

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування інституту, назва факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему РОЗРОБКА ОНЛАЙН «КУЛІНАРНОЇ КНИГИ»
ONLINE COOKBOOK DEVELOPMENT

Виконав: студент(ка) 4 курсу, групи КНТ-138
Спеціальності 121 Інженерія програмного
забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

Лях А.А.

(прізвище та ініціали)

Керівник Колпакова Т.О.

(прізвище та ініціали)

Рецензент Скрупський С.Ю.

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування закладу вищої освіти)

Інститут, факультет ФКНТ
 Кафедра програмних засобів
 Ступінь вищої освіти бакалавр
 Спеціальність 121 Інженерія програмного забезпечення
 (код і найменування)
 Освітня програма (спеціалізація) Інженерія програмного забезпечення
 (назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
 “ ” 2022 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Лях Анастасії Артемівни
 (прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Розробка онлайн кулінарної книги. Online Cookbook Development.
- керівник проєкту (роботи) Колпакова Тетяна Олексіївна, к.т.н., доцент,
 затверджені наказом закладу вищої освіти від “ 27 ” квітня 2022 року № 102
2. Строк подання студентом проєкту (роботи) 07 червня 2022 року
3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області і обґрунтування структури проєктованої системи. 2. Розробка архітектури програми. 3. Основні рішення щодо реалізації компонентів системи. 4. Експлуатація та тестування розробленої програми
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Слайди презентації.

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	Колпакова Т.О., доцент		
Нормоконтроль	Липовець М.В., асистент		

7. Дата видачі завдання “ 04 ” квітня 2022 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки.	2 тиждень	Розділ 2
4	Розробка архітектури програми.	2 тиждень	Розділ 3
5	Розробка програми.	3-4 тижні	Розділ 3, 4
6	Тестування та експериментальне дослідження програмного забезпечення.	5 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	6 тиждень	Додатки
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)

(підпис) Лях А.А.
(прізвище та ініціали)

Керівник проєкту (роботи)

(підпис) Колпакова Т.О.
(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 93 с., 32 рис., 2 табл., 2 дод., 15 джерел.

ПОШУК РЕЦЕПТІВ, СТВОРЕННЯ РЕЦЕПТІВ, ФРЕЙМВОРК, PHP, LARAVEL, PHPSTORM, CSS, SQL, HTTP, IDE, MVC.

Об'єкт дослідження – процес зберігання, відтворення та пошуку рецептів.

Предмет дослідження – програмне забезпечення для пошуку та зберігання рецептів.

Мета роботи – створення вебсайту для перенесення паперової кулінарної книги у мережу Інтернет та її вдосконалення для сучасних користувачів.

Матеріали, методи та технічні засоби: об'єктно-орієнтоване програмування, мова програмування PHP, фреймворк Laravel, база даних MySQL, система керування версіями Git, мова розмітки гіпертексту HTML, мова стилю сторінок CSS, ноутбук з процесором Intel Core i3 під управлінням операційної системи Ubuntu.

Результати. Створено максимально оптимізований вебсайт на основі фреймворку Laravel та серверного REST API, за допомогою якого користувач може публікувати власні рецепти, шукати нові та зберігати їх для подальшого використання.

Висновки. Розроблено вебсайт для опублікування нових рецептів, пошуку рецептів за вподобаннями та загальної модернізації «кулінарної книги» кожної родини.

Галузь використання – повсякденне життя людини віком від 14 років.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 93 pages, 32 figures, 2 tables, 2 appendixes, 15 sources.

RECIPE SEARCH, RECIPE CREATION, FRAMEWORK, PHP, LARAVEL, PHPSTORM, CSS, SQL, HTTP, IDE, MVC.

The object of research is the process of storing, reproducing and finding recipes.

The subject of research is the software for finding and storing recipes.

The purpose of the work – the creation of a website for transferring paper cookbook to the Internet and its improvement for modern users.

Materials, methods and technical means: object-oriented programming, PHP programming language, Laravel framework, MySQL database, Git version control system, HTML hypertext markup language, CSS style language, a laptop with an Intel Core i3 processor running Ubuntu operating system.

Results. The most optimized website based on Laravel frameworks and the server REST API which allows user to publish own recipes, add new ones and save them for future use was created.

Conclusions. A website for publishing new recipes, finding recipes by preferences and general modernization of each family's "cookbook" has been developed.

The field of use is the daily life of a person over 14 years of age.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок	8
Вступ	9
1 Аналіз предметної області і обґрунтування структури проектованої системи.....	11
1.1 Аналіз предметної області.....	11
1.2 Огляд існуючих методів вирішення завдання	11
1.2.1 Вебсайт «Russian food».....	11
1.2.2 Вебсайт «Гастрономъ»	12
1.2.3 Вебсайт «Picante Cooking»	13
1.2.4 Вебсайт «Smachno.in.ua».....	14
1.3 Порівняння функціоналу існуючих рішень	15
1.4 Постановка задачі.....	16
1.5 Висновки за розділом 1	17
2 Розробка архітектури програми	18
2.1 Обґрунтування вибору мови програмування	18
2.1.1 Мова тегів HTML	18
2.1.2 Мова стилю сторінок CSS	19
2.1.3 CSS фреймворк Bootstrap	20
2.1.4 Мова програмування PHP	21
2.2 Вибір фреймворку.....	23
2.2.1 Laravel	24
2.2.2 CodeIgniter.....	24
2.2.3 Symfony	25
2.2.4 CakePHP	25
2.2.5 Yii	26

	7
2.2.6 Zend	26
2.2.7 Phalcon.....	27
2.2.8 FuelPHP	27
2.2.9 PHPixie	28
2.3 Обґрунтування вибору фреймворку	28
2.4 Вибір середовища розробки	31
2.5 Висновки за розділом 2.....	33
3 Основні рішення щодо реалізації компонентів системи.....	34
3.1 Налаштування оточення	34
3.2 Структура проєкту	36
3.3 Структура бази даних	37
3.4 Висновки за розділом 3.....	39
4 Експлуатація та тестування розробленої програми	40
4.1 Основні розроблені функції	40
4.2 Реєстрація та авторизація користувачів.....	40
4.3 Головна сторінка	42
4.3.1 Секція останніх доданих рецептів	42
4.3.2 Секція найпопулярніших рецептів	42
4.4 Сторінка перегляду рецепту	43
4.5 Сторінка користувача	44
4.6 Створення рецепту	44
4.7 Пошук рецептів	45
4.8 Висновки за розділом	46
Висновки.....	47
Перелік джерел посилання	48
Додаток А Технічне завдання.....	50
Додаток Б Текст програми.....	54
Додаток В Слайди презентації	86

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

- API – Application Programming Interface;
- CRUD – Create Read Update Delete;
- CSS – Cascading Style Sheets;
- HTML – HyperText Markup Language;
- HTTP – Hyper Text Transfer Protocol;
- IDE – Integrated Development Environment;
- MVC – Model View Controller;
- PHP – Hypertext Preprocessor;
- REST – Representational State Transfer;
- SOLID – Single responsibility, Open–closed, Liskov substitution, Interface segregation и Dependency inversion;
- SQL – Structured Query Language;
- OC – операційна система.

ВСТУП

Ця робота присвячена вивченню та аналізу сучасного ринку вебсайтів та сервісів на кулінарну тематику. Основна її мета – осучаснення цієї галузі та допомога, у першу чергу, старшому поколінню зберегти та передати свої знання.

Усе нове – це дуже добре забуте старе. У цьому дуже легко переконатися, якщо згадати усе, чим раніше користувались і порівняти це з тим, що зараз вас оточує.

Ні в якому випадку це не означає, що не потрібно осучаснювати себе своє оточення. Досить буде просто не забувати першоджерело і не втрачати історичний зв'язок з минулим. Родина – це, безумовно, невід'ємна частка кожної людини. У кожній родині є власні звичаї, традиції, реліквії, які передаються з покоління в покоління. Саме у таких дрібницях і відчувається справжній зв'язок з предками.

У наш час, час технологій, підтримувати зв'язок та ділитися чимось можна з будь-якої точки світу, і це дуже зручно. Проте наші бабусі й дідусі не дуже добре розуміють сучасні девайси та технології, але саме вони є хранителями сімейних таємниць. Саме у них зберігаються книги та записи з усіма секретами, наприклад, смачних страв, які збирають усю родину за столом.

Вкрай важливо осучаснити та дати нове життя цим великим книжкам та зошитах з купою сімейних рецептів різних страв. Більшість з них вже ледве дихають і буквально розсипаються на очах. Багато з них втратили декілька сторінок, і тому рецепти не вдасться відтворити вже ніколи. Можна було б відсканувати сторінки, спробувати зберігати більш акуратно, але ж, я впевнена, що багато хто намагався і урешті-решт нічого не вийшло.

Для розв'язання цієї проблеми слід модернізувати підхід до зберігання та передачі таких цінних речей, проте не так багато вебсайтів можуть надати дійсно якісний сервіс. Коли дизайн сайту або додатку виглядає неактуально і

незручно, то і довіри він не викликає. Нажаль, вибір сервісів у цій галузі не такий великий, і це, безумовно, проблема, яку треба виправляти. Тож проблема розробки онлайн кулінарної книги зі зручним інтерфейсом є актуальною задачею.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ОБҐРУНТУВАННЯ СТРУКТУРИ ПРОЄКТОВАНОЇ СИСТЕМИ

1.1 Аналіз предметної області

У ХХІ столітті відбувся стрімкий розвиток у галузі технологій та Інтернету. Ще тридцять років тому, для передачі чи збереження будь-яких даних потрібно було мати при собі щонайменше: папір, ручку чи олівець та «надійне сховище», щоб запис не загубився або знищився. Нажаль, папір не надто надійний матеріал і деякі цим активно користувалися, бо багато важливих даних було втрачено саме через це. А щоб відправити комусь листа, потрібно було довго чекати поки пошта дійде, і ще більше, щоб отримати відповідь.

Зараз, навчившись на своїх помилках, люди активно оцифровують усі старі записи та створюють нові, одразу завантажуючи їх у мережу чи зберігають їх на фізичних носіях. Таким чином, ми маємо змогу безпечно зберігати, швидко обмінювати чи відновлювати видалені раніше дані.

Для кожної людини важливість тієї чи іншої інформації – річ індивідуальна. Не усі сучасні сервіси можуть надати якісний, зрозумілий та дійсно корисний продукт. Деякі не актуалізують дані, не оновлюють інтерфейс, перестають відповідати потребам користувача. Також проблемою є відсутність альтернативних варіантів для більш вузько направлених сервісів.

1.2 Огляд існуючих методів вирішення завдання

1.2.1 Вебсайт «Russian food»

RussianFood.com (рис. 1.1) – це один з найпопулярніших вебсайтів для пошуку рецептів за різною тематикою [1].

Переваги програмного застосунку полягають у декількох речах, а саме:

- великий вибір фільтрів для пошуку;
- зрозумілий інтерфейс;

- наявна покрокова інструкція з фото чи\та відео;
- багато контенту;
- реклама є, але не заважає користувачу.

Але цей вебсайт має ряд своїх недоліків:

- застарілий дизайн;
- багато візуального шуму, через який користувач просто розсіює свою увагу одразу на все і не може сфокусуватися.



Рисунок 1.1 – Вебсайт “RussianFood.com”

1.2.2 Вебсайт «Гастрономъ»

Гастроном (рис. 1.2) – це вебсайт для пошуку не тільки рецептів, а й корисних статей на різні теми [2].

Переваги програмного застосунку полягають у декількох речах, а саме:

- легкий та стриманий дизайн;
- наявність корисної інформації для будь-якого користувача;
- наявність опитувань та зворотного зв'язку з користувачем.

Але цей вебсайт має ряд своїх недоліків:

- багато візуального шуму, через який користувач просто розсіює свою увагу одразу на все і не може сфокусуватися;
- не дуже зрозумілий інтерфейс для користувача;
- велика кількість реклами.

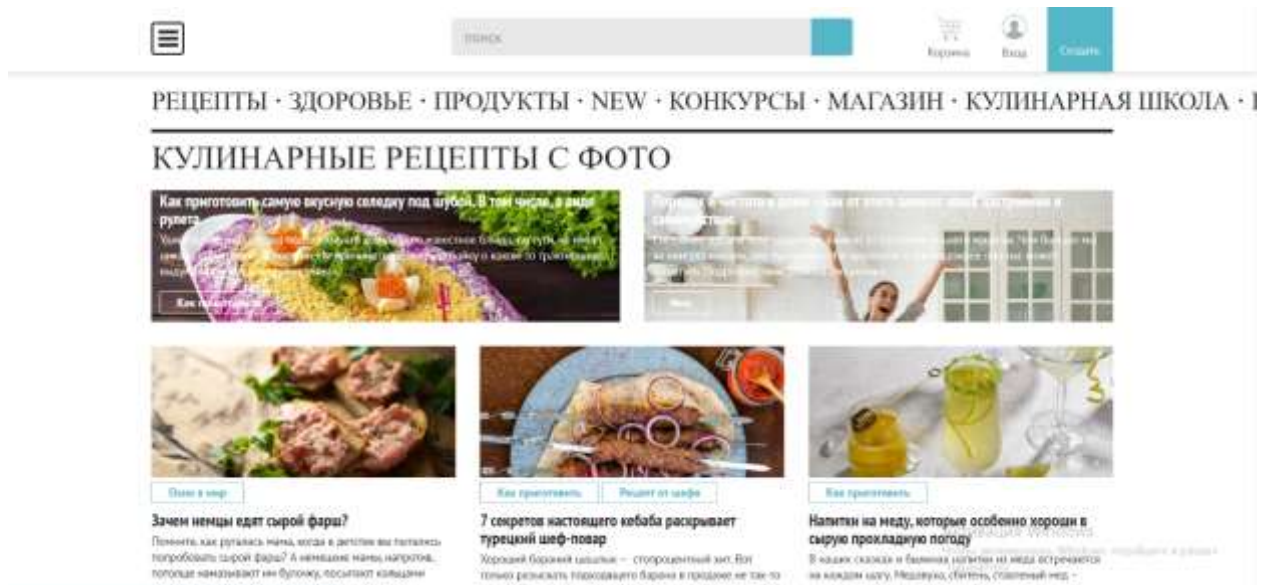


Рисунок 1.2 – Вебсайт “Гastronom”

1.2.3 Вебсайт «Picante Cooking»

Picante Cooking (рис. 1.3) – це авторський кулінарний сайт Вікторії Попін. Цей сайт вона створила для того, щоб ділитися своїми рецептами, статтями, книгами та цікавими новинами про їжу [3].

Переваги програмного застосунку полягають у декількох речах, а саме:

- цікавий дизайн;
- гарно підібрана кольорова гамма;
- зручне і зрозуміле розташування контенту;
- гарне співвідношення тексту і фото;
- є можливість перекладу сайту на різні мови;
- є посилання на соціальні мережі.

Із недоліків можу виділити лише наявність агресивної реклами, яка заважає при перегляді рецептів і статей.

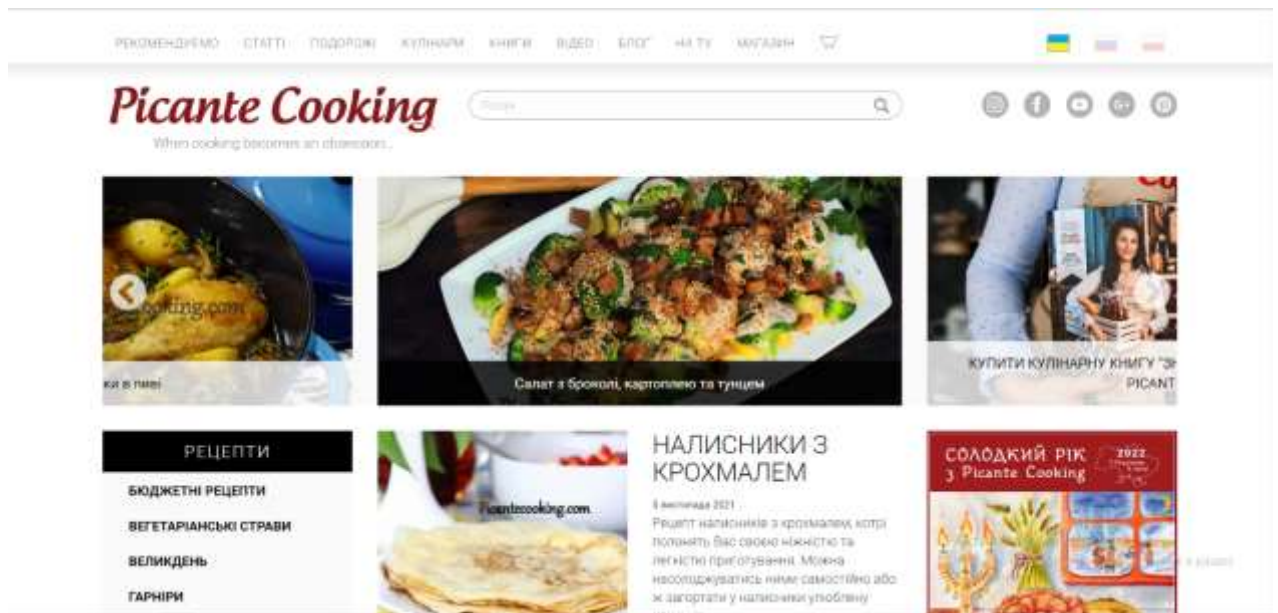


Рисунок 1.3 – Вебсайт “Picante Cooking”

1.2.4 Вебсайт «Smachno.in.ua»

Smachno.in.ua (рис. 1.4) – це авторський кулінарний сайт української кухарки [4]. Вона створила цей сайт для того, щоб показувати свої експерименти з їжею, знаходити нові смаки та ділитися цим з іншими людьми.

Переваги програмного застосунку полягають у декількох речах, а саме:

- гарне розташування контенту;
- великі фото у публікаціях;
- гарна зйомка страв;
- є посилання на відео;
- є можливість перекласти на іншу мову;
- зрозумілий інтерфейс для користувача.

Але цей вебсайт має ряд своїх недоліків:

- не дуже привабливий дизайн;
- під час переходу по розділам уся сторінка «стрибає».



Рисунок 1.4 –Вебсайт “Smachno.in.ua”

1.3 Порівняння функціоналу існуючих рішень

Проаналізувавши існуючі рішення реалізації застосунків та вебсайтів для зберігання та знаходження нових рецептів, було виділено основні переваги та недоліки усіх функціональних особливостей існуючого програмного забезпечення (табл. 1.1).

Таблиця 1.1 – Порівняльна характеристика найпоширеніших програмних продуктів

Функціональні можливості	Назва програмних продуктів			
	RussianFood .com	Гастроном	Picante Cooking	Smachno.in. ua
1	2	3	4	5
Зручний дизайн	-	+	+	-
Легкочитаємий текст	+	-	+	+
Пошук за ключовою фразою	+	+	+	+
Відсутність реклами	+	-	-	+

Продовження таблиці 1.1

1	2	3	4	5
Зручна фільтрація рецептів	+	+	+	+
Оновлення дизайну та контенту	-	+	+	-
Гарне співвідношення тексту та фото	+	-	+	+
Наявність соціальних мереж	-	-	+	-
Наявність різних типів контенту	-	+	+	+
Додатковий функціонал	-	+	+	+

Слід розробити програмний продукт, в якому буде усунуто такі недоліки конкурентів, як, наприклад, вдалий дизайн, відсутність реклами, вчасне оновлення сайту за візуальною складовою та контентом. В проєкті також буде реалізовано такі критичні функції, як пошук по усьому контенту, врахування вподобань та алергенів людини.

1.4 Постановка задачі

Програмний продукт призначений для перегляду та доповнення рецептами власної кулінарної книги. Будь-який користувач зможе зареєструватися на сайті та створити власну збірку як зі своїх рецептів, статей та порад, так і додаючи ці записи зі збірок інших людей.

Безумовно, буде враховано наявність алергенів або продуктів, які не подобаються або, навпаки подобаються користувачеві. Цю інформацію він зможе вказати при реєстрації акаунту. Відповідно до цих даних рецепти

будуть рекомендовані до перегляду або й зовсім відсутні у рекомендаціях людини.

Також в програмі буде реалізовано інші дані та функції:

- зручний та зрозумілий інтерфейс;
- врахування вподобань користувача;
- особлива увага буде приділена алергенам користувача;
- відсутність будь-якої реклами;
- наявність не тільки рецептів, а й іншого пізнавального контенту.

1.5 Висновки за розділом 1

Проаналізувавши методи та програмні засоби реалізації вебзастосунків кулінарних сайтів можна сказати, що наразі не існує гнучкого та зручного програмного засобу для пошуку, перегляду та внесення власних рецептів, статей та порад, які б стовідсотково задовільнили користувачів.

Саме тому актуальною є розробка нового програмного продукту, що має поєднати в собі функціонал, який є у існуючих альтернативах, та стати повноцінним інструментом для зберігання, використання та поширення інформації, усунувши недоліки, які є в розглянутих рішеннях.

2 РОЗРОБКА АРХІТЕКТУРИ ПРОГРАМИ

2.1 Обґрунтування вибору мови програмування

Для розробки вебзастосунку була обрана мова програмування PHP. Для створення сторінок та додавання стилістики було обрано стандартизовані мови HTML5 та CSS3. Також для зберігання даних було використано реляційну систему управління базами даних – MySQL.

2.1.1 Мова тегів HTML

HTML (HyperText Markup Language) – це мова тегів, за допомогою якої здійснюється розмітка вебсторінок для Інтернету. Елементи HTML – це будівельні блоки сторінок HTML. За допомогою конструкцій HTML, зображення та інші об'єкти, такі як інтерактивні форми можуть бути вбудовані у візуалізовану сторінку [5]. HTML надає засоби створення структурованих документів, позначаючи структурну семантику тексту, наприклад, заголовки, абзаци, списки, посилання, цитати та інші елементи.

За своєю суттю HTML – це серія коротких кодів, введених у текстовий файл. Це теги, які забезпечують можливості HTML. Текст зберігається як файл HTML і переглядається через браузер. Браузер читає файл і перекладає текст у видиму форму відповідно до кодів, використаних автором для написання того, що стає видимим відображенням. Написання HTML вимагає правильного використання тегів для створення бачення автора. Теги відрізняють звичайний текст від HTML-коду. Теги – це слова між так званими кутовими дужками, які дозволяють відображати графіку, зображення та таблиці на вебсторінці. Різні теги виконують різні функції. Найпростіші теги застосовують форматування до тексту. Оскільки вебінтерфейси повинні стати більш динамічними, можна використовувати каскадні таблиці стилів (Cascading Style Sheets, CSS) і програми JavaScript. CSS робить вебсторінки доступнішими, а JavaScript додає потужності базовому HTML.

2.1.2 Мова стилю сторінок CSS

CSS – це спеціальна мова стилю сторінок, яка використовується для опису їх зовнішнього вигляду [6]. CSS використовується авторами та відвідувачами вебсторінок, щоб визначити кольори, шрифти, верстку та інші аспекти сторінки. Однією з головних переваг є можливість розділити зміст сторінки (або контент, наповнення, зазвичай HTML, Extensible Markup Language або подібну мову розмітки) від виду документа (який описується в CSS).

До розробки CSS у 1996 році Консорціумом World Wide Web Consortium, вебсторінки були надзвичайно обмеженими як за формою, так і за функціями. Ранні браузери представляли сторінку як гіпертекст – звичайний текст, зображення та посилання на інші гіпертекстові сторінки. Не було жодного макета, про який можна було б говорити, лише абзаци, що йдуть по всій сторінці в одній колонці.

CSS дозволив впровадити кілька нововведень у макеті вебсторінки, наприклад, можливість:

- вказувати шрифти, відмінні від стандартних для браузера;
- вказувати колір і розмір тексту та посилань;
- застосовувати кольори для фону;
- містити елементи вебсторінки в полях і переміщувати ці поля на певні позиції на сторінці.

World Wide Web Consortium помістили «стиль» у таблиці стилів, і вперше можна було розробляти вебсторінки.

Щоб зрозуміти основи роботи CSS, користувач повинен спочатку трохи зрозуміти сучасний HTML. Веброзробники розкладають сторінки відповідно до «коробкової моделі» [7]. Вебсторінка — це серія блоків, кожна з яких містить дискретний елемент. Ці коробки вкладені один в інший.

HTML-код забезпечує правильне форматування тексту та зображень для Інтернет-браузера. Без HTML браузер не знав би, як відображати текст як

елементи, або завантажувати зображення чи інші елементи. HTML також забезпечує базову структуру сторінки, на яку накладаються каскадні таблиці стилів, щоб змінити її зовнішній вигляд. Можна було б думати про HTML як про кістки (структуру) вебсторінки, а CSS як про її шкіру (зовнішній вигляд).

2.1.3 CSS фреймворк Bootstrap

Bootstrap – це потужний набір інструментів HTML, CSS та JavaScript для створення та створення вебсторінок та вебдодатків.

Його основні переваги полягають у тому, що він адаптивний за дизайном, підтримує широку сумісність з браузером, пропонує послідовний дизайн за допомогою компонентів, які можна повторно використовувати, і він дуже простий у використанні та швидко навчається [8]. Він пропонує широку розширюваність за допомогою JavaScript, поставляється з вбудованою підтримкою плагінів jQuery і програмним API JavaScript. Bootstrap можна використовувати з будь-якою IDE (Integrated Development Environment) або редактором, а також з будь-якою технологією та мовою на стороні сервера, від Active Server Pages для .NET до PHP і Ruby on Rails.

Bootstrap – це не просто набір готових інструментів (HTML фрагментів, класів, компонентів та плагінів), а добре спроектований фронтенд фреймворк, який досить просто можна налаштувати під себе за допомогою редагування Sass змінних та використання міксинів.

Bootstrap складається з:

- інструментів для створення макета (обгорткових контейнерів, потужної системи сіток, гнучких медіа-об'єктів, адаптивних утилітних класів);
- класів для стилізації базового контенту: тексту, зображень, коду, таблиць та figure;
- готових компонентів: кнопок, форм, горизонтальних і вертикальних навігаційних панелей, слайдерів, списків, акордеонів, модальних вікон, спливаючих підказок та ін;

– утилітних класів для вирішення традиційних завдань, що найбільш часто виникають перед веброзробниками: вирівнювання тексту, відображення та приховування елементів, завдання кольору, фону, зовнішніх та внутрішніх відступів, і т.д.

За допомогою Bootstrap веброзробники можуть зосередитися на роботі з розробки, не турбуючись про дизайн, і швидко створити гарний вебсайт. І навпаки, це дає вебдизайнерам міцну основу для створення цікавих тем Bootstrap.

2.1.4 Мова програмування PHP

PHP (рекурсивний акронім словосполучення PHP: Hypertext Preprocessor) – це поширена мова програмування загального призначення з відкритим вихідним кодом. PHP спеціально сконструйований для веброзробок, і його код може впроваджуватися безпосередньо в HTML.

В даний час PHP є одним з лідерів серед серверних мов програмування, які застосовуються для створення динамічних вебсайтів та вебдодатків. Більшість коробочних систем управління сайтами написана саме на PHP, мова підтримується переважною більшістю хостинг-провайдерів, мова набула широкого поширення завдяки своїй простоті, швидкості, мультипарадигмальності, багатій функціональності та кросплатформенності.

PHP має деякі переваги, які зробили його таким популярним, і він є основною мовою для вебсерверів уже більше 15 років. Ось деякі з переваг PHP [9]:

– міжплатформенність. PHP не залежить від платформи. Вам не потрібно мати певну ОС (Операційна Система), щоб використовувати її, оскільки вона працює на кожній платформі, будь то Mac, Windows чи Linux;

– open source. PHP є відкритим вихідним кодом. Оригінальний код доступний для всіх, хто хоче розробити його. Це одна з причин, чому одна з її фреймворків, Laravel, настільки популярна;

- легко освоїти. PHP не важко вивчити для абсолютних початківців. Ви можете опанувати його легко, якщо у вас вже є знання програмування;

- PHP синхронізується з усіма базами даних. ви можете легко підключити PHP до всіх баз даних, реляційних і нереляційних. Тому він може швидко підключитися до MySQL, Postgress, MongoDB або будь-якої іншої бази даних;

- підтримуюча спільнота. PHP має дуже сприятливу онлайн-спільноту;

- офіційна документація містить інструкції щодо використання функцій, і ви можете легко вирішити проблему.

Мова PHP використовується для:

- автоматичне вилучення змінних оточення сервера, параметрів GET та POST у зумовлені масиви;

- взаємодія з різними системами керування базами даних (SQLite, MySQL, Microsoft SQL Server, dBase та багатьма іншими);

- відправка HTTP (Hyper Text Transfer Protocol) заголовків в автоматичному режимі (полегшує пошукову оптимізацію сайту);

- обробка файлів, що завантажуються на сервер;

- робота з віддаленими та локальними файлами, сесіями, сокетом, HTTP авторизацією, cookies, XForms.

Особливості мови [10]:

- мова виконує код всередині обмежувачів `<?php ?>`;

- дані поза цими тегами виводяться у стандартній розмітці HTML документа;

- ім'я змінної починається з символу `$` і чутливе до регістру, як і імена класів, функцій та константи;

- змінні обробляються в ув'язнених у подвійні лапки або апострофи рядках;

– значною відмінністю PHP від будь-якого коду, що виконується на стороні клієнта, наприклад JavaScript, є те, що PHP-скрипти виконуються на стороні сервера.

Як працює PHP:

- спочатку веббраузер надсилає запит HTTP на веб сервер, наприклад, index.php;
- по-друге, препроцесор PHP, який знаходиться на веб сервері, обробляє PHP-код для створення HTML-документа;
- по-третє, вебсервер надсилає HTML-документ назад у веб браузер.

2.2 Вибір фреймворку

У програмуванні фреймворки розширюють структуру, яка потрібна для створення універсального програмного застосування. Вони оптимізують процес розробки програмного забезпечення, підключаючи базові функції програми до вашої програми ще до того, як ви почнете писати код.

Є кілька речей, на які слід звертати увагу при виборі PHP-фреймворку:

- популярність та розмір спільноти. Фреймворки часто керуються спільнотою, тому наявність спільноти для підтримки та покращення фреймворку є величезною позитивною перевагою;
- філософія. Якщо цілі фреймворку не відповідають потребам та цілям вашого бізнесу, можливо, слід пошукати альтернативне рішення;
- стійкість. Чи часто підтримується фреймворк і чи підтримуватиме він ваш технологічний стек як зараз, так і в майбутньому;
- підтримка. Куди ви можете звернутися, якщо у вас виникнуть питання чи проблеми. Часто цей момент пов'язаний із критеріями популярності та розміру спільноти. Документація та онлайн-ресурси повинні бути доступні, щоб допомогти вам упоратися з будь-якими проблемами, з якими ви можете зіткнутися.

Нижче розглянуто найкращі на теперішній час фреймворки PHP.

2.2.1 Laravel

Представлений у 2011 році Laravel став найпопулярнішим у світі безкоштовним PHP-фреймворком з відкритим вихідним кодом, тому що він може безпечно обробляти складні вебпрограми значно швидше, ніж інші фреймворки [11]. Laravel спрощує процес розробки, полегшуючи загальні завдання, такі як маршрутизація, сесии, кешування та аутентифікація. Laravel підходить для розробки додатків зі складними внутрішніми вимогами, чи то маленькими, чи більшими. Установка Laravel стала простішою завдяки впровадженню Homestead. Homestead – це середовище PHP, повне функцій, які допоможуть вам налаштувати складні програми. Серед них є такі, як безшовна міграція даних, підтримка архітектури MVC (Model View Controller), безпека, маршрутизація, механізм шаблонів уявлень, автентифікація та багато іншого. Laravel дуже виразний, а його швидкість та безпека відповідають очікуванням сучасної вебпрограми.

2.2.2 CodeIgniter

Відомий своїми невеликими розмірами (всього близько 2 МБ, включаючи документацію) CodeIgniter придатний для розробки динамічних вебсайтів. Він пропонує численні готові модулі, які допомагають створювати надійні та багаторазові компоненти. CodeIgniter — це легкий та простий PHP-фреймворк, установка якого не викликає труднощів, на відміну від інших фреймворків. Завдяки простому процесу налаштування та добре ілюстрованій документації він ідеально підходить для початківців [12]. Основні характеристики включають архітектуру MVC, першокласну обробку помилок, вбудовані інструменти безпеки та просту й чудову документацію. Крім того, він створює масштабовані програми. У порівнянні з іншими фреймворками CodeIgniter значно швидше. Оскільки він також забезпечує високу продуктивність, це хороший вибір, якщо ви хочете розробляти легкі програми

для роботи на скромних серверах. Однак він має один недолік, випуски CodeIgniter є дещо нерегулярними, тому фреймворк не є чудовим варіантом для програми, яка вимагає високого рівня безпеки.

2.2.3 Symfony

Фреймворк Symfony був запущений у 2005 році, і хоча він існує набагато довше, ніж інші фреймворки з цього списку, це надійна та зріла платформа. Symfony — це розширений фреймворк PHP MVC і єдиний фреймворк, який, як відомо, повністю відповідає PHP та вебстандартам [13]. Symfony є ідеальним вибором для розробки масштабних корпоративних проєктів. Його легко встановити та налаштувати на більшості платформ. Одна з його ключових характеристик — багаторазові компоненти PHP. Він також може похвалитися незалежністю від механізму баз даних, є стабільним, відповідає більшості найкращих вебпрактик і шаблонів дизайну, а також дозволяє інтегруватися з бібліотеками інших постачальників. Symfony також дуже гнучкий і може інтегруватися з великими проєктами, такими як Drupal.

2.2.4 CakePHP

Якщо ви шукаєте простий та елегантний набір інструментів, CakePHP допоможе вам розробити візуально вражаючі, багатofункціональні вебсайти. Крім того, CakePHP є одним із найпростіших фреймворків для вивчення, особливо через його фреймворк CRUD (створення, читання, оновлення та видалення) [13]. CakePHP вийшов на ринок на початку 2000-х років, і з тих пір він отримав кращу продуктивність і багато нових компонентів. CakePHP простий і легкий в установці, оскільки для установки потрібен лише вебсервер і копія фреймворку. Він є хорошим вибором для комерційних додатків завдяки функціям безпеки, які включають запобігання ін'єкції SQL (Structured Query Language), перевірку введення, захист від підробки міжсайтових запитів

(Cross-Site Request Forgery) і захист міжсайтових сценаріїв (Cross-Site Scripting). Деякі ключові функції включають сучасну структуру, швидкі збірки, належне успадкування класів, перевірку та безпеку. Крім того, CakePHP надає чудову документацію, багато порталів підтримки та преміальну підтримку через Cake Development Corporation.

2.2.5 Yii

Це високопродуктивний PHP-фреймворк на основі компонентів для розробки для всіх видів вебпрограм. З цієї причини це універсальний фреймворк вебпрограмування [14]. Yii має простий процес встановлення. Крім того, його надійні функції безпеки роблять фреймворк придатним для високобезпечних проєктів, таких як проєкти електронної комерції, портали, Content Management System, форуми та багато інших. Він може похвалитися чудовою швидкістю та продуктивністю, він дуже розширюваний і дозволяє розробникам уникнути складності написання повторюваних операторів SQL, оскільки вони можуть моделювати дані бази даних у термінах об'єктів. Yii має основну команду розробників і експертів, які сприяють її розвитку. Оскільки його використовує велика спільнота, ви можете публікувати проблеми на форумах Yii та отримати допомогу. Yii надзвичайно розширюваний, і ви можете налаштувати майже кожен фрагмент коду ядра. Однак, якщо ви використовуєте його вперше, будьте готові до довгого навчання.

2.2.6 Zend

Framework Zend — це повна об'єктно-орієнтована структура, і той факт, що вона використовує такі функції, як інтерфейси та успадкування, робить її розширюваною. Він був побудований на основі agile методології, яка допомагає вам надавати високоякісні програми корпоративним клієнтам [13]. Zend легко налаштовується і дотримується найкращих практик PHP —

важливий момент для розробників, які хочуть додати функціональні можливості для конкретного проєкту. Framework Zend чудово підходить для складних проєктів на рівні підприємства. Це найкраща структура для великих ІТ-відділів і банків. Деякі ключові функції включають компоненти MVC, простий хмарний API, шифрування даних та керування сесіями. Він може інтегруватися із зовнішніми бібліотеками, і дозволяє використовувати лише потрібні компоненти. Фреймворк Zend поставляється з надзвичайно гарною документацією та має велику базу спільноти.

2.2.7 Phalcon

PHP фреймворк повного стеку, який використовує шаблон проєктування вебархітектури MVC, Phalcon спочатку був написаний на C і C++ і випущений у 2012 році [15]. Phalcon легко встановити, і він підходить для створення висококонфігурованих вебдодатків, які відповідають інструкціям із розвитку підприємства. Основні функції включають підвищену швидкість виконання, управління активами, універсальний автозавантажувач, а також найвищий рівень безпеки та кешування. На відміну від інших фреймворків, Phalcon оптимізує продуктивність завдяки ефективному використанню пам'яті. Якщо необхідно створити надзвичайно швидкий вебсайт, слід обирати Phalcon.

2.2.8 FuelPHP

FuelPHP – це гнучкий, повностековий фреймворк PHP, вперше випущений у 2011 році [16]. Окрім підтримки шаблону проєктування MVC, він має власну версію, яка називається контролером представлення ієрархічної моделі (Hierarchical model–view–controller). З Hierarchical model–view–controller, на відміну від MVC, вміст не потрібно дублювати, щоб він міг відображатися на кількох сторінках. В результаті це споживає менше часу і

пам'яті. FuelPHP допомагає розробникам надавати наскрізні вебрішення, різноманітні за розміром і складністю. Основні функції включають реалізацію Hierarchical model–view–controller, реалізацію RESTful, систему маршрутизації URL-адрес, захист від вразливостей і систему кешування. Він розширює свій підхід до безпеки за межі звичайних заходів безпеки, що робить його чудовим варіантом, якщо безпека має вирішальне значення для розроблюваного проєкту. Оскільки FuelPHP є відносно новим, він пропонує менше підтримки і може представляти більш круту криву навчання.

2.2.9 RNPixie

Представлений у 2012 році і, як і FuelPHP, RNPixie реалізує шаблон проєктування Hierarchical model–view–controller. Його метою було створити високопродуктивну структуру для вебсайтів лише для читання. Легко розпочати роботу з RNPixie, який підходить для вебсайтів соціальних мереж, налаштованих вебдодатків і служб розробки вебпрограм. Основні функції включають архітектуру Hierarchical model–view–controller, стандартне Object-Relational Mapping (об'єктно-реляційне відображення), перевірку введення, можливості авторизації, аутентифікацію та кешування. RNPixie побудовано з використанням незалежних компонентів. З цієї причини його можна використовувати без самого каркаса. Нажаль, RNPixie має відносно мало модулів. Крім того, у ньому відсутня підтримка компонентів, незалежно створених із залежностей. Оскільки він відносно новий, він менш популярний і має меншу спільноту користувачів, ніж інші фреймворки.

2.3 Обґрунтування вибору фреймворку

З наведених вище фреймворків найбільш перспективними для поточного проєкту є Laravel, Symfony, Phalcon, Zend та Yii. Для остаточного

визначення розглянемо порівняльну таблицю з плюсами та мінусами кожного з них (табл 2.1).

Таблиця 2.1 – Порівняльна характеристика обраних фреймворків

Фреймворк	Переваги	Недоліки
1	2	3
Laravel	Найкраща Inversion of Control Зручна система міграції Тестування модуля Шаблонний двигун Blade Гнучка маршрутизація Багато способів створення REST API Швидкий розвиток Зручна документація Велика популярність Вбудована консоль налагодження з функцією стека викликів	Вам потрібно працювати з фасадами Laravel, якщо ви хочете мати доступ до всіх функцій, IDE не розпізнають методи та властивості в деяких класах і показують попереджувальне повідомлення Крива навчання кругіша, ніж у Yii2 Немає генерації вбудованого інтерфейсу
Symfony	Дуже схожий на Yii Вбудована підтримка Codeception дозволяє писати функціональні та приймальні тести Має компонент Ain't Markup Language, що є величезною перевагою для будь-якого фреймворка Велика спільнота розробників Багато готових до використання наборів модулів, які називаються пакетами Детальна і зрозуміла документація	Symfony використовує дуже страшний Object-Relational Mapping (Propel and Doctrine) і є ресурсомістким Круга крива навчання Забгато різноманітних об'єктів Містить синтаксис анотації

Продовження таблиці 2.1

1	2	3
Phalcon	Висока ефективність Багато функцій Підходить для середовищ високого навантаження Не вимагає багато операцій з файлами Не споживає багато ресурсів Ви можете використовувати свою улюблену бібліотеку бази даних та її елементи Бази даних обробляються через Object-Relational Mapping, що призводить до підвищення продуктивності Усі процеси відбуваються швидко, оскільки фреймворк має прямий доступ до внутрішніх структур PHP	Його вихідний код написаний на C Ще не дуже популярний Це розширення (це означає, що ви не зможете запустити програму на спільному хостингі) Є багато людей, які досі не мають уявлення про Phalcon
Zend	Спадкування класів Об'єктно-орієнтований Має готові до використання рішення для багатьох завдань Ви можете інтегрувати все, що забажаєте, з чим завгодно Інструменти інтернаціоналізації Підтримка спільноти розробників Гарна документація	Не дуже підходить для швидкого розвитку Повільніше, ніж деякі інші фреймворки (але все одно достатньо швидко для 90% вебсайтів, база даних завжди є вузьким місцем) Для освоєння потрібно багато часу Ресурсомісткий

Продовження таблиці 2.1

1	2	3
Yii	Інтуїтивно зрозуміла архітектура MVC Спрощує підтримку коду, який використовує загальну архітектуру та методи Велика спільнота розробників Економить час, витрачений на звичайні завдання, такі як перевірка форми та безпеки Легко налаштувати для кращої продуктивності Не ресурсомісткий Різні варіанти кешування Сторонні бібліотеки та класи можна легко інтегрувати Гарні засоби безпеки Поведінка, помічники, можливість розширити базову функціональність тощо Підходить для проєктів будь-якого масштабу та складності	Не дуже гнучка маршрутизація Фронтенд і бекенд бібліотеки сильно заплутані

2.4 Вибір середовища розробки

В якості середовища розробки було обрано JetBrains PhpStorm.

PHPStorm – це інтелектуальний редактор для PHP, HTML і JavaScript з аналізом коду «на льоту», автоматизованими інструментами рефакторингу для PHP та JavaScript. PHPStorm підтримує специфікацію PHP 5.3/5.4/5.5/5.6/7.0/7.1/8.0 (сучасні і традиційні проєкти), включаючи генератори, співпрограми, простори імен, замикання, типажі і синтаксис

коротких масивів. Є повноцінний редактор SQL з можливістю редагування результатів запитів.

Графічний налагоджувач PHP, реалізований у PHPStorm, підтримує умовні точки зупинки, відстеження значень та автоматичне налагодження окремих процедур. Тестовий модуль PHPUnit і графічний інтерфейс для виконання тестів підтримуються для тестування додатків. Під час редагування коду виділяються синтаксичні конструкції, виконується розширене форматування конфігурації, виявлення помилок у реальному часі та доповнення коду. Редактор PHPStorm враховує коментарі до коду після його заповнення, автоматично вибираючи оптимальне рішення проблеми. Рефакторинг PHP і редагування шаблонів гарантує зміну проєкту в найкоротші терміни. PHPStorm дозволяє візуалізувати код в ієрархічній формі та забезпечує швидку навігацію по всіх елементах.

За допомогою тестів PHPUnit можна швидко переглянути результати генерації коду для окремих блоків або всієї програми. Якщо тест не пройшов, продукт дозволяє переглянути окремі рядки коду, в яких була виявлена помилка. PHPStorm забезпечує налагодження JavaScript і надає широкий спектр функцій: пошук точки зупину в HTML і JavaScript, налаштування параметрів точки зупину, тестування синтаксису коду в реальному часі тощо.

Сотні інспекцій піклуються про перевірку коду, аналізуючи проєкт під час розробки. Підтримка PHPDoc, аранжувальника коду, конфігуратора формату коду з конфігурацією стилю коду та інших функцій допомагають розробникам писати акуратний і простий у підтримці код.

Підтримуються передові технології веброботи, включаючи HTML5, CSS, Syntactically Awesome Stylesheets, Less, Stylus, Compass, CoffeeScript, TypeScript, ECMAScript Harmony, шаблони Jade, Zen Coding, Emmet і, звісно, JavaScript.

PHPStorm включає в себе всі функції WebStorm (редактор HTML / CSS, редактор JavaScript) і додає повнофункціональну підтримку PHP і баз даних / SQL [17].

Серед основних переваг я можу відзначити такі:

- швидкість;
- відмінні стандарти кодування;
- швидке налаштування проєкту;
- бездоганний контроль версій;
- неперевершена організація проєкту.

За допомогою PHPStorm ви можете впорядкувати ваші найчастіше використовувані елементи проєкту в списки улюблених. Ви також можете створювати заглушки проєкту на основі численних шаблонів, включаючи Composer, Drupal, WordPress, HTML5 Boilerplate, Twitter Bootstrap, Foundation, Node.js Express, AngularJs тощо.

2.5 Висновки за розділом 2

В розділі було проведено аналіз мов та технологій, що використовуються для вебпроєктів. Проведено порівняння найпопулярніших PHP фреймворків.

Для створення вебзастосунку було обрано найефективніший та найсучасніший набір інструментів. Мовою програмування було обрано PHP, обраний фреймворк – Laravel та середовище розробки – PHPStorm від JetBrains.

3 ОСНОВНІ РІШЕННЯ ЩОДО РЕАЛІЗАЦІЇ КОМПОНЕНТІВ СИСТЕМИ

В попередньому розділі для розробки проєкту було обрано фреймворк Laravel. Перед початком роботи Laravel потребує попередніх налаштувань сервера та оточення.

3.1 Налаштування оточення

Для встановлення програмного продукту та його запуску на локальному сервері необхідно мати встановлений PHP та PHPStorm, та Composer. Також необхідно мати акаунт на GitLab.

Для початку необхідно налаштувати сервер Nginx. Для цього потрібно зайти у `etc/nginx/sites-available`, скопіювати конфігурації у тестовому файлі та перенести їх у свій, відповідно замінивши тестові данні на свої.

Треба повернутись у попередню папку і зайти у `sites-enabled`. Потім скопіювати створений файл та додати за тестовим прикладом. Тепер перейти у `etc` й додати домен у файл `hosts`.

Потрібно перезавантажити сервер і перевірити роботоспроможність, сайту вписавши у пошукову строку назву домену.

Тепер необхідно зклонувати проєкт у окрему папку за допомогою команди `git clone` і додати `ssh`-ключ. Наступним кроком потрібно встановити пакети Composer за допомогою команди `composer install`. Далі потрібно згенерувати ключ командою `php artisan key:generate`. Для того, щоб згенерувати `js`-файли потрібно виконати команду `npm run production`.

На даному етапі потрібно створити базу даних. Найзручніший спосіб це скористатися MySQL Workbench. Це уніфікований візуальний інструмент для архітекторів баз даних та розробників. MySQL Workbench забезпечує моделювання даних, розробку SQL та комплексні інструменти

адміністрування для конфігурації сервера, адміністрування користувачів, резервного копіювання та багато іншого.

Одним з найважливіших кроків є заповнення `.env` файлу у самому проєкті. Це можна зробити як з консолі розробника, так і у самій IDE. У першому випадку потрібно зайти до папки проєкту. Створити файл `.env` за допомогою команди `touch .env`, скопіювати усе з файлу `.env.example` та перенести до файлу `.env`, замінивши потрібні данні на свої. У другому випадку потрібно зробити теж саме, але через IDE це буде простіше (рис. 3.1).

```
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:FxKh1G+7xSdIUVJTbyH7Xr+loz63bZ3r37mLSmeeAM=
4 APP_DEBUG=true
5 APP_URL=http://diplom.loc
6
7 LOG_CHANNEL=stack
8 LOG_DEPRECATIONS_CHANNEL=null
9 LOG_LEVEL=debug
10
11 DB_CONNECTION=mysql
12 DB_HOST=127.0.0.1
13 DB_PORT=3306
14 DB_DATABASE=diplom
15 DB_USERNAME=root
16 DB_PASSWORD=1q2w3e4r5t
17
18 BROADCAST_DRIVER=log
19 CACHE_DRIVER=file
20 FILESYSTEM_DRIVER=local
21 QUEUE_CONNECTION=sync
22 SESSION_DRIVER=file
23 SESSION_LIFETIME=120
24
25 MEMCACHED_HOST=127.0.0.1
26
27 REDIS_HOST=127.0.0.1
28 REDIS_PASSWORD=null
29 REDIS_PORT=6379
30
31 MAIL_MAILER=smt
32 MAIL_HOST=mailhog
33 MAIL_PORT=1025
34 MAIL_USERNAME=null
35 MAIL_PASSWORD=null
36 MAIL_ENCRYPTION=null
37 MAIL_FROM_ADDRESS=null
```

Рисунок 3.1 – Файл `.env`

3.2 Структура проєкту

Перехід між сторінками здійснюється за допомогою Routing у Laravel по структурі (рис. 3.2), що дозволяє без перезавантаження міняти контент у застосунку. Структура проєкту розроблена за програмними паттернами SOLID (Single responsibility, Open–closed, Liskov substitution, Interface segregation и Dependency inversion) і наведена на рис. 3.3.

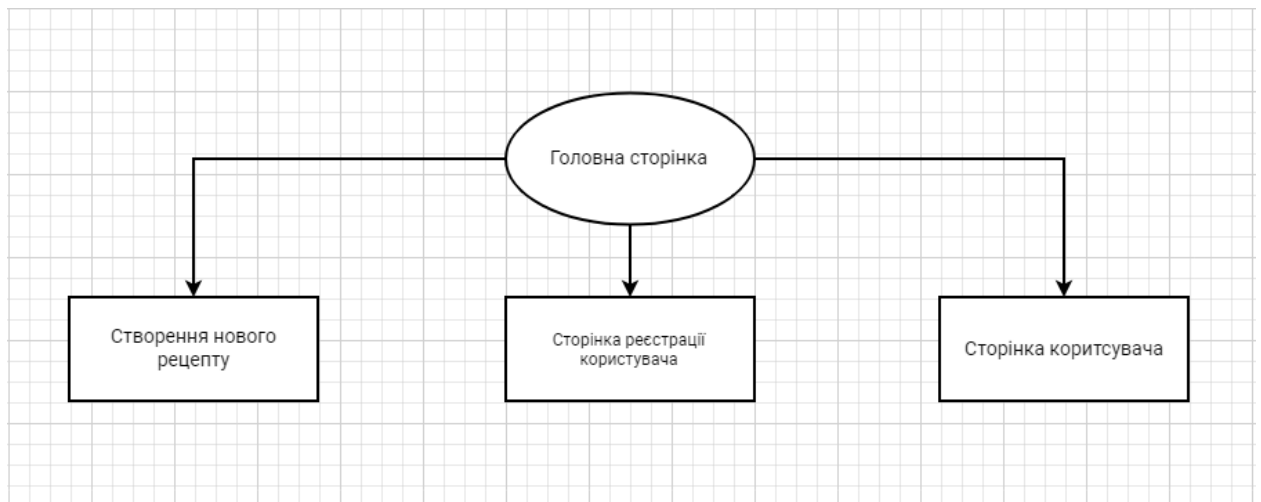


Рисунок 3.2 – Структура роутингу вебзастосунку

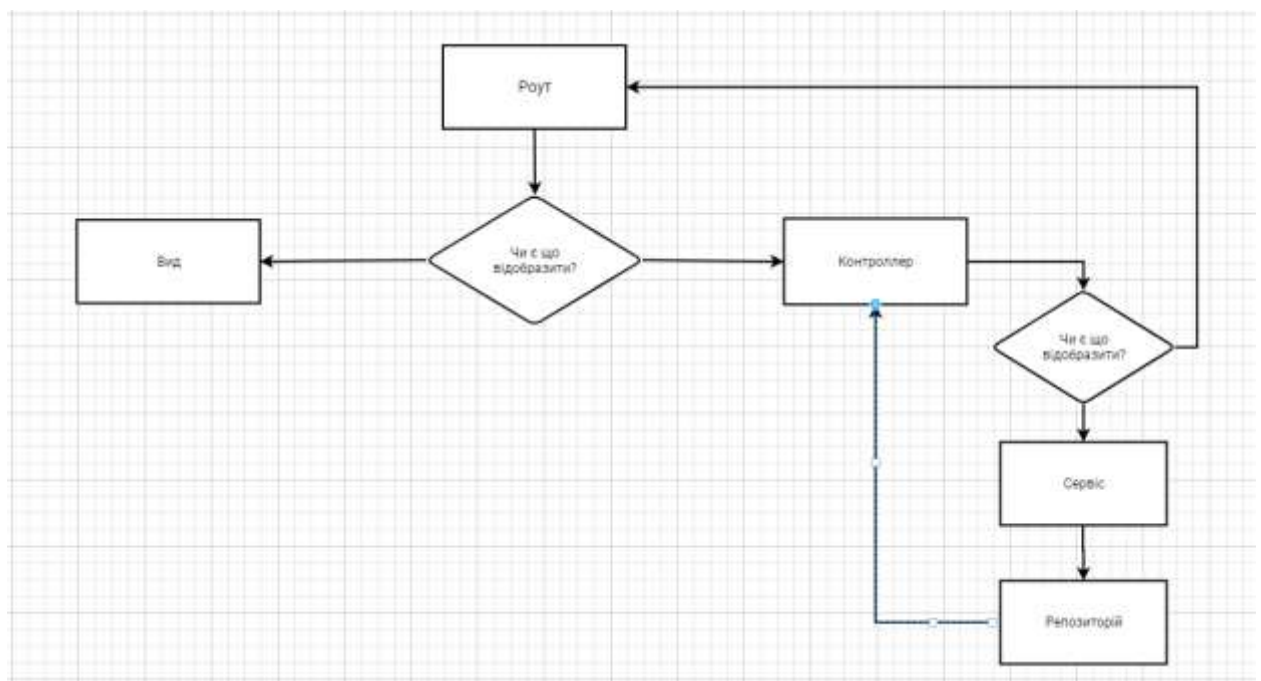


Рисунок 3.3 – Структура програми

3.3 Структура бази даних

База даних проєкту складається з п'яти таблиць. Загальна схема бази даних наведена на рисунку 3.4.

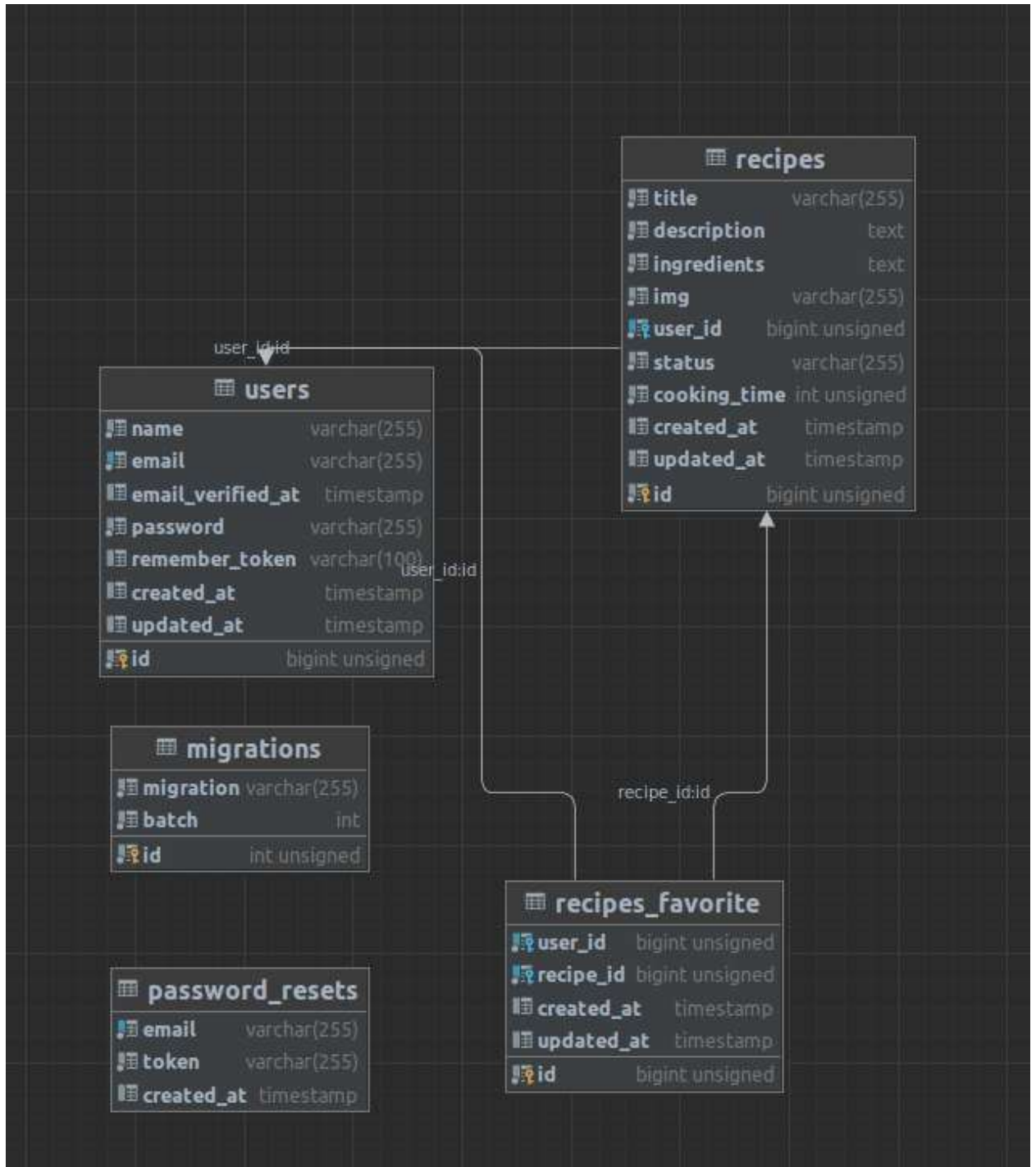


Рисунок 3.4 – Структура бази даних

Таблиця `users` містить інформацію про зареєстрованих користувачів. Кожна сутність «користувач» складається з наступних полів:

- `name` – ім'я;
- `email` – електронна пошта;
- `email_verified_at` – час, коли верифікували пошту;
- `password` (пароль);
- `remember_token` – токен користувача;
- `created_at` – дата та час створення аккаунту користувача;
- `updated_at` – дата та час будь-яких змін на аккаунті користувача.

Таблиця `recipes` містить інформацію про створені рецепти. Кожен рецепт містить у собі такі данні:

- `title` – назва рецепту;
- `description` – опис приготування;
- `ingredients` – інгредієнти;
- `img` – зображення;
- `user_id` – id автора;
- `status` – показник видалення рецепту;
- `cooking_time` – час приготування;
- `created_at` – дата та час створення рецепту;
- `updated_at` – дата та час будь-яких змін рецепту.

Таблиця `recipes_favorite` є суміжною таблицею у відношенні багато до багатьох. Кожен запис демонструє якому користувачеві який рецепт сподобався (`recipe_id`). А саме містить такі данні:

- `user_id` – користувач, якому подобається рецепт;
- `recipe_id` – рецепт, який сподобався;
- `created_at` – дата та час створення запису;
- `updated_at` – дата та час будь-яких змін запису.

3.4 Висновки за розділом 3

В цьому розділі було розглянуто структуру проєкту, роутинг, тобто навігацію між сторінками, розроблено схему бази даних та розглянуто налаштування серверу для роботи з програмою.

4 ЕКСПЛУАТАЦІЯ ТА ТЕСТУВАННЯ РОЗРОБЛЕНОЇ ПРОГРАМИ

4.1 Основні розроблені функції

Основні функції розробленої системи:

- виведення списку останніх доданих рецептів;
- виведення списку найпопулярніших рецептів;
- пошук рецептів;
- виведення останніх доданих статей;
- додавання або видалення в (з) списку вподобаних статей та рецептів;
- зручне фільтрування;
- додавання власних рецептів.

На головній сторінці проєкту знаходиться дві секції з популярними та останніми доданими рецептами. Незареєстрованому користувачеві буде доступним лише перегляд рецептів та пошук. Щоб створювати власні рецепти або зберігати колекції рецептів, користувачу необхідно зареєструватись.

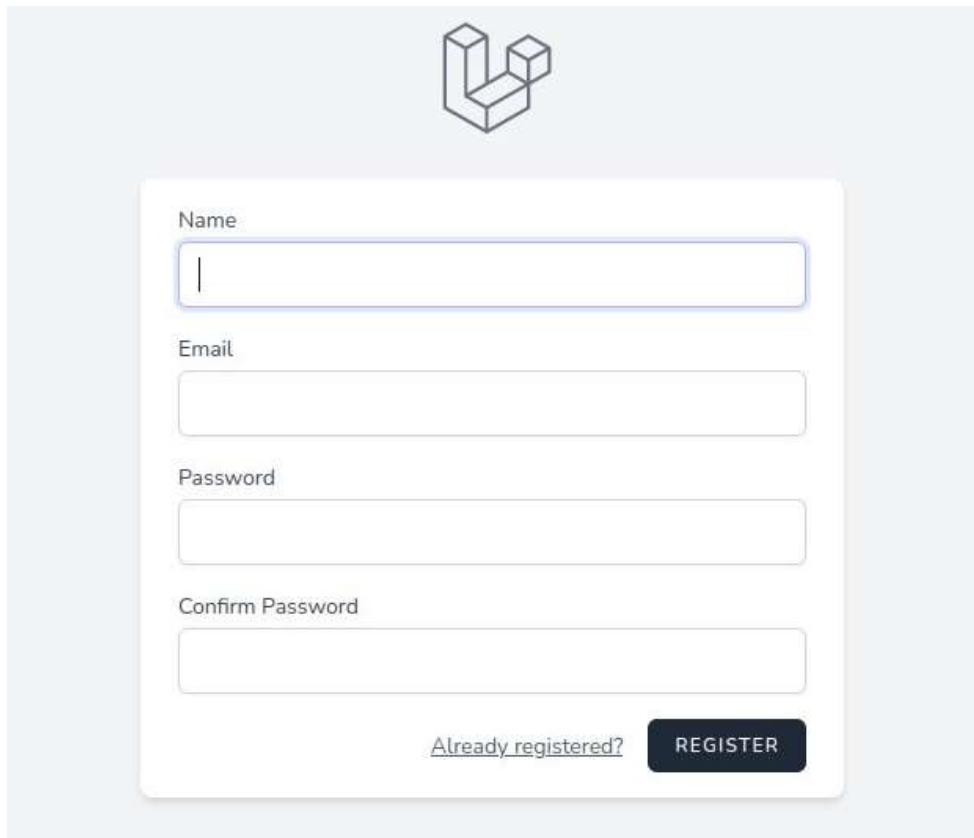
4.2 Реєстрація та авторизація користувачів

Щоб зареєструватися або зайти на існуючий акаунт необхідно натиснути на кнопку «Акаунт» та виконати відповідні дії.

Для реєстрації (рис. 4.1) користувачу необхідно надати ім'я, під яким його рецепти будуть бачити інші користувачі, електронну пошту і пароль.

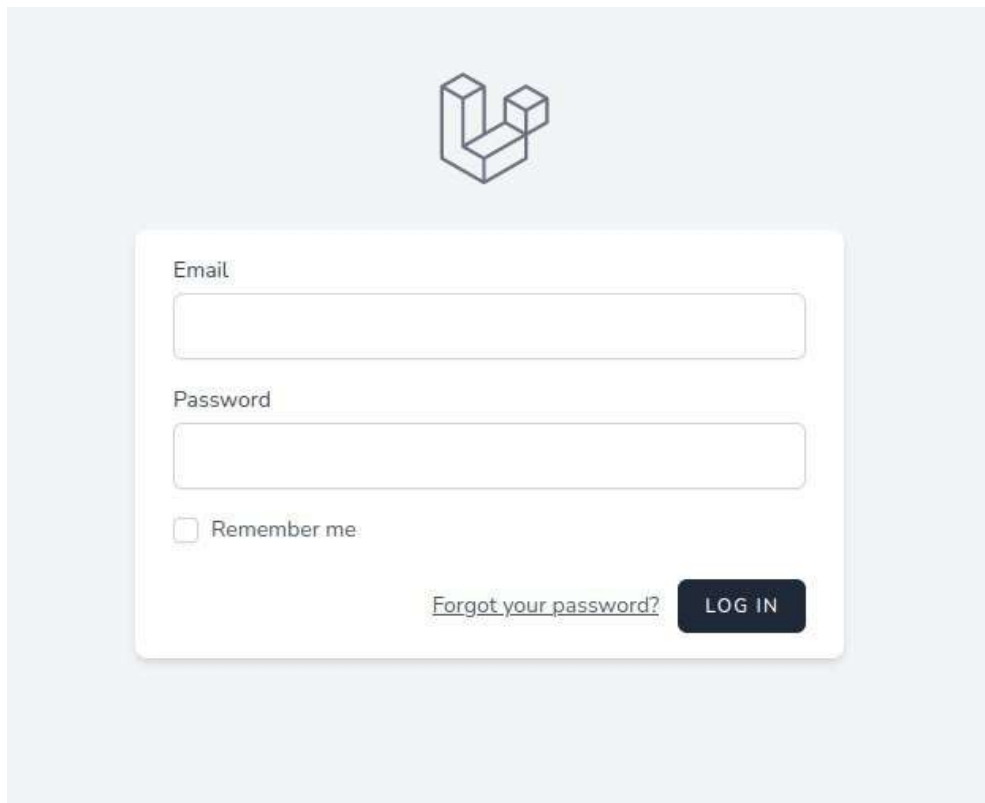
Якщо користувач вже реєструвався, він може перейти до авторизації (рис. 4.2) і ввести дані, які він вказував раніше.

Після успішної авторизації користувачу надано увесь необхідний спектр дій, наприклад, створення рецепту, збереження улюбленого тощо.



The registration form is centered on a light blue background. At the top is a logo consisting of three stacked cubes. Below the logo is a white rounded rectangle containing the following elements: a 'Name' label above a text input field with a vertical cursor; an 'Email' label above a text input field; a 'Password' label above a text input field; and a 'Confirm Password' label above a text input field. At the bottom right of the form are a link labeled 'Already registered?' and a dark blue button with the text 'REGISTER' in white.

Рисунок 4.1 – Форма реєстрації



The login form is centered on a light blue background. At the top is a logo consisting of three stacked cubes. Below the logo is a white rounded rectangle containing the following elements: an 'Email' label above a text input field; a 'Password' label above a text input field; a checkbox labeled 'Remember me'; a link labeled 'Forgot your password?'; and a dark blue button with the text 'LOG IN' in white.

Рисунок 4.2 – Форма авторизації

4.3 Головна сторінка

4.3.1 Секція останніх доданих рецептів

На головній сторінці користувач одразу бачить секцію останніх доданих рецептів. Рецепти сортуються за датою створення, від найновіших до більш давніх (рис. 4.3).

Нові рецепти:		
Назва	Час приготування	Інгредієнти
Вівся з сиrom та зеленню	35	Тісто листкове бездрожжове - 225 г Яйця - 3 шт. —
"Королівський" салат із авокадо	25	Апельсин - 1 шт. Крабові палички - 50 г —
Пованський пиріг з куркою та грибами на листковому тісті	120	Тісто листкове (безрізкове) - 200 г Борошно - 50 г —
Салат з кукурудзою, яйцями та зеленню	20	Кукурудза консервована - 100 г Листя салату - 100 г —

Рисунок 4.3 – Секція останніх доданих рецептів

4.3.2 Секція найпопулярніших рецептів

Головною відмінністю від попередньої секції є тип сортування для відображення на сторінці. У цій секції сортування виконується за показником кількості збереження рецепту користувачами (рис. 4.4).

Популярні рецепти:		
Назва	Час приготування	Інгредієнти
Салат з кукурудзою, яйцями та зеленню	20	Кукурудза консервована - 100 г Листя салату - 100 г —
Пованський пиріг з куркою та грибами на листковому тісті	120	Тісто листкове (безрізкове) - 200 г Борошно - 50 г —

Рисунок 4.4 – Секція найпопулярніших рецептів

4.4 Сторінка перегляду рецепту

Сторінки перегляду рецепту (рис. 4.5) окрім перегляду безпосередньо змісту рецепту та фото результату містять такі кнопки:

- додавання або видалення рецепту з обраного. Цю кнопку бачить тільки зареєстрований користувач;
- видалення рецепту. Цю кнопку бачить тільки автор, якщо переглядає власний рецепт.

Лоранський пиріг з куркою та грибами на листковому тісті

Опис рецепту:

Гриби замочити у холодній воді на 1 годину. Ріпчасту цибулю нарізати тонкими півкільцями. Злегка обсмажити цибулю на олії. Гриби віджати від води (якщо шматочки великі, нарізати), викласти в сковороду до цибулі, обсмажувати 10 хвилин. Курячі грудки відварити або запекти, остудити, дрібно нарізати. Додати до цибулі з грибами, посолити, поперчити, перемішати. На припаленому мукою столі розклати тісто, перенести у форму для запікання (моя форма 26 см), сформувати бортики. Поверх тіста рівномірно розподілити начинку, посипати тертим сиром. Яйця збити виделкою, додати вершки та мускатний горіх, добре перемішати, отриману сумішшю залити начинку. Випікати пиріг з куркою та грибами в розігрітій до 180 ° С духовці 30 хвилин. Готовий пиріг залишити у формі 15 хвилин, потім перенести на блюдо, розрізати на шматочки і можна подавати до столу. Смачного!



Час приготування - 120 хвилин

Інгредієнти:

Тісто листкове (безрізне) - 200 г

Борошно - 50 г

Для начинки:

Куряче філе - 300 г

Гриби сушені - 50 г (або 300 г свіжих грибів)

Цибуля ріпчаста (велика) - 1 шт.

Олія - 1 ст. ложка

Сир сулугуні - 100 г

Перець чорний мелений

Сіль

Для заливки:

Вершки 10% - 250 мл

Яйця - 2 шт.

Мускатний горіх

Рисунок 4.5 – Сторінка перегляду рецепту

4.5 Сторінка користувача

Кожний зареєстрований користувач має власну сторінку, на якій виводяться його рецепти (рис. 4.6) та колекція збережених рецептів (рис. 4.7), а також кнопка додавання нового рецепту.

Ваші рецепти:		
Назва	Час приготування	Інгредієнти
Яснiя з сиром та зеленню	35	Тiсто листкове бездрожжеве - 225 г Яйця - 3 шт. —
"Королівський" салат із авокадо	25	Апельсин - 1 шт. Крабові палички - 50 г —
Львівський пиріг з куркою та грибами на листковому тісті	120	Тiсто листкове (безрізкове) - 200 г Борошно - 50 г —
Салат з кукурудзою, яйцями та зеленню	20	Кукурудза консервована - 100 г Листя салату - 100 г —
Додати рецепт		

Рисунок 4.6 – Рецепти користувача

Ваші збережені рецепти:		
Назва	Час приготування	Інгредієнти
Салат з кукурудзою, яйцями та зеленню	20	Кукурудза консервована - 100 г Листя салату - 100 г

Рисунок 4.7 – Збережені рецепти

4.6 Створення рецепту

Створення рецепту доступне тільки для зареєстрованих користувачів. У форму користувач додає такі данні: назва рецепту, його фото, опис, інгредієнти та час приготування (рис. 4.8).

Додати рецепт



The form is titled "Додати рецепт" (Add recipe). It contains several input fields: "Назва рецепту:" (Recipe name), "Фото рецепту:" (Recipe photo) with a "Browse..." button and "No file selected." text, "Опис рецепту" (Recipe description), "Інгредієнти:" (Ingredients), "Новий інгредієнт - з нового рядка" (New ingredient - from a new line), and "Час приготування:" (Cooking time). A blue "Submit" button is at the bottom left.

Рисунок 4.8 – Форма створення рецепту

Після додавання рецепту система переправляє користувача на головну сторінку, де відображається відповідне сповіщення (рис. 4.9).

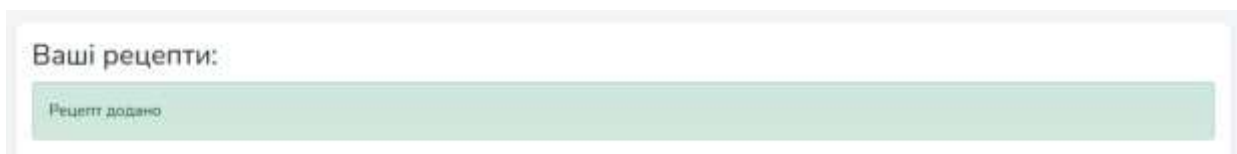


Рисунок 4.9 – Сповіщення про додавання рецепту

4.7 Пошук рецептів

При введенні ключового слова у пошукову строку або зміні параметру виконується запит до API, та програма відображає отримані результати за наявності (рис. 4.10).



Рисунок 4.10 – Результати пошуку

4.8 Висновки за розділом 4

В розділі було протестоване використання системи зареєстрованим користувачем, а саме розглянуто функції реєстрації, авторизації, виведення списку останніх доданих рецептів, виведення списку найпопулярніших рецептів, пошук рецептів, перегляд рецептів.

ВИСНОВКИ

За результатами виконання дипломної кваліфікаційної роботи виконано такі задачі:

- проведено аналіз огляд та аналіз існуючих кулінарних сайтів;
- визначено функціональні вимоги для реалізації програмного забезпечення, обрано мову програмування PHP, фреймворк Laravel для реалізації серверної частини, CSS фреймворк Bootstrap, створено структуру бази даних проєкту;
- реалізовано весь запланований функціонал програмного забезпечення, створено структуру вебсервісу;
- проведено тестування розробленої системи.

Розроблений вебсервіс призначений для ведення онлайн кулінарної книги, створення нових рецептів, а також перегляду та збереження вже існуючих рецептів.

Розроблений продукт повністю виконує поставлені в роботі задачі та може бути використаний будь-яким користувачем з мінімальним набором навичок та досвіду.

Завдання дипломної кваліфікаційної роботи в результаті повністю виконано.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Сайт “Russian food” [Електрон. ресурс]. – Режим доступу: <https://www.russianfood.com/>.
2. Сайт “Гастрономъ” [Електрон. ресурс]. – Режим доступу: <https://www.gastronom.ru/>.
3. Сайт “Picante Cooking” [Електрон. ресурс]. – Режим доступу: <https://picantecooking.com/ua/>.
4. Сайт “Smachno.in.ua” [Електрон. ресурс]. – Режим доступу: <https://www.smachno.in.ua/>.
5. Freeman, A. The Definitive Guide to HTML5 / Adam Freeman. – Apress, 2011. – 1115 p.
6. McFarland, D. CSS: The Missing Manual 4th ed. London / David Sawyer McFarland. – Editorial, 2016. – 720 p.
7. Фрейн, Б. HTML5 і CSS3. Розробка сайтів для будь-яких браузерів та пристроїв: навч. посіб. / Бен Фрейн. – Санкт-Петербург : Пітер, 2017. – 272 с.
8. Perna, M. A. Bootstrap: Related Tools & Skills / Maria Antonietta Perna, David Attard, Callum Hopkins. – Sitepoint, 2018. – 78 p.
9. Reiersol, D. PHP in Action: Objects, Design, Agility / D. Reiersol, M. Baker, C. Shiflett. – Manning Publications, 2007. – 552 p.
10. Зандстра, М. PHP 8: об'єкти, шаблони та методики програмування. 6-те вид. / М. Зандстра. – Київ : Наука, 2021. – 864 с.
11. Стаффер М. Laravel. Повне керівництво. 2-ге видання / Мэтт Стаффер. – Канада : Пітер Прес, 2003. – 132 с.
12. Foster, R. CodeIgniter Web Application Blueprints / Rob Foster. – Packt Publishing, 2015. — 330 p.
13. Porebski, B. Building PHP Applications with Symfony, CakePHP, and Zend Framework / B. Porebski, K. Przystalski. – Wiley Publishing, Inc., 2011. — 576 p.

14. Makarov, A. Yii Application Development Cookbook / A. Makarov. – Packt Publishing, 2013. – 408 p.
15. Miller, S. Getting Started with Phalcon / Stephan A. Miller. – Birmingham : Packt Publishing, 2014. – 138 p.
16. Tweedie, R. Learning FuelPHP for Effective PHP Development / R. Tweedie. – Packt Publishing, 2013. – 104 p.
17. Файзрахманов, А. Архитектура сложных веб-приложений. С примерами на Laravel / Адель Файзрахманов. – Leanpub, 2020. – 270 с.

ДОДАТОК А
Технічне завдання

Вступ

Вебсайти з кулінарними рецептами є достатньо відокремленими рішеннями. Найчастіше такі рішення мають функціональність тільки перегляду рецептів, збереження їх і не завжди мають навіть можливість авторизації, відповідно це не сприяє постійному залученню користувача. З іншого боку таке відокремлення призводить до того, що існуючі функції не піддаються індивідуалізації. Необхідно знайти можливості індивідуалізації статистичних функцій для тіснішого залучення користувачів до роботи програми.

А.1 Підстави для розробки

Підставою для розробки було завдання на дипломну кваліфікаційну роботу за темою «Розробка онлайн «Кулінарної книги»» на основі наказу №102 від 27 квітня 2022 р. за Національним університетом «Запорізька політехніка».

А.2 Призначення розробки

Програма призначена для надання користувачам можливості створювати, зберігати та ділитися улюбленими рецептами. Головною метою було надати можливість доступу до рецептів у будь-який час та надійність їх зберігання.

А.3 Основні вимоги до програми

А.3.1 Вимоги до функціональних характеристик

Вимоги до функціональних характеристик складаються з наступних:

- додавання рецепту;

- виведення списку останніх доданих рецептів;
- виведення списку найпопулярніших рецептів;
- пошук рецептів;
- виведення останніх доданих статей;
- додавання або видалення в (з) списку вподобаних статей та рецептів;
- зручне фільтрування.

A.3.2 Вимоги до надійності

Внесення даних має відбуватися тільки через адміністративну панель, доступ до якої має відбуватися за окремою адресою. Доступ звичайних користувачів, авторизованих і неавторизованих, має реалізовуватися через вебсервер стандартно. Користувачі повинні мати можливість тільки додавати власні записи, які ніяк не впливають на дані щодо самих рецептів.

A.3.3 Умови експлуатації

Для експлуатації програми потрібна підтримка зі сторони менеджера. Ним може бути відповідальна особа з правами адміністратора щодо роботи з системою. Менеджерів може бути декілька для підтримки зручності роботи. Внесення менеджерів у базу даних має відбуватися системним адміністратором.

A.3.4 Вимоги до складу та параметрів технічних засобів

Апаратні вимоги до сервера:

- жорсткий диск з вільним простором – 1 Гб;
- оперативна пам'ять – 4 Гб;
- процесор: 4 ядра, 2,8 ГГц частота.

Клієнт представлений браузером користувача. Для виконання вимог до нього потрібний встановлений сучасний браузер з доступом до мережі інтернет.

A.4 Порядок контролю та приймання

Контроль за виконанням дипломної кваліфікаційної роботи покладається на керівника. Остаточне рішення приймається екзаменаційною комісією під час захисту роботи.

ДОДАТОК Б
Текст програми

Б.1 Package.json – конфігурація проекту

```
{
  "name": "laravel/laravel",
  "type": "project",
  "description": "The Laravel Framework.",
  "keywords": ["framework", "laravel"],
  "license": "MIT",
  "require": {
    "php": "^7.3|^8.0",
    "fruitcake/laravel-cors": "^2.0",
    "guzzlehttp/guzzle": "^7.0.1",
    "laravel/framework": "^8.65",
    "laravel/sanctum": "^2.11",
    "laravel/tinker": "^2.5"
  },
  "require-dev": {
    "facade/ignition": "^2.5",
    "fakerphp/faker": "^1.9.1",
    "laravel/breeze": "^1.4",
    "laravel/sail": "^1.0.1",
    "mockery/mockery": "^1.4.4",
    "nunomaduro/collision": "^5.10",
    "phpunit/phpunit": "^9.5.10"
  },
  "autoload": {
    "psr-4": {
      "App\\": "app/",
      "Database\\Factories\\": "database/factories/",
      "Database\\Seeders\\": "database/seeders/"
    }
  },
  "autoload-dev": {
    "psr-4": {
```

```

        "Tests\\": "tests/"
    }
},
"scripts": {
    "post-autoload-dump": [

"Illuminate\\Foundation\\ComposerScripts::postAutoloadDump",
        "@php artisan package:discover --ansi"
    ],
    "post-update-cmd": [
        "@php artisan vendor:publish --tag=laravel-assets --ansi"
    ],
    "post-root-package-install": [
        "@php -r \"file_exists('.env') || copy('.env.example',
'.env');\""
    ],
    "post-create-project-cmd": [
        "@php artisan key:generate --ansi"
    ]
},
"extra": {
    "laravel": {
        "dont-discover": []
    }
},
"config": {
    "optimize-autoloader": true,
    "preferred-install": "dist",
    "sort-packages": true
},
"minimum-stability": "dev",
"prefer-stable": true
}

```

Б.2 Модель Recipe.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Recipe extends Model
{
    use HasFactory;

    protected $table = 'recipes';

    protected $guarded = [];

    public function favorite()
    {
        return $this->hasMany(RecipeFavorite::class, 'recipe_id',
'id');
    }

    public function getPreparedIngredients($limit = 2)
    {
        if (is_array($this->ingredients)) {
            $ingrArr = array_slice($this->ingredients, 0, $limit);
        } else {
            $ingrArr = array_slice(json_decode($this->ingredients), 0,
$limit);
        }

        array_push($ingrArr, '...');
    }
}
```

```

        return $ingrArr;
    }

    public function getIngredientsAsArray()
    {
        if (is_array($this->ingredients)) {
            $ingrArr = $this->ingredients;
        } else {
            $ingrArr = json_decode($this->ingredients);
        }

        return $ingrArr;
    }
}

```

Б.3 Модель RecipeFavorite.php

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class RecipeFavorite extends Model
{
    use HasFactory;

    protected $table = 'recipes_favorite';

    public function recipe()
    {
        return $this->belongsTo(Recipe::class);
    }
}

```

```
}
```

Б.4 Модель User.php

```
<?php

namespace App\Models;

use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;
use Laravel\Sanctum\HasApiTokens;

class User extends Authenticatable
{
    use HasApiTokens, HasFactory, Notifiable;

    /**
     * The attributes that are mass assignable.
     *
     * @var string[]
     */
    protected $fillable = [
        'name',
        'email',
        'password',
    ];

    /**
     * The attributes that should be hidden for serialization.
     *
     * @var array
     */
```

```

protected $hidden = [
    'password',
    'remember_token',
];

/**
 * The attributes that should be cast.
 *
 * @var array
 */
protected $casts = [
    'email_verified_at' => 'datetime',
];

public function recipes()
{
    return $this->hasMany(Recipe::class, 'user_id', 'id');
}

public function favoriteRecipes()
{
    return $this->hasMany(RecipeFavorite::class, 'user_id', 'id');
}
}

```

Б.5 Роути web.php

```

<?php

use App\Http\Controllers\DefaultController;
use App\Http\Controllers\RecipeController;
use App\Http\Controllers\UserController;
use Illuminate\Support\Facades\Route;

```

```

/*
|-----
-----
| 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('/', [UserController::class, 'home'])->name('home');
```

```
//Authenticated
```

```
Route::middleware('auth')->group(function () {
```

```
    //User
```

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

```
    //Recipe
```

```
    Route::get('/recipes/add', [RecipeController::class, 'create'])->
name('recipe.create');
```

```
    Route::post('/recipes/store', [RecipeController::class, 'store'])->
name('recipe.store');
```

```
    Route::get('/recipes/favorite/add/{id}', [RecipeController::class,
'addToFavorite'])->name('recipe.addToFavorite');
```

```
    Route::get('/recipes/delete/{id}', [RecipeController::class,
'delete'])->name('recipe.delete');
```

```
});
```

```
//Not Authenticated
```

```
//Recipe
Route::get('/recipes/{id}', [RecipeController::class, 'show'])->name('recipe.show');

//Other
Route::get('/search', [DefaultController::class, 'search']);

//Get Image
Route::get('/recipes/img/{filename}', [DefaultController::class, 'getImage']);

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

Б.6 Контроллер RecipeController.php

```
<?php

namespace App\Http\Controllers;

use App\Http\Requests\Recipe\RecipeStoreRequest;
use App\Services\RecipeService;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Illuminate\View\View;

class RecipeController extends Controller
{
    private $recipeService;

    public function __construct(RecipeService $recipeService)
    {
        $this->recipeService = $recipeService;
    }
}
```

```

public function create(): View
{
    return view('recipes.create');
}

public function store(RecipeStoreRequest $request)
{
    $validated = $request->validated();

    $this->recipeService->store($validated);

    return redirect()->route('account')->with('success', 'Рецепт
додано');
}

public function show(int $id)
{
    $recipe = $this->recipeService->getById($id);

    return view('recipes.show')->with([
        'recipe' => $recipe
    ]);
}

public function addToFavorite(int $id)
{
    $message = $this->recipeService-
>addOrRemoveFromFavorite(Auth::id(), $id);

    return redirect()->route('recipe.show', ['id' => $id])
        ->with('success', $message);
}

public function delete(int $id)

```

```

    {
        $this->recipeService->delete($id);

        return redirect()->route('account')->with('success', 'Рецепт
видалено');
    }
}

```

Б.7 Контроллер UserController.php

```

<?php

namespace App\Http\Controllers;

use App\Repositories\RecipeRepository;
use Illuminate\Support\Facades\Auth;
use Illuminate\View\View;

class UserController extends Controller
{
    private $recipeRepository;

    public function __construct(RecipeRepository $recipeRepository)
    {
        $this->recipeRepository = $recipeRepository;
    }

    public function home(): View
    {
        return view('home')->with([
            'allRecipes' => $this->recipeRepository->getAll(5),
            'favoriteRecipes' => $this->recipeRepository->getFavoriteRecipes(5)
        ]);
    }
}

```

```

    }

    public function account(): View
    {
        return view('account')->with([
            'recipes' => $this->recipeRepository->getRecipesForCurrentUser(),
            'favoriteRecipes' => $this->recipeRepository->getFavoriteForCurrentUser(Auth::id())
        ]);
    }
}

```

Б.8 Контроллер DefaultController.php

```

<?php

namespace App\Http\Controllers;

use App\Http\Requests\SearchRequest;
use App\Services\RecipeService;
use Illuminate\Support\Facades\File;
use Illuminate\Support\Facades\Response;
use Illuminate\View\View;

class DefaultController extends Controller
{
    private $recipeService;

    public function __construct(RecipeService $recipeService)
    {
        $this->recipeService = $recipeService;
    }
}

```

```

public function search(SearchRequest $request)
{
    $recipes = $this->recipeService->getByTitle($request-
>get('search'), true);

    return view('search')->with([
        'search_phrase' => $request->get('search'),
        'recipes' => $recipes
    ]);
}

public function getImage(string $filename)
{
    $path = storage_path() . '/app/images/' . $filename;

    if(!File::exists($path)) {
        return response()->json(['message' => 'Image not found.'],
404);
    }

    $file = File::get($path);
    $type = File::mimeType($path);

    $response = Response::make($file, 200);
    $response->header("Content-Type", $type);

    return $response;
}
}

```

Б.9 Репозиторій RecipeRepository.php

```
<?php
```

```

namespace App\Repositories;

use App\Models\Recipe;
use App\Models\RecipeFavorite;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\DB;

class RecipeRepository
{
    public function store(Recipe $recipe)
    {
        $recipe->save();
    }

    public function getAll($limit = 30)
    {
        return Recipe::query()
            ->orderBy('id', 'desc')
            ->limit($limit)
            ->get();
    }

    public function getFavoriteRecipes($limit = 10)
    {
        return RecipeFavorite::select('recipe_id',
DB::raw('COUNT(recipe_id)'))
            ->with(['recipe'])
            ->groupBy('recipe_id')
            ->orderBy(DB::raw('COUNT(recipe_id)'), 'desc')
            ->limit($limit)
            ->get();
    }

    public function findOneBy(array $params)

```

```

{
    $query = Recipe::query()->whereRowValues(array_keys($params),
    '=', array_values($params));

    return $query->first();
}

public function getRecipesForCurrentUser()
{
    $recipes = Recipe::query()
        ->orderBy('id', 'desc')
        ->where('user_id', Auth::id())
        ->get();

    foreach ($recipes as $recipe)
    {
        $recipe->ingredients = json_decode($recipe->ingredients);
    }

    return $recipes;
}

public function getByTitle(string $title, bool $like)
{
    $recipe = Recipe::query();

    if ($like) {
        $recipe->where('title', 'like', '%'.$title.'%');
    } else {
        $recipe->where('title', $title);
    }

    return $recipe->get();
}

```

```

public function findFavorite(int $userId, int $recipeId)
{
    return RecipeFavorite::query()
        ->where('user_id', $userId)
        ->where('recipe_id', $recipeId)
        ->first();
}

public function addToFavorite(int $userId, int $recipeId)
{
    $favorite = new RecipeFavorite();
    $favorite->user_id = $userId;
    $favorite->recipe_id = $recipeId;

    $favorite->save();
}

public function getFavoriteForCurrentUser(int $userId)
{
    return RecipeFavorite::with(['recipe'])->where('user_id',
$userId)->get();
}
}

```

Б.10 Сербic RecipeService.php

```

<?php

namespace App\Services;

use App\Models\Recipe;
use Illuminate\Support\Facades\Auth;
use App\Repositories\RecipeRepository;

```

```

class RecipeService
{
    public const STATUS_ACTIVE = 'Active';
    public const UNDER_REVIEW = 'Under Review';

    private $repository;

    public function __construct(RecipeRepository $recipeRepository)
    {
        $this->repository = $recipeRepository;
    }

    public function store(array $params)
    {
        $imageName = Auth::user()->getAuthIdentifier() . '_' . time()
        . '.' . $params['recipeImage']->extension();
        $params['recipeImage']->storeAs('images', $imageName);

        $ingredients = explode("\r\n", $params['ingredients']);

        $recipe = new Recipe();
        $recipe->title = $params['title'];
        $recipe->description = $params['description'];
        $recipe->ingredients = json_encode($ingredients);
        $recipe->img = $imageName;
        $recipe->user_id = Auth::id();
        $recipe->status = self::UNDER_REVIEW;
        $recipe->cooking_time = $params['cookingTime'];

        $this->repository->store($recipe);
    }

    public function getById(int $id)

```

```

{
    return $this->repository->findOneBy(['id' => $id]);
}

public function getByTitle(string $title, bool $like)
{
    return $this->repository->getByTitle($title, $like);
}

public function addOrRemoveFromFavorite(int $userId, int
$recipeId)
{
    $isFavorite = $this->repository->findFavorite($userId,
$recipeId);

    if (!$isFavorite) {
        $this->repository->addToFavorite($userId, $recipeId);
        $message = 'Рецепт додано до улюблених';
    } else {
        $isFavorite->delete();
        $message = 'Рецепт видалено з улюблених';
    }

    return $message;
}

public function delete(int $id)
{
    $recipe = $this->getById($id);
    $recipe->delete();
}
}

```

Б.11 Міграція CreateRecipesFavoriteTable

```

<?php

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

class CreateRecipesFavoriteTable extends Migration
{
    /
    * Run the migrations.
    *
    * @return void
    */
    public function up()
    {
        Schema::create('recipes_favorite', function (Blueprint $table)
        {
            $table->id();
            $table->unsignedBigInteger('user_id');
            $table->unsignedBigInteger('recipe_id');
            $table->timestamps();

            $table->foreign('user_id')->references('id')->on('users')->
onDelete('cascade');
            $table->foreign('recipe_id')->references('id')->
on('recipes')->onDelete('cascade');

            $table->unique(['user_id', 'recipe_id']);
        });
    }

    /

```

```

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

```

Б.12 Міграція CreateRecipesTable

```

<?php

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

class CreateRecipesTable extends Migration
{
    /
    * Run the migrations.
    *
    * @return void
    */
    public function up()
    {
        Schema::create('recipes', function (Blueprint $table) {
            $table->id();
            $table->string('title')->nullable(false);
            $table->text('description');
            $table->text('ingredients');
            $table->string('img');
            $table->unsignedBigInteger('user_id');

```

```

        $table->string('status');
        $table->unsignedInteger('cooking_time');
        $table->timestamps();

        $table->foreign('user_id')->references('id')->on('users');
    });
}

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

```

Б.13 Сторінка акаунту користувача

```

@extends('layouts.main')
@section('title', 'Панель управління')
@section('content')
    <div class="bg-white p-3 mb-3 overflow-hidden shadow-sm
sm:rounded-lg">
        <h3>Ваші рецепти:</h3>
        @if ($recipes->isEmpty())
            @if (\Session::has('success'))
                <div class="alert alert-success">
                    {!! \Session::get('success') !!}
                </div>
            @endif
            <table class="table">

```

```

<thead>
<tr>
    <th>Назва</th>
    <th>Час приготування</th>
    <th>Інгредієнти</th>
</tr>
</thead>
<tbody>
@foreach($recipes as $recipe)
    <tr>
        <td><a href="{{route('recipe.show', ['id' =>
$recipe->id])}}">{{ $recipe->title }}</a></td>
        <td>{{ $recipe->cooking_time }}</td>
        <td>
            <ul>
                @foreach($recipe-
>getPreparedIngredients() as $ingr)
                    <li>{{ $ingr }}</li>
                @endforeach
            </ul>
        </td>
    </tr>
@endforeach
</tbody>
</table>
@else
    <p>Ви не додали власних рецептів.</p>
@endif

    <a href="{{ route('recipe.create') }}" class="btn btn-
primary">Додати рецепт</a>
</div>

@if ($favoriteRecipes->isEmpty())

```

```

<div class="bg-white p-3 overflow-hidden shadow-sm sm:rounded-lg">
  <h3>Ваші збережені рецепти:</h3>
  <table class="table">
    <thead>
      <tr>
        <th>Назва</th>
        <th>Час приготування</th>
        <th>Інгредієнти</th>
      </tr>
    </thead>
    <tbody>
      @foreach($favoriteRecipes as $recipe)
        <tr>
          <td><a href="{{route('recipe.show', ['id' =>
$recipe->recipe_id])}}">{{ $recipe->recipe->title }}</a></td>
          <td>{{ $recipe->recipe->cooking_time }}</td>
          <td>
            <ul>
              @foreach($recipe->recipe-
>getPreparedIngredients() as $ingr)
                <li>{{ $ingr }}</li>
              @endforeach
            </ul>
          </td>
        </tr>
      @endforeach
    </tbody>
  </table>
</div>
@endif
@endsection

```

Б.14 Головна сторінка

```

@extends('layouts.main')
@section('title', 'Рецепти')
@section('content')
    <div class="bg-white p-3 overflow-hidden shadow-sm sm:rounded-lg">
        <h3>Нові рецепти:</h3>
        <table class="table">
            <thead>
                <tr>
                    <th>Назва</th>
                    <th>Час приготування</th>
                    <th>Інгредієнти</th>
                </tr>
            </thead>
            <tbody>
                @foreach($allRecipes as $recipe)
                    <tr>
                        <td><a href="{{route('recipe.show', ['id' =>
$recipe->id])}}">{{ $recipe->title }}</a></td>
                        <td>{{ $recipe->cooking_time }}</td>
                        <td>
                            <ul>
                                @foreach($recipe->getPreparedIngredients()
as $ingr)
                                    <li>{{ $ingr }}</li>
                                @endforeach
                            </ul>
                        </td>
                    </tr>
                @endforeach
            </tbody>
        </table>
    </div>

```

```

<br>
<div class="bg-white p-3 overflow-hidden shadow-sm sm:rounded-lg">
  <h3>Популярні рецепти:</h3>
  <table class="table">
    <thead>
      <tr>
        <th>Назва</th>
        <th>Час приготування</th>
        <th>Інгредієнти</th>
      </tr>
    </thead>
    <tbody>
      @foreach($favoriteRecipes as $recipe)
        <tr>
          <td><a href="{{route('recipe.show', ['id' =>
$recipe->recipe_id])}}">{{ $recipe->recipe->title }}</a></td>
          <td>{{ $recipe->recipe->cooking_time }}</td>
          <td>
            <ul>
              @foreach($recipe->recipe-
>getPreparedIngredients() as $ingr)
                <li>{{ $ingr }}</li>
              @endforeach
            </ul>
          </td>
        </tr>
      @endforeach
    </tbody>
  </table>
</div>
@endsection

```

Б.15 Пошук

```

@extends('layouts.main')
@section('title', 'Пошук по запиту "' . $search_phrase . "'")
@section('content')
    <div class="bg-white p-3 overflow-hidden shadow-sm sm:rounded-lg">
        <table class="table">
            <tbody>
                @foreach($recipes as $recipe)
                    <tr>
                        <td><a href="{{route('recipe.show', ['id' =>
$recipe->id])}}">{{ $recipe->title }}</a></td>
                    </tr>
                @endforeach
            </tbody>
        </table>
    </div>

@endsection

```

Б.16 Створення рецепту

```

@extends('layouts.main')
@section('title', 'Додати рецепт')
@section('content')
    <div class="bg-white p-3 overflow-hidden shadow-sm sm:rounded-lg">
        <form action="{{ route('recipe.store') }}" method="POST"
enctype="multipart/form-data">
            @csrf
            <div class="mb-3">
                <label for="recipeTitle" class="form-label">Назва
рецепту:</label>
                <input
                    name="title"

```

```

        type="text"
        class="form-control @if($errors->has('title')) is-
invalid @endif"
        id="recipeTitle"
        value="{{ old('title') }}"
    >

    @if($errors->has('title'))
        <div class="alert alert-danger mt-1" role="alert">
            {{ $errors->first('title') }}
        </div>
    @endif

</div>
<div class="mb-3">
    <label for="recipeImage" class="form-label">Фото
рецепты:</label>
    <input
        name="recipeImage"
        type="file"
        class="form-control @if($errors-
>has('recipeImage')) is-invalid @endif"
        id="recipeImage"
        value="{{ old('recipeImage') }}"
    >

    @if($errors->has('recipeImage'))
        <div class="alert alert-danger mt-1" role="alert">
            {{ $errors->first('recipeImage') }}
        </div>
    @endif

</div>
<div class="mb-3">

```

```

        <label for="recipeDescription" class="form-label">Опис
рецепту</label>
        <textarea
            name="description"
            class="form-control @if($errors-
>has('description')) is-invalid @endif"
            id="recipeDescription"
            rows="3">{{ old('description') }}</textarea>

        @if($errors->has('description'))
            <div class="alert alert-danger mt-1" role="alert">
                {{ $errors->first('description') }}
            </div>
        @endif
    </div>
    <div class="mb-3">
        <label for="recipeIngredients" class="form-
label">Інгредієнти:</label>
        <textarea
            name="ingredients"
            class="form-control @if($errors-
>has('ingredients')) is-invalid @endif"
            id="recipeIngredients"
            rows="3">{{ old('ingredients') }}</textarea>
        <small id="ingredientsHelp" class="form-text text-
muted">Новий інгредієнт – з нового рядка</small>

        @if($errors->has('ingredients'))
            <div class="alert alert-danger mt-1" role="alert">
                {{ $errors->first('ingredients') }}
            </div>
        @endif
    </div>
    <div class="mb-3">

```

```

        <label for="recipeCookingTime" class="form-label">Час
приготування:</label>
        <input
            name="cookingTime"
            type="text"
            class="form-control @if($errors-
>has('cookingTime')) is-invalid @endif"
            id="recipeCookingTime"
            value="{{ old('cookingTime') }}"
        >

        @if($errors->has('cookingTime'))
            <div class="alert alert-danger mt-1" role="alert">
                {{ $errors->first('cookingTime') }}
            </div>
        @endif
    </div>
    <button type="submit" class="btn btn-
primary">Submit</button>
</form>
</div>
@endsection

```

Б.17 Показ рецепту

```

@extends('layouts.main')
@section('title', 'Додати рецепт')
@section('content')
    <div class="bg-white p-3 overflow-hidden shadow-sm sm:rounded-lg">
        <form action="{{ route('recipe.store') }}" method="POST"
enctype="multipart/form-data">
            @csrf
            <div class="mb-3">

```

```

        <label for="recipeTitle" class="form-label">Назва
рецепту:</label>
        <input
            name="title"
            type="text"
            class="form-control @if($errors->has('title')) is-
invalid @endif"
            id="recipeTitle"
            value="{{ old('title') }}"
        >

        @if($errors->has('title'))
            <div class="alert alert-danger mt-1" role="alert">
                {{ $errors->first('title') }}
            </div>
        @endif

    </div>
    <div class="mb-3">
        <label for="recipeImage" class="form-label">Фото
рецепту:</label>
        <input
            name="recipeImage"
            type="file"
            class="form-control @if($errors-
>has('recipeImage')) is-invalid @endif"
            id="recipeImage"
            value="{{ old('recipeImage') }}"
        >

        @if($errors->has('recipeImage'))
            <div class="alert alert-danger mt-1" role="alert">
                {{ $errors->first('recipeImage') }}
            </div>
        @endif
    </div>

```

```

@endif

</div>
<div class="mb-3">
    <label for="recipeDescription" class="form-label">Опис
рецепту</label>
    <textarea
        name="description"
        class="form-control @if($errors-
>has('description')) is-invalid @endif"
        id="recipeDescription"
        rows="3">{{ old('description') }}</textarea>

    @if($errors->has('description'))
        <div class="alert alert-danger mt-1" role="alert">
            {{ $errors->first('description') }}
        </div>
    @endif
</div>
<div class="mb-3">
    <label for="recipeIngredients" class="form-
label">Інгредієнти:</label>
    <textarea
        name="ingredients"
        class="form-control @if($errors-
>has('ingredients')) is-invalid @endif"
        id="recipeIngredients"
        rows="3">{{ old('ingredients') }}</textarea>
    <small id="ingredientsHelp" class="form-text text-
muted">Новий інгредієнт – з нового рядка</small>

    @if($errors->has('ingredients'))
        <div class="alert alert-danger mt-1" role="alert">
            {{ $errors->first('ingredients') }}

```

```

        </div>
    @endif
</div>
<div class="mb-3">
    <label for="recipeCookingTime" class="form-label">Час
приготування:</label>
    <input
        name="cookingTime"
        type="text"
        class="form-control @if($errors-
>has('cookingTime')) is-invalid @endif"
        id="recipeCookingTime"
        value="{{ old('cookingTime') }}"
    >

    @if($errors->has('cookingTime'))
        <div class="alert alert-danger mt-1" role="alert">
            {{ $errors->first('cookingTime') }}
        </div>
    @endif
</div>
    <button type="submit" class="btn btn-
primary">Submit</button>
</form>
</div>
@endsection

```

ДОДАТОК В
Слайди презентації

Національний університет «Запорізька політехніка»
Кафедра програмних засобів

Випускна кваліфікаційна робота бакалавра на тему:
РОЗРОБКА ОНЛАЙН «КУЛІНАРНОЇ КНИГИ»

Виконала:
ст. гр. КНТ-138

А.А. Лях

Керівник:
к.т.н., доцент

Т.О. Колпакова

1

Рисунок Б.1 – Слайд 1

Мета роботи

Об'єкт дослідження – процес зберігання, відтворення та пошуку рецептів.

Предмет дослідження – програмне забезпечення для пошуку та зберігання рецептів.

Мета роботи – створення вебсайту для перенесення паперової кулінарної книги у мережу Інтернет та її вдосконалення для сучасних користувачів.

2

Рисунок Б.1 – Слайд 2

Задачі роботи

- дослідити предметну область;
- виконати аналіз процесу розробки веб-застосунку;
- розробити структуру пояснювальної записки;
- виконати програмну реалізацію проекту;
- провести тестування програмного продукту.

3

Рисунок Б.3 – Слайд 3

Порівняння існуючих аналогів

Функціональні можливості	Назва програмних продуктів			
	RussianFood.com	Гастроном	Picante Cooking	Smachno.in.ua
Зручний дизайн	-	+	+	-
Легкочитаємий текст	+	-	+	+
Пошук за ключовою фразою	+	+	+	+
Відсутність реклами	+	-	-	+
Зручна фільтрація рецептів	+	+	+	+
Оновлення дизайну та контенту	-	+	+	-
Гарне співвідношення тексту та фото	+	-	+	+
Наявність соціальних мереж	-	-	+	-
Наявність різних типів контенту	-	+	+	+
Додатковий функціонал	-	+	+	+

4

Рисунок Б.4 – Слайд 4

Переваги й недоліки фреймворків

Фреймворк	Переваги	Недоліки
Laravel	Найкраща Inversion of Control Зручна система міграції Тестування модуля Шаблонний двигун Blade Гнучка маршрутизація Багато способів створення REST API Швидкий розвиток Зручна документація Велика популярність Вбудована консоль налагодження з функцією стека викликів	Вам потрібно працювати з фасадами Laravel, якщо ви хочете мати доступ до всіх функцій, IDE не розпізнають методи та властивості в деяких класах і показують попереджувальне повідомлення Крива навчання крутіша, ніж у Yii2 Немає генерації вбудованого інтерфейсу

5

Рисунок Б.5 – Слайд 5

Переваги й недоліки фреймворків

Фреймворк	Переваги	Недоліки
Symfony	Дуже схожий на Yii Вбудована підтримка Codeception дозволяє писати функціональні та приймальні тести Має компонент Ain't Markup Language, що є величезною перевагою для будь-якого фреймворка Велика спільнота розробників Багато готових до використання наборів модулів, які називаються пакетами Детальна і зрозуміла документація	Symfony використовує дуже страшний Object-Relational Mapping (Propel and Doctrine) і є ресурсомістким Крута крива навчання Забгато різноманітних об'єктів Містить синтаксис анотації

6

Рисунок Б.6 – Слайд 6

Переваги й недоліки фреймворків

Фреймворк	Переваги	Недоліки
Phalcon	Висока ефективність Багато функцій Підходить для середовищ високого навантаження Не вимагає багато операцій з файлами Не споживає багато ресурсів Ви можете використовувати свою улюблену бібліотеку бази даних та її елементи Бази даних обробляються через Object-Relational Mapping, що призводить до підвищення продуктивності Усі процеси відбуваються швидко, оскільки фреймворк має прямий доступ до внутрішніх структур PHP	Його вихідний код написаний на C Ще не дуже популярний Це розширення (це означає, що ви не зможете запустити програму на спільному хостингі) Є багато людей, які досі не мають уявлення про Phalcon

7

Рисунок Б.7 – Слайд 7

Переваги й недоліки фреймворків

Фреймворк	Переваги	Недоліки
Zend	Спадкування класів Об'єктно-орієнтований Має готові до використання рішення для багатьох завдань Ви можете інтегрувати все, що забажаєте, з чим завгодно Інструменти інтернаціоналізації Підтримка спільноти розробників Гарна документація	Не дуже підходить для швидкого розвитку Повільніше, ніж деякі інші фреймворки (але все одно достатньо швидко для 90% вебсайтів, база даних завжди є вузьким місцем) Для освоєння потрібно багато часу Ресурсомісткий

8

Рисунок Б.8 – Слайд 8

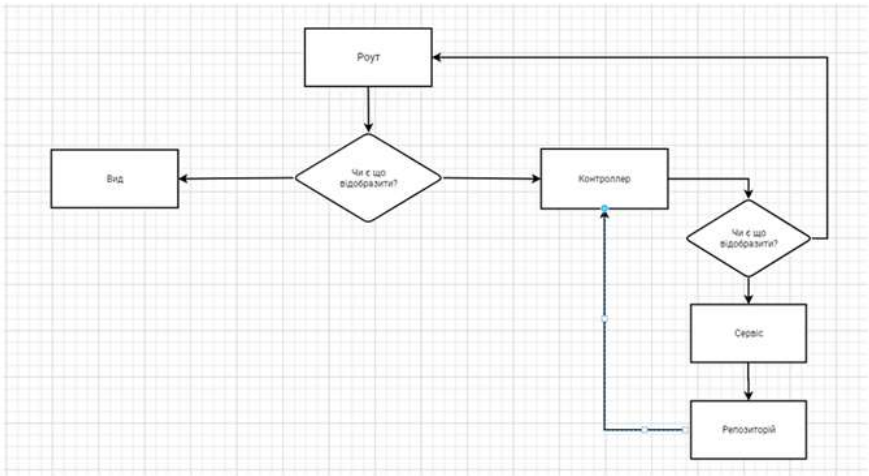
Переваги й недоліки фреймворків

Фреймворк	Переваги	Недоліки
Yii	Інтуїтивно зрозуміла архітектура MVC Спрощує підтримку коду, який використовує загальну архітектуру та методи Велика спільнота розробників Економить час, витрачений на звичайні завдання, такі як перевірка форми та безпеки Легко налаштувати для кращої продуктивності Не ресурсомісткий Різні варіанти кешування Сторонні бібліотеки та класи можна легко інтегрувати Гарні засоби безпеки Поведінка, помічники, можливість розширити базову функціональність тощо Підходить для проєктів будь-якого масштабу та складності	Не дуже гнучка маршрутизація Фронтенд і бекенд бібліотеки сильно заплутані

9

Рисунок Б.9 – Слайд 9

Структура програми



10

Рисунок Б.10 – Слайд 10

Структура бази даних

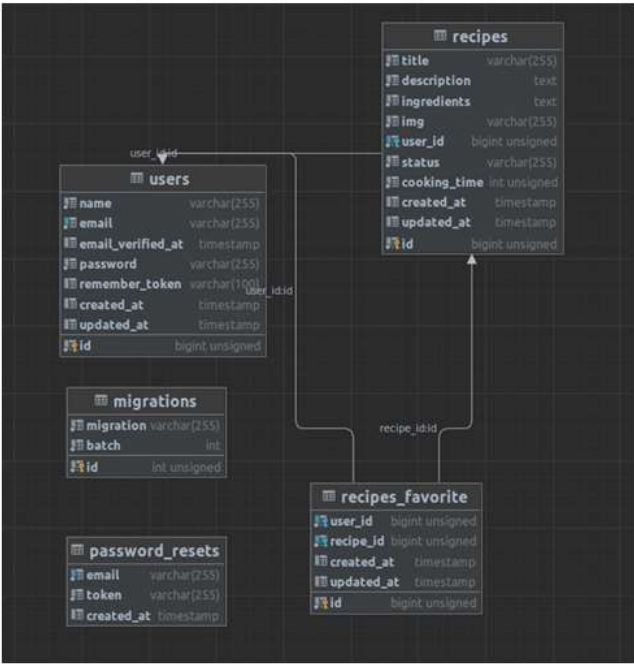


Рисунок Б.11 – Слайд 11

Блок створення рецепту

Додати рецепт

Назва рецепту:

Фото рецепту:

Browse...

No file selected.

Опис рецепту

Інгредієнти:

Новий інгредієнт - з нового рядка

Час приготування:

Submit

Рисунок Б.12 – Слайд 12

