

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування інституту, факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему РОЗРОБКА ВЕБРЕСУРСУ ДЛЯ ОРГАНІЗАЦІЇ
РОБОЧОГО ПРОЦЕСУ СТУДЕНТА

DEVELOPMENT OF A WEB RESOURCE TO ORGANIZE
A STUDENT'S WORKFLOW

Виконав: студент(ка) 4 курсу, групи КНТ-119
Спеціальності 121 Інженерія програмного
забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

ПОКРЕПОВ О.В.

(ПРІЗВИЩЕ та ініціали)

Керівник КАПЛІЄНКО Т.І.

(ПРІЗВИЩЕ та ініціали)

Рецензент ШИТІКОВА О.В.

(ПРІЗВИЩЕ та ініціали)

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

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.
Сергій Субботін
“ ” 2023 року

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

ПОКРЕПОВА Олексія Володимировича

(ПРІЗВИЩЕ, ім'я, по батькові)

1. Тема проекту (роботи) Розробка вебресурсу для організації робочого процесу студента

керівник проекту (роботи) к.т.н., доцент Тетяна Ігорівна КАПЛІЄНКО,
(науковий ступінь, вчене звання, ПРІЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “04” квітня 2023 року № 87

2. Строк подання студентом проекту (роботи) 07 червня 2022 року

3. Вихідні дані до проекту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз проблеми та постановка завдань дослідження

2. Матеріали і методи. 3. Опис програми. 4. Експлуатація, тестування та експериментальне дослідження 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	КАПЛІЄНКО Т.І., доцент		
Нормоконтроль	КАМІНСЬКА Ж.К., асистент		

7. Дата видачі завдання « 06 » лютого 2023 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)

Олексій ПОКРЕПОВ
(підпис) (Ім'я ПРІЗВИЩЕ)

Керівник проєкту (роботи)

Тетяна КАПЛІЄНКО
(підпис) (Ім'я ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 81 с., 47 рис., 6 табл., 2 дод., 18 джерел..

РОЗРОБКА ВЕБРЕСУРСУ ДЛЯ ОРГАНІЗАЦІЇ РОБОЧОГО ПРОЦЕСУ СТУДЕНТА.

Об'єкт дослідження – вебресурси для організації робочого процесу студента.

Предмет дипломної роботи – вебсайт, який допомагає в організації робочого процесу студента.

Мета роботи – розробка вебресурсу, який допоможе студентам більш ефективно організувати свій робочий процес та забезпечить можливість отримувати необхідну інформацію з будь-якого місця з доступом до Інтернету.

Матеріали, методи та технічні засоби: методологія розробки вебзастосунків, HTML, CSS, JavaScript, фреймворки Bootstrap та jQuery, система управління базами даних MySQL, вебсервер Apache, веббраузер Google Chrome.

Результати. Створено вебсайт, який містить функції для організації робочого процесу студента, такі як створення та планування розкладу занять, завдань та дедлайнів, зберігання та обмін документами, комунікація з викладачами та іншими студентами через чат, форум та електронну пошту. Всі дані зберігаються в базі даних, що забезпечує зручний та швидкий доступ до них.

Висновки. Розроблено вебсайт, який дозволяє студентам ефективно організувати свій робочий процес та забезпечує доступ до необхідної інформації з будь-якого місця з доступом до Інтернету.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 81 pages, 47 figures, 6 tables, 2 appendixes, 18 sources.

DEVELOPMENT OF A WEB RESOURCE FOR THE STUDENT WORKFLOW ORGANIZING.

Object of research is a web resources for organizing the student work process.

The subject of the application is the website, which helps in organizing the student's work process.

The aim of the work is to develop a web resource that will help students to more effectively organize their work process and provide access to necessary information from any location with Internet access.

Materials, methods, and technical tools: methodology for developing web applications, HTML, CSS, JavaScript, Bootstrap and jQuery frameworks, MySQL database management system, Apache web server, Google Chrome web browser.

Results. A web application has been created that includes features for organizing the student work process, such as creating and scheduling class schedules, tasks, and deadlines, storing and exchanging documents, communicating with teachers and other students through chat, forums, and email. All data is stored in a database, providing convenient and fast access to it.

Conclusions. A web application has been developed that allows students to effectively organize their work process and provides access to necessary information from any location with Internet access. Field of application - rehabilitation of patients in medical institutions by neurologists or in home conditions by parents.

ЗМІСТ

Перелік скорочень та умовних познач	8
Вступ	9
1 Аналіз предметної області	10
1.1 Опис предметної галузі	10
1.1.1 Оцінка в кінці семестру	12
1.1.2 ECTS	13
1.2 Аналіз завдань інформаційних систем для організації робочого процесу студента	15
1.3 Аналіз існуючих на ринку інформаційних систем для робочого процесу студента	17
1.3.1 Google Календар	17
1.3.2 Tappsk	18
1.3.3 Evernote	19
1.3.4 Notes	20
1.3.5 Any.do	21
1.4 Опис технічного завдання	23
Висновки до розділу	25
2 Опис програми	26
2.1 Вибір засобів розробки вебресурсу	26
2.2 Опис середовища	29
2.2.1 Sublime Text	29
2.2.2 Опис МАР	31
2.3 Основні вимоги до запуску та роботи з вебресурсом	33
2.3.1 Вимоги до інтерфейсів інформаційної системи	33
2.3.2 Вимоги до ТЗ	34
2.3.3 Вимоги до ПЗ	34
2.4 Робота з даними	34
2.4.1 Вхідні дані	34
2.4.2 Операції обробки вхідних даних	34
Висновки до розділу	37
3 Розробка програми	38

	7
3.1 Структура вебресурсу	38
3.2 Проєктування та розробка БД.....	39
3.3 Опис основних форм інтерфейсу вебресурсу	43
Висновки до розділу	44
4 Експлуатація та тестування.....	45
4.1 Опис функціоналу	45
4.2 Тестування розробленої інформаційної системи.....	51
Висновки до розділу	54
Висновок	55
Перелік джерел посилання	57
Додаток А Текст програми.....	59
Додаток Б Слайди презентації	76

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

ASP.NET	– Active Server Pages для .NET
CSS	– Cascading Style Sheets
ECTS	– European Credit Transfer and Accumulation System;
HTML	– Hypertext Markup Language.;
HTTP	– Hypertext Transfer Protocol;
iOS	– iPhone OS
JS	– JavaScript
JSON	– JavaScript Object Notation;
MAMP	– Macintosh, Apache, MySQL і PHP
OS	– Operating System
PHP	– Hypertext Preprocessor;
SQL	– Structured Query Language
TCL	– Tool Command Language
XML	– Extensible Markup Language
БД	– База даних
ЗВО	– Заклад вищої освіти
ПЗ	– Програмне забезпечення
ТЗ	- Технічне забезпечення
СУБД	– Система управління базами даних

ВСТУП

В сучасний період багатьом людям відсутнє уявлення про ефективне планування свого часу, що призводить до марнотратства цінного ресурсу на неважливі справи. Завдяки відсутності структури в плануванні свого дня, тижня чи місяця, безглуздо губиться час, який міг бути використаний з користю і приносити прибуток.

Сьогодення вимагає від сучасної людини ефективного управління часом, яке стає ключем до успіху. З цією метою люди почали зосереджуватися на вивченні питань організації свого дня, зокрема, як забезпечити вільний час після роботи. Відтак, в 21 столітті, епоху технологій, народжуються різноманітні календарі, планувальники та органайзери, які допомагають зберігати всю необхідну інформацію.

Якщо раніше для планування часу використовувалися фізичні носії, де зберігався облік майбутніх транзакцій, то зараз на зміну їм приходять сучасні технології, де з'являється все більше нових можливостей, таких як легке сортування, класифікація та призначення міток електронним спискам справ і встановлювання нагадувань. Список справ - це зручний інструмент для спрощення повсякденного життя. Люди 21 століття весь час проводять на смартфонах, тому всі свої плани зберігають там.

Цільовою аудиторією вебресурсу є студенти закладів вищої освіти (ЗВО) та їх викладачі.

Метою створення дипломної роботи є розробка вебресурсу, який допоможе студентам більш ефективно організувати свій робочий процес та забезпечить можливість отримувати необхідну інформацію з будь-якого місця з доступом до Інтернету. Зараз цей застосунок має багато аналогів, але його функціонал буде являти собою поєднання кількох застосунків - Календар, Нагадування, Нотатки та Чат.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Опис предметної галузі

Метою даного дипломного проєкту є створення вебресурсу, що сприятиме студентам в більш ефективній організації їх робочого процесу.

Основні вимоги та функції, які має виконувати ця програма, включають наступне:

- забезпечення зручного інтерфейсу;
- не потребувати встановлення додаткового програмного забезпечення;
- забезпечення єдиного механізму для збереження та доступу до даних;
- можливість додавання, редагування та видалення даних в таблицях.

Для реалізації цих завдань, необхідно розробити вебресурс, який буде здійснювати задані маніпуляції з даними.

Відповідно до вимог теорії управління інформація, що передається по каналах зворотного зв'язку і використовується для управління, повинна задовольняти певному набору змістових і організаційних вимог:

- повнота – обсяг і зміст інформації, необхідної для побудови діагностичного висновку, що дозволяє приймати контрольні рішення;
- об'єктивність - відповідність отриманої інформації істинному стану контрольованої підсистеми (наприклад, рівень навчальних досягнень учнів);
- актуальність та оперативність - своєчасність отримання інформації з метою аналізу та оперативного прийняття рішення щодо необхідності та характеру коригуючих дій;
- безперервність у часі – постійний потік інформації в процесі навчання для підтримки стану керованої підсистеми та можливості безперервного регулювання процесу [1].

Контроль навичок студентів має бути комплексним і включати в себе всі частини навчальної програми; він також має бути системним, об'єктивним та індивідуальним. Під час контроль, проводиться комплексна перевірка всієї

навчальної діяльності учнів, у тому числі сформованості навичок, творчих здібностей, розвивається пізнавальний інтерес.

У навчальному процесі контроль виконує ряд функцій:

- перевірна функція – вказує на ступінь засвоєння навчального матеріалу навчальної дисципліни, засвоєння знань;
- дидактична функція - забезпечує не тільки систематизацію, а й узагальнення всього навчального матеріалу з дисципліни, що вивчається;
- корекційна функція – дозволяє виявити прогалини в знаннях, навичках і вміннях учнів з метою подальшого визначення оптимальних шляхів і шляхів їх виправлення;
- методична функція – дає змогу вчителю оцінити використані методи та внести оперативні корективи в навчально-виховний процес;
- виховна функція – формує в учнів відповідальність, здатність до самоорганізації та адекватної оцінки власної навчальної діяльності;
- мотиваційна функція – стимулює активність у навчальній діяльності за умови, що учень орієнтується не стільки на отримання оцінки, скільки на самоосвіту [2].

Умовою реалізації функцій моніторингу результатів навчання є використання різноманітних форм перевірки набутих знань і умінь. У системі вищої освіти традиційно використовуються такі види контролю:

- попередній контроль спрямований на виявлення вихідного рівня засвоєння учнів у результаті вивчення матеріалу на попередніх етапах навчання;
- поточний контроль дає можливість оцінити діяльність учня протягом усього навчального процесу, відображає ступінь сприйняття поточного матеріалу;
- проміжний контроль полягає у перевірці ефективності засвоєння студентами змісту навчального матеріалу з конкретної частини дисципліни;
- підсумковий контроль проводиться з метою оцінки результатів навчання, як правило, проводиться в кінці семестру у формі заліку або екзамену і завершує вивчення навчальної дисципліни [3].

У рамках даної роботи буде розглянуто поточний та підсумковий контроль, оскільки в сукупності ці форми контролю дають змогу комплексно оцінити успішність засвоєння студентами навчальної дисципліни.

Таким чином, при побудові моделі оцінки необхідно вирішити такі завдання:

- здійснити та обґрунтувати вибір адекватних по відношенню до вимірюваного об'єкта інформаційних оціночних параметрів;
- вибрати схему оцінювання;
- визначити алгоритм побудови підсумкової оцінки.

Побудова сучасної системи управління якістю освіти неможлива без використання об'єктивних методів кількісного вимірювання успішності освіти. Сучасні моделі навчання передбачають безперервну кількісну оцінку результатів навчання. З одного боку, сучасні моделі обліку успішності повинні надавати вчителю точну, постійну та вичерпну інформацію про хід навчального процесу, а з іншого – забезпечувати об'єктивність оцінювання та підвищувати мотивацію учнів до навчання.

Існує багато математичних моделей, які враховують поточний прогрес учнів; кожен з них має свої переваги і недоліки. Вибір системи оцінювання безпосередньо впливає на такі складові навчальної діяльності, як рівень мотивації учнів і, як наслідок, кінцеві результати навчання [4]. Давайте розглянемо найпоширеніші моделі для запису поточної продуктивності, спочатку розділивши їх на групи за видами контролю, що здійснюються.

Розглянемо моделі підсумкового (екзаменаційного) оцінювання.

1.1.1 Оцінка в кінці семестру

Наразі в освітній системі вищої освіти прийнято до використання 4-бальну систему оцінювання за шкалою від 2 до 5 («незадовільно» - «відмінно»). Використання цієї системи, як правило, передбачає усне опитування за квитками,

яке проводиться в кінці вивчення навчальної дисципліни. Учитель розмовляє з учнем по заздалегідь визначеному колу питань.

Ця чотириточкова система досить проста у використанні, але має ряд істотних недоліків:

- охоплює не весь матеріал дисципліни, а лише невелику його частину, як наслідок, порушення вимоги до повноти інформації;
- крайня упередженість, тобто. наявність сторонніх факторів (наприклад, серйозність вчителя, характер відносин між учнем і викладачем тощо);
- не враховується навчальна робота студента протягом семестру.

Багато дослідників констатують той факт, що традиційна вітчизняна контрольно-оціночна модель давно застаріла і потребує серйозної модернізації [5].

1.1.2 ECTS

В останні роки академічній мобільності студентів у навчальному процесі університету надається все більшого значення. Вона виступає як одна з ключових складових інтеграційних тенденцій розвитку Європейського простору вищої освіти [2]. Академічна мобільність дає змогу підтримувати спільні освітні програми та навчальні плани, розробляти універсальні методи оцінки якості освіти, а також здійснювати освітній обмін.

Під студентською академічною мобільністю розуміється можливість переміщення студента до іншої освітньої організації (у власній країні чи за кордоном) для навчання, після чого студент може повернутися до свого основного навчального закладу [3]. Запровадження загальновизнаної системи переведення має вирішити проблему отримання студентами окремих компонентів своєї освіти в університетах інших країн. Зазначимо, що окрім можливості перерахунку загальної суми витрат праці студента на оволодіння змістом навчальної дисципліни, ця система має передбачати також можливість перерахунку оцінки якості оволодіння набутими знаннями та вміннями. студентом з даної дисципліни [3].

Наразі в багатьох країнах популярна так звана система ECTS яка представляє собою систему обліку навчальної діяльності студентів під час оволодіння освітньою програмою чи курсом. Система ECTS виступає невід'ємним інструментом для впровадження Болонського процесу та забезпечує академічну та трудову мобільність студентів, а також випускників університетів.

Проблема переоцінки якості оволодіння набутими знаннями та вміннями залишається предметом дослідження та обговорення в європейській освітній спільноті. Питання про критерії перенесення студентських оцінок, отриманих за шкалою, розробленою університетом однієї країни, на аналогічну шкалу, прийняту в іншій країні, є досить складним і неоднозначним.

Система ECTS передбачає, що успішність студента буде зафіксована відповідно до прийнятої національної шкали оцінок, але в той же час рекомендується використовувати шкалу оцінювання ECTS, запропоновану Європейська комісія в середині 1990-х років. як компонент системи ECTS в цілому [6].

Основні положення системи ECTS:

- для досягнення очікуваних результатів навчання студенти повинні враховувати витрати в кредитах; 60 кредитів ECTS відповідають одному навчальному року, що становить приблизно 1500-1800 академічних годин. Отже, одна кредитна одиниця розраховується як 25-30 академічних годин;
- щоб отримати ступінь бакалавра, студент повинен набрати загалом від 180 до 240 ECTS-кредитів (залежно від тривалості навчання), а для отримання ступеня магістра потрібно мати не менше 300 кредитів;
- кількість кредитів повинна бути цілим числом; загальна сума кредитів ECTS за семестр має становити 30;
- заліки нараховуються студентам після закінчення навчальної діяльності за навчальною програмою в цілому або за окремим її структурним елементом та вдалої оцінки результатів навчання студента;
- оцінка не має відношення до кількості нарахованих кредитів з дисципліни;

- при зарахуванні кредитів студентам враховуються різноманітні форми навчальної діяльності, такі як участь у заняттях у аудиторії, виконання самостійних завдань, активна участь у семінарах, підготовка та виступи з доповідями, проходження практики на робочому місці, виконання лабораторних робіт, написання рефератів, курсових проектів та інших відповідних завдань;
- відвідування студентом аудиторних уроків не впливає на кількість нарахованих заліків, у свою чергу адміністрація ЗВО або викладач може передбачити зменшення кількості кредитів, які вже набрали з дисципліни, що вивчається, у разі невідвідування аудиторії.

Оцінювання за системою оцінювання ECTS здійснюється в кілька етапів. Спочатку встановлюється мінімальне значення, що вказує на успішність засвоєння дисципліни та отримання кредиту (наприклад, за сумою балів, набраних за виконання певних завдань), потім за сумою набраних балів студенти діляться на два підгрупи за такими умовами:

- бал «склав» отримують студенти, сума балів яких виявилася не меншою за певне значення;
- бал «не склав» ставиться учням, які набрали менше критичного значення, іншими словами, залік цим учням не присвоюється.

1.2 Аналіз завдань інформаційних систем для організації робочого процесу студента

Щоденний звіт є завершальним етапом роботи студента.

У процесі регулювання перевіряються:

а) наявність у звіті:

1. титульні листи;
2. зовнішній огляд;
3. зміст;
4. завдання;
5. пояснювальна записка;

6. робочі креслення;
7. застосунки;
8. список використаної літератури;

б) наявність необхідних підписів, віз та дат підписання звітів та доданих до неї документів.

в) відповідність вимогам для реєстрації:

1. титульні листи;
2. завдання;
3. зміст;
4. пояснювальна записка;
5. бібліографічний список;
6. застосунки.

г) наявність орфографічних, синтаксичних, стилістичних помилок та виправлень у тексті;

д) наявність, правильність та правильність нумерації сторінок;

е) дотримання розмірів полів, шрифту, відступу абзацу;

ж) відповідність нумерації сторінок, заголовків розділів (глав) та підрозділів (абзаців) на аркуші «Зміст» їх заголовкам та номерам сторінок у тексті;

з) правильне розділення тексту на розділи (глави), підрозділи (абзаци), пункти та підпункти та дотримання їх рекомендованих обсягів;

і) дотримання дистанції між текстом та заголовком, між заголовком та підзаголовком;

к) дотримання правил оформлення та нумерації формул, таблиць та ілюстрацій, наявність посилань на них у тексті;

л) наявність у тексті та правильність оформлення посилань на бібліографічні джерела.

1.3 Аналіз існуючих на ринку інформаційних систем для робочого процесу студента

1.3.1 Google Календар

Першим і найпопулярнішим застосунком є Google Calendar. Цей сервіс створений для планування зустрічей та заходів. Бета-версія програми була випущена в 2006 році, а офіційна версія - в 2009 році. Програма розроблена для платформ iOS і macOS. Застосунок дозволяє встановлювати час зустрічі, створювати повторювані події, встановлювати нагадування та запрошувати інших учасників через повідомлення електронної пошти [7].

Інтерфейс календаря Google відповідає стилю Material Design (рис. 1.1). Загальний дизайн виглядає зрозумілим для користувача завдяки розстановці проміжків і колірних акцентів. Візуальні елементи дат і днів тижня виділяються збільшенням розміру. Панель інструментів, яка дозволяє змінювати режим перегляду та керувати календарем, розташована вгорі, а рядок пошуку представлено піктограмою.

Для того, щоб мати можливість відображати й керувати кількома календарями на одній сторінці шляхом використання стовпців, користувачу необхідно обрати тип відображення дат. В залежності від відповіді користувача на запит про подію, її представлення матиме різний вигляд: "Так" буде позначатися одним кольором, "Можливо" - нахиленим рядком, а "Ні" - пунктирним рядком. Крім того, користувач може отримувати нагадування щодо дій або подій двома способами: за допомогою push-повідомлень або електронною поштою.

Календар використовує підключення до аккаунту Gmail.



Рисунок 1.1 – Інтерфейс календаря «Google»

Переваги Google Calendar: можливість доступу та керування календарем, швидкість роботи системи, локалізація, доступність кількома мовами. Крім того, Календар Google має єдиний дизайн для всіх продуктів Google. Користувачі продуктів Google можуть легко орієнтуватися в усіх графічних знаках і меню, оскільки вони однакові в усіх продуктах [7].

1.3.2 Tappsk

Другий застосунок під назвою Tappsk був створений у 2018 році. Це щоденник і планувальник завдань, який підтримує лише платформу iOS. Ця програма поєднує в собі планувальник завдань із щоденником і трекером звичок.

Ця програма має максимально інтерактивний інтерфейс (рис. 1.2), дуже легке вирішення завдань.

Основною функцією є список завдань. Є можливість розділити їх на етапи, розподілити за категоріями та керувати повідомленнями [6].



Рисунок 1.2 – Интерфейс «Tarrsk»

Також застосунок має функцію відстеження звичок допоможе вам позначити будь-яке завдання як рутину, щоб зберегти своє здоров'я та дисципліну.

1.3.3 Evernote

Evernote — це система, доступна в кількох операційних системах (рис. 1.3).

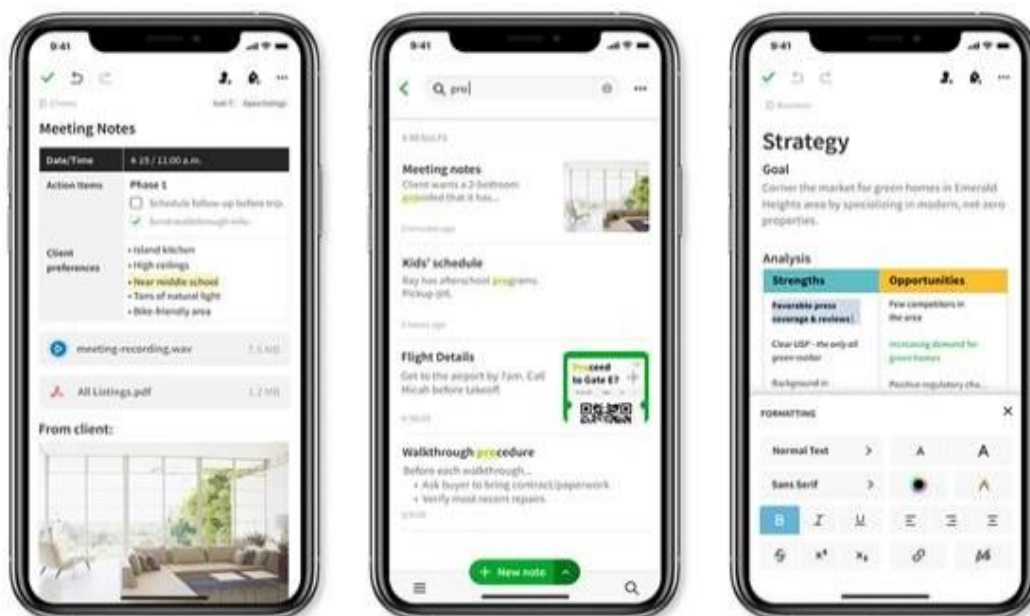


Рисунок 1.3 – Интерфейс «Evernote»

Дозволяє обробляти нотатки, ділитися нотатками з іншими користувачами та надсилати повідомлення користувачам електронною поштою [9].

Evernote демонструє свою сильну сторону в ефективному використанні текстового редактора. Його редактор пропонує розширені можливості форматування тексту, використання вбудованих таблиць та списків, шаблонів, а також підтримку Google Drive, голосового вводу, атачментів файлів та фотографій. Ще однією значною перевагою Evernote є зручна синхронізація між різними пристроями в рамках одного облікового запису.

1.3.4 Notes

Notes — це настільна та мобільна програма для операційних систем iOS та macOS [10]. За допомогою програми Notes користувачі можуть записувати свої думки, створювати ескізи, списки та миттєво додавати до них фотографії тощо (рис. 1.4).

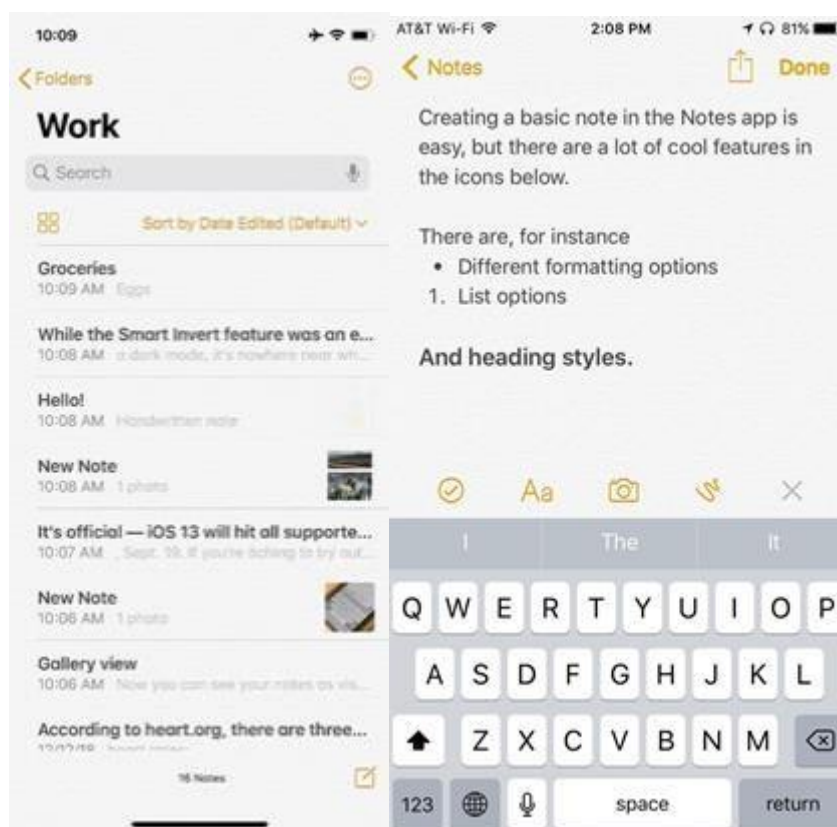


Рисунок 1.4 – Інтерфейс «Note»

Оскільки це стандартна програма для iOS, нотатки будуть синхронізовані на всіх пристроях користувача за допомогою iCloud.

Програма є важливою при використанні з операційною системою macOS, а її продуктивність, інтеграція зі стандартним набором програм операційної системи. Привабливість цього продукту для користувачів зумовлена його простим дизайном. Але недоліком програми є функціональність текстового редактора та незручність його використання. Викликати верхній регістр можна лише через контекстне меню, інші способи недоступні..

1.3.5 Any.do

Any.do - це програмний продукт, призначений для ефективного управління нотатками та нагадуваннями. Основна ідея застосунку полягає в тому, що він не розрізняє ці дві концепції. Згідно з концепцією програми, головна відмінність між нагадуваннями і замітками полягає лише в наявності часу повідомлення для нагадувань, в інших аспектах вони майже однакові.

Any.do організовує завдання за датою нагадування: сьогодні, завтра, майбутні та колись. Синхронізація відбувається вручну, і користувач може легко перетягувати завдання між списками та нотатками. Крім того, програма дозволяє прикріплювати файли, записувати звуки та додавати нагадування до будь-якого елемента (рис. 1.5). Користувач також може переглядати та керувати своїм календарем пристрою разом із завданнями в програмі.

Any.do пропонує багато корисних додатків, включаючи додаткову "панель швидкого додавання", яка дозволяє розміщувати щоденні завдання на панелі сповіщень, а також набір привабливих віджетів головного екрана та розумну систему спливаючих сповіщень, що нагадують про пропущені дзвінки.

Програма має зрозумілий та простий у навігації інтерфейс з безліччю корисних функцій для створення, організації та, що найважливіше, просування практично будь-якого типу списків. Інтерфейс Any.do зображен на рисунку 1,5 [11].

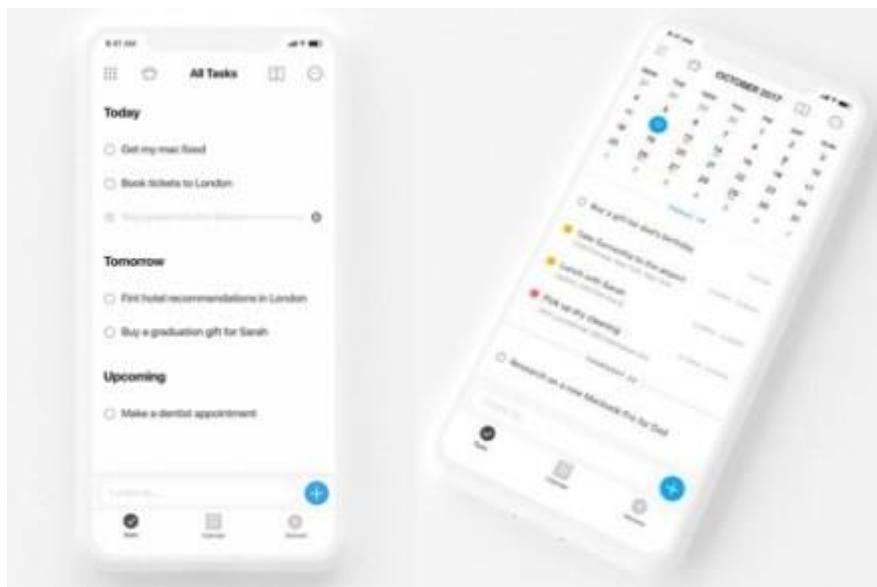


Рисунок 1.5 – Інтерфейс «Any.do»

Мінімалістичний дизайн робить інтерфейс дуже простим і зручним для всіх. У нижній панелі навігації є три вкладки: «Завдання», «Календар» і «Моменти». На сторінці «Завдання» всі замітки відсортовані за датою, і користувач одразу може побачити не лише свої плани на сьогодні, а й на весь тиждень. Користувачі можуть виконати завдання, натиснувши на порожнє коло, і виконане завдання буде перекреслено. Під час написання тексту для нотаток і нагадувань можна використовувати кольорові мітки та позначати користувачів.

Основні функції прості, але платна підписка на продукт включає такі функції, як додавання повторюваних завдань, кольорові мітки та нотатки, геолокаційні нагадування тощо.

Аналіз аналогів популярних мобільних застосунків показав, що жодне з них не має функції чату [12]. Тому включення цієї можливості при створенні програми-організатора Ostorus було ключовою відмінною рисою. Ця програма дозволяє зв'язати кілька облікових записів (наприклад, сім'ї чи друзів), щоб користувач міг бачити, у кого які плани на сьогодні, додавати учасників до власної події, але,

найголовніше, це можливість спілкуватися один з одним. Чат у Календарі також є зручним способом надсилання документів, фотографій чи іншого вмісту.

Основна мета цього дослідження – розробка вебресурсу, який допоможе студентам більш ефективно організувати свій робочий процес та забезпечить можливість отримувати необхідну інформацію з будь-якого місця з доступом до Інтернету.

Об'єктом дослідження даної роботи є вебресурси для організації робочого процесу студента.

Теми дослідження – методи розробки вебресурсу для організації робочого процесу студента.

Для досягнення поставленої мети необхідно вирішити ці завдання:

- аналіз предметної галузі;
- сучасний підхід до аналізу дизайну інтерфейсу органайзера;
- аналіз існуючих стандартів і показників якості інтерфейсів мобільних застосунків;
- розробка методів проєктування інтерфейсу органайзера, експериментальна оцінка технічної ефективності.

1.4 Опис технічного завдання

Метою кваліфікаційної роботи є розробка зручного вебресурсу для організації робочого процесу студента.

Основна ціль програми полягає в частковій автоматизації процесу роботи студентів, з метою полегшення їхньої роботи у закладах вищої освіти. Система, що має високий рівень реалізації, спроектована для полегшення роботи студентів у ЗВО.

Система передбачає розділення прав доступу та призначена для двох категорій користувачів: адміністраторів та відвідувачів сайту. Гості сайту мають реєструватися у системі.

Система для відвідувачів сайту має надавати можливість:

- реєстрація у системі;
- пошук послуг за певними фільтрами;
- перегляд інформації про звіти;
- перегляд списку коментарів до звіту;
- зручний та зрозумілий для користувача інтерфейс з підказками.

Одна з функціональних складових системи створена для забезпечення користувачеві наступних можливостей:

- автентифікація у системі користувача;
- перегляд списку звітів.
- додавання звіту;
- перегляд та зміна профілю користувача;
- перегляд історії роботи зі звітами, зроблених користувачем;
- видалення звіту;
- зміна пароля;
- додавання коментарів;
- усі можливості, доступні для відвідувачів вебсайту.

Система для адміністрації сайту має надавати наступні можливості:

- перекриття доступу до системи для інших гостей;
- можливість додавати, видаляти та змінювати дані про користувачів;
- здатність додавати, видаляти та змінювати інформацію про звіти;
- всі можливості, які надаються користувачам.

При роботі з введенням та редагуванням даних ця система має перевіряти коректність введених даних та уникати ситуацій, що можуть спричинити помилкові дії з боку гостей.

Для полегшення підтримки, обслуговування та подальшого розширення програми, її архітектура повинна бути заснована на трьох рівнях (клієнтський рівень, сервер застосунків та база даних), що дозволяє побудувати програмні системи з клієнтською та серверною частинами. Серверна частина відповідає за

можливість доступу до БД та містить бізнес-логіку програми, тоді як клієнтська частина реалізує користувацький інтерфейс.

Для розробки вебресурсу рекомендовано використовувати базу даних як рівень доступу до даних, що забезпечує централізоване та структуроване зберігання всіх системних даних. Крім того, база даних надає низькорівневі сервіси, такі як читання, збереження та зміна структури даних. На цьому рівні буде створена база даних для зберігання всіх даних системи. Щоб забезпечити цілісність та узгодженість, необхідно використовувати відповідні процедури, тригери та інші об'єкти, які надає сервер, щоб контролювати та забезпечувати сумісність..

На сервері застосунків буде встановлене ядро системи, яке включатиме рівень бізнес-логіки. Цей рівень має фокусуватися на таких аспектах системи:

- аутентифікація користувачів системи та перевірка їхнього доступу;
- правила обробки інформації, такі як перевірка точності введення даних користувачем та організація зв'язків між даними;
- клас, який забезпечує підключення до бази даних;
- класи та алгоритми для роботи з таблицями бази даних та виконання збережених процедур та функцій на сервері.

Інтерфейс, створений для даного проєкту, може використовувати гіпертекстову розмітку, каскадні таблиці стилів та скриптову мову браузера для його реалізації.

Висновки до розділу

Таким чином, у цьому розділі було проаналізовано завдання інформаційних систем вебресурсу. Розглянуто існуючі на ринку аналогів інформаційних систем. В результаті було зроблено висновок та постановку технічного завдання.

2 ОПИС ПРОГРАМИ

2.1 Вибір засобів розробки вебресурсу

Перш ніж описати методи та інструменти, які використовуються для створення сайту, розглянемо кілька термінів.

Front-end (інтерфейс) включає такі вебтехнології як CSS, HTML, JavaScript, що може використовуватися для створення інтерфейсу користувача.

Back-end це сторона сервера, де знаходиться ядро вебсайту, і це ядро забезпечує основні функції та підтримку інтерфейсу. На серверній частині використовуйте мову програмування сервера, таку як PHP, ASP.NET, Python, Ruby, Perl тощо. Серверні мови програмування призначені для роботи з такими базами даних, як MySQL, MongoDB та ін.

Загалом, найпопулярнішими мовами програмування Back-end є:

- PHP — мова програмування на стороні сервера, розроблена спеціально для написання вебзастосунків. Спочатку він був розроблений для відстеження відвідувачів вебсайту. З часом PHP перетворився з набору інструментів у повноцінну ООП (об'єктно-орієнтовану) мову програмування, код якої також можна вбудовувати в HTML. Коли користувач відвідує вебсервер, PHP створює сторінку HTML і надсилає її назад у браузер користувача;

- Python — це високорівнева, інтерпретована мова програмування, створена Гвідо ван Россумом у 1991 році. Вона має простий і зрозумілий синтаксис, що робить її дуже популярною серед початківців і досвідчених програмістів;

- Ruby - це динамічна, об'єктно-орієнтована мова програмування з елегантним синтаксисом, створена Юкіхіро Мацумото у 1995 році. Вона надає розширені можливості для розробки вебзастосунків та скриптів;

- MySQL — це система керування базами даних (СУБД). База даних являє собою структурований набір даних. В ньому може міститися будь-яка інформація — від простих списків покупок до великих обсягів даних, які використовують в корпоративних мережах.

Аналіз мов програмування представлено на рисунку (рисунок 2.1) [13].

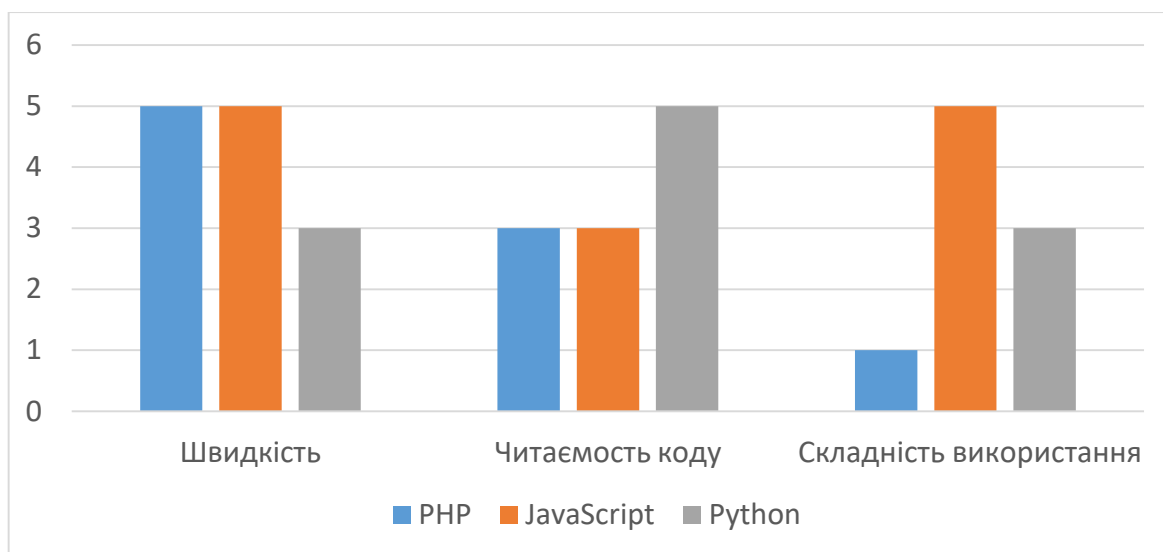


Рисунок 2.1 – Аналіз мов програмування

Розробниками також найчастіше використовуються такі мови програмування Front-end:

- HTML — це мова розмітки гіпертекстових документів. HTML описує структуру вебсторінки за допомогою тегів (елементів HTML), які визначають розміщення тексту, графіки, таблиць тощо;
- CSS являє собою правила, які описують стиль (вигляд) документа HTML. За допомогою CSS вебсторінки отримують візуально зрозумілий інтерфейс. Тобто CSS описує правила тегів HTML, тексту та зображень, тим самим змінюючи відносне розташування, розмір, колір, шрифт та інші візуальні зміни елементів;
- JavaScript — це динамічна мова програмування. Він призначений для виконання будь-яких програмних операцій у браузері, забезпечуючи тим самим інтерактивність вебсторінок. JavaScript дозволяє керувати елементами HTML, стилями CSS для вебсторінок тощо.

Обмін даними між інтерфейсом і сервером використовує протокол HTTP. Це протокол для обміну інформацією в Інтернеті. [14].

На рисунку 2.2, приклад взаємодії Front-end та Back-end. Якщо користувач відвідує вебресурс, йому у відповідь приходить вебсторінка (Front-end), яка була сформована сервером (Back-end).

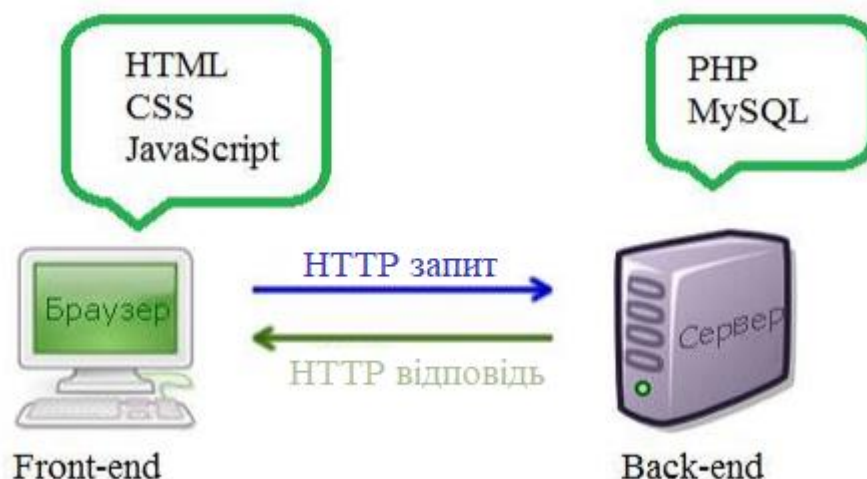


Рисунок 2.2 - Приклад взаємодії Front-end та Back-end

В роботі було використано для реалізації Front-end частини CSS та JavaScript, а для Back-end частини використовується PHP та MySQL.

В роботі також було використано технологія CSS FlexBox.

Flexbox — це відносно нова інтерфейсна функція, яка робить створення макета вебсайту (і робить його адаптивним) набагато простішим, ніж це було раніше.

На рисунку представлено рейтинг популярності серед систем управління контентом (рисунок 2.3) [15].

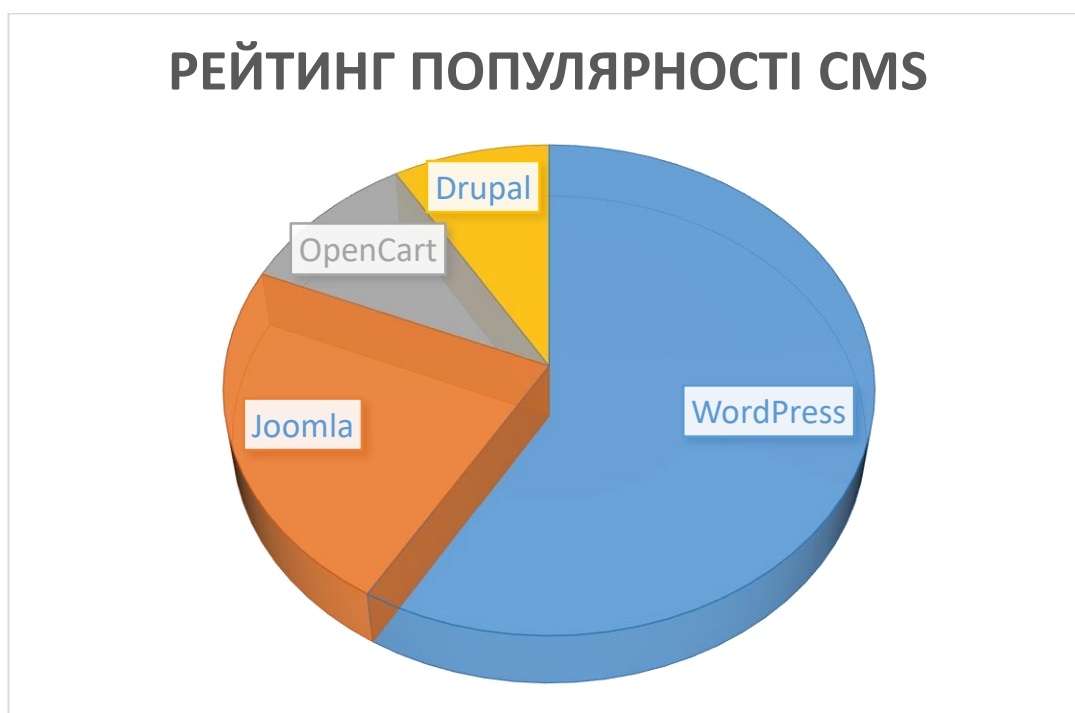


Рисунок 2.3 - Рейтинг популярності серед систем управління контентом

2.2 Опис середовища

2.2.1 Sublime Text

Для розробки вебресурсу буде використовуватися середовище розробки Sublime Text.

Sublime Text являє собою власний кросплатформний текстовий редактор, який підтримує різні плагіни, включаючи ті, що працюють з мовою програмування Python. Розробники надають можливість безкоштовного ознайомлення з програмою без будь-яких обмежень, але наявність ліцензії є обов'язковою для його використання (рисунок 2.4).



Рисунок 2.4 – Sublime Text

Sublime Text підтримує велику кількість мов програмування та забезпечує підтримку C, C++, C#, CSS, D, Dylan, Erlang, HTML, Groovy, Haskell, Java, JavaScript, LaTeX, Lisp, Lua, Markdown, MATLAB, OCaml, Perl, PHP, Python, R, Ruby, SQL, TCL і XML [16].

На додаток до тих мов програмування, які включені за замовчуванням, користувачі можуть вибрати завантаження плагінів для підтримки інших мов [17].

Sublime Text може бути оснащений менеджером пакетів, який дозволяє користувачам знаходити, встановлювати, оновлювати та видаляти пакети без перезапуску програми. Менеджер підтримує встановлені пакети в актуальному стані, завантажуючи нові версії зі сховища. Крім того, він надає команди для активації та деактивації встановлених пакетів [18].

Редактор містить різноманітні візуальні теми, а додаткові теми доступні для завантаження.

Користувачі можуть бачити всі свої коди у вигляді міні-карти в правій частині екрана, на якій вони можуть переміщатися, натискаючи на неї.

Є кілька режимів екрану. Одна з них включає від 1 до 4 панелей, і ви можете відображати до чотирьох файлів одночасно.

Повний режим (вільний режим) показує лише один файл без будь-яких інших меню навколо нього.

При введенні коду, Sublime Text пропонує різні варіанти автодоповнення, залежно від використовуваної мови. Крім того, редактор автоматично заповнює створені користувачем змінні.

Темна тема Sublime Text призначена для підвищення контрастності тексту. Основні синтаксичні елементи позначаються різними кольорами, які краще гармоніюють з темним фоном, ніж з світлим.

Sublime Text дозволяє користувачам компілювати та запускати програми без переходу до командного рядка. Крім того, користувачі можуть налаштувати свою систему збірки та ввімкнути автоматичну збірку програми кожного разу при збереженні коду.

Існує також навігаційний інструмент, який дозволяє користувачам переходити між файлами та вміщеними в них елементами за допомогою нечіткого пошуку.

Додатково була реалізована функція автоматичного збереження, щоб допомогти користувачам уникнути втрати проділеної роботи.

Налаштування клавіатурних скорочень та інструментів навігації дозволяють призначати клавіатурні комбінації для меню та панелей інструментів (лише у першій версії, у другій та третій - командній палітрі).

Функція пошуку під час введення використовується для знаходження документів.

Засіб перевірки граматики працює подібним чином, перевіряючи правильність під час введення.

Останню дію можна автоматизувати і повторити за допомогою макросів.

2.2.2 Опис MAMP

Для роботи з серверною частиною було обрано локальний сервер MAMP. MAMP означає Macintosh, Apache, MySQL і PHP (рисунок 2.5).



Рисунок 2.5 - MAMP

Після встановлення MAMP користувачу потрібні: Apache, MySQL і PHP.

Існує дві версії програми: платна MAMP PRO і безкоштовна MAMP, які відрізняються функціональністю та доступом до контенту та керування сервером.

Стандартний дистрибутив включає дві версії. Якщо ви використовуєте безкоштовну версію, досить скопіювати її в папку з програмою і почати працювати. Якщо ви купуєте ліцензію на платний продукт, щоб використовувати MAMP PRO, вам потрібно буде інсталювати обидві версії, оскільки MAMP і MAMP PRO складаються разом. В основному списку включених програм доступні всі ярлики для обох версій, але не рекомендується запускати їх одночасно.

Для роботи з таблицями даних використовується phpMyAdmin.

phpMyAdmin – це безкоштовний програмний інструмент, призначений для адміністрування СУБД MySQL через браузер (рисунок 2.6). phpMyAdmin надає розширений набір функцій для роботи з MySQL. Цей інструмент дозволяє керувати базами даних, таблицями, зв'язками, індексами, користувачами, дозволами та багато іншого.

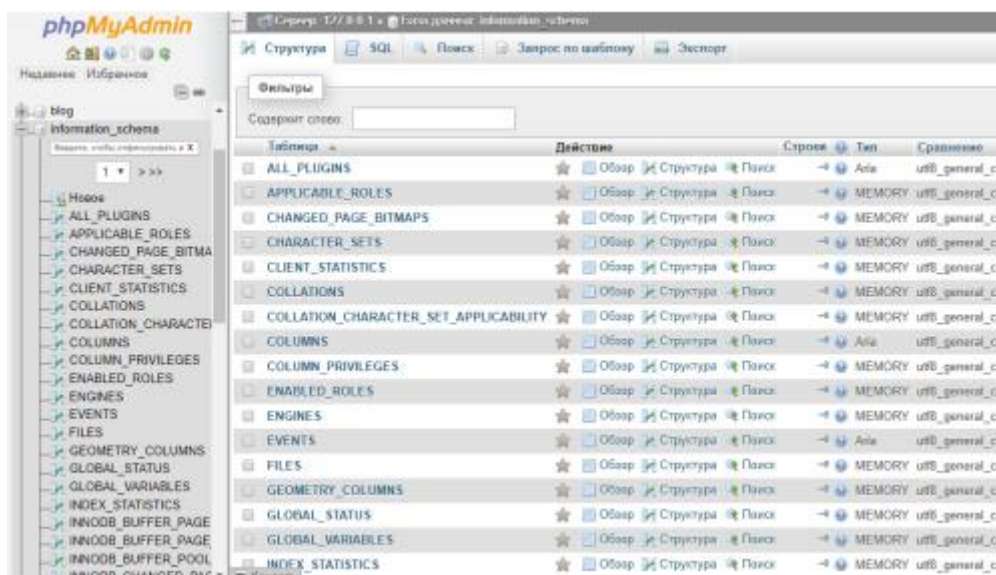


Рисунок 2.6 - phpMyAdmin

2.3 Основні вимоги до запуску та роботи з вебресурсом

2.3.1 Вимоги до інтерфейсів інформаційної системи

Для інтерактивної обробки інформації розроблена система повинна містити такі інтерфейси:

- форма введення даних;
- форма для введення запиту та параметрів для формування звітів та їх друку;
- форма вибору режимів друку вихідних документів.

Витримувати колірну палітру інтерфейсів у стилі Windows. Роздільна здатність екрану: 1280*1024 - великий шрифт.

Інтерфейс має бути захищений від некоректних дій користувача та мати систему підказок.

Передбачити можливість виклику форм редагування довідників у разі потреби з форми введення та коригування даних.

2.3.2 Вимоги до ТЗ

Для введення та корекції даних онлайн пропонується використовувати декілька комп'ютерів, підключених до локальної мережі.

Вимоги до серверної частини - персональний комп'ютер стандартної конфігурації, вільне місце на жорсткому диску 1 Гб.

Вимоги до клієнта - відповідно до рекомендованих вимог операційної системи.

2.3.3 Вимоги до ПЗ

Зі сторони сервера - операційна система Linux або Windows, СУБД MySQL, вебсервер MAMP.

Вимоги до користувача - операційна система Linux або Windows, браузер - Opera 10.

2.4 Робота з даними

2.4.1 Вхідні дані

У програмному застосунку вхідними даними є:

Основний функціонал:

- ведення обліку відвідуваності предметів;
- виставлення оцінок по предметам;
- перегляд журналу;
- додавання та видалення предметів, по яких ведеться відвідуваність.

Основні функції, які необхідно реалізувати в застосунку, впливають із вимог до програми, що розробляється.

2.4.2 Операції обробки вхідних даних

На рисунку 2.7 представлена діаграма обробки даних, що дає загальне уявлення функціонального призначення програми.



Рисунок 2.7 - Діаграма варіантів використання та обробки даних

Розглянемо більш детально варіанти використання.

Назва прецеденту: додавання предмета.

Мета: додати предмет.

Передмова: відкритий головний екран програми (список предметів).

Головна послідовність:

- користувач натискає кнопку із символом плюс;
- відкривається форма додавання предмета, що містить поле для введення назви предмета та опису;
- користувач вводить назву предмета;
- новий предмет доданий та відображений у списку предметів.

Альтернативна послідовність (повернення до списку предметів без додавання предмета):

- користувач натискає кнопку із символом плюс;
- відкривається форма додавання предмета, що містить поле для введення назви предмета та опису;
- користувач вводить дані в поля форми;

- відбувається повернення до списку предметів;

Альтернативна послідовність (додавання предмета, що вже є у списку предметів):

- користувач натискає кнопку із символом плюс;
- відкривається форма додавання предмета, що містить поле для введення назви предмета та опису;
- користувач вводить дані в поля форми;
- користувачеві відображається повідомлення про те, що даний предмет уже додано.

Назва прецеденту: видалення предмета.

Ціль: видалити предмет, неактуальний для користувача.

Передмова: користувач авторизований, відкритий головний екран програми (список предметів).

Головна послідовність:

- користувач натискає на кнопку із зображенням хрестика, що знаходиться навпроти предмета, який хоче видалити;
- відбувається видалення предмета та з'являється вікно з оновленим списком предметів.

Назва прецеденту: додавання заняття чи роботи.

Мета: додати заняття, відвідування якого відзначається або роботу, за яку виставляється оцінка.

Передумова: користувач авторизований, відкритий журнал.

Головна послідовність:

- користувач натискає на кнопку із символом плюс;
- відкривається форма додавання роботи або заняття, що містить перемикач «відвідування/оцінка», поля для введення назви та дати;
- користувач заповнює поля форми;
- нова робота або нове заняття додано та відображено в журналі групи.

Альтернативна послідовність (повернення до журналу без додавання або заняття):

- користувач натискає на кнопку із символом плюс;
- відкривається форма додавання роботи або заняття, що містить перемикач «відвідування/оцінка», поля для введення назви та дати;
- користувач заповнює поля форми;
- відбувається повернення до журналу (при цьому дані, введені в поля форми, не зберігаються).

Альтернативна послідовність (додавання предмета, що вже є у списку предметів):

- користувач натискає на кнопку із символом плюс;
- відкривається форма додавання роботи або заняття, що містить перемикач «відвідування/оцінка», поля для введення назви та дати;
- користувач заповнює поля форми;
- користувачеві відображається повідомлення про те, що ця робота або заняття вже додано.

Назва прецеденту: видалення роботи чи заняття.

Мета: видалити неактуальну роботу або заняття.

Передумова: користувач авторизований, відкритий журнал.

Головна послідовність:

- користувач натискає на кнопку із зображенням хрестика, що знаходиться навпроти заняття та роботи, яку хоче видалити;
- відбувається видалення роботи або заняття та з'являється вікно з оновленим списком робіт та занять для обраної групи.

Висновки до розділу

У другому розділі проаналізовано вибір інструментів для розробки вебсервісу, розглядається середовище та системи для розробки вебресурсу для організації робочого процесу студента. В результаті було зроблено висновки стосовно обрання засобів розробки.

3 РОЗРОБКА ПРОГРАМИ

3.1 Структура вебресурсу

Після аналізу сучасних підходів до проектування інтерфейсу було сформовано етапи розробки дизайну інтерфейсу вебресурсу (рис. 3.1):

а) аналіз користувача. Щоб розробити хороший користувацький інтерфейс для мобільної програми, користувачу спочатку потрібно знати своїх користувачів. Користувачу потрібно зрозуміти, що спонукає їх використовувати застосунок, а потім почати розробку;

б) створення макету. Warframe — модель інтерфейсу мобільних застосунків. Все починається з ескізу на папері. Він створює шаблони для кожного екрана програми;

в) дизайн інтерфейсу. Зовнішній вигляд є одним з найважливіших етапів створення мобільного застосунку. Зрештою, це перше, на що звертають увагу користувачі, заходячи в програму. Але щоб створити дизайн, який сподобається користувачам, необхідно враховувати наступні параметри:

1. особливості керівництва по платформі ;
2. унікальність;
3. зручність інтерфейсу;
4. зменшити гучність інтерфейсу;
5. основні елементи повинні бути масштабними;
6. менше тексту та більше візуалізації;
7. певні функції;
8. оброблені іконки;

г) тестування.

Після того як дизайн мобільного застосунку готовий, його залишилося протестувати. Це допомагає виявити недоліки та представити досконалість клієнтам.

3.2 Проектування та розробка БД

Для ефективного зберігання даних користувачів, документи, відгуки та комунікацію рекомендується використовувати базу даних.

У процесі розробки було створено шість таблиць для організації бази даних. Згідно з обраною технологією також для управління БД було обрано СУБД MySQL.

СУБД MySQL використовується в системі управління контентом для забезпечення функціональності бази даних. База даних включає в себе наступні таблиці:

- звіти;
- повідомлення;
- відгуки;
- користувачі;
- активні користувачі;
- зразки документів.

У таблиці «Users» (таблиця 3.1) знаходиться інформація про гостей, що зареєструвалися. Таблиця містить 8 полів.

Таблиця 3.1 - Таблиця «Users»

№	Атрибут	Семантика	Тип
1	id	Унікальний ідентифікатор	int(5)
2	login	Логін для авторизації	varchar(50)
3	password	Пароль для авторизації	varchar(50)
4	sername	Прізвище користувача	varchar(50)
5	name	Ім'я користувача	varchar(50)
6	fotourl	Посилання на фото	varchar(50)
7	address	Електронна адреса користувача	varchar(100)
8	phone	телефон користувача	varchar(20)

У таблиці «Reviews» (таблиця 3.2) зберігаються відгуки клієнтів. Таблиця складається з 4 полів.

Таблиця 3.2 – Таблиця «Reviews»

№	Атрибут	Семантика	Тип
1	id	Унікальний ідентифікатор студента	int(5)
2	textotziv	Текст відгуку	varchar(500)
3	username	Ім'я користувача	varchar(50)
4	fotourl	Фото користувача	varchar(50)

У таблиці «Docs» (таблиця 3.3) знаходиться інформація про запити користувачів, що зареєструвалися.

Таблиця 3.3 - Таблиця «Docs»

№	Атрибут	Семантика	Тип
1	id	Унікальний ідентифікатор студенту	int(5)
2	name	Фото	varchar(100)
3	disciplina	Дисципліна	varchar(100)
4	communication	Засоби комунікації	varchar(100)
5	complite	Що зроблено	varchar(100)

Аналогічна структура і для інших таблиць.

Вся інформація зберігається на сервері в окремих БД (рис. 3.1).

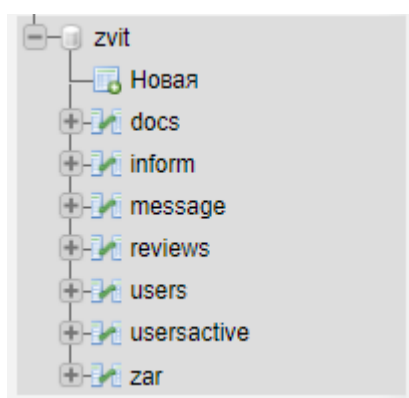


Рисунок 3.1 – Структура БД (Загальна)

Нижче вказана структура БД яка відповідає за звіт (рис.3.2).

+ Параметри

←T→	id	name	disciplina	communication	complate
☐ Изменить Копировать Удалить	1	Іванов Іван Іванович	Дипломна робота	Viber	Перевірено роботу

Рисунок 3.2 – Дані в БД (Звіт)

Нижче зображена частина БД яка включає в себе повідомлення (рис.3.3).

←T→	id	message	d	tosend
☐ Изменить Копировать Удалить	1001	Hello	2022-11-27 17:42:40	1002
☐ Изменить Копировать Удалить	1002	Hi	2022-11-27 17:42:48	1001
☐ Изменить Копировать Удалить	2		2022-12-03 15:28:39	

Рисунок 3.3 – Дані в БД (Повідомлення)

Нижче зображена структура БД яка включає в себе відгуки користувачів (рис.3.4)

+ Параметри

←T→	id	textotziv	username	fotourl
☐ Изменить Копировать Удалить	1	Це дуже добрий сайт для роботи з документами	@svitlana_m	src="img/p2.jpg"
☐ Изменить Копировать Удалить	2	Nice	@alex_v	src="img/p1.jpg"
☐ Изменить Копировать Удалить	3	Я обираю цей сайт для контролю документів	@kira2012	src="img/p3.jpg"
☐ Изменить Копировать Удалить	4	Я задоволений цим сайтом	@vlad01	src="img/p4.jpg"
☐ Изменить Копировать Удалить	5	Very well!!!	a89307	src="img/p3.jpg"

Рисунок 3.4 – Дані БД (Відгуки)

Нижче вказана частина БД яка включає в себе список користувачів (рис.3.5)

+ Параметри

←T→	id	login	password	sername	name	fotourl	adress	phone
☐ Изменить Копировать Удалить	1	admin@gmail.com	admin	admin	admin	src="uploads/p1.jpg"	admin	+1234567890

Рисунок 3.5 – Дані БД (Користувачі)

Нижче зображена структура БД яка включає в себе список активних користувачів (рис.3.6)

+ Параметры						
← T →				id	d	ready
☐		Изменить		Копировать		Удалить
				1	1682192335	0

Рисунок 3.6 – Дані БД (Активні користувачі)

На рисунку зображена структура БД яка включає в себе список зразків документів (рис.3.7)

+ Параметри

			id	fotoUrl	name	description				
☐		Изменить		Копировать		Удалить	11	src="img/zar1.jpg"	Щоденник	Щоденник з практики
☐		Изменить		Копировать		Удалить	12	src="img/zar2.jpg"	Звіт	Звіт з практики
☐		Изменить		Копировать		Удалить	13	src="img/zar3.jpg"	Заява	Заява на проживання у гуртожитку
☐		Изменить		Копировать		Удалить	14	src="img/zar4.jpg"	Заява	Заява на академічну доброчесність
☐		Изменить		Копировать		Удалить	15	src="img/zar5.jpg"	Наказ	Наказ на отримання стипендії

Рисунок 3.7 – Дані БД (Зразки)

Нижче зображено інформацію про користувачів (рис.3.8)

+ Параметры

			id	sername	name	classnum	room	program	history	
☐	 Изменить	 Копировать	 Удалить	1	Іванов	Іван	2	62	84	73
☐	 Изменить	 Копировать	 Удалить	2	Петров	Петр	3	83	91	73
☐	 Изменить	 Копировать	 Удалить	3	Федоренко	Максим	1	88	77	99
☐	 Изменить	 Копировать	 Удалить	4	Макаренко	Дмитро	2	94	79	81

Рисунок 3.8 – Дані БД (Інформація)

Підключення до бази даних відбувається за допомогою коду, наведеного у додатку Г.

3.3 Опис основних форм інтерфейсу вебресурсу

Логотип вебсайту зображено на рисунку (рис. 3.9).



Рисунок 3.9 – Логотип

Всі сторінки містять в собі header (рис. 3.10) Aside (рис. 3.11) та footer (рис. 3.13).

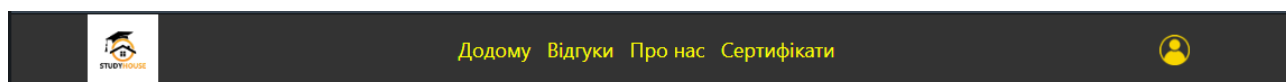


Рисунок 3.10 - Header



Рисунок 3.11 – Aside

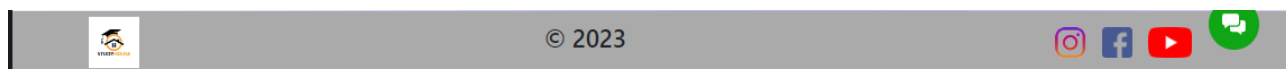


Рисунок 3.12– Footer

Висновки до розділу

В цьому розділі було описано призначення та функції розробленого вебсайту, а також наведено мінімальні апаратні вимоги для його встановлення та використання. Інтерфейс користувача вебсайту є простим і легким у використанні, не вимагає додаткового навчання для користувача.

4 ЕКСПЛУАТАЦІЯ ТА ТЕСТУВАННЯ

4.1 Опис функціоналу

Розглянемо приклад реалізації розробленої системи. При введенні адреси сайту відвідувач потрапляє на його головну сторінку (рис. 4.1). Зовнішній вигляд визначає адміністратор системи.

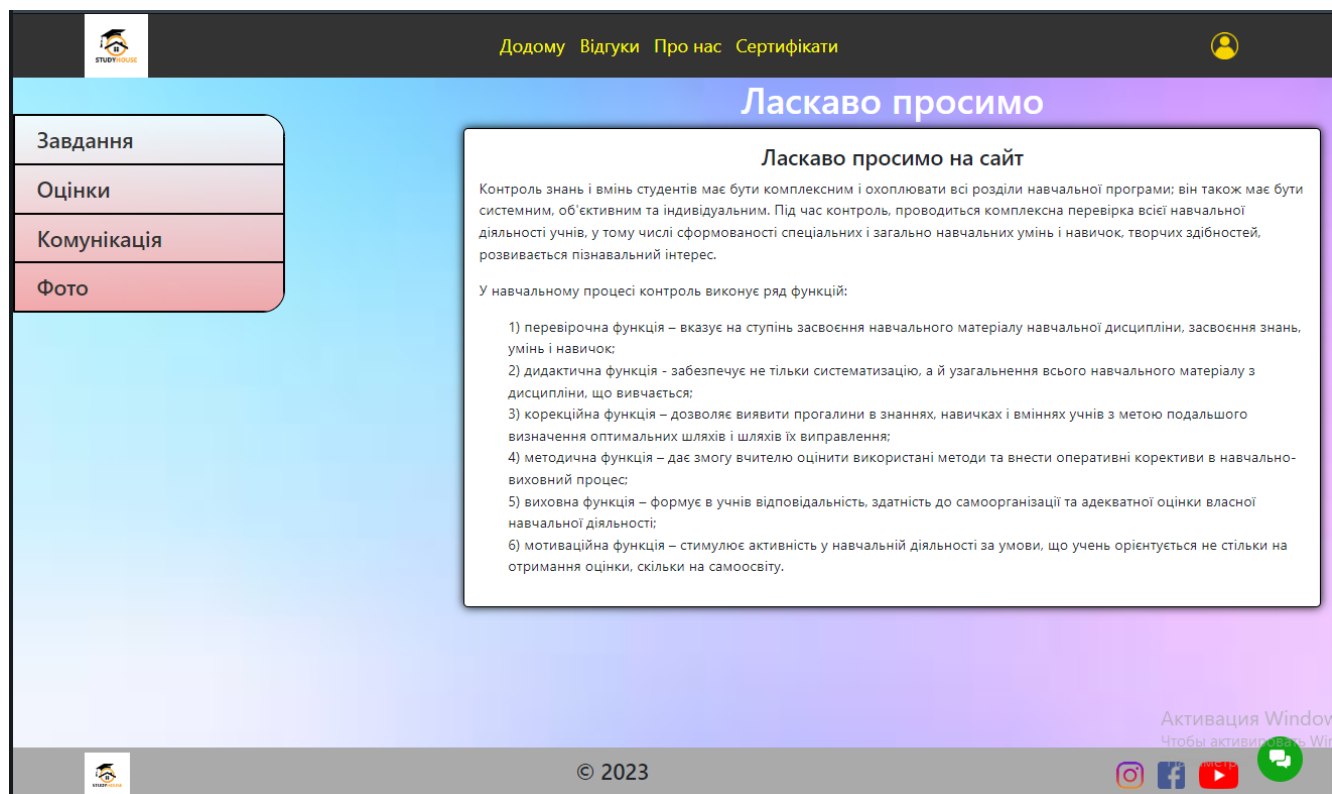


Рисунок 4.1 – Головна сторінка

З головної сторінки можна перейти на сторінку з фото та на сторінку з комунікацією (рис. 4.2).

Вид сторінки «Фото» представляє собою сторінку, куди студенти можуть за бажання додавати свої фотографії.

Вид сторінки комунікація представляє собою сторінку за допомогою якої користувачі можуть тримати зв'язок один з одним.



Рисунок 4.2 – Сторінка «Комунікація»

З головної сторінки також можна перейти на сторінку з завданнями (рис. 4.3).

Додому Відгуки Про нас Сертифікати

Завдання

Додавання завдання Зразки документів Заповнення документів

#	ПІБ Викладача	№ курсу	Дисципліна	Комунікація	Що зроблено
1	Іванов Іван Іванович	5	Історія	Аудиторія	Отримано навчальні матеріали
2	Петров Петр Петрович	2	Програмування	Zoom	Прослухана лекція
3	Павел Сергійович	4	Математика	Онлайн	Отримана консультація

ПІБ № курсу Дисципліна Засоби комунікації Що зроблено

Внести завдання

Номер

Очистити таблицю Очистити за номером

Зберегти

© 2023

Рисунок 4.3 – Сторінка «Завдання»

На сторінці завдання, можна додати завдання, якщо користувач увійшов у систему як зареєстрований користувач, заповнивши відповідні поля. Також є вкладка «Зразки документів», на якій зображено перелік документів, які є в базі даних (рис. 4.4).

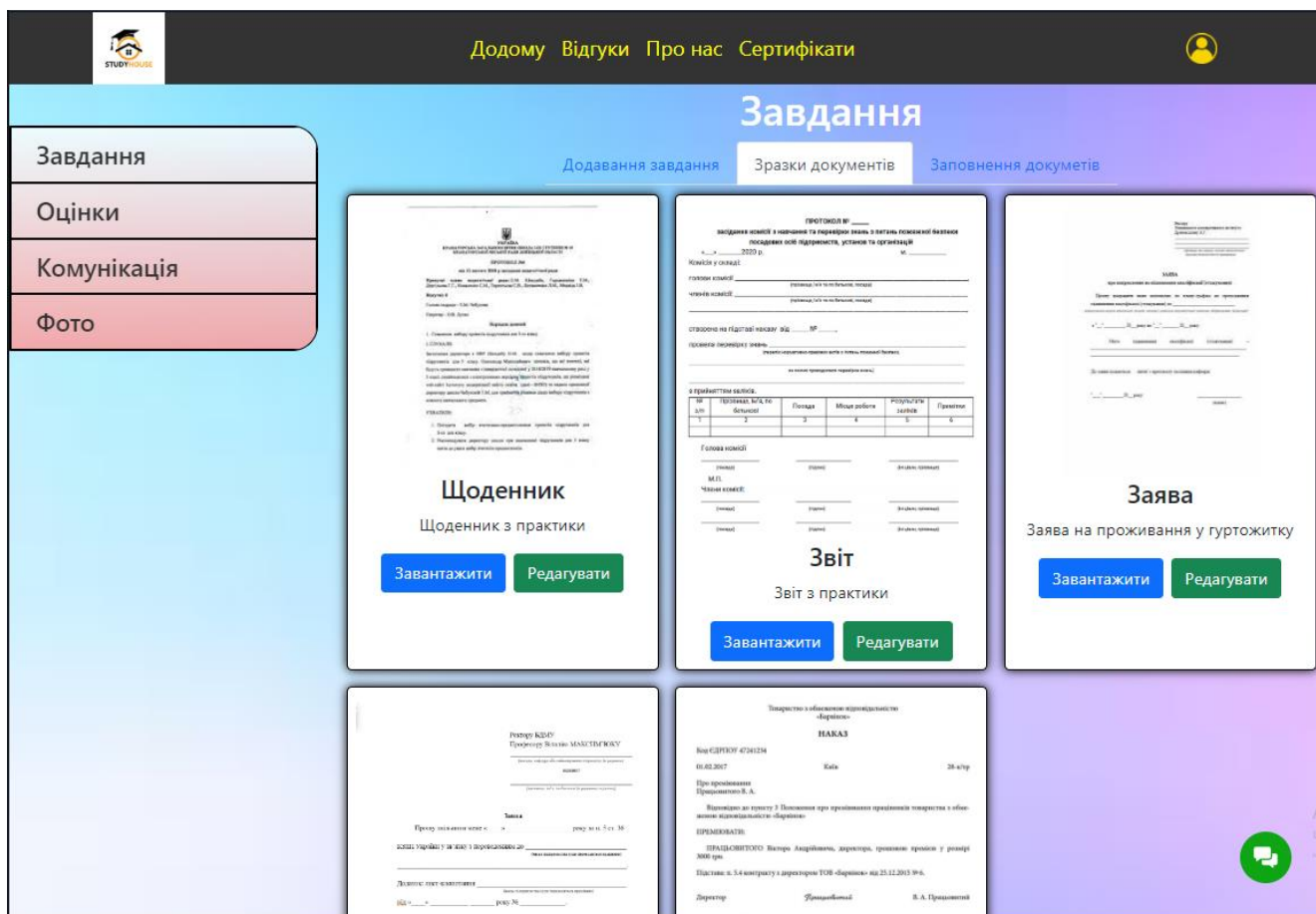


Рисунок 4.4 – Сторінка «Зразки документів»

А також можна заповнити документи за шаблоном, заповнивши відповідні поля необхідними даними (рис. 4.5).

Додому Відгуки Про нас Сертифікати

Завдання

Додавання завдання Зразки документів **Заповнення документів**

Тип документа **Оберіть тип документа**

Назва документа

Прізвище

Ім'я

По батькові

Посада

Дата

Внести данні

Зберегти

© 2023

Рисунок 4.5 – Сторінка «Заповнення документів»

На головній сторінці є можливість переглянути відгуки (рисунок 4.6).

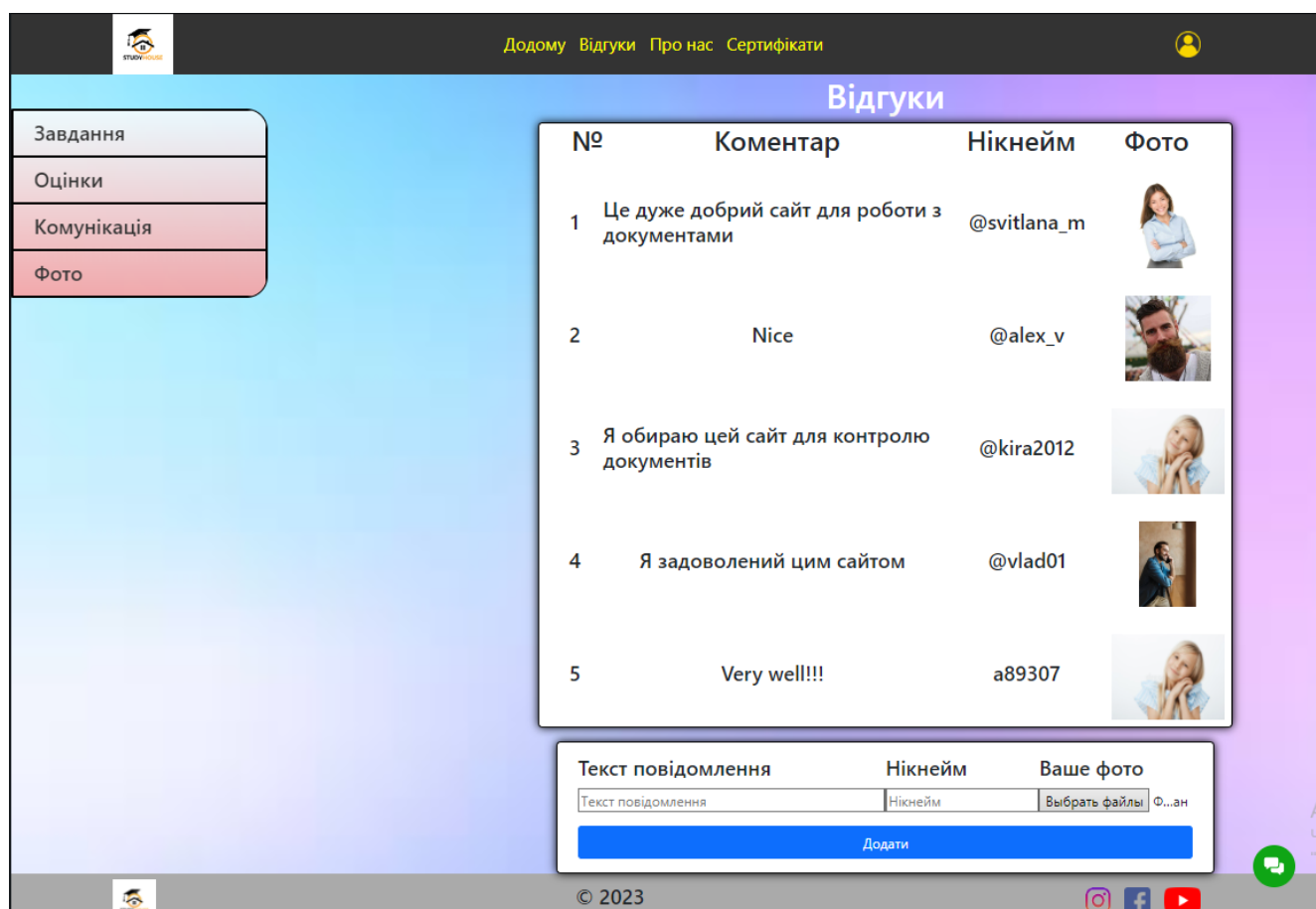


Рисунок 4.6 – Сторінка «Відгуки»

Також є сторінка з оцінками, де кожен студент може переглянути список своєї групи (рис. 4.7).

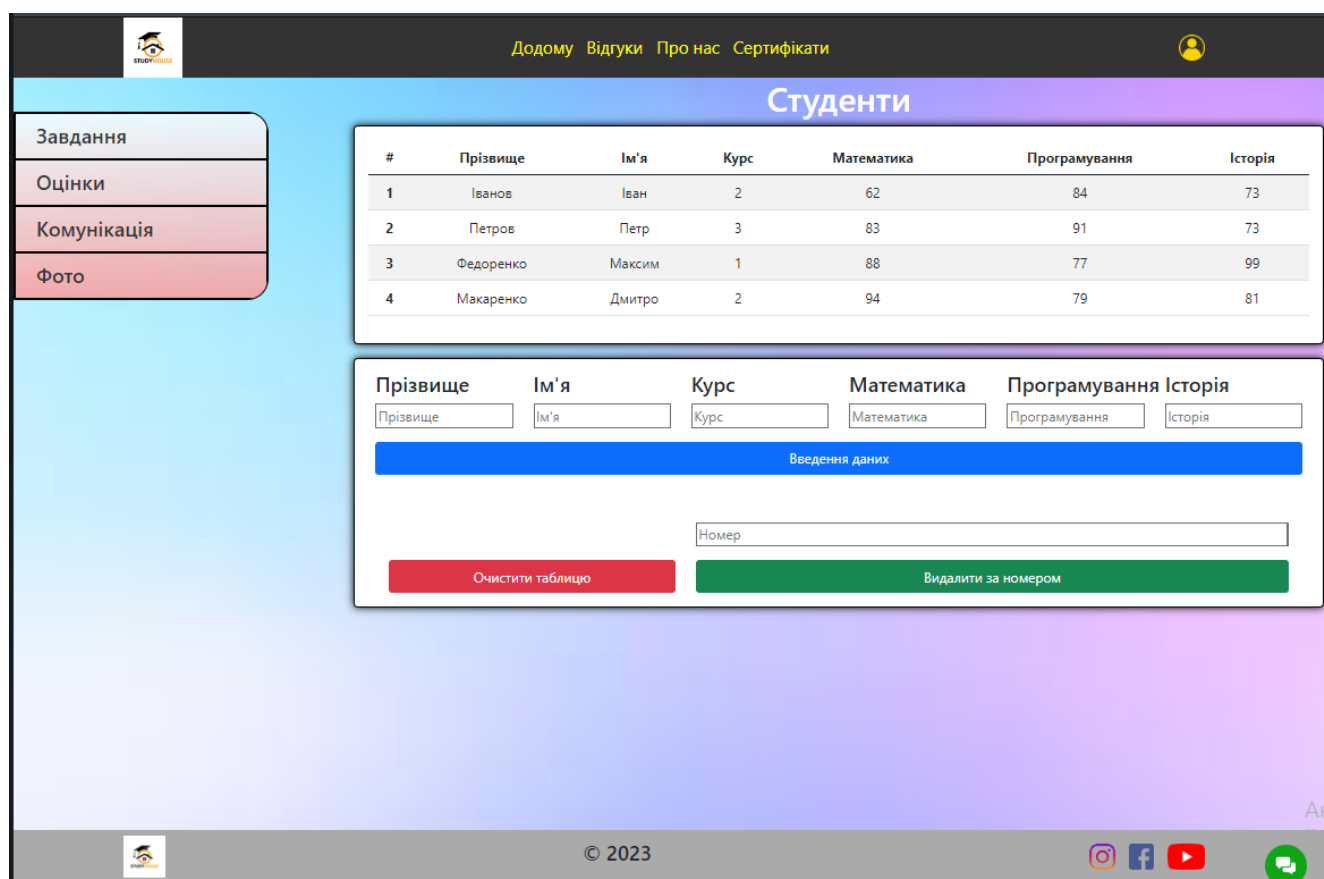


Рисунок 4.7 – Сторінка «Оцінки»

4.2 Тестування розробленої інформаційної системи

Під час тестування, усі виникаючі помилки одразу ж виправлялися. Нижче наведено декілька прикладів тестування роботи програми (табл. 4.1, 4.2).

Таблиця 4.1 - Тест на реєстрацію користувача

Ідентифікатор тест-варіанту	Користувач реєструється в системі
Набір вхідних даних	введення власних даних
Очікувані результати	Користувач успішно зареєструвався у системі
Виконувані дії	Користувач заповнює відповідні поля та натискає на кнопку Реєстрація

Результат тесту наведено на рис. 4.8.

Рисунок 4.8 – Результати тестування

Таблиця 4.2 - Тест на вхід у систему

Ідентифікатор тест-варіанту	Вхід в систему
Набір вхідних даних	Логін та пароль
Очікувані результати	Користувач входить у систему
Виконувані дії	Користувач вводить логін та пароль

Результат тесту наведено на рис. 4.9 - 4.10.

Рисунок 4.9 – Результати тестування

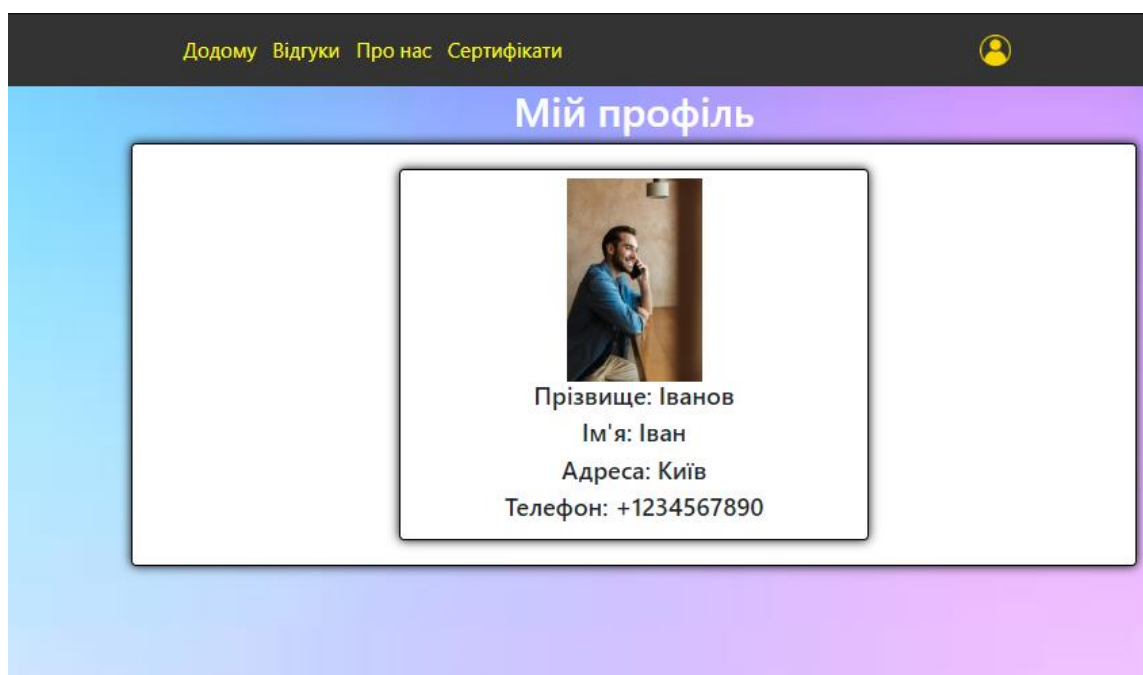


Рисунок 4.10 – Сторінка «Мій профіль»

Було проведено тест на додавання завдання (табл. 4.3).

Таблиця 4.3 - Тест на додавання завдання

Ідентифікатор тест-варіанту	Користувач додає завдання до сайту
Набір вхідних даних	ПБ, Дисципліна, Комунікація, Що зроблено
Очікувані результати	Користувач успішно додає завдання
Виконувані дії	Користувач заповнює відповідні поля, та натискає «Внести дані»

Результат тесту наведено на (рис. 4.11).

Додому Відгуки Про нас Сертифікати

Завдання

Додавання завдання Зразки документів Заповнення документів

#	ПІБ Викладача	№ курсу	Дисципліна	Комунікація	Що зроблено
1	Іванов Іван Іванович	5	Історія	Аудиторія	Отримано навчальні матеріали
2	Петров Петр Петрович	2	Програмування	Zoom	Прослухана лекція
3	Павел Сергійович	4	Математика	Онлайн	Отримана консультація

ПІБ № курсу Дисципліна Засоби комунікації Що зроблено

Внести завдання

Номер

Очистити таблицю Очистити за номером

Зберегти

Ак

© 2023

Рисунок 4.11 – Результати тестування

Тестування по всім наведеним тестам пройдено успішно.

Висновки до розділу

В четвертому розділі було проаналізовано структура вебзастосунку. Було розглянуто бази даних для вебсайту. В результаті було зроблено базу даних та вебсайт для організації робочого процесу студента.

ВИСНОВОК

В результаті дипломної роботи було вивчено процес розробки програми органайзер для студента.

Проаналізовано аналоги, розглянуто підхід до проєктування інтерфейсу вебресурсу, досліджено існуючі стандарти та показники якості інтерфейсів вебресурсів.

В результаті був розроблений спеціальний вебресурс «електронний органайзер», який містить необхідні інструменти для покращення планування та організації діяльності людини.

В ході даної роботи були досягнуті наступні результати:

- Наведено огляд існуючих програмних рішень у сфері мобільних застосунків, що показує актуальність створення нового програмного продукту, що включає повний набір засобів, необхідних для забезпечення безпеки даних шляхом розгортання сервісів на стороні клієнта.

- Для розробки сервісу було використано спеціальне програмне середовище та реалізовано інтерфейс, який взаємодіє з користувачем.

- Було розроблено архітектурне рішення для створення клієнтської та серверної частини програми. Проведено тестування застосунку. Результати тестування свідчать про належну функціональність програми та відповідність вимогам.

- Розроблений інтерфейс користувача має зручну навігацію та просту структуру. Детально описано, як користувач взаємодіє з застосунком, зокрема описано різні функціональні можливості та приклади взаємодії з системою.

- Створений сервіс дозволяє користувачам робити нотатки та анотації в календарі, редагувати наявні записи, позначати завдання як виконані або видаляти їх з бази даних.

У першому розділі було розглянуто предметну область баз даних. Також розглянуто інформаційну складову вебсайтів. Проаналізовано існуючі на ринку аналоги вебсайтів.

У другому розділі було розглянуто етапи проєктування та методологію вебсайтів. Також було розглянуто, які існують на даний час програмні засоби для проєктування баз даних та створення вебсайтів.

В третьому розділі було розглянуто засоби проєктування та створення вебсайтів. Спроектовано логічну модель БД. На основі логічної моделі було створено інформаційну систему.

Цей вебсайт був створений за допомогою HTML+CSS+JS, а також програмувальних мов PHP та MySQL бази даних. Вибір цих інструментів був обумовлений їх безкоштовністю та врахуванням різних інших критеріїв, що робить їх найоптимальнішим варіантом.

Вебсайт вийшов працездатний, та пройшов всі тестування і показав відмінний результат. Вебсайт може бути використаний на практиці. В подальшому, вебсайт можна доповнити різними методами та функціями.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Загальна інформація про базу даних MySQL // Електронна версія на сайті <https://ru.wikipedia.org/wiki/MySQL>
2. Робін Ніксон // Створення динамічних вебсайтів за допомогою PHP, MySQL, JavaScript, CSS та HTML5, 2016. 768с.
3. Веллінг, Томсон // Розробка веб-додатків за допомогою PHP та MySQL, 2017. 768 с.
4. Ромашов, В. CMS Drupal Система управління вмістом сайту/В. Ромашов. - Запоріжжя, 2020 254 с.
5. Гаєвський А.Ю. // 100% підручник зі створення веб-сторінок і вебсайтів: HTML і JavaScript - Одеса.: 2008. 457 с.
6. Стратегія та основні кроки при розробці вебсайту [Електронний ресурс]. – Режим доступу: <http://ruszura.in.ua/neobdidno-znaty/stratehiya-i-osnovni-kroky-rryrozo>.
7. Макфарланд Д. Велика книга CSS3: навчальний посібник / Д. Макфарланд. - Київ : 2014. 608 с. - ISBN 978-5-496-00428-2.
8. Горнаков С.Г. Освоюємо популярні системи керування сайтом / С.Г. Гірників. - Київ: ДМК Прес, 2009. - 336 с. - ISBN 978-5-94074-469-6.
9. «Evernote» [Електронний ресурс]. – Режим доступу: <https://evernote.com/intl/ru>.
10. Євдокимов Н.С. Створення сайтів/Н.С. Євдокимов. - Санкт-Петербург: Пітер, 2014. 410 с. - ISBN 978-5-496-00933-1.
11. «Any.do» [Електронний ресурс]. – Режим доступу: <https://www.any.do/>
12. Інформація про Any.do [Електронний ресурс]. – Режим доступу: <https://apix-drive.com/ua/anydo>
13. Найпопулярніші мови програмування [Електронний ресурс]. – Режим доступу: <https://highload.today/uk/top-10-najpopulyarnishih-mov-programuvannya/>.

14. Запит HTTP [Електронний ресурс]. – Режим доступу: <https://training.qatestlab.com/blog/technical-articles/http-protocol-what-and-where-to-test/>.
15. Кращі CMS платформи для запуску веб-сайту в 2020 році [Електронний ресурс]. – Режим доступу: <https://www.hostinger.com.ua/rukovodstva/luchshie-cms-platformy-2019/>.
16. Фрейн Бен. HTML5 та CSS3. Розробка сайтів для будь-яких браузерів та пристроїв / Бен Ф. – Київ : Пітер, 2017. 304 с. - ISBN 978-5-8459-1715-7.
17. Матросов А., Сергєєв А., Чаунін М., «HTML 4.0» / А. Матросов. - Київ: 2019. 304 с.
18. Мілецьку J.I. Принтери про Інтернет Маркетинг: New Tools i Methods for Web Develoers: Course Technologu Press / J.I. Мілецьку. – Boston, MA, United States, 2019. 480.

ДОДАТОК А
Текст програми


```

        <a      class="nav-link"      href="about.php">Про
нає</a>

    </li>

    <li class="hp nav-item">

        <a                                  class="nav-link"
href="certificate.php">Сертифікати</a>

    </li>

</ul>

</div>

</div>

</nav>

<ul class="d-flex m-3 p-0">
    <li  class="mx-2"><a      data-bs-toggle="modal"
data-bs-target="#exampleModal"></a></li>

</ul>

</div>

</div>

</div>

</header>

<?php include('modal.php')?>

```

A.2 (aside.php)

```

<div class="col-12 aside scrollmenu p-0">
    <ul class="px-0 d-flex flex-row flex-md-column m-0">
        <li                                class="py-2                                px-4"><a
href="documentation.php">Завдання</a></li>

        <li class="py-2 px-4"><a href="classinfo.php">Оцінки</a></li>

```

```

        <li class="py-2 px-4"><a href="communication.php">Комунікація</a></li>
        <li class="py-2 px-4"><a href="photo.php">Фото</a></li>
    </ul>
</div>

```

A.3 (footer.php)

```

<footer>
    <div class="container">
        <div class="row">
            <div class="col-12 d-flex justify-content-between align-items-center">
                

                <h4>© 2023</h4>

                <ul class="p-0 m-3 d-flex">
                    <li class="mx-2"><a href="https://www.instagram.com"></a></li>
                    <li class="mx-2"><a href="https://www.facebook.com"></a></li>
                    <li class="mx-2"><a href="https://www.youtube.com"><img alt="youtube"></a></li>
                </ul>
            </div>
        </div>
    </div>
</footer>

```

A.4 (database.php)

```
<?php

$link = mysqli_connect('localhost','root','root','zvit');
if (mysqli_connect_errno())
{
    echo 'Mistake ('.mysqli_connect_errno().'): '.mysqli_connect_error();
    exit();
}

?>
```

A.5 (index.php)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.0/dist/css/bootstrap.min.css"
rel="stylesheet"                                integrity="sha384-
KyZXEAg3QhqLMpG8r+8fhAXLRk2vvoC2f3B09zVXn8CA5QIVfZ0J3BCsw2P0p/
We" crossorigin="anonymous">
    <?php include('css.php') ?>
    <title>Home</title>
</head>
<body>
```

```

<?php include('header.php') ?>
<main>
    <div class="row m-0">
        <div class="col-12 col-md-4 p-0">
            <?php include('aside.php') ?>
        </div>
        <div class="main_section col-12 col-md-8">
            <div class="col-12 d-flex flex-column align-items-
center">

                <h1>Ласкаво просимо</h1>
                <div class="wrapper p-3 col-12">

                    <h4 class="text-center">Ласкаво просимо
на сайт</h4>

                    <p>Контроль знань і вмінь студентів має
бути комплексним і охоплювати всі розділи навчальної програми; він також має
бути системним, об'єктивним та індивідуальним. Під час контроль, проводиться
комплексна перевірка всієї навчальної діяльності учнів, у тому числі
сформованості спеціальних і загально навчальних умінь і навичок, творчих
здібностей, розвивається пізнавальний інтерес.</p>

                    <p>У навчальному процесі контроль
виконує ряд функцій:</p>

                    <ul>

                        <li>1) перевірна функція – вказує
на ступінь засвоєння навчального матеріалу навчальної дисципліни, засвоєння
знань, умінь і навичок;</li>

                        <li>2) дидактична функція -
забезпечує не тільки систематизацію, а й узагальнення всього навчального
матеріалу з дисципліни, що вивчається;</li>

```

3) корекційна функція – дозволяє виявити прогалини в знаннях, навичках і вміннях учнів з метою подальшого визначення оптимальних шляхів і шляхів їх виправлення;

4) методична функція – дає змогу вчителю оцінити використані методи та внести оперативні корективи в навчально-виховний процес;

5) виховна функція – формує в учнів відповідальність, здатність до самоорганізації та адекватної оцінки власної навчальної діяльності;

6) мотиваційна функція – стимулює активність у навчальній діяльності за умови, що учень орієнтується не стільки на отримання оцінки, скільки на самоосвіту.

</div>

</div>

</div>

</div>

</main>

<div class="chatbot__btn">

<div class="chatbot__tooltip d-none">Є питання?</div>

</div>

<?php include('footer.php') ?>

<?php include('script.php') ?>

</body>

</html>

A.6 (style.css)

```

.main_section{
    min-height: 84vh;
}
header{
    background: #333;
}

li{
    list-style-type: none;
}

.aside{
    z-index: 100;
    position: sticky;
    top:110px;
    background-image:linear-gradient(rgba(255, 255, 255, 0.8),rgba(255,151,
151, 0.8) );
    max-width: 300px;
    border: 1px solid black;
    max-height: 80vh;
    border-radius: 0 25px 25px 0;
}
@media(max-width: 768px){
    .aside{
        max-width: 100vh;
    }
}

.aside a{
    text-decoration: none;
    font-weight: 600;
}

```

```
        font-size: 24px;
        color: #333;
    }
    .aside li{
        border: 1px solid black;
    }
    .scrollmenu {
        overflow: auto;
        white-space: nowrap;
    }
    .scrollmenu::-webkit-scrollbar {
        height: 0px;
        width: 0;
    }
    .scrollmenu {
        -ms-overflow-style: none;
    }
    .scrollmenu {
        overflow: -moz-scrollbars-none;
    }
    .wrapper{
        border: 2px solid black;
        border-radius: 5px;
        box-shadow: 0 0 10px black;
        height: 100%;
        background-color: #fff;
    }
    .wrapper:hover{
        background-color: #aaa;
        transition-duration:1s;
```

```
}
```

```
img[alt="comp"]{  
    height: 200px;  
}
```

```
footer{  
    background-color: #aaa;  
}
```

```
.hp a{  
    color: yellow !important;  
    font-size: 20px;  
    text-decoration: none;  
}
```

```
.navbar{  
    background: transparent!important;  
}
```

```
.slider{  
    background: #FFFFFFF;  
    box-shadow: 0px 0px 50px rgba(64, 74, 84, 0.06);  
    border-radius: 20px;  
}
```

```
.circle{  
    border: 0px!important;  
    border-radius: 7px;  
    width: 14px!important;  
    height: 14px!important;
```

```

        background: #E0E0E0!important;
    }
    .circle.active{
        background: #CFB88C!important;
    }
    #arrow_left2{
        display: none;
    }
    .carousel-control-prev:hover #arrow_left{
        display: none;
    }
    .carousel-control-prev:hover #arrow_left2{
        display: flex;
    }
    #arrow_right2{
        display: none;
    }
    .carousel-control-next:hover #arrow_right{
        display: none;
    }
    .carousel-control-next:hover #arrow_right2{
        display: flex;
    }
    main{
        background-image:url(../img/BG.jpg);
        background-repeat: no-repeat;
        background-size: cover;
        min-height: 84vh;
    }
    .menu a{

```

```

    text-decoration: none;
    color: orange;
    font-size: 24px;
}
.wrapper2, .wrapper3{
    border: 2px solid black;
    border-radius: 5px;
    box-shadow: 0 0 10px black;

    background-color: #fff;
}
.wrapper2 img{
    height: 100px;
}
#block-sort{
    display: none;
}
.wrapper3 img{
    height: 500px;
}
h1{
    color: white;
}

```

A.7 (script.js)

```

(function (factory) {
  if (typeof define === 'function' && define.amd) {
    // AMD (Register as an anonymous module)
    define(['jquery'], factory);
  } else if (typeof exports === 'object') {
    // Node/CommonJS

```

```

    module.exports = factory(require('jquery'));
  } else {
    // Browser globals
    factory(jQuery);
  }
})(function ($) {

var pluses = /\+/g;

function encode(s) {
  return config.raw ? s : encodeURIComponent(s);
}

function decode(s) {
  return config.raw ? s : decodeURIComponent(s);
}

function stringifyCookieValue(value) {
  return encode(config.json ? JSON.stringify(value) : String(value));
}

function parseCookieValue(s) {
  if (s.indexOf('"') === 0) {
    // This is a quoted cookie as according to RFC2068, unescape...
    s = s.slice(1, -1).replace(/\\"/g, '"').replace(/\\'/g, "'");
  }

  try {
    // Replace server-side written pluses with spaces.
    // If we can't decode the cookie, ignore it, it's unusable.
    // If we can't parse the cookie, ignore it, it's unusable.
    s = decodeURIComponent(s.replace(pluses, ' '));
    return config.json ? JSON.parse(s) : s;
  } catch(e) {}
}

function read(s, converter) {
  var value = config.raw ? s : parseCookieValue(s);
  return $.isFunction(converter) ? converter(value) : value;
}

var config = $.cookie = function (key, value, options) {

  // Write

```

```

if (arguments.length > 1 && !$.isFunction(value)) {
    options = $.extend({}, config.defaults, options);

    if (typeof options.expires === 'number') {
        var days = options.expires, t = options.expires = new Date();
        t.setMilliseconds(t.getMilliseconds() + days * 864e+5);
    }

    return (document.cookie = [
        encode(key), '=', stringifyCookieValue(value),
        options.expires ? '; expires=' + options.expires.toUTCString() :
        "", // use expires attribute, max-age is not supported by IE
        options.path ? '; path=' + options.path : "",
        options.domain ? '; domain=' + options.domain : "",
        options.secure ? '; secure' : ""
    ].join(""));
}

// Read

var result = key ? undefined : {},
    // To prevent the for loop in the first place assign an empty array
    // in case there are no cookies at all. Also prevents odd result when
    // calling $.cookie().
    cookies = document.cookie ? document.cookie.split('; ') : [],
    i = 0,
    l = cookies.length;

for (; i < l; i++) {
    var parts = cookies[i].split('='),
        name = decode(parts.shift()),
        cookie = parts.join('=');

    if (key === name) {
        // If second argument (value) is a function it's a converter...
        result = read(cookie, value);
        break;
    }

    // Prevent storing a cookie that we couldn't decode.
    if (!key && (cookie = read(cookie)) !== undefined) {
        result[name] = cookie;
    }
}

```

```

        return result;
    };

    config.defaults = {};

    $.removeCookie = function (key, options) {
        // Must not alter options, thus extending a fresh object...
        $.cookie(key, "", $.extend({}, options, { expires: -1 }));
        return !$.cookie(key);
    };

    }));

    function cookie(){
        var name = $("#login").val();
        var description = $("#pass").val();
        $.cookie('login',name);
        $.cookie('pass',description);
    }

    function createCookie(name, value, days) {
        var expires;
        if (days) {
            var date = new Date();
            date.setTime(date.getTime() + (days * 24 * 60 * 60 * 1000));
            expires = "; expires=" + date.toGMTString();
        }
        else {
            expires = "";
        }
        document.cookie = name + "=" + value + expires + "; path=/";
    }

    function getCookie(c_name) {
        if (document.cookie.length > 0) {
            c_start = document.cookie.indexOf(c_name + "=");
            if (c_start != -1) {
                c_start = c_start + c_name.length + 1;
                c_end = document.cookie.indexOf(";", c_start);
                if (c_end == -1) {
                    c_end = document.cookie.length;
                }
            }
            return unescape(document.cookie.substring(c_start, c_end));
        }
    }

```

```

    }
    return "";
}

function exituser(){
    createCookie('user',"",-1);
}
function mySortUp(){
    var nav=document.querySelector('.product_list');
    for (var i = 0; i < nav.children.length-1; ++i) {
        for (var j = i+1; j < nav.children.length; ++j) {
            if (+nav.children[i].getAttribute('data-
sort')>+nav.children[j].getAttribute('data-sort')){
                replaceNode      =      nav.replaceChild(nav.children[j],
nav.children[i]);
                insertAfter(replaceNode, nav.children[i]);
            }
        }
    }
}
function mySortDown(){
    var nav=document.querySelector('.product_list');
    for (var i = 0; i < nav.children.length-1; ++i) {
        for (var j = i+1; j < nav.children.length; ++j) {
            if (+nav.children[i].getAttribute('data-
sort')<+nav.children[j].getAttribute('data-sort')){
                replaceNode      =      nav.replaceChild(nav.children[j],
nav.children[i]);
                insertAfter(replaceNode, nav.children[i]);
            }
        }
    }
}

function insertAfter(elem, refElem){
    return refElem.parentNode.insertBefore(elem,refElem.nextSibling);
}

var show=0;
function block_show(){
    var ss=document.getElementById('block-sort');
    if (show==0)
    {
        show=1;
    }
}

```

```

        ss.style.display='grid';
    }else{
        show=0;
        ss.style.display='none';
    }
}
function mySortPrice(){
    var nav=document.querySelector('.product_list');
    var min=+document.querySelector('#minprice').getAttribute('value');
    var max=+document.querySelector('#maxprice').getAttribute('value');

    for (var i = 0; i < nav.children.length; ++i) {
        if
            (+nav.children[i].getAttribute('data-sort')<min
+nav.children[i].getAttribute('data-sort')>max){
            nav.children[i].style.display='none';
        }
        else
            nav.children[i].style.display='flex';
    }
}
function inputMaxVal(a){
    $('#maxprice').attr('value',a.value);
}
function inputMinVal(a){
    $('#minprice').attr('value',a.value);
}

```

||

ДОДАТОК Б
Слайди презентації

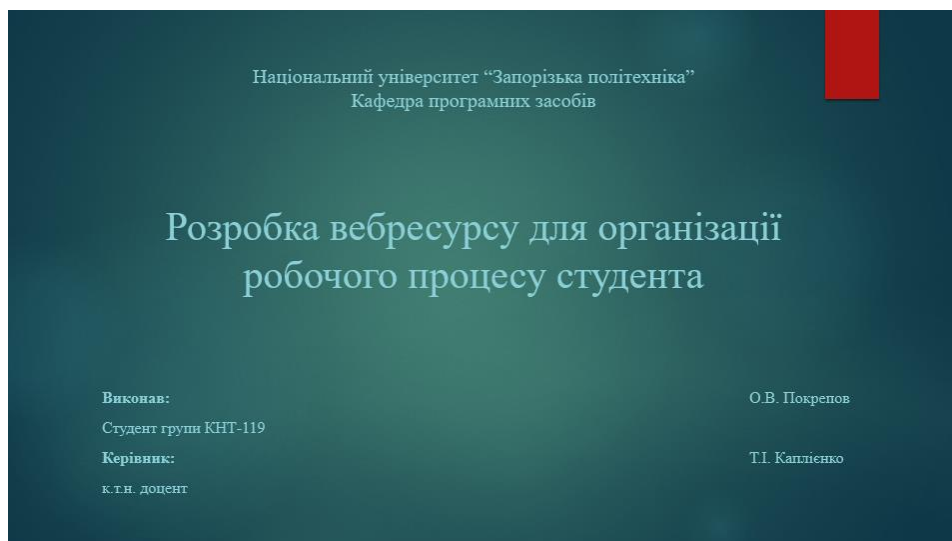


Рисунок Б.1 – Слайд №1

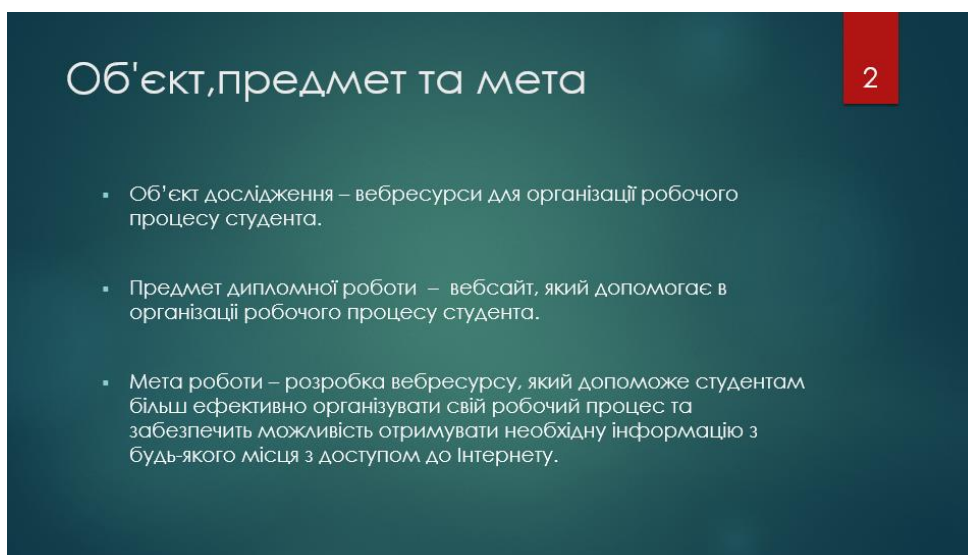


Рисунок Б.2 – Слайд №2

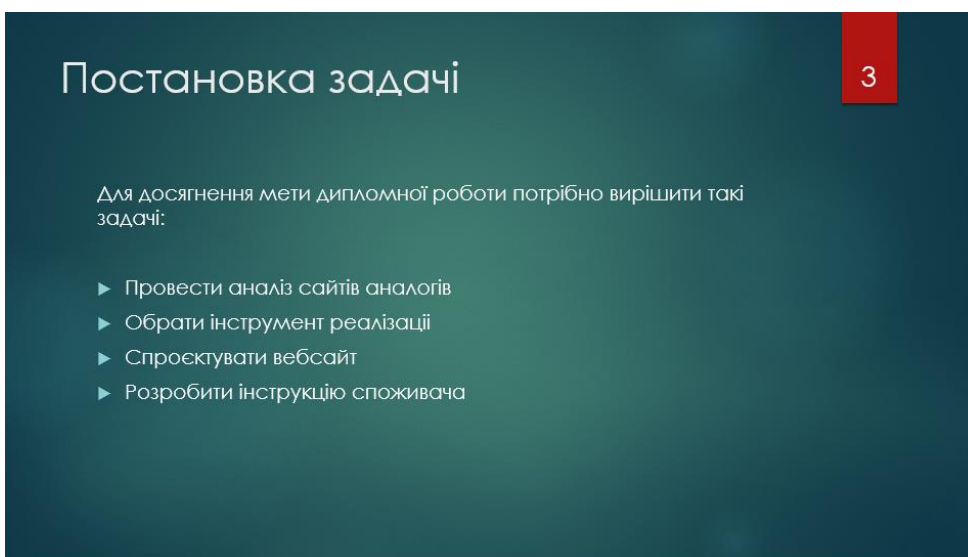


Рисунок Б.3 – Слайд №3

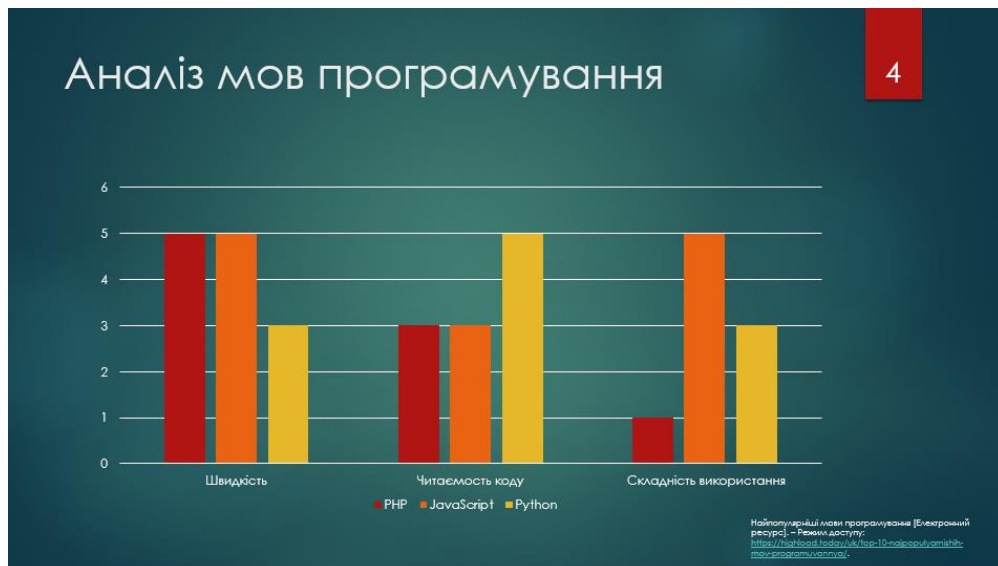


Рисунок Б.4 – Слайд №4



Рисунок Б.5 – Слайд №5

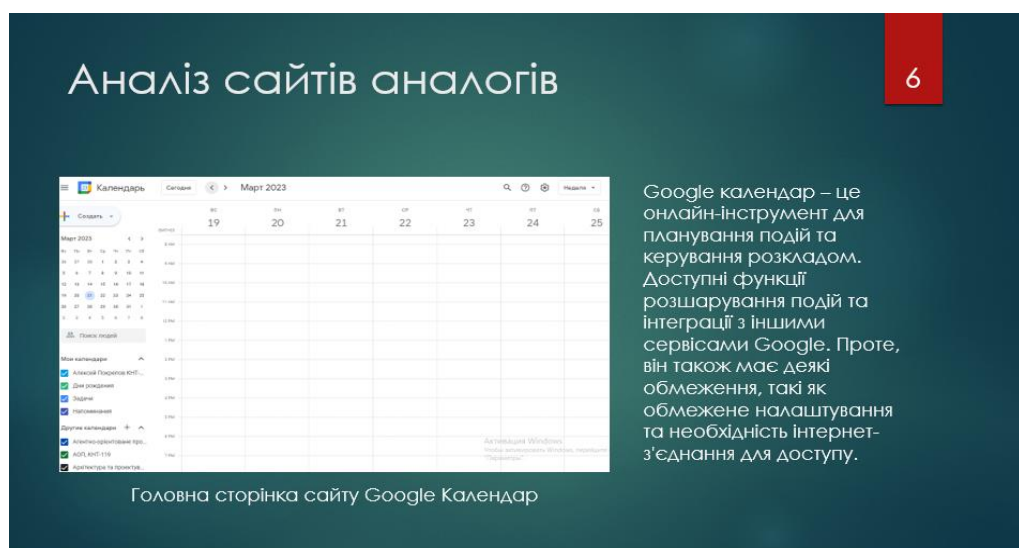


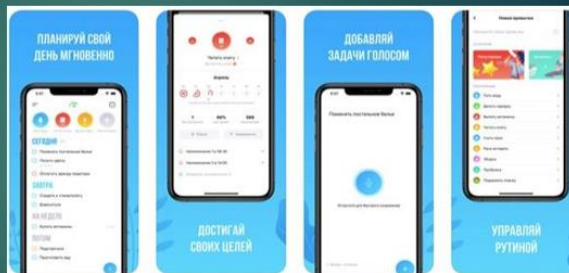
Рисунок Б.6 – Слайд №6

Аналіз сайтів аналогів

7

Tappsk - організатор і планувальник задач з зручним інтерфейсом та гнучкими функціями планування. Проте він має ряд мінусів:

- обмежені функціональні можливості
- обмежена інтеграція
- обмежена масштабованість



Інтерфейс Tappsk

Рисунок Б.7 – Слайд №7

Діаграма обробки даних

8



Зліва представлено загальне уявлення функціонального призначення програми.

Рисунок Б.8 – Слайд №8

Структура БД

9

+ Параметры				
	id	textotziv	username	fotourl
☐ Изменить ☐ Копировать ☐ Удалить	1	Це дуже добрий сайт для роботи з документами	@svitlana_m	src="img/p2.jpg"
☐ Изменить ☐ Копировать ☐ Удалить	2	Nice	@alex_v	src="img/p1.jpg"
☐ Изменить ☐ Копировать ☐ Удалить	3	Я обираю цей сайт для контролю документів	@kira2012	src="img/p3.jpg"
☐ Изменить ☐ Копировать ☐ Удалить	4	Я задоволений цим сайтом	@vlad01	src="img/p4.jpg"
☐ Изменить ☐ Копировать ☐ Удалить	5	Very well!!!	a89307	src="img/p3.jpg"

Дані БД (Відуки)

Рисунок Б.9 – Слайд №9

Приклад таблиці БД

10

№	Атрибут	Семантика	Тип
1	id	Унікальний ідентифікатор студента	Int(5)
2	textotziv	Текст відгуку	varchar(500)
3	username	Ім'я користувача	varchar(50)
4	fotourl	Фото користувача	varchar(50)

Таблиця «Відгуки»

Рисунок Б.10 – Слайд №10

Розроблений сайт

11

Сторінка «Завдання»

Додати завдання | Завдання | Завдання | Завдання

#	ПІБ Викладача	№ курсу	Дисципліна	Категорія	Що зроблено
1	Іванов Іван Іванович	5	Дисципліна роботи	Увім	Підготував роботу
2	Петров Петро Петрович	2	Курсова робота	Теоретич	Провів лекцію

ПІБ: № курсу: Дисципліна: Категорія: Що зроблено:

© 2023

Рисунок Б.11 – Слайд №11

Висновки

12

- ▶ В результаті дипломної роботи було вивчено процес розробки програми органайзер для студента.
- ▶ Проаналізовано аналоги, розглянуто підхід до проектування інтерфейсу вебресурсу, досліджено існуючі стандарти та показники якості інтерфейсів вебресурсів.
- ▶ В результаті був розроблений спеціальний вебресурс «електронний органайзер», який містить необхідні інструменти для покращення планування та організації діяльності людини.

Рисунок Б.12 – Слайд №12