;
- правила поведінки здобувача освіти в закладі освіти;

- план заходів, спрямованих на запобігання та протидію булінгу (цькуванню) в закладі освіти;
- порядок подання та розгляду (з дотриманням конфіденційності) заяв про випадки булінгу (цькування) в закладі освіти;
- порядок реагування на доведені випадки булінгу (цькування) в закладі освіти та відповідальність осіб, причетних до булінгу (цькування);
- інша інформація, що оприлюднюється за рішенням закладу освіти або на вимогу законодавства.

Перелік інформації, обов'язкової для оприлюднення закладами освіти, може бути розширеним, а додаткова інформація може визначатися спеціальними законами.

Для досягнення мети було визначено наступні цілі:

- провести аналіз предметної області;
- проаналізувати конкурентів з метою визначення зручності та практичності сайтів;
- проаналізувати сучасні методи розробки сайтів;
- розробити дизайн сайту;
- спроектувати базу даних ;
- розробити Панель Адміністратора;
- провести тестування та виправити помилки при їх наявності.

1.2 Огляд літератури

Для проектування та реалізації сайту, було розглянуто певний список джерел для аналізу та порівняння сучасних технологій для розробки серверної та клієнтської частини сайту, такі як Laravel, WordPress, Django тощо. Далі наведено певний список джерел розділених по категоріях, що використовуються для написання дипломної роботи:

- Статті з порівнянням основних та популярних технологій у сфері розробки веб-сайтів або веб-додатків [2-4];
- Офіційна документації сучасних фреймворків для розробки серверної та клієнтської частини [5-7];
- Статті та поради про те, що таке база даних та як краще працювати з ними [8, 9];
- Книги, що забезпечують основну інформацію про необхідні технології [10-12].

Під час розгляду літератури будується чітке та реалістичне уявлення про те, який повинен бути веб-сайт або веб-додаток, який веб-фреймворк обрати та як працювати з базами даних.

Також надаються практичні рекомендації, які стосуються веб-фреймворків та роботи з базами даних для розробки ефективного проекту, в плані того, що він не буде «Провисати» від великої кількості користувачів, також він буде доволі безпечний за рахунок окремих технологій в кожному з визначених фреймворків.

1.3 Аналіз схожих сайтів, з метою визначення зручності використання їх сайтів

Для прийняття вірних рішень розробки слід подивитися на аналогічні вже створені веб-сторінки і виділити в них ряд рішень, які позитивно впливають на використання сайту, та недоліки, які потрібно уникнути під час розробки власного продукту.

Оскільки, готових рішень в даній нам області майже не існує, а якщо вони навіть існують, то про них немає ніякої згадки в жодній статті або документації, тому для кращого визначення основних пунктів, що повинно знаходитись на сайті та методи оптимізації цього, ми проаналізуємо декілька конкурентів в сфері надання освіти.



Рисунок 1.1 – Сайт «Запорізького медичного фахового коледжу»

Розглядаючи сайт «Запорізького медичного фахового коледжу» ми можемо побачити наступне:

- цікавий дизайн з фіолетовими тонами;
- поки сайт завантажується – з’являється цікава анімація;
- на головній сторінці є основна інформація у вигляді меню, або при гортанні сайту з правої сторони, а також новини;
- сайт розроблений на базі WordPress, цей висновок ми можемо зробити оскільки в самому низу сайту нам про це прямо написано;
- застосовано велику кількість анімацій при гортанні, що є як і плюсом так і мінусом. З плюсів – можемо виділити естетичність і те, що сайт виглядає «Живим», з мінусів – якщо на сайт заїде людина зі слабким інтернетом, він може почати «Ламатися», тобто можуть з’явитися блоки з новинами без анімації, або просто залишиться білий екран.

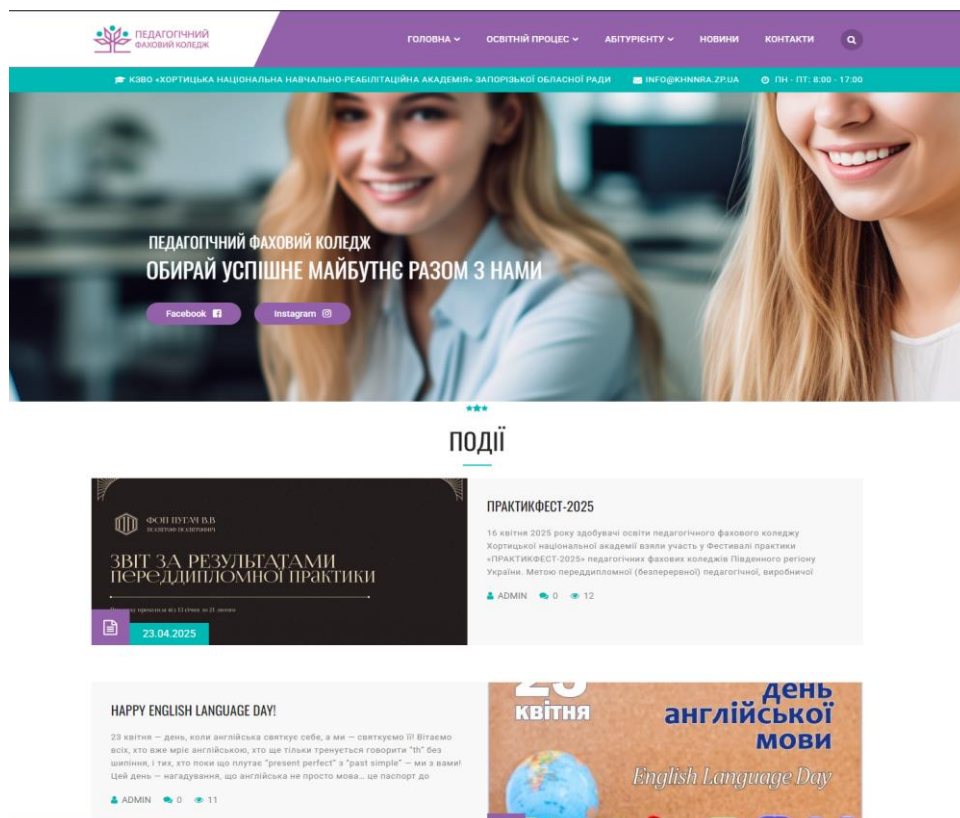


Рисунок 1.2 – Сайт «Педагогічного фахового коледжу»

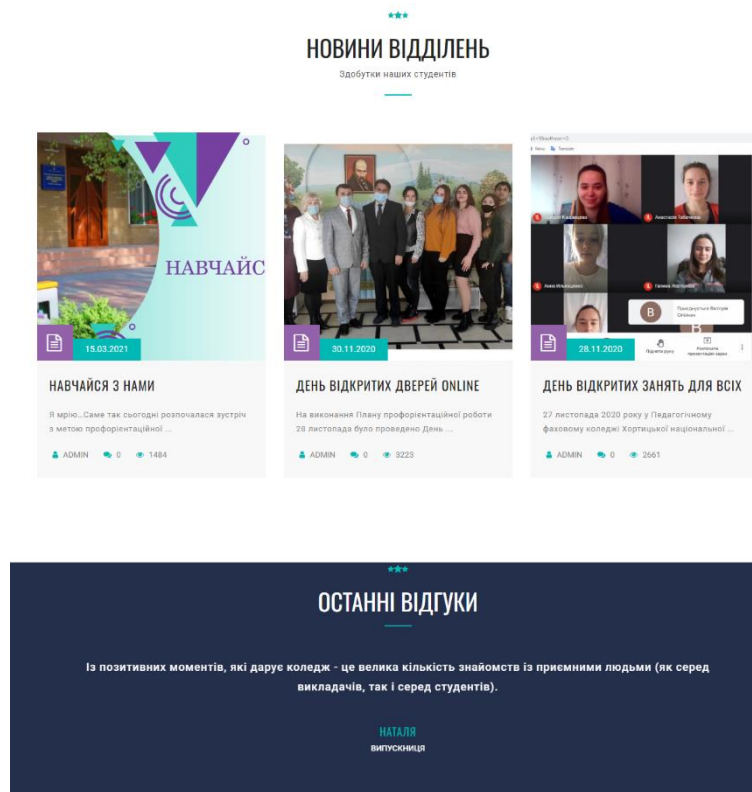


Рисунок 1.3 – Друга частина головної сторінки

Ознайомившись із сайтом «Педагогічного фахового коледжу», можемо сказати, що сайт виконаний максимально просто та лаконічно. На зверху сайту ми можемо побачити Меню сайту, за допомогою якого ми можемо перейти до інших сторінок.

На головній сторінці ми можемо побачити тільки інформацію про події коледжу, новини відділень та відгуки (рис. 1.3). Цієї інформації замало, якщо дивитись від обличчя майбутнього абітурієнту.

Проаналізувавши лише невелику кількість найближчих конкурентів, оскільки коледжів та їх дуже багато, дійдемо до висновку, що кожен сайт унікальний, але в усіх є спільні риси які використовуються всюди:

- верхня частина сайту (Header) – в ньому розташовуються основні посилання на інші сторінки сайту;
- блок з новинами – зберігаються усі новини коледжу за весь час;

— нижня части сайту – тут розташовуються посилання на основні/популярні сторінки сайту, соціальні мережі.

1.4 Постанова задач

На основі проведеного аналізу сайтів коледжів-конку rentів, склався список певних задач, які потрібно виконати для реалізації якісного продукту, у вигляді веб-сайту. Цей список складається з наступних пунктів:

- інтуїтивно зрозумілий дизайн. Щоб кожен користувач, незалежно від своєї ролі (студент, абітурієнт, викладач), зміг би зорієнтуватися куди саме йому потрібно клікнути, або прогорнути щоб знайти потрібну йому інформацію;
- зони розмежування. Сайт повинен розбиватися на дві категорії користувачів:
 - звичайний користувач – це студент, абітурієнт або викладач, які не мають доступу до Панелі Адміністратора;
 - адміністратор – це користувач, який має набагато більше можливостей, а саме: робота з новинами, додавати, редагувати або видаляти інформацію про викладачів або керівництво.
- відповідність до статті 30 «Прозорість та інформаційна відкритість закладу освіти» Закону України «Про освіту»;
- інформація про новини, події або заходи коледжу. Для того щоб користувачі були проінформовані про новини, або могли прийняти

участь в певних заходах, наприклад таких як – турніри, олімпіади тощо. Все це буде додаватися через Панель Адміністратора;

- інформація про діючих викладачів та керівництво коледжу. Для того, щоб прийняти рішення до якої саме спеціальності вступити – абітурієнт може зайти на сторінку потрібної спеціальності та ознайомитись з викладачами, інформацією про їх освіту або подивитися наукові роботи тощо.

Виходячи з цих пунктів, при дотриманні виконання кожного з них – ми отримаємо сайт, який підкреслить імідж коледжу серед інших та збільшить рівень взаємодії зі своїми нинішніми та майбутніми студентами.

2 ВИБІР ЗАСОБІВ РОЗРОБКИ ПРОГРАМНОГО ПРОДУКТУ

2.1 Аналіз особливостей роботи сайту

На сьогодні існує декілька варіантів систем роботи сайтів, такі як:

— CRM-система (Customer Relationship Management) – це система для керування взаємовідносинами з клієнтами. Іншими словами, за допомогою сайтів, які використовують дану технологію, клієнт може мати «Особистий кабінет користувача» та робити певні дії. Наприклад, запис на урок онлайн-школи, запис до лікаря тощо;

— CMS-системи (Content Management System) – це система керування вмістом сайту. Тобто, маючи таку систему, веб-сайт має окремо адміністративну панель, через яку він може керувати сайтом, а саме його змістом, картинками тощо. Існують готові CMS-системи, такі як:

- Wordpress;
- Joomla;
- Shopify.

Кожна з цих систем мають свої переваги та недоліки. Також, дуже часто при розробці веб-сайту, створюють власну CMS, оскільки таким чином, можна повністю підлаштувати її під потреби сайту, без зайвих плагінів та функцій.

Веб-сайт як спосіб знайомства користувачів з комерційною пропозицією з'явився нещодавно. Це може бути односторінковий сайт, за допомогою якого відвідувача можна підвести до покупки. Багато хто вважає такий формат роботи зі споживачами дуже перспективним і що односторінковий сайт кращий за багатосторінковий. Але незважаючи на схожість кожен із них створюється для реалізації різних завдань.

Односторінковий сайт – лендинг – має чітку структуру, що відповідає наступній схемі: цікавий заголовок, опис усіх переваг продукту з акцентом на

вигоду, яку може отримати клієнт, та заклик до дії, підкріплений формою для зворотного зв'язку.

Головне завдання лендингу – створити умови безпосереднього контакту з клієнтом. Будь-яка деталь односторінкового сайту повинна змусити людину вчинити якусь дію. Це можливо заповнення форми заявки, звернення за допомогою консультанту, оплата покупки. Структура односторінкового сайту вибудовується так, щоб відвідувач зміг повністю зосередитись на просуванні до покупки. Всі дії повинні виконуватися швидко і без будь-яких труднощів.

На відміну від лендингу, багатосторінковий сайт містить кілька розділів, тому покупцю необхідно переглядати сторінки, повертаючись в меню і переходячи по відповідним посиланням.

Завдання багатосторінкового сайту – продемонструвати покупцю якнайбільше товарів та послуг, які пропонують великі компанії. Для них формат лендингу не актуальний – він просто не зможе вмістити потрібну кількість даних. Звичайний комерційний сайт вони розглядають не тільки як базу для просування свого продукту, але і як спосіб продемонструвати статус компанії, наголосити на її спроможності.

Кількість сторінок може бути довільним. Їх додають у міру необхідності, і це залежить від обсягу інформації, яку потрібно донести до відвідувачів. У процесі просування багатосторінкових сайтів активно застосовуються принципи SEO та платна реклама. Він повинен потрапити в перші рядки пошукової видачі, що потребує серйозних витрат часу та сил, тому для вирішення короткострокових завдань багатосторінкові не підходять.

Такі сайти можна використовувати практично для будь-яких типів проектів. Вони однаково добре працюють як на сайтах електронної комерції, так і на освітніх ресурсах.

Багатосторінковий сайт – результат діяльності програмістів, дизайнерів, маркетологів, копірайтерів, верстальників та інших фахівців. Крім того, у процесі подальшої роботи з сайтом потрібна постійна оптимізація.

Зробити вибір на користь багатосторінкового чи односторінкового вебсайту непросто. З одного боку, односторінкові відмінно вписуються в сучасні умови, що передбачають зростання активності користувачів у соціальних мережах та мобільних додатках. З іншого боку, багатосторінкові сайти дозволяють надати відвідувачам набагато більше інформації та чудово вписуються у рамки традиційних потоків навігації

Наведемо кілька очевидних плюсів багатосторінкового сайту:

— високий потенціал у SEO. Істотний обсяг унікального контенту дає змогу врахувати максимальну кількість пошукових запитів. Пошукові системи відображають відповідну семантику як корисну та піднімають вебсайт у видачі. Семантичне проектування дозволяє навести на сайт значну кількість щодо дешевого пошукового трафіку;

— велика кількість корисної інформації. Будь-який аспект роботи компанії може бути висвітлений окремим розділом. Можна докладно розповісти про умови співпраці, показати портфоліо, описати переваги, розмістити відгуки задоволених клієнтів тощо;

— необмежена масштабованість. Для регулярної зміни контент-стратегії необхідно мати рішення, що масштабується. Багатосторінковий сайт надає таку можливість;

— високий рівень довіри. За рахунок великої кількості якісної інформації на багатосторінкових сайтах у споживачів складається думка про позитивний імідж компанії. Тут вони можуть побачити не лише комерційну пропозицію, а й новини фірми, дипломи співробітників, фотографії зразків продукції та відеоролики, що розповідають про процес виробництва. Наявність експертного контенту говорить про компетентність власників сайту, а відгуки клієнтів – якість запропонованого продукту. Все це підвищує ступінь довіри та викликає у користувачів бажання стати клієнтом цієї компанії;

— звичність для користувача. Незважаючи на те, що лендинг зовні виглядає як мобільний додаток і зрозуміти, як з ним поводитися, дуже легко, його простота не може переkritи переваг звичайного, грамотно структурованого веб-сайту. Користувачі вже давно засвоїли принципи роботи з меню навігації і легко орієнтуються на багатосторінкових сайтах.

Заради справедливості згадаємо про мінуси багатосторінкового сайту:

— тривалий процес розробки. Для того, щоб розробити багатосторінковий сайт, потрібно близько місяця. Точні терміни залежатимуть від того, скільки часу потрібно замовнику, щоб надати необхідний контент, наскільки оперативно він даватиме зворотній зв'язок, які зміни та в якій кількості попросить внести до вже розробленого проекту. Процес узгодження деталей при серйозному підході замовника, який працює над тим, як зробити багатосторінковий сайт по-справжньому ефективним, іноді триває кілька місяців;

— необхідність технічної підтримки. Діючий сайт необхідно постійно поповнювати цікавим та корисним контентом. Для цього потрібно знайти людей, які добре володіють матеріалом і розуміються на тонкощах вашого бізнесу. Також необхідно постійно оновлювати стрічку новин, модерувати коментарі та роботу форуму. Вся розміщена раніше інформація повинна постійно перевірятися ще раз і уточнюватися, тому для ведення сайту вам потрібні послуги контент-менеджера;

— велика кількість відволікаючих факторів. Велика кількість сторінок викликає у користувача бажання перестрибнути з одного місця на інше, що нерідко відводить його в бік від основної мети зробити вибір і зробити покупку;

— висока вартість розробки. При створенні багатосторінкового сайту ціна визначається в основному кількістю самостійних елементів – чим більше

сторінок, тим дорожче замовлення. Нижня межа вартості становить приблизно 30 тисяч гривень.

Незважаючи на всі перераховані вище недоліки, вам не обійтися без багатосторінкового сайту, якщо ви маєте широкий асортимент товарів або послуг, вирішуєте завдання поліпшення іміджу компанії і впізнаваності бренду.

В нашому випадку знадобиться саме багатосторінковий сайт для відображення всієї потрібної інформації.

2.2 Види, структура та типи багатосторінкових сайтів

Залежно від того, для вирішення яких завдань призначені багатосторінкові сайти, існує їх поділ на групи:

- корпоративні сайти – це найпоширеніший вид сайтів, на яких можна знайти дуже докладну інформацію про компанію, продукти, послуги, ціни, способи доставки, можливі варіанти контактів та контрактів. Тут розміщуються фотографії готової продукції, відгуки клієнтів, графіки запланованих заходів тощо. Є розділи, що дозволяють оформити покупку або замовити послугу, а також поставити запитання та отримати відповідь, запропонувати свої варіанти співпраці. Корпоративні сайти максимально наповнені інформацією, але іноді можуть виглядати як сайти-візитки, що пропонують отримати додаткову інформацію, зателефонувавши за вказаним номером телефону або відвідавши офіс компанії. Також до різновидів корпоративних сайтів відносяться сайтгалереї та каталоги;

— інформаційні багатосторінкові сайти – такі сайти створюються з метою заробітку на рекламі або розміщення платного контенту.

Це можуть бути:

- новинкові сайти – їх наповнюють фактами про події та тенденції сучасного життя, що відбуваються в різних сферах;
 - тематичні сайти – вони можуть розглядати конкретні теми або постійно змінювати коло запитань, що цікавлять аудиторію користувачів;
 - тематичні портали – дозволяють не тільки отримувати інформацію з теми, що цікавить, але й брати участь у її обговоренні в чаті або на форумах;
 - блог – корпоративний або особистий сайт, за допомогою якого його власник не лише доносить інформацію до своїх передплатників, а й висловлює власну думку.
- Веб-сервіси – ресурс, що дозволяє обмінюватися інформацією, вступати у комунікації тощо. буд. Існує дуже багато різновидів веб-сервісів. Серед них найпоширенішими є:
- пошукові сервіси;
 - поштові послуги;
 - сайти-агрегатори, що зводять разом продавців та покупців найрізноманітніших послуг;
 - дошки оголошень;;
 - відеохостинг;
 - соцмережі.

- Інтернет-магазини – спеціальні сайти, за допомогою яких здійснюється продаж товарів та послуг. Вони можуть бути прив'язані до існуючого в офлайн-магазині або працювати виключно на інтернет-платформі. Останній варіант представлений широко відомими маркетплейсами AliExpress, Prom, Temu та ін.;
- сайти-представництва – Це сайти, перед якими не стоїть завдання організації продажу. Вони створюються для того, щоб у людей була можливість отримати необхідні їм відомості і забезпечують роботу державних структур, реалізацію соціальних програм тощо.

Також існує кілька різновидів структур багатосторінкових сайтів. Найпоширенішими та найзручнішими для користувачів є:

Деревоподібна структура сайту.

Цей універсальний варіант структури використовується найчастіше інших. Він передбачає наступну логіку дій – спочатку користувач потрапляє на головну сторінку, потім переходить у потрібний розділ, з нього потрапляє в підрозділ і т. д. (рис. 2.1). Всі переміщення між головною сторінкою і розділами відбуваються легко і швидко, надаючи людині можливість переглядати будь-які сторінки.

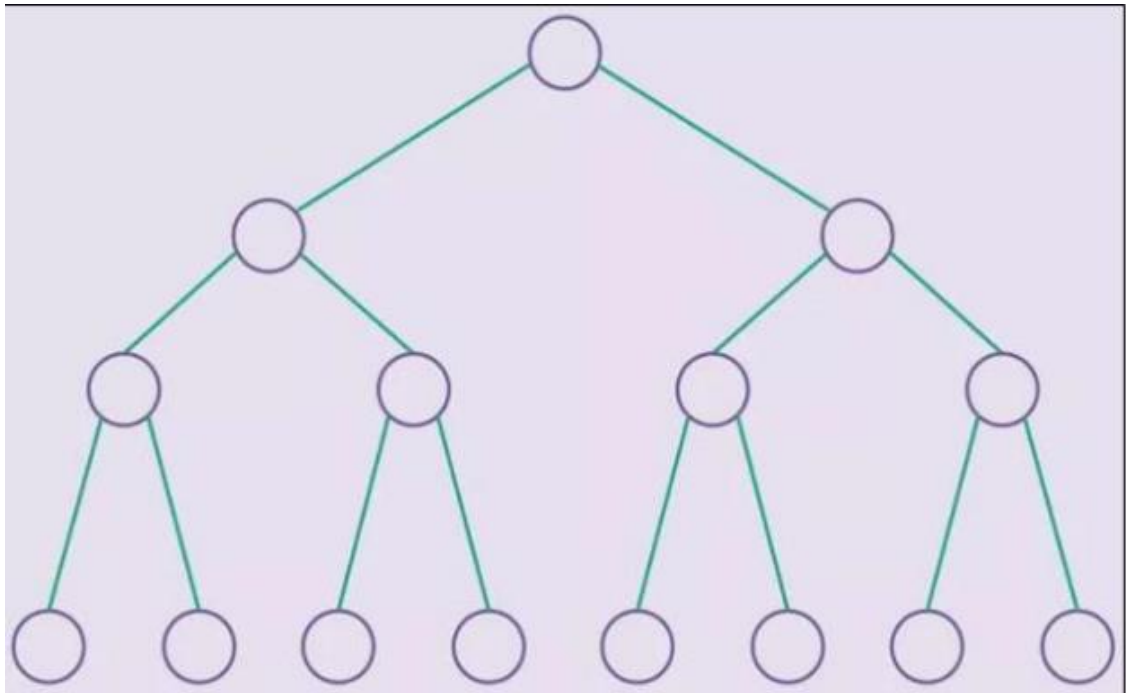


Рисунок 2.1 – Деревоподібна структура сайту

За наявності великої кількості переваг у деревоподібній структурі є суттєві недоліки. Через надмірне розширення в глибину користувачеві доводиться робити занадто багато переходів. В результаті він змушений перегортати зайві сторінки, що викликає у нього роздратування. За наявності широкої деревоподібної структури також виникають незручності – доводиться витрачати час на пошуки потрібної гілки. Отже, розробляючи макет багатосторінкового сайту, слід вибирати структуру, яка забезпечить швидкий доступ до інформації.

Лінійна структура сайту.

Логічна структура, де сторінки йдуть одна за одною (рис. 2.2). Такий сайт нагадує звичайну книгу, яку можна перегортати вперед і назад. З відкритої сторінки можна швидко перейти до наступної, попередньої або головної. Поданий контент не поділяється на рівні, що забезпечує його рівні права.

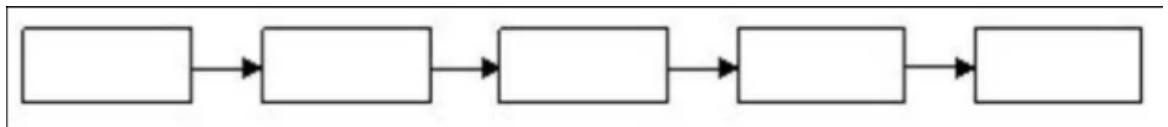


Рисунок 2.2 – Лінійна структура сайту

Застосовується така структура у випадках, коли для знайомства відвідувачів з інформацією необхідна певна послідовність. Так оформляються презентації та портфоліо, сайти, де розміщуються навчальні матеріали та посібники. У цілому нині, область застосування лінійної структури досить вузька, але є варіанти, які передбачають її розширення.

Лінійна структура сайту з альтернативними варіантами.

За наявності такої структури у відвідувача з'являється можливість переходу між двома гілками сторінок (рис. 2.3). Це полегшує процес пошуку необхідної інформації. Наприклад, відвідувачу, що опинився на головній сторінці, пропонується зареєструватися і вказати, ким він є – приватним користувачем або представником комерційної організації. Це не тільки спрямує його подальші дії, а й допоможе зібрати необхідні власнику сайту дані.

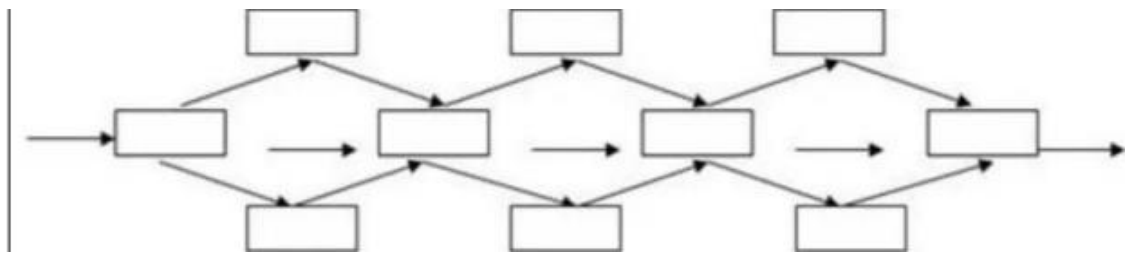


Рисунок 2.3 – Лінійна структура сайту з альтернативними варіантами

Ця структура не тільки оптимізує роботу з актуальною інформацією, але й надає можливість грамотно спрямовувати дії клієнта на потрібне русло.

Лінійна структура сайту з відгалуженнями.

Звичайна лінійна структура може бути доповнена і значною кількістю відгалужень (рис. 2.4). Кожна з них дозволяє відвідувачу докладніше

ознайомитися з важливою йому інформацією, та був знову повернутися на основну лінію.

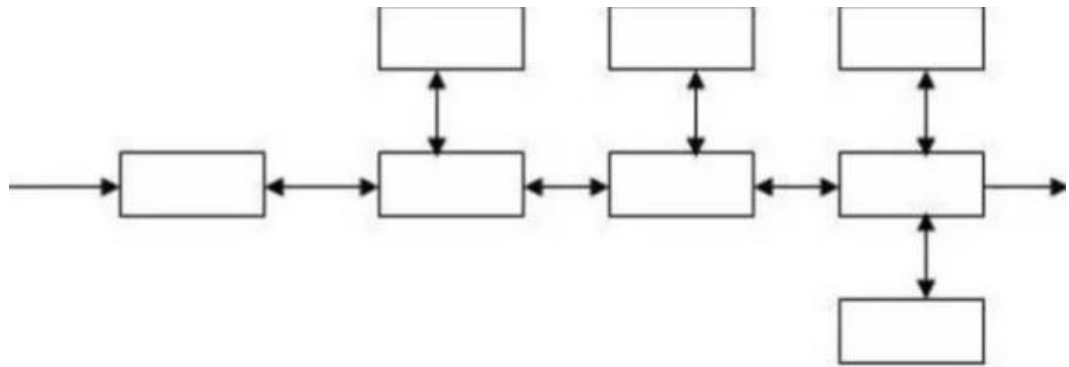


Рисунок 2.4 – Лінійна структура сайту з відгалуженнями

Відгалуження можна створити, трохи переробивши просту лінійну структуру. Внесені зміни не зроблять багатосторінковий сайт менш зручним та зрозумілим для перегляду користувачів.

Гратчаста структура сайту.

Досить складний у порівнянні з попередніми вид структури. Він надає користувачам можливість переходу як із розділу до підрозділів, так і між різними підрозділами (рис. 2.5).

Найчастіше цю складну у процесі розробки структуру застосовують упорядкування каталогів. Зрозумілий макет такого сайту розробити досить важко. Переходи можуть бути настільки заплутаними, що не кожен відвідувач захоче у них орієнтуватися.

Структура обирається згідно завдань сайту та потреб замовника. В випадку з нашим веб-сайтом, обрано саме гратчасту структуру – оскільки має переходи не тільки в середині сторінок, а й між блоками різних сторінок

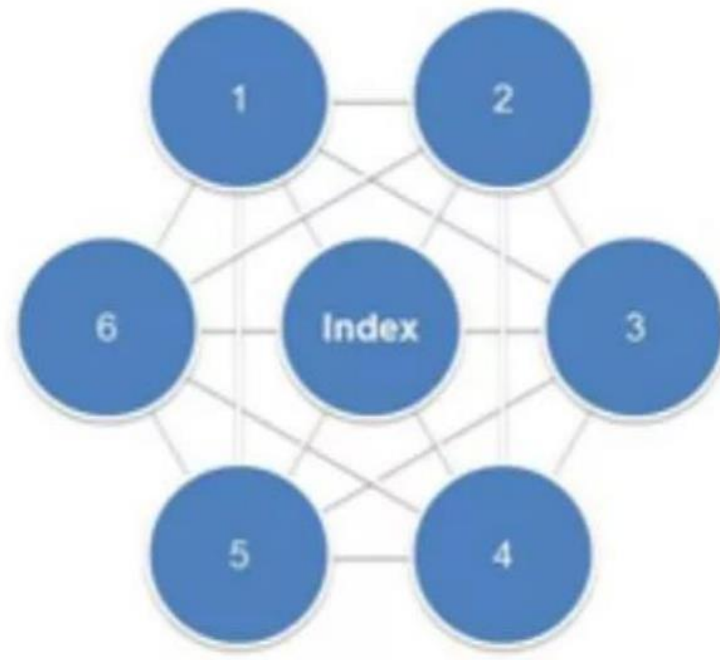


Рисунок 2.5 – Гратчаста структура сайту

2.3 Огляд та вибір фреймворку для розробки серверної частини веб-сайту

Проведемо невеликий аналіз існуючих фреймворків для розробки веб-сайтів. Оскільки розробка такого великого проекту потребує великої кількості часу – ми будемо використовувати вже готові рішення у вигляді фреймворку.

Фреймворк – це готове рішення для розробки, на основі якої можна написати свій власний код. Більшість з них вже перевірені часом та довели свою надійність та ефективність на практиці.

Основними перевагами фреймворків є:

- ефективність – використання фреймворків значно прискорює розробку, спрощує процес та знижує витрати. Вони дозволяють застосовувати готову архітектуру та автоматизовано отримувати частину функціоналу, залишаючи можливість додавати власні унікальні елементи;

— надійність – самописні рішення часто страждають від нестачі структурованості — якщо над ними по черзі працювали різні команди. Підтримувати чистоту та читання коду в таких умовах складно. Фреймворки вирішують цю проблему: вони задають чітку архітектуру, спрощують тестування та налагодження, а багато перевірок можна автоматизувати;

— безпека – фреймворки оснащені вбудованими інструментами та бібліотеками, які допомагають забезпечити захист даних та відповідність вимогам кібербезпеки. Серед них – безпечні методи автентифікації та засоби для відстеження спроб злому. Завдяки цьому фреймворки підходять для комерційних та корпоративних рішень, де безпека є критично важливою.

Але все ж таки у фреймворків також є свої мінуси, серед яких можна виділити те – що вони дають лише базову архітектуру або модель, а все інше треба дороблювати самостійно.

Виходячи з цього можемо зробити висновок – що фреймворк це дуже зручний, гнучкий та швидкий інструмент для створення веб-сайту, яке підходить для вирішення більшості проблем.

Проведемо аналіз популярних, на сьогодні, фреймворків:

WordPress – це популярна, безкоштовна та універсальна CMS-система для створення, наповнення та адміністрування сайту. Також, він є нескладний для освоєння, оскільки він набагато простіший у використанні в порівнянні із стандартним кодуванням завдяки інтуїтивно зрозумілій панелі керування (Адміністратора). У WordPress, як і в інших є свої переваги та недоліки. До переваг можемо віднести:

— нескладний у вивченні. Для вивчення базових функцій знадобиться всього декілька днів і після цього створити невеликий сайт-візитку тощо;

— універсальність. За допомогою WordPress можна створити як звичайний сайт-візитку або невеликий корпоративний сайт, так і великий

інтернет магазин, використовуючи додаткові плагіни, наприклад, Woocommerce або схожі;

— готові рішення. Оскільки WordPress являє собою систему з відкритим вихідним кодом – будь-який розробник може створювати свої теми, плагіни тощо. На «місцевому маркетплейсі», де кожен розробник може викласти свій плагін, можна знайти будь-що, від плагінів, які додають анімації, до плагінів, по типу Woocommerce, або навіть є плагін «Нової пошти», за допомогою якого, на сайт завантажуються всі відділення пошти, це дуже зручно для інтернет-магазинів;

— активне ком'юніті. Якщо виникає якась проблема в процесі розробки, то її вирішення легко знайдеться в мережі Інтернету або на форумі.

Але звісно все не так ідеально, та є свої недоліки, серед яких хочеться виділити:

— проблеми з оптимізацією. Існує багато готових рішень у вигляді плагінів, тем, тощо. Це безумовно плюс, але якщо не знати межі, та додати дуже багато плагінів, то сайт буде дуже повільний. Він буде довго завантажуватись, або не провантажуватимуться картинки, або не працюватимуть певні плагіни тощо. Все це впливає на утримання користувача на сайті;

— безпека. Оскільки WordPress це платформа з відкритим кодом, то це є плюс як для користувачів, так і для шахраїв. Вони можуть шукати певні вразливості платформи та користуватися ними. Також певна кількість корисних плагінів – платні, і звичайно є «Піратські сайти», де є всі плагіни безкоштовно, але з великою вірогідністю в них можуть вшити «майнери» або код для отримання чужих персональних даних;

— немає технічної підтримки. Якщо, при створенні сайту, ви зустрінетесь з певною кількістю проблем, вам доведеться вирішувати її

самостійно. Але, для цього стане в нагоді активне ком'юніті, про яке вказано в перевагах.

Django – це безкоштовний фреймворк, який має відкритий код. Він застосовується для створення серверної частини веб-додатків на основі мови Python. Використовується у таких великих додатках як – YouTube, Google, Instagram тощо. Як і WordPress має свої переваги та недоліки. До переваг можемо віднести:

- вбудована ORM. Допомогає прискорити розробку веб-сайту, використовуючи базову взаємодію між Базами даних та об'єктами коду, також можна працювати з декількома базами одночасно;

- налаштування URL та оптимізація SEO. Django дає можливість створювати семантичні або «Кастомні» URL-адреси для будь-якої сторінки, це дуже добре SEO-оптимізації. Сам фреймворк був спочатку створений для сайтів з новинами, тому для оптимізації під пошукові системи (Google, Yahoo тощо) було приділено багато уваги;

- безпека. Він має вбудовані системи захисту. Наприклад проти SQL-ін'єкцій, підробки запитів тощо. Це дає змогу, спокійно, створювати складні системи, залишаючи гарну швидкість завантаження сайту;

- велика кількість бібліотек. Django має велику кількість вбудованих бібліотек, яких буде повністю достатньо для створення непоганої системи, наприклад:

- Auth – для авторизації користувачів;
- Admin – для реалізації адмін-панелі;
- ORM – для роботи з базами даних;
- та багато інших.

Але якщо їх недостатньо для розробки, то зазвичай використовуються сторонні плагіни, наприклад, для авторизації користувачів за допомогою соціальних мереж, завантаження файлів тощо.

Як бачимо вище, переваг дуже багато, але є і свої недоліки:

- важкий для освоєння. Він має багато вбудованих рішень для створення системи, але мінусом цього є те, що потрібно вивчати все і відразу, що може займати велику кількість часу, навіть якщо є щось непотрібне для використання, оскільки більшість функцій взаємно пов'язанні;

- перенасиченість. Оскільки у Django є багато систем для створення складних та навантажених систем, не завжди є раціональність використання для невеликих проєктів;

- WebSocket. Хоч сам фреймворк позиціонує себе «універсальним», він не має підтримку обміну інформацією між клієнтом та сервером у режимі реального часу. Існує реалізація, тільки, за допомогою сторонніх рішень.

Laravel – фреймворк, написаний на основі PHP, використовується для створення складних сайтів та веб-додатків, Він створює певний «каркас» для сайту, та розширює функціонал PHP. До його переваг можна віднести:

- продуктивність. За допомогою фреймворку можна керувати кешуванням, має власний механізм черговості процесу, що дозволяє сайту швидко отримати інформацію з пам'яті сату, має підтримку noSQL. Все це разом, пришвидшує роботу системи;

- безпека. Має вбудовані механізми від SQL-ін'єкцій за допомогою власної ORM;

- налаштування мови. Laravel підтримує дуже багато мов, тому налаштування під різні мови реалізується достатньо швидко;

— маршрутизація. Він дозволяє створювати цілі групи маршрутів по яким інформація потрапляє з однієї точки до іншої, використовує багато способів перевірки, керуванням назвами, та застосовувати аргументи;

— вбудований шаблонізатор. Laravel має вбудований шаблонізатор Blade, який також зроблений на основі PHP, та розширює його функціонал. За допомогою нього зручно працювати з інформацією з баз даних, робити шаблони елементів, які повторюються, та зручно підключати їх до потрібних сторінок.

А також є свої певні недоліки, такі як:

— складність. Оскільки цей фреймворк має дуже багато вбудованих готових інструментів та функцій – це створює велику проблему для новачків при його вивченні, що може займати багато часу;

— оновлення. Коли виходить нове оновлення, приблизно раз на рік-півтора, для багатьох проектів виникають проблеми при оновленні, оскільки код, який вже є, може бути просто несумісний з новою версією, через що, бувають випадки, при яких треба перероблять певну кількість функціоналу;

— багатофункціональність. Це є як плюс так і мінус. Оскільки Laravel має багато вбудованих функцій, це є не дуже добре для невеликих проектів. Більше половини функціоналу може навіть не знадобитися для розробки. А для середніх – великих проектів це плюс, вони допоможуть економити велику кількість часу при розробці проекту. [10, 12, 14, 16, 18, 20]

Проаналізувавши вже існуючі готові рішення для проектування і створення систем, потрібно порівняти та обрати, що буде використовуватись при розробці. Для цього створимо таблицю:

Таблиця 2.1 – Аналіз готових систем.

	WordPress	Django	Laravel
Безпека	±	+	+
Документація	-	+	+
Складність освоєння	+	-	-
Функціональність	±	+	+
Продуктивність	-	+	+

Виходячи з результатів порівнянь, з впевненістю можемо сказати, що кожна із цих систем має свої переваги та недоліки для кожного проекту, і треба обирати саме під конкретну задачу, оскільки немає універсального рішення для всіх.

В випадку з нашим проектом нам важлива зручність розробки та безпека – тому в цьому плані лідером є Laravel, за рахунок:

- захист від SQL-ін'єкцій за рахунок власних ORM та Query Builder систем;
- захист від XSS-атак, що реалізовано за рахунок Blade-шаблонізатору, де змінні екрануються в звичайний текст;
- CSRF-токени. Додаються обов'язково до тегу Form, де методом є – POST, PUT, PATCH, DELETE;
- використання PHP. На сьогодні ця мова є одна з найпопулярніших та успішно процвітає в тих межах, де вона потрібна;
- використання Blade-шаблонізатора. За допомогою нього дуже зручно працювати, обробляти та виводити інформацію на сторінках, не використовуючи додаткові фреймворки.

2.4 Огляд та вибір фреймворку для розробки клієнтської частини веб-сайту

Для зручного використання веб-сайту, а також для швидкості розробки, буде використовуватись Bootstrap.

Bootstrap – це безкоштовний CSS, HTML та JS-фреймворк, що використовується для швидкої та адаптивної розробки веб-сайтів. Включає в себе багато різних готових рішень – від готових кнопок до готових форм вводу інформації. А також він дуже легкий для вивчення, це не займе дуже багато часу. До його переваг можна виділити:

- Кроссбраузерність. Оскільки кожен браузер, який на сьогодні є – може обробляти HTML та CSS-код трошки по своєму, то компоненти Bootstrap’у враховують цей момент та дуже добре оптимізовані;

- Простота. Bootstrap можна дуже легко та швидко вивчити, оскільки існує багато готових уроків по ньому, а також є дуже гарно описана документація;

- Адаптивність. Фреймворк має готову 12-колонну сітку, за допомогою якої розташовуються елементи, а використовуючи приставки – md, lg, sm – дуже легко адаптується під кожен розмір екрану.

Виходячи з цього – Bootstrap це універсальне рішення для багатьох задач. [11, 13, 15, 17, 19, 21].

3 ПРОЕКТУВАННЯ АРХІТЕКТУРИ СИСТЕМИ

3.1 Проектування логічної та фізичної структури сайту

Для зручного використання веб-сайтом, як користувачами, так і адміністратору, перед його розробкою потрібно розібратися із структурою проекту, як логічною, тобто які сторінки будуть входити до цього сайту, так і фізичною.

До логічної структури сайту відноситься – кількість та які сторінки будуть використовуватися, що матимуть один дизайн, тощо.

Оскільки функціонал сайту розбивається на дві ролі, а саме – адміністратор та користувач, то і функціонал вони будуть мати трошки різний.

До адміністратора має відношення не тільки сайт з інформацією, а також адміністративна панель яка буде складатися з наступних розділів:

- сторінка авторизації. Для входу в аккаунт адміністратора;
- головна. Сторінка, де змінюватиметься загальна інформація. Наприклад, розклад занять, заміни, графік консультацій тощо;
- новини. Сторінка, де адміністратор може додавати, змінювати, або видаляти новини за необхідністю. Для додавання та редагування новин – існуватиме окрема сторінка;
- циклові комісії. Сторінка, де додаватимуться циклові комісії, групи та студенти цих груп;
- викладачі. Сторінка, де додаватимуться нові викладачі, або в разі звільнення прибиратися вже існуючі.

Для звичайних користувачів будуть доступні наступні розділи:

- головна. Тут буде інформація про доступні курси навчання, новини, інформація про розклад, зміни, гуртки тощо. Також форма для зворотного зв'язку;

— коледж. Цей розділ буде відображати підрозділи, а саме – історія, структура та органи управління, забезпечення освітнього процесу, проф. спілка, публічна інформація, моніторинг якості освіти;

— освітній процес. В цьому розділі будуть розташовуватись підрозділи, які будуть мати важливу інформацію про діяльність коледжу. А саме – мова навчання, коледж online, безпека життєдіяльності, охорона праці та цивільний захист, психологічна служба, академічна доброчесність, навчальна робота, навчально-методична робота, навчально-виробнича робота, виховна робота, педагогічна рада, методична рада, циклові комісії, відділення;

— викладачам. Розташовуватимуться розділи, які будуть інформувати викладачів про – графік освітнього процесу, розклад занять, зміни у розкладі занять, розклад екзаменів, НМКД;

— абітурієнтам. В цьому розділі будуть знаходитись декілька підрозділів, які будуть мати інформацію про – Дні відкритих дверей, підготовчі курси, приймальну комісію, перелік додаткових послуг та їх вартість, розмір плати за навчання;

— студентам. Цей розділ матиме доволі широкий вибір інформації, для зручності студентів, а саме інформація про – розклад занять, дзвінків, консультацій, екзаменів, захисту кваліфікаційних робіт, ліквідації академічної заборгованості. Інформацію про ДПА/ЗНО/НМТ, рейтинги студентів, правила поведінки, призначення стипендій, гуртожиток, студентську раду, конкурси тощо;

— випускникам. Розділ який матиме інформацію про випуски груп за різні роки, а також моніторинг випускників;

— бібліотека. Це розділ, який декілька років назад став прогресивним, він зберігає в собі не тільки інформацію про те, що існує

внутрішня бібліотека та про бібліотекарів, а й всі підручники, які використовуються при навчанні, а саме:

- дисципліни повної загальної середньої освіти;
- дисципліни, що формують загальні компетентності;
- дисципліни, що формують професійні компетентності;
- вибіркові освітні дисципліни;
- періодичні видання.

— **Контакти.** Розділ, де кожен користувач, незалежно від статусу, будь-то абітурієнт, студент або викладач – може зв’язатися з необхідним департаментом, номери або адреси електронних пошт яких, вказано на цій сторінці;

— **INTERNATIONAL COOPERATION.** Розділ для міжнародних партнерів, який призначен для інформування про готовність до міжнародних відносин з іншими коледжами, університетами або компаніями, а також інформування про статут, ліцензії тощо.

Нижче наведено логічну структуру сайту та взаємозв’язки між сторінками, де з’являється інформація з панелі адміністратора на сайті (рис. 3.1–3.2):

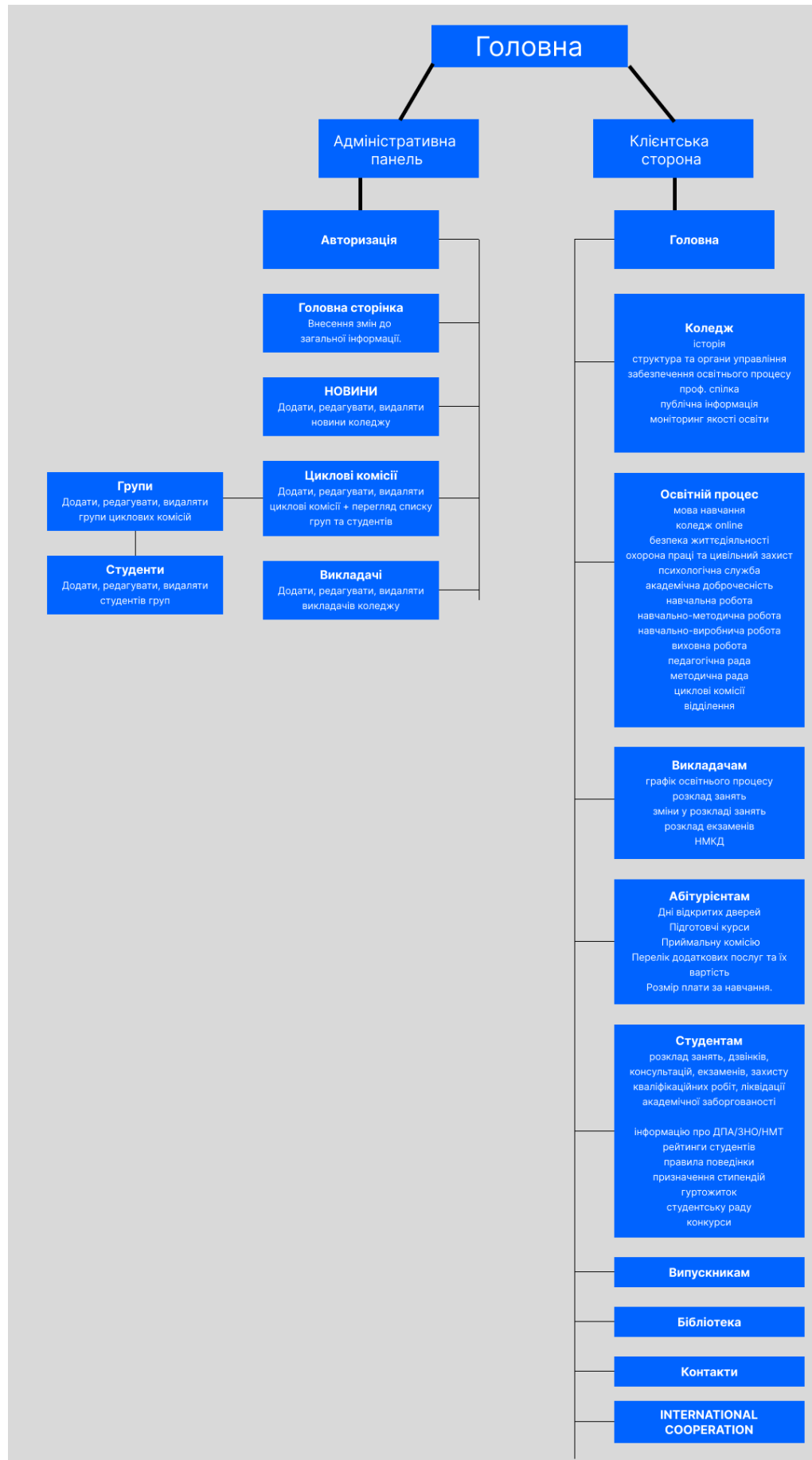


Рисунок 3.1 – Логічна структура сайту

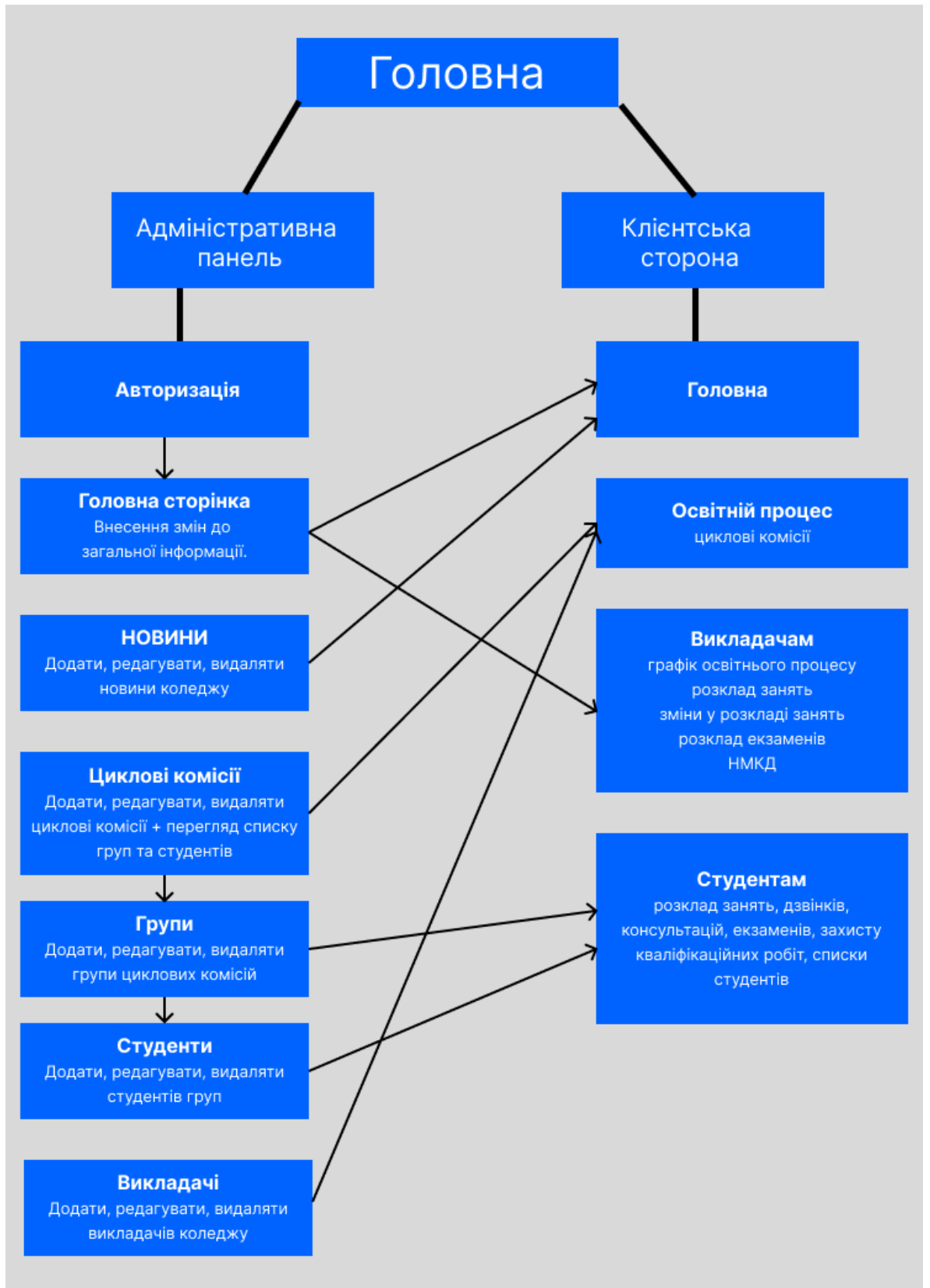


Рисунок 3.2 – Взаємозв'язки між сторінками, де з'являється інформація з панелі адміністратора

Фізична структура сайту відповідає за файлову систему сайту, а точніше, за правильне розташування файлів (Моделі, Контролери, CSS-файли, PHP-файли тощо). Оскільки ми використовуємо фреймворк Laravel, для розробки системи, ми будемо використовувати його фізичну структуру (рис. 3.3), яка розрахована на безпеку даних сайту, тому його структура відрізняється від інших своєю кількістю.

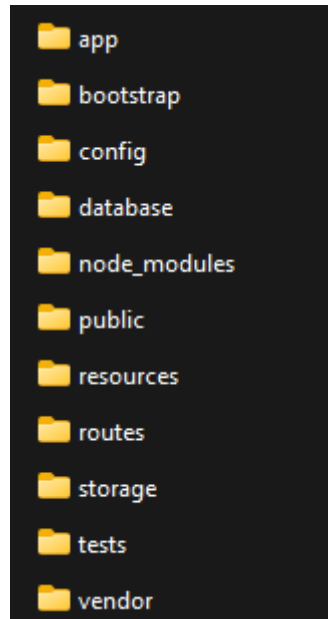


Рисунок 3.3 – Фізична структура Laravel

Не дивлячись на велику кількість каталогів, кожен з них важливий для сайту. А саме:

- каталог `app` – це «серце» проекту. Він зберігає всю основну логіку проекту – Моделі, Контролери, тощо. Без них проект не зможе працювати;
- каталог `bootstrap` – відповідає за ініціалізацію проекту, а також за «кеш» конфігурацій та маршрутів, для більш швидкої роботи фреймворку. P.s. цей каталог ніяк не пов'язаний з css-фреймворком Bootstrap;
- каталог `config` – зберігає конфігурації або ж налаштування баз даних, пошти, сесій тощо. Частина інформації потрапляю сюди через файл `.env` в основному каталозі;

- каталог `database` – тут створює та зберігаються міграції, сіди тощо. Тобто все те, що відповідає за роботу з базою даних;
- каталог `node_modules` – використовується для npm-залежностей.
- каталог `public` – ключова папка. Тут зберігаються `css`-, `js`-файли, картинки тощо. Все що потрібно для візуального відображення сайту, окрім самих сторінок;
- каталог `resources` – зберігає `blade`-шаблони, сторінки, локалізації;
- каталог `routes` – зберігає маршрути для створення URL-адрес та API – маршрутів;
- каталог `storage` – сюди Laravel зберігає логи помилок, кеш та завантажуванні файли;
- каталог `tests` – відповідає за тестування додатків;
- каталог `vendor` – зберігає `composer`-залежності, а саме бібліотеки для Laravel, що були встановлені через Composer.

3.2 Проектування інтерфейсу

Перед розробкою будь-якого веб-проекту не менш важливим є – проектування інтерфейсу, а точніше – створення схеми, або створення попереднього дизайну. Це зручно тим, що під час проектування, можливо виправити можливі неточності або переробити якийсь «шматочок» сайту.

За допомогою попереднього проектування, можна оцінити приблизний об'єм роботи, та вирішити на чому концентруватися в першу чергу, а на чому під кінець розробки.

Нижче наведено проектування інтерфейсів сайту (рис 3.4–3.18):

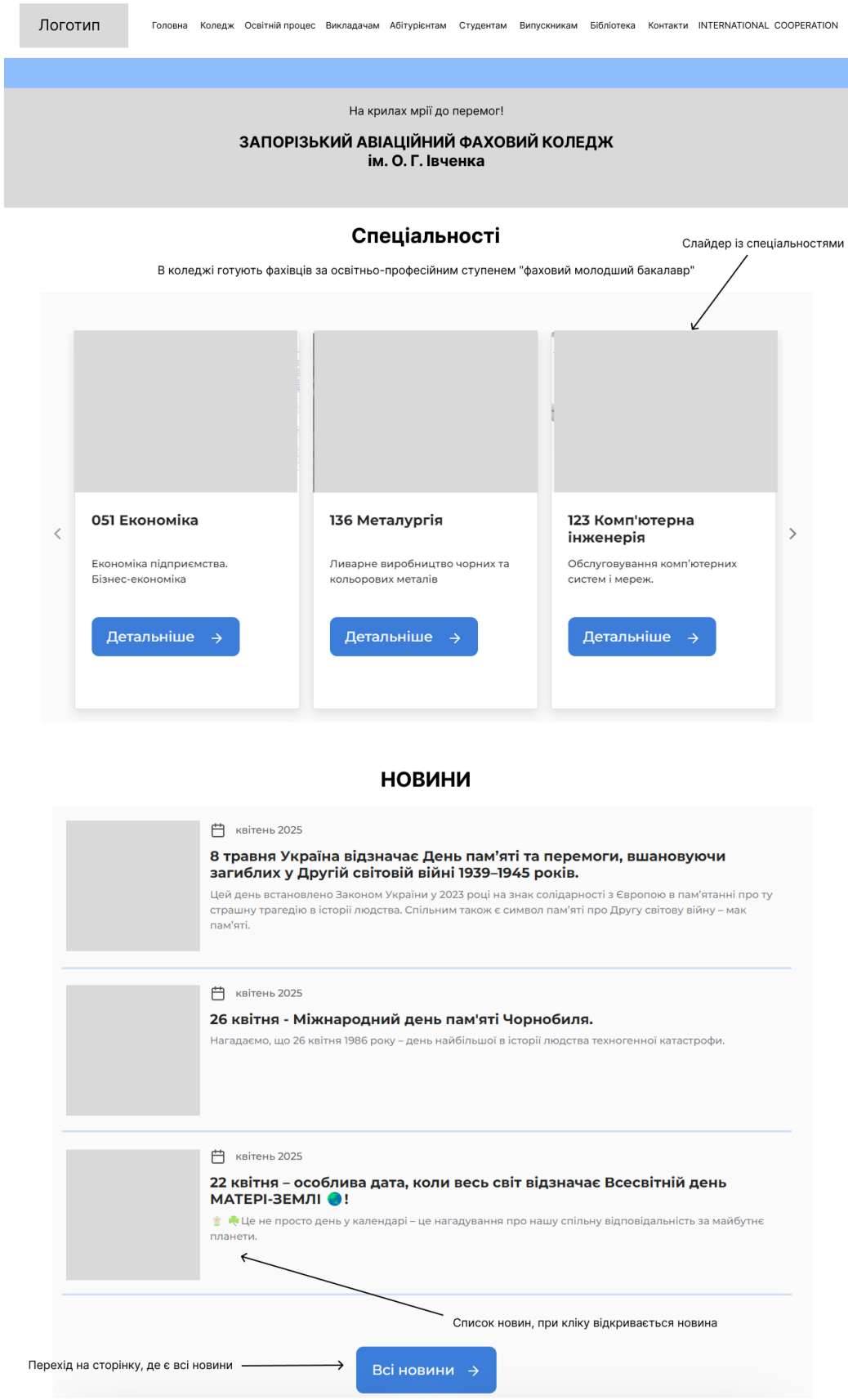


Рисунок 3.4 – Структура головної сторінки. Перша частина



Декілька причин навчатися з нами



Залишилися питання?

Надішліть нам листа на нашу пошту з цікавим для вас питанням.

Ім'я

Ел.пошта

Повідомлення

МАПА

Адреса коледжу: 69068, м.Запоріжжя, вул. Іванова 97Б



Рисунок 3.5 – Структура головної сторінки. Друга частина

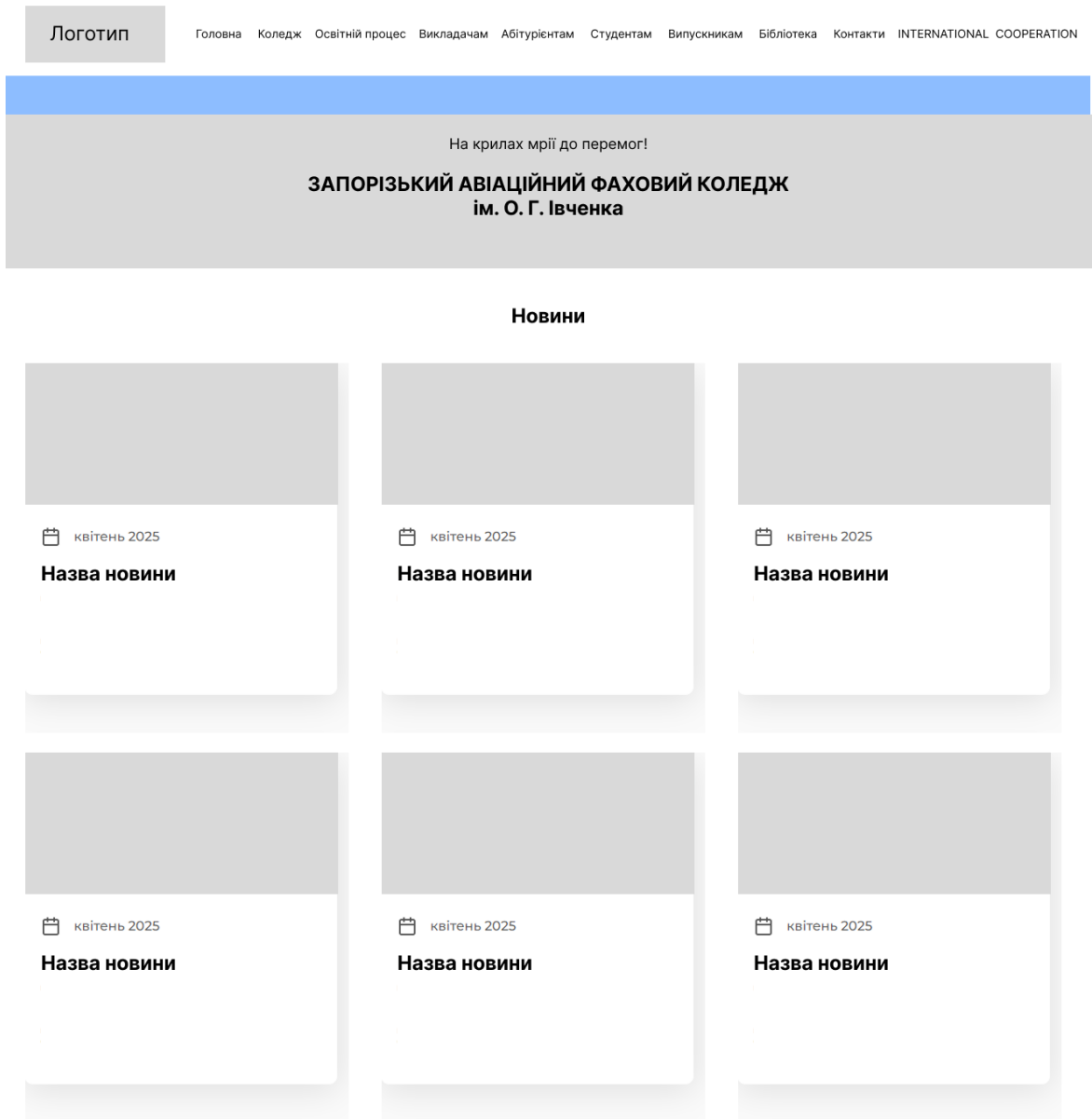
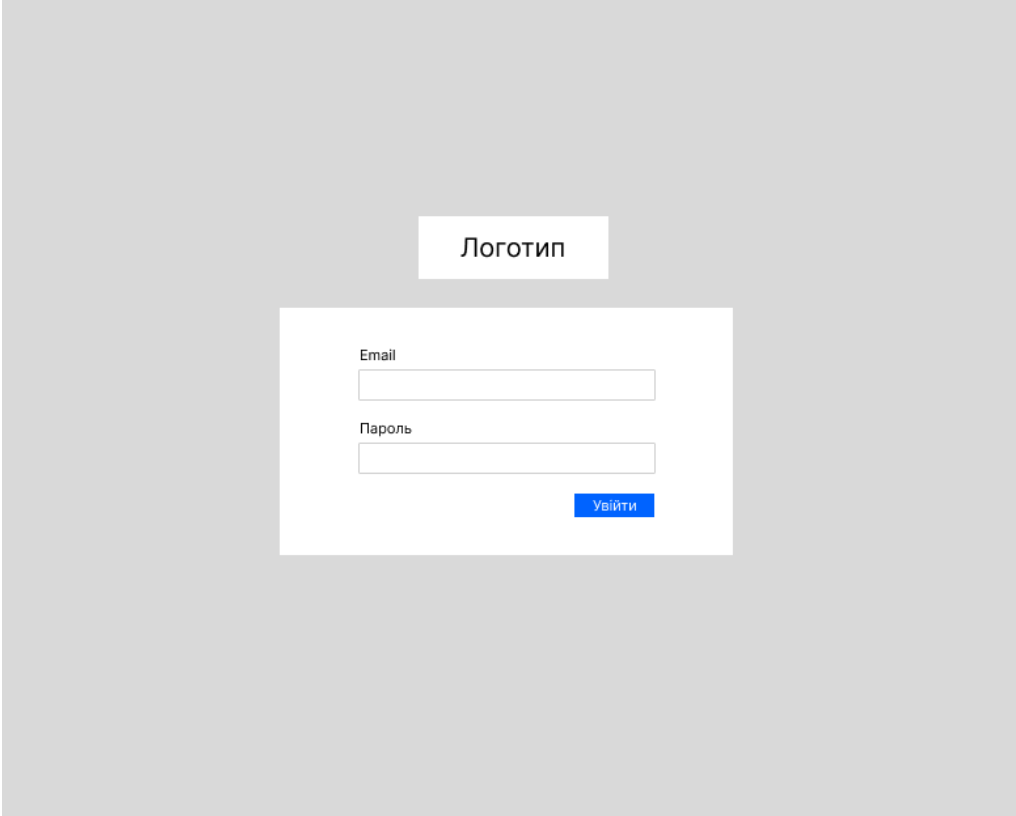
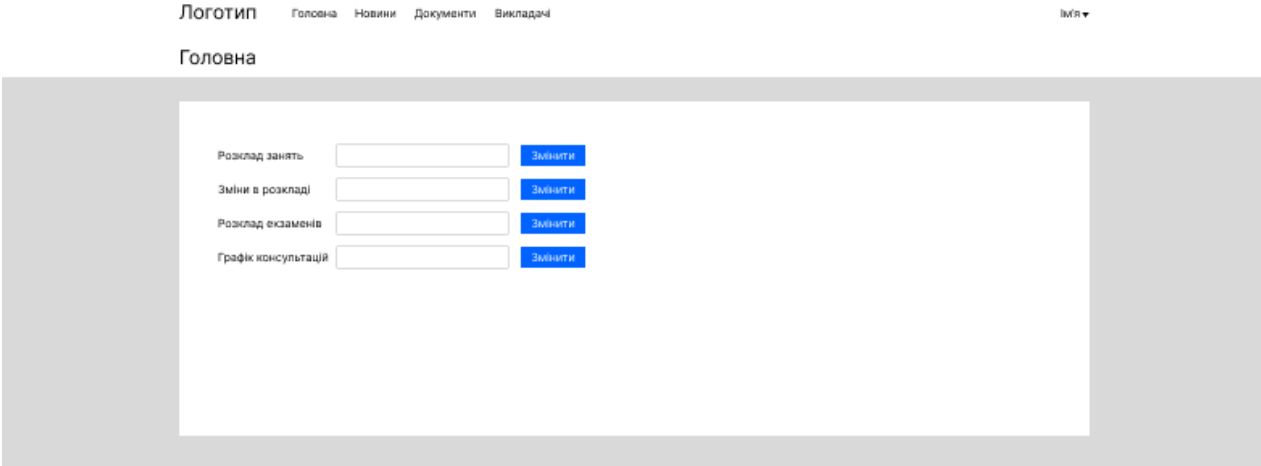


Рисунок 3.6 – Структура сторінки з новинами



The image shows a login form centered on a gray background. At the top, there is a white box containing the text "Логотип". Below this, there is a larger white box containing two input fields: "Email" and "Пароль". To the right of the "Пароль" field is a blue button with the text "Увійти".

Рисунок 3.7 – Структура сторінки авторизації в панель адміністратора



The image shows the main page of an administrator panel. At the top, there is a navigation bar with the text "Логотип" on the left and "Головна Новини Документи Викладачі" in the center. On the right side of the navigation bar, there is a user profile icon labeled "ІМЯ" with a downward arrow. Below the navigation bar, the word "Головна" is displayed. The main content area contains a white box with four rows of controls, each consisting of a text label, an input field, and a blue "Змінити" button:

Розклад занять	<input type="text"/>	Змінити
Зміни в розкладі	<input type="text"/>	Змінити
Розклад екзаменів	<input type="text"/>	Змінити
Графік консультацій	<input type="text"/>	Змінити

Рисунок 3.8 – Структура головної сторінки в панелі адміністратора

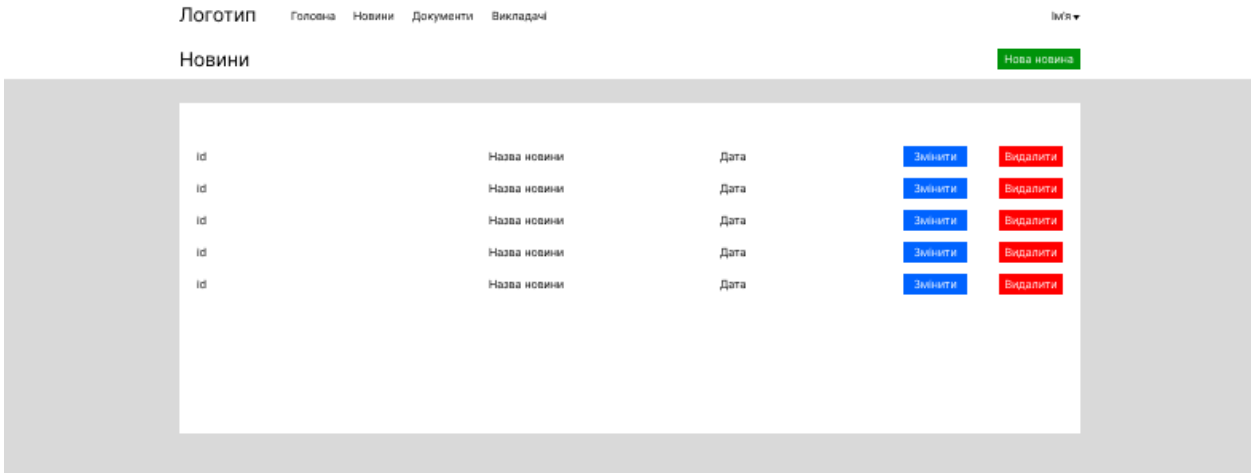


Рисунок 3.9 – Структура сторінки з новинами в панелі адміністратора

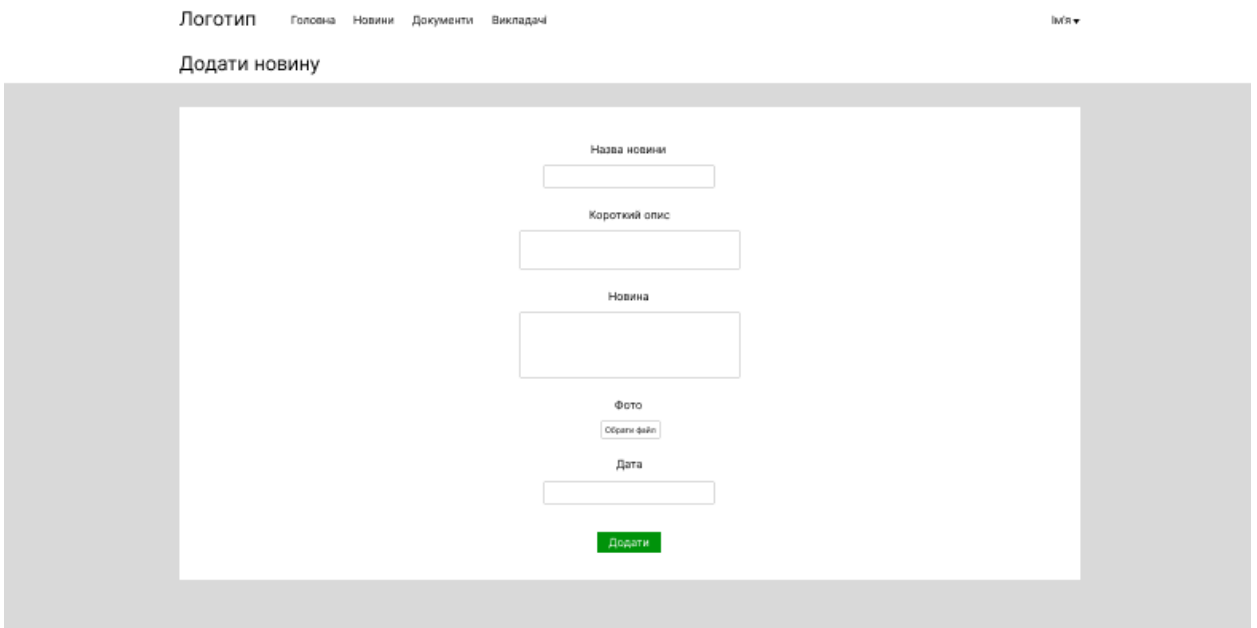


Рисунок 3.10 – Структура сторінки з додаванням новини в панелі адміністратора

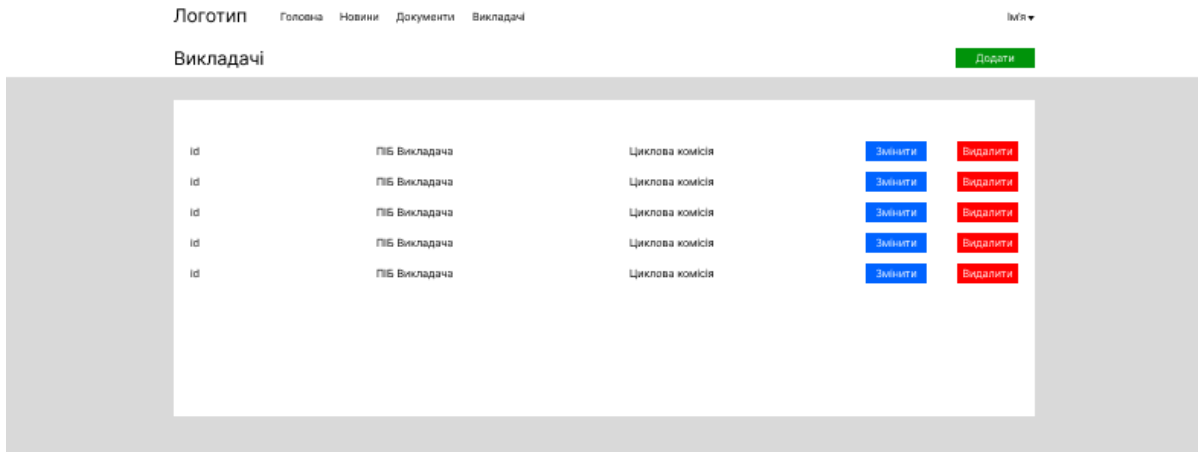


Рисунок 3.11 – Структура сторінки з викладачами в панелі адміністратора

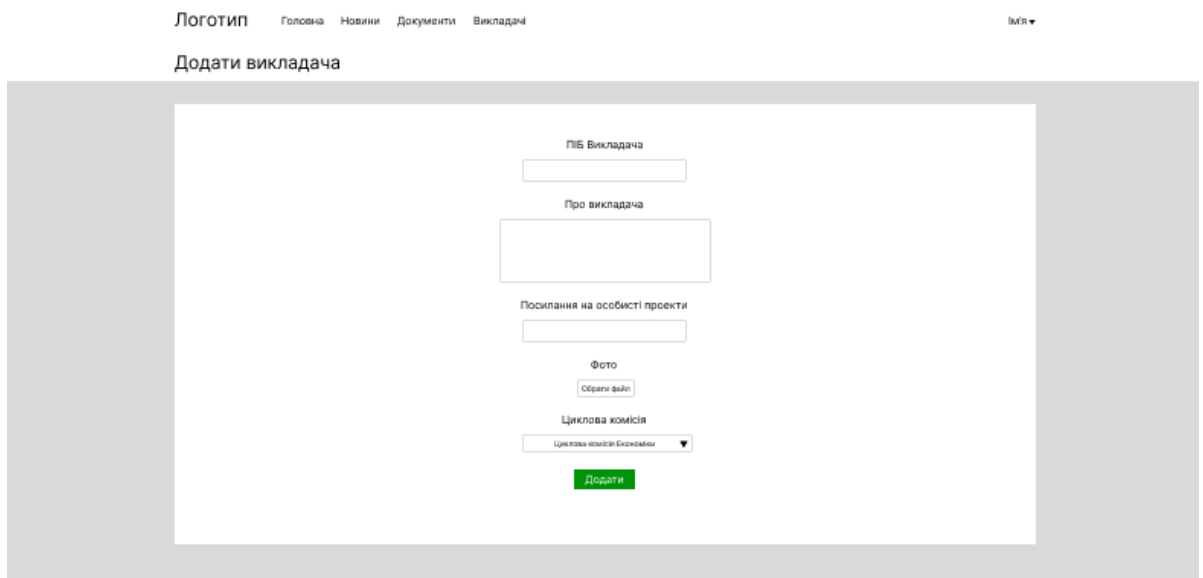


Рисунок 3.12 – Структура сторінки з додаванням викладача в панелі адміністратора

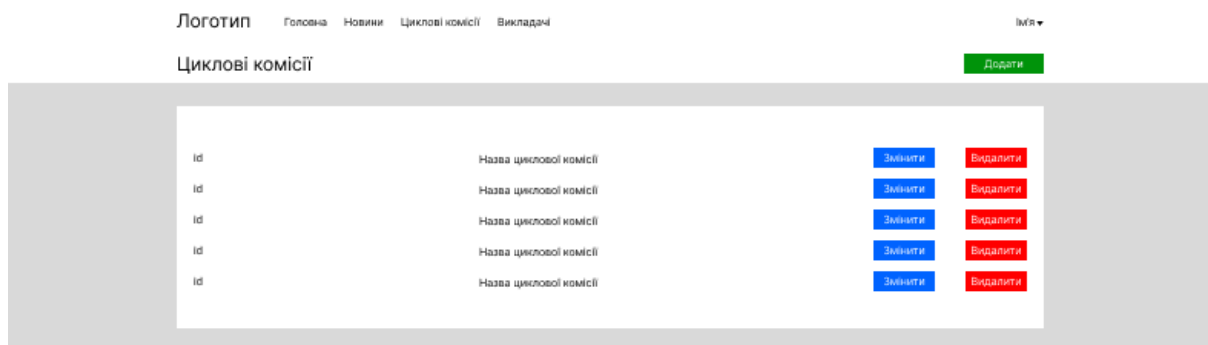


Рисунок 3.13 – Структура сторінки з цикловими комісіями в панелі адміністратора

Логотип Головна Новини Документи Викладачі Ім'я ▾

Додати циклову комісію

Назва циклової комісії

Фото

Рисунок 3.14 – Структура сторінки з додаванням циклової комісії в панелі адміністратора

Логотип Головна Новини Циклові комісії Викладачі Ім'я ▾

Групи

id	Назва групи	Назва циклової комісії	<input type="button" value="Змінити"/>	<input type="button" value="Видалити"/>
id	Назва групи	Назва циклової комісії	<input type="button" value="Змінити"/>	<input type="button" value="Видалити"/>
id	Назва групи	Назва циклової комісії	<input type="button" value="Змінити"/>	<input type="button" value="Видалити"/>
id	Назва групи	Назва циклової комісії	<input type="button" value="Змінити"/>	<input type="button" value="Видалити"/>
id	Назва групи	Назва циклової комісії	<input type="button" value="Змінити"/>	<input type="button" value="Видалити"/>

Рисунок 3.15 – Структура сторінки з групами в панелі адміністратора

Логотип Головна Новини Документи Викладачі Ім'я ▾

Додати групу

Назва групи

Циклова комісія

Рисунок 3.16 – Структура сторінки з додаванням групи в панелі адміністратора

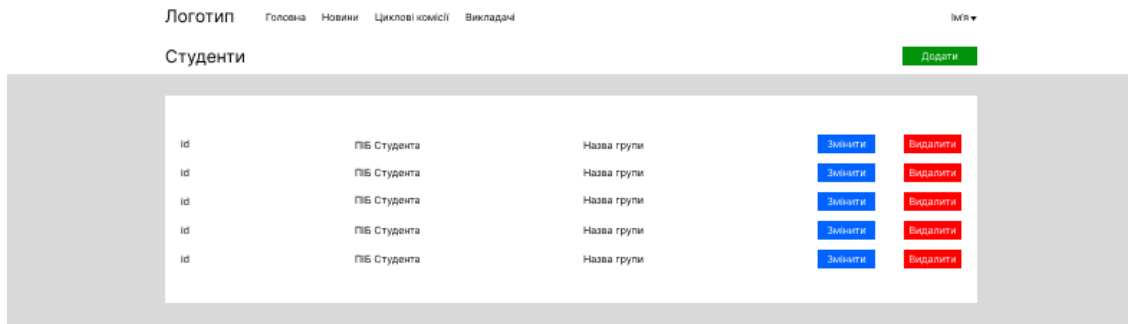


Рисунок 3.17 – Структура сторінки зі студентами в панелі адміністратора

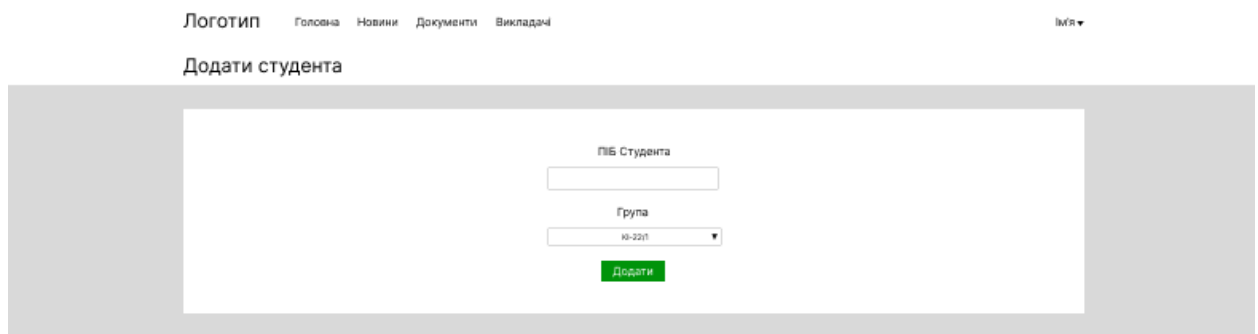


Рисунок 3.18 – Структура сторінки з додаванням студентів в панелі адміністратора

3.3 Проектування структури таблиць бази даних

База даних – ключ зручного керування сайтом, все що зберігається в ній, може якимось чином відобразитися на сайті. Зручність є в тому, що при великому об’ємі інформації, наприклад, про викладачів, їх дуже багато, база даних допомагає реалізувати «шаблон», де зберігається потрібна інформація і потім зручно відображається в тому місці на сайті, де треба.

Кожна база даних – розбивається на таблиці. Проаналізувавши наші потреби та структуру панелі адміністратора, знадобиться створити наступні таблиці – «Користувачі», «Базова інформація», «Новини», «Викладачі», «Документи».

Нижче наведено структуру таблиць (таб. 3.1–3.7):

Users – зберігає інформацію про адміністраторів сайту.

Таблиця 3.1 – Структура полів таблиці Users

Назва	Тип	Зміст
id	bigInt	Унікальний ідентифікатор запису
name	varchar	Ім'я
email	varchar	Адреса електронної пошти
email_verified_at	timestamp	Дата підтвердження електронної пошти
password	varchar	Пароль
created_at	timestamp	Дата реєстрації профілю
updated_at	timestamp	Дата оновлення профілю

Informations – зберігає базову інформацію на сайті, розклад тощо.

Структуру наведено в таблиці 3.2

Таблиця 3.2 – Структура полів таблиці Informations

Назва	Тип	Зміст
id	bigInt	Унікальний ідентифікатор запису
name	varchar	Назва інформації
link	varchar	Посилання

Cyclical_commissions – зберігає інформацію щодо наявних циклових комісій на сайті. Структуру наведено в таблиці 3.3

Таблиця 3.3 – Структура полів таблиці Cyclical_commissions

Назва	Тип	Зміст
id	bigInt	Унікальний ідентифікатор запису
name	varchar	Назва циклової комісії
photo	varchar	Фото для комісії

News – зберігає новини сайту, які відображаються на головній сторінці.

Структура наведено в таблиці 3.4

Таблиця 3.4 – Структура полів таблиці News

Назва	Тип	Зміст
id	bigInt	Унікальний ідентифікатор запису

title	varchar	Назва новини
description	varchar	Короткий опис новини
text	text	Текст новини
date	varchar	Дата новини
photo	varchar	Фото для новини
slug	varchar	Назва URL-адреси
created_at	timestamp	Дата створення новини
updated_at	timestamp	Дата оновлення новини

Groups – зберігає список груп, що знаходяться на кожній з циклових комісій. Структуру наведено в таблиці 3.5.

Таблиця 3.5 – Структура полів таблиці Groups

Назва	Тип	Зміст
id	bigInt	Унікальний ідентифікатор запису
name	varchar	Назва групи
commission_id	bigInt	Код циклової комісії

Teachers– зберігає інформацію щодо викладачів, що викладають на кожній з циклових комісій. Структуру наведено в таблиці 3.6

Таблиця 3.6 – Структура полів таблиці Teachers

Назва	Тип	Зміст
id	bigInt	Унікальний ідентифікатор запису
full_name	varchar	ПІБ викладача
desc	varchar	Опис викладача
photo	varchar	Фото
link	varchar	Посилання на роботи викладача
position	bigInt	Код циклової комісії

Students – зберігає список студентів, що знаходяться в групах. Структуру наведено в таблиці 3.7.

Таблиця 3.7 – Структура полів таблиці Students

Назва	Тип	Зміст
id	bigInt	Унікальний ідентифікатор запису
full_name	varchar	ПІБ студента
group_id	bigInt	Код групи

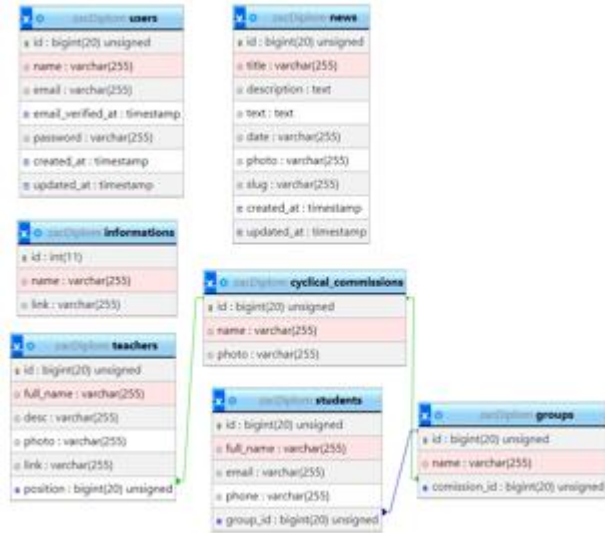


Рисунок 3.14 – Модель бази даних

4 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ

4.1 Створення проекту та початкові налаштування Laravel

Laravel – дуже «сильний» фреймворк, який може виконувати масштабні задачі, такі як – обробка будь-якої кількості запитів одночасно, забезпечення безпеки або ж працювати з запитами або подіями в реальному часі. Тому перед початком створення проекту, необхідно підготуватися.

Оскільки Laravel створений на основі PHP – необхідно мати «Локальний сервер», який має підтримку PHP, консолі та баз даних. Один з популярних серверів – OpenServer [10]

OpenServer – локальне середовище, що встановлюється на ПК користувача. Ця програма включає в себе графічний інтерфейс та може працювати з: Apache, Nginx, MySQL, MariaDB, MongoDB, PostgreSQL, PHP, PHPMyAdmin. Цього вистачить для розробки повноцінного веб-додатку.

Інтерфейс програми (рис 4.1) достатньо простий, він працює наступним чином:

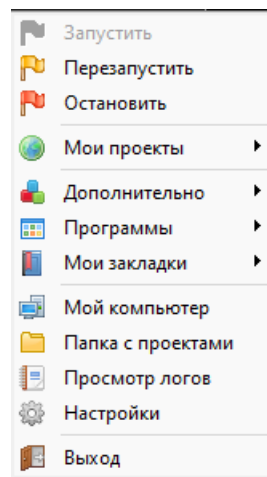


Рисунок 4.1 – Графічний інтерфейс OpenServer

— Запустити, перезавантажити, зупинити – відповідають за роботу програми;

- Мої проекти – зберігає та запускає веб-додатки які було створено у папці domains, що знаходиться в каталозі з OpenServer;
- Додатково – зберігає доступ до баз даних, консолі тощо;
- Програми – зберігає основні програми для створення веб-додатку, а саме – редактори коди, браузері тощо;
- Мої закладки – зберігає посилання на сайти, які хотів би зберегти користувач;
- Мій комп'ютер – відкриває каталог «Мій комп'ютер»;
- Папка з проектами – відкриває каталог domains, що зберігає всі проекти, які були створенні;
- Перегляд логів – відкриває додаткову панель, де можемо побачити помилки при роботі веб-сайту або самого серверу;
- Налаштування – відкриває панель з налаштуваннями серверу;
- Вихід – закриває програму.

Після успішного встановлення OpenServer, наступний етап – встановлення Laravel [5, 10, 12]. Для цього необхідно запустити консоль (рис 4.2), яка вбудована в OpenServer.

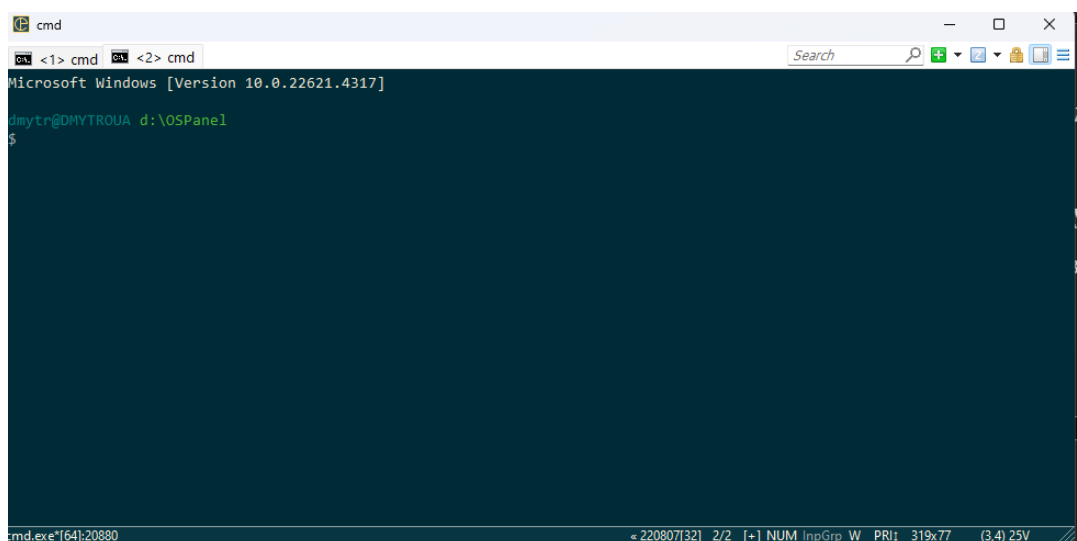


Рисунок 4.2 – Консоль OpenServer'у

Для встановлення необхідно вписати 2 команди, а саме:

`composer global require laravel/installer` – додає установник Laravel, для подальшого встановлення.

`laravel new example-app` – встановлює проект, де «example-app» – назва проекту.

Після цього проект успішно встановлено. Наступним етапом є – базове налаштування веб-проекту. Для цього в першу чергу необхідно створити базу даних, для цього необхідно перейти до PHPMyAdmin (рис. 4.3).

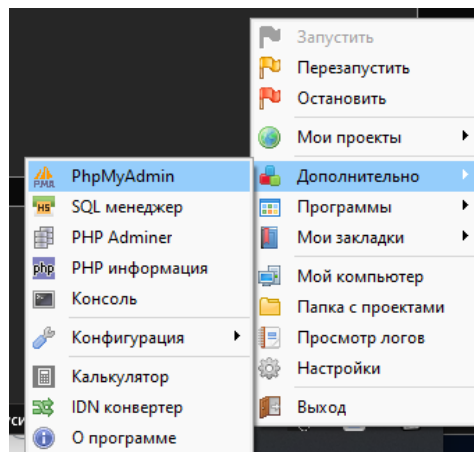


Рисунок 4.3 – Вхід до PHPMyAdmin

PHPMyAdmin – допоміжне програмне забезпечення з графічним інтерфейсом для створення та керування базами даних на основі мови MySQL. Його було обрано за надійність та швидкість роботи. Розглянемо створення бази даних в PHPMyAdmin.

Після успішної авторизації, в меню з лівої сторони необхідно натиснути на кнопку «Нова БД» та новій сторінці написати назву бази даних та натиснути кнопку «Створити» (рис. 4.4).

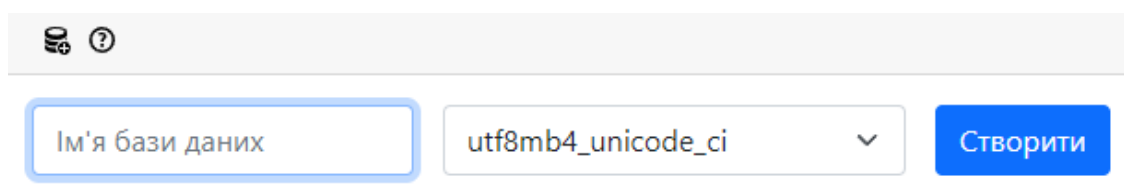


Рисунок 4.4 – Створення бази даних

Після створення бази даних, необхідно налаштувати підключення до цієї бази даних в Laravel через файл `.env`.

```
11 DB_CONNECTION=mysql
12 DB_HOST=localhost
13 DB_PORT=3306
14 DB_DATABASE=zac
15 DB_USERNAME=root
16 DB_PASSWORD=
```

Рисунок 4.5 – Підключення бази даних через `.env`

Тут ми можемо побачити наступне:

— `DB_CONNECTION` – Тип бази даних. Оскільки Laravel підтримує багато різних систем – MySQL, MariaDB, MongoDB, то треба вказати з чим саме ми працюємо;

— `DB_HOST` – Сервер , де знаходиться База даних;

— `DB_PORT` – Порт підключення до Бази даних;

— `DB_DATABASE` – Назва Бази даних;

— `DB_USERNAME` – Ім'я користувача;

— `DB_PASSWORD` – Пароль.

Тепер база даних успішно встановлено та підключено. Останній етап першого підключення – зробити Міграцію баз даних.

Міграція – спосіб керувати структурою таблиць та стовпців бази даних.

[5, 14, 20]

Розглянемо приклад на основі рис. 4.6:

```

class CreateInformationsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    0 references | 0 overrides
    public function up(): void
    {
        Schema::create(table: 'informations', callback: function (Blueprint $table): void {
            $table->id();
            $table->string(column: "name");
            $table->string(column: "link");
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    0 references | 0 overrides
    public function down(): void
    {
        Schema::dropIfExists(table: 'informations');
    }
}

```

Рисунок 4.6 – Структура файлу міграції «Informations»

Бачимо дві функції – up та down:

Функція up – створює саму таблицю, структура якої має три стовпчики:

- \$table->id() – створює стовпчик id з автоінкрементом;
- \$table->string("name") – створює стовпчик name, який має тип VARCHAR (string);
- \$table->string("link") – створює стовпчик link, який має тип VARCHAR (string).

Функція down – видаляє таблицю, якщо така існує при виконанні в консолі команди – php artisan migrate:rollback. [18]

Запуск міграції відбувається за допомогою:

```
php artisan migrate
```

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	замовчуванням
<input type="checkbox"/>	1	id 🔑	bigint(20)	UNSIGNED	Ні	Немає
<input type="checkbox"/>	2	name	varchar(255) utf8mb4_unicode_ci		Ні	Немає
<input type="checkbox"/>	3	link	varchar(255) utf8mb4_unicode_ci		Ні	Немає

Рисунок 4.7 – Результат міграції

4.2 Створення основних систем та сторінок веб-сайту

Основними система та сторінками вважаються ті, що були визначенні при створенні «Архітектури проекти». А саме – авторизація, панель адміністратора та сторінки сайта з його підрозділами.

При створенні проекту на Laravel, треба врахувати, що він працює за допомогою 3-х основних блоків:

Маршрут (Route) – механізм, який пов’язує запит користувача з необхідним методом в контролері. Маршрути зберігаються у файлі web.php. [14]

Аналізуючи приклад на рис. 4.8, приходимо висновку, що при переході на сторінку Бібліотека (url /library), то буде викликатися UkrController, а в ньому метод (функція) library.

```
Route::get(uri: '/library', action: [UkrController::class, 'library']);
```

Рисунок 4.8 – Приклад роботи маршруту

Контролер (Controlers) – мозок проекту. Саме він вирішує, що потрібно робити при отриманні запиту від користувача, незалежно від того, чи просто користувач перейшов за посиланням, або зробив відправку форми, яку контролер може обробити. [5]

Аналізуючи приклад роботи контролеру (рис. 4.9), можемо сказати, що «Метод контролеру» – це звичайна функція, яка обов’язково щось повертає назад. В цьому прикладі, вона повертає View (Сторінка сайту), який знаходиться за шляхом вказаним в лапках.

```
class UkrController extends Controller
{
    1 reference | 0 overrides
    public function library(): Factory|View{
        return view(view: 'lang.ukr.library');
    }
}
```

Рисунок 4.9 – Приклад роботи контролеру

На прикладі більш складного методу (рис 4.10), можемо побачити, що контролери вміють повертати не тільки сторінки, а ще й додатково дані з таблиць Баз даних. Можемо побачити що тут створено змінну \$commissions, в якій зберігається запит до таблиці Commissions за допомогою Commission(Модель, виділено зеленим кольором), та зроблено умову, при якій треба взяти викладачів, які займають позицію у «Цикловій комісії авіаконструкторських дисциплін», сортувати їх по id (orderBy) та отримати (get).

```
public function avia(): Factory|View{
    $teachers = Teacher::where(column: "position", operator: "Циклова комісія авіаконструкторських
    дисциплін")->orderBy(column: 'id')->get();

    return view(view: 'lang.ukr.commissions.avia', data: [
        "teachers"=>$teachers
    ]);
}
```

Рисунок 4.10 – Приклад складного методу

Додатково при поверненні (return), вказуємо не тільки view (сторінка), а й data (Об’єкт зі змінною, яку ми хочемо показати).

Для створення контролеру необхідно в консолі прописали команду:

```
php artisan make:controller UkrController|
```

Рисунок 4.11 – Створення контролеру через консоль

Модель (Model) – це спеціалізований клас, який допомагає працювати з таблиця бази даних перетворюючи їх у звичайний об'єкт. Також він являє собою «Посередника» між Контролером та Базою даних. [5, 12]

```
2 references | 0 implementations
class Teacher extends Model
{
    use HasFactory;
}
```

4.12 – Модель «Вчитель»

Особливістю систем моделей та міграцій є те, що вони – взаємопов'язані. По-друге, моделі самі визначають, яка таблиця з бази даних відноситься до них, оскільки назва моделі завжди – однина, а назва міграції – в множинні.

Існує два способи створення моделі – без міграції (рис. 4.13) та з міграцією (рис. 4.14)

```
$ php artisan make:model Teacher
```

Рисунок 4.13 – Створення моделі без міграції

```
$ php artisan make:model Teacher --migration
```

Рисунок 4.14 – Створення моделі з міграцією

4.2.1 Створення панелі адміністратора

Для розробки панелі адміністратора необхідно реалізувати систему авторизації користувачів. Оскільки входити в акаунт зможуть лише адміністратори, то для цього не потрібно створювати «Ролі».

Роль – розподілення систем в системі сайту, в залежності від типу користувача – Клієнт, Адміністратор тощо.

Візуальна частина панелі входу, було реалізовано за допомогою Blade-шаблонів Laravel, які з’являються автоматично при створенні проекту.

```

<x-guest-layout>
  <x-auth-card>
    <x-slot name="logo">
      <a href="/">
        
      </a>
    </x-slot>

    <!-- Session Status -->
    <x-auth-session-status class="mb-4" :status="session(key: 'status')" />

    <!-- Validation Errors -->
    <x-auth-validation-errors class="mb-4" :errors="$errors" />

    <form method="POST" action="{{ route(name: 'login') }}">
      @csrf

      <!-- Email Address -->
      <div>
        <x-label for="email" :value="__(key: 'Email')" />

        <x-input id="email" class="block mt-1 w-full" type="email" name="email" :value="old(key: 'email')" required autofocus />
      </div>

      <!-- Password -->
      <div class="mt-4">
        <x-label for="password" :value="__(key: 'Пароль')" />

        <x-input id="password" class="block mt-1 w-full"
          type="password"
          name="password"
          required autocomplete="current-password" />
      </div>

      <!-- Remember Me -->

      <div class="flex items-center justify-end mt-4">

        <x-button class="ml-3">
          {{ __(key: 'Увійти') }}
        </x-button>
      </div>
    </form>
  </x-auth-card>
</x-guest-layout>

```

Рисунок 4.15 – Код сторінки входу

Для реалізації входу в панель адміністратора використано тег `form`, що відповідає за форму вводу, де користувач може надіслати дані на сервер. Він має 2 основних налаштування (атрибути):

- `method` – приймає в собі `Post` (відправити або ж «запостити») та `Get` (отримати), це методи роботи форми;
- `action` – шлях до файлу-обробника, який надсилає дані з полів (`input`) при натисканні на кнопку.

Для того, щоб користувач зміг ввести та надіслати дані існує команда – `input` – сюди користувач вводить потрібну інформацію, а також кнопка (`button`), при натисканні на яку все що, написав користувач надсилається на сервер, шлях до якого вказано в тегу `form`, а саме в атрибуті `action`.

```

4 references | 0 implementations
class AuthenticatedSessionController extends Controller
{
  /**
   * Display the login view.
   *
   * @return \Illuminate\View\View
   */
  1 reference | 0 overrides
  public function create(): Factory|View
  {
      return view(view: 'auth.login');
  }

  /**
   * Handle an incoming authentication request.
   *
   * @param \App\Http\Requests\Auth\LoginRequest $request
   * @return \Illuminate\Http\RedirectResponse
   */
  1 reference | 0 overrides
  public function store(LoginRequest $request): mixed|RedirectResponse
  {
      $request->authenticate();

      $request->session()->regenerate();

      return redirect()->intended(default: RouteServiceProvider::HOME);
  }

  /**
   * Destroy an authenticated session.
   *
   * @param \Illuminate\Http\Request $request
   * @return \Illuminate\Http\RedirectResponse
   */
  1 reference | 0 overrides
  public function destroy(Request $request): Redirector|RedirectResponse
  {
      Auth::guard(name: 'web')->logout();

      $request->session()->invalidate();

      $request->session()->regenerateToken();

      return redirect(to: '/');
  }
}

```

Рисунок 4.16 – Контролер обробки входу та виходу з кабінету

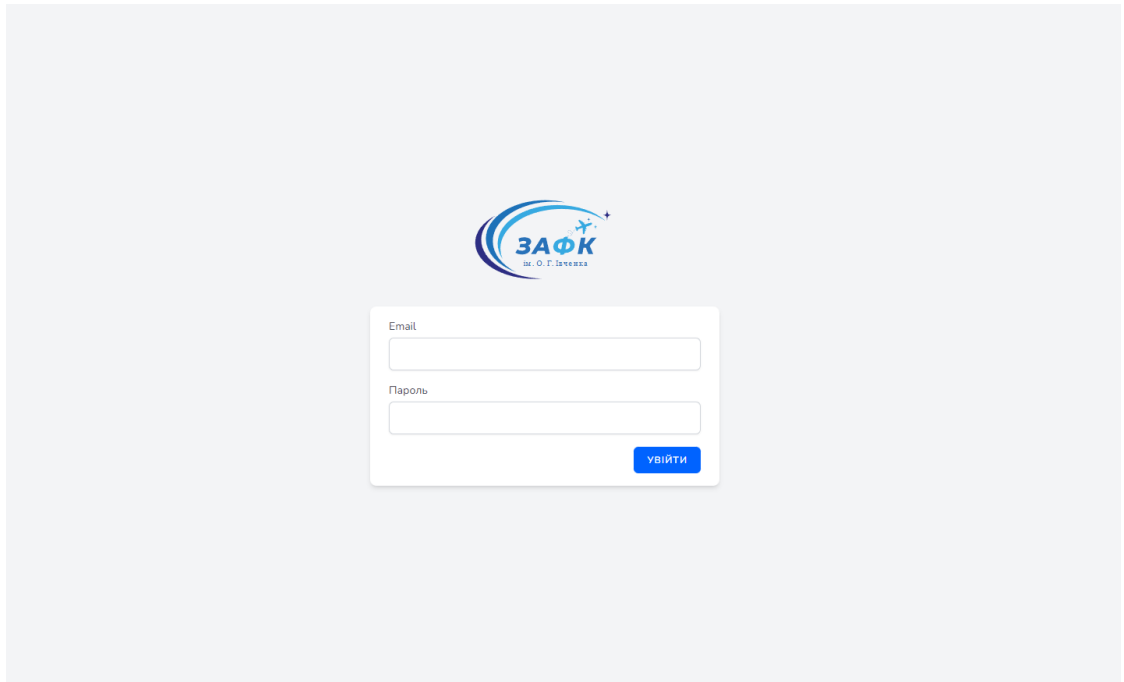


Рисунок 4.17 – Сторінка входу

Після успішної авторизації, ми потрапляємо в панель адміністратора на головну сторінку (рис. 4.18). Вона складається з головного меню зверху, а також зміна основної інформації про розклад тощо.

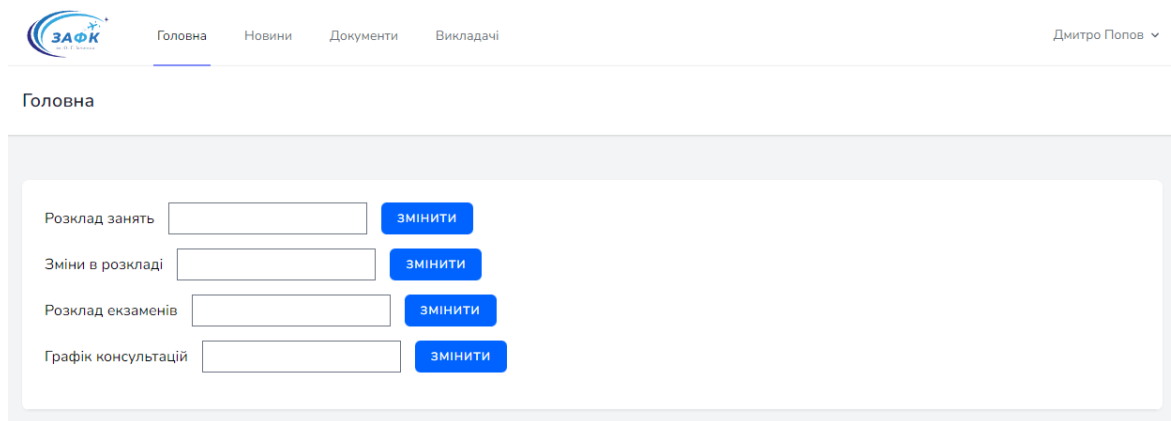


Рисунок 4.18 – Головна сторінка панелі адміністратора

Також за допомогою консолі створюємо 2 контролера – AdminController (Для обробки даних з адмін. панелі) та UkrController (Для обробки даних клієнтської частини сайту). Та після цього обов'язково підключаємо до файлу web.php для формування routes.

```
$ php artisan make:controller AdminController
Controller created successfully.
```

Рисунок 4.19 – Створення AdminController

```
$ php artisan make:controller UkrController
```

Рисунок 4.20 – Створення UkrController

```
use App\Http\Controllers\UkrController;
use App\Http\Controllers\AdminController;
```

Рисунок 4.21 – Підключення контролерів до web.php

Для реалізації зміни основних даних з головної сторінки щодо розкладу, екзаменів тощо. Необхідно у файлі головної сторінки додати 4 форми для редагування кожного пункту.

```
<form action="{{ route(name: 'admin.editschedule') }}" method="POST" class="flex items-center" style="margin-bottom: 15px;">
  @csrf
  <label for="lessonstime">Розклад занять</label>
  <input value="{{ {{ $schedule->link }}" type="text" name="lessonstime" id="lessonstime" class="block min-w-0 grow py-1 pr-3 pl-1 text-base text-gray-900 placeholder:text-gray-400 focus:outline-none sm:text-sm/6" style="margin: 0 15px;">
  <button class="inline-flex items-center px-4 py-2 border border-transparent rounded-md font-semibold text-xs text-white uppercase tracking-widest hover:bg-gray-700 active:bg-gray-900 focus:outline-none focus:border-gray-900 focus:ring ring-gray-300 disabled:opacity-25 transition ease-in-out duration-150" style="background-color: #0063ff; color: white;">Змінити</button>
</form>

<form action="{{ route(name: 'admin.editchanges') }}" method="POST" class="flex items-center" style="margin-bottom: 15px;">
  @csrf
  <label for="lessonstime">Зміни в розкладі</label>
  <input value="{{ {{ $changes->link }}" type="text" name="changetime" id="lessonstime" class="block min-w-0 grow py-1 pr-3 pl-1 text-base text-gray-900 placeholder:text-gray-400 focus:outline-none sm:text-sm/6" style="margin: 0 15px;">
  <button class="inline-flex items-center px-4 py-2 border border-transparent rounded-md font-semibold text-xs text-white uppercase tracking-widest hover:bg-gray-700 active:bg-gray-900 focus:outline-none focus:border-gray-900 focus:ring ring-gray-300 disabled:opacity-25 transition ease-in-out duration-150" style="background-color: #0063ff; color: white;">Змінити</button>
</form>

<form action="{{ route(name: 'admin.editexams') }}" method="POST" class="flex items-center" style="margin-bottom: 15px;">
  @csrf
  <label for="lessonstime">Розклад екзаменів</label>
  <input value="{{ {{ $exams->link }}" type="text" name="exams" id="lessonstime" class="block min-w-0 grow py-1 pr-3 pl-1 text-base text-gray-900 placeholder:text-gray-400 focus:outline-none sm:text-sm/6" style="margin: 0 15px;">
  <button class="inline-flex items-center px-4 py-2 border border-transparent rounded-md font-semibold text-xs text-white uppercase tracking-widest hover:bg-gray-700 active:bg-gray-900 focus:outline-none focus:border-gray-900 focus:ring ring-gray-300 disabled:opacity-25 transition ease-in-out duration-150" style="background-color: #0063ff; color: white;">Змінити</button>
</form>

<form action="{{ route(name: 'admin.editconsults') }}" method="POST" class="flex items-center" style="margin-bottom: 15px;">
  @csrf
  <label for="lessonstime">Графік консультацій</label>
  <input value="{{ {{ $consults->link }}" type="text" name="consults" id="lessonstime" class="block min-w-0 grow py-1 pr-3 pl-1 text-base text-gray-900 placeholder:text-gray-400 focus:outline-none sm:text-sm/6" style="margin: 0 15px;">
  <button class="inline-flex items-center px-4 py-2 border border-transparent rounded-md font-semibold text-xs text-white uppercase tracking-widest hover:bg-gray-700 active:bg-gray-900 focus:outline-none focus:border-gray-900 focus:ring ring-gray-300 disabled:opacity-25 transition ease-in-out duration-150" style="background-color: #0063ff; color: white;">Змінити</button>
</form>
```

Рисунок 4.22 – Код головної сторінки панелі адміністратора

Кожна форма вводу (тег Form) має свій шлях до обробнику, що записано у налаштуванні action, а також метод POST. В середині створено поле вводу (input), де адміністратор може вписати посилання на потрібну інформацію.

```
Route::post(uri: '/edit-schedule', action: [AdminController::class, 'editchedule'])->name(name: 'admin.editschedule');
Route::post(uri: '/edit-changes', action: [AdminController::class, 'editchanges'])->name(name: 'admin.editchanges');
Route::post(uri: '/edit-exams', action: [AdminController::class, 'editexams'])->name(name: 'admin.editexams');
Route::post(uri: '/edit-consults', action: [AdminController::class, 'editconsults'])->name(name: 'admin.editconsults');
```

Рисунок 4.23 – Маршрут до кожного файлу обробника

Оскільки маршрут являє собою посередника між сторінкою та контролером, де відбувається обробка даних – необхідно перейти до AdminController’у.

У контролері (рис 4.24) ми бачимо декілька важливих функцій, а саме:

```
class AdminController extends Controller
{
    1 reference | 0 overrides
    public function dashboard(): Factory|View{
        $schedule = Information::where(column: 'name', operator: 'Розклад занять')->first();
        $changes = Information::where(column: 'name', operator: 'Зміни в розкладі')->first();
        $exams = Information::where(column: 'name', operator: 'Розклад екзаменів')->first();
        $consults = Information::where(column: 'name', operator: 'Графік консультацій')->first();
        return view(view: 'dashboard', data: ['schedule'=> $schedule, 'changes'=> $changes, 'exams'=> $exams, 'consults'=> $consults]);
    }

    1 reference | 0 overrides
    public function editchedule(Request $request): RedirectResponse{
        $info = Information::where(column: 'name', operator: 'Розклад занять')->first();
        $link = $request->lessonstime;
        $info->name = "Розклад занять";
        $info->link = $link;
        $info->save();
        return Redirect::to(path: '/dashboard');
    }

    1 reference | 0 overrides
    public function editchanges(Request $request): RedirectResponse{
        $info = Information::where(column: 'name', operator: 'Зміни в розкладі')->first();
        $link = $request->changetime;
        $info->name = "Зміни в розкладі";
        $info->link = $link;
        $info->save();
        return Redirect::to(path: '/dashboard');
    }

    1 reference | 0 overrides
    public function editexams(Request $request): RedirectResponse{
        $info = Information::where(column: 'name', operator: 'Розклад екзаменів')->first();
        $link = $request->exams;
        $info->name = "Розклад екзаменів";
        $info->link = $link;
        $info->save();
        return Redirect::to(path: '/dashboard');
    }

    1 reference | 0 overrides
    public function editconsults(Request $request): RedirectResponse{
        $info = Information::where(column: 'name', operator: 'Графік консультацій')->first();
        $link = $request->consults;
        $info->name = "Графік консультацій";
        $info->link = $link;
        $info->save();
        return Redirect::to(path: '/dashboard');
    }
}
```

Рисунок 4.24 – AdminController обробки головної сторінки

dashboard() – за допомогою моделі Information, створено 4 змінні, які зберігають інформацію про – Розклад занять, Зміни в розкладі, Розклад

екзаменів, графік консультацій. А після повертає сторінку (view) з усіма даними (data), які йдуть у вигляді масиву. Для виводу даних на сторінці через масив використовується наступна конструкція:

```
<input value="{{ $schedule->link }}"
```

Рисунок 4.25 – Приклад виводу інформації після return

Де бачимо – масив \$schedule та за допомогою стрілочки йде звертання до стовпчику link з таблиці Бази даних.

editshedule() – функція, що є обробником першої форми для зміни розкладу. Створено змінна info, яка за допомогою моделі Information отримує інформацію про строку з розкладом занять за допомогою методу where(). За допомогою змінної link отримується інформація з поля вводу під назвою lesson, з цим допомагає спеціальна конструкція Request \$request, яка вбудована в Laravel. Після йде звертання до стовпчиків name та link (\$info->name та \$info->link відповідно), де йде оновлення існуючої інформації та за допомогою \$info->save() – зберігаємо нову інформацію та робимо Редірект (перехід користувача на іншу сторінку), а саме на головну сторінку панелі адміністратора.

editchanges() – функція, яка обробляє дані з другої форми, яка відповідає за зміни в розкладі, працює за таким самим прикладом як і функція editshedule.

editexams() – функція, що змінює розклад екзаменів за допомогою третьої форми, працює аналогічно до функції editshedule.

editconsults() – функція, яка оброблює дані з останньої форми головної сторінки про графік консультації, реалізовано так само, як й editshedule.

```

9 references | 0 implementations
class Information extends Model
{
    use HasFactory;
    0 references
    protected $table = 'informations';
    0 references
    protected $fillable = ['name', 'link'];
    0 references
    public $timestamps = false;
}

```

Рисунок 4.26 – Модель Information

Файл, що зберігає інформацію про модель Information, тут зберігається декілька строк, а саме:

Use HasFactory – дозволяє створювати Factory – об’єкти для тестування.

protected \$table = 'informations' – уточнює з якою таблицею працює модель.

protected \$fillable = ['name', 'link'] – вказуються назви стовпчиків.

public \$timestamps = false – вимикає режим timestamps (стовпчики created_at, updated_at).

4.2.2 Створення системи новин

Для реалізації системи додавання, перегляду, редагування та видалення новин, було використано систему CRUD (Create, Read, Update, Delete).

Створено модель з міграцією:

```

$ php artisan make:model News -m
Model created successfully.
Created Migration: 2025_05_22_222000_create_news_table

```

Рисунок 4.27 – Створення моделі та міграції News

Також створено спеціальний ресурсний контролер (рис. 4.28), який допомагає працювати з системою CRUD, а також він був підключений до списку маршрутів у файлі web.php (рис. 4.29).

```
$ php artisan make:controller NewsController --resource --model=News
Controller created successfully.
```

Рисунок 4.28 – Створення ресурсного контролеру NewsController

```
Route::resource(name: '/news', controller: NewsController::class)->middleware(middleware: ['auth']);
```

Рисунок 4.29 – Підключення контролеру до системи маршрутизації

Оскільки було створено не звичайний контролер, а ресурсний – його базова структура відрізняється. При створенні просто контролеру він залишається пустим та не має жодних методів в порівнянні з Ресурсним, де за замовчування створюються додаткові методи – index, create, store, edit, update, show, destroy.

В першу чергу – створено сторінку з новинами. Структура коду, контролеру та візуальної частини наведено нижче:

```
<!-- app-layout -->
<!-- slot name="header" -->
<div class="flex justify-between items-center">
  <h2 class="font-semibold text-xl text-gray-800 leading-tight">
    {{ __key: 'Новини' }}
  </h2>
  <a href="/news/create" class="inline-flex items-center px-4 py-2 border border-transparent rounded-md font-semibold text-xs text-white uppercase tracking-widest hover:bg-gray-700 active:bg-gray-900 focus:outline-none focus:border-gray-900 focus:ring ring-gray-300 disabled:opacity-25 transition ease-in-out duration-150" style="background-color: #28a745;" >Додати новину</a>
</div>
</x-slot>
<div class="py-12">
  <div class="max-w-7xl mx-auto sm:px-6 lg:px-8">
    <div class="bg-white overflow-hidden shadow-sm sm:rounded-lg">
      <div class="p-6 bg-white border-b border-gray-200">
        @foreach ($posts as $post)
          <div class="flex justify-between items-center" style="margin-bottom: 15px">
            <div>
              <p>{{ $post->id }}</p>
            </div>
            <div>
              <p><a href="{{ route('news.show', parameters: ['news' => $post]) }}">{{ $post->title }}</a></p>
            </div>
            <div class="flex">
              <a href="{{ route('news.edit', parameters: ['news' => $post]) }}" class="inline-flex items-center px-4 py-2 border border-transparent rounded-md font-semibold text-xs text-white uppercase tracking-widest hover:bg-gray-700 active:bg-gray-900 focus:outline-none focus:border-gray-900 focus:ring ring-gray-300 disabled:opacity-25 transition ease-in-out duration-150" style="background-color: #007bff; margin-right: 10px;" >Редагувати</a>
              <form action="{{ route('news.destroy', parameters: ['news' => $post->id]) }}" method="POST">
                @csrf
                @method("DELETE")
                <button class="inline-flex items-center px-4 py-2 border border-transparent rounded-md font-semibold text-xs text-white uppercase tracking-widest hover:bg-gray-700 active:bg-gray-900 focus:outline-none focus:border-gray-900 focus:ring ring-gray-300 disabled:opacity-25 transition ease-in-out duration-150" style="background-color: #dc3545;" >Видалити</button>
              </form>
            </div>
          </div>
        @endforeach
      </div>
    </div>
  </div>
</x-app-layout>
```

Рисунок 4.30 – Структура сторінки index news

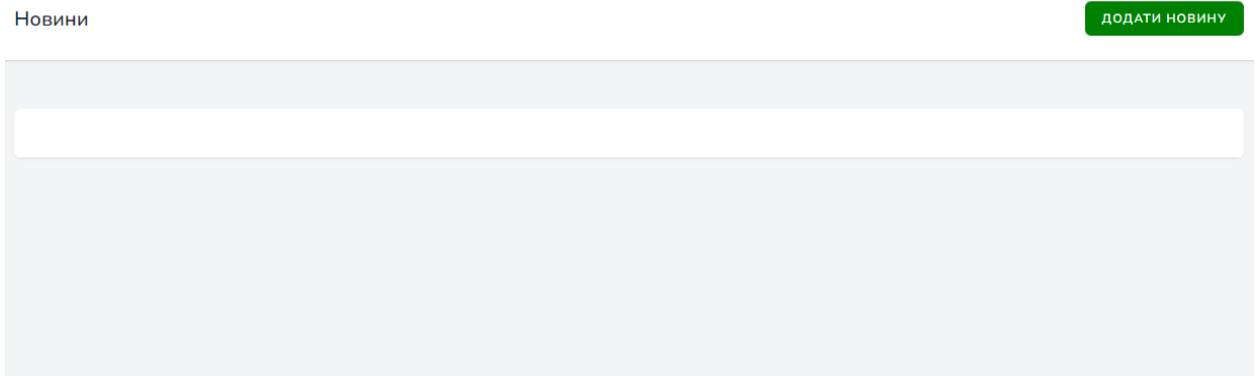


Рисунок 4.31 – Візуальна структура сторінки index news

```
public function index(): Factory|View
{
    $posts = News::all();
    return view('lang.ukr.news.index', data: ['posts'=> $posts]);
}
```

Рисунок 4.32 – Структура методу index в NewsController

Метод `index` – відповідає за головну сторінку ресурсу, де будуть відображатися всі новини, що знаходяться в Базі Даних.

Змінна `$posts` – за допомогою моделі `News::all()`, отримує все, що знаходиться в таблиці `News` (Аналог команди `SELECT * FROM` з `SQL`). Після цього відбувається повернення сторінки та її даних, для подальшої реалізації виводу.

В `Blade`-шаблоні, для перебору масиву `Posts` на кожен окремий `post`, який було отримано з Контролеру було використано метод `foreach`, підготовлений шаблон та за допомогою, наприклад, `{{ $post->titile }}` вже відображає інформацію на сайті.

Для створення нової новини було створено нову сторінку (`create`), структура коду, контролеру та візуальної частини наведено нижче:

```

<div class="p-6 bg-white border-b border-gray-200">
  <form action="{ route(name: 'news.store') }}" method="POST" class="text-center" style="margin-bottom: 15px;" enctype="multipart/form-data">
    @csrf
    <div style="margin-bottom: 10px;">
      <label for="title">Назва новини</label>
      <input value="" type="text" name="title" id="title" class="block min-w-0 grow py-1 pr-3 pl-1 text-base text-gray-900 placeholder:text-gray-400 focus:outline-none sm:text-sm/6" style="margin: 0 auto; width: 50%;">
    </div>
    <div style="margin-bottom: 10px;">
      <label for="description">Коротки опис</label>
      <textarea value="" type="text" name="description" id="description" class="block min-w-0 grow py-1 pr-3 pl-1 text-base text-gray-900 placeholder:text-gray-400 focus:outline-none sm:text-sm/6" style="margin: 0 auto; width: 50%;"></textarea>
    </div>
    <div style="margin-bottom: 10px;">
      <label for="text">Текст новини</label>
      <textarea value="" type="text" name="text" id="text" class="block min-w-0 grow py-1 pr-3 pl-1 text-base text-gray-900 placeholder:text-gray-400 focus:outline-none sm:text-sm/6" style="margin: 0 auto; width: 50%;"></textarea>
    </div>
    <div style="margin-bottom: 10px;">
      <label for="date">Дата новини</label>
      <input value="" type="text" name="date" id="date" class="block min-w-0 grow py-1 pr-3 pl-1 text-base text-gray-900 placeholder:text-gray-400 focus:outline-none sm:text-sm/6" style="margin: 0 auto; width: 50%;">
    </div>
    <div style="margin-bottom: 10px;">
      <label for="photo">Фото новини</label>
      <input value="" type="file" name="photo" id="photo" class="block min-w-0 grow py-1 pr-3 pl-1 text-base text-gray-900 placeholder:text-gray-400 focus:outline-none sm:text-sm/6" style="margin: 0 auto; width: 50%;">
    </div>
    <div style="margin-bottom: 10px;">
      <label for="slug">Назва URL (slug)</label>
      <input value="" type="text" name="slug" id="slug" class="block min-w-0 grow py-1 pr-3 pl-1 text-base text-gray-900 placeholder:text-gray-400 focus:outline-none sm:text-sm/6" style="margin: 0 auto; width: 50%;">
    </div>
    <button class="inline-flex items-center px-4 py-2 border border-transparent rounded-md font-semibold text-xs text-white uppercase tracking-widest hover:bg-gray-700 active:bg-gray-900 focus:outline-none focus:border-gray-900 focus:ring ring-gray-300 disabled:opacity-25 transition ease-in-out duration-150" style="background-color: #28a745; color: white; margin-top: 10px; width: 50%;">
      Додати новину</button>
  </form>

```

Рисунок 4.33 – Структура коду сторінки create

Додати новину

Назва новини

Коротки опис

Текст новини

Дата новини

Фото новини

Виберіть файл | Файл не вибрано

Назва URL (slug)

ДОДАТИ НОВИНУ

Рисунок 4.34 – Візуальна структура сторінки Create

Для створення нової новини на сайті, спираючись на архітектуру таблиці бази даних з новинами, додано 4 поля вводу (input) , куди записується

інформація про назву новини, дату новини, фото новини та slug (Назва URL-адреси), 2 textarea для опису та основної частини новини та кнопка додавання новини.

```

public function store(Request $request): RedirectResponse
{
    $news = new News();
    $news->title = $request->title;
    $news->description = $request->description;
    $news->text = $request->text;
    $news->date = $request->date;
    $news->slug = $request->slug;

    if ($request->hasFile(key: 'photo')) {
        $file = $request->file(key: 'photo');

        $filename = $file->getClientOriginalName();

        $file->move(directory: public_path(path: '/img/news'), name: $filename);
        $news->photo = $filename;
    }else{
        $filename = 'noimage.png';
        $news->photo = $filename;
    }
    $news->save();
    return Redirect::to(path: '/news');
}

```

Рисунок 4.35 – Структура методу store в NewsController

При створенні нової новини, використано метод store(), який знаходиться в ресурсному контролері NewsController. Змінна \$news створює нову строку в таблиці.

Використовуючи \$request – отримується інформація з input, textarea, що знаходяться в form.

Для реалізації системи додавання нової фотографії необхідно зробити додаткову умову, де йде перевірка, чи додав адміністратор нове фото, якщо так – то фото буде переміщено в спеціально папку, вказану в методі move(), та дізнаємось назву файлу, за допомогою \$file->getClientOriginalName(). Якщо

адміністратор не додав нове фото – то в назву й надалі в таблицю буде записано «noimage.png».

Після обробки полів вводу, відбувається збереження новини та повернення користувача на головну сторінку з новинами.

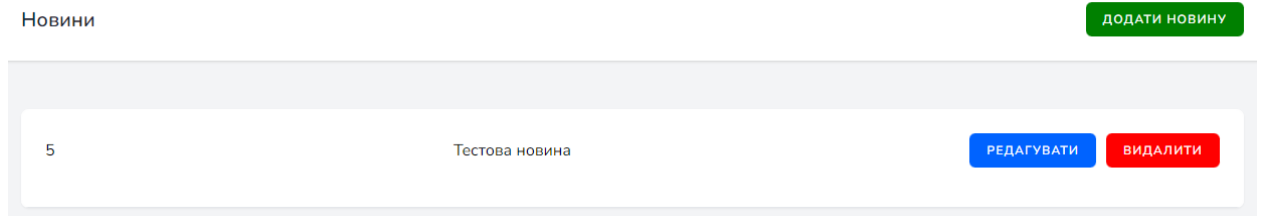


Рисунок 4.36 – Результат додавання нової сторінки

Для видалення непотрібної або застарілої новини, необхідно створити форму, яка буде мати тільки кнопку й саме головне, підключити правильне посилання на маршрут з видалення, де буде передано id новини та вказано спеціальний метод «Delete», якого потребує маршрутизація Laravel.

```
<form action="{ route(name: 'news.destroy', parameters: ['news' => $post->id]) }" method="POST">
  @csrf
  @method("DELETE")
  <button class="inline-flex items-center px-4 py-2 border border-transparent rounded-md font-semibold text-xs text-white uppercase tracking-widest hover:bg-gray-700 active:bg-gray-900 focus:outline-none focus:border-gray-900 focus:ring ring-gray-300 disabled:opacity-25 transition ease-in-out duration-150" style="background-color: red;" >Видалити</button>
</form>
```

Рисунок 4.37 – Форма видалення новини

```
public function destroy(News $news): RedirectResponse
{
    // dd($news);
    $news->delete();

    return Redirect::to(path: '/news');
}
```

Рисунок 4.38 – Метод destroy для видалення

Не менш важливою системою в CRUD – редагування, оскільки будь-хто може зробити помилку будь в чому й потрібно, щоб було де її виправити, для

цього використовуються `edit` – для створення сторінки та `update` – для оновлення.

```
<a href="{{ route(name: 'news.edit', parameters: ['news' => $post]) }}" class="inline-flex items-center px-4 py-2 border border-transparent rounded-md font-semibold text-xs text-white uppercase tracking-widest hover:bg-gray-700 active:bg-gray-900 focus:outline-none focus:border-gray-900 focus:ring ring-gray-300 disabled:opacity-25 transition ease-in-out duration-150" style="background-color: #0063ff; margin-right: 10px;" >Редагувати</a>
```

Рисунок 4.39 – Посилання на сторінку редагування новини

Використовуючи посилання та атрибут `href`, було вказано шлях до `news.edit` (сторінка для редагування) з `id` новини.

```
public function edit(News $news): Factory|View
{
    $news = News::where(column: 'id', operator: $news->id)->first();
    return view(view: 'lang.ukr.news.edit', data: ["news" => $news]);
}
```

Рисунок 4.40 – Метод `edit` в `NewsController`

Змінна `$news` – зберігає новину, яку отримав метод `edit` за допомогою `id` й після повернення на сторінку з інформацією про новину.

Для відображення інформації про новини для полів вводу додано атрибут `value`, в який додається Laravel код з інформацією з таблиці новин за допомогою спеціальних дужок `{{}}`.

```

<form action="{{ route(name: 'news.update', parameters: ['news' => $news]) }}" method="POST" class="text-center"
style="margin-bottom: 15px;">
  @csrf
  @method('PUT')
  <div style="margin-bottom: 10px;">
    <label for="title">Назва новини</label>
    <input value="{{ $news->title }}" type="text" name="title" id="title" class="block min-w-0 grow py-1 pr-3 pl-1 text-base
text-gray-900 placeholder:text-gray-400 focus:outline-none sm:text-sm/6" style="margin: 0 auto; width: 50%;">
  </div>

  <div style="margin-bottom: 10px;">
    <label for="description">Коротки опис</label>
    <textarea type="text" name="description" id="description" class="block min-w-0 grow py-1 pr-3 pl-1 text-base text-gray-900
placeholder:text-gray-400 focus:outline-none sm:text-sm/6" style="margin: 0 auto; width: 50%;">{{ $news->description }}</
textarea>
  </div>

  <div style="margin-bottom: 10px;">
    <label for="text">Текст новини</label>
    <textarea type="text" name="text" id="text" class="block min-w-0 grow py-1 pr-3 pl-1 text-base text-gray-900
placeholder:text-gray-400 focus:outline-none sm:text-sm/6" style="margin: 0 auto; width: 50%;">{{ $news->text }}</textarea>
  </div>

  <div style="margin-bottom: 10px;">
    <label for="date">Дата новини</label>
    <input value="{{ $news->date }}" type="text" name="date" id="date" class="block min-w-0 grow py-1 pr-3 pl-1 text-base
text-gray-900 placeholder:text-gray-400 focus:outline-none sm:text-sm/6" style="margin: 0 auto; width: 50%;">
  </div>

  <div style="margin-bottom: 10px;">
    <label for="slug">Назва URL (slug)</label>
    <input value="{{ $news->slug }}" type="text" name="slug" id="slug" class="block min-w-0 grow py-1 pr-3 pl-1 text-base
text-gray-900 placeholder:text-gray-400 focus:outline-none sm:text-sm/6" style="margin: 0 auto; width: 50%;">
  </div>

  <button class="inline-flex items-center px-4 py-2 border border-transparent rounded-md font-semibold text-xs text-white uppercase
tracking-widest hover:bg-gray-700 active:bg-gray-900 focus:outline-none focus:border-gray-900 focus:ring ring-gray-300
disabled:opacity-25 transition ease-in-out duration-150" style="background-color: #0063ff;">Редагувати новину</button>
</form>

```

Рисунок 4.41 – Структура сторінки редагування новини (edit)

Редагувати новину

Назва новини

Коротки опис

Текст новини

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

Дата новини

Назва URL (slug)

Рисунок 4.42 – Сторінка з редагуванням новини

4.2.3 Створення клієнтської частини

Для створення клієнтської частини сайту було додано окремий контролер UkrController. Для головної сторінки створено окремий метод index (рис 4.43), в якому створено змінну \$posts, де код звертається до моделі News, в якій отримуємо 3 новини для гарного відображення на сайті, а також вони відсортуються за датою додавання починаючи з нових.

```
1 reference | 0 overrides
public function index(): Factory|View{
    $posts = News::orderBy(column: 'created_at', direction: 'desc')->paginate(perPage: 3);
    return view(view: 'lang.ukr.index', data: [
        'posts'=>$posts
    ]);
}
```

Рисунок 4.43 – Структура головної сторінки в контролері

```
Route::get(uri: '/', action: [UkrController::class, 'index']);
```

Рисунок 4.44 – Маршрут головної сторінки

Далі необхідно створити код та візуальну складову головної сторінки. Оскільки кожна сторінка сайту має однакову верхню та нижні частини – створено спеціальний шаблон, який буде підключатися до кожної сторінки.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link href="https://fonts.googleapis.com/css2?family=Montserrat:wght@400;500;600;700;900&display=swap" rel="stylesheet">
    <link rel="stylesheet" href="/css/bootstrap.min.css">
    <link rel="stylesheet" href="/css/slick.css">
    <link rel="stylesheet" href="/css/slick-theme.css">
    <link rel="stylesheet" href="/css/style.css">
    <title>@yield(section: 'title')</title>
</head>
<body>
```

Рисунок 4.45 – тег head в шаблоні templates

В тегу Head зберігається системна інформація, яку не видно користувачу, про необхідні підключені css/less файли. Окрім стандартних файлів було підключено бібліотеку slick-slider для реалізації слайдеру на головній сторінці.

@yield('title') – місце для контенту, щоб на кожній сторінці підставити власну назву сторінки.

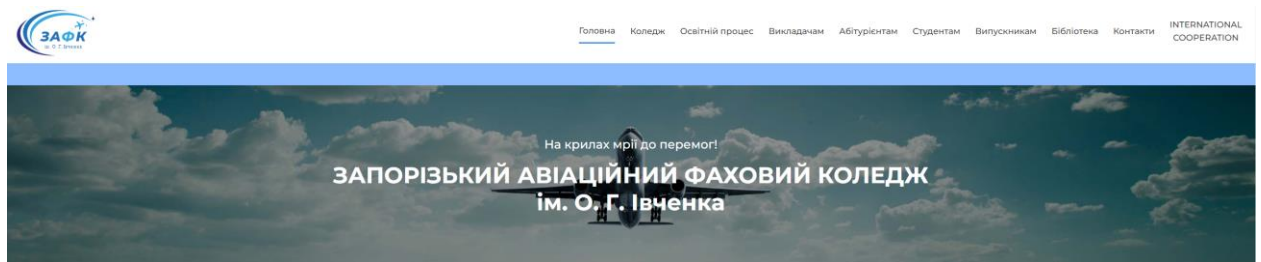


Рисунок 4.46 – Шаблон Header та Section для кожної сторінки

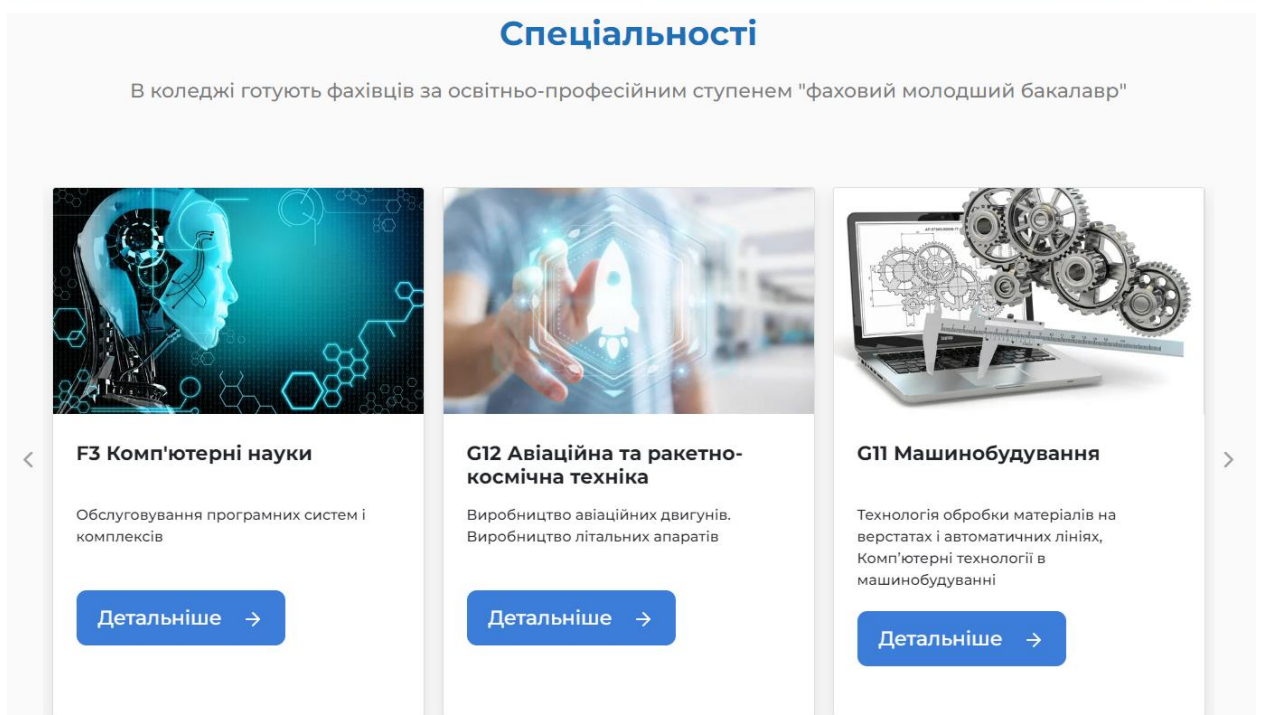



Рисунок 4.47 – Секція «Спеціальності»

При створенні цієї секції було використано бібліотеку Slick Slider для відображення всіх спеціальностей поступово, слайдер гортається

автоматично, але якщо необхідно прогорнути далі або назад, для цього зліва та справа є спеціальні стрілочки.


Новини



📅 травень 2025

Сьогодні, 19 травня, відбулась онлайн-зустріч здобувачів освіти з інспектором сектору ювенальної превенції Запорізького РУП ГУНП в Запорізькій області, лейтенантом поліції Зубачовою О.О..


Зустріч пройшла у форматі бесіди-діалогу. Студенти активно цікавились проявами вербування, Ольга Олександрівна доступно і грамотно пояснила про відповідальність неповнолітніх щодо участі у терористичної діяльності сьогодні.



📅 травень 2025

Вперше в Україні стартував Національний тиждень безбар'єрності. Він триватиме протягом 19-25 травня. До нього долучається і Запорізька область.

Національний тиждень безбар'єрності – частина державної стратегії зі створення безбар'єрного простору в Україні, ініційованої Першою леді України Оленою Зеленською.



📅 травень 2025

18.05.2025 р. - День пам'яті жертв політичних репресій

Відзначається згідно з Указом Президента від 21 травня 2007 року на вшанування десятків тисяч людей, розстріляних комуністичним режимом, сотень тисяч кинутих у в'язниці й табори, на заслання чи примусове психіатричне лікування. Кількість цих мучеників Україні досі неможливо підрахувати.

[Всі новини →](#)

Рисунок 4.48 – Секція «Новини»

Відображення новин було реалізовано за допомогою контролера, а також за допомогою методу перебору масивів `foreach`

```

@foreach($posts as $post)

<div class="col-12 news">
  <a href="{{route(name: 'show',parameters: $post->slug)}}">
    <div class="news">
      <div class="news__img">
        
      </div>
      <div class="news__text">
        <p class="card-text"><i data-feather="calendar"></i><span>{{ $post->date }}</span></p>
        <h5 class="card-title">{{ $post->title }}</h5>
        <p class="news__desc">{!! $post->description !!</p>
      </div>
    </div>
  </a>
</div>

<hr style="background-color: #4B91F1; height: 3px;" class="mt-3">

@endforeach

<div class="student-btn text-center">
  <a href="/news" class="more-btn"><span>Всі новини</span><i data-feather="arrow-right"></i></a>
</div>

```


Рисунок 4.49 – Структура коду секції «Новини»

Також додано посилання на всі новини, щоб побачити все, що відбувалося в коледжі за всі роки.

На крилах мрії до перемог!


ЗАПОРІЗЬКИЙ АВІАЦІЙНИЙ ФАХОВИЙ КОЛЕДЖ ім. О. Г. Івченка

Новини




📅 травень 2025

Сьогодні, 19 травня, відбулась онлайн-зустріч здобувачів освіти з інспектором сектору ювенальної превенції Запорізького РУП ГУНП в Запорізькій області, лейтенантом поліції Зубачовою О.О..




📅 травень 2025

Вперше в Україні стартував Національний тиждень безбар'єрності. Він триватиме протягом 19-25 травня. До нього долучається і Запорізька область.




📅 травень 2025

18.05.2025 р. - День пам'яті жертв політичних репресій




ІСТОРІЯ КРИМУ

КРИМ - ЦЕ УКРАЇНА



ДЕНЬ ЄВРОПИ



📅 травень 2025

8 травня Україна відзначає

Рисунок 4.50 – Сторінка «всі новини»

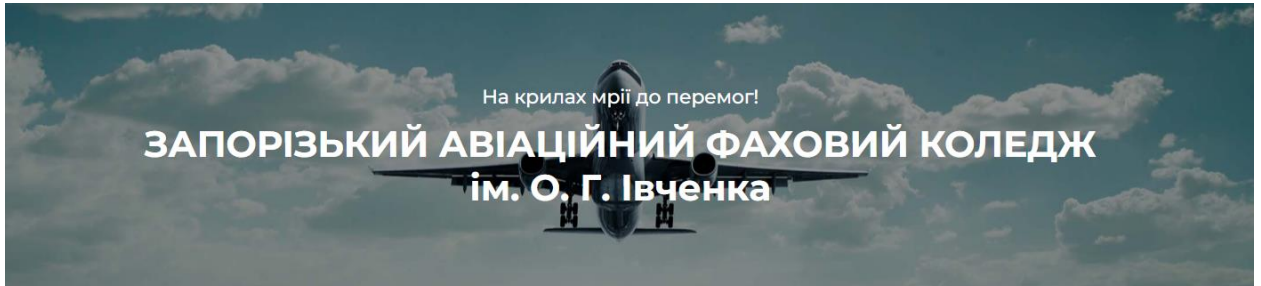
```
public function news(): Factory|View{
    $posts = News::orderBy(column: 'created_at', direction: 'desc')->get();
    return view(view: 'lang.ukr.posts.index', data: [
        'posts'=>$posts
    ]);
}
```

Рисунок 4.51 – Структура методу News в UkrController

Для отримання всіх новин та передачі їх до сторінки «Всі новини» створено змінну \$posts, яка звертається до моделі з новинами, сортує від самих

нових та отримує їх в дану змінну. А далі повертає користувача на сторінку з новинами та передає змінну posts.

При натисканні на потрібну новину відкривається окрема сторінка, де відображається повний текст новини, фото та назва.



18.05.2025 р. - День пам'яті жертв політичних репресій



Відзначається згідно з Указом Президента від 21 травня 2007 року на вшанування десятків тисяч людей, розстріляних комуністичним режимом, сотень тисяч кинутих у в'язниці й табори, на заслання чи примусове психіатричне лікування. Кількість цих мучеників Україні досі неможливо підрахувати.

[Детальніше](#)

Рисунок 4.52 – Приклад новини

```
public function show($slug): Factory|View{
    $post = Post::where(column: 'slug', operator: $slug)->first();
    return view(view: 'lang.ukr.posts.show', data: [
        'post'=>$post
    ]);
}
```

Рисунок 4.53 – Структура методу show в UkrController

Змінна \$post отримує новину по унікальній Url-назві (Slug) й передає всю інформацію про неї до сторінки Show, яка працює як шаблон для виводу новини (рис. 4.54).

```
@extends(view: "layouts.templates.ukr.template")
@section(section: 'title', content: "{$post->title}")

@section(section: 'content')
  <section class="post__show">
    <div class="container">
      <h1 class="main-h1">{{$post->title}}</h1>
      <div class="text-center">
        
      </div>

      {!! $post->text !!}
    </div>
  </section>
@endsection
```

Рисунок 4.54 – Шаблон-файл для новини

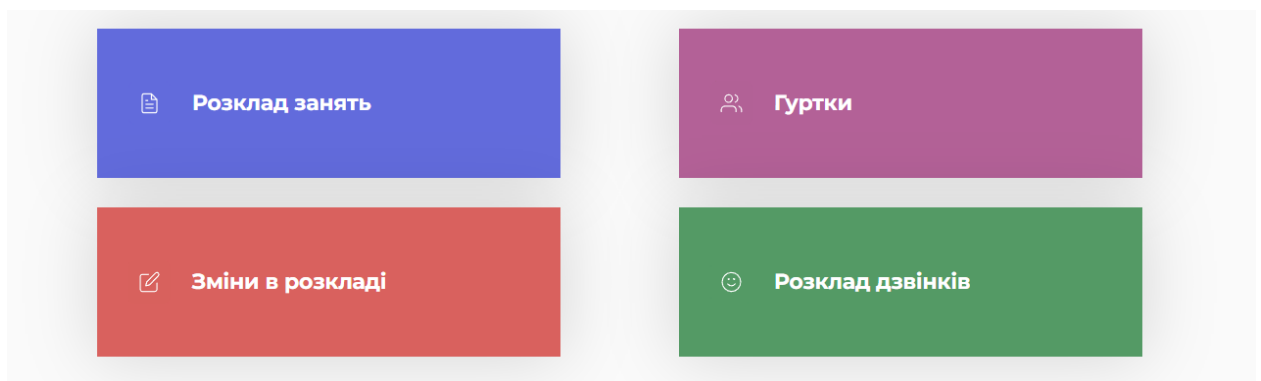


Рисунок 4.55 – Секція «Основна інформація»

Декілька причин навчатися з нами

Безкоштовне навчання

Навчання у коледжі повністю безоплатне, за виключенням окремих спеціальностей

Висока стипендія

Виплачується стипендія. Регулярні виплати без затримок

Спортивне життя

Студенти мають всі умови для здорового активного дозвілля й змістовного та затишного відпочинку

Гарний старт

З дипломом молодшого спеціаліста та фахового молодшого бакалавра продовжуйте здобувати Вищу освіту.

Робота

Випускники забезпечуються робочими місцями.

Досвідчені педагоги

Освітній процес забезпечується висококваліфікованим викладацьким колективом та кураторським супроводом протягом усіх років навчання

Відповідальність

Відповідальне ставлення до навчання, самостійне планування навчального графіка, розподіл грошових витрат швидко перетворює безпорадну дитину на дорослого.

Виробнича практика

Практичне навчання студенти проходять на кращих підприємствах міста.

Матеріальна підтримка

Соціально вразливі категорії забезпечуються соціальними стипендіями, а активні студенти мають можливість отримати матеріальне заохочення

Рисунок 4.56 – Секція «Декілька причин навчатися з нами»

Залишилися запитання?

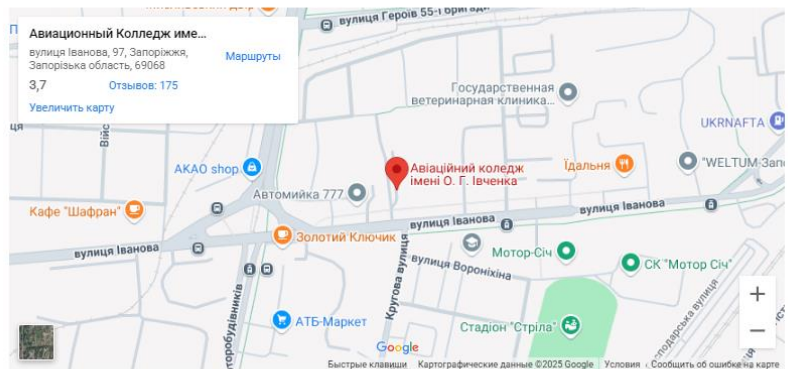
Надішліть нам листа на нашу пошту з цікавим для вас питанням.

Ім'я

Ел.пошта

Повідомлення

Надіслати



Адреса коледжу: 69068, м.Запоріжжя, вул. Іванова 97Б

Рисунок 4.57 – Секція «Контактна інформація»

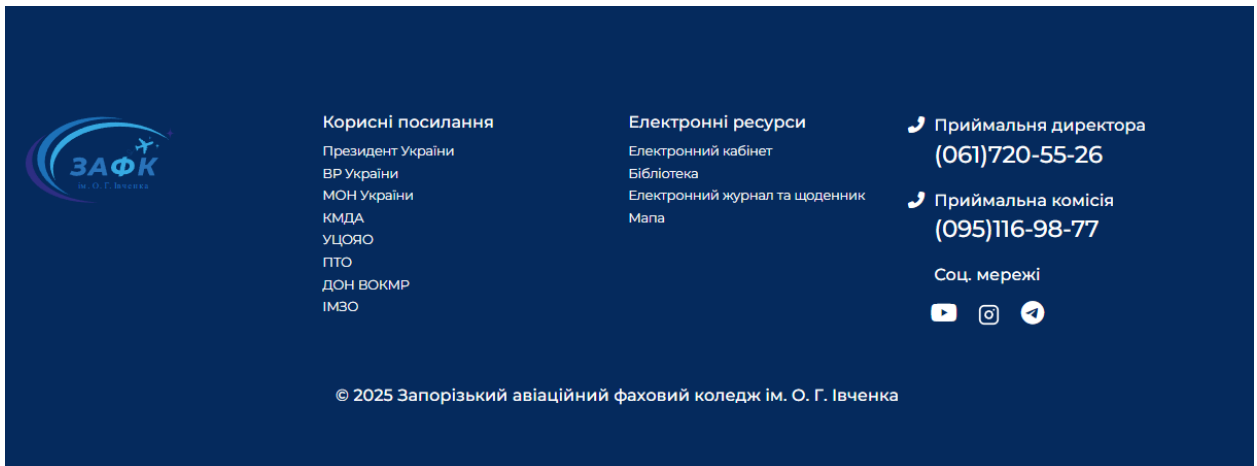


Рисунок 4.58 – Footer

Оскільки сайт коледжу є інформаційним – кількість сторінок дуже велика, нижче приведено файл web.php (рис 4.59–4.60), який демонструє інші сторінки сайту:

```
Route::get(uri: '/', action: [UkrController::class, 'index']);
Route::get(uri: '/library', action: [UkrController::class, 'library']);
Route::get(uri: '/contacts', action: [UkrController::class, 'contacts']);
Route::get(uri: '/team-teachers', action: [UkrController::class, 'teamlead']);
Route::get(uri: '/studkom', action: [UkrController::class, 'studkom']);
Route::get(uri: '/provisions', action: [UkrController::class, 'provisions']);
Route::get(uri: '/rating', action: [UkrController::class, 'rating']);
Route::get(uri: '/social-scholarship', action: [UkrController::class, 'sociasolar']);
Route::get(uri: '/academic-scholarship', action: [UkrController::class, 'academsolar']);
Route::get(uri: '/dorm', action: [UkrController::class, 'dorm']);
Route::get(uri: '/news', action: [UkrController::class, 'news']);
Route::get(uri: '/lang-ukr', action: [UkrController::class, 'langukr']);
Route::get(uri: '/graduates', action: [UkrController::class, 'graduates']);
Route::get(uri: '/news/{slug}', action: [UkrController::class, 'show']->name(name: 'show'));
Route::get(uri: '/educational-events', action: [UkrController::class, 'eduevents']);
Route::get(uri: '/material-and-technical-support', action: [UkrController::class, 'materials']);
Route::get(uri: '/prospective-student', action: [UkrController::class, 'prospectivestudent']);
Route::get(uri: '/psychological-service', action: [UkrController::class, 'psychologicalservice']);
Route::get(uri: '/prevention-of-corruption', action: [UkrController::class, 'corruption']);
Route::get(uri: '/help-the-student', action: [UkrController::class, 'helpStudent']);
Route::get(uri: '/psychology-parents', action: [UkrController::class, 'psychParent']);
Route::get(uri: '/psychology-students', action: [UkrController::class, 'psychStudent']);
Route::get(uri: '/psychology-teachers', action: [UkrController::class, 'psychTeacher']);
Route::get(uri: '/competitions', action: [UkrController::class, 'competitions']);
Route::get(uri: '/managment', action: [UkrController::class, 'managment']);
Route::get(uri: '/history', action: [UkrController::class, 'history']);
Route::get(uri: '/qualifications', action: [UkrController::class, 'qualifications']);
Route::get(uri: '/certificates', action: [UkrController::class, 'certificates']);
Route::get(uri: '/certification-of-teaching-staff', action: [UkrController::class, 'certificateStaff']);
Route::get(uri: '/licenses', action: [UkrController::class, 'licenses']);
Route::get(uri: '/computer-and-economic-department', action: [UkrController::class, 'departmentComputer']);
Route::get(uri: '/cyclic-commissions', action: [UkrController::class, 'cyclicCommissions']);
Route::get(uri: '/cycle-commission-of-aircraft-design-disciplines', action: [UkrController::class, 'avia']);
Route::get(uri: '/cycle-commission-of-economic-disciplines', action: [UkrController::class, 'economic']);
Route::get(uri: '/cycle-commission-of-electrotechnical-disciplines', action: [UkrController::class, 'electrotechincs']);
Route::get(uri: '/cycle-commission-of-computer-disciplines', action: [UkrController::class, 'computers']);
Route::get(uri: '/cycle-commission-of-metallurgical-disciplines', action: [UkrController::class, 'metalurg']);
Route::get(uri: '/cycle-commission-of-special-technological-disciplines', action: [UkrController::class, 'spectech']);
Route::get(uri: '/cycle-commission-of-socio-political-disciplines', action: [UkrController::class, 'sociopolit']);
Route::get(uri: '/cycle-commission-of-physical-and-mathematical-disciplines', action: [UkrController::class, 'physmath']);
Route::get(uri: '/cycle-commission-of-physical-culture-disciplines', action: [UkrController::class, 'physcult']);
Route::get(uri: '/cycle-commission-of-philological-disciplines', action: [UkrController::class, 'phylogy']);
Route::get(uri: '/department-of-industrial-engineering-and-applied-mechanics', action: [UkrController::class, 'departmentGM']);
```

Рисунок 4.59 – Файл Web.php

```

Route::get(uri: '/department-of-aerospace-and-metalurgy', action: [UkrController::class, 'departmentAKT']);
Route::get(uri: '/reporting', action: [UkrController::class, 'reporting']);
Route::get(uri: '/feedback', action: [UkrController::class, 'feedback']);
Route::get(uri: '/opp', action: [UkrController::class, 'opp']);
Route::get(uri: '/information-about-material-and-technical', action: [UkrController::class, 'infabsup']);
Route::get(uri: '/life-safety-labor-protection-and-civil-protection', action: [UkrController::class, 'bespeka']);
Route::get(uri: '/achivments', action: [UkrController::class, 'comPos']);
Route::get(uri: '/educational-programs-of-specialized-secondary-education', action: [UkrController::class, 'oppsk']);
Route::get(uri: '/disciplines-forming-general-competences', action: [UkrController::class, 'generalCompetences']);
Route::get(uri: '/disciplines-forming-professional-competences', action: [UkrController::class, 'profCompetences']);
Route::get(uri: '/elective-educational-disciplines', action: [UkrController::class, 'electiveCompetences']);
Route::get(uri: '/educational-plans', action: [UkrController::class, 'eduPlans']);
Route::get(uri: '/students', action: [UkrController::class, 'students']);
Route::get(uri: '/regulatory-support', action: [UkrController::class, 'regulatory']);
Route::get(uri: '/test-pages', action: [UkrController::class, 'test']);
Route::get(uri: '/pedagogical-council', action: [UkrController::class, 'teachrada']);
Route::get(uri: '/metodical-council', action: [UkrController::class, 'metrada']);
Route::get(uri: '/in-progress', action: [UkrController::class, 'notFound']);
Route::get(uri: '/college-today', action: [UkrController::class, 'collegeToday']);
Route::get(uri: '/ensuring-the-educational-process', action: [UkrController::class, 'osv']);
Route::get(uri: '/monitoring-the-quality-of-education', action: [UkrController::class, 'quality']);
Route::get(uri: '/nmkd', action: [UkrController::class, 'nmkd']);
Route::get(uri: '/educational-work', action: [UkrController::class, 'nvr']);
Route::get(uri: '/preparatory-courses', action: [UkrController::class, 'preparatory']);
Route::get(uri: '/public-information', action: [UkrController::class, 'publicInf']);
Route::get(uri: '/academic-integrity', action: [UkrController::class, 'academicintegrity']);
Route::get(uri: '/academic-integrity/students', action: [UkrController::class, 'acs']);
Route::get(uri: '/academic-integrity/teachers', action: [UkrController::class, 'act']);
Route::get(uri: '/educational-work/work-group', action: [UkrController::class, 'nvrW']);
Route::get(uri: '/educational-work/students', action: [UkrController::class, 'nvrS']);
Route::get(uri: '/educational-work/vacancies', action: [UkrController::class, 'nvrV']);
Route::get(uri: '/educational-work/employers', action: [UkrController::class, 'nvrE']);
Route::get(uri: '/educational-work/cooperation-with-partners', action: [UkrController::class, 'nvrC']);
Route::get(uri: '/information-on-information-provision-of-educational-activities', action: [UkrController::class, 'zInfoP']);
Route::get(uri: '/information-about-teaching-staff-who-provide-the-educational-process', action: [UkrController::class, 'zTeacherP']);
Route::get(uri: '/international-cooperation', action: [UkrController::class, 'international']);
Route::get(uri: '/open-days', action: [UkrController::class, 'opendays']);

Route::post(uri: '/edit-schedule', action: [AdminController::class, 'editshedule'])->name(name: 'admin.editschedule');
Route::post(uri: '/edit-changes', action: [AdminController::class, 'editchanges'])->name(name: 'admin.editchanges');
Route::post(uri: '/edit-exams', action: [AdminController::class, 'editexams'])->name(name: 'admin.editexams');
Route::post(uri: '/edit-consults', action: [AdminController::class, 'editconsults'])->name(name: 'admin.editconsults');

Route::resource(name: '/news', controller: NewsController::class)->middleware(middleware: ['auth']);

```

Рисунок 4.60 – файл Web.php. Продовження.

5 ТЕСТУВАННЯ СИСТЕМИ

Тестування являє собою не менш важливий етап під час розробки систем, оскільки в програмній реалізації, все може виглядати чудово, на перший погляд, але надалі будуть помилки.

Для тесту системи новин – додамо тестову новину та після видалимо її.

В панелі адміністратора перейдемо до сторінки «Новини» та натиснемо на посилання «Додати новину» та заповнимо інформацію.

Назва новини
<input type="text" value="Тестова новина"/>
Короткий опис
<input type="text" value="Тестовий короткий опис"/>
Текст новини
<input type="text" value="Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum."/>
Дата новини
<input type="text" value="25 травня 2025"/>
Фото новини
<input type="text" value="Виберіть файл"/> icons8-зуб-80.png
Назва URL (slug)
<input type="text" value="testing-news"/>
<input type="button" value="ДОДАТИ НОВИНУ"/>

Рисунок 5.1 – Заповнена сторінка з новиною

Після успішного додавання, новина відображається на сторінці «Всі новини»

6

Тестова новина

РЕДАГУВАТИ

ВИДАЛИТИ

Рисунок 5.2 – Новина в списку всіх новин

Для успішного тестування, натиснемо на кнопку «Видалити».

Новини

ДОДАТИ НОВИНУ

Рисунок 5.3 – «Всі новини» після видалення

Тестування успішно виконано.

ВИСНОВКИ

У межах даної дипломної роботи було реалізовано інформаційну систему керування контентом веб-сайту «Запорізький Авіаційний Фаховий коледж» з адміністративною панеллю, що базується на сучасному PHP-фреймворку Laravel. Основну увагу було приділено створенню зручної структури бази даних, яка дозволяє легко додавати, редагувати та відображати важливу інформацію на сайті, таку як новини, документи, розклад, а також відомості про викладачів.

У процесі роботи було проведено аналіз вимог до функціоналу, визначено необхідні таблиці для збереження контенту, та спроектовано їх структуру. Створення бази даних і підключення її до Laravel-проекту здійснено з використанням інструменту PHPMyAdmin і системи міграцій, що забезпечило надійність та простоту керування схемою бази.

Веб-додаток розроблено з урахуванням стандартів MVC-архітектури: реалізовано маршрути, контролери, моделі та представлення. Особливу увагу приділено безпеці аутентифікації, а також зручності керування вмістом через адміністративну панель.

Підсумовуючи, можна стверджувати, що поставлені цілі були досягнуті: створено функціональну та масштабовану систему для управління освітнім контентом, яка є придатною до подальшого розвитку та впроваджено в реальне використання для потреб коледжу.

ПЕРЕЛІК ПОСИЛАНЬ

1. Про Освіту. Закон України від 05.09.2017 № 2145-VIII [Електронний ресурс]. — Режим доступу: <https://zakon.rada.gov.ua/laws/show/2145-19#Text> (дата звернення: 30.04.2025).
2. Ключові відмінності фреймворків Laravel та Django: який з них кращий для вашого проекту? [Електронний ресурс]. — Режим доступу: <https://wezom.com.ua/blog/django-ili-laravel>.
3. ОГЛЯД РІЗНИЙ МІЖ LARAVEL І WORDPRESS: ЯКИЙ ФРЕЙМВОРК ВИБРАТИ ДЛЯ РОЗРОБКИ САЙТУ? [Електронний ресурс]. — Режим доступу: <https://seo-evolution.com.ua/ru/blog/razrabotka/yakyy-framework-vybraty-dlya-rozrobky-sajtu> (дата звернення: 30.04.2025).
4. ПЕРЕВАГИ DJANGO НАД ІНШИМИ ФРЕЙМВОРКАМИ ТА ПОРІВНЯННЯ З CMS-СИСТЕМОЮ WORDPRESS. НТУ «Дніпровська політехніка» [Електронний ресурс]. — Режим доступу: <https://ir.nmu.org.ua/server/api/core/bitstreams/6c216113-3167-44bf-87f4-36e7e6406550/content> (дата звернення: 30.04.2025).
5. Laravel Documentation [Електронний ресурс]. — Режим доступу: <https://laravel.com/docs/12.x> (дата звернення: 30.04.2025).
6. WordPress документація [Електронний ресурс]. — Режим доступу: <https://wordpress.org/documentation/> (дата звернення: 30.04.2025).
7. Django документація [Електронний ресурс]. — Режим доступу: <https://docs.djangoproject.com/en/5.2/> (дата звернення: 30.04.2025).
8. Як працювати з SQL [Електронний ресурс]. — Режим доступу: <https://robotdreams.cc/uk/blog/540-17-porad-yak-pratsyuvaty-z-sql> (дата звернення: 30.04.2025)
9. Що таке база даних і чому вона важлива для бізнесу [Електронний ресурс]. — Режим доступу: <https://hostkoss.com/b/uk/database/> (дата звернення: 30.04.2025).

10. Петров І. Ларавел: Повний посібник для початківців [Текст] / І. Петров. — Київ: Вид. дім "Технології", 2020. — 256 с.
11. Сидоренко. О. Bootstrap 5: Сучасний дизайн веб-сайтів. [Текст] / О. Сидоренко. — Харків: Видавництво "ВебСвіт", 2019. — 180 с.
12. Коваленко. С. Розробка веб-додатків на Laravel [Текст] / С. Коваленко. — Львів: "Програміст", 2021. — 320 с.
13. Мельник. М. Основи Bootstrap та CSS [Текст] / М. Мельник. — Київ: "ІТ-Література", 2018. — 150 с.
14. Гриценко А. Laravel: Професійні рішення [Текст] / А. Гриценко. — Одеса: "КодКнига", 2022. — 290 с.
15. Шевченко Т. Адаптивний дизайн з Bootstrap 4 [Текст] / Т. Шевченко. — Харків: "ТехноЛіт", 2017. — 210 с.
16. Бойко Д. Розробка сайтів на Laravel 8 [Текст] / Д. Бойко. — Київ: "ВебАкадемія", 2021. — 275 с.
17. Іваненко Н. Bootstrap для початківців [Текст] / Н. Іваненко. — Львів: "Сучасні Технології", 2020. — 160 с.
18. Павленко О. Laravel та API: Поглиблений курс [Текст] / О. Павленко. — Дніпро: "DevPress", 2022.— 340 с.
19. Савченко Ю. Веб-дизайн з Bootstrap та JavaScript [Текст] / Ю. Савченко. — Київ: "ІнфоМедіа", 2019. — 230 с.
20. Лисенко В. Laravel: Швидкий старт [Текст] / В. Лисенко. — Харків: "ІТ-Книга", 2021. — 195 с.
21. Морозов А. Сучасні тенденції у Bootstrap 5 [Текст] / А. Морозов. — Львів: "ТехноВидав", 2023. — 200 с.

ДОДАТОК А

Програмний код

Код файлу маршрутизації web.php:

```
<?php
```

```
use App\Http\Controllers\NewsController;
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\UkrController;
use App\Http\Controllers\AdminController;
```

```
/*
```

```
|-----
```

```
| Web Routes
```

```
|-----
```

```
|
```

```
| Here is where you can register web routes for your application. These  
| routes are loaded by the RouteServiceProvider within a group which  
| contains the "web" middleware group. Now create something great!
```

```
|
```

```
*/
```

```
Route::get('/', [UkrController::class, 'index']);
Route::get('/library', [UkrController::class, 'library']);
Route::get('/contacts', [UkrController::class, 'contacts']);
Route::get('/team-teachers', [UkrController::class, 'teamlead']);
Route::get('/studkom', [UkrController::class, 'studkom']);
Route::get('/provisions', [UkrController::class, 'provisions']);
Route::get('/rating', [UkrController::class, 'rating']);
Route::get('/social-scholarship', [UkrController::class, 'socioasolar']);
```

```

Route::get('/academic-scholarship', [UkrController::class, 'academsolar']);
Route::get('/dorm', [UkrController::class, 'dorm']);
Route::get('/news', [UkrController::class, 'news']);
Route::get('/lang-ukr', [UkrController::class, 'langukr']);
Route::get('/graduates', [UkrController::class, 'graduates']);
Route::get('/news/{slug}', [UkrController::class, 'show'])->name('show');
Route::get('/educational-events', [UkrController::class, 'eduevents']);
Route::get('/material-and-technical-support', [UkrController::class, 'materials']);
Route::get('/prospective-student', [UkrController::class, 'prospectivestudent']);
Route::get('/psychological-service', [UkrController::class, 'psychologicalservice']);
Route::get('/prevention-of-corruption', [UkrController::class, 'corruption']);
Route::get('/help-the-student', [UkrController::class, 'helpStudent']);
Route::get('/psychology-parents', [UkrController::class, 'psyhParent']);
Route::get('/psychology-students', [UkrController::class, 'psyhStudent']);
Route::get('/psychology-teachers', [UkrController::class, 'psyhTeacher']);
Route::get('/competitions', [UkrController::class, 'competitions']);
Route::get('/managment', [UkrController::class, 'managment']);
Route::get('/history', [UkrController::class, 'history']);
Route::get('/qualifications', [UkrController::class, 'qualifications']);
Route::get('/certificates', [UkrController::class, 'certificates']);
Route::get('/certification-of-teaching-staff', [UkrController::class,
'certificateStaff']);
Route::get('/licenses', [UkrController::class, 'licenses']);
Route::get('/computer-and-economic-department', [UkrController::class,
'departmentComputer']);
Route::get('/cyclic-commissions', [UkrController::class, 'cyclicCommissions']);
Route::get('/cycle-commission-of-aircraft-design-disciplines',
[UkrController::class, 'avia']);
Route::get('/cycle-commission-of-economic-disciplines', [UkrController::class,
'economic']);

```

```

Route::get('/cycle-commission-of-electrotechnical-disciplines',
[UkrController::class, 'electrotechincs']);
Route::get('/cycle-commission-of-computer-disciplines',    [UkrController::class,
'computers']);
Route::get('/cycle-commission-of-metallurgical-disciplines', [UkrController::class,
'metalurg']);
Route::get('/cycle-commission-of-special-technological-disciplines',
[UkrController::class, 'spectech']);
Route::get('/cycle-commission-of-socio-political-disciplines',
[UkrController::class, 'sociopolit']);
Route::get('/cycle-commission-of-physical-and-mathematical-disciplines',
[UkrController::class, 'physmath']);
Route::get('/cycle-commission-of-physical-culture-disciplines',
[UkrController::class, 'physcult']);
Route::get('/cycle-commission-of-philological-disciplines', [UkrController::class,
'phylogy']);
Route::get('/department-of-industrial-engineering-and-applied-mechanics',
[UkrController::class, 'departmentGM']);
Route::get('/department-of-aerospace-and-metalurgy',    [UkrController::class,
'departmentAKT']);
Route::get('/reporting', [UkrController::class, 'reporting']);
Route::get('/feedback', [UkrController::class, 'feedback']);
Route::get('/opp', [UkrController::class, 'opp']);
Route::get('/information-about-material-and-technical',    [UkrController::class,
'infabsup']);
Route::get('/life-safety-labor-protection-and-civil-protection',
[UkrController::class, 'bespeka']);
Route::get('/achivments', [UkrController::class, 'comPos']);
Route::get('/educational-programs-of-specialized-secondary-education',
[UkrController::class, 'oppsk']);

```

```
Route::get('/disciplines-forming-general-competences', [UkrController::class,
'generalCompetences']);
Route::get('/disciplines-forming-professional-competences', [UkrController::class,
'profCompetences']);
Route::get('/elective-educational-disciplines', [UkrController::class,
'electiveCompetences']);
Route::get('/educational-plans', [UkrController::class, 'eduPlans']);
Route::get('/students', [UkrController::class, 'students']);
Route::get('/regulatory-support', [UkrController::class, 'regulatory']);
Route::get('/test-pages', [UkrController::class, 'test']);
Route::get('/pedagogical-council', [UkrController::class, 'teachrada']);
Route::get('/metodical-council', [UkrController::class, 'metrada']);
Route::get('/in-progress', [UkrController::class, 'notFound']);
Route::get('/college-today', [UkrController::class, 'collegeToday']);
Route::get('/ensuring-the-educational-process', [UkrController::class, 'osv']);
Route::get('/monitoring-the-quality-of-education', [UkrController::class, 'quality']);
Route::get('/nmkd', [UkrController::class, 'nmkd']);
Route::get('/educational-work', [UkrController::class, 'nvr']);
Route::get('/preparatory-courses', [UkrController::class, 'preparatory']);
Route::get('/public-information', [UkrController::class, 'publicInf']);
Route::get('/academic-integrity', [UkrController::class, 'academicintegrity']);
Route::get('/academic-integrity/students', [UkrController::class, 'acs']);
Route::get('/academic-integrity/teachers', [UkrController::class, 'act']);
Route::get('/educational-work/work-group', [UkrController::class, 'nvrW']);
Route::get('/educational-work/students', [UkrController::class, 'nvrS']);
Route::get('/educational-work/vacancies', [UkrController::class, 'nvrV']);
Route::get('/educational-work/employers', [UkrController::class, 'nvrE']);
Route::get('/educational-work/cooperation-with-partners', [UkrController::class,
'nvrC']);
```

```
Route::get('/information-on-information-provision-of-educational-activities',
[UkrController::class, 'zInfoP']);
```

```
Route::get('/information-about-teaching-staff-who-provide-the-educational-
process', [UkrController::class, 'zTeacherP']);
```

```
Route::get('/international-cooperation', [UkrController::class, 'international']);
```

```
Route::get('/open-days', [UkrController::class, 'opendays']);
```

```
Route::post('/edit-shedule', [AdminController::class, 'editshedule']-
>name('admin.editschedule');
```

```
Route::post('/edit-changes', [AdminController::class, 'editchanges']-
>name('admin.editchanges');
```

```
Route::post('/edit-exams', [AdminController::class, 'editexams']-
>name('admin.editexams');
```

```
Route::post('/edit-consults', [AdminController::class, 'editconsults']-
>name('admin.editconsults');
```

```
Route::resource('/news', NewsController::class)->middleware(['auth']);
```

```
Route::get('/periodicals/computers', [UkrController::class, 'perComp']);
```

```
Route::get('/dashboard', [AdminController::class, 'dashboard']-
>name('dashboard');
```

```
Route::get('/posts', function () {
    return view('posts');
})->middleware(['auth']->name('posts');
```

```
require __DIR__.'./auth.php';
```

Код контролерів AdminController, NewsController та UkrController

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use Illuminate\Http\Request;
```

```
use App\Models\Information;
```

```
use Illuminate\Support\Facades\Redirect;
```

```
class AdminController extends Controller
```

```
{
```

```
    public function dashboard(){
```

```
        $schedule = Information::where('name', 'Розклад занять')->first();
```

```
        $changes = Information::where('name', 'Зміни в розкладі')->first();
```

```
        $exams = Information::where('name', 'Розклад екзаменів')->first();
```

```
        $consults = Information::where('name', 'Графік консультацій')->first();
```

```
        return view('dashboard', ['schedule'=> $schedule, 'changes'=> $changes
, 'exams'=> $exams, 'consults'=> $consults]);
```

```
    }
```

```
    public function editshedule(Request $request){
```

```
        $info = Information::where('name', 'Розклад занять')->first();
```

```
        $link = $request->lessonstime;
```

```
        $info->name = "Розклад занять";
```

```
        $info->link = $link;
```

```
        $info->save();
```

```
        return Redirect::to('/dashboard');
```

```
    }
```

```
    public function editchanges(Request $request){
```

```
        $info = Information::where('name', 'Зміни в розкладі')->first();
```

```

$link = $request->changetime;
$info->name = "Зміни в розкладі";
$info->link = $link;
$info->save();
return Redirect::to('/dashboard');
}

public function editexams(Request $request){
    $info = Information::where('name', 'Розклад екзаменів')->first();
    $link = $request->exams;
    $info->name = "Розклад екзаменів";
    $info->link = $link;
    $info->save();
    return Redirect::to('/dashboard');
}

public function editconsults(Request $request){
    $info = Information::where('name', 'Графік консультацій')->first();
    $link = $request->consults;
    $info->name = "Графік консультацій";
    $info->link = $link;
    $info->save();
    return Redirect::to('/dashboard');
}
}
}

```

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use App\Models\News;
```

```
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Redirect;

class NewsController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $posts = News::all();
        return view('lang.ukr.news.index', ['posts'=> $posts]);
    }

    /**
     * Show the form for creating a new resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function create()
    {
        return view('lang.ukr.news.create');
    }

    /**
     * Store a newly created resource in storage.
     *

```

```
* @param \Illuminate\Http\Request $request
* @return \Illuminate\Http\Response
*/
public function store(Request $request)
{
    $news = new News();
    $news->title = $request->title;
    $news->description = $request->description;
    $news->text = $request->text;
    $news->date = $request->date;
    $news->slug = $request->slug;

    if ($request->hasFile('photo')) {
        $file = $request->file('photo');

        $filename = $file->getClientOriginalName();

        $file->move(public_path('/img/news/'), $filename);
        $news->photo = $filename;

    } else {
        $filename = 'noimage.png';
        $news->photo = $filename;
    }
    $news->save();
    return Redirect::to('/newss');
}

/**
```

```

* Display the specified resource.
*
* @param \App\Models\News $news
* @return \Illuminate\Http\Response
*/
public function show(News $newss)
{
    $news = News::where('id', $newss->id)->first();
    return view('lang.ukr.news.show', ["news"=> $news]);
}

/**
* Show the form for editing the specified resource.
*
* @param \App\Models\News $news
* @return \Illuminate\Http\Response
*/
public function edit(News $news)
{
    $news = News::where('id', $news->id)->first();
    return view('lang.ukr.news.edit', ["news"=> $news]);
}

/**
* Update the specified resource in storage.
*
* @param \Illuminate\Http\Request $request
* @param \App\Models\News $news
* @return \Illuminate\Http\Response

```

```

*/
public function update(Request $request, News $newss)
{
    $news = News::where('id', $newss->id)->first();
    $news->title = $request->title;
    $news->description = $request->description;
    $news->text = $request->text;
    $news->date = $request->date;
    $news->slug = $request->slug;
    $news->save();
    return Redirect::to('/newss');
}

/**
 * Remove the specified resource from storage.
 *
 * @param \App\Models\News $news
 * @return \Illuminate\Http\Response
 */
public function destroy(News $newss)
{
    // dd($news);
    $newss->delete();

    return Redirect::to('/newss');
}
}

```

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use Illuminate\Http\Request;
```

```
use App\Models\Post;
```

```
use App\Models\Employee;
```

```
use App\Models\Commision;
```

```
use App\Models\CommisionPost;
```

```
use App\Models\Opp;
```

```
use App\Models\Library;
```

```
use App\Models\News;
```

```
class UkrController extends Controller
```

```
{
```

```
    public function library(){
```

```
        return view('lang.ukr.library');
```

```
    }
```

```
    public function avia(){
```

```
        $commisions = Commision::where("position", "Циклова комісія  
авіаконструкторських дисциплін")->orderBy('id')->get();
```

```
        return view('lang.ukr.commisions.avia',[
```

```
            "commisions"=>$commisions
```

```
        ]);
```

```
    }
```

```
    public function index(){
```

```

    $posts = News::orderBy('created_at','desc')->paginate(3);
    return view('lang.ukr.index',[
        'posts'=>$posts
    ]);
}

public function news(){
    $posts = News::orderBy('created_at','desc')->get();
    return view('lang.ukr.posts.index',[
        'posts'=>$posts
    ]);
}

public function show($slug){
    $post = Post::where('slug', $slug)->first();
    return view('lang.ukr.posts.show',[
        'post'=>$post
    ]);
}

public function comPos(Request $request){
    $spec = $request->input('specs');
    $title = CommisionPost::where('spec', $spec)->first();
    $specs = CommisionPost::where('spec', $spec)->orderBy('date','desc')->get();
    return view('lang.ukr.commissions.achivments',[
        'specs'=>$specs,
        'title'=>$title
    ]);
}

```

```
public function teachrada() {  
    return view('lang.ukr.teachrada');  
}
```

```
public function metrada() {  
    return view('lang.ukr.metrada');  
}
```

```
public function competitions() {  
    return view('lang.ukr.competitions');  
}
```

```
public function infabsup() {  
    return view('lang.ukr.infabsup');  
}
```

```
public function graduates() {  
    return view('lang.ukr.graduates');  
}
```

```
public function nvr() {  
    return view('lang.ukr.nvr');  
}
```

```
public function acs() {  
    return view('lang.ukr.acs');  
}
```

```
public function publicInf(){  
    return view('lang.ukr.publicInf');  
}
```

```
public function act(){  
    return view('lang.ukr.act');  
}
```

```
public function nvrW(){  
    return view('lang.ukr.nvrW');  
}
```

```
public function nvrS(){  
    return view('lang.ukr.nvrS');  
}
```

```
public function nvrV(){  
    return view('lang.ukr.nvrV');  
}
```

```
public function nvrE(){  
    return view('lang.ukr.nvrE');  
}
```

```
public function nvrC(){  
    return view('lang.ukr.nvrC');  
}
```

```
public function contacts(){  
    return view('lang.ukr.contact');
```

```
}
```

```
public function teamlead(){  
    return view('lang.ukr.teamlead');  
}
```

```
public function certificateStaff(){  
  
    $opps = Opp::where("category", "Атестація педагогічних працівників")-  
>get();  
    return view('lang.ukr.certificateStaff',[  
        'opps'=>$opps  
    ]);  
}
```

```
public function studkom(){  
    return view('lang.ukr.studkom');  
}
```

```
public function rating(){  
    return view('lang.ukr.rating');  
}
```

```
public function test(){  
    return view('lang.ukr.testPage');  
}
```

```
public function sociasolar(){  
    return view('lang.ukr.sociasolar');
```

```
}
```

```
public function academsolar(){  
    return view('lang.ukr.acadsolar');  
}
```

```
public function qualifications(){  
    return view('lang.ukr.qualifications');  
}
```

```
public function opendays(){  
    return view('lang.ukr.opendays');  
}
```

```
public function feedback(){  
    return view('lang.ukr.feedback');  
}
```

```
public function international(){  
    return view('lang.ukr.international ');  
}
```

```
public function quality(){
```

```
    $students = Opp::where("category", "Результати опитування здобувачів  
освіти")->get();
```

```
    $parents = Opp::where("category", "Результати опитування батьків  
студентів")->get();
```

```
    $teachers = Opp::where("category", "Результати моніторингу  
працевлаштування та подальшого навчання випускників")->get();
```

```
return view('lang.ukr.quality',[
    'students'=>$students,
    'parents'=>$parents,
    'teachers'=>$teachers
]);
}

public function dorm(){
    return view('lang.ukr.dorm');
}

public function eduevents(){
    return view('lang.ukr.eduevents');
}

public function materials(){
    return view('lang.ukr.materials');
}

public function prospectivestudent(){
    return view('lang.ukr.prospectivestudent');
}

public function psychologicalservice(){

    $opps = Opp::where("category", "Психологічна служба")->get();
    return view('lang.ukr.psychologicalservice',[
        'opps'=>$opps
    ]);
}
```

```
public function langukr(){
    return view('lang.ukr.ukrlang');
}
```

```
public function corruption(){
    return view('lang.ukr.corruption');
}
```

```
public function licenses(){
    return view('lang.ukr.licenses');
}
```

```
public function preparatory(){
    return view('lang.ukr.preparatory');
}
```

```
public function academicintegrity(){
    return view('lang.ukr.dobr');
}
```

```
public function managment(){
    $managers = Employee::all();
    return view('lang.ukr.managment',[
        'managers'=>$managers
    ]);
}
```

```
public function history(){
```

```

return view('lang.ukr.history');
}

```

```

public function certificates() {
    return view('lang.ukr.certificates');
}

```

```

public function departmentComputer() {
    return view('lang.ukr.departments.computer');
}

```

```

public function cyclicCommissions() {
    return view('lang.ukr.commissions.main');
}

```

```

public function economic() {

    $commissions = Commission::where("position", "Циклова комісія
економічних дисциплін")->orderBy('id')->get();

    return view('lang.ukr.commissions.economic',[
        "commissions"=>$commissions
    ]);
}

public function electrotechincs() {

```

```

    $commissions = Commission::where("position", "Циклова комісія
електротехнічних дисциплін")->orderBy('id')->get();

    return view('lang.ukr.commissions.electrotechincs',[

```

```

        "commisions"=>$commisions
    ];
}

public function computers(){

    $commisions = Commision::where("position", "Циклова комісія
комп'ютерних дисциплін")->orderBy('id')->get();
    return view('lang.ukr.commissions.computers',[
        "commisions"=>$commisions
    ]);
}

public function metalurg(){

    $commisions = Commision::where("position", "Циклова комісія
металургійних дисциплін")->orderBy('id')->get();
    return view('lang.ukr.commissions.metalurg',[
        "commisions"=>$commisions
    ]);
}

public function spectech(){

    $commisions = Commision::where("position", "Циклова комісія
спецтехнологічних дисциплін")->orderBy('id')->get();
    return view('lang.ukr.commissions.spectech',[
        "commisions"=>$commisions
    ]);
}

```

```

public function physmath() {

    $commisions = Commision::where("position", "Циклова комісія фізико-
математичних дисциплін")->orderBy('id')->get();
    return view('lang.ukr.commissions.physmath',[
        "commisions"=>$commisions
    ]);
}

```

```

public function sociopolit() {

    $commisions = Commision::where("position", "Циклова комісія суспільно-
політичних дисциплін")->orderBy('id')->get();
    return view('lang.ukr.commissions.sociopolit',[
        "commisions"=>$commisions
    ]);
}

```

```

public function physcult() {

    $commisions = Commision::where("position", "Циклова комісія
фізкультурних дисциплін")->orderBy('id')->get();
    return view('lang.ukr.commissions.physcult',[
        "commisions"=>$commisions
    ]);
}

```

```

public function phylogy() {

```

```
$commissions = Commission::where("position", "Циклова комісія  
філологічних дисциплін")->orderBy('id')->get();  
return view('lang.ukr.commissions.phylogy',[  
    "commissions"=>$commissions  
]);  
}  
  
public function departmentGM(){  
  
    return view('lang.ukr.departments.gm');  
}  
  
public function departmentAKT(){  
  
    return view('lang.ukr.departments.akt');  
}  
  
public function notFound(){  
  
    return view('lang.ukr.notfound');  
}  
  
public function osv(){  
  
    return view('lang.ukr.osv');  
}  
  
public function collegeToday(){  
  
    return view('lang.ukr.today');
```

```
}
```

```
public function reporting(){  
  
    return view('lang.ukr.reporting');  
}
```

```
public function psyhParent(){  
  
    return view('lang.ukr.psychology.parents');  
}
```

```
public function psyhStudent(){  
  
    return view('lang.ukr.psychology.students');  
}
```

```
public function psyhTeacher(){  
  
    return view('lang.ukr.psychology.teacher');  
}
```

```
public function bespeka(){  
  
    return view('lang.ukr.bespeka');  
}
```

```
public function regulatory(){  
  
    return view('lang.ukr.regulatory');
```

```
}
```

```
public function students(){
```

```
    return view('lang.ukr.students');
```

```
}
```

```
public function nmkd(){
```

```
    $opps = Opp::where("category", "НМКД")->get();
```

```
    return view('lang.ukr.nmkd',[
```

```
        'opps'=>$opps
```

```
    ]);
```

```
}
```

```
public function helpStudent(){
```

```
    $opps = Opp::where("category", "На допомогу студенту")->get();
```

```
    return view('lang.ukr.helpStudent',[
```

```
        'opps'=>$opps
```

```
    ]);
```

```
}
```

```
public function zInfoP(){
```

```
    $opps = Opp::where("category", "Відомості про інформаційне забезпечення  
освітньої діяльності")->get();
```

```
    return view('lang.ukr.zinfop',[
```

```
        'opps'=>$opps
```

```

    });
}

```

```

public function zTeacherP(){

```

```

    $opps = Opp::where("category", "Відомості про педагогічних працівників,
які забезпечують освітній процес")->get();

```

```

    return view('lang.ukr.zteacherp',[

```

```

        'opps'=>$opps

```

```

    ]);

```

```

}

```

```

public function opp(Request $request){

```

```

    $category = $request->input("category");

```

```

    $opps = Opp::where("category", "Освітньо-професійні програми")-
>where("name", $category)->get();

```

```

    return view('lang.ukr.opp',[

```

```

        'opps'=>$opps

```

```

    ]);

```

```

}

```

```

public function provisions(){

```

```

    $opps = Opp::where("category", "Положення")->get();

```

```

    return view('lang.ukr.provisions',[

```

```

        'opps'=>$opps

```

```

    ]);

```

```

}

```

```

public function oppsk(Request $request){

```

```

    $category = $request->input("category");

```

```

    $opps = Opp::where("category", "Освітні програми профільної середньої
освіти")->where("name", $category)->get();
    return view('lang.ukr.oppsk',[
        'opps'=>$opps
    ]);
}

```

```

public function eduPlans(Request $request){
    $category = $request->input("category");
    $opps = Opp::where("category", "Навчальні плани")-
>where("name", $category)->get();
    return view('lang.ukr.eduplans',[
        'opps'=>$opps
    ]);
}

```

```

public function generalCompetences(){
    $generals = Library::where("category", "Загальні предмети")->get();
    $specialities = Library::where("category", "Загальні предмети
спеціальності")->get();
    return view('lang.ukr.library.general',[
        'generals'=>$generals,
        'specialities' => $specialities
    ]);
}

```

```

public function profCompetences(){
    $specialities = Library::where("category", "Професійні компетентності")-
>get();
    return view('lang.ukr.library.profLibrary',[

```

```

        'specialities' => $specialities
    ];
}

public function electiveCompetences() {
    $specialities = Library::where("category", "Вибіркові освітні дисципліни")-
>get();
    return view('lang.ukr.library.electiveCompetences', [
        'specialities' => $specialities
    ]);
}

public function perComp() {

    return view('lang.ukr.periodicals.computers');
}

}

```

Код файлів міграції

```

<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateNewsTable extends Migration
{
    /**
     * Run the migrations.

```

```

*
* @return void
*/
public function up()
{
    Schema::create('news', function (Blueprint $table) {
        $table->id();
        $table->string('title');
        $table->text('description');
        $table->text('text');
        $table->string('date');
        $table->string('photo');
        $table->string('slug')->unique();
        $table->timestamps();
    });
}

/**
 * Reverse the migrations.
 *
 * @return void
 */
public function down()
{
    Schema::dropIfExists('news');
}
}

```

```
<?php
```

```
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateInformationsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('informations', function (Blueprint $table) {
            $table->id();
            $table->string("name");
            $table->string("link");
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('informations');
    }
}
```

```
}
```

Код клієнтської частини головної сторінки

```
@extends("layouts.templates.ukr.template")
```

```
@section('title','Головна')
```

```
@section('content')
```

```
<div class="modal fade" id="akt" tabindex="-1" aria-labelledby="akt" aria-
hidden="true">
```

```
  <div class="modal-dialog modal-lg">
```

```
    <div class="modal-content">
```

```
      <div class="modal-header">
```

```
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-
label="Close"></button>
```

```
      </div>
```

```
      <div class="modal-body">
```

```
        <div class="row">
```

```
          <h3 class="fw-bold text-center">134 Авіаційна та ракетно-космічна
техніка</h3>
```

```
          <h4 class="fw-bold text-center">Освітньо-професійні програми:</h4>
```

```
            <h4 class="fw-bold text-center">Виробництво авіаційних
двигунів</h4>
```

```
            <h4 class="fw-bold text-center">Виробництво літальних апаратів</h4>
```

```
              <h5 class="fst-italic text-center">Кваліфікація – технік-технолог
(механіка)</h5>
```

```
        <p>Хочеш підкорити космос? Мрієш будувати космічні ракети та
літаки? Завжди прагнув розібратись, з чого складаються супутники і як
```

долетіти до Марса? Тоді спеціальність «Авіаційна та ракетно-космічна техніка» – твій вибір!

Випускники цієї спеціальності – інженери найвищого класу! Це люди, які здатні проектувати і діагностувати будь-які технічні системи будь-якої складності. Авіаційні агрегати – дуже складні і відповідальні системи, що працюють в умовах великих перепадів температур, перевантажень і вібрацій. До їх надійності пред’являються підвищені вимоги. Тому метою навчання є формування навичок і умінь розрахунку комп’ютерного проектування паливних, масляних, повітряних і протиобмерзних систем, систем автоматичного управління, виконавчих механізмів систем автоматичного управління і регулювання авіаційних двигунів.

Випускникам цієї спеціальності пропонується широкий вибір при визначенні місця своєї подальшої роботи і розвитку. Це можуть бути як підприємства авіаційної промисловості, так і освітні та науково-дослідні інститути, такі як Національний аерокосмічний університет «Харківський авіаційний інститут», Київський національний авіаційний університет, Інститут проблем машинобудування Національної академії наук України та інші.

</div>

</div>

</div>

</div>

</div>

<div class="modal fade" id="gm" tabindex="-1" aria-labelledby="gm" aria-hidden="true">

<div class="modal-dialog modal-lg">

<div class="modal-content">

```

<div class="modal-header">
  <button type="button" class="btn-close" data-bs-dismiss="modal" aria-
label="Close"></button>
</div>
<div class="modal-body">
  <div class="row">
    <h3 class="fw-bold text-center">133 Галузеве машинобудування</h3>
    <h4 class="fw-bold text-center">Освітньо-професійні програми:</h4>
    <h4 class="fw-bold text-center">Технологія обробки матеріалів на
верстатах та автоматичних лініях</h4>
    <h4 class="fw-bold text-center">Комп'ютерні технології в
машинобудуванні</h4>
    <h5 class="fst-italic text-center">Кваліфікація технік-технолог
(механіка)</h5>

```

<p>Існує міф, що машинобудування – це спеціальність, на якій навчають будувати машини, але насправді це можливість одержати найкращу технічну освіти й стати справжнім майстром інженерних вершин. </p>

<p>Галузеве машинобудування – це спеціальність широкого профілю, яка дозволить навчитись знатися на складових частинах великих промислових виробництв, зрозуміти інноваційні технології виготовлення деталей для літаків і ракетно-космічної техніки, формувати засоби автоматизації сучасних підприємств.</p>

<p>Ви будете знати, як впроваджуються новітні комп'ютерні технології на виробництві, вільно оперувати всіма потрібними програмами й розробляти надсучасні підходи до оптимізації української промисловості, Ми готуємо фахівців високого рівня для успішного кар'єрного зростання на

підприємствах, де здобуті знання та навички високо цінуються роботодавцями. </p>

</div>

</div>

</div>

</div>

</div>

<div class="modal fade" id="pm" tabindex="-1" aria-labelledby="pm" aria-hidden="true">

<div class="modal-dialog modal-lg">

<div class="modal-content">

<div class="modal-header">

<button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>

</div>

<div class="modal-body">

<div class="row">

<h3 class="fw-bold text-center">131 Прикладна механіка</h3>

<h4 class="fw-bold text-center">Освітньо-професійні програми:</h4>

<h4 class="fw-bold text-center">Обслуговування верстатів з програмним управлінням і робототехнічних комплексів</h4>

<h5 class="fst-italic text-center">Кваліфікація технік-технолог (механіка)</h5>

<p>Ця спеціальність дає фундаментальні знання в галузі прикладної механіки інформаційних систем та технологій машинобудування </p>

<p>Здобувачі освіти вивчають системи автоматизованого проектування електричних та електронних систем, розробку та виготовлення

друкованих плат та їх 3D-моделювання, автоматизовані та комп'ютеризовані виробничі комплекси на базі сучасних цифрових технологій і комп'ютерної техніки та займаються розробкою сучасних роботизованих систем. Опановують основи роботи з платформами Arduino, NodeMCU та системи ЧПУ фірм Sinumeric, Fanuc, Heidenhain, WestLabs

Випускники працюють на підприємствах авіабудівної та машинобудівної галузей.

</div>

</div>

</div>

</div>

</div>

<div class="modal fade" id="metalurg" tabindex="-1" aria-labelledby="metalurg" aria-hidden="true">

<div class="modal-dialog modal-lg">

<div class="modal-content">

<div class="modal-header">

<button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>

</div>

<div class="modal-body">

<div class="row">

<h3 class="fw-bold text-center">136 Металургія</h3>

<h4 class="fw-bold text-center">Освітньо-професійні програми:</h4>

<h4 class="fw-bold text-center">Ливарне виробництво чорних та кольорових металів</h4>

<h5 class="fst-italic text-center">Кваліфікація – технік-технолог (лиття металів)</h5>

<p>Хочеш пізнати мистецтво виготовлення і застосування чорних та кольорових металів та сплавів? Тоді спеціальність «Металургія» – саме для тебе! </p>

<p>Секрети виготовлення і обробки металів – одна з найстародавніших процесій, яку опанувало людство. Зараз метал використовується для виготовлення різноманітних товарів, приладів та устаткування. Продукцією ливарного виробництва є литі деталі різноманітної конфігурації та розмірів для авіаційних двигунів, складні деталі для всіх галузей машинобудування та навіть ювелірні прикраси та вироби для стоматології. </p>

<p>Металургія – складна галузь промисловості, до якої належать ріні за організацією і технологією виробництва: виробництво чорних та кольорових металів та сплавів, отримання і використання композиційних матеріалів, адитивні технології у вигляді 3D-моделювання та 3D-друку. </p>

</div>

</div>

</div>

</div>

</div>

<div class="modal fade" id="compeng" tabindex="-1" aria-labelledby="compeng" aria-hidden="true">

<div class="modal-dialog modal-lg">

<div class="modal-content">

<div class="modal-header">

```
<button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
```

```
</div>
```

```
<div class="modal-body">
```

```
<div class="row">
```

```
<h3 class="fw-bold text-center">123 Комп'ютерна інженерія</h3>
```

```
<h4 class="fw-bold text-center">Освітньо-професійні програми:</h4>
```

```
<h4 class="fw-bold text-center">Обслуговування комп'ютерних систем і мереж</h4>
```

```
<h5 class="fst-italic text-center">Кваліфікація – технік з комп'ютерної інженерії</h5>
```

```
<p>Спеціальність для тих, хто більше любить «програмувати залізо»! </p>
```

```
<p>Ця спеціальність пов'язана з апаратним забезпеченням комп'ютерної техніки і водночас підтримується потужною підготовкою з програмування. Здобувачі освіти отримають професійні знання з розробки, експлуатаційного обслуговування, ремонту та модернізації системних програмних і технічних засобів ІТ, їх локально-мережових реалізаціях в офісах, установах, організаціях та на підприємствах. </p>
```

```
<p>Наші студенти набувають професійних навичок, що допоможуть реалізувати себе в найрізноманітніших сферах діяльності в Україні та за кордоном. </p>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div class="modal fade" id="compnau" tabindex="-1" aria-labelledby="compnau"
aria-hidden="true">
```

```
  <div class="modal-dialog modal-lg">
```

```
    <div class="modal-content">
```

```
      <div class="modal-header">
```

```
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-
label="Close"></button>
```

```
      </div>
```

```
    <div class="modal-body">
```

```
      <div class="row">
```

```
        <h3 class="fw-bold text-center">122 Комп'ютерні науки</h3>
```

```
        <h4 class="fw-bold text-center">Освітньо-професійні програми:</h4>
```

```
        <h4 class="fw-bold text-center">Обслуговування програмних систем і
комплексів</h4>
```

```
        <h5 class="fst-italic text-center">Кваліфікація – технік-програміст</h5>
```

```
      <p>Ти прагнеш брати участь у розвитку комп'ютерних технологій,
створювати цікаві і корисні програми, веб-сайти, 3D-моделі, комп'ютерні
ігри? Тоді спеціальність «Комп'ютерні науки» - саме для тебе! </p>
```

```
      <p>Сьогодні важко уявити собі життя без комп'ютерів. Темпи
розвитку комп'ютерних технологій вражають. </p>
```

```
      <p>Комп'ютерні науки – це дослідження того, як працюють
інформаційні технології з точки зору теоретичної та математичної
перспективи, з точки зору забезпечення наукового обґрунтування процесів
розгортання, інтеграції та взаємодії інформаційних технологій. Наші студенти
вивчають алгоритми і структури даних, мови програмування, операційні
```

системи та комп'ютерні мережі, розробку програмного забезпечення, бази даних. </p>

</div>

</div>

</div>

</div>

</div>

<div class="modal fade" id="economics" tabindex="-1" aria-labelledby="economics" aria-hidden="true">

<div class="modal-dialog modal-lg">

<div class="modal-content">

<div class="modal-header">

<button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>

</div>

<div class="modal-body">

<div class="row">

<h3 class="fw-bold text-center">051 Економіка</h3>

<h4 class="fw-bold text-center">Освітньо-професійні програми:</h4>

<h4 class="fw-bold text-center">Економіка підприємства</h4>

<h4 class="fw-bold text-center">Бізнес-економіка</h4>

<h5 class="fst-italic text-center">Кваліфікація – економіст</h5>

<p>Економіст – це освіта сучасності, яка має такі унікальні риси як перспективність, універсальність та прибутковість. </p>

<p>Економіка в усіх її проявах міцно увійшла в життя людства – це наука про використання обмежених ресурсів, виробництво, розподіл, обмін, продаж та споживання товарів і послуг. </p>

<p>Здобувачі освіти вивчають функціонування будь-якого підприємства, незалежно від його розміру та організаційно-правової форми господарювання, закономірності функціонування та розвитку соціально-економічних систем і процесів їх моделювання, прогнозування та регулювання, мотивацію і поведінку економічних суб'єктів. </p>

<p>Ця спеціальність надає можливість працевлаштування у будь-якій сфері, де потрібні економічні знання. </p>

</div>

</div>

</div>

</div>

</div>

<div class="modal fade" id="dzvinks" tabindex="-1" aria-labelledby="dzvinks" aria-hidden="true">

<div class="modal-dialog modal-lg">

<div class="modal-content">

<div class="modal-header">

<button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>

</div>

<div class="modal-body">

<div class="row">

</div>

```

    </div>
  </div>
</div>
</div>

<div class="modal fade" id="groups" tabindex="-1" aria-labelledby="groups" aria-
hidden="true">
  <div class="modal-dialog modal-lg">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-
label="Close"></button>
      </div>
      <div class="modal-body">
        <div class="row">
          
        </div>
      </div>
    </div>
  </div>
</div>
</div>

<!-- <div class="home __main-section">
  <div class="container">
    <div class="first">

      <div>
        <div class="row align-items-center justify-content-md-between px-3">

```

```
<div class="col-12 col-lg-6 main-section__text text-center text-lg-start">
```

```
<h1>134 Авіаційна та ракетно-космічна техніка</h1>
```

```
<p>Хочеш підкорити космос? Мрієш будувати ракети та літаки? Завжди прагнув розібратись, з чого складаються супутники і як долетіти до Марса? Тоді спеціальність «Авіаційна та ракетно-космічна техніка» – твій вибір! </p>
```

```
<div class="div-btn">
```

```
<a href="#" class="more-btn" data-bs-toggle="modal" data-bs-target="#akt">Детальніше <i data-feather="arrow-right"></i></a>
```

```
</div>
```

```
<div class="main-section__icons">
```

```
<a href="#"><i data-feather="facebook"></i></a>
```

```
<a href="#"><i data-feather="instagram"></i></a>
```

```
</div>
```

```
</div>
```

```
<div class="col-lg-6 d-none d-lg-flex main-section_img justify-content-end">
```

```

```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div>
```

```
<div class="row align-items-center justify-content-md-between px-3">
```

```
<div class="col-12 col-lg-6 main-section__text text-center text-lg-start">
```

```
<h1>133 Галузеве машинобудування</h1>
```

```
<p>Існує міф, що машинобудування – це спеціальність, на якій навчають будувати машини, але насправді це можливість одержати найкращу технічну освіти й стати справжнім майстром інженерних вершин. </p>
```

```
<div class="div-btn">
```

```
<a href="#" class="more-btn" data-bs-toggle="modal" data-bs-target="#gm">Детальніше <i data-feather="arrow-right"></i></a>
```

```
</div>
```

```
<div class="main-section__icons">
```

```
<a href="#"><i data-feather="facebook"></i></a>
```

```
<a href="#"><i data-feather="instagram"></i></a>
```

```
</div>
```

```
</div>
```

```
<div class="col-lg-6 d-none d-lg-flex main-section_img justify-content-end">
```

```

```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div>
```

```
<div class="row align-items-center justify-content-md-between px-3">
```

```
<div class="col-12 col-lg-6 main-section__text text-center text-lg-start">
```

```
<h1>131 Прикладна механіка</h1>
```

<p>Ця спеціальність дає фундаментальні знання в галузі прикладної механіки інформаційних систем та технологій машинобудування
</p>

<div class="div-btn">

Детальніше <i data-feather="arrow-right"></i>

</div>

<div class="main-section__icons">

<i data-feather="facebook"></i>

<i data-feather="instagram"></i>

</div>

</div>

<div class="col-lg-6 d-none d-lg-flex main-section_img justify-content-end">

</div>

</div>

</div>

<div>

<div class="row align-items-center justify-content-md-between px-3">

<div class="col-12 col-lg-6 main-section__text text-center text-lg-start">

<h1>051 Економіка</h1>

<p>Економіст – це освіта сучасності, яка має такі унікальні риси як перспективність, універсальність та прибутковість.</p>

<div class="div-btn">

```

    <a href="#" class="more-btn" data-bs-toggle="modal" data-bs-
target="#economics">Детальніше <i data-feather="arrow-right"></i></a>

```

```

</div>

```

```

<div class="main-section__icons">

```

```

    <a href="#"><i data-feather="facebook"></i></a>

```

```

    <a href="#"><i data-feather="instagram"></i></a>

```

```

</div>

```

```

</div>

```

```

    <div class="col-lg-6 d-none d-lg-flex main-section_img justify-
content-end">

```

```

```

```

</div>

```

```

</div>

```

```

</div>

```

```

<div>

```

```

    <div class="row align-items-center justify-content-md-between px-3">

```

```

        <div class="col-12 col-lg-6 main-section__text text-center text-lg-
start">

```

```

            <h1>136 Металургія</h1>

```

```

            <p>Хочеш пізнати мистецтво виготовлення і застосування
чорних та кольорових металів та сплавів? Тоді спеціальність «Металургія» –
саме для тебе!</p>

```

```

            <div class="div-btn">

```

```

                <a href="#" class="more-btn" data-bs-toggle="modal" data-bs-
target="#metalurg">Детальніше <i data-feather="arrow-right"></i></a>

```

```

</div>

<div class="main-section__icons">
  <a href="#"><i data-feather="facebook"></i></a>
  <a href="#"><i data-feather="instagram"></i></a>
</div>
</div>

<div class="col-lg-6 d-none d-lg-flex main-section_img justify-
content-end">
  
</div>
</div>
</div>

<div>
  <div class="row align-items-center justify-content-md-between px-3">
    <div class="col-12 col-lg-6 main-section__text text-center text-lg-
start">
      <h1>123 Комп'ютерна інженерія</h1>
      <p>Спеціальність для тих, хто більше любить «програмувати
залізо»!</p>
      <div class="div-btn">
        <a href="#" class="more-btn" data-bs-toggle="modal" data-bs-
target="#compeng">Детальніше <i data-feather="arrow-right"></i></a>
      </div>
    </div>
  </div>
  <div class="main-section__icons">
    <a href="#"><i data-feather="facebook"></i></a>

```

```

        <a href="#"><i data-feather="instagram"></i></a>
    </div>
</div>

<div class="col-lg-6 d-none d-lg-flex main-section_img justify-
content-end">
    
</div>
</div>
</div>

<div>
    <div class="row align-items-center justify-content-md-between px-3">
        <div class="col-12 col-lg-6 main-section__text text-center text-lg-
start">
            <h1>122 Комп'ютерні науки</h1>
            <p>Ти прагнеш брати участь у розвитку комп'ютерних
технологій, створювати цікаві і корисні програми, веб-сайти, 3D-моделі,
комп'ютерні ігри? Тоді спеціальність «Комп'ютерні науки» - саме для
тебе!</p>
            <div class="div-btn">
                <a href="#" class="more-btn" data-bs-toggle="modal" data-bs-
target="#compnau">Детальніше <i data-feather="arrow-right"></i></a>
            </div>
        </div>
    </div>
    <div class="main-section__icons">
        <a href="#"><i data-feather="facebook"></i></a>
        <a href="#"><i data-feather="instagram"></i></a>
    </div>

```

```

</div>

<div class="col-lg-6 d-none d-lg-flex main-section_img justify-
content-end">
  
</div>
</div>
</div>
</div>
</div>

</div>
</div>

<section class="home __about-section">
  <div class="container">
    <div class="row">
      <div class="col-12 col-md-5">
        
      </div>
      <div class="col-12 col-md-7 home __about-section __text text-center text-
md-start mt-5 mt-md-0">
        <h1>Про коледж</h1>
        <p class="first">Запорізький авіаційний фаховий коледж ім. О. Г.
Івченка-заклад із майже віковою історією: в травні 2020 року йому
виповнилось 90 років! І весь цей час заклад готує та випускає
висококваліфікованих спеціалістів, які є конкурентоспроможними на
сучасному ринку праці. </p>
        <p>Серед закладів освіти міста Фаховий коледж має один з
найбільших обсягів прийому на бюджетні місця, що забезпечує безкоштовне
навчання за спеціальностями, дозволяє отримати диплом державного зразка,

```

надає можливість продовжити навчання у вищих навчальних закладах, забезпечує гарантоване працевлаштування.

```
</p>
```

```
<a href="/history" class="more-btn">Детальніше <i data-feather="arrow-right"></i></a>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</section> -->
```

```
<section class="home __specialties">
```

```
<div class="container">
```

```
<div class="home __title text-center">
```

```
<h1>Спеціальності</h1>
```

```
<p>В коледжі готують фахівців за освітньо-професійним ступенем "фаховий молодший бакалавр"</p>
```

```
</div>
```

```
<div class="slid">
```

```
<!-- По классу получаем те элементы которые нужны для слайдер в JS ->
```

```
<div class="your-class">
```

```
<div class="card" style="width: 300px;">
```

```

```

```
<div class="card-body">
```

```
<h5 class="card-title">134 Авіаційна та ракетно-космічна техніка</h5>
```

```
<p class="card-text">Виробництво авіаційних двигунів. Виробництво літальних апаратів.</p>
```

```

    <a href="#" class="more-btn" data-bs-toggle="modal" data-bs-
target="#akt">Детальніше <i data-feather="arrow-right"></i></a>

```

```

    </div>

```

```

</div>

```

```

<div class="card" style="width: 300px;">

```

```

```

```

    <div class="card-body big-card-body">

```

```

        <h5 class="card-title">133 Галузеве машинобудування</h5>

```

```

        <p class="card-text">Технологія обробки матеріалів на верстатах і
автоматичних лініях, Комп'ютерні технології в машинобудуванні

```

```

    </p>

```

```

    <a href="#" class="more-btn" data-bs-toggle="modal" data-bs-
target="#gm">Детальніше <i data-feather="arrow-right"></i></a>

```

```

    </div>

```

```

</div>

```

```

<div class="card" style="width: 300px;">

```

```

```

```

    <div class="card-body">

```

```

        <h5 class="card-title">131 Прикладна механіка</h5>

```

```

        <p class="card-text">Обслуговування верстатів з програмним
управлінням та робото-технічних комплексів.</p>

```

```

    <a href="#" class="more-btn" data-bs-toggle="modal" data-bs-
target="#pm">Детальніше <i data-feather="arrow-right"></i></a>

```

```

    </div>

```

```

</div>

```

```

<div class="card" style="width: 300px;">

```

```

```

```

<div class="card-body">
  <h5 class="card-title">051 Економіка</h5>
  <p class="card-text">Економіка підприємства.<br> Бізнес-
економіка</p>
  <a href="#" class="more-btn" data-bs-toggle="modal" data-bs-
target="#economics">Детальніше <i data-feather="arrow-right"></i></a>
</div>
</div>

```

```

<div class="card" style="width: 300px;">
  
  <div class="card-body">
    <h5 class="card-title">136 Металургія</h5>
    <p class="card-text">Ливарне виробництво чорних та кольорових
металів</p>
    <a href="#" class="more-btn" data-bs-toggle="modal" data-bs-
target="#metalurg">Детальніше <i data-feather="arrow-right"></i></a>
  </div>
</div>

```

```

<div class="card" style="width: 300px;">
  
  <div class="card-body">
    <h5 class="card-title">123 Комп'ютерна інженерія</h5>
    <p class="card-text">Обслуговування комп'ютерних систем і
мереж.</p>
    <a href="#" class="more-btn" data-bs-toggle="modal" data-bs-
target="#compeng">Детальніше <i data-feather="arrow-right"></i></a>
  </div>

```

```

</div>

<div class="card" style="width: 300px;">
  
  <div class="card-body">
    <h5 class="card-title">122 Комп'ютерні науки</h5>
    <p class="card-text">Обслуговування програмних систем і
комплексів</p>
    <a href="#" class="more-btn" data-bs-toggle="modal" data-bs-
target="#compnau">Детальніше <i data-feather="arrow-right"></i></a>
  </div>
</div>
</div>
</div>
</div>
</div>
</div>
</section>

<section class="home __news">
  <div class="container">
    <div class="home __title text-center">
      <h1>Новини</h1>
      <!-- <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Magni,
iste.</p> -->
    </div>
    <div class="row">
      { {-- @foreach($posts as $post)

      <div class="col-12 col-lg-4 student-work">
        <a href="{{route('show',$post->slug)}}">
          <div class="card" style="width: 380px;">

```

```

        
        <div class="card-body">
            <p class="card-text"><i data-
feather="calendar"></i><span>{{ $post->date }}</span></p>
            <h5 class="card-title">{{ $post->title }}</h5>
        </div>
    </div>
</a>
</div>

```

```
@endforeach --}}
```

```
@foreach($posts as $post)
```

```

<div class="col-12 student-work">
    <a href="{{ route('show',$post->slug) }}">

        <div class="news">
            <div class="news__img">
                
            </div>

            <div class="news__text">
                <p class="card-text"><i data-
feather="calendar"></i><span>{{ $post->date }}</span></p>
                <h5 class="card-title">{{ $post->title }}</h5>
                <p class="news__desc">{!! $post->description !!</p>

```

```

        </div>
    </div>
</a>
</div>

<hr style="background-color:#4B91F1; height: 3px;" class="mt-3">

@endforeach

<div class="student-btn text-center">
    <a href="/news" class="more-btn"><span>Всі новини</span><i data-
feather="arrow-right"></i></a>
</div>

</div>
</section>

<section class="home __main-buttons">
    <div class="container">

        <div class="row">
            <div class="col-12 col-lg-6 justify-content-start">
                <a href="https://docs.google.com/spreadsheets/d/1ZbsbHcS-
MH0B0pkMs613QXvYAlt3O4cF/edit?usp=sharing&oid=110861569292838174
490&rtpof=true&sd=true" target="_blank">
                    <div class="main-button" style="background-color:#626bdc;
color:white;">
                </div>
            </div>
            <div class="row">
                <div class="col-2">

```

```

    
  </div>
  <div class="col-10 d-flex align-items-center">
    <h1 class="mb-0">Розклад занять</h1>
  </div>
</div>
</div>
</a>
<a
href="https://docs.google.com/document/d/15sTjEGGQPuYWYICcPlbZNSkSwo
C1Aubdu5dAVtF1hzs/edit?usp=sharing">
  <div class="main-button" style="background-color:#d9615e;
color:white;">
  <div class="row">
    <div class="col-2">
      
    </div>
    <div class="col-10 d-flex align-items-center">
      <h1 class="mb-0">Зміни в розкладі</h1>
    </div>
  </div>
</div>
</a>
</div>
<div class="col-12 col-lg-6">
  <a href="#" data-bs-toggle="modal" data-bs-target="#groups">
    <div class="main-button" style="background-color:#b36197;
color:white;">

```

```

<div class="row">
  <div class="col-2">
    
  </div>
  <div class="col-10 d-flex align-items-center">
    <h1 class="mb-0">Гуртки</h1>
  </div>
</div>
</div>
</a>

<a href="#" data-bs-toggle="modal" data-bs-target="#dzvinks">
  <div class="main-button" style="background-color:#549a65;
color:white;">
  <div class="row">
    <div class="col-2">
      
    </div>
    <div class="col-10 d-flex align-items-center">
      <h1 class="mb-0">Розклад дзвінків</h1>
    </div>
  </div>
</div>
</a>
</div>
</div>
</div>
</div>
</section>

```

```

<!-- <section class="home __more-info">
  <div class="container">
    <div class="row">
      <div class="col-12 col-lg-4">
        <div class="card text-center" style="width: 350px;">
          
          <div class="card-body">
            <h5 class="card-title">Абітурієнтам</h5>
            <a href="/prospective-student" class="more-btn">Детальніше <i
data-feather="arrow-right"></i></a>
          </div>
        </div>
      </div>
    </div>
    <div class="col-12 col-lg-4 d-lg-flex justify-content-md-center mt-5 mt-
lg-0">
      <div class="card text-center" style="width: 350px;">
        
        <div class="card-body">
          <h5 class="card-title">Студентам</h5>
          <a href="#" class="more-btn">Детальніше <i data-feather="arrow-
right"></i></a>
        </div>
      </div>
    </div>
    <div class="col-12 col-lg-4 d-lg-flex justify-content-md-end mt-5 mt-lg-
0">
      <div class="card text-center" style="width: 340px;">

```

```


<div class="card-body">
  <h5 class="card-title">Викладачам</h5>
  <a href="#" class="more-btn">Детальніше <i data-feather="arrow-
right"></i></a>
  </div>
</div>
</div>
</div>
</div>
</div>
</section> -->

```

```

<section class="home __ reasons">
<div class="container">
  <h1 class="home __ reasons-title">Декілька причин навчатися з нами</h1>
  <div class="row">
    <div class="col-12 col-lg-4">
      <div class="reason reason-violet">
        <div class="row">
          <div class="col-2">
            
          </div>
          <div class="col-10">
            <h1>Безкоштовне навчання</h1>
          </div>
          <p>Навчання у коледжі повністю безоплатне, за виключенням
окремих спеціальностей</p>
        </div>
      </div>
    </div>
  </div>
  <div class="reason reason-violet">

```

```

<div class="row">
  <div class="col-2">
    
  </div>
  <div class="col-10">
    <h1>Гарний старт</h1>
  </div>
  <p>З дипломом молодшого спеціаліста та фахового молодшого
бакалавра продовжуйте здобувати Вищу освіту.</p>
</div>
</div>

<div class="reason reason-violet">
  <div class="row">
    <div class="col-2">
      
    </div>
    <div class="col-10">
      <h1>Відповідальність</h1>
    </div>
    <p>Відповідальне ставлення до навчання, самостійне планування
навчального графіка, розподіл грошових витрат швидко перетворює
безпорадну дитину на дорослого.</p>
  </div>
</div>

</div>
<div class="col-12 col-lg-4">
  <div class="reason reason-blue">
    <div class="row">

```

```

<div class="col-2">
  
</div>

<div class="col-10">
  <h1>Висока стипендія</h1>
</div>

<p>Виплачується стипендія. Регулярні виплати без затримок</p>
</div>
</div>

<div class="reason reason-blue">
  <div class="row">
    <div class="col-2">
      
    </div>
    <div class="col-10">
      <h1>Робота</h1>
    </div>
    <p>Випускники забезпечуються робочими місцями.</p>
  </div>
</div>

<div class="reason reason-blue">
  <div class="row">
    <div class="col-2">
      
    </div>
    <div class="col-10">
      <h1>Виробнича практика</h1>
    </div>
    <p>Практичне навчання студенти проходять на кращих підприємствах міста.</p>
  </div>
</div>

```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div class="col-12 col-lg-4">
```

```
<div class="reason reason-sky">
```

```
<div class="row">
```

```
<div class="col-2">
```

```

```

```
</div>
```

```
<div class="col-10">
```

```
<h1>Спортивне життя</h1>
```

```
</div>
```

```
<p>Студенти мають всі умови для здорового активного дозвілля  
й змістовного та затишного відпочинку</p>
```

```
</div>
```

```
</div>
```

```
<div class="reason reason-sky">
```

```
<div class="row">
```

```
<div class="col-2">
```

```

```

```
</div>
```

```
<div class="col-10">
```

```
<h1>Досвідчені педагоги</h1>
```

```
</div>
```

```
<p>Освітній процес забезпечується висококваліфікованим  
викладацьким колективом та кураторським супроводом протягом усіх років  
навчання</p>
```

```
</div>
```

```

</div>
<div class="reason reason-sky">
  <div class="row">
    <div class="col-2">
      
    </div>
    <div class="col-10">
      <h1>Матеріальна підтримка</h1>
    </div>
    <p>Соціально вразливі категорії забезпечуються соціальними
стипендіями, а активні студенти мають можливість отримати матеріальне
заохочення</p>
  </div>
</div>
</div>
<!-- <div class="col-12">

</div> -->
</div>
</div>
</section>

<!-- <section class="home __specialties">
  <div class="container">
    <div class="home __title text-center">
      <h1>Спеціальності</h1>
      <p>В коледжі готують фахівців освітньо-кваліфікаційного рівня
«фаховий молодший спеціаліст»</p>
    </div>
  </div>
</section>

```

```
<div class="slid">
```

По классу получаем те элементы которые нужны для слайдер в JS

```
<div class="your-class">
```

```
<div class="card" style="width: 300px;">
```

```

```

```
<div class="card-body">
```

```
<h5 class="card-title">134 Авіаційна та ракетно-космічна  
техніка</h5>
```

```
<p class="card-text">Виробництво авіаційних двигунів.  
Виробництво літальних апаратів.</p>
```

```
<a href="#" class="more-btn" data-bs-toggle="modal" data-bs-  
target="#akt">Детальніше <i data-feather="arrow-right"></i></a>
```

```
</div>
```

```
</div>
```

```
<div class="card" style="width: 300px;">
```

```

```

```
<div class="card-body big-card-body">
```

```
<h5 class="card-title">133 Галузеве машинобудування</h5>
```

```
<p class="card-text">Технологія обробки матеріалів на верстатах і  
автоматичних лініях</p>
```

```
<a href="#" class="more-btn" data-bs-toggle="modal" data-bs-  
target="#gm">Детальніше <i data-feather="arrow-right"></i></a>
```

```
</div>
```

```
</div>
```

```
<div class="card" style="width: 300px;">
```

```

```

```
<div class="card-body">
```

```
<h5 class="card-title">131 Прикладна механіка</h5>
```

```
<p class="card-text">Обслуговування верстатів з програмним управлінням та робото-технічних комплексів.</p>
```

```
<a href="#" class="more-btn" data-bs-toggle="modal" data-bs-target="#pm">Детальніше <i data-feather="arrow-right"></i></a>
```

```
</div>
```

```
</div>
```

```
<div class="card" style="width: 300px;">
```

```

```

```
<div class="card-body">
```

```
<h5 class="card-title">051 Економіка</h5>
```

```
<p class="card-text">Економіка підприємства.<br> Бізнес-економіка</p>
```

```
<a href="#" class="more-btn" data-bs-toggle="modal" data-bs-target="#economics">Детальніше <i data-feather="arrow-right"></i></a>
```

```
</div>
```

```
</div>
```

```
<div class="card" style="width: 300px;">
```

```

```

```
<div class="card-body">
```

```
<h5 class="card-title">136 Металургія</h5>
```

```
<p class="card-text">Ливарне виробництво чорних та кольорових металів</p>
```

```
<a href="#" class="more-btn" data-bs-toggle="modal" data-bs-target="#metalurg">Детальніше <i data-feather="arrow-right"></i></a>
```

```
</div>
```

```
</div>
```

```

<div class="card" style="width: 300px;">
  
  <div class="card-body">
    <h5 class="card-title">123 Комп'ютерна інженерія</h5>
    <p class="card-text">Обслуговування комп'ютерних систем і
мереж.</p>
    <a href="#" class="more-btn" data-bs-toggle="modal" data-bs-
target="#conpeng">Детальніше <i data-feather="arrow-right"></i></a>
  </div>
</div>

<div class="card" style="width: 300px;">
  
  <div class="card-body">
    <h5 class="card-title">122 Комп'ютерні науки</h5>
    <p class="card-text">Обслуговування програмних систем і
КОМПЛЕКСІВ</p>
    <a href="#" class="more-btn" data-bs-toggle="modal" data-bs-
target="#compnau">Детальніше <i data-feather="arrow-right"></i></a>
  </div>
</div>
</div>
</div>
</section> -->

<!--
<section class="home __students-work">
  <div class="container">

```

```

<div class="home__title text-center">
  <h1>Роботи студентів</h1>
  <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Magni,
iste.</p>
</div>
<div class="row">
  <div class="col-12 col-lg-4 student-work text-center text-lg-start">
    
    <h3>Ілля Нагін</h3>
    <p>Студент курсу КН-17</p>
  </div>
  <div class="col-12 col-lg-4 student-work text-center text-lg-start">
    
    <h3>Катарина Андрусейко</h3>
    <p>Студентка курсу ГМ-16</p>
  </div>
  <div class="col-12 col-lg-4 student-work text-center text-lg-start">
    
    <h3>Марина Батейко</h3>
    <p>Студент курсу КІ-19</p>
  </div>
</div>
<div class="student-btn text-center">
  <a href="#" class="more-btn"><span>Показати ще</span><i data-
feather="arrow-right"></i></a>
</div>

</div>
</section> -->

```

```
<!--  
<section class="home __events">  
  <div class="container">  
    <div class="home __title text-center">  
      <h1>Головні заходи коледжу</h1>  
      <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Aperiam,  
fugiat, suscipit assumenda eum asperiores pariatur cumque odio sint temporibus  
ducimus a eius nulla mollitia, atque cupiditate maiores voluptates. Id, repellat.</p>  
    </div>  
  
    <div class="home __events-slider">  
      <div class="event">  
        <div class="row">  
          <div class="col-12 col-lg-2">  
            <div class="date">  
              <p class="cif">23</p>  
              <p class="month">Гру</p>  
            </div>  
          </div>  
          <div class="col-12 col-lg-10 event-title">  
            <h1>День відкритих дверей</h1>  
          </div>  
  
          <div class="event-body">  
            <div class="overlay"></div>  
            <p>Оголошено набір на підготовчі курси!</p>
```

```

    <ul>
      <li>покращити рівень знань;</li>
      <li>повторити повний шкільний курс;</li>
      <li>пройти підготовку відповідно до програми вступних
випробувань;</li>
    </ul>

```

```

</div>

```

```

<div class="event-footer d-flex justify-content-between">

```

```

  <div class="link">

```

```

    <a href="#">Читати далі</a>

```

```

  </div>

```

```

<div class="time">

```

```

  <i data-feather="clock"></i>

```

```

  <span>10:00</span>

```

```

</div>

```

```

</div>

```

```

</div>

```

```

</div>

```

```

<div class="event">

```

```

  <div class="row">

```

```

    <div class="col-12 col-lg-2">

```

```

      <div class="date">

```

```

        <p class="cif">23</p>

```

```

        <p class="month">Гру</p>

```

```

      </div>

```

```
</div>
```

```
<div class="col-12 col-lg-10 event-title">
```

```
<h1>День відкритих дверей</h1>
```

```
</div>
```

```
<div class="event-body">
```

```
<div class="overlay"></div>
```

```
<p>Оголошено набір на підготовчі курси!</p>
```

```
<ul>
```

```
<li>покращити рівень знань;</li>
```

```
<li>повторити повний шкільний курс;</li>
```

```
<li>пройти підготовку відповідно до програми вступних  
випробувань;</li>
```

```
</ul>
```

```
</div>
```

```
<div class="event-footer d-flex justify-content-between">
```

```
<div class="link">
```

```
<a href="#">Читати далі</a>
```

```
</div>
```

```
<div class="time">
```

```
<i data-feather="clock"></i>
```

```
<span>10:00</span>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```

<div class="event">
  <div class="row">
    <div class="col-12 col-lg-2">
      <div class="date">
        <p class="cif">23</p>
        <p class="month">Гру</p>
      </div>
    </div>
    <div class="col-12 col-lg-10 event-title">
      <h1>День відкритих дверей</h1>
    </div>

    <div class="event-body">
      <div class="overlay"></div>
      <p>Оголошено набір на підготовчі курси!</p>
      <ul>
        <li>покращити рівень знань;</li>
        <li>повторити повний шкільний курс;</li>
        <li>пройти підготовку відповідно до програми вступних
випробувань;</li>
      </ul>

    </div>

    <div class="event-footer d-flex justify-content-between">
      <div class="link">
        <a href="#">Читати далі</a>
      </div>

```

```

        <div class="time">
            <i data-feather="clock"></i>
            <span>10:00</span>
        </div>
    </div>
</div>
</div>
</div>

</div>

</div>
</section>
-->

<section class="home __ contacts">
    <div class="container">
        <div class="home __ title text-center">
            <h1>Залишилися запитання?</h1>
            <p>Надішліть нам листа на нашу пошту з цікавим для вас
питанням.</p>
        </div>
        @include("layouts.templates.ukr.contacts")
    </div>
</section>

@endsection

```

Код сторінки «Всі новини»

```
@extends("layouts.templates.ukr.template")
```

```
@section('title','Новини')
```

```
@section('content')
```

```
<section class="home__news">
```

```
  <div class="container">
```

```
    <div class="home__title text-center mb-5">
```

```
      <h1>Новини</h1>
```

```
    </div>
```

```
  <div class="row">
```

```
    @foreach($posts as $post)
```

```
      <div class="col-12 col-lg-4 student-work mb-5">
```

```
        <a href="{{route('show',$post->slug)}}">
```

```
          <div class="card" style="width: 380px;">
```

```
            
```

```
            <div class="card-body">
```

```
              <p class="card-text"><i data-
```

```
feather="calendar"></i><span>{{ $post->date }}</span></p>
```

```
              <h5 class="card-title">{{ $post->title }}</h5>
```

```
            </div>
```

```
          </div>
```

```
        </a>
```

```
      </div>
```

```
    @endforeach
```

```
</div>
```

```
</section>
```

```
@endsection
```

Код сторінки «Новина»

```
@extends("layouts.templates.ukr.template")
```

```
@section('title','$post->title')
```

```
@section('content')
```

```
<section class="post__show">
```

```
<div class="container">
```

```
<h1 class="main-h1">{{ $post->title }}</h1>
```

```
<div class="text-center">
```

```

```

```
</div>
```

```
    {!! $post->text !!
```

```
</div>
```

```
</section>
```

```
@endsection
```

Код головної сторінки панелі адміністратора

```
<x-app-layout>
```

```
<x-slot name="header">
```

```
<h2 class="font-semibold text-xl text-gray-800 leading-tight">
```

```
    {{ __( 'Головна' ) }}
```

```
</h2>
```

```
</x-slot>
```

```

<div class="py-12">
  <div class="max-w-7xl mx-auto sm:px-6 lg:px-8">
    <div class="bg-white overflow-hidden shadow-sm sm:rounded-lg">
      <div class="p-6 bg-white border-b border-gray-200">
        <form action="{{ route('admin.editschedule') }}" method="POST"
class="flex items-center" style="margin-bottom: 15px;">
          @csrf
          <label for="lessonstime">Розклад занять</label>
            <input value="{{ $schedule->link }}" type="text"
name="lessonstime" id="lessonstime" class="block min-w-0 grow py-1 pr-3 pl-1
text-base text-gray-900 placeholder:text-gray-400 focus:outline-none sm:text-sm/6"
style="margin: 0 15px;">
            <button class="inline-flex items-center px-4 py-2 border border-
transparent rounded-md font-semibold text-xs text-white uppercase tracking-widest
hover:bg-gray-700 active:bg-gray-900 focus:outline-none focus:border-gray-900
focus:ring ring-gray-300 disabled:opacity-25 transition ease-in-out duration-150"
style="background-color:#0063ff;">ЗМІНИТИ</button>
          </form>

          <form action="{{ route('admin.editchanges') }}" method="POST"
class="flex items-center" style="margin-bottom: 15px;">
            @csrf
            <label for="lessonstime">Зміни в розкладі</label>
              <input value="{{ $changes->link }}" type="text" name="changetime"
id="lessonstime" class="block min-w-0 grow py-1 pr-3 pl-1 text-base text-gray-900
placeholder:text-gray-400 focus:outline-none sm:text-sm/6" style="margin: 0
15px;">
              <button class="inline-flex items-center px-4 py-2 border border-
transparent rounded-md font-semibold text-xs text-white uppercase tracking-widest

```

```

hover:bg-gray-700 active:bg-gray-900 focus:outline-none focus:border-gray-900
focus:ring ring-gray-300 disabled:opacity-25 transition ease-in-out duration-150"
style="background-color:#0063ff;">ЗМІНИТИ</button>

```

```
</form>
```

```

<form action="{{ route('admin.edite exams') }}" method="POST"
class="flex items-center" style="margin-bottom: 15px;">

```

```
@csrf
```

```
<label for="lessonstime">Розклад екзаменів</label>
```

```

<input value="{{ $exams->link }}" type="text" name="exams"
id="lessonstime" class="block min-w-0 grow py-1 pr-3 pl-1 text-base text-gray-900
placeholder:text-gray-400 focus:outline-none sm:text-sm/6" style="margin: 0
15px;">

```

```

<button class="inline-flex items-center px-4 py-2 border border-
transparent rounded-md font-semibold text-xs text-white uppercase tracking-widest
hover:bg-gray-700 active:bg-gray-900 focus:outline-none focus:border-gray-900
focus:ring ring-gray-300 disabled:opacity-25 transition ease-in-out duration-150"
style="background-color:#0063ff;">ЗМІНИТИ</button>

```

```
</form>
```

```

<form action="{{ route('admin.editconsults') }}" method="POST"
class="flex items-center" style="margin-bottom: 15px;">

```

```
@csrf
```

```
<label for="lessonstime">Графік консультацій</label>
```

```

<input value="{{ $consults->link }}" type="text" name="consults"
id="lessonstime" class="block min-w-0 grow py-1 pr-3 pl-1 text-base text-gray-900
placeholder:text-gray-400 focus:outline-none sm:text-sm/6" style="margin: 0
15px;">

```

```

<button class="inline-flex items-center px-4 py-2 border border-
transparent rounded-md font-semibold text-xs text-white uppercase tracking-widest

```

```

hover:bg-gray-700 active:bg-gray-900 focus:outline-none focus:border-gray-900
focus:ring ring-gray-300 disabled:opacity-25 transition ease-in-out duration-150"
style="background-color:#0063ff;">Змінити</button>
    </form>
  </div>
</div>
</div>
</div>
</x-app-layout>

```

Код сторінки для роботи з новинами в панелі адміністратора

```

<x-app-layout>
  <x-slot name="header">
    <div class="flex justify-between items-center">
      <h2 class="font-semibold text-xl text-gray-800 leading-tight">
        {{ __( 'Новини' ) }}
      </h2>
      <a href="/newss/create" class="inline-flex items-center px-4 py-2 border
border-transparent rounded-md font-semibold text-xs text-white uppercase tracking-
widest hover:bg-gray-700 active:bg-gray-900 focus:outline-none focus:border-
gray-900 focus:ring ring-gray-300 disabled:opacity-25 transition ease-in-out
duration-150" style="background-color: green;" >Додати новину</a>
    </div>
  </x-slot>
  <div class="py-12">

```

```

<div class="max-w-7xl mx-auto sm:px-6 lg:px-8">
  <div class="bg-white overflow-hidden shadow-sm sm:rounded-lg">
    <div class="p-6 bg-white border-b border-gray-200">
      @foreach ($posts as $post)
        <div class="flex justify-between items-center" style="margin-bottom:
15px">
          <div>
            <p>{{ $post->id }}</p>
          </div>
          <div>
            <p><a href="{{ route('newss.show', ['newss' => $post]) }}">{{
$post->title }}</a></p>
          </div>
          <div class="flex">
            <a href="{{ route('newss.edit', ['newss' => $post]) }}"
class="inline-flex items-center px-4 py-2 border border-transparent rounded-md
font-semibold text-xs text-white uppercase tracking-widest hover:bg-gray-700
active:bg-gray-900 focus:outline-none focus:border-gray-900 focus:ring ring-gray-
300 disabled:opacity-25 transition ease-in-out duration-150" style="background-
color: #0063ff; margin-right: 10px;" >Редагувати</a>
            <form action="{{ route('newss.destroy', ['newss' => $post->id])
}}" method="POST">
              @csrf
              @method("DELETE")
              <button class="inline-flex items-center px-4 py-2 border
border-transparent rounded-md font-semibold text-xs text-white uppercase tracking-
widest hover:bg-gray-700 active:bg-gray-900 focus:outline-none focus:border-
gray-900 focus:ring ring-gray-300 disabled:opacity-25 transition ease-in-out
duration-150" style="background-color: red;" >Видалити</button>
            </form>
          </div>
        </div>
      @endforeach
    </div>
  </div>
</div>

```

```

        </div>
    </div>
    @endforeach

</div>
</div>
</div>
</div>
</x-app-layout>

```

Код сторінки з додаванням новини

```

<x-app-layout>
  <x-slot name="header">
    <div class="flex justify-between items-center">
      <h2 class="font-semibold text-xl text-gray-800 leading-tight">
        {{ __('Додати новину') }}
      </h2>
    </div>
  </x-slot>

  <div class="py-12">
    <div class="max-w-7xl mx-auto sm:px-6 lg:px-8">
      <div class="bg-white overflow-hidden shadow-sm sm:rounded-lg">
        <div class="p-6 bg-white border-b border-gray-200">
          <form action="{{ route('newss.store') }}" method="POST" class="text-
center" style="margin-bottom: 15px;" enctype="multipart/form-data">

```

```
@csrf
```

```
<div style="margin-bottom: 10px;">
  <label for="title">Назва новини</label>
  <input value="" type="text" name="title" id="title" class="block
min-w-0 grow py-1 pr-3 pl-1 text-base text-gray-900 placeholder:text-gray-400
focus:outline-none sm:text-sm/6" style="margin: 0 auto; width: 50%;">
</div>
```

```
<div style="margin-bottom: 10px;">
  <label for="description">Коротки опис</label>
  <textarea value="" type="text" name="description" id="description"
class="block min-w-0 grow py-1 pr-3 pl-1 text-base text-gray-900 placeholder:text-
gray-400 focus:outline-none sm:text-sm/6" style="margin: 0 auto; width:
50%;"></textarea>
</div>
```

```
<div style="margin-bottom: 10px;">
  <label for="text">Текст новини</label>
  <textarea value="" type="text" name="text" id="text" class="block
min-w-0 grow py-1 pr-3 pl-1 text-base text-gray-900 placeholder:text-gray-400
focus:outline-none sm:text-sm/6" style="margin: 0 auto; width: 50%;"></textarea>
</div>
```

```
<div style="margin-bottom: 10px;">
  <label for="date">Дата новини</label>
  <input value="" type="text" name="date" id="date" class="block
min-w-0 grow py-1 pr-3 pl-1 text-base text-gray-900 placeholder:text-gray-400
focus:outline-none sm:text-sm/6" style="margin: 0 auto; width: 50%;">
</div>
```

```

<div style="margin-bottom: 10px;">
  <label for="photo">Фото новини</label>
  <input value="" type="file" name="photo" id="photo" class="block
min-w-0 grow py-1 pr-3 pl-1 text-base text-gray-900 placeholder:text-gray-400
focus:outline-none sm:text-sm/6" style="margin: 0 auto; width: 50%;">
</div>

```

```

<div style="margin-bottom: 10px;">
  <label for="slug">Назва URL (slug)</label>
  <input value="" type="text" name="slug" id="slug" class="block
min-w-0 grow py-1 pr-3 pl-1 text-base text-gray-900 placeholder:text-gray-400
focus:outline-none sm:text-sm/6" style="margin: 0 auto; width: 50%;">
</div>

```

```

<button class="inline-flex items-center px-4 py-2 border border-
transparent rounded-md font-semibold text-xs text-white uppercase tracking-widest
hover:bg-gray-700 active:bg-gray-900 focus:outline-none focus:border-gray-900
focus:ring ring-gray-300 disabled:opacity-25 transition ease-in-out duration-150"
style="background-color: green;">Додати новину</button>

```

```

  </form>
</div>
</div>
</div>
</div>
</x-app-layout>

```

Код сторінки редагування новин

```

<x-app-layout>
  <x-slot name="header">
    <div class="flex justify-between items-center">
      <h2 class="font-semibold text-xl text-gray-800 leading-tight">
        {{ __('Редагувати новину') }}
      </h2>
    </div>
  </x-slot>

  <div class="py-12">
    <div class="max-w-7xl mx-auto sm:px-6 lg:px-8">
      <div class="bg-white overflow-hidden shadow-sm sm:rounded-lg">
        <div class="p-6 bg-white border-b border-gray-200">
          <form action="{{ route('newss.update', ['newss' => $news]) }}"
method="POST" class="text-center" style="margin-bottom: 15px;">
            @csrf
            @method('PUT')
            <div style="margin-bottom: 10px;">
              <label for="title">Назва новини</label>
              <input value="{{ $news->title }}" type="text" name="title"
id="title" class="block min-w-0 grow py-1 pr-3 pl-1 text-base text-gray-900
placeholder:text-gray-400 focus:outline-none sm:text-sm/6" style="margin: 0 auto;
width: 50%;">
            </div>

            <div style="margin-bottom: 10px;">
              <label for="description">Коротки опис</label>

```

```

        <textarea type="text" name="description" id="description"
class="block min-w-0 grow py-1 pr-3 pl-1 text-base text-gray-900 placeholder:text-
gray-400 focus:outline-none sm:text-sm/6" style="margin: 0 auto; width: 50%;">{{
$news->description }}</textarea>

```

```

    </div>

```

```

    <div style="margin-bottom: 10px;">

```

```

        <label for="text">Текст новини</label>

```

```

        <textarea type="text" name="text" id="text" class="block min-w-
0 grow py-1 pr-3 pl-1 text-base text-gray-900 placeholder:text-gray-400
focus:outline-none sm:text-sm/6" style="margin: 0 auto; width: 50%;">{{ $news-
>text }}</textarea>

```

```

    </div>

```

```

    <div style="margin-bottom: 10px;">

```

```

        <label for="date">Дата новини</label>

```

```

        <input value="{{ $news->date }}" type="text" name="date"
id="date" class="block min-w-0 grow py-1 pr-3 pl-1 text-base text-gray-900
placeholder:text-gray-400 focus:outline-none sm:text-sm/6" style="margin: 0 auto;
width: 50%;">

```

```

    </div>

```

```

    <div style="margin-bottom: 10px;">

```

```

        <label for="slug">Назва URL (slug)</label>

```

```

        <input value="{{ $news->slug }}" type="text" name="slug"
id="slug" class="block min-w-0 grow py-1 pr-3 pl-1 text-base text-gray-900
placeholder:text-gray-400 focus:outline-none sm:text-sm/6" style="margin: 0 auto;
width: 50%;">

```

```

    </div>

```

```

        <button class="inline-flex items-center px-4 py-2 border border-
transparent rounded-md font-semibold text-xs text-white uppercase tracking-widest
hover:bg-gray-700 active:bg-gray-900 focus:outline-none focus:border-gray-900
focus:ring ring-gray-300 disabled:opacity-25 transition ease-in-out duration-150"
style="background-color: #0063ff;">Редагувати новину</button>
    </form>
</div>
</div>
</div>
</div>
</div>
</x-app-layout>

```

Код сторінки короткого перегляду новини через панель адміністратора

```

<x-app-layout>
  <x-slot name="header">
    <div class="flex justify-between items-center">
      <h2 class="font-semibold text-xl text-gray-800 leading-tight">
        {{ __( 'Новина' ) }}
      </h2>
    </div>
  </x-slot>
  <div class="py-12">
    <div class="max-w-7xl mx-auto sm:px-6 lg:px-8">
      <div class="bg-white overflow-hidden shadow-sm sm:rounded-lg">

```

```

<div class="p-6 bg-white border-b border-gray-200">
  <div class="flex justify-between">
    <p style="font-weight: 700">{{ $news->date }}</p>
    <p style="font-weight: 700">{{ $news->slug }}</p>

  </div>

  <h1><span style="font-weight:700;"> Назва:</span> {{ $news->title
}}</h1>

  <p style="margin: 15px 0;"><span style="font-weight:700;">Короткий
опис:</span> {{ $news->description }}</p>

  <p><span style="font-weight:700;">Текст новини:</span> {{ $news-
>text }}</p>

  </div>
</div>
</div>
</div>
</div>
</x-app-layout>

```