

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

бакалавр

(ступінь вищої освіти)

на тему РОЗРОБКА ВЕБЗАСТОСУНКУ ОНЛАЙН ЩОДЕННИКА ЗІ
СПИСКОМ ЗАДАЧ

DEVELOPMENT OF A WEB APPLICATION FOR AN ONLINE DIARY
WITH A TO-DO LIST

Виконав(ла): студент(ка) 4 курсу, групи КНТ-139

Спеціальності 121 Інженерія програмного

(код і найменування спеціальності)

забезпечення

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

КАЛЬНИЧЕНКО А.Р.

(ПРІЗВИЩЕ та ініціали)

Керівник КОЛПАКОВА Т.О.

(ПРІЗВИЩЕ та ініціали)

Рецензент КОРОТКИЙ 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 ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

КАЛЬНИЧЕНКО Ангеліни Русланівни

(ПРІЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Розробка вебзастосунку Онлайн щоденника зі списком задач. Development of a Web Application for an Online Dairy with To-Do List

керівник проєкту (роботи) к.т.н., доцент, КОЛПАКОВА Тетяна Олексіївна,
(науковий ступінь, вчене звання, ПРІЗВИЩЕ, ім'я, по батькові)

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

2. Строк подання студентом проєкту (роботи) 01 червня 2023 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. 2. Аналіз та вибір засобів розробки програмного продукту. 3. Основні рішення щодо реалізації системи. 4. Керівництво програміста. 5. Керівництво користувача

5. Перелік графічного матеріалу (з з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)
Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-5 Основна частина	КОЛПАКОВА Т.О., доцент		
Нормоконтроль	ЛИПОВЕЦЬ М.В., асистент		

7. Дата видачі завдання « 04 » квітня 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, 5
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) Ангеліна КАЛЬНИЧЕНКО
(Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

(підпис) Тетяна КОЛПАКОВА
(Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 125 с., 1 табл., 35 рис., 3 дод., 19 джерел.

ОНЛАЙН ЩОДЕННИК ЗІ СПИСКОМ ЗАДАЧ, ВЕБЗАСТОСУНОК, ЗАПИС, ЗАДАЧА, OPEN SERVER, MYSQL, PHPSTORM.

Об'єкт дослідження – процес зберігання та систематизації записів і задач.

Предмет дослідження – вебзастосунок для ведення списку задач.

Мета роботи – розробити онлайн щоденник з можливістю створення, редагування та видалення записів та задач.

Матеріали, методи та технічні засоби: серверне середовище Open Server, інтегроване середовище розробки (IDE) PhpStorm, мови програмування HTML, CSS, JavaScript та PHP, PhpMyAdmin для адміністрування серверів баз даних MySQL, персональний комп'ютер з процесором Intel Core i5 під управлінням операційної системи Microsoft Windows 10.

Результати. Розроблений вебзастосунок онлайн щоденник зі списком задач, який дає можливість створення, редагування та видалення записів та задач.

Висновки. Результатом розробки щоденника зі списком задач є функціональний продукт, який забезпечує користувачів можливістю вести свій щоденник, створення списку задач та їх виконанням.

Галузь використання. Розроблений щоденник може бути використаний у різних сферах життя, де необхідно вести щоденник та планувати свої завдання. Він знадобиться в освітній сфері, наприклад студентам, також в сфері бізнесу, працівникам, яким потрібно вести записи про зустрічі, дзвінки. Також буде в нагоді в особистій сфері для зберігання своїх думок, планів та нагадувань.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 125 pages, 1 tables, 35 figures, 3 appendixes, 19 sources.

ONLINE DIARY WITH TASK LIST, WEB APPLICATION, ENTRY, TASK, OPEN SERVER, MYSQL, PHPSTORM.

Object of study is the process of storing and organizing records and tasks.

Subject of study a web application for managing a to-do list).

The purpose of the work is to develop an online diary with the ability to create, edit, and delete entries and tasks.

Materials, methods, and tools: Open Server as the server environment, PhpStorm as the integrated development environment (IDE), programming languages HTML, CSS, JavaScript, and PHP, personal computer with an Intel Core i5 processor running the Microsoft Windows 10 operating system.

Results. A developed web application, an online diary with a to-do list, that enables users to create, edit, and delete entries and tasks.

Conclusions. The development of the diary with a to-do list has resulted in a functional product that allows users to maintain their diary, create task lists, and track their completion.

Field of Application: The developed diary can be used in various domains where diary keeping and to-do planning are necessary. It can be beneficial in the educational sector, such as for students, as well as in the business sector for recording meetings and calls. It can also be valuable in personal life for storing thoughts, plans, and reminders.

ЗМІСТ

	С.
Перелік скорочень та умовних познач.....	8
Вступ.....	9
1 Аналіз предметної області.....	10
1.1 Онлайн нотатник та щоденник.....	10
1.2 Вимоги до створення інтерфейсу онлайн щоденника.....	11
1.3 Аналіз існуючих онлайн щоденників.....	13
1.3.1 Trello.....	13
1.3.2 Todoist.....	15
1.3.3 Google Keep.....	15
1.3.4 Asana.....	16
1.3.5 Microsoft To Do.....	17
1.4 Постановка завдання.....	18
1.5 Висновки за розділом 1.....	19
2 Аналіз та вибір засобів розробки програмного продукту.....	20
2.1 Аналіз видів веброзробки та мов програмування.....	20
2.1.1 Види веброзробки.....	20
2.1.2 Мови програмування.....	21
2.2 Зберігання даних.....	25
2.2.1 Open Server.....	26
2.2.2 PhpMyAdmin.....	26
2.3 Вибір інструментів розробки.....	28
2.4 Висновки за розділом 2.....	29
3 Основні рішення щодо реалізації системи.....	31
3.1 Розробка схеми отримання даних.....	31
3.2 Розробка вебінтерфейсу.....	33
3.3 Опис реалізованих функцій.....	35
3.3.1 Вхід та реєстрація користувача.....	35
3.3.2 Робота з цілями у списку задач.....	38

3.3.3 Робота з текстом в щоденнику	43
3.4 Висновки за розділом 3	47
4 Керівництво програміста	48
4.1 Призначення програми.....	48
4.2 Характеристики програми	48
4.3 Умови застосування програми.....	48
4.4 Звернення до програми	49
4.5 Вхідні та вихідні дані	49
4.5.1 Вхідні дані.....	49
4.5.2 Вихідні дані.....	49
4.6 Розгортання вебзастосунку на Open Server.....	50
4.7 Висновки за розділом 4	51
5 Керівництво користувача	52
5.1 Призначення та умови застосування програми	52
5.1.1 Призначення програми.....	52
5.1.2 Умови виконання програми	52
5.2 Характеристики програми	52
5.3 Структура програми	53
5.3.1 Сторінка авторизації.....	53
5.3.2 Сторінка «список задач»	56
5.3.3 Сторінка «щоденник».....	58
5.3.4 Сторінка «профіль»	60
5.4 Висновки за розділом 5	61
Висновки.....	62
Перелік джерел посилань.....	63
Додаток А Технічне завдання.....	65
Додаток Б Текст програми.....	70
Додаток В Слайди презентації	120

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

API	– Application Programming Interface;
CSV	– Comma-Separated Values;
CSS	– Cascading Style Sheets,
DOM	– Document Object Model;
HTML	– HyperText Markup Language;
IDE	– Integrated Development Environment;
PDF	– Portable Document Format;
PHP	– Hypertext Preprocessor;
QBE	– Query-by-example;
SQL	– Structured Query Language;
SPA	– Single Page Application;
XML	– eXtensible Markup Language.

ВСТУП

У сучасному світі, де технології розвиваються у швидкому темпі та люди зайняті вирішенням багатьох задач, ефективне управління часом та задачами є дуже важливим. Щоб впоратися з робочими справами, зустрічами, дзвінками та домашньою рутиною було винайдено тайм-менеджмент – планування свого дня з правильним розподілом часу.

На допомогу приходять щоденники. Паперові зошити, в яких люди складала списки задач вручну, викликали ряд незручностей. Наприклад, планер потрібно завжди носити з собою, можна було його загубити, конфіденційність також була під загрозою. Він міг бути пошкодженим і частина записів була б неможливою для читання, а редагування переліку завдань викликала ряд незручностей. Якщо немає з собою щоденника, то немає і календаря з плануванням.

З появою онлайн планувальників стало набагато простіше створювати та коригувати свої плани, зберігати багато записів та доступ до даних став доволі швидким. Також щоденник захищен від чужих очей завдяки паролям.

Задачі, які було виконано в рамках даної дипломної кваліфікаційної роботи:

- проведено аналіз існуючого програмного забезпечення;
- проведено аналіз та вибір засобів розробки програмного продукту;
- розроблено та реалізовано онлайн щоденник зі списком задач з можливістю створення, редагування та видалення записів та завдань;
- керівництво програміста під час розробки проєкту;
- керівництво користувача щодо використання розробленої системи.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

У нинішній час, коли інформаційні технології відіграють значну роль у житті кожної людини, онлайн щоденник стає дедалі популярним і займає важливе місце в нашому житті. Онлайн щоденник – це доступний і зручний інструмент для планування задач та керування часом.

У цьому розділі буде розглянуто особливості та переваги онлайн щоденників, як інструмент для збереження та організації персональної інформації.

1.1 Онлайн нотатник та щоденник

В сучасному світі, коли віртуальний простір став незамінною частиною нашого життя, з'явилась можливість вести записи з будь-яких сфер життя електронно. З цією метою було створено велику кількість різноманітних застосунків для нотування та зберігання інформації, такі як онлайн нотатники та щоденники. Хоч і на перший погляд ці дві програми можуть здаватись подібними, вони мають ряд відмінностей.

Щоденник – це особистий журнал, у якому людина веде запис свого життя. Це може бути щоденник для запису подій та емоцій, або професійний щоденник для запису задач та завдань [1]. Тепер, щоденник більше схожий на програму, яка дозволяє користувачам вести облік щоденних подій, створювати та зберігати нотатки та записи у хмарному сховищі. Як правило, щоденники мають простий інтерфейс, спрощені функції редагування та підтримують різні формати зображень.

Нотатник – це звичайний блокнот, у якому можна вести записи про різні справи, такі як замітки, список покупок, плани та інше. Нотатники можуть мати різну форму та розмір, від звичайного блокнота до електронного ноутбука, на якому можна вести записи в електронному форматі. Зараз, електронний нотатник представляє з себе програму, яка

дозволяє користувачам створювати та зберігати різноманітні типи нотаток, включаючи тексти, зображення, звуки, відео, посилання та інше. Нотатники зазвичай мають більш розширені функції редагування, організації та пошуку, ніж щоденники.

В таблиці 1.1 показаний розширений список відмінностей між нотатником та щоденником [2].

Таблиця 1.1 – Схожість та відмінність між щоденником та нотатником

Критерії	Щоденник	Нотатник
Тип документів	Щоденні записи, події, завдання	Записи, списки, нагадування
Функціонал	Створення, редагування, збереження, видалення щоденних записів, управління списком завдань	Створення, редагування, збереження, видалення записів
Форматування	Великий список можливостей форматування тексту	Обмежені можливості форматування тексту
Призначення	Ведення щоденника, планування, відстеження завдань	Зберігання та організація інформації
Онлайн/офлайн	Можливе використання як в онлайн-режимі, так і в офлайн	
Хмарні можливості	Можливість синхронізувати дані на різних пристроях, доступ до даних з будь-якого місця	
Рівень приватності	Можливість захисту записів паролем	

1.2 Вимоги до створення інтерфейсу онлайн щоденника

Створення інтерфейсу є важливим аспектом у створенні сайту. З появою онлайн щоденників, користувачі стають все більш вимогливими щодо функціональності та зручності використання таких систем. Щоб

задовольнити бажання користувачів, інтерфейс онлайн щоденника має включати наступні компоненти:

- головне меню – містить кнопки для доступу до основних функцій програми, таких як створення нового завдання, перегляд списку завдань та редагування налаштувань тощо;

- список завдань – містить всі списки завдань, що були створені користувачем, з можливістю перегляду, редагування та відмітки виконання завдань;

- фільтри – дозволяють відсортовувати задачі за різними параметрами, наприклад такими як дата створення тощо;

- календар – дозволяє побачити планування завдань на певний період часу та розпланувати їх відповідно до термінів виконання.

Інтерфейс повинен бути простим, лаконічним, зручним і зрозумілим для користувача, і не завантажувати його зайвими елементами. Є основний список вимог для інтерфейсу щоденника, який бажано дотримуватись:

- простота та зрозумілість інтерфейсу: інтерфейс повинен бути простим та зрозумілим для користувача. Користувач повинен з легкістю знайти потрібну функцію та зрозуміти її призначення [3]-[4];

- мінімалістичний дизайн: оскільки основна мета щоденника – збереження та організація інформації, то інтерфейс повинен бути мінімалістичним та не заважати користувачу. Занадто яскраві кольори та багато деталей можуть відволікти від основної мети та заважати при використанні додатку [3]-[4];

- респонсивний дизайн: оскільки онлайн щоденник повинен бути доступним з будь-якого пристрою з Інтернетом, інтерфейс повинен бути респонсивним та адаптивним до різних розмірів екранів [3];

- легкість використання: інтерфейс повинен бути простим та зрозумілим для користувача. Користувач повинен з легкістю знайти потрібну функцію та зрозуміти її призначення [3];

- доступність: інтерфейс повинен бути доступним для користувачів з різними особливостями – візуальними, моторними, аудіотехнічними, та ін [3];
- естетика: зважаючи на те, що користувачі зазвичай звертають увагу не тільки на функціональність, а й на зовнішній вигляд програми, естетичність і зручність її інтерфейсу важливі для успішної реалізації проєкту [3];
- ергономіка: елементи управління повинні бути розташовані зручно і логічно. Використання гарячих клавіш повинно бути мінімальним [3];
- навігація: вона повинна бути зрозумілою та легкою для користувача. Користувач повинен мати можливість швидко перейти до будь-якої частини програми [3]-[4];
- інформаційний вміст: вміст має бути доступним та зрозумілим. Інформація повинна бути викладена лаконічно та зрозуміло. Має бути відсутня непотрібна інформація, яка може забрати увагу користувача від головної мети програми [4].

1.3 Аналіз існуючих онлайн щоденників

Наразі існує багато онлайн щоденників, які допомагають людям організувати свій час та вирішити завдання. Огляд існуючих онлайн щоденників зі списками задач був проведений на основі аналізу функціональності, інтерфейсу та користувацького досвіду.

1.3.1 Trello

Переваги:

- надзвичайно легкий та зручний інтерфейс;
- безліч додаткових функцій, таких як коментарі, прикріплення файлів тощо;

- можливість спільної роботи з іншими користувачами.

Недоліки:

- для повноцінного використання необхідно зрозуміти принцип роботи Канбан-дошки (рис. 1.1), що може зайняти час [5]-[6].

Вигляд головного вікна Trello наведено на рис. 1.2.

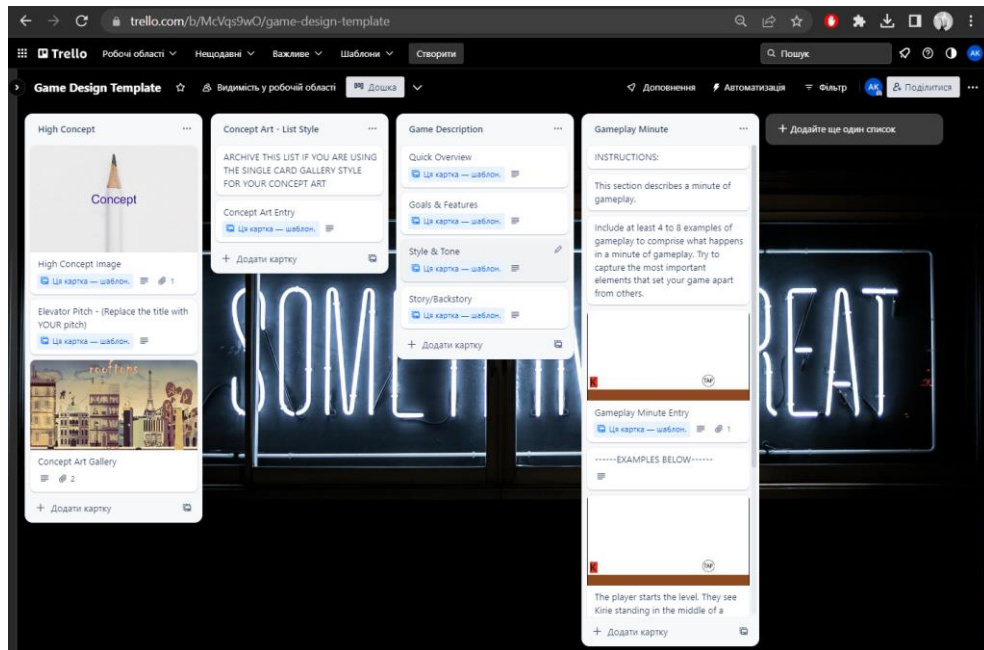


Рисунок 1.1 – Приклад Канбан-дошки в Trello [5]

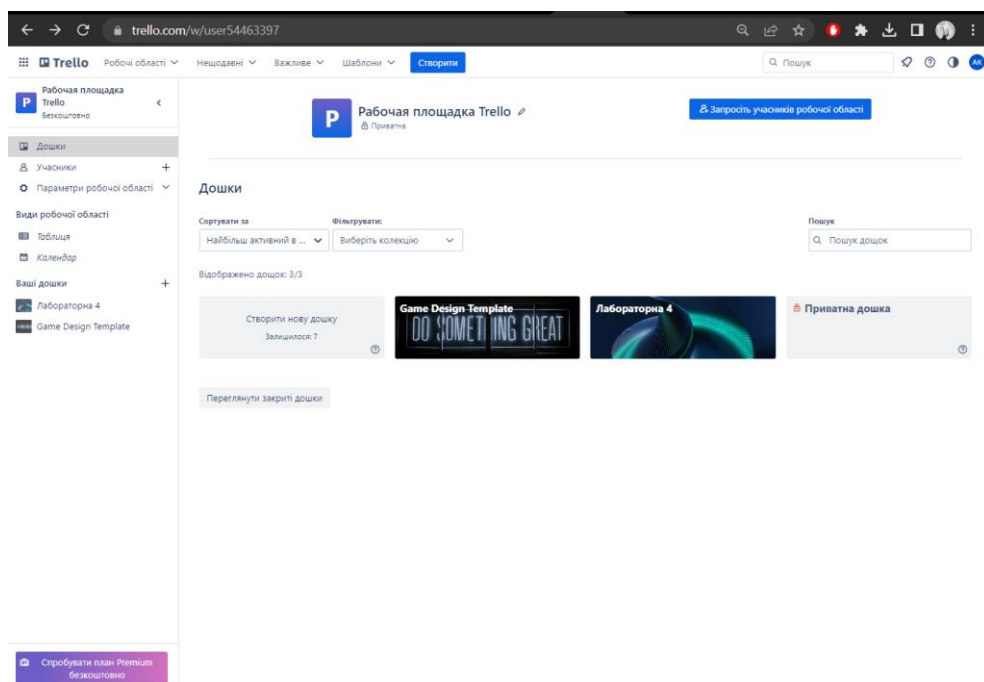


Рисунок 1.2 – Робоче вікно Trello [5]

1.3.2 Todoist

Переваги:

- досить простий та зрозумілий інтерфейс;
- багато функцій, таких як теги, пріоритети, дедлайни тощо;
- можливість додавати коментарі та взаємодіяти з іншими

користувачами.

Недоліки:

- для використання всіх функцій необхідно придбати платну версію;
- інтерфейс може здаватись перевантаженим (рис. 1.3);
- відсутня українська мова [6]-[7].

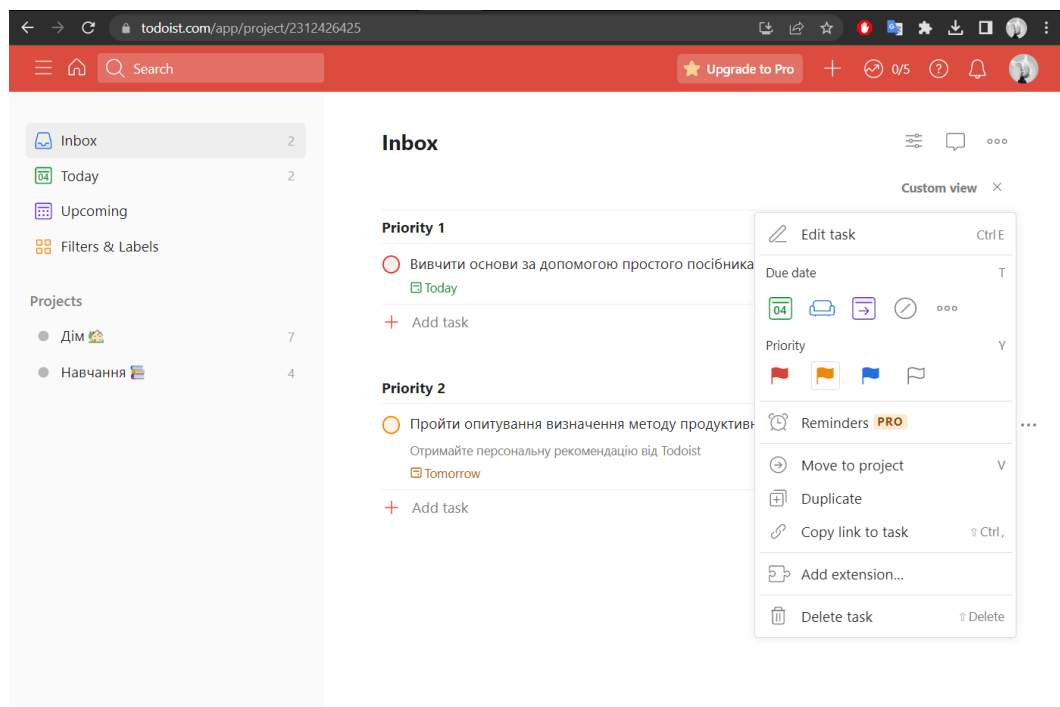


Рисунок 1.3 – Робоче вікно Todoist [8]

1.3.3 Google Keep

Переваги:

- надзвичайно простий та зрозумілий інтерфейс;

– можливість додавати звичайні нотатки, списки задач, малювати тощо;

– швидкий доступ до нотаток через будь-який пристрій з обліковим записом Google.

Недоліки:

– обмежена кількість функцій порівняно з іншими додатками;

– не забезпечує можливість спільної роботи з іншими користувачами

[6]-[7].

Вигляд робочого вікна Google Keep показаний на рис. 1.4.

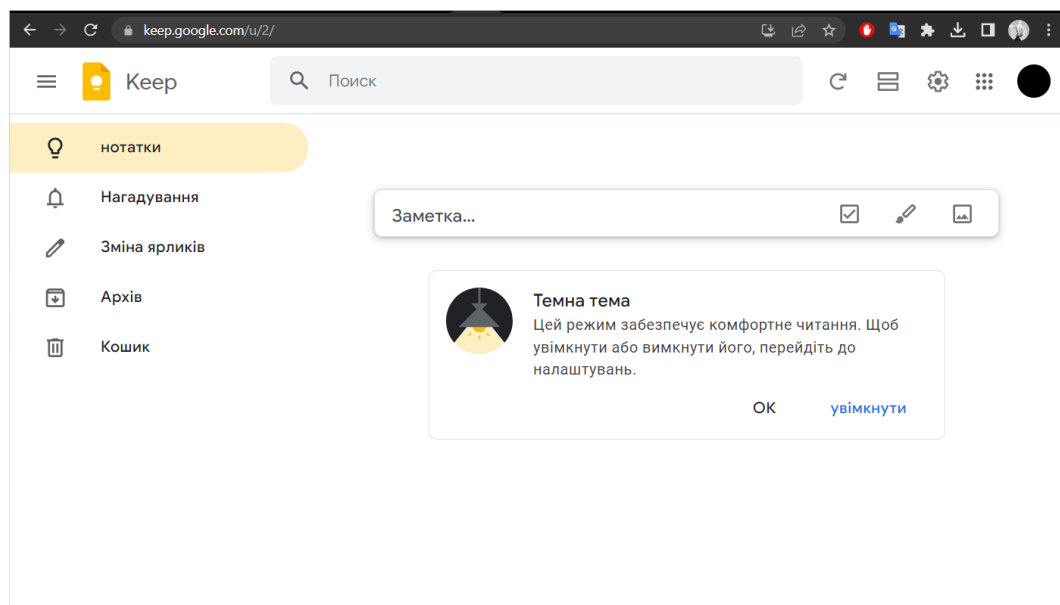


Рисунок 1.4 – Робоче вікно Google Keep [9]

1.3.4 Asana

Переваги:

– можливість керувати завданнями проєкту в команді;

– можливість присвоєння завдань, дедлайнів, коментарів тощо;

– зручна структура проєктів та завдань.

Недоліки:

– висока вартість платної версії додатку;

– складний інтерфейс (рис. 1.5) для новачків [6]-[7].

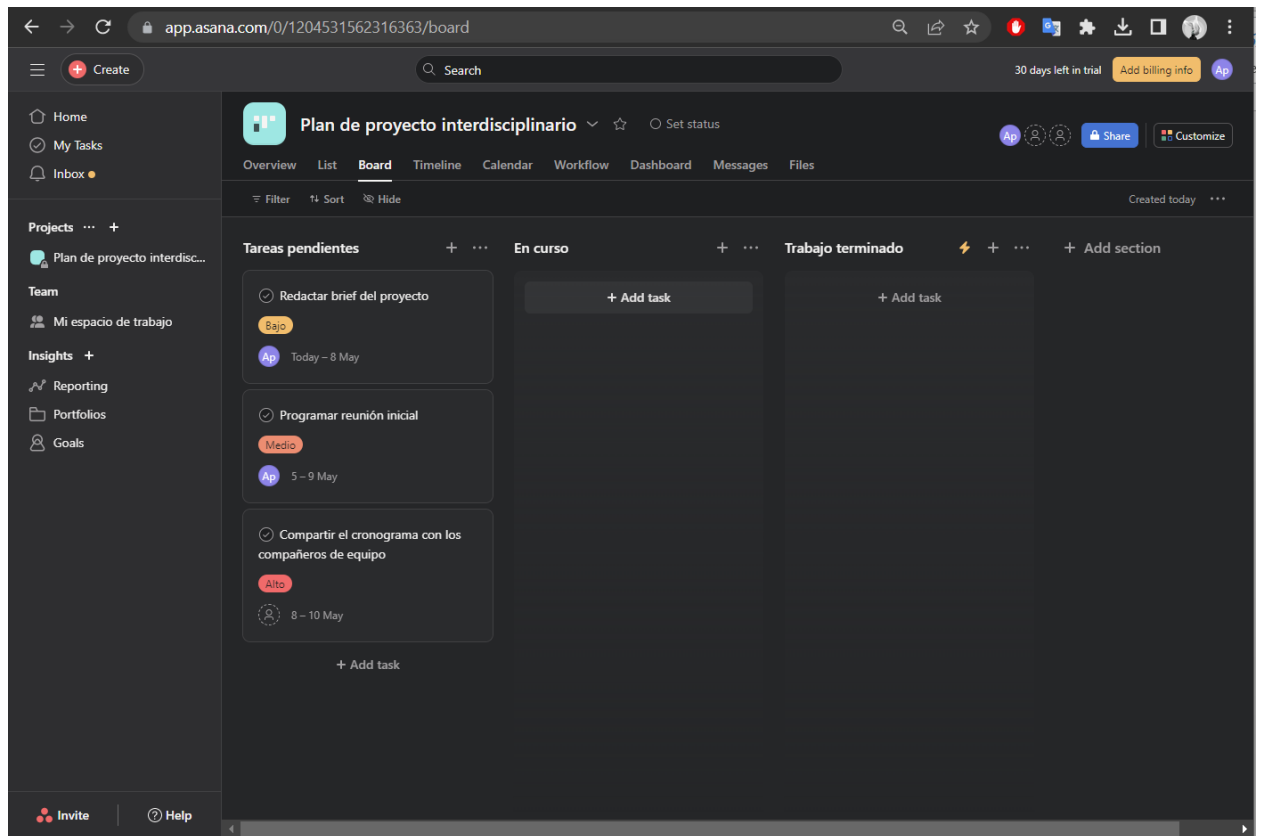


Рисунок 1.5 – Робоче вікно Asana [10]

1.3.5 Microsoft To Do

Переваги:

- інтеграція з Microsoft 365 та Outlook;
- простий та зрозумілий інтерфейс;
- можливість додавати списки завдань та розклади.

Недоліки:

- обмежена кількість функцій порівняно з іншими додатками;
- відсутність можливості спільної роботи з іншими користувачами [6]-[7].

Робоче вікно Microsoft To Do наведений на рис. 1.6.

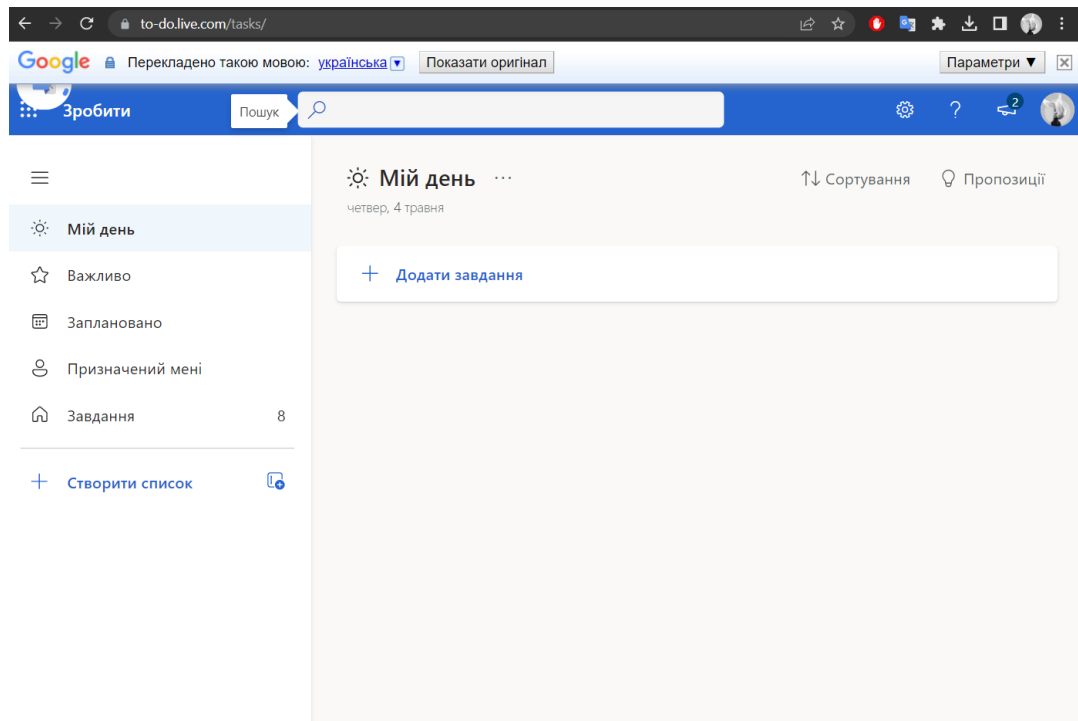


Рисунок 1.6 – Робоче вікно Microsoft To Do [11]

В цілому, всі ці додатки мають свої плюси та мінуси, і вибір залежить від потреб користувача. Деякі додатки мають більш розширені функції та можливості для спільної роботи з іншими користувачами, проте можуть бути складнішими у використанні, тоді як інші додатки мають більш простий інтерфейс, але менші можливості.

1.4 Постановка завдання

На основі аналізу існуючих додатків онлайн щоденників було вирішено створити застосунок який би містив основний та необхідний функціонал, а саме:

- можливість створення, редагування та видалення записів та планів в щоденнику та в списку задач;
- календар, що дозволяє відстежувати задачі на визначений тиждень;
- можливість додавати до кожного пункту списку відмітку про виконання, що допомагає відслідковувати ваш прогрес і результати;
- реєстрація або вхід користувача.

1.5 Висновки за розділом 1

У даному розділі було проведено огляд технологій створення онлайн щоденників та списків задач. Були проаналізовані переваги та недоліки популярних на даний момент онлайн щоденників. Було також визначено вимоги до створення інтерфейсу онлайн щоденника та встановлено вимоги до функціональних характеристик. Створення інтерфейсу та функціоналу онлайн щоденника є дуже високими, оскільки користувачі мають певні очікування від таких продуктів.

У рамках дипломної кваліфікаційної роботи необхідно розробити онлайн щоденник з можливістю створення списків задач. Цей продукт має на меті надати користувачам зручний інструмент для планування їхнього часу, контролю над задачами та досягнення їх цілей.

2 АНАЛІЗ ТА ВИБІР ЗАСОБІВ РОЗРОБКИ ПРОГРАМНОГО ПРОДУКТУ

Серед найважливіших етапів створення програмного продукту є зважування різних факторів і елементів, які входять у процес планування та розробки. Необхідно приділяти особливу увагу інструментам розробки, оскільки вибір відповідного інструменту може суттєво вплинути на результат проєкту з точки зору його якості, ефективності та загального успіху.

У розділі детально розглядається кожен аспект засобів розробки, оцінюються їхні переваги та недоліки, які з рештою допоможуть зробити усвідомлений вибір інструменту, що підходить для вимог та успішного розроблення програмного продукту.

2.1 Аналіз видів веброзробки та мов програмування

Веброзробка відноситься до створення вебсайтів в Інтернеті та стосується недизайнерських аспектів вебсайту, таких як створення функціональних можливостей і функцій за допомогою мов програмування, розмітки та сценаріїв. Розробники зосереджуються на технічних аспектах створення вебсайту, таких як архітектура, програмування та інтеграція додатків, а також графіка [7].

2.1.1 Види веброзробки

Існує три види веброзробки [12]:

– backend – розробники зосереджені на розробці серверної частини застосунку, включно з обробкою даних, взаємодією з базами даних та розробкою логіки додатку. Вони використовують такі мови програмування,

як Python, Ruby, Java, PHP та інші, для створення та керування інфраструктурою backend;

- frontend – розробники створюють користувацький інтерфейс, з яким користувачі взаємодіють у браузері. Вони створюють та оформляють вебсторінки за допомогою HTML, CSS та JavaScript, додають динамічну функціональність, перевірку даних та взаємодію з сервером;

- fullstack – розробники поєднують у собі навички як Backend, так і Frontend і мають можливість брати участь в обох частинах процесу розробки. Вони можуть виконувати весь цикл розробки вебдодатка, включно з розробкою серверної частини додатка, створення користувацького інтерфейсу та роботою з базою даних.

У ході аналізу цих видів веброзробки було виявлено основні особливості кожного з них. При розгляді цих видів та їх застосування, було прийняте рішення обрати Frontend розробку з використанням PHP використовується для серверної обробки та генерації вебсторінок. Використання HTML, CSS та JavaScript дозволять творити ефективний та зручний інтерфейс для взаємодії з користувачем. Це дасть можливість розміщувати елементи на сторінці, стилізувати їх та надавати динамічну функціональність. Також це дасть змогу продуктивно взаємодіяти з користувачем та забезпечити корисний та приємний досвід використання вебдодатку.

Враховуючи ці фактори, Frontend є найкращим вибором для проєкту. Можна зосередитись на створенні інтерфейсу і забезпечені комфортної взаємодії з користувачами без необхідності заглиблюватися у розробку серверної частини.

2.1.2 Мови програмування

Одним із цікавих аспектів розробки сайту є різноманітність мов програмування. Існує безліч різних мов, кожна з яких має свої особливості та

переваги. Тому в розробників є можливість вибрати мову, яка найкращим чином відповідає їхнім потребам і проєкту. Для виконання дипломної кваліфікаційної роботи було вирішено використовувати PHP, JavaScript, HTML та CSS.

Розглянемо мови програмування для Frontend:

JavaScript — це мова програмування для розробки динамічних інтерактивних вебсайтів. Це дозволяє впливати на поведінку сторінки, маніпулювати елементами DOM (Document Object Model) і взаємодіяти з користувачем.

Основні функції JavaScript включають:

- простота використання: JavaScript має простий синтаксис, який полегшує написання та розуміння коду;
- вебфункції: JavaScript використовується для динамічної зміни вебсторінок, перевірки форм, обробки подій, анімації та взаємодії з користувачем;
- кросплатформенність: JavaScript може працювати на різних платформах, таких як веббраузери, сервери, мобільні пристрої та навіть IoT-пристрої;
- об'єктно-орієнтоване програмування: JavaScript підтримує об'єктно-орієнтований підхід, де функції та об'єкти є основними будівельними блоками коду;
- великі можливості розширення: за допомогою додаткових бібліотек і фреймворків, таких як React, Angular або Vue.js, ви можете створювати складні програми та SPA (односторінкові програми);
- інтеграція з HTML і CSS: JavaScript може взаємодіяти з документами HTML і змінювати їх структуру, стиль і вміст;
- асинхронний код: JavaScript підтримує виконання асинхронного коду за допомогою зворотних викликів, обіцянок і функцій `async/await` для ефективного обробки тривалих операцій [13].

HTML (Hypertext Markup Language) — стандартна мова розмітки, яка використовується для створення вебсторінок і вебдокументів. Він використовується для структурування та організації вмісту вебсторінок, включаючи текстовий вміст, зображення, посилання, таблиці, форми, мультимедіа та інші елементи.

Основні переваги HTML включають:

- простота використання: синтаксис HTML простий і легкий для вивчення навіть для новачків;
- кросплатформенність: усі сучасні веббраузери підтримують HTML і можуть працювати на різних платформах, таких як комп'ютери, мобільні пристрої, планшети та інші пристрої;
- семантична розмітка: HTML надає спеціальні елементи та атрибути, які допомагають зрозуміти семантичну структуру вмісту. Це покращує доступність вебсторінки, оптимізацію SEO та загальну зрозумілість;
- розширюваність: HTML можна розширити за допомогою CSS (каскадних таблиць стилів) для стилізації зовнішнього вигляду та JavaScript для додавання динамічних функцій. Він дозволяє створювати багатофункціональні вебсторінки та додавати інтерактивні ефекти;
- сумісність із SEO: із семантичною розміткою та правильною структурою вебсторінки, створені за допомогою HTML, швидше за все, будуть проіндексовані пошуковими системами та відображені в результатах пошуку [14].

CSS (Cascading Style Sheets) — це мова стилів для розробки вебсторінок. Це дозволяє відокремити стиль вебдокумента від його структури, дозволяючи змінювати зовнішній вигляд вебсторінки, не змінюючи сам код HTML.

Основні переваги CSS:

- розділення стилю та вмісту: CSS дозволяє відокремити стиль вебсторінки від її структури та вмісту. Це дозволяє легко змінити зовнішній вигляд сторінки, не впливаючи на вміст;

- зручність і багаторазове використання: за допомогою CSS можна створювати стилі, які можна застосовувати до багатьох елементів на сторінці. Це полегшує роботу зі стилями та забезпечує ефективне повторне використання коду;

- легко змінити зовнішній вигляд: CSS дозволяє легко змінити зовнішній вигляд вебсторінки, просто змінивши стиль. Це дозволяє швидко та легко змінювати дизайн без редагування структури HTML;

- сумісність із різними браузерами: CSS підтримується всіма сучасними веббраузерами та гарантує, що сторінки виглядатимуть однаково на різних платформах і пристроях;

- адаптивний дизайн: за допомогою CSS ви можете створювати адаптивний дизайн, забезпечуючи найкращий вигляд ваших сторінок на різних пристроях і екранах;

- анімація та переходи: CSS дозволяє створювати ефекти анімації та переходи між станами елементів на сторінці, покращуючи їх візуальну привабливість. [15].

PHP (Hypertext Preprocessor) — мова програмування на стороні сервера для розробки вебдодатків. Він дозволяє створювати динамічні та інтерактивні вебсторінки, обробляти форми, працювати з базами даних, керувати сесіями користувачів тощо.

До основних можливостей PHP відносяться:

- робота з формами: PHP дозволяє отримувати дані з форм, обробляти їх і виконувати перевірку на сервері перед збереженням у базі даних або подальшою обробкою;

- робота з базами даних: PHP забезпечує можливість взаємодії з різними системами керування базами даних, такими як MySQL, PostgreSQL, Oracle тощо. Він дозволяє робити запити до бази даних, додавати, змінювати та видаляти дані;

- створення динамічних вебсторінок: PHP дозволяє вставляти динамічний вміст у шаблони HTML, дозволяючи створювати інтерактивні сторінки, які змінюються залежно від умов і даних.

- робота з файлами: PHP дозволяє читати, писати, видаляти та редагувати файли на сервері. Це корисно, коли потрібно зберігати й обробляти дані, завантажені користувачем;

- керування сесіями користувачів: PHP надає можливість створювати сесії користувачів і керувати ними, дозволяючи зберігати стан і дані між відвідуваннями вебсторінок;

- робота з API: PHP чудово підходить для взаємодії із зовнішніми службами та API. Він дозволяє виконувати HTTP-запити, отримувати та обробляти дані в JSON, XML та інших форматів;

- обробка зображень: PHP має такі функції обробки зображень, як зміна розміру, обрізка, створення мініатюр тощо [16].

Frontend є однією з ключових складових розробки вебдодатків, яка займається створенням інтерфейсу користувача та забезпеченням його взаємодії з додатком. Зважаючи на особливості роботи, буде використано HTML для створення структури та розміщення елементів на сторінці, CSS для стилізації елементів та надання їм зовнішнього вигляду, JavaScript для надання динамічної функціональності та взаємодії з користувачем та PHP для серверної обробки інформації та створення вебдодатку.

2.2 Зберігання даних

В процесі розробки вебдодатків, збереження інформації є важливою задачею. Коли користувачі взаємодіють з додатком, їхні налаштування, стан додатка та інші дані можуть потребувати збереження, щоб забезпечити постійність та персоналізацію досвіду. Для роботи з сайтом та збереження даних було вирішено обрати локальне збереження даних за допомогою Open Server.

2.2.1 Open Server

Open Server — це серверне середовище, яке дозволяє легко налаштувати та запустити вебсервер на комп'ютері для локальної розробки та тестування вебсайтів. Використовуючи Open Server, можна створити повноцінне середовище вебсервера, яке включає всі необхідні компоненти, такі як вебсервер Apache, база даних MySQL і мова програмування PHP.

Open Server можна налаштувати за допомогою простого та зручного інтерфейсу. Можна вибрати потрібні версії компонентів, встановити необхідні розширення та налаштувати параметри сервера відповідно до ваших вимог. Він також підтримує використання віртуальних хостів, що дозволяє запускати кілька вебсайтів одночасно з різними конфігураціями.

Однією з переваг Open Server є можливість швидкого перемикання між різними версіями вебсерверів, баз даних і PHP. Це дозволяє легко перевірити свій вебсайт у середовищах різних версій і перевірити їх сумісність.

Open Server також має додаткові функції, які полегшують розробку вебсайтів. Наприклад, він має вбудований phpMyAdmin, який дозволяє зручно керувати базами даних MySQL. Існують також такі засоби розробки, як Xdebug для налагодження коду PHP і Composer для керування залежностями проєкту.

Отже, Open Server є зручним і потужним інструментом для локальної розробки та тестування вебсайтів. Він забезпечує швидкий спосіб налаштування середовища вебсервера та забезпечує зручність і гнучкість у роботі над проєктами [17].

2.2.2 PhpMyAdmin

PhpMyAdmin — це безкоштовний інструмент, написаний на PHP, призначений для адміністрування серверів баз даних MySQL або MariaDB.

Його можна використовувати для виконання більшості адміністративних завдань, включаючи створення баз даних, виконання запитів і додавання облікових записів користувачів.

Зараз phpMyAdmin може:

- створювати, переглядати, редагувати та видаляти бази даних, таблиці, подання, стовпці та індекси;
- відображати кілька наборів результатів через збережені процедури або запити;
- обслуговування сервера, баз даних і таблиць, з пропозиціями по конфігурації сервера;
- виконувати, редагувати та створювати закладки для будь-яких інструкцій SQL , навіть пакетних запитів;
- робити імпорт даних і структур MySQL з електронних таблиць OpenDocument, а також файлів XML , CSV і SQL;
- адмініструвати декілька серверів;
- додавати, редагувати та видаляти облікові записи та привілеї користувачів MySQL;
- використовуючи Query-by-example (QBE), створювати складні запити, автоматично з'єднуючи необхідні таблиці;
- створювати PDF- графічні зображення макета бази даних;
- здійснювати глобальний пошук у базі даних або її підмножині;
- перетворювати збережені дані в будь-який формат за допомогою набору попередньо визначених функцій;
- відстежувати зміни в базах даних і таблицях;
- підтримка таблиць InnoDB і зовнішніх ключів;
- підтримка mysqli, покращене розширення MySQL [18].

Приклад збереження та передачі даних показаний на рис. 2.1.

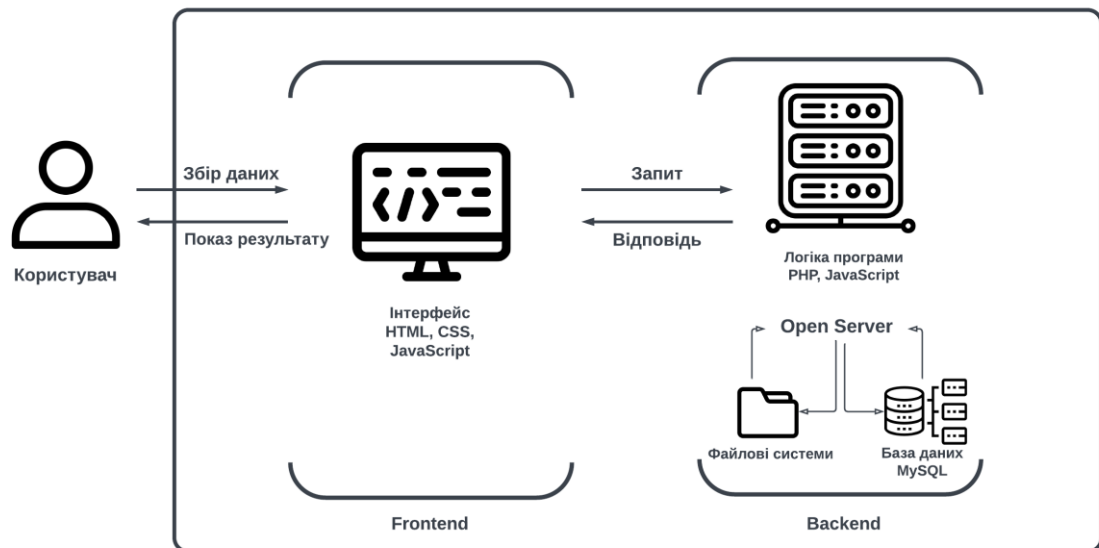


Рисунок 2.1 – Робота програми

2.3 Вибір інструментів розробки

Вибір засобів розробки є важливим етапом розробки вебдодатків. Інструменти розробки включають набір програмних засобів, фреймворків, бібліотек, середовищ розробки та інших інструментів, які допомагають розробникам створювати, тестувати та розгортати вебдодатки більш ефективно.

Вибираючи інструмент розробки, повинно бути враховувано багато факторів, таких як потреби проєкту, власні навички та досвід, масштабованість, підтримка спільноти розробників тощо. Інструменти розробки можуть включати мови програмування, фреймворки, редактори коду, системи контролю версій, засоби автоматизації тестування тощо. Для виконання дипломної кваліфікаційної роботи був вибраний PhpStorm.

PhpStorm — це IDE (інтегроване середовище розробки) для PHP і веброзробників, розроблене JetBrains.

Основними характеристиками PhpStorm є:

- ідеальне середовище розробки: більшість розробників PHP використовують не лише PHP, але й HTML, CSS, JavaScript та інші мови.

PhpStorm полегшує роботу з усіма мовами, надаючи підтримку автозавершення коду та підсвічування синтаксису;

- підтримує кілька фреймворків PHP: якщо використовується фреймворк PHP, PhpStorm стане ідеальним вибором. Він ефективно інтегрується з багатьма фреймворками, такими як Laravel, CakePHP, Symfony2 і Yii — сервер додатків PHP Framework від Future Hosting також підтримує їх усі;

- відмінна інтеграція контролю версій: PhpStorm повністю підтримує різні системи контролю версій, включаючи GitHub і Git. Розробники можуть використовувати IDE для розгалуження та злиття. Крім того, PhpStorm включає відмінну підтримку FTP, що дозволяє завантажувати код на віддалений сервер за лічені секунди;

- підтримка бази даних: PhpStorm надає інструменти та функції підтримки коду для роботи з SQL і базами даних у ваших проєктах. Він підключається до баз даних, редагує табличні дані, виконує запити та навіть аналізує діаграми за допомогою діаграм UML;

- налагодження та тестування: налагодження без конфігурації спрощує налагодження програми. PHPUnit підтримує розробку та виконання модульних тестів безпосередньо з вашої IDE. Профілі програми за допомогою Xdebug і Zend Debugger і перевірте підсумковий звіт у PhpStorm [19].

2.4 Висновки за розділом 2

В цьому розділі було проведено аналіз технологій, що використовуються для розробки подібних онлайн щоденників. На основі аналізу було прийняте рішення реалізувати розробку сайт, а саме онлайн щоденник зі списком задач.

Проаналізувавши вибір засобів розробки програмного продукту було прийнято рішення використовувати HTML для створення структури та

розташування елементів, CSS для стилізації сайту, JavaScript для забезпечення динамічних функцій та взаємодії з користувачем та PHP для можливості обробки та генерації динамічного контенту на серверному рівні.

Дані будуть зберігатися в базі даних MySQL у локальному сервері phpMyAdmin. Ці механізми дозволять зберігати дані безпосередньо в браузері користувача та забезпечують локальне зберігання та доступ до цих даних під час наступних сеансів використання програми.

3 ОСНОВНІ РІШЕННЯ ЩОДО РЕАЛІЗАЦІЇ СИСТЕМИ

У цьому розділі розглядаються основні рішення щодо розробки класів, основні алгоритми розробки, рішення щодо розробки інтерфейсу користувача, рішення щодо зберігання даних та використання бази даних.

3.1 Розробка схеми отримання даних

Основна мета системи онлайн-щоденника — забезпечити зручне та надійне зберігання записів користувачів. Для цього було вирішено використовувати базу даних MySQL, потужне і широко використовуване середовище зберігання реляційних даних. Кожен запис — це окремий рядок таблиці, що містить інформацію про вміст тексту, дату створення та інші додаткові поля. Мова програмування PHP використовується для отримання даних з бази даних MySQL і подальшої їх обробки. PHP надає зручні інструменти для підключення до баз даних, виконання запитів і отримання необхідної інформації.

Щоб почати використовувати щоденник обов'язковим є реєстрація або вхід, якщо користувач був вже зареєстрований. При реєстрації у користувача просять його нік, ім'я та просять вигадати пароль. Ці дані будуть зберігатися у базі (рис. 3.1).

Параметр “id” повертає унікальний ідентифікатор користувача, який буде використовуватись для подальших збережень інформації про цього користувача.

Параметр “username” повертає нік користувача, який в подальшому буде використовуватись для входу в щоденник.

Параметр “full name” повертає ім'я користувача, який потім буде відображатись у його профілі.

Параметр “password” повертає пароль від щоденника, який в подальшому буде використовуватись для входу в щоденник.

Після вдалого збереження даних, користувача просять повторити нік та пароль для входу в щоденник.

←T→	id	username	full name	password
☐	4	user	Full Name	password

Рисунок 3.1 – Таблиця даних для зберігання інформації про користувача

Після входу в систему користувачу доступне створення списку задач на певний день. Йому доступне поле для запису задачі та обрання дати, на коли він хоче створити запис. Після внесення задачі у список, в базі (рис. 3.2) зберігаються данні про цей пункт списку.

Параметр “id” повертає унікальний ідентифікатор створеної задачі, який буде використовуватись для подальшого редагування, видалення та відмітки про виконання.

Параметр “id_user” повертає унікальний ідентифікатор користувача, який створив пункт.

Параметр “task” повертає текст пункту списку.

Параметр “date_task” повертає дату, на яку було створено задачу.

Параметр “checked” повертає індикатор виконання завдання. Якщо в колонці стоїть 0, це означає, що завдання не виконане, якщо стоїть 1 – виконане.

←T→	id	id_user	task	date_task	checked
☐	22	5	створити завдання	2023-05-29	0

Рисунок 3.2 – Таблиця даних для зберігання інформації про пункти списку задач користувача

На іншій сторінці онлайн щоденника зі списком задач є щоденник тижня, в якому користувач на певний день може записати свої плани,

міркування і тд. Після запису та збереження, дані будуть направлені в базу даних (рис. 3.3).

Параметр “id” повертає унікальний ідентифікатор створеного запису, який буде використовуватись для подальшого редагування та видалення.

Параметр “id_user” повертає унікальний ідентифікатор користувача, який створив цей запис.

Параметр “text” повертає текст з елементами (тегами) HTML-розмітки, які використовуються для створення структури та форматування тексту.

Параметр “date_diary” повертає дату, на яку було створено допис.

	id	id_user	text	date_diary
1			ще	2023-05-25

Рисунок 3.3 – Таблиця даних для зберігання інформації про записи користувача

3.2 Розробка вебінтерфейсу

Розробка вебінтерфейсу є невід’ємною частиною процесу створення сучасних вебдодатків. Під час розробки вебінтерфейсів використовуються різноманітні зовнішні бібліотеки JavaScript і стилі CSS, щоб покращити функціональність і зовнішній вигляд вебсторінок.

Однією з ключових бібліотек є jQuery, яка спрощує взаємодію з HTML-документами, обробку подій, анімацію та AJAX. Це дозволяє досягти кількох функцій з меншими зусиллями та скоротити час розробки.

```
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js
"></script>
```

Також для роботи з датами та часом використовується Moment.js, який надає розширені можливості для роботи з датами, часом, форматами та маніпуляціями.

```
<script
src="https://cdn.jsdelivr.net/momentjs/2.10.3/moment.min.js"></script>
```

Bootstrap Datetimepicker використовується для вибору дати й часу на вебсторінках, він забезпечує зручний інтерфейс для вибору дати й часу з інтерактивним календарем і годинником.

```
<script
src="https://cdnjs.cloudflare.com/ajax/libs/moment.js/2.10.3/moment-
with-locales.js" ></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/bootstrap-
datetimepicker/4.17.37/js/bootstrap-datetimepicker.min.js"
integrity="sha512-
Y+0b10RbVUTf3Mi0EgJue0FoheNzentTMMIE20reNbqnUPNbQj8zmjK3fs5D2WhQeGWIem
2G2UkKjAL/bJ/UXQ==" crossorigin="anonymous" referrerpolicy="no-
referrer"></script>
```

Крім того, TinyMCE — це бібліотека, яка дозволяє легко інтегрувати розширений текстовий редактор у вебсторінку чи програму, надаючи розширені можливості форматування та редагування тексту.

```
<script
src="https://cdn.tiny.cloud/1/qagffr3pkuv17a8on1afax661irst1hbr4e6tbv8
88sz91jc/tinymce/5-stable/tinymce.min.js"></script>
<script
src="https://cdn.tiny.cloud/1/{YOUR_API_KEY}/tinymce/5/tinymce.min.js"
referrerpolicy="origin"></script>
```

Крім того, для використання піктограм на вебсторінках використовується Font Awesome Icons, набір піктограм, який пропонує широкий вибір графічних символів, щоб додати вищий рівень привабливості та функціональності вебсторінок.

```
<script      src="https://kit.fontawesome.com/df07cb0f03.js"
crossorigin="anonymous"></script>
```

Для стилізації тексту та назви шрифтів використовується Google Fonts, що дозволяє підключати та використовувати різні шрифти з великої кількості доступних варіантів.

```
<link rel="preconnect" href="https://fonts.googleapis.com">
<link      rel="preconnect"      href="https://fonts.gstatic.com"
crossorigin>
<link
href="https://fonts.googleapis.com/css2?family=Abhaya+Libre:wght@700;800&family=Atma:wght@500;600&family=Lato:wght@300;400;700&display=swap"
rel="stylesheet">
```

3.3 Опис реалізованих функцій

3.3.1 Вхід та реєстрація користувача

Даний код відображає форму входу та форму реєстрації на вебсторінці.

У формі входу користувач повинен ввести свій нікнейм (логін) та пароль, які передаються на сервер для перевірки за допомогою POST-запиту до файлу "login-validation.php". Якщо введені дані некоректні або не відповідають вимогам, повідомлення про помилку виводяться на сторінку.

```
<!-- форма входу -->
<div id="login-register">
```

```

<div class="field">
    <h3 style="width:100px">Вхід</h3>
    <form          action="../controller/login-validation.php"
method="POST">
        <input          type="text"          class="form-general"
name="username" placeholder="Нік користувача">

        <input          type="password"       class="form-general"
name="password" placeholder="Пароль">

        <!-- не правильний вхід -->
        <?php
        if (isset($_GET['login'])) {
            if ($_GET['login'] == 'user_not_logged_in'){
                echo  '<span style="color:red">Не вдалось
увійти!<br></span>';
            }else {
                echo      '<span style="color:red">'.
$_GET['login'] .'<br></span>';
            }
        }
        ?>
        <input type="submit" class="buttons" value="УВІЙТИ">
    </form>
</div>

```

У формі реєстрації користувач повинен ввести нікнейм, повне ім'я та пароль, які також передаються на сервер за допомогою POST-запиту до файлу "register-validation.php". Якщо введені дані некоректні або не відповідають вимогам, повідомлення про помилку виводяться на сторінку.

```

<!-- форма реєстрації -->
<div class="field">
    <h3>реєстрація</h3>

```

```

        <form        action="../../../controller/register-validation.php"
method="POST">
            <input        type="text"        class="form-general"
name="username" placeholder="Ім'я користувача" required>

            <input type="text" class="form-general" name="full-
name" placeholder="Ім'я" required>

            <input        type="password"        class="form-general"
name="password" placeholder="Пароль" required>

            <!-- не правильна форма реєстрації -->
            <?php
            if (isset($_GET['register'])) {
                if ($_GET['register'] == 'user_not_logged_in'){
                    echo  '<span style="color:red">Не вдалось
увійти!<br></span>';
                }else {
                    echo      '<span style="color:red">'.
$_GET['register'] .'<br></span>';
                }
            }
            ?>
            <input        type="submit"        class="buttons"
value="ЗАРЕЄСТРУВАТИСЯ">
        </form>
    </div>
</div>

```

При наявності помилок входу або реєстрації (якщо вони передаються через GET-параметр), відповідне повідомлення про помилку виводиться на сторінку з червоним кольором.

Форми мають відповідні HTML-класи та стилі для відображення на вебсторінці, а також кнопки "УВІЙТИ" та "ЗАРЕЄСТРУВАТИСЯ" для підтвердження введених даних.

3.3.2 Робота з цілями у списку задач

Цей код відповідає за відображення списку завдань на основі днів тижня.

Починаючи з рядка, де включений файл "tasks.php", код отримує список завдань з сервера. Потім визначається масив `$week_days`, який містить назви днів тижня українською мовою.

Далі виконується цикл `foreach`, який пробігає по кожному елементу масиву `$week_days` (тобто по кожному дню тижня). Для кожного дня створюється блок `<div class="field">`, в якому виводиться назва дня `<h4><?php echo $day; ?></h4>`.

Усередині кожного блоку днів тижня відбувається ще один цикл `foreach`, який перебирає кожне завдання зі списку `$tasks`. Завдання відображаються як пункти списку `<li>`, де назва завдання виводиться за допомогою `<?php echo $task['task'] ?>`.

Крім того, для кожного завдання відображаються додаткові елементи у вигляді іконок. Наприклад, іконка видалення, яка виконує посилання на `"../controller/delete_task.php"` для видалення завдання, а також іконки для позначення стану виконання завдання.

Для редагування завдання також використовується вікно, яке відображається при натисканні на іконку зміни завдання. Вікно містить форму, в якій користувач може відредагувати назву завдання і зберегти зміни. Для збереження використовується форма з методом `POST`, яка вказує на `"../controller/edit_task.php"` для обробки змін.

Код генерує відповідний HTML-код, який відображає список завдань організований за днями тижня. Кожен день має свій власний блок, в якому


```

        <a
href="../../../controller/delete_task.php?task_id=<?php    echo    $task['id']
?>">

                <i    class="fa    fa-eraser"    aria-
hidden="true"></i></a>

        <?php

                //перевірка виконання завдання
                if ($task['checked'])
                {
                        ?>
                        <a
href="../../../controller/do_task.php?task_id=<?php            echo            $task['id']
?>&check=0">

                                <i    class="fas    fa-circle-
xmark"></i></a>

                        <?php
                } else
                {
                        ?>
                        <a
href="../../../controller/do_task.php?task_id=<?php            echo            $task['id']
?>&check=1">

                                <i    class="fas    fa-circle-
check"></i></a>

                        <?php
                }
                ?>

                <!-- кнопка змінити завдання -->
                <a    href="#popup?nmr_task=<?php    echo
$task['id']    ?>"    class="edit_btn">

                        <i    class="fa    fa-pencil"></i></a>

```


Цей код відповідає за додавання нового завдання користувачем.

Внутрішній блок `<div class="field add-task">` створює блок для додавання завдання. Нижче знаходиться форма `<form>`, яка використовує метод POST і має атрибут `action="../controller/add_task.php"`. Цей атрибут вказує на шлях до файлу "add_task.php", який відповідає за обробку доданого завдання.

У формі є наступні елементи введення:

- `<input type="text" name="task-text" class="texts" placeholder="Ціль на день..." maxlength="50" required>`: це текстове поле, де користувач може ввести опис або назву завдання. Введений текст буде передано на сервер з параметром "task-text". Також, встановлено максимальну довжину тексту у 50 символи та обов'язковість заповнення.

- `<input type="date" name="date-task" id="date-task">`: це поле вводу дати, де користувач може вибрати дату для завдання. Введена дата буде передана на сервер з параметром "date-task".

Коли користувач заповнює форму та натискає кнопку "Внести", дані з форми будуть відправлені на сервер для обробки. Файл "add_task.php" візьме введені дані та збереже нове завдання у відповідному місці (наприклад, в базі даних або файловій системі)..

```
<div class="field add-task">
    <h4>Додати задачу</h4>

    <form action="../controller/add_task.php" method="POST">
        <input type="text" name="task-text" class="texts"
placeholder="Ціль на день..." maxlength="50" required>
        <h4>На яку дату задача:</h4>
        <input type="date" name="date-task" id="date-task">

        <button type="submit">Внести</button>
    </form> </div>
```

3.3.3 Робота з текстом в щоденнику

Цей код відповідає за відображення записів у щоденнику користувача на кожен день тижня.

Внутрішній блок `<div class="field add-days">` створює блок для відображення записів у щоденнику. У цьому блоку є посилання `<a href="#popup-create" class="create_btn"><i class="fa fa-plus"></i></a>`, яке представляє кнопку для створення нового запису у щоденнику. При натисканні на цю кнопку відкривається вікно для створення запису.

Після цього блоку знаходиться список `<ul>`, в якому відображаються записи щоденника користувача для конкретного дня тижня. Кожен запис представлений елементом списку `<li>`, який містить текст запису.

Крім того, для кожного запису є наступні опції:

– кнопка видалення: `<a href=" ../controller/delete_diary_record.php?task_id=?php echo $record['id'] ?>"><i class="fas fa-eraser"></i></a>`. При натисканні на цю кнопку виконується видалення запису з щоденника;

– кнопка редагування: `<a href="#popup?nmr_task=?php echo $record['id'] ?>" class="edit_btn"><i class="fa fa-pencil"></i></a>`. При натисканні на цю кнопку відкривається вікно для редагування запису;

– вікно редагування запису: `<div class="edit-task" id="popup?nmr_task=?php echo $record['id'] ?>">`. У цьому вікні користувач може змінити текст запису і зберегти його з новими змінами.

Код також включає обробку записів для кожного дня тижня за допомогою циклу `foreach ($week_days as $index => $day)`. Змінна `$week_days` містить масив з назвами днів тижня, а змінна `$index` представляє індекс поточного дня. Цикл перебирає кожен день тижня і відображає відповідні записи щоденника для кожного дня.

```

<?php
    //Переглядайте дні тижня, навіть якщо в цей день у користувача
    ще немає завдань
    include '../controller/diary.php';
    $week_days = ['ПОНЕДІЛОК', 'ВІВТОРОК', 'СЕРЕДА', 'ЧЕТВЕР',
    "П'ЯТНИЦЯ", 'СУБОТА', 'НЕДІЛЯ'];
    foreach ($week_days as $index => $day)
    {
        ?>
        <div class="field add-days">

            <h4><?php echo $day; ?></h4>

            <!-- Додана кнопка створення запису -->
            <a href="#popup-create" class="create_btn">
                <i class="fa fa-plus"></i></a>

            <ul>
                <?php
                // Цикл навколо записів користувача
                foreach ($diary_records as $record)
                {
                    // Організація по днях тижня
                    if (intval(date('w',
    strtotime($record['date_diary']))) == $index + 1)
                    {
                        ?>
                        <div style="clear:both"></div>
                        <!-- Видалення задачі -->
                        <a
    href="../controller/delete_diary_record.php?task_id=<?php echo
    $record['id'] ?>">

                            <i class="fas fa-eraser"></i></a>
                        <!-- Кнопка редагування задачі -->

```

```

        <a href="#popup?nmr_task=<?php echo
$record['id'] ?>" class="edit_btn">
            <i class="fa fa-pencil"></i></a>
        <!-- Текст -->
        <li><?php echo $record['text']; ?></li>

        <!-- Редагування задачі -->
        <div class="edit-task"
id="popup?nmr_task=<?php echo $record['id'] ?>">

            <!-- кнопка закрити -->
            <a href="#" class="close"><i
class="fa fa-close"></i></a>

            <div style="clear:both"></div>

            <!-- вікно редагування -->
            <form
action="../controller/edit_diary.php" method="POST">
                <h3 class="rewrite">Запис</h3>
                <textarea name="diary-text"
rows="10" cols="80" class="texts" id="tiny-mce"><?php echo
$record['text'] ?></textarea>
                <input type="hidden"
name="task_id" value="<?php echo $record['id'] ?>">

                <button type="submit"
class="rewrite">Зберегти</button>

            </form>
        </div>
        <?php
    }
} ?>
</ul>
</div>

```

```

    <?php
}
?>

```

Цей код відповідає за відображення вікна для створення нового запису у щоденнику користувача.

Блок `<div class="edit-task1" id="popup-create">` створює вікно, яке містить форму для створення запису. Вікно має своє унікальне ідентифікатор `id="popup-create"`, що дозволяє викликати це вікно при натисканні на відповідну кнопку.

У вікні розміщені наступні елементи:

- кнопка закриття вікна: `<a href="#" class="close"><i class="fa fa-close"></i></a>`. При натисканні на цю кнопку вікно закривається;

- форма для створення запису: `<form action="../controller/add_record_diary.php" method="POST">`. Дана форма має метод POST і вказує на шлях до файлу `add_record_diary.php`, який оброблює дані форми.

У формі знаходяться такі поля:

- поле для введення тексту запису: `<textarea type="text" name="diary-text" rows="10" cols="80" class="texts" id="tiny-mce"></textarea>`. Користувач може ввести текст запису у це поле;

- поле для вибору дати запису: `<input type="date" name="date-diary" id="date-diary">`. Користувач може вибрати дату для запису;

- кнопка "Додати": `<button type="submit" class="rewrite">Додати</button>`. При натисканні на цю кнопку відбувається відправка форми з введеними даними.

Після заповнення форми і натискання кнопки "Додати", дані з форми будуть відправлені до файлу `add_record_diary.php` для подальшої обробки.

```

<div class="edit-task1" id="popup-create">
    <!-- кнопка закрити -->

```

```

        <a href="#" class="close"><i class="fa fa-
close"></i></a>
        <div style="clear:both"></div>
        <h3 class="rewrite">Запис</h3>

        <!-- вікно редагування -->
        <form action="../controller/add_record_diary.php"
method="POST">
            <textarea type="text" name="diary-text" rows="10"
cols="80" class="texts" id="tiny-mce"></textarea>
            <input type="date" name="date-diary" id="date-
diary">

            <button type="submit"
class="rewrite">Додати</button>
        </form>
    </div>

```

3.4 Висновки за розділом 3

У цьому розділі описано основні компоненти системи розробки.

Цей розділ містить опис коду, який відповідає за вхід систему, опис інтерфейсу програми та модульних шаблонів для її компонентів. Також наведено приклади реалізації основних функцій онлайн щоденника.

4 КЕРІВНИЦТВО ПРОГРАМІСТА

4.1 Призначення програми

Онлайн щоденник зі списком задач – це вебзастосунок, призначений для керування та організації завдань, розкладів та щоденних справ. Його основна мета полягає в тому, щоб дозволити користувачам створювати, редагувати та видаляти записи в щоденнику та пункти у списку задач.

4.2 Характеристики програми

Онлайн щоденник зі списком задач був розроблений за допомогою таких мов програмування як HTML для створення структури та розташування елементів, CSS для стилізації сайту, JavaScript для забезпечення динамічних функцій та взаємодії з користувачем та PHP для можливості обробки та генерації динамічного контенту на серверному рівні.

Застосунок виконує авторизацію користувача в щоденник. Має перегляд створення, редагування та видалення завдань у списку задач та записів у щоденнику. Також має фільтрацію по тижнях.

4.3 Умови застосування програми

До комп'ютеру, на якому виконується програма, висуваються наступні вимоги:

- операційна система Windows 7, 8, 10;
- наявність дисплею;
- Google Chrome;
- наявність миші та клавіатури;
- до 10 МБ оперативної пам'яті;
- встановлений Open Server Panel.

4.4 Звернення до програми

Для запуску програми потрібно попередньо упевнитися в відповідності характеристик комп'ютера, на який встановлена програма, до системних вимог програми. Потрібно упевнитися у наявності всіх файлів програми.

Звернення до сайту передбачене відбувається за допомогою додавання всіх файлів до Open Server, додавання бази даних в phpMyAdmin та запуску локального сервісу Open Server Panel.

4.5 Вхідні та вихідні дані

4.5.1 Вхідні дані

Введення даних є важливою частиною онлайн щоденика. Ці дані дозволяють користувачам заходити в свій аккаунт, перевіряти, додавати, редагувати та видаляти свої завдання та записи в журналі. Ось приклади введення, необхідного для різних функцій щоденника:

- автентифікація користувача: нік користувача, ім'я, пароль;
- задача: додавання та редагування завдання та додавання дати;
- запис у щоденнику: додавання та редагування тексту та додавання дати.

4.5.2 Вихідні дані

Вихідні дані є результатом взаємодії користувача зі застосунокм. Ці дані надають користувачам відгуки та інформацію про їхні завдання, записи в журналі та профілі користувачів. Ось приклад вихідних даних, які відображаються в щоденнику:

- задача: текстовий опис завдання та показ дати та статусу завдання;
- запис у щоденнику: текстовий зміст щоденника та показ дати;

- профіль користувача: відображення ніку та ім'я користувача.

4.6 Розгортання вебзастосунку на Open Server

Щоб працювати з локальним сайтом на Open Server, необхідно виконати наступні кроки:

Завантаження даних у папку Open Server: скопіювати папку з онлайн щоденником, де зберігаються всі його файли, до відповідної папки на локальному сервері, де встановлено Open Server. Зазвичай ця папка знаходиться в директорії 'openserver/domains/' або 'OSPanel\domains'.

Завантаження бази даних:

- запустити Open Server та впевнитись, що сервер бази даних (наприклад, MySQL) працює;
- виконати імпорт бази даних з файлу 'diary.sql'. Це можна зробити через інтерфейс адміністрування бази даних, наприклад, phpMyAdmin або командний рядок MySQL.

Налаштування з'єднання з базою даних:

- відкрити файл конфігурації онлайн щоденника 'bd-connection.php' та знайти розділ, де встановлюються параметри з'єднання з базою даних;
- вписати правильні значення для параметрів, таких як хост ('mysql:host=127.0.0.1;dbname=diary'), ім'я користувача бази даних ('diary'), пароль ('1qazxsw2') та назва бази даних ('diary').

Перевірка роботи:

- запустити локальний сервер та відкрийте веббраузер;
- якщо все налаштовано правильно, повинен відобразитись онлайн щоденник у браузері.

4.7 Висновки за розділом 4

У цьому розділі було надано основні характеристики, призначення та умови застосування онлайн щоденника.

Було розглянуто вимоги до інформативних структур та вхідні і вихідні дані проєкту.

Описаний процес звернення до програми та процес розгортання вебзастосунку у системі Open Server.

5 КЕРІВНИЦТВО КОРИСТУВАЧА

5.1 Призначення та умови застосування програми

5.1.1 Призначення програми

Онлайн щоденник зі списком задач – це вебзастосунок, призначений для керування та організації завдань, розкладів та щоденних справ. Його основна мета полягає в тому, щоб дозволити користувачам створювати, редагувати та видаляти записи, створювати розділи для впорядкування записів. Також онлайн щоденник дає змогу користувачам створювати, редагувати та видаляти списки задач.

5.1.2 Умови виконання програми

До комп'ютеру, на якому виконується програма, висуваються наступні вимоги:

- операційна система Windows 7, 8, 10;
- наявність дисплею;
- Google Chrome;
- наявність миші та клавіатури;
- до 10 МБ оперативної пам'яті;
- встановлений Open Server Panel.
- До користувачів висуваються наступні вимоги:
- володіти українською мовою на базовому рівні;
- володіння комп'ютером на базовому рівні.

5.2 Характеристики програми

Застосунок виконує авторизацію користувача в щоденник. Має перегляд створення, редагування та видалення завдань у списку задач та записів у щоденнику. Також має фільтрацію по тижнях.

5.3 Структура програми

5.3.1 Сторінка авторизації

При першому вході в сайт користувач потрапляє на головну сторінку, де він може дізнатися функції сайту та вигляд щоденника та списку задач (рис. 5.1). Щоб зайти в щоденник, потрібно зареєструватись. Для цього треба натиснути кнопку «Вхід» у верхньому правому кутку.

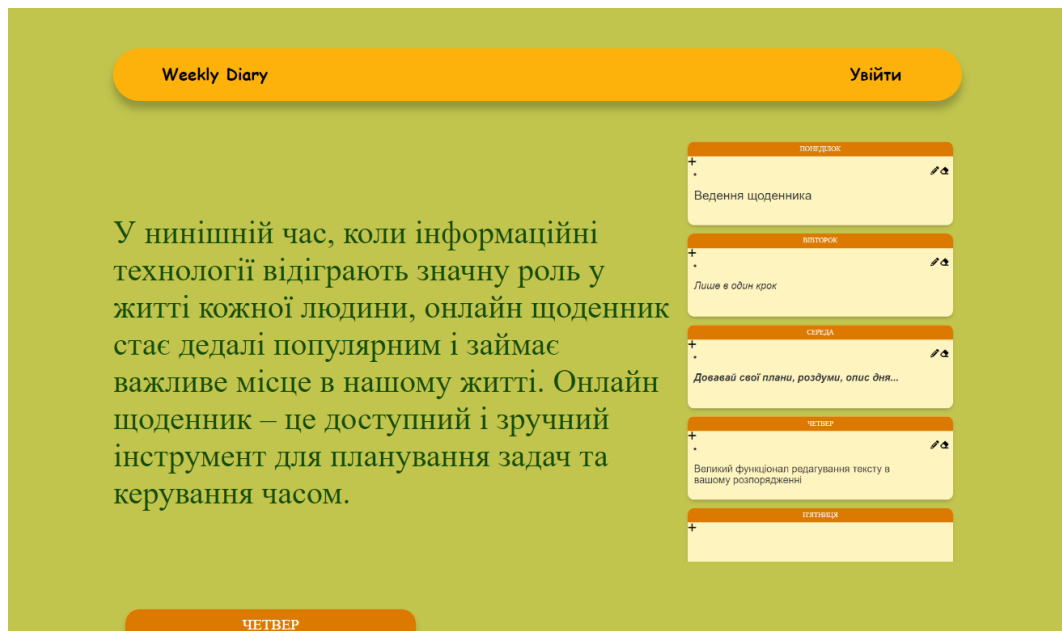


Рисунок 5.1 – Головна сторінка сайту

Після входу в щоденник, у верхній частині пропаде кнопка «Увійти». Замість неї з'являться кнопки «Щоденник», «Список задач» та «Вийти» (рис.5.2).

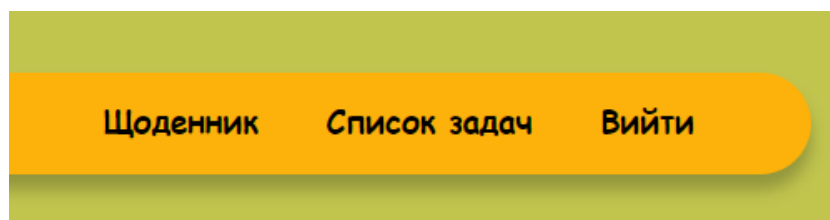


Рисунок 5.2 – Зміна головної сторінки

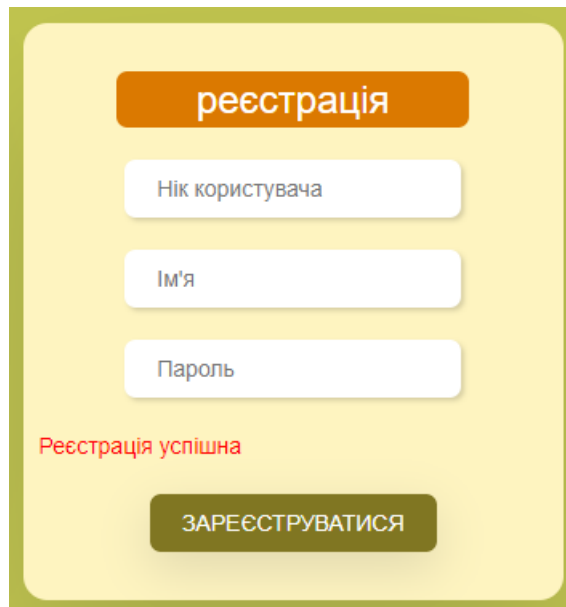
Сторінка авторизації містить 2 форми (рис. 5.3).

Рисунок 5.3 – Сторінка авторизації

З правої сторони форма реєстрації для нового користувача. Ця форма містить поля: нік користувача, ім'я та пароль. Всі поля є обов'язковими для заповнення (рис.5.4).

Рисунок 5.4 – Обов'язкове введення даних

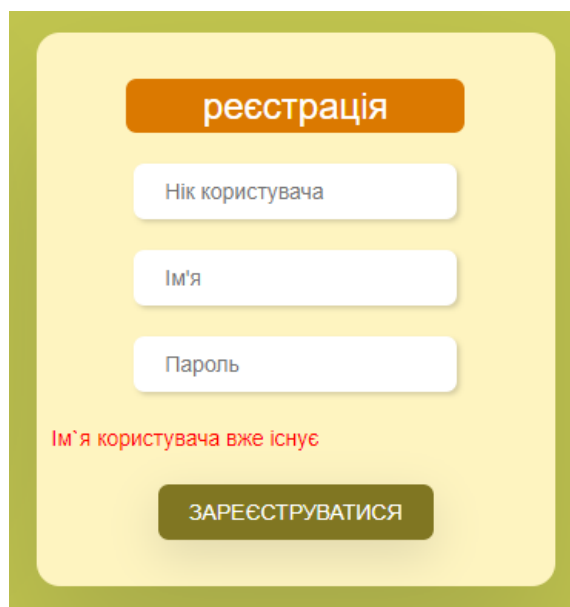
При вдалій реєстрації впливає червоний напис «Реєстрація успішна» (рис. 5.5).



The image shows a registration form titled "реєстрація" (registration) in an orange box. Below the title are three input fields: "Нік користувача" (username), "Ім'я" (name), and "Пароль" (password). A red message "Реєстрація успішна" (Registration successful) is displayed below the fields. At the bottom is a green button labeled "ЗАРЕЄСТРУВАТИСЯ" (REGISTER).

Рисунок 5.5 – Вдала реєстрація

При невдалій спробі впливає червоний напис «Ім'я користувача вже існує» (рис. 5.6).



The image shows the same registration form as in Figure 5.5, but with a red error message "Ім'я користувача вже існує" (Username already exists) displayed below the input fields. The button "ЗАРЕЄСТРУВАТИСЯ" remains at the bottom.

Рисунок 5.6 – Невдала реєстрація

Після вдалої реєстрації потрібно увійти в систему. З лівої сторони знаходиться форма входу. Ця форма містить поля: нік користувача та пароль. При невдалому вході впливає червоний напис «Користувач не знайдений» (рис.5.7).

The screenshot shows a login interface with a yellow background. At the top is an orange button labeled "ВХІД". Below it are two white input fields: "Нік користувача" and "Пароль". A red error message "Користувач не знайдений" is displayed below the password field. At the bottom is a dark green button labeled "УВІЙТИ".

Рисунок 5.7 – Невдалий вхід

5.3.2 Сторінка «список задач»

Після входу користувач одразу потрапляє на сторінку списку задач (рис. 5.8).

The screenshot shows the "Weekly Diary" application interface. At the top is a yellow header bar with the text "Weekly Diary" and icons for calendar, list, user, and home. Below the header is a dark green button labeled "список задач". A date range "29.05.2023 - 04.06.2023" is displayed. The main area contains seven yellow task cards for the days of the week: ПОНЕДІЛОК, ВІВТОРОК, СЕРЕДА, ЧЕТВЕР, П'ЯТНИЦЯ, СУБОТА, and НЕДІЛЯ. To the right of these cards is a form to add a task, labeled "ДОДАТИ ЗАДАЧУ". The form includes a text input for "Ціль на день...", a date input for "НА ЯКУ ДАТУ ЗАДАЧА:" with a calendar icon, and an orange "ВНЕСТИ" button. The footer is a dark orange bar with the text "Weekly Diary", contact information "Кальниченко Ангеліна КНТ-139 @kalnychenkoangel", and the university name "Національний університет 'Запорізька політехніка' сайт університету".

Рисунок 5.8 – Сторінка «список задач»

З правої сторони вгорі є значки для переходу на сторінки: щоденнику, списку задач, профіля користувача та на головну сторінку відповідно.

На сторінці списку задач розташовані 7 блоків, які відповідають за дні тижня, та один блок – додавання задач. Щоб створити пункт списку потрібно в полі «Ціль на день...» написати свою задачу, а нижче відмітити дату, на яку ви хочете записати цю ціль (рис. 5.9).

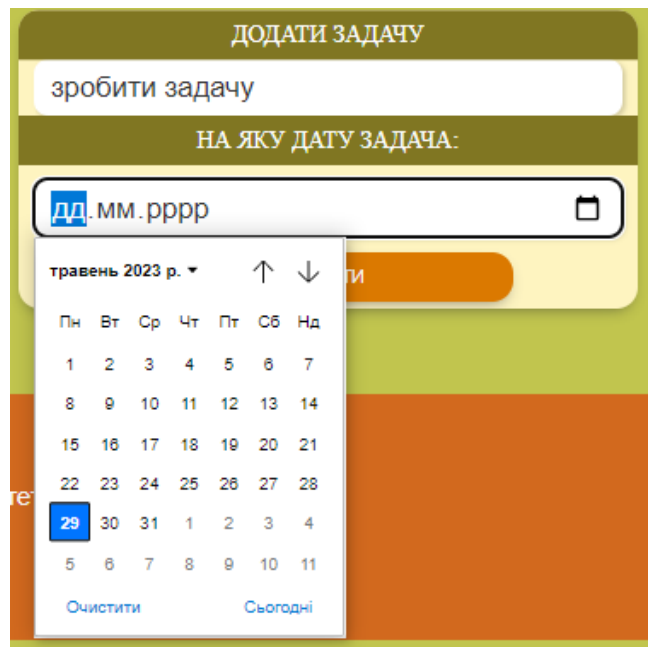


Рисунок 5.9 – Створення задачі

Якщо не вказати дату, то задача буде автоматично додана на сьогоднішній день.

Біля створеної задачі є три значка, які відповідають за виконання, редагування та видалення завдання. При натисканні виконання завдання, галочка перетворюється на хрестик (рис. 5.10).

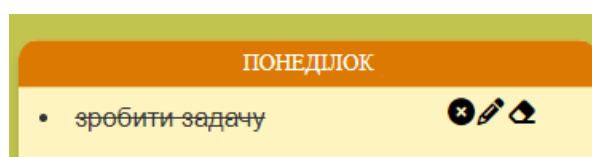


Рисунок 5.10 – Виконана задача

При натисканні редагування (олівця), впливає нове поле для редагування (рис. 5.11).

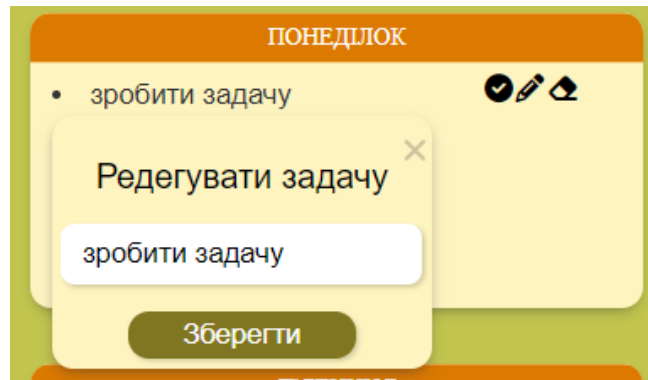


Рисунок 5.11 – Редагування задачі

На сторінці є фільтрація по тижням. Зверху вказується тиждень, який наразі вказаний на сторінці (рис. 5.12).

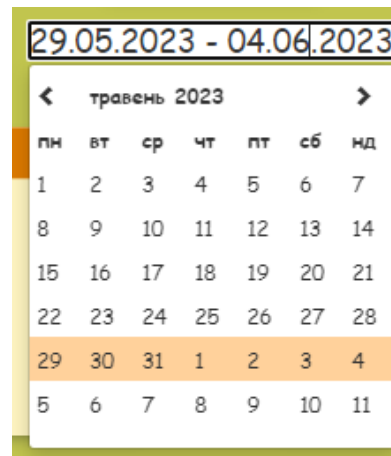


Рисунок 5.12 – Пошук по тижню

5.3.3 Сторінка «щоденник»

Ця сторінка представляє собою тижневий щоденник. В ньому знаходяться 7 блоків, які розташовані вертикально в один ряд і відповідають дням тижня (рис. 5.13).

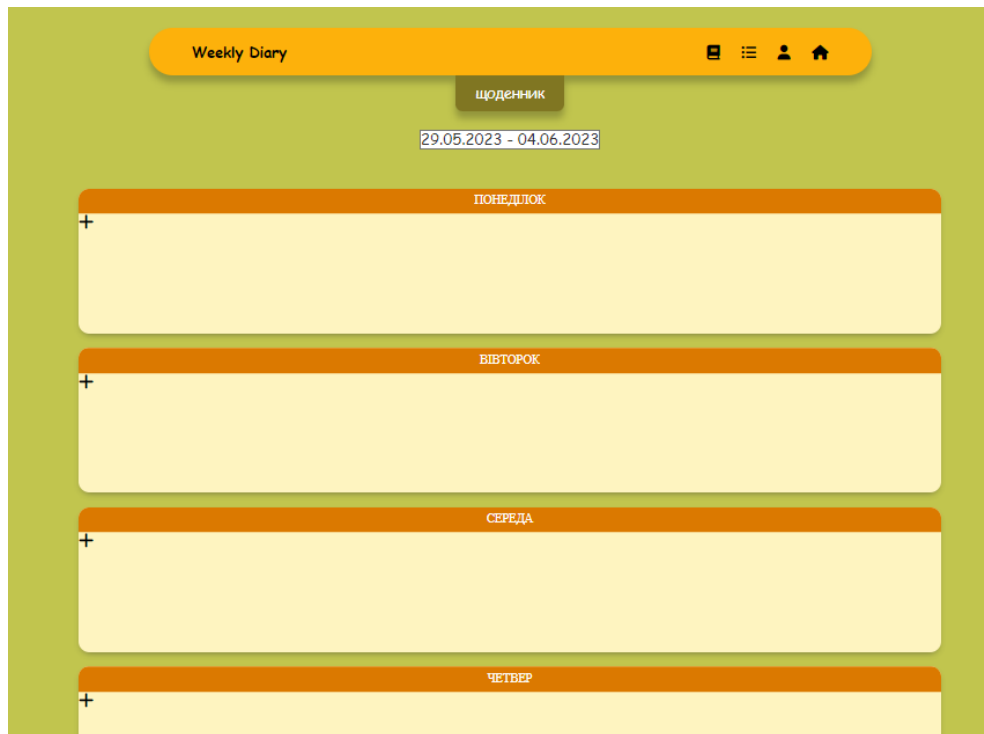


Рисунок 5.13 – Сторінка «щоденник»

У кожного блока у верхньому лівому кутку є плюс, який відповідає за створення нового запису. При натисканні плюса, впливає поле, де можна робити запис з різним редагуванням тексту (рис 5.14).

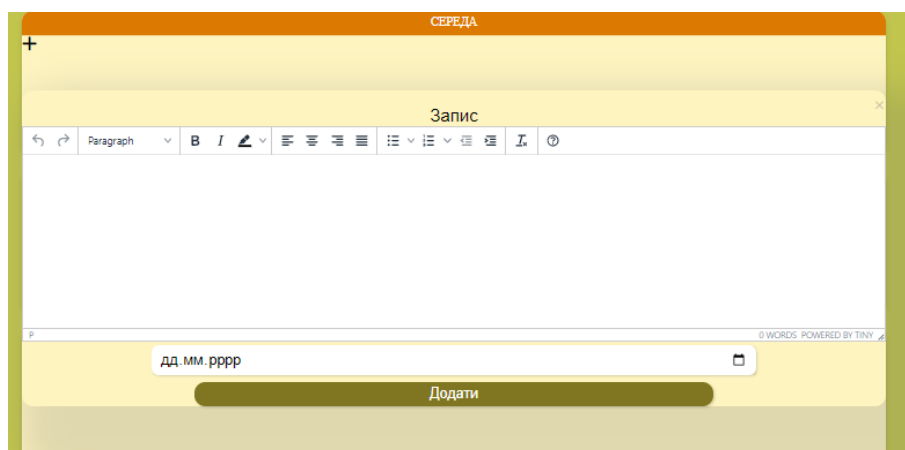


Рисунок 5.14 – Створення запису

Після створення запису є можливість зберегти його на будь яку дату, якщо дату не обирати, то запис збережеться автоматично на той день, де був натиснутий плюс.

Біля створеного запису є дві іконки, які відповідають за редагування та віддалення запису. При натисканні редагування, з'являється поле, де можна змінити допис (рис 5.15).

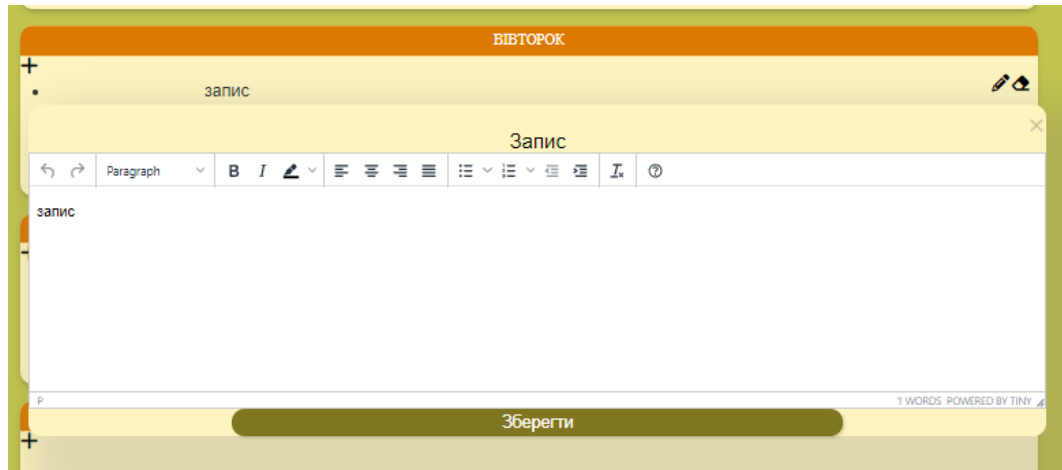


Рисунок 5.15 – Редагування запису

На сторінці є також фільтрація по тижням, як і у списка задач. Зверху вказується тиждень, який наразі показаний на сторінці.

5.3.4 Сторінка «профіль»

Сторінка профіля містить нік та ім'я користувача та кнопку виходу з акаунту користувача (рис. 5.16).



Рисунок 5.16 – Сторінка «профіль»

5.4 Висновки за розділом 5

В розділі було описано процес взаємодії користувача з онлайн щоденником зі списком задач, структуру сторінок та користувацькі дії з сайтом.

Також було розглянуто процес отримання повідомлень для користувача під час експлуатації програми.

ВИСНОВКИ

У результаті виконання дипломної кваліфікаційної роботи детально розглядалися техніки створення онлайн-щоденників і списків завдань, аналізувалися сильні та слабкі сторони існуючих рішень. На основі проведеного аналізу визначено вимоги до інтерфейсу та функціональних характеристик розробленого онлайн-щоденника зі списком завдань.

Для розробки проєкту HTML використовувався для створення структури та позиціонування елементів, CSS для стилізації сайту, JavaScript для забезпечення динамічних функцій і взаємодії з користувачем, а PHP для обробки та генерації динамічного вмісту на рівні сервера. Дані зберігались в базі даних MySQL, що дозволило зберігати дані локально та отримувати доступ під час наступних сеансів програми. Було виконано реалізацію та тестування програмного забезпечення.

Розроблений сайт дозволяє створювати, редагувати та видаляти пункти в списку задач та записи в щоденнику.

Розробленою програмною реалізацією може користуватись будь-яка людина яка потребує впорядкування своїх планів, цілей, думок тощо.

Проведені випробування показують, що створена програмна реалізація відповідає всім вимогам потенційних користувачів та являється економічно доцільною для впровадження.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Tatum M. What is a Diary? [Electronic resource] / Malcolm Tatum // Language Humanities. – Access mode: <https://www.languagehumanities.org/what-is-a-diary.htm>.
2. Holloway S. Diaries Versus Journals: What's the Difference? [Electronic resource] / Sadie Holloway // FeltMagnet. – Access mode: <https://feltmagnet.com/drawing/Diary-vs-Journal-Whats-the-Difference-Between-a-Diary-and-a-Journal>.
3. Nielsen J. Enhancing the explanatory power of usability heuristics / J. Nielsen. – Boston, MA, April 24-28, 1994. – С. 152-158.
4. 10 UI Design Patterns You Should Be Paying Attention To — Smashing Magazine. (2009, June 23). Smashing Magazine. [Electronic resource]. – Access mode: <https://www.smashingmagazine.com/2009/06/10-ui-design-patterns-you-should-be-paying-attention-to/>.
5. Керуйте проєктами своєї команди звідусіль | Trello. Trello. [Electronic resource]. – Access mode: <https://trello.com/uk>.
6. Pot, J. (2022). The 8 best to do list apps in 2023. zapier.com. [Electronic resource]. – Access mode: <https://zapier.com/blog/best-todo-list-apps/> .
7. 16 Best Online To Do Lists & Apps in 2023. Friday.app. [Electronic resource]. – Access mode: <https://friday.app/p/to-do-lists> .
8. Todoist: список справ для організації роботи. Todoist. [Електрон. ресурс]. – Режим доступу: <https://todoist.com/ru>.
9. Sign in - google accounts. Google Keep. [Electronic resource]. – Access mode: <https://keep.google.com>.
10. Manage your team's work, projects, & tasks online. Asana. [Electronic resource]. – Access mode: <https://asana.com/>.
11. Microsoft to do. To Do. [Electronic resource]. – Access mode: https://to-do.microsoft.com/tasks/login?redirectUrl=/en-us?source=product_page&culture=uk-ua&country=UA.

12. Wilkins J. What is a Full Stack Developer? Back End + Front End = Full Stack Engineer [Electronic resource] / Jessica Wilkins // freeCodeCamp.org. – Access mode: <https://www.freecodecamp.org/news/what-is-a-full-stack-developer-back-end-front-end-full-stack-engineer/>.

13. An Introduction to JavaScript. The Modern JavaScript Tutorial. [Electronic resource] // Javascript.info. – Access mode: <https://javascript.info/intro>.

14. Lutkevich B. HTML (Hypertext Markup Language) [Electronic resource] / Ben Lutkevich // TheServerSide. – 2020. – Access mode: <https://www.theserverside.com/definition/HTML-Hypertext-Markup-Language>.

15. Domantas G. What Is CSS and How Does It Work? [Electronic resource] / G. Domantas // Hostinger Tutorials. – 2023. – Access mode: <https://www.hostinger.com/tutorials/what-is-css>.

16. What can PHP do? [Electronic resource] // Manual – Access mode: <https://www.php.net/manual/en/intro-whatcando.php>.

17. Налаштування Open Server [Електрон. ресурс] // ArmedSoft. – 2017. – Режим доступу: <https://armedsoft.com/ua/blog/nalashtuvannya-open-server>.

18. Introduction — PhpMyAdmin 5.1.4 Documentation [Electronic resource] // phpMyAdmin's documentation – Access mode: <https://docs.phpmyadmin.net/en/latest/intro.html>.

19. What is PhpStorm? [Electronic resource] // JavaTPoint – Access mode: <https://www.javatpoint.com/phpstorm>.

ДОДАТОК А
Технічне завдання

A.1 Вступ

Вебзастосунок онлайн щоденник зі списком задач призначений для впорядкування планів, цілей, завдань та думок користувачів.

A.2 Підстава для розроблення

Підставою для розробки є завдання на дипломний кваліфікаційний проєкт на тему: «Розробка вебзастосунку онлайн щоденника зі списком задач», затверджене наказом №87 від 04 квітня 2023 р.

A.3 Призначення розробки

Вебзастосунок онлайн щоденник зі списком задач призначений для надання можливості зручного планування свого дня та думок. Вебзастосунок володіє необхідним функціоналом, що пов'язаний з фільтрацією задач та записів за тижнями.

A.4 Основні вимоги до програмного забезпечення

A.4.1 Вимоги до функціональних характеристик

Розроблюваний вебзастосунок повинен виконувати такі функції:

- можливість створення, редагування та видалення записів та планів в щоденнику та в списку задач;
- календар, що дозволяє відстежувати задачі на визначений тиждень;
- можливість додавати до кожного пункту списку відмітку про виконання, що допомагає відслідковувати ваш прогрес і результати;
- реєстрація або вхід користувача.

A.4.2 Вимоги до надійності

Сервіс повинен розпізнавати ситуації нерозуміння сервісом користувача (користувачем програми), а також обробку збійних ситуацій.

A.4.3 Умови експлуатації

До експлуатаційних вимог на розроблювану програму входять:

- мережа Інтернет;
- встановлений Open Server;
- персональний комп'ютер;
- актуальний браузер.

A.4.4 Вимоги до складу та параметрів технічних засобів

Для експлуатації програмного продукту необхідні такі мінімальні програмно-технічні засоби як: ПК на апаратній платформі x86 під керуванням ОС сімейства WINDOWS (Windows 7/8/8.1/10), обсяг ОЗУ – не менш 256 Мб, обсяг вільного дискового простору – не менш 250 Мб, маніпулятор миша.

A.4.5 Вимоги до маркування та упакування

Вебзастосунок онлайн щоденник зі списком задач може бути упакований у віддалений git-репозиторій. На якій будуть знаходитись всі необхідні файли з кодом та документацією.

A.4.6 Вимоги до транспортування та збереження

Вимоги до транспортування й зберігання аналогічні вимогам, пропонованим системою віддаленого git-репозиторія, на якому зберігається програма.

A.4.7 Вимоги до програмної документації

Вебзастосунок повинен бути з керівництвом користувача, керівництвом програміста.

A.5 Стадії та етапи розробки

A.5.1 Етап розробки аналіз вимог

Це етап на якому збирається вся необхідна інформація для розробки сервісу. Робиться наліз існуючих програмних систем.

A.5.2 Етап розробки проєктування

На етапі проєктування відбувається розробка користувацького інтерфейсу та проєктування схем отримання даних.

A.5.3 Етап розробки програмування і верстка

Це основний етап розробки на якому відбувається розробка компонентів сторінок вебзастосунку, проходять перші тести на реальних даних, код може дороблятися, фіксуються помилки розроблюваного сервісу.

A.5.4 Етап розробки запуск і супровід

Це завершальний етап розробки на якому формується на відправлення замовникові вся документація, у вебзастосунок вноситься необхідний контент, вибирається певна платформа, та весь проєкт відсилається замовникові для тестування.

A.5.5 Етап розробки закриття проєкту

На цьому етапі замовник повідомляє після своїх тестів на своєму обладнанні, що все працює та він згоден підписати акт передачі проєкту та отримує від замовника ПЗ та документацію.

A.6 Порядок контролю та приймання системи

Замовник повинен провести тест на власному технічному забезпеченні. У разі знайдення помилок чи не відповідності ТЗ виконавець зобов'язується виправити всі помилки у певний термін. Якщо тестування було успішним замовник підписує акт передачі проєкту та отримує від замовника ПЗ та документацію.

ДОДАТОК Б
Текст програми

Б.1 Текст програми пов'язаний з обробкою даних

Б.1.1 Текст файлу access-validation.php

```
<?php
    /* перевірка реєстрації користувача */
    global $connection;
    if (session_status() == PHP_SESSION_NONE)
    {
        session_start();
    }
    require_once '../model/bd-connection.php';

    if (!isset($_SESSION['authenticated']) || $_SESSION['authenticated']
    != 'YES')
    {
        header('Location: ../templates/login-
        register.php?login=Помилка!');
    }else
    {
        // Вибір поточного користувача
        $query = "SELECT * FROM users where id = :identify";
        $stmt = $connection->prepare($query);
        $stmt->bindValue(':identify', $_SESSION['id']);
        $stmt->execute();
        $user = $stmt->fetch();
        /* Полегшує залучення поточного користувача на інші сторінки
    */
        $user_id = $user['id'];
    }
?>
```

Б.1.2 Текст файлу add_record_diary.php

```
<?php
    // Підключення до бази даних і отримання поточного користувача
    global $connection, $user_id;
    require_once 'access-validation.php';

    // Зберігання дня тижня, обраного користувачем
    $text = $_POST['diary-text'];
    $date_diary = !empty($_POST['date-diary'])?$_POST['date-
    diary']:'NOW';

    try
    {
        // Введення користувача в базу даних
        $query = '
```

```

        INSERT INTO diary (text, id_user, date_diary)
        VALUES (:text, :id_user, :date_diary)
    ';
    $stmt = $connection->prepare($query);
    $stmt->bindValue(':text', $text);
    $stmt->bindValue(':date_diary', date('Y-m-
d',strtotime($date_diary)));
    $stmt->bindValue(':id_user', $user_id);
    $stmt->execute();
    header('Location: ../templates/diary.php');
} catch(PDOException $e)
{
    $error = 'Error: '.$e->getMessage();
    echo '<pre>';
    print_r($e);
    echo '</pre>';
    header('Location: ../templates/diary.php?add_task='.$error);
} ?>

```

Б.1.3 Текст файлу add_task.php

```

<?php
    // Підключення до бази даних і отримання поточного користувача
    global $connection, $user_id;
    require_once 'access-validation.php';

    // Зберігання в день тижня, обраного користувачем
    $task = $_POST['task-text'];
    $date_task = !empty($_POST['date-task'])?$_POST['date-
task']: 'NOW';

    try
    {
        // Введення користувача в базу даних
        $query = '
            INSERT INTO tasks (task, id_user, date_task)
            VALUES (:task, :id_user, :date_task)
        ';

        $stmt = $connection->prepare($query);

        $stmt->bindValue(':task', $task);
        $stmt->bindValue(':date_task', date('Y-m-
d',strtotime($date_task)));
        $stmt->bindValue(':id_user', $user_id);

        $stmt->execute();

        header('Location: ../templates/to-do_list.php');
    }
}

```

```

}catch(PDOException $e)
{
    $error = 'Error: '.$e->getMessage();

    echo '<pre>';
    print_r($e);
    echo '</pre>';

    header('Location: ../templates/to-
do_list.php?add_task='.$error);
}
?>

```

Б.1.4 Текст файлу delete_diary_record.php

```

<?php
    // Підключення до бази даних і отримання поточного користувача
    global $user_id, $connection;
    require_once 'access-validation.php';

    // Зберігання ідентифікатора завдання, яке буде видалено
    $task_id = $_GET['task_id'];

    try
    {
        // Видалення завдання
        $query = "
            DELETE FROM diary
            WHERE id = :id_task and id_user = :id_user
        ";

        $stmt = $connection->prepare($query);

        $stmt->bindValue(':id_task', $task_id);
        $stmt->bindValue(':id_user', $user_id);

        $stmt->execute();

        header('Location: ../templates/diary.php');
    }catch(PDOException $e)
    {
        $error = 'Error: '.$e->getMessage();

        echo '<pre>';
        print_r($e);
        echo '</pre>';

        header('Location:
../templates/diary.php?delete_task='.$error); } ?>

```

Б.1.5 Текст файлу delete_task.php

```
<?php
    // Підключення до бази даних і отримання поточного користувача
    global $connection, $user_id;
    require_once 'access-validation.php';

    // Зберігання ідентифікатора завдання, яке буде видалено
    $task_id = $_GET['task_id'];

    try
    {
        // Видалення завдання
        $query = "
            DELETE FROM tasks
            WHERE id = :id_task and id_user = :id_user
        ";

        $stmt = $connection->prepare($query);

        $stmt->bindValue(':id_task', $task_id);
        $stmt->bindValue(':id_user', $user_id);

        $stmt->execute();

        header('Location: ../templates/to-do_list.php');
    } catch(PDOException $e)
    {
        $error = 'Error: '.$e->getMessage();

        echo '<pre>';
        print_r($e);
        echo '</pre>';

        header('Location: ../templates/to-
do_list.php?delete_task='.$error);
    }
?>
```

Б.1.6 Текст файлу diary.php

```
<?php
    // Підключення до бази даних і отримання поточного користувача
    global $connection, $user_id;

    require_once 'access-validation.php';
```

```

$diaryData=
isset($_GET['firstDate'])?$_GET['firstDate']:(isset($_SESSION['dateDiary'])?$_SESSION['dateDiary']:date('d.m.Y'));
$_SESSION['dateDiary']=$diaryData;

$day = date('w',strtotime($diaryData));
$week_start = date('Y-m-d', strtotime($diaryData.'-'. $day.' days'));
$week_end = date('Y-m-d', strtotime($diaryData.'+'.(6-$day).' days'));

// Зберігання всіх завдань від поточного користувача
$query = "
    SELECT * FROM diary where id_user = :user_id AND date_diary
    BETWEEN :week_start AND :week_end
    ";

$stmt = $connection->prepare($query);

$stmt->bindValue(':user_id', $user_id);
$stmt->bindValue(':week_start', $week_start);
$stmt->bindValue(':week_end', $week_end);
$stmt->execute();

$diary_records = $stmt->fetchAll();

?>

```

Б.1.7 Текст файлу do_task.php

```

<?php
// Підключення до бази даних і отримання поточного користувача
global $connection;
require_once 'access-validation.php';

// Зберігання ідентифікатора та тексту завдання, яке потрібно
редагувати
$task_id = $_GET['task_id'];
$checked = $_GET['check'];
try
{
    // Редагування завдання
    $query = "
        UPDATE tasks
        SET checked= :checked
        WHERE id = :id_task
    ";

    $stmt = $connection->prepare($query);

```

```

$stmt->bindValue(':id_task', $task_id);
$stmt->bindValue(':checked', $checked);

$stmt->execute();

header('Location: ../templates/to-do_list.php');
} catch(PDOException $e)
{
    $error = 'Error: '.$e->getMessage();

    echo '<pre>';
    print_r($e);
    echo '</pre>';

    header('Location: ../templates/to-
do_list.php?delete_task='.$error);
}
?>

```

Б.1.8 Текст файлу edit_diary.php

```

<?php
// Підключення до бази даних і отримання поточного користувача
global $connection, $user_id;
require_once 'access-validation.php';

// Зберігання ідентифікатора та тексту завдання, яке потрібно
редагувати
$task_id = $_POST['task_id'];
$rewritten_task = $_POST['diary-text'];

try
{
    // Редагування завдання
    $query = "
        UPDATE  diary
        SET text= :task
        WHERE id = :id_task and id_user = :id_user
    ";

    $stmt = $connection->prepare($query);

    $stmt->bindValue(':id_task', $task_id);
    $stmt->bindValue(':id_user', $user_id);
    $stmt->bindValue(':task', $rewritten_task);

    $stmt->execute();
}

```

```

        header('Location: ../templates/diary.php');
    }catch(PDOException $e)
    {
        $error = 'Error: '.$e->getMessage();

        echo '<pre>';
        print_r($e);
        echo '</pre>';

        header('Location:
../templates/diary.php?delete_diary='.$error);
    }
?>

```

Б.1.9 Текст файлу edit_task.php

```

<?php
    // Підключення до бази даних і отримання поточного користувача
    global $connection, $user_id;
    require_once 'access-validation.php';

    // Зберігання ідентифікатора та тексту завдання, яке потрібно
    редагувати
    $task_id = $_POST['task_id'];
    $rewritten_task = $_POST['task'];

    try
    {
        // Редагування завдання
        $query = "
            UPDATE tasks
            SET task= :task
            WHERE id = :id_task and id_user = :id_user
        ";

        $stmt = $connection->prepare($query);

        $stmt->bindValue(':id_task', $task_id);
        $stmt->bindValue(':id_user', $user_id);
        $stmt->bindValue(':task', $rewritten_task);

        $stmt->execute();

        header('Location: ../templates/to-do_list.php');
    }catch(PDOException $e)
    {
        $error = 'Error: '.$e->getMessage();

        echo '<pre>';
    }

```

```

        print_r($e);
        echo '</pre>';

        header('Location: ../templates/to-
do_list.php?delete_task='.$error);
    }
?>

```

Б.1.10 Текст файлу login-validation.php

```

<?php
global $connection, $user_authenticated;
session_start();

// Підключення до бази даних
require_once '../model/bd-connection.php';

// Зберігання даних користувачів
$username_user = $_POST['username'];
$password_user = $_POST['password'];

// Змінні для контролю автентифікації користувача
$user_authenticated = false;
$user_id = null;

// Вибір поточного користувача
$query = "
    SELECT * FROM users where username = :username
    AND password = :password
";

$stmt = $connection->prepare($query);

$stmt->bindValue(':username', $username_user);
$stmt->bindValue(':password', $password_user);

$stmt->execute();

$user = $stmt->fetch();

if (!empty($user))
{
    $user_authenticated = true;
    $user_id = $user['id'];
}else{
    header('Location: ../templates/login-register.php?login=Не
знайден користувач');
}

```

```

if($user_authenticated)
{
    $_SESSION["authenticated"] = "YES";
    $_SESSION["id"] = $user_id;
    header('Location: ../templates/to-do_list.php');
}else
{
    $_SESSION['authenticated'] = 'NO';
    header('Location: ../templates/login-
register.php?login=Користувач не знайдений');
}
?>

```

Б.1.11 Текст файлу register-validation.php

```

<?php
global $connection;
error_reporting(E_ALL);
ini_set('display_errors', '1');
// Підключення до бази даних
require_once '../model/bd-connection.php';

// Змінні для контролю автентифікації користувача
$user_authenticated = false;
$user_id = null;

// Зберігання даних користувачів
$username_user = $_POST['username'] ;
$name_user = $_POST['full-name'] ;
$password_user = $_POST['password'];

try {
    // Перевірка чи користувач з введеним іменем вже існує в базі
    даних
    $query = "
        SELECT * FROM users WHERE username = :username
    ";
    $stmt = $connection->prepare($query);
    $stmt->bindValue(':username', $username_user);
    $stmt->execute();
    $existingUser = $stmt->fetch();

    if ($existingUser) {
        header('Location: ../templates/login-
register.php?register=Ім`я користувача вже існує');
        exit;
    }

    // Вставка нового користувача в базу даних

```

```

$query = '
    INSERT INTO users (username, full_name, password)
    VALUES (:username, :full_name, :password)
';
$stmt = $connection->prepare($query);
$stmt->bindValue(':username', $username_user);
$stmt->bindValue(':full_name', $name_user);
$stmt->bindValue(':password', $password_user);
$stmt->execute();

// Перенаправлення після успішної реєстрації
header('Location: ../templates/login-
register.php?register=Реєстрація успішна');
} catch (PDOException $e) {
    $error = 'Error: ' . $e->getMessage();
    echo '<pre>';
    print_r($e);
    echo '</pre>';
    header('Location: ../templates/login-
register.php?register.php?register=' . $error);
}

echo '<br><br><hr><pre>';
print_r($_FILES);
echo '</pre>';
?>

```

Б.1.12 Текст файлу tasks.php

```

<?php
// Підключення до бази даних і отримання поточного користувача
global $connection, $user_id;

require_once 'access-validation.php';
$firstDate=
isset($_GET['firstDate'])?$_GET['firstDate']:(isset($_SESSION['planerDate'])?$_SESSION['planerDate']:date('d.m.Y'));
$_SESSION['planerDate']=$firstDate;

$day = date('w',strtotime($firstDate));
$week_start = date('Y-m-d', strtotime($firstDate.'-'. $day.'
days'));
$week_end = date('Y-m-d', strtotime($firstDate.'+'.(6-$day).'
days'));

// Зберігання всіх завдань від поточного користувача
$query = "
    SELECT * FROM tasks where id_user = :user_id AND date_task
    BETWEEN :week_start AND :week_end

```

```

";

$stmt = $connection->prepare($query);

$stmt->bindValue(':user_id', $user_id);
$stmt->bindValue(':week_start', $week_start);
$stmt->bindValue(':week_end', $week_end);
$stmt->execute();

$tasks = $stmt->fetchAll();

?>

```

Б.2 Текст програми пов'язаний з даними

Б.2.1 Текст файлу bd-connection.php

```

<?php
// Creating variables to connect with the database
$dsn = 'mysql:host=127.0.0.1;dbname=diary';
$user = 'diary';
$password = '1qazxsw2';

$connection = new PDO($dsn, $user, $password);
define ('SITE_ROOT', realpath(dirname(__FILE__)));

?>

```

Б.2.2 Текст файлу diary.sql

```

-- phpMyAdmin SQL Dump
-- version 5.1.1deb5ubuntu1
-- https://www.phpmyadmin.net/
--
-- Хост: localhost:3306
-- Час створення: Трв 25 2023 р., 20:22
-- Версія сервера: 8.0.33-0ubuntu0.22.04.2
-- Версія PHP: 8.1.18

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;

```

```

/*!40101 SET NAMES utf8mb4 */;

--
-- База даних: `diary`
--

--
-- Структура таблиці `diary`
--

CREATE TABLE `diary` (
  `id` int NOT NULL,
  `id_user` int NOT NULL,
  `text` text CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT
NULL,
  `date_diary` date NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

--
-- Структура таблиці `tasks`
--

CREATE TABLE `tasks` (
  `id` int NOT NULL,
  `id_user` int NOT NULL,
  `task` text COLLATE utf8mb4_unicode_ci NOT NULL,
  `date_task` date NOT NULL,
  `checked` tinyint NOT NULL DEFAULT '0'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

--
-- Структура таблиці `users`
--

CREATE TABLE `users` (
  `id` int NOT NULL,
  `username` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `full_name` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `password` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

--
-- Індекси збережених таблиць
--

```

```

--
-- Індокси таблиці `diary`
--
ALTER TABLE `diary`
  ADD PRIMARY KEY (`id`);

--
-- Індокси таблиці `tasks`
--
ALTER TABLE `tasks`
  ADD PRIMARY KEY (`id`);

--
-- Індокси таблиці `users`
--
ALTER TABLE `users`
  ADD PRIMARY KEY (`id`);

--
-- AUTO_INCREMENT для збережених таблиць
--

--
-- AUTO_INCREMENT для таблиці `diary`
--
ALTER TABLE `diary`
  MODIFY `id` int NOT NULL AUTO_INCREMENT;

--
-- AUTO_INCREMENT для таблиці `tasks`
--
ALTER TABLE `tasks`
  MODIFY `id` int NOT NULL AUTO_INCREMENT;

--
-- AUTO_INCREMENT для таблиці `users`
--
ALTER TABLE `users`
  MODIFY `id` int NOT NULL AUTO_INCREMENT;
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

```

Б.3 Текст програми пов'язаний з роботою щоденника

Б.3.1 Текст файлу diary.php

```

<?php
global $tasks, $diary_records;
include_once 'navigation.php';

// Сторінка лише для користувачів
require_once '../controller/access-validation.php';
?>
<!DOCTYPE html>
<html lang="uk">

<body>

<div class="marker">щоденник</div>
<div class="marker-week marker-week-hover">
    <input type='text' id='weeklyDatePicker' placeholder="Виберіть
тиждень" />
</div>

<!-- Створення нового запису -->
<div id="diary">

    <?php
    //Переглядайте дні тижня, навіть якщо в цей день у користувача ще
    немає завдань
    include '../controller/diary.php';
    $week_days = ['ПОНЕДІЛОК', 'ВІВТОРОК', 'СЕРЕДА', 'ЧЕТВЕР',
"П'ЯТНИЦЯ", 'СУБОТА', 'НЕДІЛЯ'];
    foreach ($week_days as $index => $day)
    {
        ?>
        <div class="field add-days">

            <h4><?php echo $day; ?></h4>

            <!-- Додана кнопка створення запису -->
            <a href="#popup-create" class="create_btn">
                <i class="fa fa-plus"></i></a>

            <ul>
                <?php
                // Цикл навколо записів користувача
                foreach ($diary_records as $record)
                {
                    // Організація по днях тижня
                    if (intval(date('w',
strtotime($record['date_diary']))) == $index + 1)

```

```

        {
            ?>
            <div style="clear:both"></div>
            <!-- Видалення задачі -->
            <a
href=" ../controller/delete_diary_record.php?task_id=<?php echo
$record['id'] ?>">
                <i class="fas fa-eraser"></i></a>
            <!-- Кнопка редагування задачі -->
            <a href="#popup?nmr_task=<?php echo
$record['id'] ?>" class="edit_btn">
                <i class="fa fa-pencil"></i></a>
            <!-- Текст -->
            <li><?php echo $record['text']; ?></li>

            <!-- Редагування задачі -->
            <div class="edit-task"
id="popup?nmr_task=<?php echo $record['id'] ?>">

                <!-- кнопка закрити -->
                <a href="#" class="close"><i class="fa fa-
close"></i></a>

                <div style="clear:both"></div>

                <!-- вікно редагування -->
                <form
action=" ../controller/edit_diary.php" method="POST">
                    <h3 class="rewrite">Запис</h3>
                    <textarea name="diary-text" rows="10"
cols="80" class="texts" id="tiny-mce"><?php echo $record['text']
?></textarea>

                    <input type="hidden" name="task_id"
value="<?php echo $record['id'] ?>">

                    <button type="submit"
class="rewrite">Зберегти</button>
                </form>
            </div>
            <?php
        }
    } ?>
</ul>
</div>
<?php
}
?>

<div class="edit-task1" id="popup-create">
    <!-- кнопка закрити -->
    <a href="#" class="close"><i class="fa fa-close"></i></a>
    <div style="clear:both"></div>

```

```

<h3 class="rewrite">Запис</h3>

<!-- вікно редагування -->
<form action="../../controller/add_record_diary.php"
method="POST">
    <textarea type="text" name="diary-text" rows="10"
cols="80" class="texts" id="tiny-mce"></textarea>
    <input type="date" name="date-diary" id="date-diary">

    <button type="submit" class="rewrite">Додати</button>
</form>
</div>

</div>

<?php
include_once 'footer.php';
?>
</body>
</html>

```

Б.3.2 Текст файлу footer.php

```

<footer>
    <div class="name">
        <h3>Weekly Diary</h3>
    </div>

    <div class="contact">
        <h4>Кальниченко Ангеліна КНТ-139</h4>
        <a href="https://www.instagram.com/kalnynchenkoangel/"
target="_blank"><h4>@kalnynchenkoangel</h4></a>
    </div>

    <div class="contact">
        <h4>Національний університет "Запорізька політехніка"</h4>
        <a href="https://zp.edu.ua/" target="_blank"><h4>сайт
університету </h4></a>
    </div>
</footer>

<!-- посилання на зовнішні JavaScript-бібліотеки, які використовуються
в проєкті -->

<!--jQuery:спрощує взаємодію з HTML-документами, обробку подій,
анімацію та роботу з AJAX-->
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js
"></script>

```

```

<!--Moment.js: для роботи з датами та часом-->
<script
src="https://cdn.jsdelivr.net/momentjs/2.10.3/moment.min.js"></script>
<!--Bootstrap Datetimepicker: надає можливість вибору дати та часу з
допомогою інтерактивного календаря та годинника-->
<script
src="https://cdnjs.cloudflare.com/ajax/libs/moment.js/2.10.3/moment-
with-locales.js" ></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/bootstrap-
datetimepicker/4.17.37/js/bootstrap-datetimepicker.min.js"
integrity="sha512-
Y+0b10RbVUTf3Mi0EgJue0FoheNzentTMMIE20reNbqnUPNbQj8zmjK3fs5D2WhQeGWIem
2G2UkKjAL/bJ/UXQ==" crossorigin="anonymous" referrerpolicy="no-
referrer"></script>
<!--TinyMCE: дозволяє легко інтегрувати розширений редактор у вашу
вебсторінку або додаток-->
<script
src="https://cdn.tiny.cloud/1/qagffr3pkuv17a8on1afax661irst1hbr4e6tbv8
88sz91jc/tinymce/5-stable/tinymce.min.js"></script>
<script
src="https://cdn.tiny.cloud/1/{YOUR_API_KEY}/tinymce/5/tinymce.min.js"
referrerpolicy="origin"></script>

<script>
    tinymce.init({
        selector: 'textarea.texts',
        height: 300,
        menubar: false,
        plugins: [
            'advlist autolink lists link image charmap print preview
anchor',
            'searchreplace visualblocks code fullscreen',
            'insertdatetime media table paste code help wordcount'
        ],
        toolbar: 'undo redo | formatselect | bold italic backcolor | \
            alignleft aligncenter alignright alignjustify | \
            bullist numlist outdent indent | removeformat |
help',
        content_css: 'path/to/your/custom.css'
    });
</script>

<script>

    $(document).ready(function ()
    {
        var firstDate = <?php echo
json_encode(isset($_SESSION['planerDate'])?$_SESSION['planerDate']: '',
JSON_HEX_TAG); ?>;

```

```

        var diaryData = <?php echo
json_encode(isset($_SESSION['dateDiary'])?$_SESSION['dateDiary']:'',
JSON_HEX_TAG); ?>;
        var dateNow = new Date();
        $("#weeklyDatePicker").datetimepicker(
            {
                format: 'DD.MM.YYYY',
                defaultDate: dateNow,
                locale: moment.locale('uk')
            });

        $("#weeklyDatePickerDiary").datetimepicker(
            {
                format: 'DD.MM.YYYY',
                defaultDate: dateNow,
                locale: moment.locale('uk')
            });

        var value = firstDate ? firstDate :
moment().format("DD.MM.YYYY");
        var firstDateStart = moment(value,
"DD.MM.YYYY").locale('uk').day(1).format("DD.MM.YYYY");
        var lastDate = moment(value,
"DD.MM.YYYY").locale('uk').day(7).format("DD.MM.YYYY");

        $("#weeklyDatePicker").val(firstDateStart + " - " + lastDate);
        $('#weeklyDatePicker').on('dp.change', function (e)
        {
            var value = $("#weeklyDatePicker").val();
            var firstDate = moment(value,
"DD.MM.YYYY").locale('uk').day(1).format("DD.MM.YYYY");
            var lastDate = moment(value,
"DD.MM.YYYY").locale('uk').day(7).format("DD.MM.YYYY");
            $("#weeklyDatePicker").val(firstDate + " - " + lastDate);
            console.log(value, firstDate + " - " + lastDate);
            if (firstDateStart != firstDate)
                window.location = window.location.origin +
window.location.pathname + '?firstDate=' + firstDate;
        });

        var value1 = diaryData ? diaryData :
moment().format("DD.MM.YYYY");

        var firstDateStartDiary = moment(value1,
"DD.MM.YYYY").locale('uk').day(1).format("DD.MM.YYYY");
        var lastDateDiary = moment(value1,
"DD.MM.YYYY").locale('uk').day(7).format("DD.MM.YYYY");

        ($("#weeklyDatePickerDiary").val(firstDateStartDiary + " - " +
lastDateDiary);
        $('#weeklyDatePickerDiary').on('dp.change', function (e)

```

```

    {
        var value = $("#weeklyDatePickerDiary").val();
        var firstDate = moment(value,
"DD.MM.YYYY").locale('uk').day(1).format("DD.MM.YYYY");
        var lastDate = moment(value,
"DD.MM.YYYY").locale('uk').day(7).format("DD.MM.YYYY");
        $("#weeklyDatePickerDiary").val(firstDate + " - " +
lastDate);
        console.log(value, firstDate + " - " + lastDate);
        if (firstDateStartDiary != firstDate)
            window.location = window.location.origin +
window.location.pathname + '?diaryData=' + firstDate;
    });

    tinymce.init({
        selector: 'textarea#tiny-mce',
        height: 500,
        menubar: false,
        plugins: [
            'advlist autolink lists link image charmap print
preview anchor',
            'searchreplace visualblocks code fullscreen',
            'insertdatetime media table paste code help wordcount'
        ],
        toolbar: 'undo redo | formatselect | ' +
            'bold italic backcolor | alignleft aligncenter ' +
            'alignright alignjustify | bullist numlist outdent
indent | ' +
            'removeformat | help',
        content_style: 'body { font-family:Helvetica,Arial,sans-
serif; font-size:14px }'
    });
});
</script>

```

Б.3.3 Текст файлу login-register.php

```

<?php
require_once 'navigation.php';
?>
<!DOCTYPE html>
<html lang="uk">

<body>
<div class="marker">пофіль</div>

<!-- форма входу -->
<div id="login-register">
    <div class="field">

```

```

        <h3 style="width:100px">Вхід</h3>
        <form action="../../controller/login-validation.php"
method="POST">
            <input type="text" class="form-general" name="username"
placeholder="Нік користувача"required >

            <input type="password" class="form-general"
name="password" placeholder="Пароль"required >

            <!-- не правильний вхід -->
            <?php
            if (isset($_GET['login'])) {
                if ($_GET['login'] == 'user_not_logged_in'){
                    echo '<span style="color:red">Не вдалось
увійти<br></span>';
                }else {
                    echo '<span style="color:red">'. $_GET['login']
.'<br></span>';
                }
            }
            ?>
            <input type="submit" class="buttons" value="УВІЙТИ">
        </form>
    </div>

    <!-- форма реєстрації -->
    <div class="field">
        <h3>реєстрація</h3>
        <form action="../../controller/register-validation.php"
enctype="multipart/form-data" method="POST">
            <input type="text" class="form-general" name="username"
placeholder="Нік користувача" required>

            <input type="text" class="form-general" name="full-name"
placeholder="Ім'я" required>

            <input type="password" class="form-general"
name="password" placeholder="Пароль" required>

            <!-- не правильна форма реєстрації -->
            <?php
            if (isset($_GET['register'])) {
                echo '<span style="color:red">'. $_GET['register']
.'<br></span>';
            }
            ?>
            <input type="submit" class="buttons"
value="ЗАРЕЄСТРУВАТИСЯ">
        </form>
    </div>
</div>

```

```
<?php
include_once 'footer.php';
?>
</body>
</html>
```

Б.3.4 Текст файла logout.php

```
<?php
    session_start();
    session_destroy();
    header('Location: login-register.php');
?>
```

Б.3.5 Текст файла navigation.php

```
<?php
error_reporting(E_ALL);
ini_set('display_errors', '1');
    session_start();
?>
<!DOCTYPE html>
<html lang="uk">
    <head>
        <meta charset="UTF-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <meta name="viewport" content="width=device-width, initial-
scale=1.0">

        <!-- Main CSS -->
        <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.mi
n.css">
        <link rel="stylesheet" href="../static/css/style.css">

        <!-- Normalize CSS -->
        <link rel="stylesheet" href="../static/css/normalize.css">

        <!--[if lt IE 9]>
        <script
src="https://cdnjs.cloudflare.com/ajax/libs/html5shiv/3.7.3/html5shiv-
printshiv.min.js"></script>
        <![endif]-->

        <!-- ICONS -->
```

```

        <script src="https://kit.fontawesome.com/cb18e7956b.js"
crossorigin="anonymous"></script>

        <!-- FONTS -->
        <link rel="preconnect" href="https://fonts.googleapis.com">
        <link rel="preconnect" href="https://fonts.gstatic.com"
crossorigin>
        <link
href="https://fonts.googleapis.com/css2?family=Atma:wght@400;600&displ
ay=swap" rel="stylesheet">
    </head>

    <body>
        <nav> <!-- навігація в бачі -->

            <a href="../index.php"><span class="name">Weekly
Diary</span></a>

            <a href="../index.php" title="Головна сторінка"
class="icons"><i class="fas fa-home"></i></a>

            <?php
                if (!isset($_SESSION['authenticated']) ||
$_SESSION['authenticated'] != 'YES')
                {
                    ?>
                        <a href="../templates/login-register.php"
title="Головна сторінка" class="icons"><i class="fas fa-user"></i></a>
                    <?php
                        }else
                        {
                            ?>
                                <a href="../templates/user-page.php"
title="Профіль" class="icons"><i class="fas fa-user"></i></a>
                            <?php
                                }
                            ?>

                                <a href="../templates/to-do_list.php" title="Список задач"
class="icons"><i class="fas fa-list-ul"></i></a>

                                <a href="../templates/diary.php" title="Щоденник"
class="icons"><i class="fas fa-book"></i></a>
                            </nav>
                        </body>
                    </html>

```



```

        <a
href="../../../controller/delete_task.php?task_id=<?php echo $task['id']
?>">
        <i class="fa fa-eraser" aria-
hidden="true"></i></a>

        <!-- кнопка змінити завдання -->
        <a href="#popup?nmr_task=<?php echo
$task['id'] ?>" class="edit_btn">
            <i class="fa fa-pencil"></i></a>
        <div class="edit-tasks"
id="popup?nmr_task=<?php echo $task['id'] ?>">

            <!-- кнопка закрити -->
            <a href="#" class="close"><i class="fa
fa-close"></i></a>

            <div style="clear:both"></div>

            <!-- вікно редагування -->
            <form
action="../../../controller/edit_task.php" method="POST">
                <div class="">
                    <h3 class="rewrite">Редагувати
задачу</h3>
                    <input type="text"
class="rewrite" value="<?php echo $task['task'] ?>"
                        name="task" required>
                    <input type="hidden"
name="task_id" value="<?php echo $task['id'] ?>">
                </div>
                <div class="">
                    <div class=""></div>
                    <button type="submit"
class="rewrite">Зберегти</button>
                </div>
            </form>
        </div>

        <?php

        //перевірка виконання завдання
        if ($task['checked'])
        {
            ?>
            <a
href="../../../controller/do_task.php?task_id=<?php echo $task['id']
?>&check=0">
                <i class="fas fa-circle-
xmark"></i></a>

            <?php
        } else

```

```

        {
            ?>
            <a
href="../controller/do_task.php?task_id=<?php echo $task['id']
?>&check=1">
                <i class="fas fa-circle-
check"></i></a>
            <?php
            }
            ?>
        <?php
        }
    } ?>
</ul>
</div>
<?php
}
?>

<!-- поле для додавання завдань -->
<div class="field add-task">
    <h4>Додати задачу</h4>

    <form action="../controller/add_task.php" method="POST">
        <input type="text" name="task-text" class="texts"
placeholder="Ціль на день..." maxlength="50" required>
        <h4>На яку дату задача:</h4>
        <input type="date" name="date-task" id="date-task">

        <button type="submit">Внести</button>
    </form>
</div>
</div>

<?php
include_once 'footer.php';
?>

</body>
</html>

```

Б.3.7 Текст файлу user-page.php

```

<?php
global $tasks;
include_once 'navigation.php';

// Сторінка лише для користувачів
require_once '../controller/access-validation.php';

```

```

?>
<!DOCTYPE html>
<html lang="uk">

<body>
<div class="marker">список задач</div>
<div class="marker-week marker-week-hover">
    <input type='text' id='weeklyDatePicker' placeholder="Виберіть
тиждень" />
</div>

<div id="todolist">
    <?php
    //Переглядайте дні тижня, навіть якщо в цей день у користувача ще
немає завдань
    include '../controller/tasks.php';
    $week_days = ['ПОНЕДІЛОК', 'ВІВТОРОК', 'СЕРЕДА', 'ЧЕТВЕР',
"П'ЯТНИЦЯ", 'СУБОТА', 'НЕДІЛЯ'];
    foreach ($week_days as $index => $day)
    {
        ?>

        <div class="field">
            <h4><?php echo $day; ?></h4>
            <ul>
                <?php
                // Цикл навколо завдань користувача
                foreach ($tasks as $task)
                {
                    // Організація по днях тижня
                    if (intval(date('w',
strtotime($task['date_task']))) == $index + 1)
                    {
                        ?>
                        <div style="clear:both"></div>

                        <!-- Зміст завдання -->
                        <li> <?php echo !$task['checked']?
$task['task']:"<s>".$task['task']."</s>" ?>

                        <!-- кнопка видалення -->
                        <a
href="../controller/delete_task.php?task_id=<?php echo $task['id']
?>">

                            <i class="fa fa-eraser" aria-
hidden="true"></i></a>

                        <!-- кнопка змінити завдання -->
                        <a href="#popup?nmr_task=<?php echo
$task['id'] ?>" class="edit_btn">
                            <i class="fa fa-pencil"></i></a>

```

```

        <div class="edit-tasks"
id="popup?nmr_task=<?php echo $task['id'] ?>">

                <!-- кнопка закрити -->
                <a href="#" class="close"><i class="fa
fa-close"></i></a>

                <div style="clear:both"></div>

                <!-- вікно редагування -->
                <form
action="../controller/edit_task.php" method="POST">
                        <div class="">
                                <h3 class="rewrite">Редегувати
задачу</h3>
                                <input type="text"
class="rewrite" value="<?php echo $task['task'] ?>"
                                name="task" required>
                                <input type="hidden"
name="task_id" value="<?php echo $task['id'] ?>">
                                </div>
                                <div class="">
                                        <div class=""></div>
                                        <button type="submit"
class="rewrite">Зберегти</button>
                                </div>
                        </form>
                </div>

                <?php

                //перевірка виконання завдання
                if ($task['checked'])
                {
                        ?>
                        <a
href="../controller/do_task.php?task_id=<?php echo $task['id']
?>&check=0">
                                <i class="fas fa-circle-
xmark"></i></a>

                                <?php
                                } else
                                {
                                        ?>
                                        <a
href="../controller/do_task.php?task_id=<?php echo $task['id']
?>&check=1">
                                                <i class="fas fa-circle-
check"></i></a>

                                <?php
                                }
                                ?>

```

```

                                <?php
                                }
                            } ?>
                        </ul>
                    </div>
                <?php
            }
        ?>

        <!-- поле для додавання завдань -->
        <div class="field add-task">
            <h4>Додати задачу</h4>

            <form action="../../controller/add_task.php" method="POST">
                <input type="text" name="task-text" class="texts"
placeholder="Ціль на день..." maxlength="50" required>
                <h4>На яку дату задача:</h4>
                <input type="date" name="date-task" id="date-task">

                <button type="submit">Внести</button>
            </form>
        </div>
    </div>

    <?php
    include_once 'footer.php';
    ?>

</body>
</html>

```

Б.3.8 Текст файлу index.php

```

<?php
    session_start();
?>
<!DOCTYPE html>
<html>
    <head>
        <!-- Встановлюємо кодування символів на UTF-8 -->
        <meta charset="UTF-8">
        <!-- Застосовуємо режим сумісності з Internet Explorer -->
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <!-- Налаштовуємо масштаб та розмір відображення на різних
пристроях -->
        <meta name="viewport" content="width=device-width, initial-
scale=1.0">

        <!-- Main CSS -->

```

```

        <!-- Підключаємо основний CSS-файл для стилізації домашньої
сторінки -->
        <link rel="stylesheet" href="./static/css/homepage.css">

        <!-- Normalize CSS -->
        <!-- Підключаємо CSS-файл normalize для уніфікації стилів між
різними браузерами -->
        <link rel="stylesheet" href="./static/css/normalize.css">

        <!-- ICONS -->
        <!-- Підключаємо набір іконок Font Awesome для використання
іконок у вебсторінці -->
        <script src="https://kit.fontawesome.com/df07cb0f03.js"
crossorigin="anonymous"></script>

        <!-- FONTS -->
        <!-- Встановлюємо попереднє з'єднання з Google Fonts -->
        <link rel="preconnect" href="https://fonts.googleapis.com">
        <!-- Встановлюємо попереднє з'єднання з сервером Google Fonts
-->
        <link rel="preconnect" href="https://fonts.gstatic.com"
crossorigin>
        <!-- Підключаємо шрифти з Google Fonts -->
        <link
href="https://fonts.googleapis.com/css2?family=Abhaya+Libre:wght@700;8
00&family=Atma:wght@500;600&family=Lato:wght@300;400;700&display=swap"
rel="stylesheet">

    </head>
    <body>
        <div id="roxo">
            <nav>
                <a href=""><span id="logo">Weekly Diary</span></a>

                <div class="menu">

                    <?php
                        if (!isset($_SESSION['authenticated'])) ||
$_SESSION['authenticated'] != 'YES')
                        {
                            ?>
                            <a href="./templates/login-
register.php">Увійти</a>
                            <?php
                                }else
                                {
                                    ?>
                                    <a
href="./templates/diary.php">Щоденник</a>
                                    <a href="./templates/to-
do_list.php">Список задач</a>

```

```

        <a href="./templates/user-
page.php">Вийти</a>
        <?php
        }
        ?>
    </nav>
</div>
    <div id="intro">
        <div class="texts">
            <span class="title">
                У нинішній час, коли інформаційні технології
                відіграють значну роль у житті кожної людини,
                онлайн щоденник стає дедалі популярним і
                займає важливе місце в нашому житті.
                Онлайн щоденник – це доступний і зручний
                інструмент для планування задач та керування часом.
            </span>
        </div>

        <div class="vector">
            
        </div>
    </div>

    <div id="demonstrative">
        <div class="image">
            
        </div>

        <div class="explanation">
            <h2>
                Вебзастосунок має аутентифікація, щоб збережит
                ващі дані в безпеці.
            </h2>
            <h2>
                Проста система для додавання, редагування та
                видалення записів та завдань.
            </h2>
        </div>
    </div>

    <?php
        include_once './templates/footer.php';
    ?>
</body>
</html>

```

Б.4 Текст програми пов'язаний зі стилізацією щоденника

Б.4.1 Текст файлу homepage.css

```
:root
{
  --body: #c1c54e;
  --nav: #FDB10B;
  --lighter: #fccc64;
  --green: #807622;
  --lighter-yellow: #FDB10B;
  --stronger-green: #104d06;
  --font-title: 'Josefin Sans', serif;
  --font-subtitle: 'Josefin Sans', sans-serif;
}
*
{
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}
body
{
  background-color: #c1c54e;
  font-family: var(--font-subtitle);
}
a
{
  text-decoration: none;
}
#roxo
{
  width: 80%;
  height: 65px;
  margin: 0 auto;
  margin-top: 50px;
  padding: 20px 60px;
  border-radius: 50px;
  font-family: 'Josefin Sans', cursive;
  font-weight: 600;
  background-color: var(--nav);
  text-align: left;
  font-size: 20px;
  box-shadow: rgba(0, 0, 0, 0.24) 0px 10px 10px;
}
/*навігація*/
nav{
  width: 100%;
  margin: 0 auto;
  display: flex;
```

```

        justify-content: space-between;
        align-items: flex-start;
        font-size: 20px;
    }
    nav .menu a
    {
        color: black;
        margin-left: 5px;
        padding: 10px 15px;
        border-radius: 15px;
    }
    nav .menu a:hover
    {
        background-color: #cb7103;
        box-shadow: rgba(100, 100, 100, 0.2) 0px 10px 30px 0px;
    }
    nav #logo
    {
        font-size: 20px;
        font-family: 'Josefin Sans', cursive;
        color: black;
        padding-bottom: 20px;
    }
    /*верхній блок*/
    #intro
    {
        width: 80%;
        margin: 40px auto;
        display: flex;
        justify-content: space-between;
        align-items: flex-start;
    }
    #intro .texts
    {
        flex-grow: 1;
    }
    #intro .texts .title
    {
        font-family: var(--font-title);
        display: inline-block;
        font-size: 40px;
        margin-top: 100px;
        color: var(--stronger-green);
        font-weight: 500;
    }
    #intro .vector img
    {
        width: 350px;
    }
    /*нижній блок*/
    #demonstrative

```

```

{
    width: 80%;
    margin: 10px auto;
    display: flex;
    justify-content: space-between;
    align-items: center;
}
#demonstrative .explanation
{
    width: 400px;
}
#demonstrative .explanation h2
{
    margin-bottom: 100px;
}

/*FOOTER*/
footer
{
    width: 100%;
    height: 100px;
    margin-top: 40px;
    padding: 40px 55px;
    background-color: chocolate;

    display: flex;
    justify-content: space-evenly;
    align-items: center;
}
footer .name
{
    color: black;
    font-family: 'Josefin Sans', cursive;
    font-weight: 100;
    font-size: 20px;
}
footer .contact
{
    color: black;
}
footer a, footer a:visited
{
    color: black;
    text-decoration: underline;
}
footer p
{
    font-size: 20px; }

```

Б.4.2 Текст файла normalize.css

```
html
{
    line-height: 1.15;
    -webkit-text-size-adjust: 100%;
}
body
{
    margin: 0;
}
main
{
    display: block;
}
h1
{
    font-size: 2em;
    margin: 0.67em 0;
}
hr
{
    box-sizing: content-box;
    height: 0;
    overflow: visible;
}
pre
{
    font-family: monospace, monospace;
    font-size: 1em;
}
a
{
    background-color: transparent;
}
abbr[title]
{
    border-bottom: none;
    text-decoration: underline dotted;
}
b, strong
{
    font-weight: bolder;
}
code, kbd, samp
{
    font-family: monospace, monospace;
    font-size: 1em;
}
small
```

```

{
    font-size: 80%;
}
sub, sup
{
    font-size: 75%;
    line-height: 0;
    position: relative;
    vertical-align: baseline;
}
sub
{
    bottom: -0.25em;
}
sup
{
    top: -0.5em;
}
img
{
    border-style: none;
}
button, input, optgroup, select, textarea
{
    font-family: inherit;
    font-size: 100%;
    line-height: 1.15;
    margin: 0;
}
button, input
{
    overflow: visible;
}
button, select
{
    text-transform: none;
}
button,
[type="button"],
[type="reset"],
[type="submit"]
{
    -webkit-appearance: button;
}
button::-moz-focus-inner,
[type="button"]::-moz-focus-inner,
[type="reset"]::-moz-focus-inner,
[type="submit"]::-moz-focus-inner
{
    border-style: none;
    padding: 0;
}

```

```

}
button:-moz-focusring,
[type="button"]:-moz-focusring,
[type="reset"]:-moz-focusring,
[type="submit"]:-moz-focusring
{
    outline: 1px dotted ButtonText;
}

fieldset
{
    padding: 0.35em 0.75em 0.625em;
}
legend
{
    box-sizing: border-box;
    color: inherit;
    display: table;
    max-width: 100%;
    padding: 0;
    white-space: normal;
}
progress
{
    vertical-align: baseline;
}
textarea
{
    overflow: auto;
}
[type="checkbox"],
[type="radio"]
{
    box-sizing: border-box;
    padding: 0;
}

[type="number"]::-webkit-inner-spin-button,
[type="number"]::-webkit-outer-spin-button
{
    height: auto;
}
[type="search"]
{
    -webkit-appearance: textfield;
    outline-offset: -2px;
}
[type="search"]::-webkit-search-decoration
{
    -webkit-appearance: none;
}

```

```

::-webkit-file-upload-button
{
    -webkit-appearance: button;
    font: inherit;
}
details
{
    display: block;
}
summary
{
    display: list-item;
}
template
{
    display: none;
}
[hidden]
{
    display: none;
}

```

Б.4.3 Текст файла style.css

```

:root
{
    --body:#c1c54e ;
    --nav: #FDB10B;
    --lighter: #FEF4C0;
    --green: #807622;
    --font: 'Josefin Sans', cursive;
    --stronger-purple: #d97c02;
}
*
{
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}
body
{
    background-color: var(--body);
    font-family: 'Josefin Sans', sans-serif;
}
a, a:visited
{
    color: black;
    text-decoration: none;
    cursor: pointer;
}

```

```

}

/*Навігація*/

nav
{
    width: 1000px;
    height: 65px;
    margin: 0 auto;
    margin-top: 50px;
    padding: 20px 60px;
    border-radius: 50px;
    font-family: 'Josefin Sans', cursive;
    font-weight: 600;
    background-color: var(--nav);
    text-align: left;
    font-size: 20px;
    box-shadow: rgba(0, 0, 0, 0.24) 0px 10px 10px;
}
a.icons
{
    float: right;
    color: black;
    margin-left: 30px;
}
.marker
{
    width: 150px;
    line-height: 40px;
    margin: 0 auto;
    padding: 5px;
    background-color: var(--green);
    text-align: center;
    border-radius: 0 0 10px 10px;
    font-family: 'Josefin Sans', cursive;
    font-weight: 400;
    font-size: 20px;
    color: white;
    box-shadow: rgba(0, 0, 0, 0.24) 0px 10px 10px;
}
.marker-week
{
    width: 500px;
    line-height: 20px;
    margin: 15px auto;
    padding: 10px;
    position: relative;
    text-align: center;
    border-radius: 0 0 7px 7px;
    font-family: 'Josefin Sans', cursive;
    font-size: 21px;

```

```

}
.datepicker,
.marker-week input,
.table-condensed
{
    width: 250px;
}
/*вхід/реєстрація*/
#login-register
{
    width: 750px;
    min-height: 400px;
    margin: 44px auto;
    display: flex;

    justify-content: space-between;
    align-items: flex-start;
}
#login-register .field
{
    width: 45%;
    min-height: 200px;
    padding: 10px 10px;
    background-color: var(--lighter);
    border-radius: 15px;
    box-shadow: rgba(15, 15, 50, 0.15) 0px 50px 100px 0px;
}
#login-register h3
{
    text-align: center;
    width: 220px;
    margin: 20px auto;
    line-height: 35px;
    background-color: #DB7900;
    border-radius: 7px;
    color: white;
}
/* форма */
.form-general, .profile
{
    display: block;
    padding: 10px 20px;
    border: 0;
    border-radius: 7px;
    box-shadow: rgba(0, 0, 0, 0.15) 2px 2px 3px;
    margin: 20px auto;
}
.buttons
{
    display: block;
    margin: 20px auto;
}

```

```

border: none;
padding: 10px 20px;
background-color: var(--green);
color: white;
border-radius: 7px;
cursor: pointer;
box-shadow: rgba(100, 100, 100, 0.2) 0px 10px 30px 0px;
}
.buttons:hover
{
background-color: var(--stronger-purple);
}
input[type="file"]
{
display: none;
}
/*список задач*/
#todolist
{
width: 90%;
height: 500px;
margin: 15px auto;

display: flex;
justify-content: space-evenly;
align-items: center;
flex-wrap: wrap;
overflow-wrap: break-word;
}
#diary
{
width: 100%;
height: 90%;
margin: 44px auto;

display: flex;
flex-direction: column;
justify-content: center;
align-items: flex-start;
flex-wrap: wrap;
}
#diary .field
{
width: 70%;
height: 200px;
margin: 0 auto;
margin-bottom: 20px;
background-color: var(--lighter);
border-radius: 15px;
overflow: auto;

```

```

        box-shadow: rgba(50, 50, 90, 0.25) 0px 5px 10px -5px, rgba(0, 0,
0, 0.3) 0px 5px 10px -5px;
    }
    .add-days
    {
        width: 45%;
        height: 130px;
        margin-bottom: 20px;
        margin-right: 30px;
        border-radius: 15px;
        overflow: auto;
        box-shadow: rgba(50, 50, 90, 0.25) 0px 5px 10px -5px, rgba(0, 0,
0, 0.3) 0px 5px 10px -5px;
    }
    #todolist .field
    {
        width: 21%;
        height: 210px;
        background-color: var(--lighter);
        border-radius: 15px;
        overflow: auto;
        box-shadow: rgba(50, 50, 90, 0.25) 0px 5px 10px -5px, rgba(0, 0,
0, 0.3) 0px 5px 10px -5px;
    }
    .add-days h4
    {
        position: sticky;
        top: 0;
        text-align: center;
        font-family: 'Josefin Sans';
        width: 100%;
        margin-top: -10px;
        bottom: 25px;
        line-height: 35px;
        font-size: 1.2em;
        background-color: #DB7900;
        color: white;
        text-transform: uppercase;
    }
    #todolist h4
    {
        position: sticky;
        top: 0;
        text-align: center;
        font-family: 'Josefin Sans';
        width: 100%;
        margin-top: -10px;
        bottom: 25px;
        line-height: 35px;
        font-size: 1.2em;
    }

```

```

        background-color: #DB7900;
        color: white;
        text-transform: uppercase;
    }
#todolist ul
{
    list-style-position: inside;
    font-size: 20px;
    padding: 5px 10px;
}
.add-days ul
{
    list-style-position: inside;
    font-size: 20px;
    padding: 5px 10px;
    width: 100%;
    height: 50%;
    background-color: var(--lighter);
}
#todolist li
{
    margin-top: 5px;
    margin-left: 5px;
    width: 90%;
}
.add-days li
{
    margin-top: 5px;
    margin-left: 5px;
    width: 90%;
    background-color: var(--lighter);
}
#todolist p
{
    display: inline;
    width: 81%;
    float: right;
}
.add-days p
{
    display: inline;
    width: 81%;
    float: right;
}
/* додати задачу/запис */
.add-task
{
    text-align: center;
}
#todolist .add-task h4
{

```

```

    font-family: 'Josefin Sans';
    background-color: var(--green);
}
.add-task h4
{
    font-family: 'Josefin Sans';
    background-color: var(--green);
}
#todolist.add-task form
{
    margin-top: 50px;
}
.add-dairy
{
    display: flex;
    flex-direction: column;
    justify-content: flex-start;
    align-items: center;

    width: 45%;
    height: 1000px;
    margin-top: 30px;
    margin-right: 30px;
    background-color: var(--lighter);
    border-radius: 15px;
    box-shadow: rgba(50, 50, 90, 0.25) 0px 5px 10px -5px, rgba(0, 0, 0, 0.3) 0px 5px 10px -5px;
}
.add-dairy h4
{
    position: sticky;
    top: 0;
    text-align: center;
    font-family: 'Josefin Sans';
    width: 100%;
    margin-top: -10px;
    bottom: 25px;
    line-height: 35px;
    font-size: 1.2em;
    background-color: var(--green);
    color: white;
    text-transform: uppercase;
}
.add-dairy button
{
    width: 100%;
    border: none;
    background-color: rgb(219, 121, 0);
    cursor: pointer;
    padding: 10px;
    font-size: 0.9em;
}

```

```

    border-radius: 15px;
    margin-bottom: 10px;
    color: white;
    text-transform: uppercase;
    box-shadow: rgba(50, 50, 90, 0.25) 0px 5px 10px -5px, rgba(0, 0,
0, 0.3) 0px 5px 10px -5px;
}
.add-task .texts, .add-task select
{
    width: 95%;
    display: block;
    margin: 10px auto 10px auto;
    padding: 8px 12px;
    font-size: 40px;
    border: none;
    border-radius: 10px;
    box-shadow: rgba(0, 0, 0, 0.15) 2px 2px 3px;
}
.add-task .texts, .add-task input[type='date']
{
    display: block;
    width: 95%;
    margin: 5px auto 10px auto;
    padding: 8px 12px;
    font-size: 20px;
    border: none;
    border-radius: 10px;
    box-shadow: rgba(0, 0, 0, 0.15) 2px 2px 3px;
}
.add-task button
{
    width: 60%;
    border: none;
    background-color: rgb(219, 121, 0);
    cursor: pointer;
    padding: 10px;
    font-size: 0.9em;
    border-radius: 15px;
    color: white;
    text-transform: uppercase;
    box-shadow: rgba(50, 50, 90, 0.25) 0px 5px 10px -5px, rgba(0, 0,
0, 0.3) 0px 5px 10px -5px;
}
.add-task button:hover
{
    background-color: var(--green);
}
.edit-task
{
    opacity: 0;
    visibility: hidden;

```

```

        background-color: var(--lighter);
        color: black;
        width: 70%;
        height: auto; /* Змінено на авто, щоб контент можна було
розтягнути */
        border-radius: 15px;
        position: absolute;
        z-index: 1;
        box-shadow: rgba(100, 100, 100, 0.2) 50px 50px 50px 50px;
    }
    .edit-tasks{
        opacity: 0;
        visibility: hidden;
        background-color: var(--lighter);
        color: black;
        width: 270px;
        height: 180px;
        position: absolute;
        z-index: 1;

        border-radius: 15px;
        box-shadow: rgba(50, 50, 93, 0.25) 0px 6px 12px -2px, rgba(0, 0,
0, 0.3) 0px 3px 7px -3px;
    }
    .edit-tasks:target
    {
        opacity: 1;
        visibility: visible;
    }
    .edit-task:target
    {
        opacity: 1;
        visibility: visible;
    }
    .edit-task1
    {
        opacity: 0;
        visibility: hidden;
        background-color: var(--lighter);
        color: black;
        width: 70%;
        height: auto; /* Змінено на авто, щоб контент можна було
розтягнути */
        margin-left: 15%;
        border-radius: 15px;
        position: absolute;
        z-index: 1;
        box-shadow: rgba(100, 100, 100, 0.2) 50px 50px 50px 50px;
    }

```

```

.edit-task1 .texts, .edit-task1 input[type='date']
{
    display: block;
    width: 70%;
    margin: 5px auto 10px auto;
    padding: 8px 12px;
    font-size: 20px;
    border: none;
    border-radius: 10px;
    box-shadow: rgba(0, 0, 0, 0.15) 2px 2px 3px;
}
.edit-task1:target
{
    opacity: 1;
    visibility: visible;
}
h3.rewrite
{
    position: relative;
    display: block;
    text-align: center;
    margin: 0 auto;
}
input[type='text'].rewrite
{
    width: 95%;
    display: block;
    padding: 10px 15px;
    border: 0;
    border-radius: 10px;
    box-shadow: rgba(0, 0, 0, 0.15) 2px 2px 3px;
    margin: 20px auto;
    background-color: white;
}
button.rewrite
{
    width: 60%;
    display: block;
    padding: 5px 5px;
    border: 0;
    margin: 0 auto;
    box-shadow: rgba(0, 0, 0, 0.15) 2px 2px 3px;
    background-color: var(--green);
    color: white;
    font-size: 20px;
    border-radius: 15px;
    cursor: pointer;
}
button.rewrite:hover
{
    opacity: 0.9;
}

```

```

}

/* иконки */
.fa-thumbtack
{
    margin-left: -20px;
    margin-right: 15px;
}
.fa-eraser
{
    float: right;
    cursor: pointer;
}
.fa-eraser:hover
{
    color: rgb(35, 82, 124);
}
.fa-plus
{
    float: left;
    cursor: pointer;
    font-size: 24px;
}
.fa-pencil
{
    float: right;
    cursor: pointer;
    margin-right: 2px;
}
.fa-circle-check
{
    float: right;
    cursor: pointer;
    margin-right: 2px;
}
.fa-circle-xmark
{
    float: right;
    cursor: pointer;
    margin-right: 2px;
}
.fa-pencil:hover
{
    color: rgb(35, 82, 124);
}
.fa-save
{
    margin-right: 10px;
}
.close{
    position: relative;

```

```

        top: 10px;
        cursor: pointer;
        display: inline;
        padding: 3px;
        font-size: 1.2em;
    }
    /*профіль/вихід*/
    #logout .field
    {
        display: block;
        width: 350px;
        height: 250px;
        margin: 95px auto 70px auto;
        background-color: var(--lighter);
        border-radius: 15px;
        box-shadow: rgba(100, 100, 100, 0.2) 0px 10px 30px 0px;
    }
    #logout h2
    {
        width: 100%;
        text-align: center;
        position: sticky;
        bottom: 25px;
        font-size: 30px;
        padding-top: 25px;
    }
    #logout ul
    {
        width:100%;
        list-style: none;
        display: block;
        text-align: center;
        margin-top: 20px;
        font-size: 20px;
    }
    #logout .buttons
    {
        width:50%;
        margin-top: 60px;
        line-height: 30px;
        font-size: 20px;
        background-color: var(--green);
        border: none;
        border-radius:15px;
    }
    #logout .buttons:hover
    {
        background-color: var(--green);
        text-decoration: none;
    }
    /* ICONS */

```

```

.fa-id-card, .fa-pencil-alt, .fa-clock
{
    margin-right: 5px;
    font-size: 1em;
}
/*FOOTER*/
footer
{
    width: 100%;
    height: 173px;
    margin-top: 40px;
    padding: 40px 55px;
    background-color: chocolate;
    display: flex;
    justify-content: space-evenly;
    align-items: center;
}
footer .name
{
    color: white;
    font-family: 'Josefin Sans', cursive;
    font-weight: 600;
    font-size: 30px;
}
footer .contact
{
    color: white;
}
footer a, footer a:visited
{
    color: white;
    text-decoration: underline;
}
footer p
{
    font-size: 1.5em;
}
.marker-week-hover tr:hover
{
    background-color: #FFD19B;
}
.marker-day-hover td:hover
{
    background-color: #FFD19B;
}
.marker-day-hover t
.bootstrap-datetimepicker-widget tr:hover
{
    cursor: pointer;
}

```

ДОДАТОК В
Слайди презентації

Розробка вебзастосунку онлайн щоденника зі списком задач

Виконала: студентка групи КНТ-139
КАЛЬНИЧЕНКО Ангеліна
Керівник: доцент к.н.т.
КОЛПАКОВА Тетяна

Рисунок В.1 – Слайд № 1

Огляд проєкту

Об'єкт дослідження – процес зберігання та систематизації записів і задач.

Предмет дослідження – вебзастосунок для ведення списку задач.

Мета роботи – розробити онлайн щоденник з можливістю створення, редагування та видалення записів та задач.

2

Рисунок В.2 – Слайд № 2

Аналіз предметної області

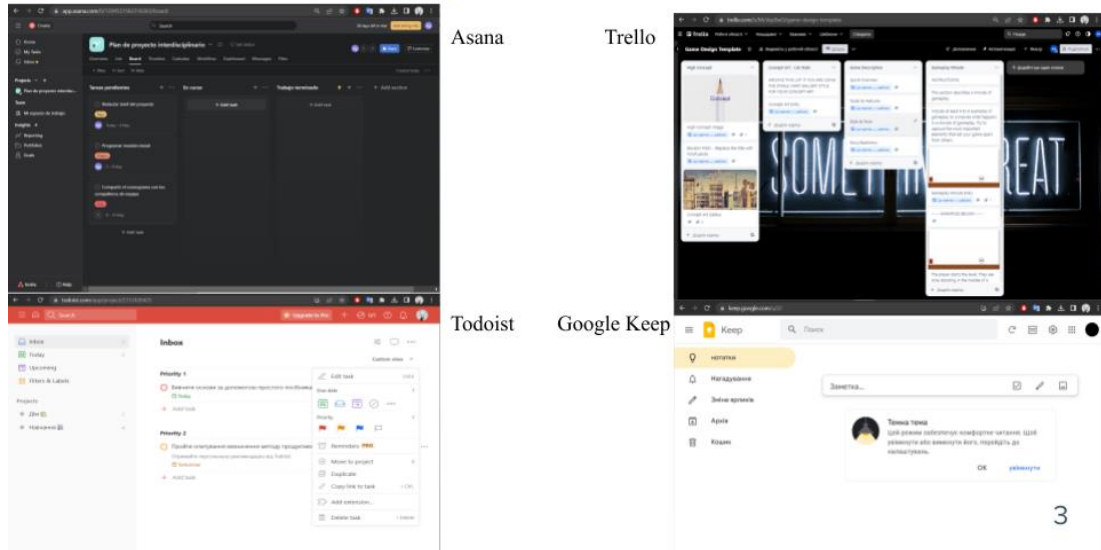


Рисунок В.3 – Слайд № 3

Технології для розробки сайту

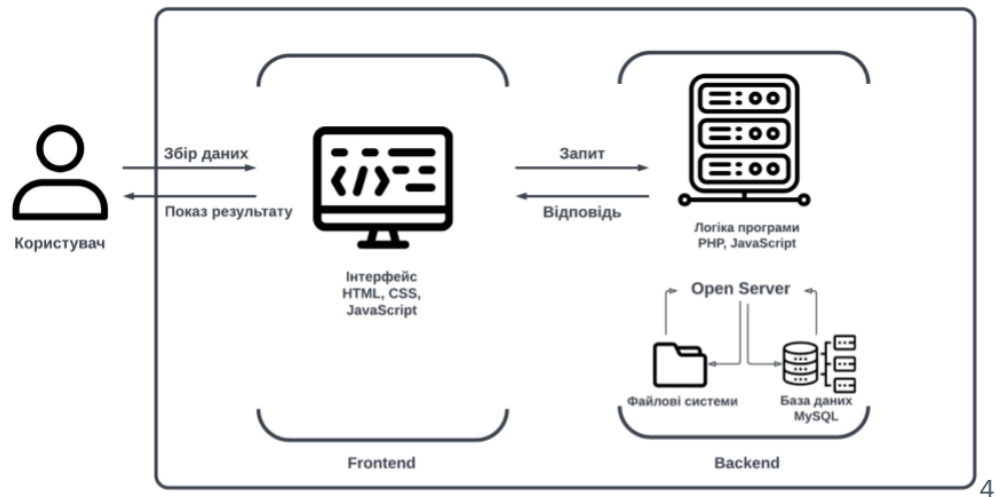
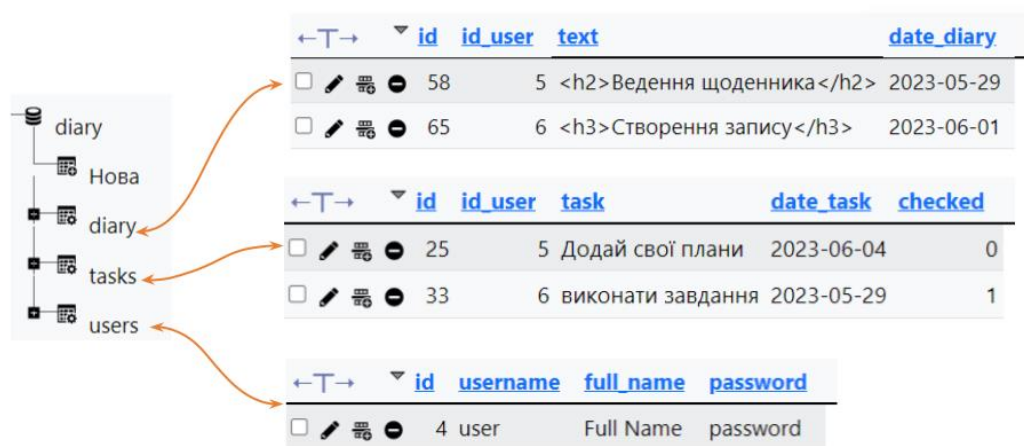


Рисунок В.4 – Слайд № 4

База даних



5

Рисунок В.5 – Слайд № 5

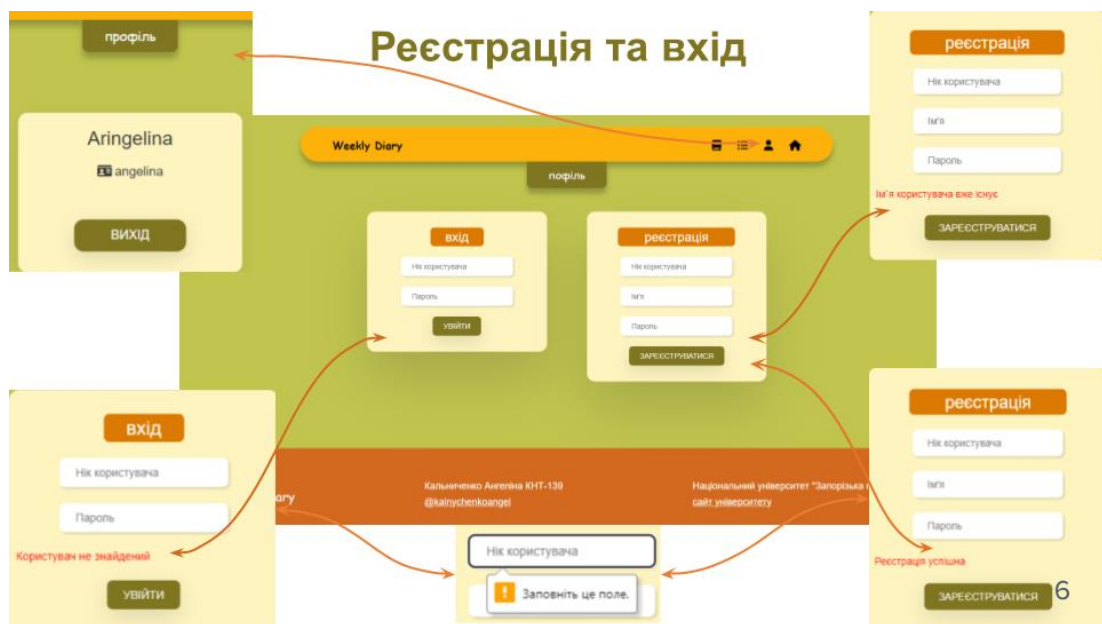


Рисунок В.6 – Слайд № 6

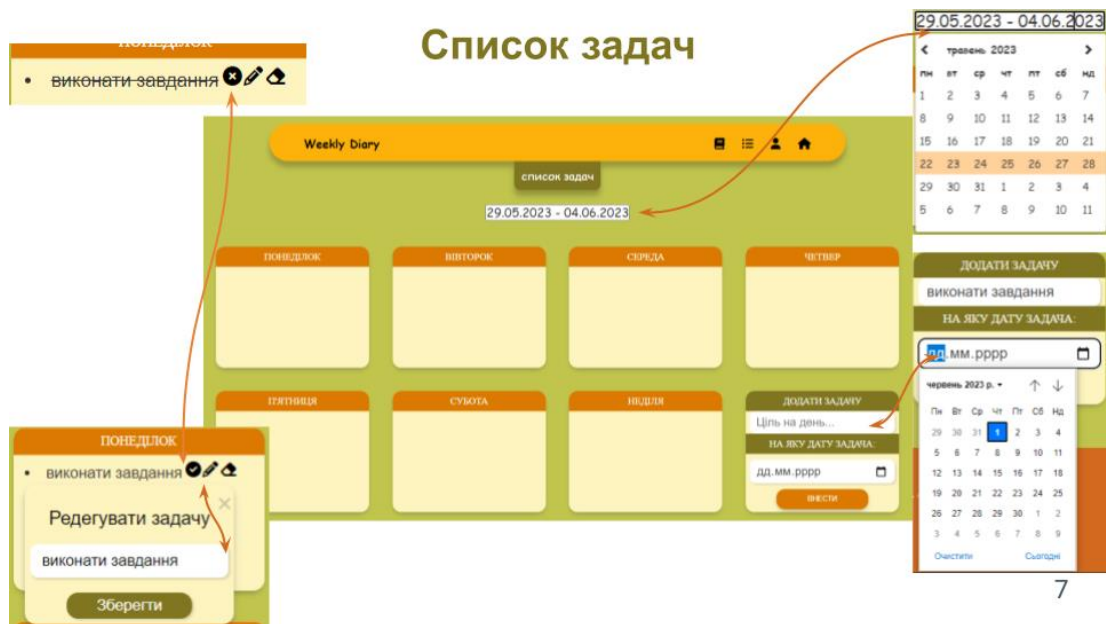


Рисунок В.7 – Слайд № 7

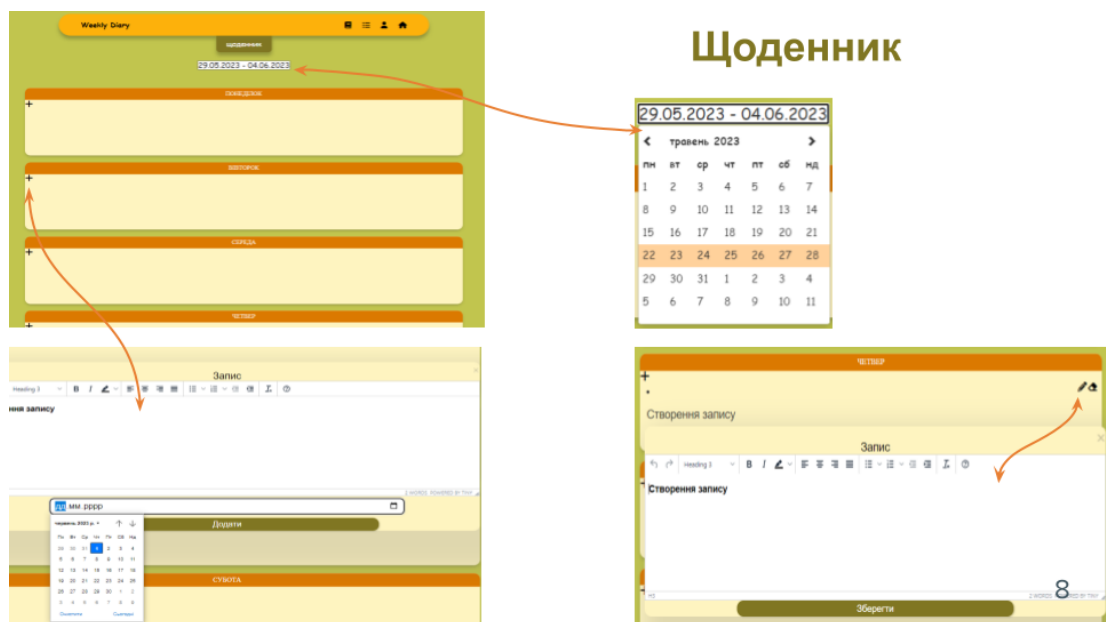


Рисунок В.8 – Слайд № 8

Висновки

Задачі, які було виконано в рамках даної дипломної кваліфікаційної роботи:

- проведено аналіз існуючого програмного забезпечення;
- проведено аналіз та вибір засобів розробки програмного продукту;
- розроблено та реалізовано онлайн щоденник зі списком задач з можливістю створення, редагування та видалення записів та завдань;
- керівництво програміста під час розробки проекту;
- керівництво користувача щодо використання розробленої системи.

9

Рисунок В.9 – Слайд № 9