МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) Факультет автоматизації та комп'ютерних систем
Кафедра Інформаційних технологій, штучного інтелекту і кібербезпеки

«До захисту в ЕК»	**«До захисту допущено»**
Директор інституту (декан факультету)	Завідувач кафедри
_____ Андрій ФОРСЮК	_____ Сергій ГРИБКОВ
(підпис) (ім'я та прізвище)	(підпис) (ім'я та прізвище)
« 05 » червня 2023р.	« 05 » червня 2023р.

КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА

зі спеціальності 122 «Комп'ютерні науки»
(код і назва спеціальності)
освітньо-професійної програми «Комп'ютерні науки»

на тему: " Розроблення веб-додатку для управління проектами та робочим часом аутсорсингової компанії по створенню та супроводі сайтів "

Виконав: здобувач 4 курсу, групи КН-4-2 .

Бельдій Дмитро Олександрович _____
(прізвище, ім'я та по батькові повністю) (підпис)

Керівник Сільвестров Антон Миколайович _____
(прізвище, ім'я та по батькові повністю) (підпис)

Консультанти _____ _____
(ім'я та прізвище) (підпис)

_____ _____
(ім'я та прізвище) (підпис)

_____ _____
(ім'я та прізвище) (підпис)

Рецензент Овсієнко Олександр _____
(ім'я та прізвище) (підпис)

Я як здобувач(ка) Національного університету харчових технологій розумію і підтримую політику університету з академічної доброчесності. Я не надавав(-ла) і не одержував(-ла) недозволеної допомоги під час підготовки цієї роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Здобувач_____
(підпис)

Київ — 2023р

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) Автоматизація і комп'ютерних систем
Кафедра Інформаційних технологій, штучного інтелекту і кібербезпеки
Освітній ступінь Бакалавр
Спеціальність 122 «Комп'ютені науки»
(код і назва)
Освітньо-професійна програма «Комп'ютерні науки»
(назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри
Інформаційних технологій, штучного інтелекту і кібербезпеки
Сергій ГРИБКОВ

« 04 » квітня 2023року

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА

Бельдія Дмитра Олександровича
(прізвище, ім'я, по батькові)

1. Тема роботи " Розроблення веб-додатку для управління проектами та робочим часом аутсорсингової компанії по створенню та супроводі сайтів"

керівник роботи Сільвестров Антон Миколайович, професор, доктор технічних наук,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом закладу вищої освіти від «03_»квітня 2023 р. № 204-кс
2. Строк подання здобувачем роботи: 01.06.2023 р.
3. Вихідні дані до роботи: Вимоги користувачів: вимоги від робітників компанії, які будуть використовувати систему. Потреби в плануванні задач, слідкуванні за часом, веденні проектів.

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити): Функціональні вимоги до системи, модулі та компоненти, доступні операції та функції, зберігання та відстеження даних, візуалізація, безпеки даних, інтеграція з іншими системами, критерії успішності, інтерфейс користувача.

5. Перелік графічного матеріалу:
Структура компаніїї Lytvynov Production, скріншоти інтерфесу системи, скріншоти різних програм

6. Консультанти розділів роботи:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Сільвестров А.М., професор		
2	Сільвестров А.М., професор		
3	Сільвестров А.М., професор		
4	Сільвестров А.М., професор		

7. Дата видачі завдання 4 квітня 2023 року

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів виконання кваліфікаційної роботи	Строк виконанняетапів роботи	Примітка
1	Фаза аналізу та планування	01.04.2023-15.04.2023	
2	Розробка концепції та дизайну	15.04.2023-25.04.2023	
3	Розробка базового функціоналу	25.04.2023-05.05.2023	
4	Розробка додаткового функціоналу	05.04.2023-20.05.2023	
5	Впровадження та тестування	20.05.2023-07.06.2023	
6	Завершення робіт та налагодження	07.05.2023-25.06.2023	
7	Тестування	25.06.2023-30.06.2023	

Здобувач _____ Бельдій Д.О.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Сільвестров А.М.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Квалифікаційна робота: 58 с., 19 рисунків, 25 джерел, 2 таблиці.

Ключові слова: JavaScript, TypeScript, Angular, JsonServer, TailwindCss, CSS, HTML.

"Розроблення веб-додатку для управління проектами та робочим часом в аутсорсинговій компанії “Lytvynov Production"

Дана кваліфікаційна робота присвячена розробленню веб-додатку для ефективного управління проектами та робочим часом в аутсорсинговій компанії Lytvynov Production. Основною метою дослідження є полегшення процесів планування, виконання та моніторингу проектів, а також ефективного управління робочим часом для співробітників компанії.

У роботі буде проведено аналіз поточних проблем, з якими стикаються працівники аутсорсингової компанії Lytvynov Production у контексті управління проектами та робочим часом. Будуть визначені основні вимоги та потреби працівників компанії щодо веб-додатку для оптимізації цих процесів.

Далі будуть розглянуті сучасні технології та методики, що використовуються для ефективного управління проектами та робочим часом. Будуть досліджені інструменти для планування проектів, відстеження завдань, контролю робочого часу та генерації звітів.

На основі проведеного аналізу та використання сучасних технологій буде розроблений веб-додаток, який має наступні функціональні можливості, адаптовані до потреб компанії Lytvynov Production.

ANNOTATION

Qualification work: 61 pages, 19 figures, 25 sources, 2 tables.

Keywords: JavaScript, TypeScript, Angular, JsonS, erver, TailwindCss, CSS, HTML.

Abstract of the study "Development of a web application for project and working time management in the outsourcing company Lytvynov Production"

This qualification work is devoted to the development of a web application for effective project and working time management in the outsourcing company Lytvynov Production. The main goal of the study is to facilitate the processes of planning, execution and monitoring of projects, as well as effective management of the working time of the company's employees.

The work will analyze the current problems faced by the outsourcing company Lytvynov Production in the context of project and working time management. The main requirements and needs of the company regarding the web application will be determined to optimize these processes.

Next, modern technologies and methods used for effective project and work time management will be considered. Tools for project planning, task tracking, time tracking and report generation will be explored.

Based on the analysis and the use of modern technologies, a web application will be developed that has the following functionalities, adapted to the needs of the Lytvynov Production company.

ЗМІСТ

ВСТУП 8
РОЗДІЛ 1. АНАЛІЗ ОБ'ЄКТА ДОСЛІДЖЕННЯ ТА ЗАДАчі АВТОМАТИЗАЦІЇ 10
1.1. Загальна характеристика компанії Lytvynov Production 10
1.2. Структура компаніїї Lytvynov Production 11
1.3. Робота та взаємодія відділів 12
1.4. Структура відділу розробки 14
1.5. Аналіз роботи IT-відділу 15
1.6. Аналіз проблем автоматизації IT-відділу 16
1.7. Пошук існуючих рішень 18
1.8. Аналіз nTask 19
1.9. Аналіз Asana 21
1.10. Порівняння систем-аналогів 22
1.11. Доцільність проєктування й розроблення системи 24
1.12. Концептуальна модель системи 25
РОЗДІЛ 2. ТЕХНІЧНЕ ЗАВДАННЯ НА ПРОЄКТУВАННЯ 27
2.1. Загальні положення 27
2.2. Призначення і цілі розробки 27
2.3. Характеристика об'єкту автоматизщації 28
2.4. Вимоги до системи 29
2.5. Вимоги до структури системи 30
2.6. Вимоги до функцій системи 31
2.7. Вимоги до збереження даних 32
2.8. Вимоги до надійності 33
2.9. Вимоги до лінгвістичного забезпечення 34
2.10. Вимоги до програмного забезпечення 35
2.11. Мінімальні системні та апаратні вимоги 36
РОЗДІЛ 3 КОМПЛЕКС ЗАДАЧ АВТОМАТИЗАЦІЇ 38
3.1. Інформаційне забезпечення системи 38

3.2. Алгоритмізація та реалізація комплексу задач автоматизації. 39
3.3. Інструкція користувача. 39
3.4. Технічне та системне забезпечення розробки. 46
3.5. Обґрунтування вибору технічних засобів 47
3.6. Топологія комп’ютерної мережі 48
РОЗДІЛ 4. ОХОРОНА ПРАЦІ 49
ВИСНОВОК 53
ВИКОРИСТАНІ ДЖЕРЕЛА: 55
ДОДАТКИ 58
Додаток А. Скріншоти програми 58
Додаток В. Код програми: 60

ВСТУП

У сучасному світі, де технології швидко розвиваються, а конкуренція стає все жорсткішою, ефективне управління проектами та робочим часом є ключовими факторами успіху для багатьох організацій. Особливо важливою ця проблематика є для аутсорсингових компаній, які займаються створенням та супроводом веб-сайтів.

Ця кваліфікаційна робота присвячена розробленню веб-додатку, який спрямований на управління проектами та робочим часом в аутсорсинговій компанії Lytvynov Production. Головна мета роботи - розробити інструмент, який допоможе оптимізувати процеси планування, виконання та моніторингу проектів, а також забезпечити ефективне використання робочого часу працівників.

Аутсорсингова компанія Lytvynov Production має значний досвід у розробці та супроводі веб-сайтів, проте з часом постає потреба вдосконалити свою систему управління проектами та робочим часом. Традиційні методи планування та відстеження виявляються недостатньо ефективними і не забезпечують необхідного рівня контролю та координації роботи.

Зважаючи на це, у роботі буде проведено аналіз поточних проблем, з якими зіштовхується компанія Lytvynov Production у контексті управління проектами та робочим часом. Будуть виявлені основні вимоги та потреби працівників компанії, а також проаналізовані сучасні технології та методики, які використовуються для ефективного управління проектами та робочим часом.

На основі проведеного аналізу буде розроблений веб-додаток, який відповідатиме вимогам і потребам аутсорсингової компанії Lytvynov Production. Цей додаток надасть працівникам компанії зручні та потужні інструменти для планування, виконання та контролю проектів, а також для ефективного використання робочого часу. Це дозволить компанії поліпшити

свою продуктивність, знизити ризики й помилки та забезпечити якісне та своєчасне виконання проектів.

Важливість ефективного управління проектами та робочим часом для аутсорсингових компаній, зокрема таких, як Lytvynov Production, не може бути переоцінена. У сучасному світі, де технології швидко розвиваються, а конкуренція стає все жорсткішою, кожна хвилина і кожна ресурсо-часова одиниця мають велике значення.

Ця кваліфікаційна робота спрямована на вирішення цієї проблеми шляхом розробки веб-додатку, який допоможе компанії Lytvynov Production управляти проектами та робочим часом її працівників з максимальною ефективністю. Розроблений додаток буде фокусуватися не тільки на підвищенні продуктивності, але й на поліпшенні комунікації та співпраці між різними командами та працівниками.

Аналізуючи поточні проблеми, з якими зіштовхується Lytvynov Production в управлінні проектами та робочим часом, будуть визначені основні вимоги та потреби працівників компанії. Дослідження також включатиме огляд сучасних технологій та методик, що використовуються в сфері управління проектами, зокрема веб-додатків, що дозволяють зручно планувати завдання, відстежувати прогрес та управляти робочим часом.

РОЗДІЛ 1. АНАЛІЗ ОБ'ЄКТА ДОСЛІДЖЕННЯ ТА ЗАДАЧІ АВТОМАТИЗАЦІЇ

1.1. Загальна характеристика компанії Lytvynov Production

В якості об'єкту дослідження було взято аутсорсингову компанію по розробці сайтів «Lytvynov Production». Це компанія, яка повністю працює віддалено, що дає її певні переваги над класичними фірмами. Компанія може реалізувати повноцінний цикл розробки сайту, від опису та формування технічного завдання до запуску готового проекту та подальша його підтримка.

Lytvynov Production є аутсорсинговою компанією, яка спеціалізується на створенні та супроводі веб-сайтів. Компанія має значний досвід у галузі веб-розробки та надає послуги клієнтам з різних секторів індустрії.

Основною метою Lytvynov Production є надання якісних та інноваційних рішень у сфері веб-розробки, допомога клієнтам у створенні привабливих та функціональних веб-сайтів, а також їх подальший супровід і підтримка.

Компанія пропонує широкий спектр послуг, включаючи веб-дизайн, програмування, розробку електронної комерції, оптимізацію для пошукових систем (SEO) та інтеграцію з різноманітними платформами та системами.

Команда професіоналів Lytvynov Production поєднує технічні знання з творчістю, що дозволяє їм реалізовувати індивідуальні потреби клієнтів та досягати високої якості веб-проектів.

Крім створення та супроводу веб-сайтів, Lytvynov Production також забезпечує консультаційні послуги щодо стратегії веб-присутності, допомагає клієнтам визначити їх цілі та потреби, а також розробляє індивідуальні рішення для досягнення успіху в онлайн-середовищі.

Компанія Lytvynov Production прагне постійно вдосконалюватись та застосовувати нові технології, щоб забезпечити своїм клієнтам інноваційні та конкурентні веб-рішення.

1.2. Структура компаніїї Lytvynov Production

У компанії Lytvynov Production існують два основних відділи, що виконують ключові функції:

1. Відділ розробки:

- Веб-дизайнери: відповідають за створення естетичного та привабливого дизайну веб-сайтів. Вони працюють над виглядом, компонуванням елементів та користувацьким інтерфейсом, щоб забезпечити зручну та привабливу взаємодію користувачів з сайтом.
- Веб-розробники: займаються програмуванням та розробкою функціональності веб-сайтів. Вони використовують різні мови програмування та технології, щоб реалізувати потрібні функції, інтерактивність та зв'язок з базою даних.
- Тестувальники програмного забезпечення: проводять тестування та перевірку якості веб-додатків та сайтів, щоб впевнитися, що вони працюють належним чином та задовольняють вимоги клієнтів.

2. Відділ менеджменту:

- Менеджери проектів: відповідають за планування, керування та контроль проектів. Вони спілкуються з клієнтами, визначають вимоги та терміни виконання проектів, розподіляють ресурси та забезпечують виконання завдань в рамках встановлених обмежень.
- Бізнес-аналітики: аналізують потреби та вимоги клієнтів, визначають обсяг робіт та варіанти виконання проектів. Вони допомагають уточнити технічні вимоги, розробляють стратегію проекту та визначають його метрики успіху.
- Координатори проектів: забезпечують взаємодію між різними відділами та учасниками проекту, здійснюють планування ресурсів та контроль за виконанням завдань. Вони відповідають за забезпечення ефективності та вчасного виконання проектів.

Ця організаційна структура (Рис. 1.1.) дозволяє компанії Lytvynov Production ефективно керувати розробкою веб-сайтів та забезпечувати виконання проектів, здійснюючи поєднання технічного розробки та проектного управління.

Рис.1.1 - Схема організаційної структури Lytvynov Production

1.3. Робота та взаємодія відділів

Відділ розробки та відділ менеджменту у компанії Lytvynov Production взаємодіють для успішного виконання проектів і забезпечення високої якості розробки веб-сайтів. Робота цих відділів базується на тісній співпраці та обміні інформацією, що дозволяє забезпечити ефективне управління проектами та досягнення поставлених цілей. Основні аспекти їх роботи та взаємодії включають:

1. Запити та вимоги клієнтів: Відділ менеджменту взаємодіє з клієнтами, збирає їх запити та вимоги щодо розробки веб-сайту. Ця інформація передається до відділу розробки, де веб-дизайнери та бізнес-аналітики аналізують вимоги та перетворюють їх на конкретні дизайн-концепції та технічні вимоги.

2. Планування проектів: Менеджери проектів спільно з бізнес-аналітиками розробляють стратегію проекту, визначають його обсяг, терміни

виконання та розподіляють ресурси. Вони співпрацюють з веб-розробниками, веб-дизайнерами та тестувальниками для оцінки ресурсного планування та визначення етапів розробки.

3. Дизайн та розробка: Веб-дизайнери створюють естетичний та привабливий дизайн веб-сайту, враховуючи вимоги клієнта та бренду компанії. Веб-розробники працюють над програмуванням та розробкою функціональності веб-сайту, дотримуючись заданих технічних вимог та стандартів.

4. Комунікація та звітність: Відділ менеджменту забезпечує постійну комунікацію між відділами розробки та клієнтом, надаючи оновлення про стан проектів, вирішення поточних питань та врахування змін в вимогах. Він також забезпечує звітність про хід проектів, включаючи часові рамки, бюджет та досягнення мілістоунів.

5. Тестування та якість: Тестувальники програмного забезпечення співпрацюють з веб-розробниками для проведення тестування веб-додатків та сайтів. Вони спостерігають за якістю роботи, виявляють та виправляють помилки, а також забезпечують відповідність вимогам та стандартам якості.

Ця взаємодія між відділом розробки та відділом менеджменту допомагає забезпечити високу ефективність проектів, виконання проектів в зазначені терміни та досягнення високої якості розробки веб-сайтів.

Крім тісної співпраці між відділом розробки та відділом менеджменту, компанія Lytvynov Production також прагне створити сприятливу робочу атмосферу, де співробітники можуть розкрити свій потенціал та внести вагомий внесок у розвиток проектів. Комунікація, колаборація та взаємодія між різними відділами стимулюються шляхом проведення регулярних зборів, спільних проектів та обміну знаннями. Компанія також надає співробітникам можливості для професійного розвитку, організовує навчальні заходи та тренінги, що допомагають підтримувати високий рівень компетентності та навичок у кожного співробітника.

Ці зусилля спрямовані на досягнення колективної мети компанії, а саме, надання клієнтам високоякісних веб-сайтів та послуг, забезпечення ефективного управління проектами та робочим часом і відповідь на змінні потреби ринку. Компанія Lytvynov Production завжди націлені на постійне вдосконалення своїх процесів та використання новітніх технологій для досягнення найкращих результатів і задоволення потреб своїх клієнтів.

1.4. Структура відділу розробки

Відділ розробки у аутсорсингової компанії Lytvynov Production складається з декількох ключових посад, кожна з яких відповідає за конкретні аспекти веб-розробки та дизайну. Ось загальна структура цього відділу:

1. Веб-розробник (Front-end Developer): Ця посада відповідає за розробку та реалізацію фронтенд-частини веб-додатків та сайтів. Веб-розробник забезпечує коректне відображення та взаємодію елементів інтерфейсу, використовуючи мови програмування, такі як HTML, CSS і JavaScript. Вони враховують дизайн та взаємодію з користувачем, забезпечуючи зручність та естетичний вигляд веб-додатку.

2. Бекенд-розробник (Back-end Developer): Бекенд-розробник займається розробкою серверної частини веб-додатків. Вони використовують різні мови програмування та технології, такі як PHP, Python або Java, для створення та оптимізації логіки програми, баз даних та серверних компонентів. Бекенд-розробник відповідає за безпеку, швидкодію та надійність функціоналу веб-додатка.

3. Full-stack розробник (Full-stack Developer): Full-stack розробник має знання та навички як у фронтенді, так і у бекенді розробки. Вони здатні працювати з усіма аспектами веб-додатка, включаючи як клієнтську, так і серверну частини. Full-stack розробники можуть бути відповідальними за

розробку повноцінного функціоналу, від користувацького інтерфейсу до бази даних та серверної інфраструктури.

4. UX/UI дизайнер (UX/UI Designer): UX/UI дизайнер займається створенням зручного та привабливого користувацького досвіду для веб-додатків. Вони проводять дослідження, створюють макети, прототипи та дизайн інтерфейсів, забезпечуючи, що веб-додаток є інтуїтивно зрозумілим, ефективним та естетично привабливим для користувачів.

5. QA інженер (Quality Assurance Engineer): QA інженер відповідає за забезпечення якості веб-додатків. Вони розробляють тести, проводять відповідні перевірки та аналізують результати, щоб виявити та виправити можливі помилки, баги та недоліки в роботі додатка. QA інженери допомагають забезпечити стабільну та безпроблемну роботу веб-додатка перед його впровадженням та підтримкою.

Ці ролі відділу розробки співпрацюють між собою для досягнення спільної мети - створення високоякісних та функціональних веб-додатків, які задовольняють потреби клієнтів та користувачів. Кожна посада вносить свій внесок у процес розробки, дизайну, тестування та оптимізації, забезпечуючи успішну реалізацію проектів компанії Lytvynov Production.

1.5. Аналіз роботи IT-відділу

Робота відділу розробки компанії Lytvynov Production демонструє високий рівень компетентності та професіоналізму у розробці веб-додатків. Відділ успішно співпрацює з іншими відділами компанії, такими як відділ менеджменту, для забезпечення успішної реалізації проектів.

Команда розробників, що включає в себе веб-розробників, бекенд-розробників, full-stack розробників, UX/UI дизайнерів та QA інженерів, працює скоординовано та ефективно. Кожен член команди виконує свою роль і вносить свій внесок у процес розробки веб-додатків.

Веб-розробники, бекенд-розробники та full-stack розробники володіють необхідними технічними навичками та знаннями у веб-розробці, що дозволяє їм ефективно створювати функціональні та високоякісні веб-додатки. Вони працюють над різними аспектами розробки, включаючи розробку фронтенду, бекенду, баз даних та інтеграцію зовнішніх сервісів.

UX/UI дизайнери відповідають за створення зручного та привабливого користувацького інтерфейсу веб-додатків. Вони проводять дослідження, розробляють макети та прототипи, забезпечуючи позитивний користувацький досвід і забезпечуючи відповідність дизайну бренду компанії.

QA інженери виконують важливу роль у забезпеченні якості веб-додатків. Вони розробляють тести, проводять перевірки та аналізують результати, щоб виявити та виправити можливі помилки.

1.6. Аналіз проблем автоматизації ІТ-відділу

Проблеми автоматизації включають

1. Слідкування за часом: Робітники можуть стикатися з викликом ефективного відстеження свого робочого часу, особливо якщо вони працюють над кількома проектами одночасно. Відсутність зручного та централізованого інструменту для відстеження та обліку витраченого часу може призводити до неправильного розподілу ресурсів і недооцінки реального часу, необхідного для завершення завдань.

2. Велика кількість проектів: У відділі розробки часто виникає ситуація, коли працівникам доводиться одночасно працювати над декількома проектами. Це може призвести до розкиданості уваги та втрати концентрації. Відсутність зручних інструментів для управління цими багатьма проектами може ускладнювати планування та призводити до недооцінки або неправильної пріоритизації завдань.

3. Плутання в проектах Слідкування за часом: Робітники можуть стикатися з викликом ефективного відстеження свого робочого часу, особливо якщо вони працюють над кількома проектами одночасно. Відсутність зручного та централізованого інструменту для відстеження та обліку витраченого часу може призводити до неправильного розподілу ресурсів і недооцінки реального часу, необхідного для завершення завдань.

4. Велика кількість проектів: У відділі розробки часто виникає ситуація, коли працівникам доводиться одночасно працювати над декількома проектами. Це може призвести до розкиданості уваги та втрати концентрації. Відсутність зручних інструментів для управління цими багатьма проектами може ускладнювати планування та призводити до недооцінки або неправильної пріоритизації: Без відповідних інструментів і систематичного підходу до управління проектами, робітники можуть стикатися з плутанням і незрозумілими пріоритетами. Це може призводити до затримок у виконанні завдань, невирішених конфліктів та загального незадоволення як у команді, так і у клієнтів.

5. Нагадування про наступні кроки: Відсутність систематичних нагадувань та повідомлень щодо наступних дій та завдань може призводити до пропуску дедлайнів та незавершених завдань. Робітникам може бути важко відстежувати, що потрібно робити далі та в якій послідовності, особливо при великій кількості завдань і проектів.

6. Складність оцінки пріоритетів: Без систематичного підходу до оцінки пріоритетів та важливості завдань, робітники можуть витрачати багато часу на менш важливі або невідкладні завдання, відводячи мало уваги на критичні завдання. Це може призводити до затримок у проектах та загального зниження продуктивності.

7. Відсутність інформації про стан реалізації: Відсутність централізованої системи для відстеження стану реалізації завдань та проектів може призводити до недоінформованості щодо поточного прогресу та

проблем, які можуть виникати під час виконання робіт. Робітники можуть бути змушені витрачати багато часу на пошук інформації та звітування про стан робіт.

1.7. Пошук існуючих рішень

У процесі пошуку існуючих рішень для вирішення проблеми недостатку персонального менеджменту проектів робітника, одними з популярних варіантів є nTask і Asana. Давайте розглянемо їх і кілька інших аналогічних інструментів:

1. nTask: nTask є потужним інструментом для управління проектами та завданнями. Він надає можливості планування, відстеження та контролю завдань, а також спільної роботи над проектами. nTask пропонує функції, такі як календар, панель завдань, контроль часу, спільна робота та звіти, що дозволяє робітникам ефективно керувати своїм робочим часом та завданнями.
2. Asana: Asana є популярною платформою для управління проектами та спільної роботи. Вона дозволяє створювати, призначати та відстежувати завдання, устанавлювати пріоритети, керувати календарем проектів та отримувати оновлення про стан проектів. Asana також надає зручні можливості спільної роботи, коментування та обміну документами.
3. Trello: Trello є інструментом, який дозволяє створювати дошки, списки та картки для управління завданнями та проектами. Він пропонує перегляд дошки, перетягування та відстежування завдань, додавання коментарів, установлення строків та спільну роботу з колегами.
4. Monday.com: Monday.com - це колаборативна платформа для управління проектами та командної роботи. Вона надає інтуїтивний інтерфейс, різноманітні шаблони для управління проектами, засоби планування та контролю завдань, а також можливості спільної роботи та звітування.

Якщо врахувати проблеми, пов'язані зі слідкуванням за часом, великою кількістю проектів, плутанням в проектах, нагадуванням про наступні кроки,

складністю оцінки пріоритетів та відсутністю інформації про стан реалізації, то найбільше підійдуть такі інструменти як : nTask і Asana. Обидва ці інструменти пропонують широкий спектр функціональності для ефективного управління проектами та завданнями. nTask забезпечує зручні модулі для планування, відстеження часу, спільної роботи та звітності, що сприяє кращому організаційному процесу відділу розробки. З іншого боку, Asana володіє багатофункціональною платформою, яка дозволяє створювати проекти, розподіляти ресурси, спілкуватися з командою та керувати завданнями. Обидва інструменти є зручними, простими у використанні та надійними варіантами для поліпшення управління проектами та роботою відділу розробки.

1.8. Аналіз nTask

nTask - це онлайн-платформа для управління проектами, завданнями та спільною роботою, яка надає широкий спектр функціональних можливостей. Проаналізуємо основні аспекти цього інструменту:

- Планування проектів: nTask дозволяє створювати проекти, визначати їх мету, об'єктиви та структуру. Можна встановлювати терміни виконання, призначати відповідальних та визначати пріоритети завдань.
- Управління завданнями: інструмент надає можливість створювати завдання, призначати їх виконавців та встановлювати дедлайни. Також можна відстежувати статус завдань, додавати коментарі та вкладати файли до завдань.
- Спільна робота та комунікація: nTask забезпечує зручні інструменти для спільної роботи та комунікації в команді. Можна створювати коментарі, обговорювати завдання та обмінюватися файлами прямо в рамках платформи.

- Відстеження часу: інструмент надає можливість відстежувати час, витрачений на виконання завдань. Це дозволяє краще контролювати робочий час та оцінювати продуктивність.
- Звітність: nTask генерує різноманітні звіти про виконання проектів, прогрес завдань та витрачений час. Це допомагає аналізувати продуктивність, виявляти проблемні аспекти та приймати обґрунтовані рішення.
- Інтеграції: nTask підтримує інтеграцію з різними іншими інструментами, такими як Google Календар, Slack, Dropbox, Zapier та інші. Це дозволяє зручно об'єднувати різні робочі процеси та забезпечувати зв'язок з іншими інструментами.
- Оглянувши функціональність nTask, можна визначити, що цей інструмент має потужний набір функцій для управління проектами, завданнями та спільною роботою. Він може бути корисним для відділу розробки, допомагаючи покращити планування, відстеження часу та спілкування в команді. Однак, варто також розглянути інші інструменти, такі як Asana, для порівняння та вибору оптимального рішення для конкретних потреб та вимог команди (Рис 1.2) .

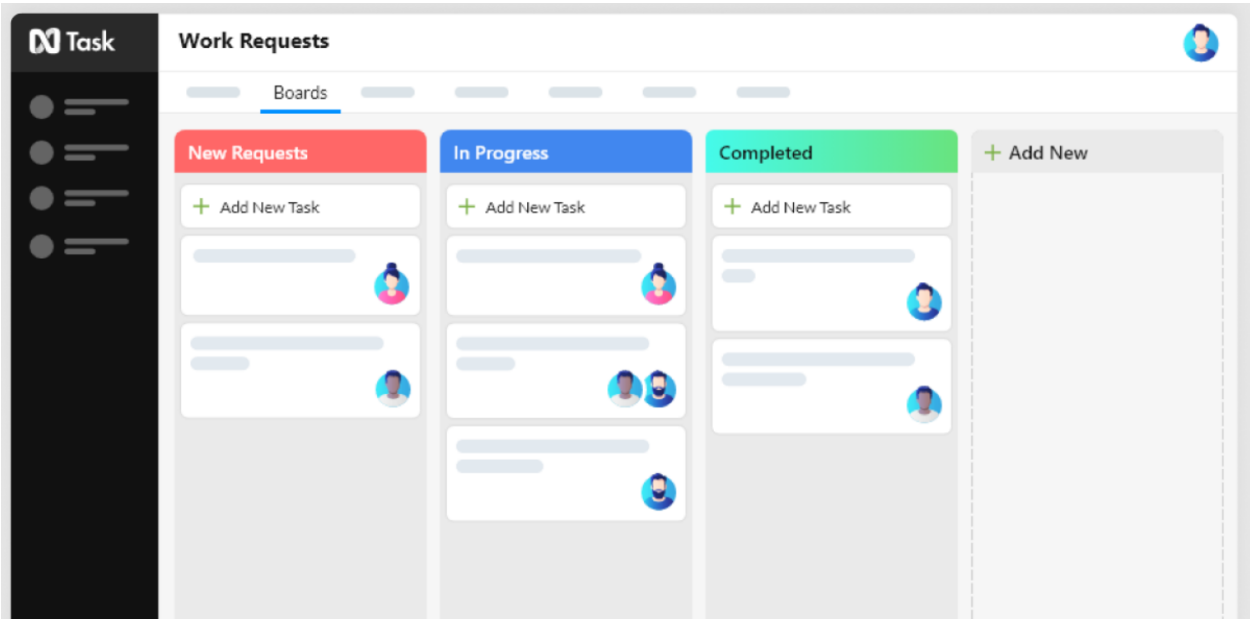

Рисунок 1.2 - Робочий єкран nTask

1.9. Аналіз Asana

Asana - це інтегрована платформа для управління проектами та завданнями, яка надає розширені можливості для організації роботи команди. Давайте розглянемо основні аспекти цього інструменту:

- Структура проектів: Asana дозволяє створювати проекти та підпроекти, встановлювати відносини між ними та створювати ієрархію завдань. Це сприяє організації проектів та кращому розподілу завдань між командою.
- Управління завданнями: інструмент надає можливість створювати завдання, визначати відповідальних, встановлювати дедлайни та пріоритети. Завдання можуть мати підзавдання, коментарі та вкладені файли для більшої деталізації та спілкування в команді.
- Календар та графіки: Asana має вбудований календар та графіки, що дозволяють відстежувати терміни виконання завдань та визначати завантаженість команди. Це сприяє плануванню та визначенню пріоритетів роботи.
- Спільна робота та комунікація: інструмент надає можливість коментувати завдання, спілкуватися з командою та обмінюватися файлами. Це сприяє ефективній комунікації та спільній роботі над завданнями.
- Інтеграції: Asana підтримує інтеграцію з іншими популярними інструментами, такими як Google Календар, Slack, Dropbox, Jira та інші. Це дозволяє зручно об'єднувати різні робочі процеси та забезпечувати зв'язок з іншими інструментами.
- Оглянувши функціональність Asana, можна зрозуміти, що цей інструмент є потужним рішенням для управління проектами та завданнями. Він надає широкий набір функцій, які можуть бути корисними для відділу

розробки, сприяючи організації роботи, спільній комунікації та відстеженню завдань. Однак, перед вибором певного інструменту, важливо зробити порівняльний аналіз з іншими аналогами та врахувати конкретні потреби та вимоги вашої команди (Рис 1.3).

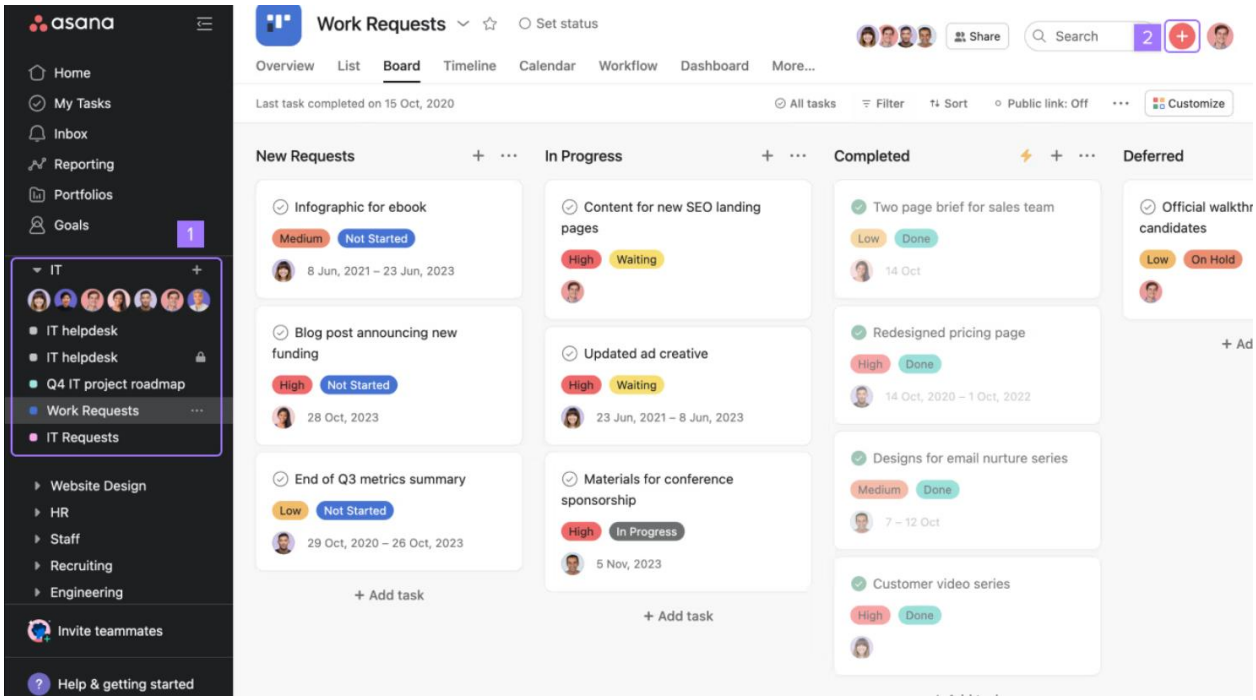

Рисунок 1.3 - Робочий єкран Asana

1.10. Порівняння систем-аналогів

Порівняємо Asana та nTask з точки зору їхньої придатності для відділу розробки і виявимо певні обмеження, які можуть зробити їх непідходящими в деяких випадках:

1. Обмежена функціональність: Хоча Asana та nTask надають базовий набір функцій для управління проектами та завданнями, вони можуть бути недостатніми для складних проектів або команд з високим рівнем технічної спеціалізації. Відділ розробки може потребувати більш продвинутих функцій, таких як інтеграція з розробницькими інструментами, контроль версій або спеціалізовані функції для керування кодом.

2. Відсутність спеціалізованих інструментів для розробників: Якщо відділ розробки працює з певними технологіями або має специфічні потреби, то Asana та nTask можуть бути менш придатними. Вони не надають спеціалізовані інструменти для розробників, такі як інтегровані IDE, системи контролю версій або спеціальні функції для веб-розробки.

3. Обмежені можливості відстеження часу та звітності: Якщо точне відстеження часу та звітності є критичними для відділу розробки, Asana та nTask можуть не надавати достатньо розширених можливостей. Вони можуть мають базові функції відстеження часу, але не пропонують додаткових інструментів, таких як автоматичний облік робочого часу, розподіл часу за проектами чи генерація детальних звітів.

4. Інтерфейс та користування: Обидва інструменти мають інтуїтивний і дружелюбний інтерфейс, що полегшує їх використання. Asana має більш витончений та сучасний дизайн, в той час як nTask пропонує більш простий та зрозумілий інтерфейс. Вибір між ними може залежати від особистих вподобань користувача.

5. Комунікація та спільна робота: Asana та nTask надають інструменти для комунікації та спільної роботи, такі як коментарі, обговорення та спільний доступ до файлів. Asana має більш розвинуті можливості для комунікації, такі як звернення до конкретних користувачів та відстеження обговорень. nTask зосереджується на базових можливостях спільної роботи, що може бути достатнім для деяких команд.

6. Інтеграції: Обидва інструменти підтримують інтеграцію з різними зовнішніми службами і інструментами, такими як Google Drive, Slack, GitHub тощо. Asana має більш широкий спектр інтеграцій та пропонує власний API для розробників. nTask також надає деякі популярні інтеграції, але може бути менш розширеним у цьому аспекті.

7. Цінова політика: Asana пропонує різні плани ціноутворення, включаючи безкоштовний план для невеликих команд. Вартість платних

планів Asana залежить від кількості користувачів та функціональних можливостей. nTask також має безкоштовний план та платні плани з додатковими функціями

1.11. Доцільність проєктування й розроблення системи

Доцільність проєктування й розроблення своєї системи управління проектами та робочим часом може бути обґрунтованою з кількох причин:

Специфічні потреби та вимоги: Компанія Lytvynov Production має унікальні потреби та вимоги щодо управління проектами й робочим часом, оскільки вона є аутсорсинговою компанією, що займається створенням та супроводом веб-сайтів. Готові рішення, такі як nTask або Asana, можуть не повністю відповідати їх специфічним потребам.

Контроль і конфіденційність: Розроблення власної системи дозволить компанії зберігати повний контроль над своїми даними та інформацією про проекти. Це особливо важливо для аутсорсингових компаній, де конфіденційність клієнтської інформації є важливою.

Гнучкість та налаштування: Розроблення власної системи дає можливість враховувати унікальні потреби та процеси компанії Lytvynov Production. Це дозволить налаштувати систему управління проектами та робочим часом так, щоб вона ідеально відповідала їх специфічним вимогам.

Інтеграція з внутрішніми системами: Власна система може бути легко інтегрована з існуючими внутрішніми системами, які вже використовуються в компанії Lytvynov Production. Це дозволить забезпечити безперебійний обмін даними та інформацією між різними системами.

У результаті, проєктування й розроблення власної системи управління проектами та робочим часом може бути доцільним для компанії Lytvynov Production, оскільки це дозволить відповідати їх специфічним потребам, забезпечити контроль і конфіденційність, мати гнучкість налаштування та забезпечити інтеграцію з внутрішніми системами.

1.12. Концептуальна модель системи

Система управління проектами та робочим часом, доступна у вигляді веб-додатку для робітників, є надійним та ефективним інструментом для планування, виконання та контролю завдань та робочого часу. Вона має ряд складових, які забезпечують зручну роботу та точне слідкування за робочим процесом.

Авторизація та персоналізація є важливим етапом використання системи. Кожен робітник має свій власний особистий обліковий запис, який дозволяє йому безпечно увійти в систему та мати доступ до своїх завдань та робочого часу. Безпека та конфіденційність даних є головним пріоритетом, а персоналізовані налаштування дозволяють кожному користувачеві налаштувати систему під свої потреби.

Управління задачами є ключовою функцією системи. Робітники можуть створювати нові завдання, редагувати існуючі та видаляти їх за потребою. Кожне завдання має назву, опис, визначений термін виконання та відповідального виконавця. Крім того, робітники можуть прикріплювати файли до завдань, встановлювати пріоритети та стежити за прогресом викон Ця система управління проектами та робочим часом є незамінним інструментом для організації робочих процесів і підвищення продуктивності. Вона надає робітникам широкий набір функцій та можливостей для ефективного планування, виконання та контролю завдань.

Завдяки цій системі, робітники можуть з легкістю створювати нові проекти та завдання, встановлювати їм терміни виконання та пріоритети. Вони можуть також прикріплювати до завдань необхідні файли та додаткову інформацію. Робітники мають змогу стежити за прогресом виконання кожного завдання та вчасно вносити необхідні зміни.

Окрім того, система забезпечує точний контроль над робочим часом. Робітники можуть відстежувати час, витрачений на кожне завдання, а також

встановлювати нагадування та таймери для керування своїм робочим графіком. Це допомагає уникнути зайвих перерв і зосередитися на важливих завданнях.

Завдяки високій функціональності та інтуїтивному користувацькому інтерфейсу, ця система сприяє більшій організованості та ефективності робочих процесів. Робітники можуть більш точно планувати свій робочий час, віддавати пріоритети важливим завданням та досягати успіху в реалізації проектів.

Таким чином, ця система управління проектами та робочим часом стає незамінним помічником для робітників, допомагаючи їм організувати робочі процеси, планувати завдання та керувати часом ефективно і результативно.ання.

РОЗДІЛ 2. ТЕХНІЧНЕ ЗАВДАННЯ НА ПРОЄКТУВАННЯ

2.1. Загальні положення

Веб-додаток, призначений для організації робочого часу під час розробки проектів, є потужним інструментом, що допомагає ефективно управляти проектами та раціонально використовувати час. Його функціонал розширюється на низку корисних можливостей, які полегшують процес роботи.

Цей веб-додаток дозволяє користувачам створювати нові проекти та додавати завдання, визначаючи для них терміни виконання та пріоритети. Він також надає можливість зручного відстеження часу, витраченого на кожне завдання, що дозволяє здійснювати ефективний контроль над прогресом проекту.

Окрім свого основного функціоналу, цей веб-додаток може використовуватись і для організації особистого часу. Це означає, що користувачі можуть використовувати його для створення особистих завдань, встановлення нагадувань, ведення щоденників або планування подій. Така гнучкість дозволяє ефективно організовувати час як у сфері роботи, так і у особистих справах.

Завдяки цим функціям, веб-додаток стає незамінним помічником у плануванні та виконанні завдань. Він допомагає забезпечити структурованість та організованість робочого процесу, що сприяє підвищенню продуктивності та досягненню успіху у роботі.

2.2. Призначення і цілі розробки

Призначення розробки веб-додатка полягає в тому, щоб створити зручний та централізований інструмент для роботи з розробками проектів. Веб-додаток допомагатиме розробникам ефективно організовувати свій робочий час, планувати завдання, ставити пріоритети, встановлювати терміни

та відстежувати їх виконання. Він надасть засоби для розподілу завдань між командами, спільної роботи та спілкування між учасниками проекту.

Основна мета розробки веб-додатка - забезпечити розробникам зручні інструменти для керування процесом розробки, виявлення потенційних проблем та ефективного рішення їх. Додаток буде забезпечувати функціонал для організації інформації про проекти, включаючи опис завдань, виконавців, терміни, пріоритети, коментарі та звіти про прогрес роботи.

Крім робочих функцій, веб-додаток також надає можливість користувачам планувати та керувати своїм особистим часом. Користувачі зможуть створювати особисті завдання, встановлювати нагадування, відслідковувати свій прогрес та отримувати повідомлення про наближення термінів. Це дозволить їм бути організованими, управляти своїм часом ефективно та досягати своїх особистих цілей.

Узагалі, розроблений веб-додаток буде спрямований на поліпшення продуктивності розробників, забезпечення якісного керування проектами та робочим часом, а також покращення організації особистого часу користувачів. Він надасть необхідні інструменти та функціональні можливості для ефективної роботи та досягнення успіху в професійній та особистій сфері.

2.3. Характеристика об’єкту автоматизщації

Lytvynov Production - це висококваліфікована аутсорсингова компанія, що спеціалізується на розробці та супроводі веб-сайтів для своїх клієнтів. Завдяки багатому досвіду в галузі веб-розробки, компанія має надійну репутацію та здатність надавати професійні послуги найвищої якості.

Команда Lytvynov Production складається з талановитих та досвідчених фахівців, які працюють в різних напрямках. Веб-розробники володіють глибокими знаннями та вміннями, щоб створювати функціональні та естетичні веб-сайти. Дизайнери розробляють привабливі інтерфейси, враховуючи

вимоги клієнтів та останні тенденції веб-дизайну. Тестувальники виконують ретельне тестування, щоб забезпечити високу якість та безпеку роботи веб-сайтів.

Команда Lytvynov Production докладає всіх зусиль, щоб забезпечити успішне виконання проектів. Вони працюють у тісному співробітництві зі своїми клієнтами, розуміючи їхні потреби та бажання, а також надають консультації та рекомендації для досягнення найкращих результатів.

Lytvynov Production володіє надійною інфраструктурою та сучасними інструментами розробки, що дозволяє їм ефективно виконувати проекти та забезпечувати своїх клієнтів зручними та інноваційними рішеннями. Компанія гордиться своєю здатністю дочасно завершувати проекти та надавати надійну підтримку після впровадження.

Звертаючись до Lytvynov Production, клієнти можуть бути впевнені в якісному та професійному виконанні робіт з розробки та супроводу веб-сайтів, що відповідають їхнім потребам та вимогам.

2.4. Вимоги до системи

У вимоги до системи входять такі пункти:

1. Функціональність: Система повинна забезпечувати можливість організації та керування робочим часом розробників, включаючи створення та відстеження завдань, планування ресурсів, контроль прогресу та виконання проектів.

2. Взаємодія з користувачем: Система має мати інтуїтивний та зручний інтерфейс, що дозволяє користувачам легко навігувати та виконувати необхідні дії. Вона повинна бути доступна через веб-браузер та мобільні пристрої для зручності використання.

3. Безпека: Система повинна забезпечувати захист конфіденційної інформації та даних користувачів, використовуючи відповідні механізми аутентифікації, авторизації та шифрування.

4. Масштабованість: Система повинна бути гнучкою та масштабованою, здатною впоратися з ростом обсягу роботи та кількості користувачів. Вона має мати ефективну архітектуру, що дозволяє розширювати функціонал та оптимізувати продуктивність.

5. Надійність: Система повинна бути стійкою до помилок та збоїв, забезпечуючи надійну роботу без втрати даних або порушення функціональності. Вона має мати вбудовані механізми резервного копіювання, відновлення та моніторингу для забезпечення безперебійної роботи.

6. Сумісність: Система повинна бути сумісною з різними операційними системами, веб-браузерами та мобільними платформами, щоб користувачі могли використовувати її на різних пристроях та платформах.

7. Підтримка: Система повинна мати механізми для надання технічної підтримки користувачам, включаючи документацію, онлайн-допомогу, чат-підтримку або інші доступні канали спілкування.

8. Вартість: Розробка та підтримка системи повинна бути ефективною з точки зору вартості, з урахуванням бюджетних обмежень та економічної доцільності.

2.5. Вимоги до структури системи

У вимоги до структури системи входять такі пункти:

9. Інтуїтивний і зручний інтерфейс: Структура додатку повинна бути логічною та інтуїтивно зрозумілою для користувача. Навігація має бути зручною, а розташування елементів — логічним, щоб користувачі легко знаходили необхідні функції.

10. Розширюваність: Структура повинна бути модульною, щоб нові функціональні можливості можна було додати без значних змін у вже

існуючий код. Це дозволить легко розширювати додаток у майбутньому, додавати нові функції або змінювати існуючі.

11. Безпека: Структура повинна забезпечувати безпеку даних та доступу до системи. Розробка повинна включати в себе заходи безпеки, такі як аутентифікація та авторизація користувачів, захист від атак, шифрування даних тощо.

12. Швидкодія та масштабованість: Структура повинна бути оптимізована для швидкого відгуку та ефективної роботи. Додаток повинен бути масштабованим, здатним працювати зі зростанням обсягу даних та користувачів.

13. Інтеграція з іншими системами: Якщо необхідно, структура повинна забезпечувати можливість інтеграції з іншими системами або зовнішніми сервісами, які використовуються компанією. Це може включати інтеграцію з базою даних, зовнішніми API, сервісами хмарного зберігання тощо.

14. Мобільна сумісність: У випадку потреби додаток повинен бути адаптований для мобільних пристроїв, забезпечуючи зручне використання і на мобільних пристроях. Це може включати розробку респонсивного дизайну або створення окремого мобільного додатку.

2.6. Вимоги до функцій системи

У вимоги до функцій входять такі пункти:

1. Авторизація та ідентифікація: Система повинна мати механізми для реєстрації та авторизації робітника, щоб визначити його ідентичність та забезпечити безпеку доступу до системи.

2. Особистий профіль: Система має надавати можливість створення та налаштування особистого профілю робітника, де він може вказати свої персональні дані, навички, контактну інформацію та інші важливі деталі.

3. Задачі та проекти: Система повинна дозволяти робітникові створювати та керувати своїми задачами та проектами. Вона має мати можливість створювати нові задачі, присвоювати їм терміни виконання, встановлювати пріоритети та відстежувати їх прогрес.

4. Календар та графік: Система повинна мати функціонал календаря, що дозволяє робітникові планувати свій робочий час, визначати дати та години для виконання задач, встановлювати нагадування та інтегрувати графік роботи.

5. Відстеження часу: Система має надавати можливість робітнику слідкувати за своїм часом роботи, включаючи витрачений час на кожну задачу або проект. Вона повинна мати зручний інтерфейс для початку та закінчення роботи над конкретними завданнями.

6. Сповіщення та нагадування: Система повинна мати можливість надсилати робітникові сповіщення та нагадування про невиконані завдання, наближення термінів виконання, а також інші важливі події чи повідомлення.

7. Аналітика: Система має надавати зручні інструменти для аналізу продуктивності робітника. Вона повинна відображати статистику виконаних задач.

8. Налаштування та інтеграція: Система повинна мати можливість налаштування параметрів відповідно до потреб робітника, а також підтримувати інтеграцію з іншими інструментами або сервісами, які використовуються компанією або робітником.

2.7. Вимоги до збереження даних

У вимоги до збереження даниї входять такі пункти:

1. Коректність формату: Всі дані, збережені у форматі JSON, повинні відповідати стандартному синтаксису JSON і мати правильну структуру. Це забезпечить правильне збереження і обробку даних.

2. Безпека: Необхідно забезпечити захист даних, збережених у форматі JSON, від несанкціонованого доступу та модифікації. Це може включати застосування механізмів шифрування, контролю доступу і автентифікації користувачів.

3. Резервне копіювання: Дані у форматі JSON повинні регулярно резервно копіюватися для захисту від можливої втрати або пошкодження. Це дозволить відновити дані в разі непередбачуваних ситуацій.

4. Цілісність даних: Система повинна забезпечувати цілісність даних у форматі JSON, переконуючись, що дані не пошкоджені або втрачені під час збереження або передачі. Це можна досягти за допомогою механізмів перевірки цілісності даних та контролю цілісності під час операцій збереження та витягування даних.

5. Швидкість доступу: Система повинна забезпечувати швидкий доступ до даних у форматі JSON для ефективного використання та обробки. Це можна досягти оптимізацією операцій збереження, витягування та маніпуляції даними в форматі JSON.

2.8. Вимоги до надійності

У вимоги до надійності входять такі пункти:

1. Стабільність: Додаток повинен працювати без збоїв і відмов, забезпечуючи безперебійну роботу для користувачів. Необхідно виявляти і усувати можливі помилки та проблеми швидко і ефективно.

2. Безпека: Додаток повинен мати вбудовані заходи безпеки для захисту даних користувачів, запобігання несанкціонованому доступу і мінімізації ризиків кібератак. Це може включати застосування механізмів аутентифікації, авторизації, шифрування даних та захисту від вразливостей.

3. Швидкість та продуктивність: Додаток повинен працювати швидко і ефективно, надаючи користувачам миттєвий доступ до функціоналу.

Відповідність вимогам продуктивності дозволить запобігти затримкам і непередбаченим відмовам.

4. Резервне копіювання та відновлення: Важливо мати механізми резервного копіювання та відновлення даних, щоб у разі втрати чи пошкодження даних була можливість відновити їх до попереднього стану. Це допоможе запобігти втраті важливої інформації та забезпечить безперебійну роботу додатка.

5. Моніторинг та журналювання: Для забезпечення надійності додатка необхідно мати систему моніторингу, яка відстежує працездатність, виявляє можливі проблеми і сповіщає адміністраторів про них. Також важливо вести журнал подій для аналізу і відстеження дій користувачів та виявлення потенційних проблем.

6. Оновлення і підтримка: Для забезпечення надійності веб-додатка важливо мати систему оновлень і підтримки, яка забезпечує вчасне виправлення помилок, вдосконалення функціоналу і підтримку нових версій залежностей.

2.9. Вимоги до лінгвістичного забезпечення

У вимоги до лінгвістичного забезпечення входять такі пункти:

1. Валідація та коректори: Додаток повинен мати вбудовані механізми валідації текстових полів і коректори для покращення якості введення ко ристувачів. Це може включати перевірку правопису, синтаксичних помилок, валідність формату електронної пошти і інші перевірки для забезпечення введення коректних даних.

2. Підтримка мовних правил: Додаток повинен враховувати мовні правила і конвенції різних мов, такі як написання, правила граматики, розділові знаки тощо. Це допоможе забезпечити правильну відображення і обробку тексту відповідно до мовних особливостей.

3. Підтримка міжнародних стандартів: Додаток повинен дотримуватися міжнародних стандартів, що визначають правила для лінгвістичного забезпечення. Це може включати стандарти кодування символів, перекладу, форматування дат і часу, міжнародних стандартів інтернаціоналізації та локалізації.

2.10. Вимоги до програмного забезпечення

У вимоги до програмного забезпечення входять такі пункти:

1. Додаток повинен бути сумісним з різними веб-браузерами і операційними системами, такими як Windows, macOS, Linux і мобільні платформи, забезпечуючи широку доступність для користувачів.

2. Додаток повинен працювати стабільно і безперебійно, запобігаючи збоїв і помилок. Він повинен вміти ефективно обробляти непередбачувані ситуації і відновлюватися після відмов.

3. Додаток повинен мати вбудовані механізми безпеки для захисту від несанкціонованого доступу, зламів і витоку конфіденційної інформації. Це включає захист від SQL-ін'єкцій, кросс-сайтових скриптів, аутентифікації і авторизації користувачів.

4. Додаток повинен бути швидким і відзначатися швидким відгуком на дії користувачів. Використання оптимізованих алгоритмів, кешування даних і ефективного використання ресурсів сервера допоможуть забезпечити оптимальну продуктивність.

5. Додаток повинен бути готовим до масштабування, здатним працювати зі зростаючою кількістю користувачів і обсягом даних. Він повинен використовувати архітектурні підходи, які дозволяють гнучко розширювати його можливості і продуктивність.

6. Додаток повинен мати інтуїтивно зрозумілий і зручний інтерфейс користувача, що дозволяє легко навігувати та виконувати зав дання без необхідності великої кількості інструкцій і навчання.

7. Сумісність: Додаток повинен бути сумісним з різними сторонніми сервісами і API, які можуть використовуватися для інтеграції з іншими системами.

8. Підтримка: Додаток повинен мати механізми підтримки, які дозволяють користувачам звертатися за допомогою, відправляти запити і отримувати відповіді. Це може включати систему тікетів, форуми спільноти або онлайн-чат.

2.11. Мінімальні системні та апаратні вимоги

У мінімальні системні вимоги входять:

1. Операційна система: Asana підтримується на різних операційних системах, включаючи Windows, macOS, Linux, а також мобільні платформи Android і iOS. Для оптимального використання рекомендується використовувати останні версії операційних систем, які підтримуються розробником.

2. Веб-браузер: Asana сумісна з багатьма популярними веб-браузерами, такими як Google Chrome, Mozilla Firefox, Safari і Microsoft Edge. Рекомендується використовувати останні версії браузерів для забезпечення найкращої продуктивності та безпеки.

3. Інтернет-з'єднання: Для використання Asana необхідне постійне інтернет-з'єднання. Це дозволяє здійснювати доступ до системи та зберігати дані в хмарному середовищі, що дозволяє користувачам працювати з Asana з будь-якого місця і пристрою.

Мінімальні апаратні вимоги для використання Asana на комп'ютері зазвичай включають:

Таблиця 2.1 – Мінімальні апаратні вимоги до системи

Характеристика	**Вимога**
Операційна система	Windows 7 або новіша, macOS 10.13 або новіша, Linux
Процесор	1,5 ГГц або швидше
Оперативна пам'ять	4 ГБ
Відеокарта	сумісна з DirectX 10
Дисковий простір	Стабільне інтернет-з'єднання
Монітор	Мінімальна роздільна здатність 1024x768 пікселів
Веб-браузер	Mozilla Firefox, Safari, Microsoft Edge, Google Chrome та інші останні версії

Це мінімальні вимоги, і для оптимальної продуктивності може знадобитися більш потужне обладнання. (Табл. 2.2)

Таблиця 2.2 – Оптимальні апаратні вимоги до системи

Характеристика	**Вимога**
Операційна система	Windows 7 або новіша, macOS 10.13 або новіша, Linux
Процесор	Intel Core i5 або еквівалентний, 2,5 ГГц або швидше
Оперативна пам'ять	8 ГБ або більше
Відеокарта	сумісна з DirectX 11 або новішою версією
Дисковий простір	10 ГБ вільного місця на жорсткому диску або більше
Монітор	1920x1080 пікселів або вище
Веб-браузер	Mozilla Firefox, Safari, Microsoft Edge, Google Chrome та інші останні версії

РОЗДІЛ 3 КОМПЛЕКС ЗАДАЧ АВТОМАТИЗАЦІЇ

3.1. Інформаційне забезпечення системи

Інформаційне забезпечення системи є критично важливим аспектом для безпечної та ефективної роботи веб-додатку. Воно включає в себе низку заходів, процедур і політик, спрямованих на захист, збереження та доступність інформації. У контексті цього веб-додатку, інформаційне забезпечення повинно враховувати такі аспекти, як зберігання даних, керування доступом та захист від кібератак.

По-перше, зберігання даних є важливим елементом інформаційного забезпечення. Веб-додаток повинен мати надійний механізм збереження даних, що забезпечує цілісність та конфіденційність інформації. Для цього можуть використовуватися різні методи, включаючи бази даних, файлові системи або зовнішні сервіси збереження даних. Важливо забезпечити резервне копіювання даних, щоб у разі випадкової втрати або пошкодження інформац ії була можливість відновлення.

По-друге, керування доступом є необхідним для забезпечення конфіденційності та обмеження доступу до інформації лише авторизованим користувачам. Система повинна мати механізм аутентифікації та авторизації, що дозволяє визначати рівні доступу користувачів до функціоналу та даних. Керування доступом може включати в себе використання паролів, ролей користувачів, двофакторної аутентифікації та інших методів ідентифікації та авторизації.

По-третє, система повинна бути захищеною від кібератак і небажаного доступу до даних. Це означає використання захисних заходів, таких як фаєрволи, шифрування даних, системи виявлення вторгнень та регулярне оновлення програмного забезпечення. Крім того, важливо встановити механізми моніторингу та журналювання, які дозволяють виявляти та

аналізувати недоліки в системі, некоректну поведінку та потенційні загрози безпеці.

3.2. Алгоритмізація та реалізація комплексу задач автоматизації.

Функціональні вимоги до системи автоматизації включають створення списку задач для ефективного управління роботою, встановлення строків виконання для кожного завдання, відстеження прогресу роботи та можливість пріоритезації завдань залежно від їх важливості.

У процесі проектування архітектури додатку розробляються необхідні модулі, включаючи створення інтуїтивного та зручного користувацького інтерфейсу, який спрощує взаємодію з системою та забезпечує зручну навігацію та виконання завдань.

Тестування системи є важливою складовою для забезпечення якості та надійності її роботи. Воно включає функціональне тестування, під час якого перевіряються правильність функцій системи, тестування взаємодії з користувачем, щоб переконатися, що користувачі можуть зручно використовувати систему, а також тестування продуктивності, яке оцінює швидкодію та відповідність системи поставленим вимогам.

Після успішного тестування система впроваджується в робоче середовище, а користувачам надається необхідна підтримка. Це включає надання інструкцій щодо використання системи, вирішення технічних питань та вдосконалення системи на основі отриманого фідбеку від користувачів. Підтримка також може включати надання додаткових навчальних матеріалів та регулярне оновлення системи для поліпшення її функціональності та забезпечення задоволення потреб користувачів.

3.3. Інструкція користувача.

Система є веб-додатком, створеним з метою надання зручного та ефективного інструменту для керування проектами.Веб-додаток має назву

"ProjectSeek" і надає широкий набір функціональності для ефективного управління проектами.

1) При вході на веб-додаток "ProjectSeek", користувач автоматично переадресовується на початкову сторінку, де надається загальна інформація про функціональні можливості та основні переваги системи. На цій сторінці користувач знайомиться з основними концепціями та принципами роботи "ProjectSeek" (Рис. 3.1).

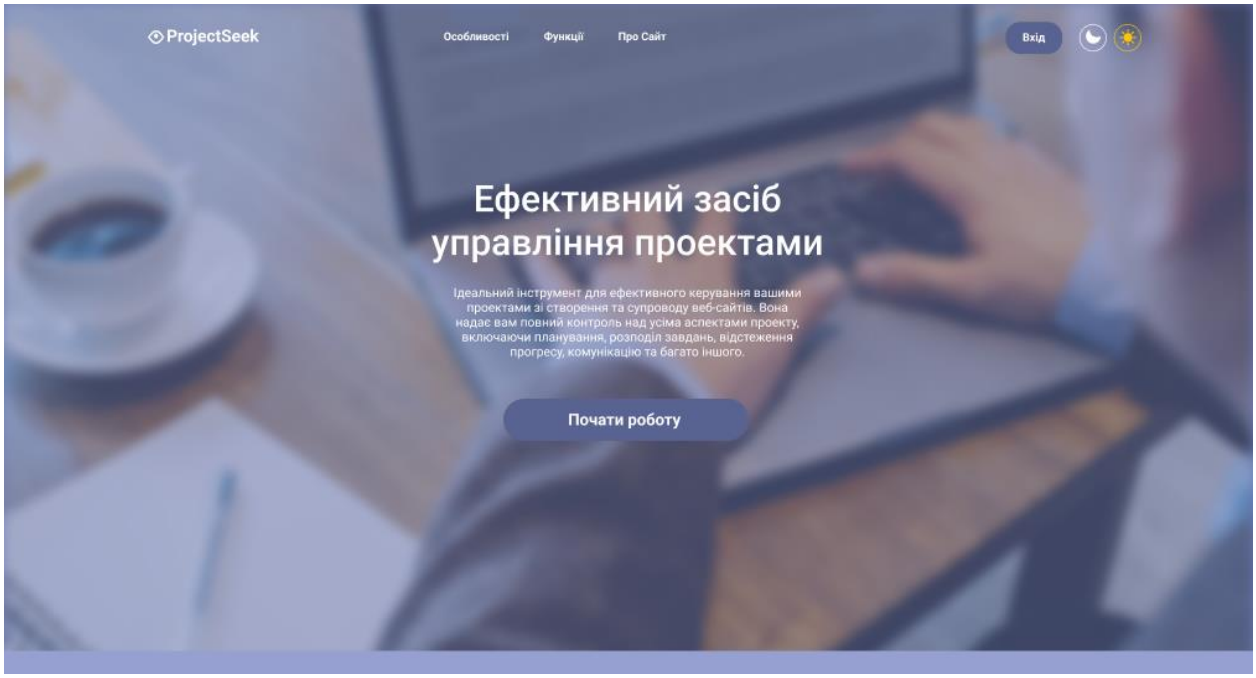

Рисунок 3.1 – Головна сторінка додатку

2) При натисканні кнопки створити профіль, на сторінці з'являється вікно на якому користувачу дається вибір: зайна на вже створений обліковий запис, або створити (Рис.3.2).

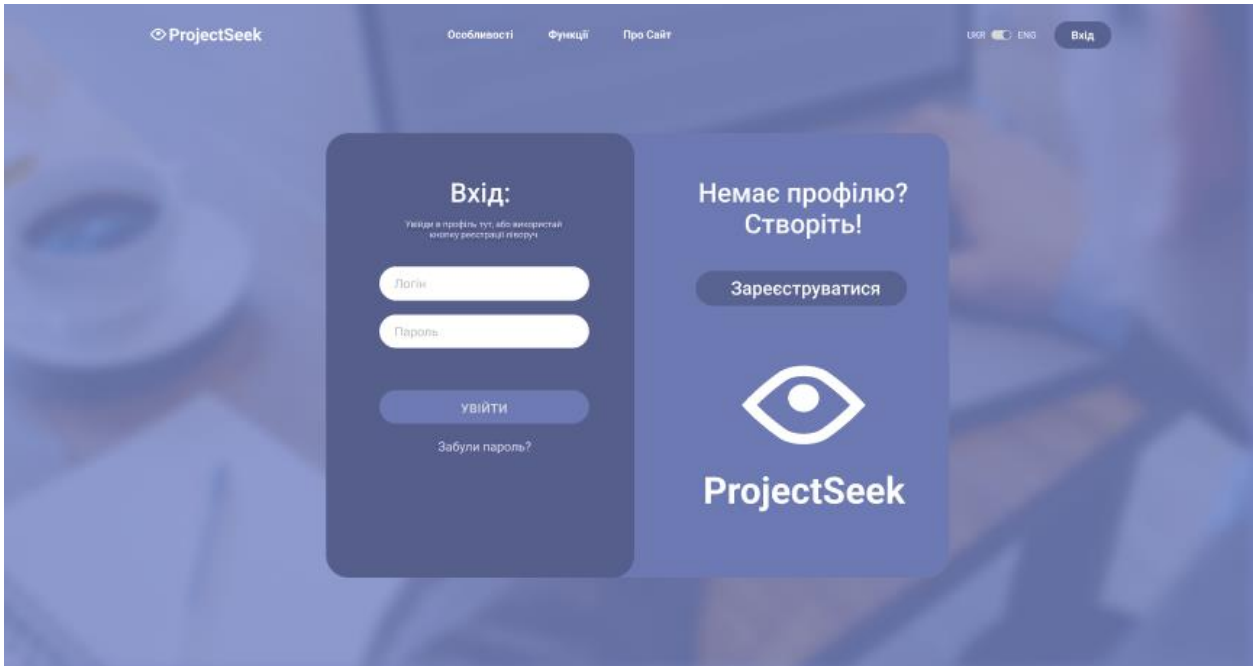

Рисунок 3.2 – Вікно входу

3)При натисканні кнопки “Зареэструватися”, користувач перенаправляється на окрему сторінку для процесу реєстрації. На цій сторінці користувачі можуть ввести свої особисті дані, такі як ім'я, електронну пошту та пароль, щоб створити обліковий запис у системі (Рис.3.3).

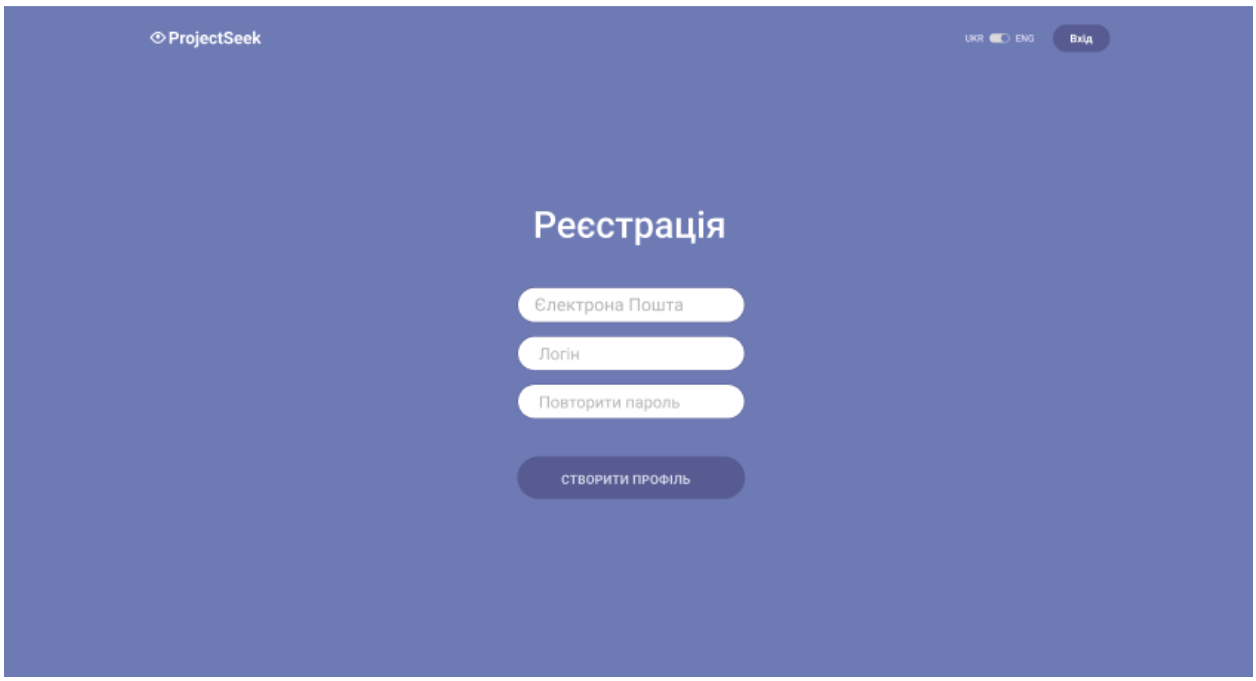

Рисунок 3.3 – Сторінка реєстрації

4) Після того, як користувач успішно увійшов у свій обліковий запис або зареєструвався на веб-додатку "ProjectSeek", він автоматично перенаправляється на головну сторінку сайту. Головна сторінка пропонує користувачеві зручний та організований перегляд всіх його задач.

На головній сторінці "ProjectSeek" знаходиться список усіх задач, які розподілені на три стовпчика, які показують стан виконання конкретної задачі. Кожна задача відображається у вигляді окремого блоку, де вказана назва, та дата створення задачі. Користувач може легко проглядати список задач, оцінювати їх пріоритетність та оновлювати їх стан (Рис. 3.4).

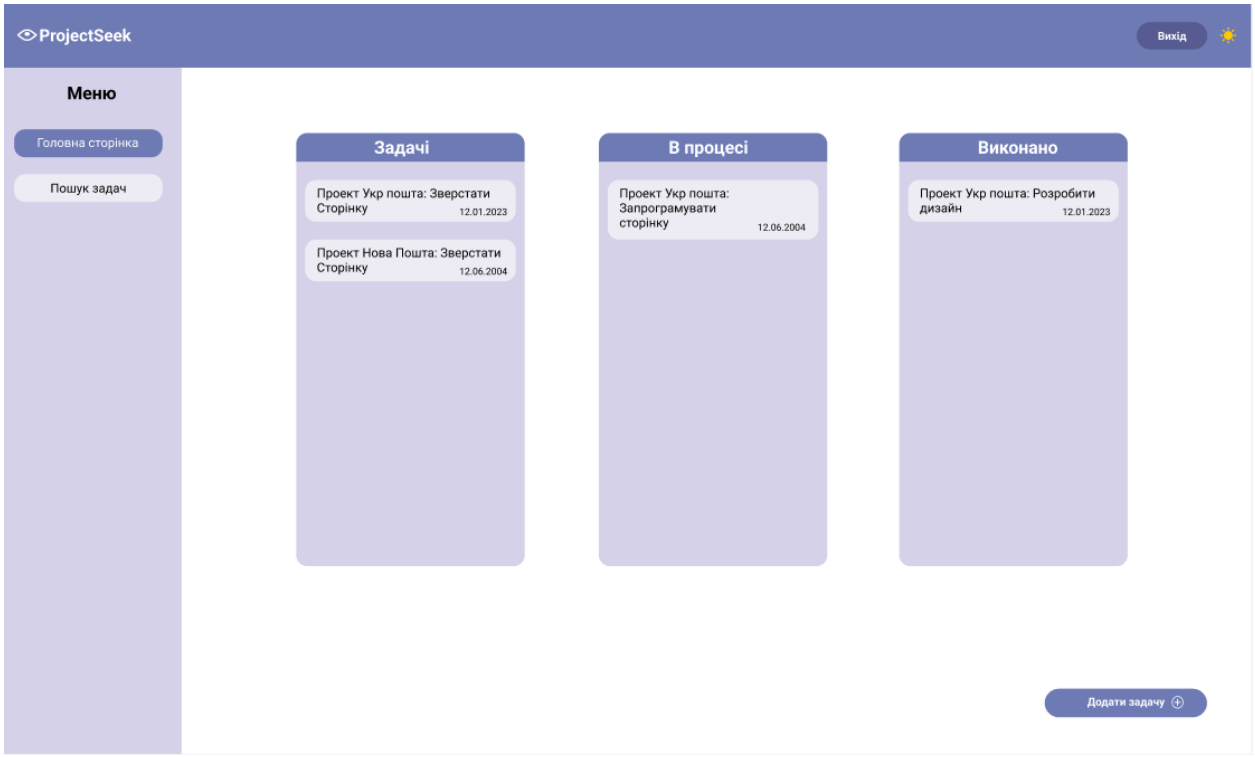

Рисунок 3.4 – Головна сторінка веб-додатку

5) Перейшовши на сторінку списка задач, користувач може увімкнути увімкнути темну тему, натиснувши спеціальну кнопку у верхній частині єкрану (Рис.3.5).

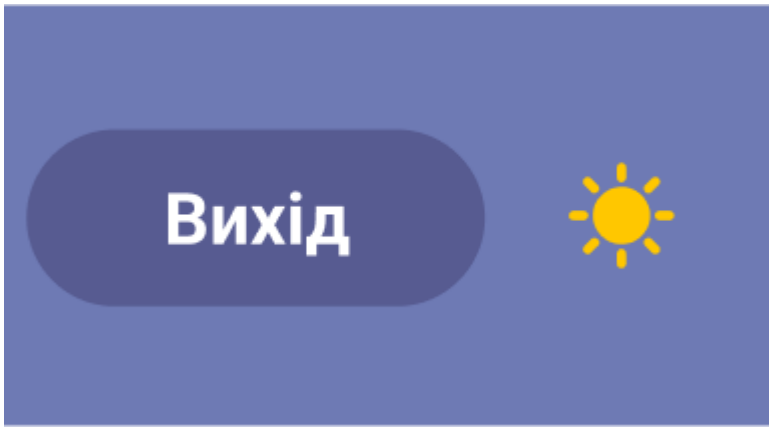

Рисунок 3.5 – Кнопка теми

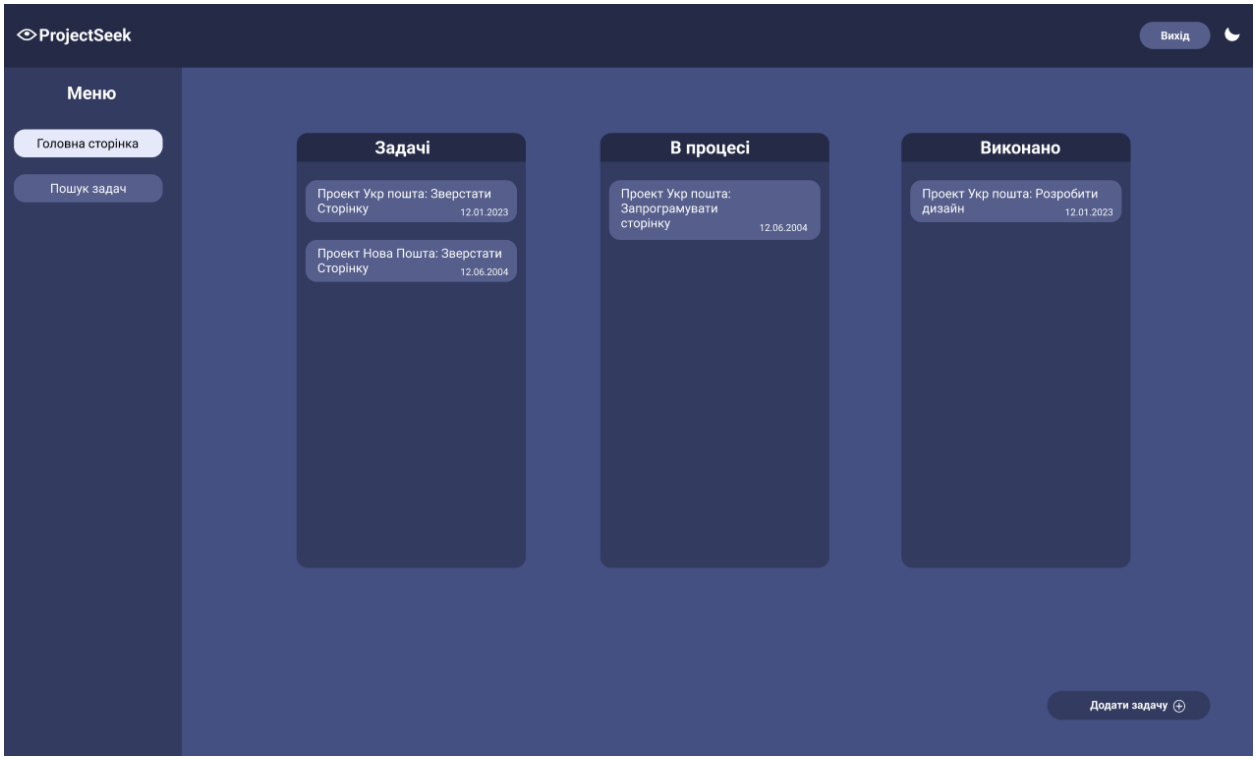

Рисунок 3.6 – Темна тема

6) Після входу в обліковий запис, користувач може використати також функцію пошуку конкретної задачі, натиснувши на спеціальну кнопку в правій частині єкрану (Рис. 3.6, 3.7) .

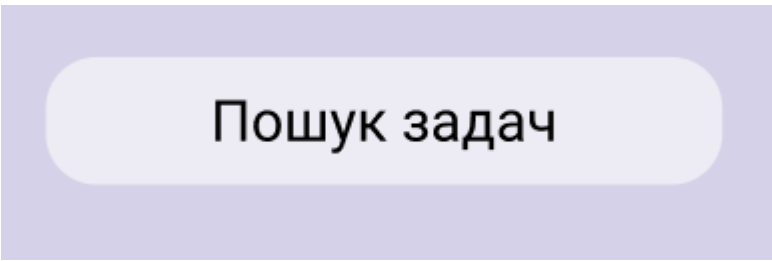

Рисунок 3.6 – Кнопка пошуку

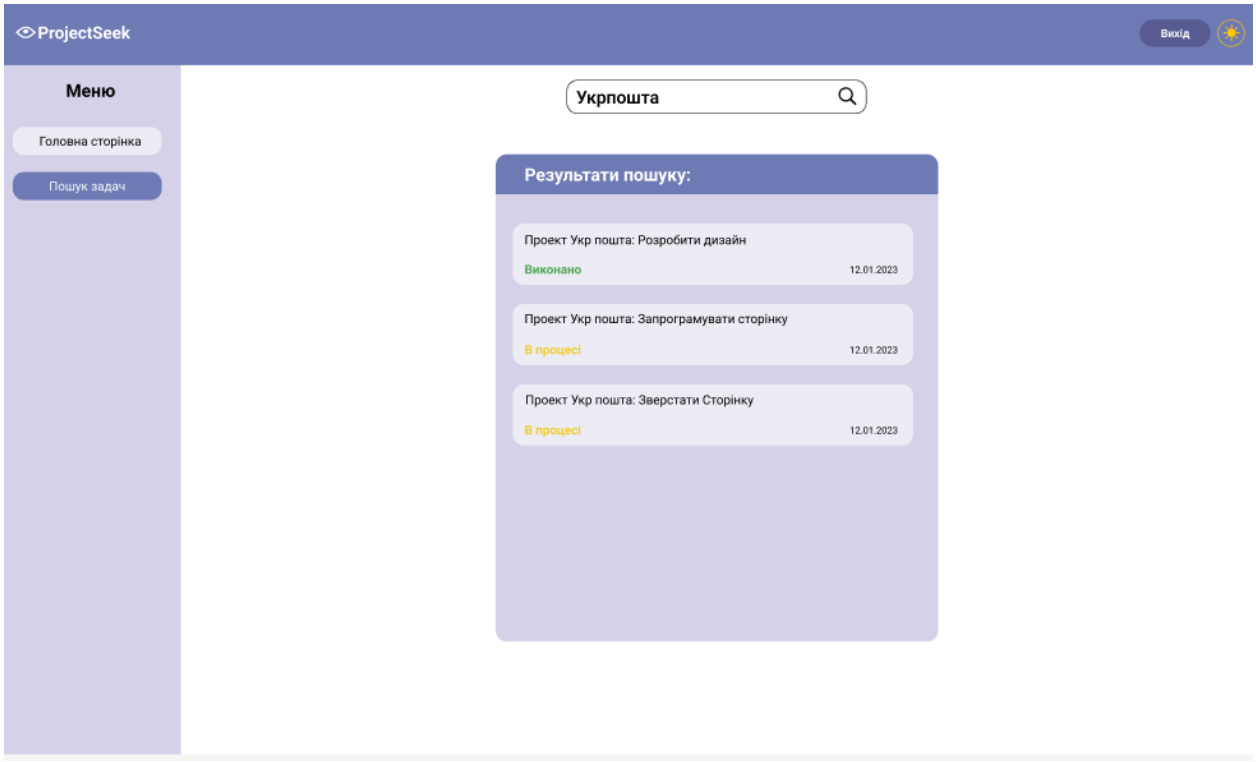

Рисунок 3.7 – Результати пошуку

Після використання кнопки Пошук задач, користуваич перейде на іншу сторінку, де після введних даних, в результатах пошуку (Рис.3.7) отримає задачі із такою назвою, а також інформацієь про стан їх виконання.

7) На головній сторінці користувач може також створити нову задачу, нажав на спеціальну кнопку внизу єкрану (Рис.3.8).

Рисунок 3.8 – Кнопка створення задачі

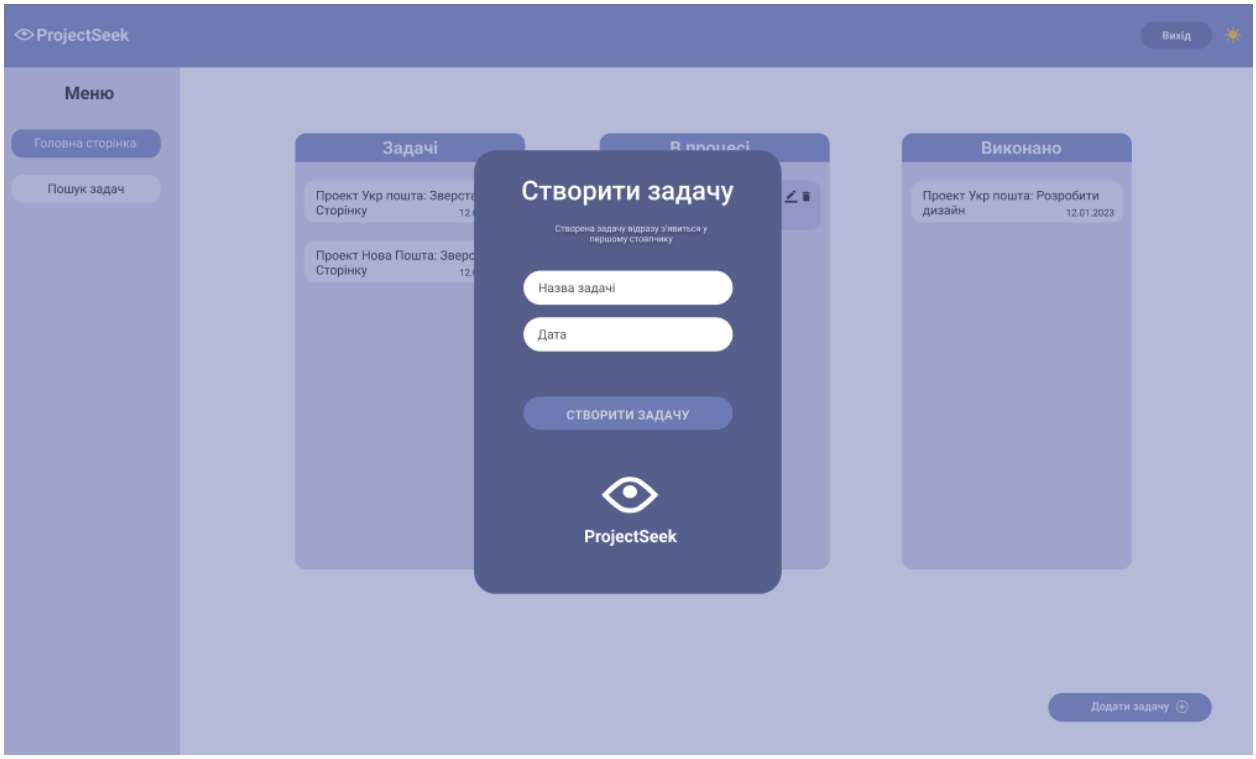

Рисунок 3.9 – Вікно створення задачі

Після створення задачі, вона автоматично переміщюється уперший стовпчик “Задачі”.

8) Для подальшої взаємодії, користувач може навести курсор миші на будь-яку задачу, для отримання додаткових функції, таких як: видалення та зміна назви задачі (Рис.3.10).

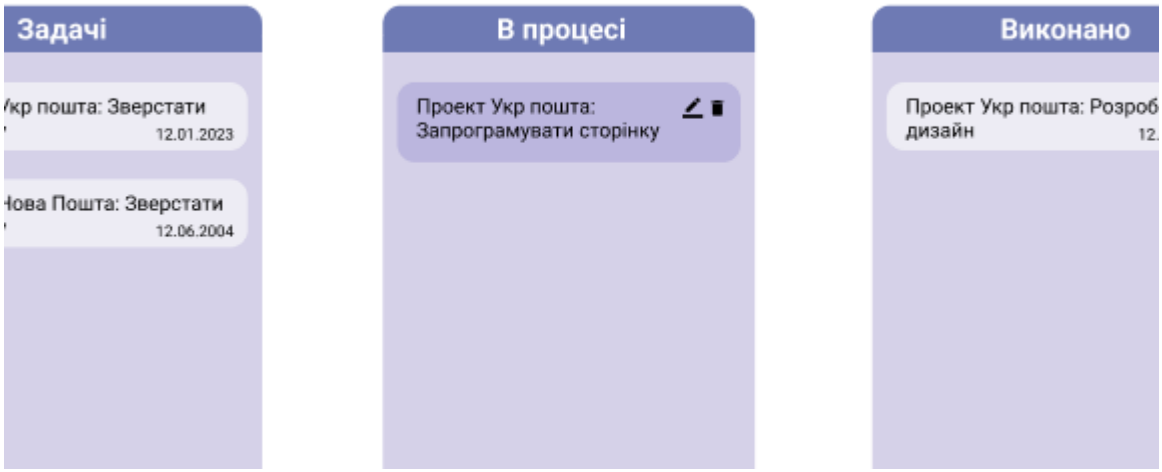

Рисунок 3.10 – Виділена задача

Також користувач може швидко перенести задачу із одного стовпця в інший. Для цього треба навести курсор на задачу, затиснути ліву кнопку миші, та перенести її в інший стовпець, після чого стан задачі буде змінений.

3.4. Технічне та системне забезпечення розробки.

В розробці системи були використані такі технології як:

- HTML/CSS(SCSS)
- JavaScript
- TypeScript(Angular 16)
- TailwindCss
- JsonServer

Для розробки інтерфейсу веб-додатку були використані HTML та CSS, включаючи препроцесор SCSS для полегшення процесу стилізації та забезпечення більшої гнучкості. Додатково, JavaScript використовувався для динамічної поведінки сторінки, такої як обробка подій та взаємодія з сервером.

Однак, для більш структурованої та масштабованої розробки, використання TypeScript було важливим аспектом проекту. TypeScript, який базується на JavaScript, дозволяє використовувати сильну типізацію та забезпечує більшу надійність і зрозумілість коду. Angular 16 був вибраний як основний фреймворк для створення веб-додатку, надаючи розширені можливості для розробки інтерфейсу користувача та управління станом додатку.

Щодо стилізації, використання Tailwind CSS дозволило швидко та ефективно створити привабливий та сучасний дизайн. Tailwind CSS пропонує набір готових класів, які дозволяють швидко застосовувати стилі до елементів і створювати належний вигляд.

У зв'язку з необхідністю зберігання та отримання даних, використання JsonServer було вирішальним. JsonServer забезпечує легку налаштування

імітації сервера, що дозволяє зберігати та отримувати дані в форматі JSON. Це дозволяє розробникам працювати з даними без реального сервера та спрощує процес розробки.

3.5. Обґрунтування вибору технічних засобів

У розробці веб-додатку "ProjectSeek" були обрані певні технології з метою забезпечення оптимального результату та задоволення вимог проекту. Кожна з цих технологій має свої унікальні переваги і внесок у розробку системи.

HTML/CSS(SCSS) є основою для будь-якого веб-додатку. HTML використовується для створення структури та розмітки сторінок, а CSS дозволяє стилізувати їх зовнішній вигляд. Використання препроцесора SCSS (Sass) дозволяє писати більш структурований та модульний CSS код, що полегшує розробку та підтримку стилів.

JavaScript є мовою програмування, яка надає динамічність та взаємодію на стороні клієнта. Вона використовується для реалізації різноманітних функцій та додаткових можливостей веб-додатку. JavaScript дозволяє обробляти події, маніпулювати DOM-елементами та взаємодіяти з сервером за допомогою AJAX-запитів.

TypeScript (Angular 16) використовується як розширення JavaScript, що надає сильну типізацію та додаткові можливості розробки. Angular 16 є потужним фреймворком для створення складних веб-додатків з багатьма компонентами та функціями. Використання TypeScript та Angular 16 сприяє структурованій та масштабованій розробці, забезпечує покращену надійність коду та полегшує підтримку проекту в майбутньому.

Tailwind CSS є модульним CSS-фреймворком, який пропонує готові класи для швидкого та зручного стилізування елементів. Використання

Tailwind CSS дозволяє швидко створювати елегантний та сучасний дизайн, забезпечує легку кастомізацію та покращує продуктивність розробки.

JsonServer використовується як простий сервер для збереження та обміну даними у форматі JSON. Це дозволяє легко встановити мокап-сервер для тестування та розробки без необхідності налаштування повноцінного сервера. JsonServer забезпечує зручну роботу з даними, такими як задачі та користувачі, забезпечуючи маніпуляцію та збереження їх у зручному форматі.

Використання цих технологій дозволяє створити потужний, ефективний та функціональний веб-додаток "ProjectSeek". Комбінація HTML/CSS(SCSS), JavaScript, TypeScript(Angular 16), Tailwind CSS та JsonServer забезпечує гнучкість, зручність у розробці, надійність та зручний інтерфейс користувача.

3.6. Топологія комп’ютерної мережі

Топологія комп'ютерної мережі веб-додатку "ProjectSeek", який знаходиться на хостингу, базується на розподіленій архітектурі. Користувачі, використовуючи свої пристрої, такі як комп'ютери, смартфони або планшети, встановлюють з'єднання з серверами хостингу через Інтернет. Сервери хостингу розташовані у дата-центрі і забезпечують обробку запитів від клієнтів, зберігання та надання доступу до даних, а також виконання обчислювальних процесів, необхідних для роботи веб-додатку.

У мережевій інфраструктурі хостингу використовуються комутатори та маршрутизатори для забезпечення з'єднання між серверами та зовнішнім Інтернетом. Ця інфраструктура може включати резервні з'єднання та механізми балансування навантаження для забезпечення надійності та оптимальної продуктивності додатку. Користувачі отримують доступ до веб-додатку через Інтернет, а взаємодіють з ним за допомогою клієнтських браузерів, що дозволяє їм використовувати функціональні можливості "ProjectSeek" з будь-якого місця і в будь-який час.

РОЗДІЛ 4. ОХОРОНА ПРАЦІ

Охорона праці або безпека праці - це комплекс заходів та правил, спрямованих на запобігання нещасним випадкам, професійним захворюванням і покращення умов праці з метою забезпечення безпеки та здоров'я працівників на робочому місці. Вона включає виконання законодавчих вимог, стандартів та правил, розроблення та впровадження безпечних технологій та процесів, а також підвищення свідомості та навичок працівників у питаннях безпеки. Організації зобов'язані забезпечити безпечне та здорове робоче середовище для своїх працівників.

Основні принципи охорони праці включають виявлення ризиків, запобігання ризикам та управління безпекою. Виявлення ризиків включає оцінку ризиків, пов'язаних з діяльністю організації, виявлення потенційних небезпек і визначення заходів для їх запобігання. Запобігання ризикам передбачає розроблення та впровадження стратегій, які дозволяють запобігти нещасним випадкам та професійним захворюванням. Управління безпекою передбачає встановлення системи управління безпекою, включаючи створення політики безпеки, планування та впровадження заходів безпеки, навчання працівників і оцінку ефективності заходів безпеки.

Охорона праці є важливою складовою успішного функціонування будь-якої організації. Вона не тільки забезпечує безпеку та здоров'я працівників, але й сприяє підвищенню продуктивності, зниженню втрат часу та ресурсів, покращенню репутації організації та створенню сприятливої робочої атмосфери. Ретельна дотримання норм і принципів охорони праці є необхідною умовою для досягнення успіху і стабільності в сфері праці.

Охорона праці є невід'ємною частиною будь-якого робочого середовища і має на меті захистити працівників від можливих ризиків, забезпечити безпеку та здоров'я на робочому місці. Це комплексний підхід, що включає в себе виявлення, оцінку та контроль ризиків, надання необхідної освіти та навчання

працівників, встановлення протоколів безпеки, а також впровадження заходів з попередження нещасних випадків та захворювань.

Головною метою охорони праці є запобігання травмам, професійним захворюванням і негативним наслідкам для здоров'я працівників. Це досягається шляхом виявлення потенційних небезпек у робочому середовищі, таких як небезпечні речовини, механічні чинники, шум, вібрація, неправильні робочі пози, стрес тощо. Після виявлення цих ризиків проводиться їх оцінка та розробка заходів для їх зниження або усунення.

Охорона праці також передбачає надання необхідної освіти та навчання працівникам. Це включає ознайомлення з правилами та процедурами безпеки, навчання використанню захисного спорядження та обладнання, а також надання інформації про потенційні ризики та способи їх уникнення. Професійні консультанти з охорони праці можуть проводити навчальні семінари, тренінги та інструктажі, спрямовані на підвищення свідомості та знань працівників щодо безпеки на робочому місці.

Охорона праці є важливим елементом культури безпеки в організації і має бути постійною практикою. Це включає постійний моніторинг та оновлення системи охорони праці, виявлення потенційних проблем, аналіз нещасних випадків та впровадження заходів для запобігання їх повторенню. Організації повинні співпрацювати з відповідними органами контролю та регуляторами, щоб забезпечити відповідність законодавству та стандартам охорони праці.

Вимоги до охорони праці включають ряд обов'язкових заходів та стандартів, які повинні бути дотримані організацією. Деякі загальні вимоги до охорони праці включають:

1. Визначення політики охорони праці: Організація повинна мати чітку і документовану політику охорони праці, яка встановлює зобов'язання та цілі стосовно безпеки та здоров'я працівників.

2. Оцінка ризиків: Проведення систематичної оцінки ризиків на робочих місцях, виявлення потенційних небезпек і встановлення необхідних заходів для їх усунення або зниження до прийнятного рівня.

3. Навчання та навчання працівників: Забезпечення відповідного навчання та навичок працівникам, що стосуються безпеки та здоров'я на робочому місці. Це включає інструктаж з правил безпеки, навчання використанню обладнання та захисного спорядження, а також свідомості щодо потенційних ризиків і способів їх уникнення.

4. Забезпечення безпечного робочого середовища: Встановлення та підтримка безпечних робочих умов, включаючи належне функціонування обладнання, вентиляцію, освітлення, акустичний комфорт та регулювання робочих процесів.

5. Медичний контроль: Забезпечення регулярного медичного контролю працівників, включаючи періодичні огляди з метою виявлення і попередження професійних захворювань.

6. Управління нещасними випадками та аваріями: Розробка планів дій для надання першої допомоги, реагування на нещасні випадки та аварії, а також розслідування подій з метою уникнення їх повторення.

7. Співпраця з відповідними органами контролю: Забезпечення взаємодії та співпраці з органами державного контролю, регуляторами та іншими сторонами з метою забезпечення відповідності законодавству та стандартам охорони праці.

Вимоги до робочого місця включають ряд факторів, які повинні бути враховані для створення безпечного та комфортного середовища для працівників. Деякі загальні вимоги до робочого місця включають:

1. Освітлення: Робоче місце повинно мати достатнє природне або штучне освітлення, яке забезпечує належну якість видимості та запобігає напруженню очей. Необхідно також уникати блисків та тіней, які можуть вплинути на робочу ефективність та безпеку.

2. Вентиляція: Робоче місце повинно мати належну вентиляцію, щоб забезпечити свіже повітря та видалити шкідливі або неприємні запахи. Це особливо важливо в приміщеннях, де використовуються хімічні речовини або де можливе накопичення шкідливих випарів.

3. Розмір та конфігурація робочої зони: Робоче місце повинно мати достатню площу та організацію для забезпечення зручності та ефективності роботи. Необхідно враховувати розмір робочого столу, наявність місця для розміщення обладнання та документів, а також зручний доступ до необхідних матеріалів.

4. Ергономіка: Робоче місце повинно бути організоване з урахуванням принципів ергономіки, що сприяє забезпеченню правильної пози тіла працівника, підтримці здоров'я та запобіганню травмам, пов'язаним з неправильним положенням тіла або неприродними рухами.

5. Звукове середовище: Робоче місце повинно мати контрольований рівень шуму, що дозволяє працівникам концентруватися на своїй роботі без непотрібних перешкод або відволікань. Деякі робочі області можуть вимагати додаткових заходів для зменшення шуму, наприклад використання звукоізоляційних матеріалів або використання звукозахисних навушників.

6. Безпека: Робоче місце повинно відповідати вимогам безпеки, включаючи наявність пожежних систем, засобів пожежогасіння та додаткових заходів безпеки, які можуть бути необхідні для конкретного виду діяльності компанії.

ВИСНОВОК

В результаті проведеного дослідження потреб компанії "Lytvynov Production" було виявлено значну потребу у персональному веб-додатку, який би допоміг ефективно організувати розробку конкретного розробника. Організація процесу розробки вимагає систематизації завдань, контролю часу та координації зусиль команди. Таким чином, розробка цього веб-додатку стала важливим завданням для поліпшення продуктивності та якості роботи розробників.

Для реалізації даної задачі було обрано кілька ключових технологій: HTML/CSS(SCSS), JS, Angular 16 та JsonServer. Використання HTML/CSS(SCSS) дозволяє створити структуру та вигляд веб-додатку, забезпечуючи гнучкість і зручність для користувача. Мова програмування JavaScript забезпечує динамічну функціональність додатку, дозволяючи здійснювати взаємодію з користувачем та обробку даних. Angular 16, як потужний фреймворк, надає засоби для побудови складних веб-додатків з багатьма компонентами та модулями. Використання JsonServer дозволяє створити мокап-сервер для тестування та розробки без реального серверного API.

Завдяки вибраним технологіям було розроблено ефективний веб-додаток, який забезпечує виконання необхідних задач. Додаток дозволяє розробникам створювати та керувати завданнями, відстежувати прогрес роботи, планувати ресурси та слідкувати за виконанням проектів. Важливою особливістю додатку є можливість організації особистого часу, хоча це не його основна функція. Це дозволяє розробникам бути більш продуктивними та керувати своїм робочим графіком для досягнення кращих результатів.

У майбутньому, розширення функціоналу веб-додатку може бути розглянуто. Наприклад, додавання двомовності для підтримки різних мов використання або розширення функціональних можливостей для покриття більш широкого спектру потреб розробників. Враховуючи постійний розвиток

технологій та зростання потреб користувачів, збільшення функціоналу може підвищити конкурентоспроможність додатку та забезпечити задоволення потреб користувачів у більш широкому контексті.

ВИКОРИСТАНІ ДЖЕРЕЛА:

1. Tailwind (CSS) Docs [Електронний ресурс] – Режим доступу до ресурсу: https://tailwindcss.com/docs/installation Дата звернення (27.05.2023)
2. Sass Documentation [Електронний ресурс] – Режим доступу до ресурсу: https://sass-lang.com/documentation/ Дата звернення (26.05.2023)
3. JavaScript Docs [Електронний ресурс] – Режим доступу до ресурсу: https://developer.mozilla.org/en-US/docs/Web/JavaScript Дата звернення (02.06.2023)
4. Проектування інформаційних систем : лабораторний практикум для студ.освіт. ступ. "Бакалавр" спец. 122 "Комп'ютерні науки " ден. і заоч. форм навч. Частина 2 "Проектування клієнтського додатку" / уклад. : О. М. М'якшило, О.В. Харкянен; Нац. ун-т харч. технол. - Київ : НУХТ, 2017. - 33 с.
5. ДСТУ 3321:2003. Системи конструкторської документації. Терміни та визначення основних понять.
6. Documentation for the npm registry, website, and command-line interface[Електронний ресурс] – Режим доступу до ресурсу: https://docs.npmjs.com/ Дата звернення (29.05.2023)
7. ДСТУ ISO 6309:2007. Пожежна безпека. Загальні вимоги.
8. ДСТУ ISO/IEC 29155-1:2015. Розроблення систем і програмного забезпечення. Платформи для тестування проектів з розроблення інформаційних систем. Частина 1. Концепції та визначення.
9. ДСТУ ISO/IEC 12207:2014. Інженерія систем і програмного забезпечення. Процеси життєвого циклу програмного забезпечення.
10. TypeScript Documentation. o [Електронний ресурс] – Режим доступу до ресурсу: https://www.typescriptlang.org/docs/ Дата звернення (28.05.2023)
11. Angular Material [Електронний ресурс] – Режим доступу до ресурсу: https://material.angular.io Дата звернення (03.06.2023)

12. Охорона праці офісних працівників. Безпечна експлуатація електроустаткування. [Електронний ресурс] – Режим доступу до ресурсу: https://hrliga.com/index.php?module=news&op=view&id=1174 1
13.. Робота в офісі: основні санітарно-гігієнічні вимоги [Електронний ресурс] – Режим доступу до ресурсу: http://oppb.com.ua/news/robota-vofisi-osnovni-sanitarno-gigiyenichni-vymogy. Дата звернення (05.06.2023)
14. Заходи пожежної безпеки у службових приміщеннях (офісі) [Електронний ресурс] – Режим доступу до ресурсу: http://www.cgntb.dp.ua/menu_459.html. Дата звернення (29.05.2023)
15. CSS веб-довідник [Електроний ресурс] – Режим доступу до ресурсу: https://css.in.ua/ Дата звернення (26.05.2023)
16. HTML docs [Електроний ресурс] – Режим доступу до ресурсу: https://developer.mozilla.org/en-US/docs/Web/HTML Дата звернення (03.06.2023)
17. ДСТУ 2941:1994. Системи оброблення інформації. Розроблення систем.Терміни та визначення.
18. ДСТУ 1.0:2003. СТУ 1.0:2003. Національна стандартизація. Основні положення.
19. ДСТУ 3918:1999 (ISO/IEC 12207:2008). Інформаційні технології. Процесижиттєвого циклу програмного забезпечення. – 57 с.
20. ДСТУ 3008:2015 — Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання. – К.: ДП «УкрНДНЦ», 2015. – 32 с.
21. М’якшило О.М. CASE-технології у проектуванні інформаційних систем: електронний навчальний посібник для студ. вищих навч. закладів / О.М. М’якшило, Л.Г. Загоровська,– К.: НУХТ, 2017. – 190 с.

22. ДСТУ ISO/IEC/IEEE 29119-1:2017. Інженерія систем і програмних засобів. Тестування програмних засобів. Частина 1. Поняття та визначення (ISO/IEC/IEEE 29119-1:2013, IDT).
23. Angular docs [Електронний ресурс] – Режим доступу до ресурсу:https://angular.io/docs Дата звернення (31.05.2023)
24. JavaScriptInfo [Електронний ресурс] – Режим доступу до ресурсу: https://javascript.info/ Дата звернення (01.06.2023)
25. freecodecamp [Електронний ресурс] – Режим доступу до ресурсу: https://www.freecodecamp.org/ Дата звернення (01.06.2023)

ДОДАТКИ

Додаток А. Скріншоти програми

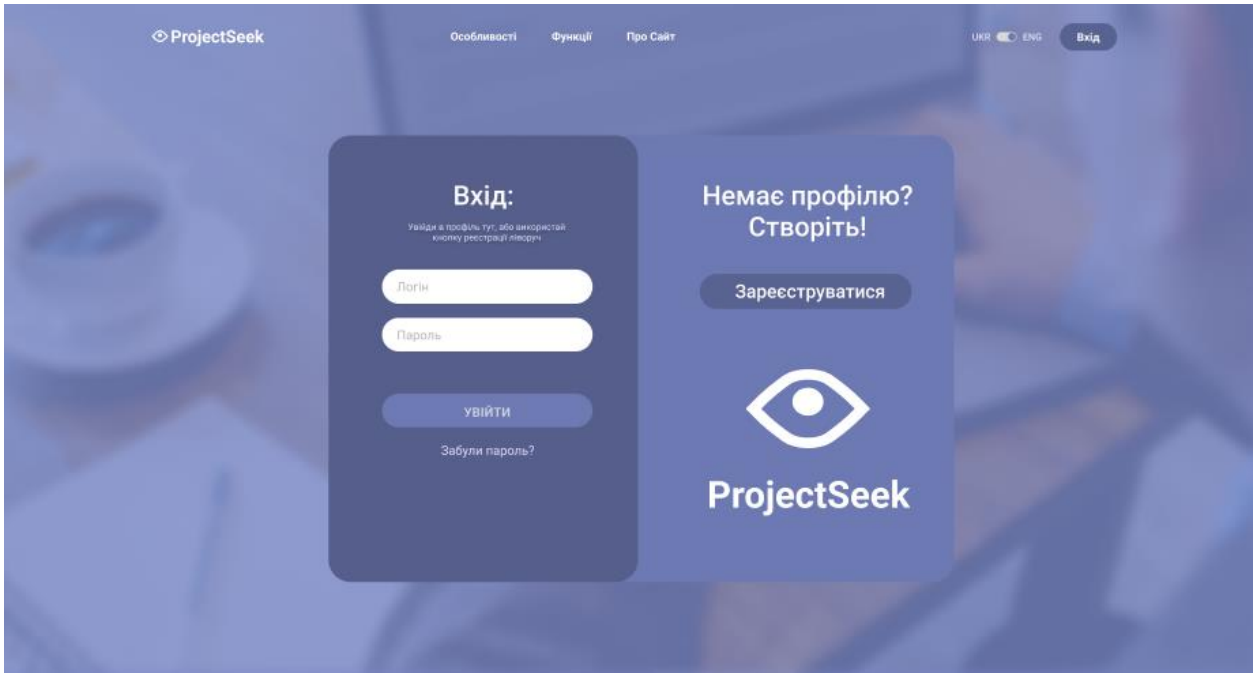

Рисунок А.1 – Вхід в додаток

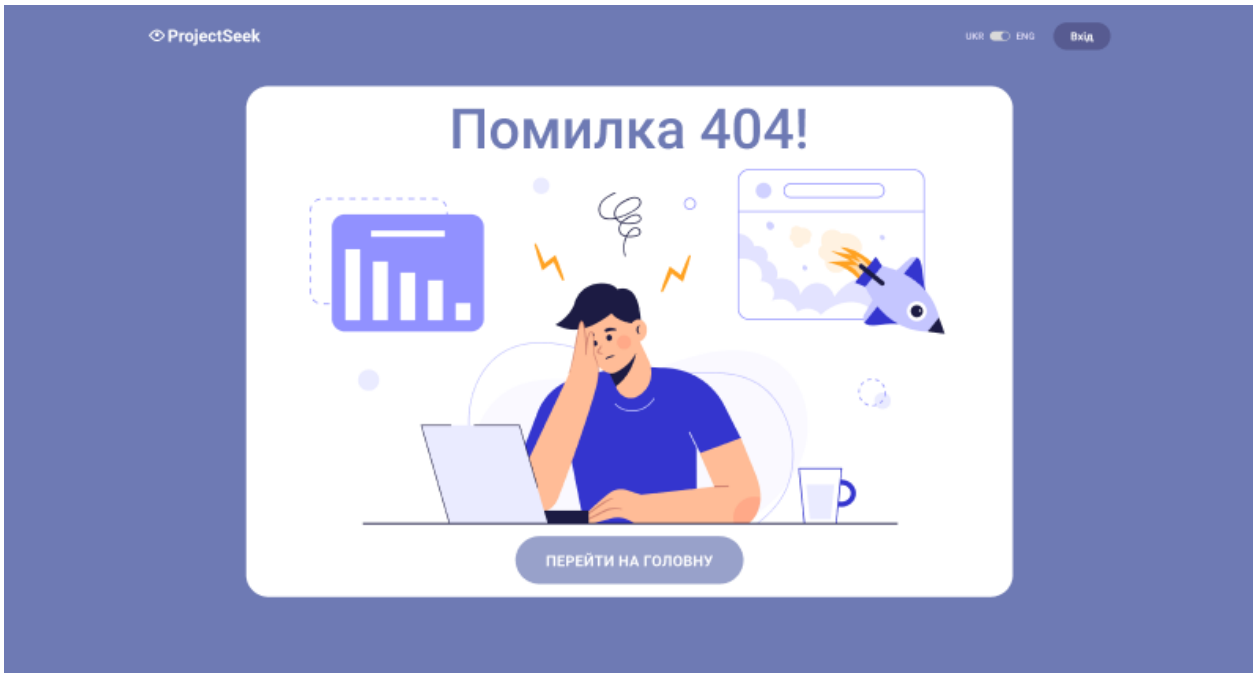

Рисунок А.2 – Помилка завантаження

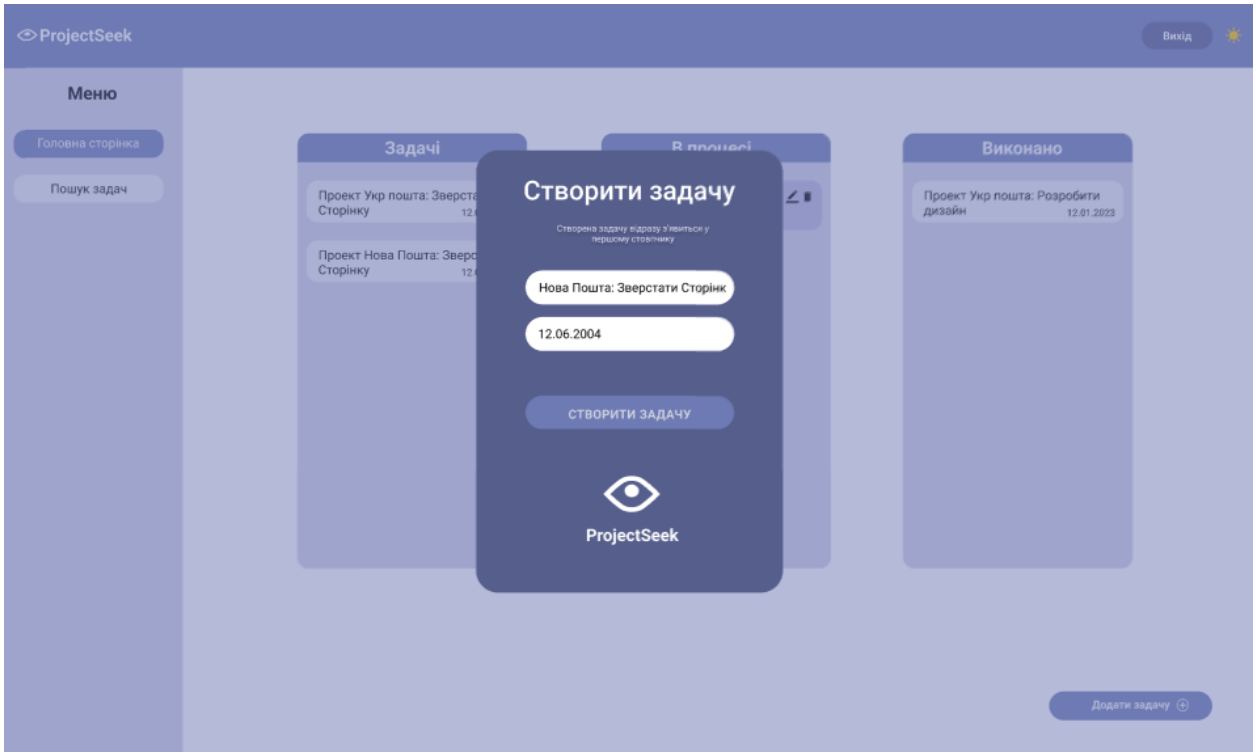

Рисунок А.3 – Заповнення форми

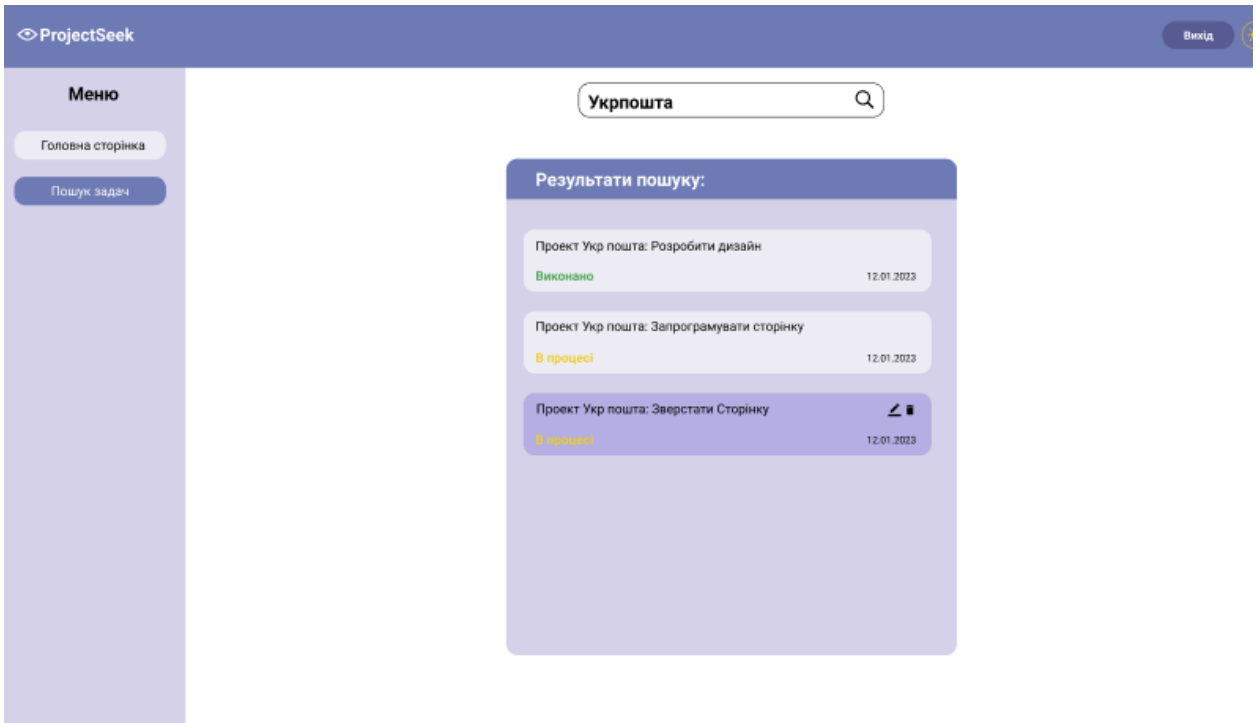

Рисунок А.4 – Взаємодія з задачею

Додаток В. Програмні коди

Головний файл стилів src/styles.scss

```
@tailwind base;
@tailwind components;
@tailwind utilities;
:root {
  --header-padding: 96px;
}
* {
  transition: background-color 0.3s linear, color 0.3s linear;
}
body {
  font-family: "Roboto";
  @apply text-base;
  overflow-x: hidden;
}
.container {
  max-width: 1330px;
  margin: 0 auto;
  padding: 0 15px;
}
.button {
  @apply rounded-3xl py-3 px-8 bg-primary text-white w-fit dark:bg-primary-dark cursor-pointer;
  &:hover {
```

```
    @apply bg-primary-hover dark:bg-primary-dark-hover;
  }
  &.button-intro {
    @apply px-20;
  }
  &.button-login {
    @apply w-full ;
  }
}
h1 {
  @apply text-5xl font-semibold;
  line-height: 59px;
}
h2 {
  @apply text-4xl font-semibold;
}
h3 {
  @apply text-3xl font-semibold;
}
.dialog {
  @apply bg-transparent;
}
.mat-mdc-dialog-container .mdc-dialog__surface {
  background: transparent !important;
  box-shadow: none !important;
```

```
}
.cdk-overlay-dark-backdrop {
  background: rgba(126, 140, 207, 0.58) !important;
}
.cdk-global-scrollblock {
  overflow-y: auto !important;
}
.cdk-drag-animating {
  transition: 250ms cubic-bezier(0, 0, 0.2, 1) !important;
}
.cdk-drop-list-dragging .task-list__list-item:not(.cdk-drag-placeholder) {
  transition: transform 250ms cubic-bezier(0, 0, 0.2, 1);
}
.mat-mdc-form-field {
  width: 100%;
}
.mdc-text-field--filled:not(.mdc-text-field--disabled){
  border-radius: 6px;
}
```

Файл головного компоненту src/app/app.component.ts

```
import {Component} from '@angular/core';
import {NavigationEnd, Router} from "@angular/router";
import {Dialog} from "@angular/cdk/dialog";
import {AuthService} from "./services/auth.service";
```

```
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.scss']
})
export class AppComponent {
  public dark!: any;
  constructor(private router: Router, private dialog: Dialog, private auth:
AuthService) {
  }
  ngOnInit() {
    this.router.events.subscribe(event => {
      if (event instanceof NavigationEnd) {
        this.dialog.closeAll();
        this.auth.checkUser();
      }
    });
  }
  darkChange() {
    this.dark = !this.dark;
  }
}
```

Файл компоненту intro src/app/intro/intro.component.html

```
<main class="intro-page">
```

```
  <div class="intro-page-start-block" style="background-image:
url(/assets/image/intro-bg.png)">
    <div class="container">
      <h1>Ефективний засіб управління проектами</h1>
      <div class="intro-sub-title">Ідеальний інструмент для ефективного
керування вашими проектами зі створення та
        супроводу веб-сайтів.</div>
      <button *ngIf="!logged" class="button button-intro"
routerLink="/registration">Почати роботу</button>
      <button *ngIf="logged" class="button button-intro" routerLink="/main">Ваш
профіль</button>
    </div>
  </div>
</main>
```

src/app/intro/intro.component.ts

```
import {Component, OnDestroy, OnInit} from '@angular/core';
import {AuthService} from "../services/auth.service";
@Component({
  selector: 'app-intro',
  templateUrl: './intro.component.html',
  styleUrls: ['./intro.component.scss']
})
export class IntroComponent implements OnInit, OnDestroy{
  public logged!: boolean;
  constructor(public auth: AuthService) {
  }
```

```
  ngOnInit() {
    this.auth.loggedIn.subscribe(res => {
      this.logged = res;
    });
  }
  ngOnDestroy() {
    this.auth.loggedIn.unsubscribe();
  }
}
```

Файл стилів компоненту intro src/app/intro/intro.component.scss

```
.intro-page {
  @apply bg-bg h-screen w-screen;
  > .intro-page-start-block {
    padding-top: var(--header-padding);
    background-position: center;
    background-repeat: no-repeat;
    background-size: cover;
    @apply  text-white flex justify-center items-center flex-col h-full text-center;
    .container {
      max-width: 550px;
      @apply flex flex-col gap-11 items-center;
    }
  }
}
```