

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»

Директор інституту (декан факультету)

Андрій ФОРСЮК

(підпис)

(прізвище та ініціали)

« 05 » червня 2023р.

«До захисту допущено»

Завідувач кафедри

Сергій ГРИБКОВ

(підпис)

(прізвище та ініціали)

« 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 інформаційної системи для керування проектами ІТ-компанії”

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

Плачков Ігор Сергійович

(прізвище, ім'я та по батькові повністю)

(підпис)

Керівник Самсонов Валерій Васильович

(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'ютерні науки»
(код і назва)

Освітньо-професійна програма «Комп'ютерні науки»

(назва)

ЗАТВЕРДЖУЮ

Завідувач

кафедри Інформаційних технологій,
штучного інтелекту і кібербезпеки

Сергій ГРИБКОВ

« 04 » квітня 2023 року

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА

Плачкова Ігоря Сергійовича

(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення інформаційної системи для керування проектами ІТ-компанії

керівник роботи Самсонов Валерій Васильович, професор, кандидат технічних наук

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

затверджені наказом закладу вищої освіти від «03» квітня 2023р. № 204-кс

2. Строк подання здобувачем роботи: 01.06.2023 р.

3. Вихідні дані до роботи: Дані про айти підрозділ компанії, Поточні процеси управління проектами, Вимоги до інформаційної системи, Аналіз існуючих інформаційних систем, Вимоги до даних.

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити): Розробка інформаційної системи для управління проектами в ІТ-компанії. Аналіз потреб і вимог до системи, включаючи функціональні можливості, типи даних, та архітектуру системи. Вибір компонентів та технологій, розробка інтерфейсу користувача.

5. Перелік графічного матеріалу: Структура компанії ВЕНКОН, скріншоти інтерфесу системи, скріншоти інших програм та систем

6. Консультанти розділів роботи:

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|--------|---|----------------|------------------|
| | | Завдання видав | завдання прийняв |
| 1 | Самсонов В.В., професор | | |
| 2 | Самсонов В.В., професор | | |
| 3 | Самсонов В.В., професор | | |
| 4 | Самсонов В.В., професор | | |

Дата видачі завдання: 4 квітня 2023 року

КАЛЕНДАРНИЙ ПЛАН

| № | Назва етапів виконання кваліфікаційної роботи | Строк виконання етапів роботи | Примітка |
|---|--|-------------------------------|----------|
| 1 | Аналіз літератури про управління проектами в ІТ-компаніях. | 02.04.2023-10.04.2023 | виконано |
| 2 | Вивчення існуючих інформаційних систем для управління проектами. | 10.04.2023-15.04.2023 | виконано |
| 3 | Визначення об'єкта дослідження та формулювання цілей дипломної роботи. | 15.04.2023-25.04.2023 | виконано |
| 4 | Аналіз поточних процесів управління проектами в обраній ІТ-компанії. | 25.04.2023-10.05.2023 | виконано |
| 5 | Визначення вимог до інформаційної системи для управління проектами в ІТ-компанії. | 20.05.2023-25.05.2023 | виконано |
| 6 | Розробка архітектури інформаційної системи. | 25.05.2023-10.06.2023 | виконано |
| 7 | Розробка інтерфейсу користувача. | 10.06.2023-15.06.2023 | виконано |
| 8 | Виконання програмування та розробки необхідних функціональних можливостей системи. | 15.06.2023-29.06.2023 | виконано |
| 9 | Внесення коректив і вдосконалень до системи, | 29.06.2023-05.06.2023 | виконано |

Здобувач

_____ (підпис)

Плачков І.С.

_____ (прізвище та ініціали)

Керівник роботи

_____ (підпис)

Самсонов В.В.

_____ (прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційна робота: 60 с., 13 рисунків, 27 джерел, 3 таблиці.

Ключові слова: CSS, JSON, Angular, JavaScript, Framework, Less, HTML5.

Кваліфікаційна робота присвячена розробці інформаційної системи для керування проектами в ІТ-компанії.

Метою даної роботи є розробка інформаційної системи, яка сприятиме поліпшенню процесів управління проектами в ІТ-компанії. Система буде забезпечувати зручність використання, ефективну координацію робіт, планування ресурсів та керування термінами. Вона допоможе підвищити продуктивність, зменшити ризики затримок та витрат, а також забезпечить високу якість виконання проектів.

Для досягнення цілей роботи будуть використані сучасні методи та підходи до управління проектами, такі як Agile, Scrum та інші. Важливою частиною роботи буде аналіз поточних процесів управління проектами в ІТ-компанії та ідентифікація проблемних аспектів, які потребують вдосконалення. На основі отриманих результатів буде розроблена інформаційна система, яка враховуватиме специфіку ІТ-галузі та потреби компанії.

Основні результати роботи будуть включати розробку архітектури інформаційної системи, реалізацію необхідних функціональних можливостей, інтеграцію з існуючими системами компанії та проведення тестування для перевірки працездатності та ефективності.

ANNOTATION

Qualification work: 60 pages, 13 figures, 27 sources, 3 tables.

Keywords: CSS, JSON, Angular, JavaScript, Framework, Less, HTML5.

The diploma thesis is dedicated to the development of an information system for project management in an IT company. The aim of this work is to develop an information system that will contribute to the improvement of project management processes in the IT company. The system will provide convenience of use, efficient coordination of work, resource planning, and deadline management. It will help increase productivity, reduce risks of delays and costs, and ensure high quality project execution.

To achieve the objectives of this work, modern methods and approaches to project management such as Agile, Scrum, and others will be employed. An important part of the work will involve analyzing the current project management processes in the IT company and identifying problematic aspects that require improvement. Based on the obtained results, an information system will be developed, taking into account the specific requirements of the IT industry and the company's needs.

The main outcomes of this work will include the development of the information system architecture, implementation of necessary functional capabilities, integration with existing company systems, and conducting testing to verify its functionality and efficiency.

ЗМІСТ

| | |
|--|----|
| ВСТУП | 7 |
| РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ ОБ’ЄКТА ДОСЛІДЖЕННЯ ТА ВИЯВЛЕННЯ ЗАДАЧ АВТОМАТИЗАЦІЇ | 8 |
| 1.1. Загальна характеристика компанії ВЕНКОН | 8 |
| 1.2. Організаційна структура ВЕНКОН | 8 |
| 1.3. Взаємодія підрозділів | 10 |
| 1.4. Структура ІТ-відділу | 13 |
| 1.5. Аналіз роботи ІТ-відділу | 16 |
| 1.6. Аналіз проблем автоматизації ІТ-відділу | 17 |
| 1.7. Пошук існуючих рішень | 18 |
| 1.8. Аналіз Jira | 19 |
| 1.9. Аналіз Trello | 21 |
| 1.10. Порівняння систем-аналогів | 22 |
| 1.11. Обґрунтування доцільності проектування й розроблення системи | 24 |
| 1.12. Концептуальна модель системи | 25 |
| РОЗДІЛ 2. ТЕХНІЧНЕ ЗАВДАННЯ НА ПРОЄКТУВАННЯ | 26 |
| 2.1. Загальні положення | 26 |
| 2.2. Призначення і цілі розробки | 26 |
| 2.3. Характеристика об’єкту автоматизації | 27 |
| 2.4. Загальні вимоги до системи | 28 |
| 2.5. Вимоги до структури системи | 29 |
| 2.6. Вимоги до функції | 30 |
| 2.7. Вимоги до збереження даних | 31 |
| 2.8. Вимоги до надійності | 32 |
| 2.9. Вимоги до лінгвістичного забезпечення | 33 |
| 2.10. Вимоги до програмного забезпечення | 34 |
| 2.11. Мінімальні вимоги до ресурсів технічних засобів | 35 |
| РОЗДІЛ 3 ОПИС КОМПЛЕКСУ ЗАДАЧ АВТОМАТИЗАЦІЇ | 37 |
| 3.1. Основні задачі автоматизації | 37 |

| | |
|--|----|
| 3.2. Алгоритмізація та реалізація комплексу задач автоматизації..... | 38 |
| 3.3. Інструкція користувача..... | 39 |
| 3.4. Технічне та системне забезпечення розробки..... | 45 |
| 3.5. Обґрунтування вибору технічних засобів | 46 |
| 3.6. Топологія комп'ютерної мережі..... | 47 |
| 3.7. Обґрунтування вибору ОС та протоколу обміну даними..... | 48 |
| РОЗДІЛ 4. ОХОРОНА ПРАЦІ | 50 |
| ВИСНОВОК..... | 56 |
| ВИКОРИСТАНІ ДЖЕРЕЛА: | 58 |
| ДОДАТКИ..... | 61 |
| Додаток А. Скріншоти програми | 61 |
| Додаток В. Код системи: | 63 |

ВСТУП

Сучасний розвиток галузі інформаційних технологій вимагає від компаній ефективного управління проектами для досягнення успіху та збереження конкурентної переваги. Одним з ключових аспектів успішного управління проектами є використання інформаційних систем, які сприяють оптимізації процесів планування, організації та контролю.

У цьому контексті, об'єктом дослідження даної дипломної роботи є айти підрозділ компанії ВЕНКОН, який займається розробкою та впровадженням інформаційних рішень. Шляхом вивчення процесів, методів та вимог цього підрозділу, було виявлено потребу в розробці спеціалізованої інформаційної системи для управління проектами на мові JavaScript.

На основі детального аналізу поточних процесів та ідентифікації проблемних аспектів управління проектами, була розроблена інформаційна система, яка враховує специфіку айти галузі та вимоги компанії. Ця система була впроваджена в айти підрозділ компанії з метою поліпшення процесів управління проектами, забезпечення зручності використання, ефективної координації робіт, планування ресурсів та керування термінами. Вона сприяла підвищенню продуктивності, зменшенню ризиків затримок та витрат, а також забезпечила високу якість виконання проектів.

РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ ОБ'ЄКТА ДОСЛІДЖЕННЯ ТА ВИЯВЛЕННЯ ЗАДАЧ АВТОМАТИЗАЦІЇ

1.1. Загальна характеристика компанії ВЕНКОН

Компанія ТОВ "ВЕНКОН" зареєстрована 13.03.2006. VENCON – центр інженерних рішень. В цьому інтернет-магазині представлений великий асортимент кліматичної техніки, адже вони спеціалізуються в галузі: систем кондиціонування та опалення, систем водопостачання та вентиляції, очищення води, водонагрівачів, зволожувачів, очисників повітря тощо. У ВЕНКОН також проводять консультацію, а також приймають проекти для реалізації різної складності. Також компанія проводить монтаж усього, що продає. Для цього в штаті є 10 власних монтажних бригад.

Компанія забезпечує стабільну соціальну ситуацію для своїх працівників завдяки високому рівню заробітної плати та сприятливим умовам праці та відпочинку. За 16 років на ринку підприємство збільшило свій штат з 45 до 100 осіб, розширило регіони збуту та налагодило зв'язки з багатьма постачальниками та перевізниками.

1.2. Організаційна структура ВЕНКОН

Компанія має структуру, яка складається з п'ятих підрозділів:

- Відділ електромонтажних робіт;
- Відділ будівельних робіт;
- Відділ торгівлі товарами для спорту та дому;
- Відділ технічного консультування.
- ІТ-відділ;

Головним на підприємстві є в.о. директора, йому підпорядковуються начальники відділів.

Кожен відділ має свою специфіку діяльності та фахівців з відповідними знаннями та навичками. Організаційна структура компанії виглядає наступним чином:

1. Генеральний директор
2. Відділ електромонтажних робіт:
 - Керівник відділу
 - Електромонтажні інженери та техніки
3. Відділ будівельних робіт:
 - Керівник відділу
 - Будівельні інженери та техніки
4. Відділ торгівлі товарами для спорту та дому:
 - Керівник відділу
 - Менеджери з продажу
 - Спеціалісти зі складського обліку та логістики
5. Відділ технічного консультування:
 - Керівник відділу
 - Консультанти
6. IT відділ
 - Розробка сайтів
 - Оновлення інформації на сайтах

Генеральний директор відповідає за загальне керівництво діяльністю компанії та координацію роботи підрозділів. Кожен відділ має свого керівника, який відповідає за роботу свого підрозділу. Електромонтажний та будівельний відділи займаються виконанням робіт зі своєї спеціалізації, а відділ торгівлі займається продажем товарів для спорту та дому та координацією роботи. (Рис. 1.1)

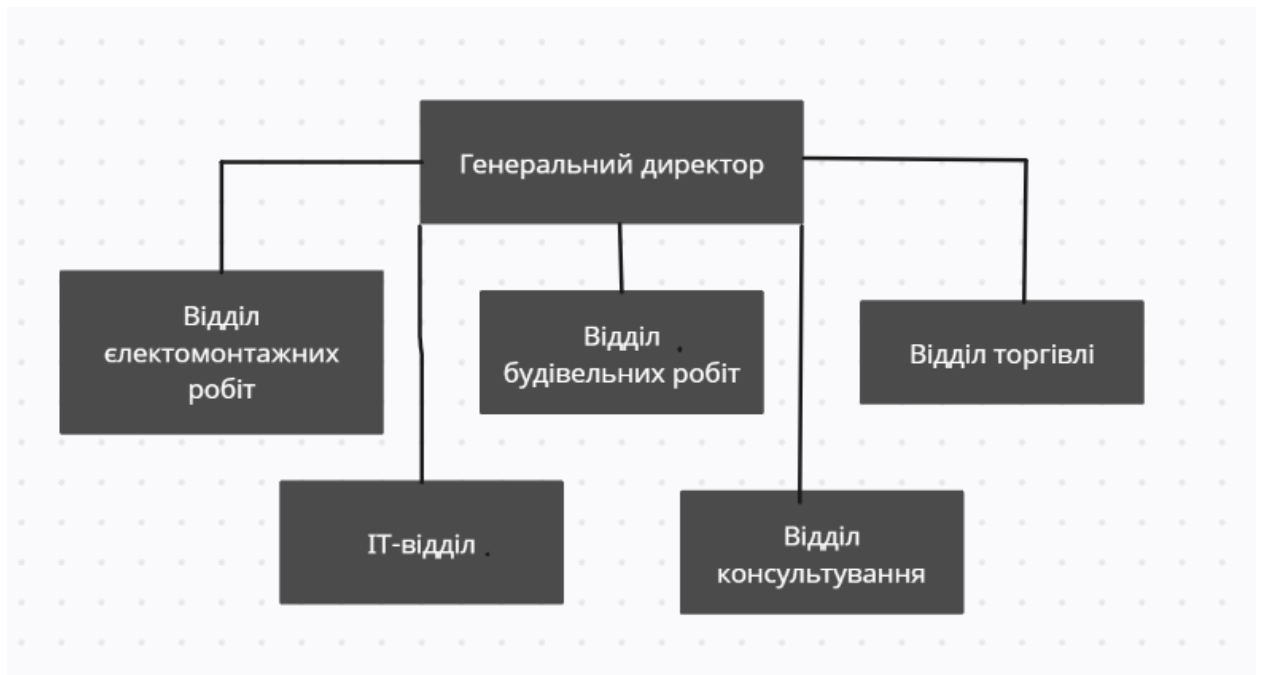


Рис.1.1 - Схема організаційної структури ТОВ «ВЕНКОН»

1.3. Взаємодія підрозділів

На підприємстві часто виникає багато задач, які потрібно вирішити. Це пов'язано з різноманітними факторами, такими як зміна ринкових умов, зміна законодавства, технологічні зміни, конкуренція, проблеми з поставками або виробництвом, проблеми зі збутом тощо.

Для ефективного вирішення задач на підприємстві можуть використовуватися різні методи та інструменти.

Також є системний контроль за виконанням поставлених завдань, вчасно коригуються стратегія та дії з урахуванням змін у ситуації. (Табл. 1.1)

Таблиця 1.1 – Функції та задачі працівників відділу по роботі з клієнтами та обробки замовлень з продажу металопродукату

| № | Задачі | Функції |
|---|--|---|
| 1 | Взаємодія з клієнтом | Прийом вхідних звернень, організація зустрічей, організація консультацій, історія взаємодій, пошук клієнтів. |
| 2 | Забезпечення ритмічної роботи відділу. | Раціональна організація робочої сил. Регулярний оперативний контроль за роботою відділу. |
| 3 | Продаж | Продаж товару, дослідження покупців, оптовиків, дилерів (дзвінки, зустрічі, конференції). Організація зворотного зв'язку для планування продажу. |
| 4 | Підготовка інформації | Організація зворотного зв'язку з посередниками. Підготовка даних для ІТ-відділу. |
| 5 | Ремонт | Ремонт товарів, електромонтаж товарів. |
| 6 | Аналіз роботи | Аналіз історії роботи з клієнтом, , аналіз прибутку. |

Ефективна взаємодія між відділами є дуже важливою для успішної роботи компанії. Це дозволяє забезпечити координацію дій між різними

відділами та зменшити можливість помилок або затримок у виконанні проектів. (Табл. 1.2)

Для забезпечення ефективної взаємодії між відділами в компанії використовують наступні методи:

- Регулярні зустрічі та наради між керівниками відділів. Це дозволяє обговорити поточні питання та спільно приймати рішення.
- Створення загальних процедур та стандартів для всієї компанії. Це дозволяє забезпечити єдиний підхід до вирішення різних питань та зменшити можливість конфліктів між відділами.
- Використання спеціалізованих програм та інструментів для обміну інформацією між відділами. Наприклад, це системи управління проектами, електронні поштові сервіси, спільні сховища даних тощо.
- Організація спільних тренінгів та семінарів для працівників різних відділів. Це дозволяє збільшити рівень спільного розуміння та взаємодії між працівниками різних відділів.

Таблиця 1.2 - Взаємодія відділів один з одним в компанії

| | <i>Підрозділ/ Підприємство</i> | <i>Одержання</i> | <i>Надання</i> |
|--|------------------------------------|---|--|
| | Відділ електромонтажних робіт | 1. Забезпечення необхідних підготовчих робіт перед початком електромонтажу 2. Обговорення технічних аспектів проектів та вибору необхідного обладнання з відділом продажу; | 1. Надання потрібних інструментів та приборів відділу будівельних робіт; |

Продовження таблиці 1.2

| | | | |
|--|---|---|--|
| | Підрозділ ремонту і технічного обслуговування електрообладнання | 1. Обговорення технічних аспектів проектів та вибору необхідного обладнання; | 2. Послуги технічного обслуговування електрообладнання; |
| | Відділ будівельних робіт | 1. З підрозділом закупівель для закупівлі необхідних матеріалів та обладнання для будівництва | 1. З відділом електромонтажних робіт для внесення необхідних змін до проектів та їх відповідність електричним вимогам планів продажів; |
| | Підрозділ контролю якості | 1. Отримання всієї потрібної інформації від інших підрозділів | 1. Забезпечення відповідністю виконаних робіт. |

1.4. Структура ІТ-відділу

Структура ІТ-відділу компанії ВЕНКОН, що спеціалізується на розробці веб-сайтів і складається з 10 співробітників, може мати наступний опис:

- 1) Керівник проекту:

- Відповідає за керівництво та координацію робіт у відділі.
- Забезпечує планування та контроль проектів веб-сайтів.
- Взаємодіє зі замовниками та іншими підрозділами компанії.

2) Веб-дизайнер:

- Відповідає за створення дизайну веб-сайтів.
- Розробляє макети інтерфейсу та графічні елементи.
- Враховує потреби замовника та вимоги ринку.

3) Front-end розробник:

- Відповідає за реалізацію клієнтської частини веб-сайтів.
- Використовує мови програмування, такі як HTML, CSS та JavaScript.
- Забезпечує коректне відображення та функціональність веб-сайтів на різних пристроях.

4) Back-end розробник:

- Відповідає за реалізацію серверної частини веб-сайтів.
- Використовує мови програмування, такі як PHP, Python або Node.js.
- Забезпечує обробку запитів користувачів, взаємодію з базами даних та іншими серверними задачами.

5) QA-інженер (Тестувальник):

- Відповідає за тестування веб-сайтів на виявлення помилок та недоліків.
- Розробляє тестові сценарії та виконує функціональне та регресійне тестування.
- Забезпечує якість розроблених веб-продуктів перед їх випуском.

6) Project Manager:

- Відповідає за планування, керування та контроль проектів веб-розробки.

- Взаємодіє зі замовниками для визначення вимог та встановлення термінів.
- Координує роботу команди та забезпечує досягнення поставлених цілей.

7) База даних розробник:

- Відповідає за проектування та розробку баз даних для веб-сайтів.
- Забезпечує ефективне зберігання, організацію та доступ до даних.
- Розробляє та підтримує SQL-запити та процедури для маніпулювання даними.

8) DevOps-спеціаліст:

- Відповідає за налагодження та підтримку інфраструктури для веб-сайтів.
- Розробляє та налаштовує автоматизовані процеси розгортання та моніторингу.
- Забезпечує безпеку та високу доступність веб-продуктів.

9) UX-дизайнер:

- Відповідає за виконання дослідження користувацького досвіду та розробку веб-інтерфейсів, що забезпечують зручну та ефективну навігацію для користувачів.
- Розробляє прототипи та забезпечує їх тестування з метою покращення взаємодії користувача з веб-сайтами.

10) Системний адміністратор:

- Відповідає за підтримку та налагодження інфраструктури компанії.
- Забезпечує безперебійну роботу серверів, мережевих пристроїв та іншого обладнання.
- Вирішує проблеми зі збереженням даних, забезпечує безпеку мережі та систем компанії.

ІТ-відділ компанії ВЕНКОН є ключовим підрозділом, що спеціалізується на розробці веб-сайтів. Відділ об'єднує команду з 10 висококваліфікованих спеціалістів, які працюють разом для створення інноваційних і привабливих веб-продуктів. Команда включає веб-дизайнерів, фронтенд та бекенд розробників, QA-інженера, проектного менеджера, бази даних розробника, DevOps-спеціаліста, UX-дизайнера та системного адміністратора. Ця комбінація експертів дозволяє забезпечити повний цикл розробки веб-сайтів, починаючи від концепції та дизайну, до реалізації, тестування та підтримки. Висока професійність та співпраця в команді дозволяють досягати високої якості продуктів, відповідати вимогам замовників та забезпечувати успішну реалізацію проектів в області веб-розробки.

1.5. Аналіз роботи ІТ-відділу

Основна робота відділу полягає у створенні зручного та привабливого веб-інтерфейсу для клієнтів. Веб-дизайнери відповідають за розробку естетичного та інтуїтивно зрозумілого дизайну, який відображає корпоративний стиль компанії та забезпечує позитивний користувацький досвід. Фронтенд та бекенд розробники перетворюють цей дизайн у функціональні веб-сторінки та додатки, використовуючи сучасні технології, зокрема JavaScript з Angular Material. QA-інженери забезпечують високу якість продукту шляхом проведення ретельного тестування та виявлення та усунення помилок.

Відділ також займається підтримкою та оптимізацією веб-сайтів, забезпечуючи їх безперебійну роботу та швидкий доступ для користувачів. DevOps-спеціалісти відповідають за розгортання та моніторинг інфраструктури, а системні адміністратори забезпечують безперебійну роботу серверів та мережевих пристроїв.

Результатом роботи відділу є створення веб-сайтів високої якості, які задовольняють потреби клієнтів, відповідають сучасним технологічним стандартам та забезпечують їх успішну присутність в онлайн-середовищі. Команда ІТ-відділу ВЕНКОН є ефективною та динамічною, завдяки чому вона здатна виконувати різноманітні проекти веб-розробки та забезпечити задоволення потреб клієнтів.

1.6. Аналіз проблем автоматизації ІТ-відділу

Аналіз проблем автоматизації ІТ-відділу виявив декілька проблемних аспектів, зокрема пов'язаних з управлінням проектами та нерівномірним розподілом робочого часу між працівниками

1. Координація та планування завдань: В компанії є проблема з плануванням та розподілом завдань між працівниками, з визначенням пріоритетів, термінів виконання та контролю прогресу над кожним проектом.
2. Стеження за прогресом: В компанії не вистачає постійного моніторингу прогресу виконання проектів і завдань. Вона дозволяє менеджеру та іншим учасникам команди в режимі реального часу відстежувати стан робіт, виявляти можливі затримки і приймати вчасні заходи для їх вирішення.
3. Малоефективне призначення ресурсів: Через велику кількість проектів, менеджеру складно оптимізувати використання ресурсів компанії, відсутність структурованої інформації про проекти, не дозволяє ефективно використовувати час для реалізації проектів.

1.7. Пошук існуючих рішень

На ринку існує велика кількість рішень для управління проектами та організації робочих процесів, що схожі на Jira, Trello та інші відомі інструменти. Ці рішення мають широкий спектр функціональності та можуть задовольнити різноманітні потреби IT-відділу та команд розробників.

Наприклад, Jira є одним з найпопулярніших інструментів управління проектами, який дозволяє створювати й відстежувати задачі, розподіляти ресурси, контролювати прогрес та спілкуватися з командою. Він надає широкий спектр функціональних можливостей, таких як графіки Ганта, управління багами, звіти про продуктивність та інтеграція з іншими інструментами розробки.

Trello, зі свого боку, відомий своєю простотою та легкістю використання. Це онлайн-дошка, де завдання можна організувати у вигляді карток, які можна переміщувати між різними стовпцями. Він також підтримує спільну роботу та спілкування в команді, дозволяючи просто відстежувати прогрес проекту.

Крім Jira та Trello, існує також безліч інших інструментів, які пропонують подібну функціональність, включаючи Asana, Basecamp, Monday.com, GitLab та багато інших. Кожен з цих інструментів має свої унікальні особливості та переваги, і вибір залежить від конкретних потреб та вимог компанії та відділу.

Ці існуючі рішення надають зручні та ефективні інструменти для організації процесів, підвищення продуктивності та поліпшення комунікації в IT-відділі. Вони можуть допомогти вирішити проблеми, пов'язані з управлінням проектами та забезпечити більш ефективну роботу команди. З моєї практики, важливо пам'ятати, що робота над проектом зазвичай включає в себе не тільки технічні завдання, а й вимоги замовника, переговори зі стейкхолдерами, планування інтеграції з іншими системами та багато іншого. Тому, обираючи інструмент для управління проектами, важливо ретельно

розглядати всі можливості та переваги кожного з них, щоб знайти той, який найкраще підходить для компанії та проектів.

1.8. Аналіз Jira

Jira є однією з найпопулярніших інструментів управління проектами, спеціально розробленою для команд розробників програмного забезпечення. Вона надає широкі можливості для планування, відстеження і керування проектами. Ось огляд Jira з плюсами і недоліками:

Плюси Jira:

1. Розширена функціональність: Jira має багато потужних інструментів для управління завданнями, відстеження прогресу, планування проектів, керування ресурсами та звітності. Вона надає велику кількість налаштувань і можливостей для адаптації до потреб конкретної команди.
2. Гнучкість і розширюваність: Jira може бути розширена за допомогою різних плагінів і розширень, що дозволяє додати додаткові функції та інтеграції з іншими інструментами, такими як системи контролю версій, засоби спілкування тощо. Це дозволяє пристосувати Jira до конкретних потреб команди.
3. Візуалізація і звітність: Jira надає розширені засоби для візуалізації даних проекту, такі як графіки, діаграми, таблиці, що допомагають зрозуміти прогрес проекту та виявити можливі проблеми. Засоби звітності дозволяють створювати різні звіти, які полегшують аналіз даних та прийняття рішень.

Недоліки Jira:

1. Складність в освоєнні: Jira має досить велику кількість функцій і налаштувань, тому вона може бути складною для освоєння для нових

користувачів. Вона вимагає трохи часу та навчання, щоб повністю зрозуміти всі можливості та використовувати їх ефективно.

2. Великий обсяг інформації: З великою кількістю функцій і даних, Jira може бути перенасиченою інформацією. Для кращого використання інструменту необхідно уважно відслідковувати та організовувати дані, щоб уникнути замішання та путаниці.
3. Вартість: Jira є платним продуктом, і вартість його ліцензій може бути значною, особливо для великих команд або організацій. Це може бути фінансовим обмеженням для деяких команд або стартапів.

Jira є потужним інструментом управління проектами, який дозволяє командам ефективно планувати, відстежувати та керувати своїми проектами.

(Рис. 1.1)

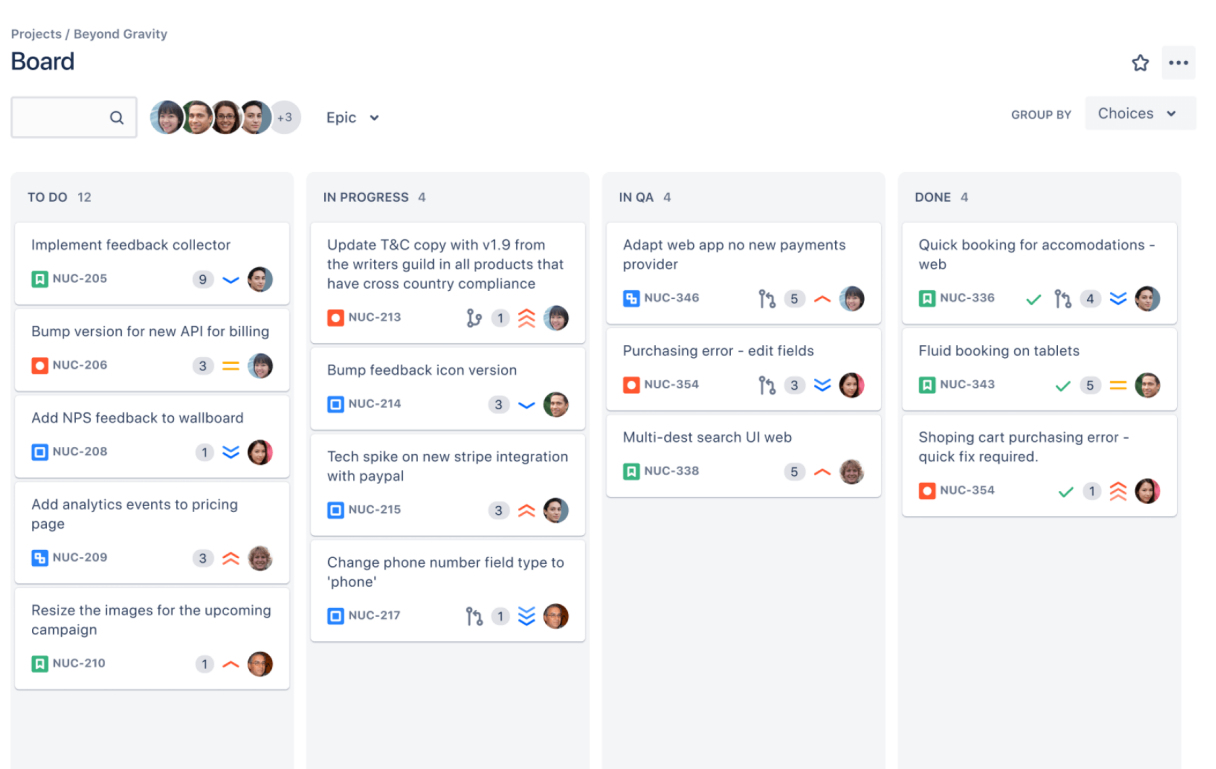


Рисунок 1.1 - Робочий екран Jira

1.9. Аналіз Trello

Trello - це інструмент управління проектами, який пропонує простий і інтуїтивно зрозумілий спосіб організації роботи та співпраці в команді. Заснований на концепції "дошки" і "карток", Trello надає візуальну платформу, де команди можуть створювати, переміщувати та оновлювати завдання у режимі реального часу.

Основні переваги Trello включають простоту використання, добре структуровану інтерфейс, яка дозволяє командам швидко організувати та відстежувати завдання. Кожна "картка" представляє окреме завдання, і її можна переміщати між різними статусами або колонками, що відповідають різним етапам процесу роботи.

Trello також надає можливість додавати коментарі, прикріплювати файли та документацію до карточок, встановлювати терміни виконання та нагадування, а також здійснювати спільну роботу в команді, дозволяючи кожному члену бачити стан проекту та внести свій вклад.

Однак, варто зазначити, що Trello може бути менш підходящим для складних проектів з великою кількістю завдань та потребою в детальному управлінні ресурсами. В порівнянні з деякими іншими інструментами управління проектами, Trello може бути більш обмеженим в розширених функціях та можливостях звітності.

В цілому, Trello є простим, зручним інструментом для організації та відстежування завдань в команді, особливо для проектів з невеликим обсягом та простими вимогами. (Рис. 1.2)

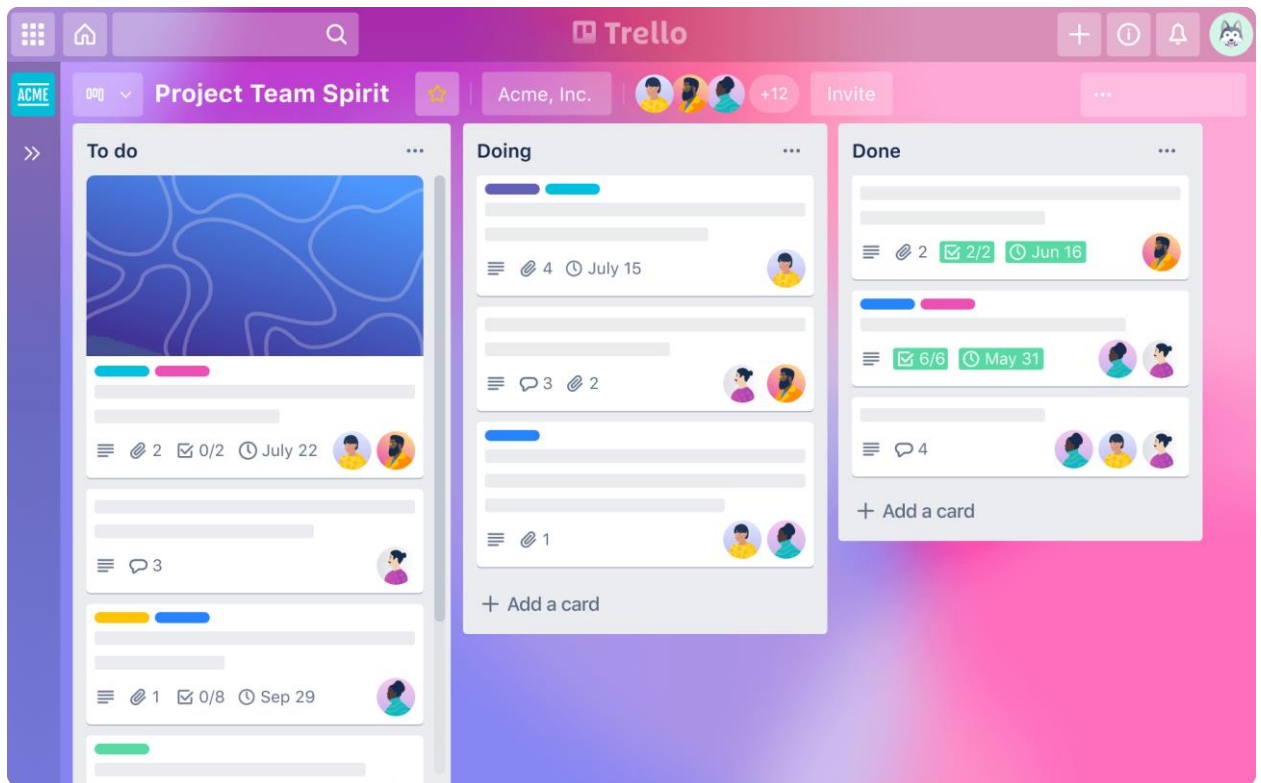


Рисунок 1.2 - Робочий екран Trello

1.10. Порівняння систем-аналогів

Існує кілька інших аналогів Jira та Trello, які також популярні в сфері управління проектами. Давайте порівняємо їх:

1. Asana: Asana - це інструмент для управління проектами, який пропонує розширені функціональні можливості поряд з інтуїтивно зрозумілим інтерфейсом. Він надає широкий спектр інструментів для планування, спільної роботи, відстеження завдань та звітності. У порівнянні з Jira та Trello, Asana зазвичай має більш широкі можливості управління завданнями та комунікації, але може бути складнішим у використанні для новачків.
2. Microsoft Project: Microsoft Project - це програмне забезпечення для управління проектами, яке надає більш розширений інструментарій для планування, відстеження та керування проектами. В порівнянні з Jira та Trello, Microsoft Project є потужним інструментом зі змогами

розкладування завдань, ресурсного планування та звітності, але може бути складним у використанні та вимагати більшої експертизи.

3. Basecamp: Basecamp - це колаборативний інструмент для управління проектами, який зосереджується на спільній роботі та комунікації. Він пропонує простий інтерфейс, де команди можуть спілкуватися, ділитися файлами та відстежувати завдання. В порівнянні з Jira та Trello, Basecamp покладає більший акцент на комунікацію та спільну роботу, але може бути менш гнучким у плануванні та керуванні більш складними проектами.

Хоча Jira, Trello та їх аналоги мають схожі цілі - сприяти управлінню проектами, вони мають свої особливості та переваги. Вибір між ними залежить від конкретних потреб команди або організації. Важливо врахувати функціональність, зручність використання, командну спільноту та інші фактори при прийнятті рішення щодо вибору найбільш підходящого інструменту для проекту.

Продовжуючи порівняння, підсумуємо основні переваги та недоліки Jira, Trello та їх аналогів:

Jira:

- Переваги:
 1. Розширена функціональність управління проектами, включаючи спринти, трекери завдань, планування ресурсів та багато іншого.
 2. Гнучка налаштування та розширення за допомогою плагінів та розширень.
 3. Можливість інтеграції з іншими інструментами та сервісами.
- Недоліки:
 1. Вимагає певного часу та зусиль для оволодіння та налаштування.
 2. Для деяких користувачів може виглядати складним через розширену функціональність.

Trello:

- Переваги:
 1. Простий та інтуїтивно зрозумілий інтерфейс, що полегшує початок роботи.
 2. Зручна організація завдань у вигляді дошок та списків.
 3. Легка спільна робота та спілкування між командами.
- Недоліки:
 1. Обмежений функціонал порівняно з більш розширеними інструментами управління проектами.
 2. Може бути складно використовувати для більших та складніших проектів.

Враховуючи це, Jira часто використовується для складних проектів, де потрібна гнучкість та розширені можливості управління, тоді як Trello зазвичай популярний для менших команд або проектів, де простота та швидкість важливі.

1.11. Обґрунтування доцільності проектування й розроблення системи

Головні мінуси безкоштовних сайтів, таких як Trello та Jira, включають наступні аспекти:

1. Обмежена функціональність: Безкоштовні версії Trello та Jira мають обмежений набір функцій порівняно з платними аналогами. Вони можуть не мати всіх необхідних інструментів та можливостей для повноцінного управління складними проектами. Це може призвести до обмеженого контролю над завданнями та обмежених можливостей аналізу та звітності.
2. Обмежена масштабованість: Безкоштовні версії сайтів, таких як Trello та Jira, зазвичай мають обмежену масштабованість, що означає, що вони можуть бути неефективними для великих команд або проектів з великим

обсягом завдань та ресурсів. Вони можуть не забезпечувати достатньої продуктивності та швидкості роботи при великому обсязі даних.

3. Відсутність індивідуального підходу: Безкоштовні сайти, такі як Trello та Jira, надають загальні рішення для управління проектами, які можуть не враховувати специфічні потреби вашого відділу або команди. Вони можуть бути неефективними для налаштування та пристосування до унікальних процесів та вимог вашого проекту.
4. Відсутність підтримки: Безкоштовні версії Trello та Jira не надають підтримку на такому рівні, як платні аналоги. У разі виникнення проблем або потреби в допомозі користувачам може бути важко отримати оперативну та професійну підтримку.

Ураховуючи ці мінуси, розробка власної інформаційної системи управління проектами, спеціально побудованої для менеджера відділу, може мати численні переваги. Можна буде настроїти функціонал, враховуючи конкретні потреби ІТ-відділу, забезпечити високу продуктивність, підвищити мотивацію співробітників та отримати підтримку, яка відповідає вашим потребам.

1.12. Концептуальна модель системи

Концептуальна модель системи сайту для управління проектами включає в себе основні функціональні компоненти, які сприяють зручному та ефективному управлінню проектами. Реєстрація користувачів дозволяє створити особисті облікові записи і отримати доступ до системи. Створення проектів дозволяє менеджеру визначити назву, опис і склад команди проекту, що сприяє чіткому визначенню завдань та відповідальності. Розподіл завдань між працівниками забезпечує більшу організацію та координацію роботи, дозволяючи визначити пріоритети, терміни виконання і відслідковувати прогрес роботи.

РОЗДІЛ 2. ТЕХНІЧНЕ ЗАВДАННЯ НА ПРОЄКТУВАННЯ

2.1. Загальні положення

Інформаційна система для управління проектами ІТ-компанії у вигляді сайту з реєстрацією, створенням профілів працівників, відстежуванням і управлінням проектами та розподілом роботи між працівниками.

2.2. Призначення і цілі розробки

Призначення і цілі розробки сайту для управління проектами включають:

1. **Забезпечення ефективного управління проектами:** Основним призначенням сайту є створення інформаційної системи, яка допомагає в управлінні проектами. Сайт надає зручний інтерфейс для створення, відстеження і оцінки проєктів, розподілу завдань між працівниками, відстеження прогресу та визначення пріоритетів.
2. **Покращення спілкування та співпраці:** Сайт для управління проектами створює централізовану платформу, де всі учасники проєкту можуть спілкуватися, обмінюватися інформацією, ділитися файлами та коментувати завдання. Це сприяє покращенню комунікації та співпраці всередині команди та замовниками.
3. **Збереження та організація даних про проєкти:** Сайт дозволяє зберігати всю необхідну інформацію про проєкти, включаючи деталі про завдання, терміни виконання, призначених працівників, документацію та інші ресурси. Це допомагає відстежувати прогрес роботи, забезпечує доступ до актуальних даних та сприяє легкому пошуку необхідної інформації.
4. **Підвищення ефективності та продуктивності:** Основною метою розробки сайту для управління проектами є покращення ефективності роботи команди, зменшення затрат часу та ресурсів на адміністративні процеси, автоматизація завдань та забезпечення структурованого підходу до

управління проектами. Це сприяє підвищенню продуктивності всієї команди і досягненню кращих результат

5. Моніторинг і звітність: Сайт для управління проектами забезпечує можливість моніторити прогрес проектів, виконання завдань та дотримання термінів. Він надає звіти і аналітичні дані, які допомагають менеджерам приймати обґрунтовані рішення та реагувати на зміни вчасно. Моніторинг дозволяє виявляти можливі проблеми та затримки, що дозволяє уникнути негативного впливу на проекти.
6. Налаштування процесів та стандартизація: Сайт дозволяє встановити та реалізувати стандарти та процеси управління проектами всередині організації. Він сприяє впровадженню уніфікованих методологій, шаблонів та процедур, що полегшує управління проектами, забезпечує однаковий підхід до роботи та сприяє покращенню якості та результативності проектів.
7. Резервування та безпека даних: Сайт для управління проектами забезпечує збереження та резервне копіювання даних про проекти, завдання та інші важливі елементи системи. Він також забезпечує захист даних та контроль доступу, щоб забезпечити конфіденційність і цілісність інформації.

2.3. Характеристика об'єкту автоматизації

Об'єктом автоматизації є діяльність ІТ-відділу компанії ВЕНКОН що є ключовим підрозділом, та який займається розробкою веб-сайтів. Команда включає 10 висококваліфікованих спеціалістів з різних областей, таких як веб-дизайн, фронтенд та бекенд розробка, QA, проектне управління, бази даних, DevOps, UX-дизайн та системне адміністрування. Відділ забезпечує повний цикл розробки веб-сайтів, включаючи концепцію, дизайн, реалізацію, тестування та підтримку.

2.4. Загальні вимоги до системи

Загальні вимоги до системи включають:

1. Реєстрація користувачів: Система повинна мати можливість реєструвати користувачів, щоб вони могли отримати доступ до функціональності системи.
2. Створення проектів: Користувачі повинні мати можливість створювати нові проекти в системі. Це дозволить організувати роботу над конкретними завданнями та проектами.
3. Розподіл завдань: Система повинна надавати можливість розподіляти завдання між працівниками. Менеджер повинен мати зручний інтерфейс, щоб легко призначати завдання, встановлювати терміни виконання та відстежувати прогрес роботи.
4. Керування термінами: Система повинна давати змогу встановлювати та відстежувати терміни виконання завдань. Це допоможе уникнути затримок та забезпечити своєчасне завершення проектів.
5. Комунікація та співпраця: Система повинна містити функціонал для комунікації та співпраці між членами команди. Це може включати можливість обміну повідомленнями, коментування завдань та спільне вирішення проблем.
6. Візуалізація прогресу: Система повинна надавати інструменти для візуалізації прогресу проектів, наприклад, графіки, діаграми чи звіти. Це дозволить менеджеру та учасникам проекту легко оцінити стан та успішність роботи.
7. Зберігання даних: Система повинна забезпечувати надійне зберігання даних про проекти, завдання, користувачів та їх взаємодію. Резервне копіювання та захист інформації повинні також бути враховані.
8. Масштабованість: Система повинна бути готовою до масштабування, здатною працювати з ростом числа проектів, користувачів та обсягу даних.

9. **Безпека:** Система повинна забезпечувати високий рівень безпеки, включаючи автентифікацію користувачів, захист даних та контроль доступу до функціоналу системи.
10. **Зручний інтерфейс:** Система повинна мати зрозумілий, інтуїтивно зрозумілий та зручний для використання інтерфейс, щоб користувачі могли легко орієнтуватися та взаємодіяти з системою.

2.5. Вимоги до структури системи

Вимоги до структури включають:

1. **Модульність:** Система повинна мати модульну структуру, яка дозволяє легко розширювати та модифікувати функціонал. Це дозволить адаптувати систему під специфічні потреби вашого ІТ-відділу.
2. **Ієрархічна організація:** Структура системи повинна включати ієрархічну організацію проектів та завдань. Це дозволить керівникам різних рівнів отримувати доступ до відповідної інформації та здійснювати керування проектами на різних рівнях.
3. **Ролева структура:** Система повинна мати можливість визначати ролі та права доступу для користувачів. Це дозволить обмежити доступ до конфіденційної інформації та забезпечити відповідну організацію робочих процесів.
4. **Зв'язки та залежності:** Структура системи повинна відображати зв'язки та залежності між проектами, завданнями та працівниками. Це допоможе визначити логічні зв'язки між різними елементами системи та впорядкувати робочий процес.
5. **Форматування та візуалізація:** Структура системи повинна дозволяти формувати та візуалізувати інформацію, щоб забезпечити зрозумілість та зручність використання. Це може включати відображення графіків, діаграм та інші елементи, що полегшують сприйняття даних.

6. Легкість навігації: Структура системи повинна бути логічною та зручною для навігації між різними розділами, проектами та завданнями

2.6. Вимоги до функції

Вимоги до функцій включають:

1. Створення проектів: Система повинна мати можливість створювати нові проекти та визначати їх основні параметри, такі як назва, опис, дедлайни та бюджет.
2. Розподіл завдань: Система повинна дозволяти розподіляти завдання між працівниками відповідно до їх ролей та компетенцій. Керівники проектів повинні мати можливість призначати завдання, встановлювати пріоритети та визначати дедлайни.
3. Моніторинг прогресу: Система повинна надавати можливість відстежувати прогрес виконання завдань та проектів. Це може включати відображення відсотка виконання, часу, витрат ресурсів та інших метрик, що допомагають оцінити ефективність та вчасність виконання завдань.
4. Керування документами: Система повинна забезпечувати зручну організацію та збереження документів, пов'язаних з проектами. Це може включати можливість завантаження, збереження та спільного доступу до документів, таких як специфікації, плани проекту, звіти тощо.
5. Звітність та аналітика: Система повинна забезпечувати збір та аналіз даних проектів для подальшої звітності та аналізу. Це може включати генерацію звітів про стан проектів, витрати ресурсів, прогрес виконання, ефективність роботи тощо.
6. Інтеграція з іншими системами: Система повинна мати можливість інтегруватися з іншими інструментами та системами, що використовуються в компанії. Наприклад, інтеграція з системою управління задачами, електронною поштою, календарем тощо.

7. **Безпека та доступ:** Система повинна забезпечувати належний рівень безпеки, контроль доступу та захисту конфіденційної інформації. Це може включати автентифікацію користувачів, ролеву базовану систему доступу та шифрування даних.
8. **Мобільний доступ:** Система повинна мати можливість доступу через мобільні пристрої, що дозволить користувачам отримувати доступ до інформації та здійснювати керування проектами навіть поза офісом.

2.7. Вимоги до збереження даних

Вимоги до збереження включають:

1. **Цілісність даних:** Система повинна забезпечувати цілісність даних, що означає, що дані повинні бути коректними та не пошкодженими. Важливо, щоб під час збереження та зчитування даних відбувалися перевірки на відповідність схемі JSON та унікальності ідентифікаторів.
2. **Безпека даних:** Забезпечення безпеки даних є важливою вимогою. Для цього можуть використовуватись різні підходи, такі як шифрування даних, контроль доступу та автентифікація користувачів. Важливо, щоб лише авторизовані користувачі мали доступ до даних та їх редагування.
3. **Ефективне збереження та отримання даних:** Система повинна забезпечувати ефективне збереження та отримання даних з використанням JSON-структури. Це може включати оптимізацію запитів до бази даних, використання індексів та кешування даних, щоб забезпечити швидкий доступ до інформації.
4. **Резервне копіювання та відновлення даних:** Важливо мати механізм резервного копіювання даних та можливість відновлення у разі випадку втрати або пошкодження даних. Це дозволить зберегти цінну інформацію та уникнути втрати даних в разі непередбачуваних ситуацій.

5. Масштабованість: Планування майбутнього росту та масштабування системи є важливою вимогою. Система повинна бути готовою до збільшення обсягу даних та кількості користувачів без втрати продуктивності. Можуть застосовуватися технології горизонтального та вертикального масштабування для оптимального використання ресурсів.

2.8. Вимоги до надійності

Вимоги до надійності включають:

1. Робота без збоїв: Система повинна функціонувати стабільно та безперебійно, забезпечуючи постійний доступ до функціоналу та даних навіть при високих навантаженнях або випадках непередбачуваних ситуацій.
2. Захист від втрати даних: Важливо мати механізми резервного копіювання, архівації та відновлення даних, щоб уникнути втрати цінної інформації. Система повинна забезпечувати надійне збереження даних та відновлення у разі випадку витоку або пошкодження даних.
3. Захист від несанкціонованого доступу: Система повинна мати механізми автентифікації, авторизації та контролю доступу для запобігання несанкціонованому доступу до даних та функціоналу. Це включає захист від хакерських атак, злому паролів та недостатньої захищеності даних.
4. Відновлення після збоїв: Система повинна мати можливість швидкого відновлення після збоїв або випадків аварійного завершення роботи. Це може включати автоматичну перезавантаження системи, відновлення стану даних та відновлення активності проектів.
5. Моніторинг та логування: Система повинна забезпечувати моніторинг роботи, щоб виявляти можливі проблеми або збої, а також логування подій для аналізу та відстеження подій. Це допоможе вчасно виявляти проблеми та вживати заходів для їх вирішення.

6. Відповідність стандартам: Система повинна відповідати вимогам стандартів безпеки, які можуть включати GDPR, HIPAA, ISO 27001 та інші відповідні нормативні вимоги.

2.9. Вимоги до лінгвістичного забезпечення

Вимоги до лінгвістичного забезпечення включають:

1. Мультиязиковість: Система повинна підтримувати різні мови для користувачів, щоб забезпечити зручну роботу та комунікацію в міжнародних командних проектах. Вона повинна мати можливість перекладу інтерфейсу, повідомлень та інших текстових елементів на різні мови.
2. Локалізація: Система повинна бути здатна адаптуватися до різних культурних контекстів та відповідати локальним вимогам. Це означає, що формати дат, часу, валют, одиниць вимірювання та інші локалізовані елементи повинні бути налаштовані відповідно до країни або регіону користувача.
3. Підтримка специфічних символів та пунктуації: Система повинна коректно обробляти та відображати спеціальні символи, пунктуацію та особливості різних мов, щоб забезпечити точність та зрозумілість текстової інформації.
4. Підтримка міжнародних стандартів: Система повинна враховувати міжнародні стандарти транслітерації, транскрипції та інших методів перетворення тексту з однієї мови на іншу. Це допоможе забезпечити правильність відображення та розпізнавання текстової інформації, незалежно від мови.
5. Підтримка специфічних мовних особливостей: Система повинна бути здатна до правильного врахування специфічних мовних особливостей, таких як граматики, складність морфології, синтаксис тощо.

2.10. Вимоги до програмного забезпечення

Вимоги до програмного забезпечення включають:

1. **Функціональність:** Система повинна мати широкий спектр функцій, що включає створення проектів, розподіл завдань, відстеження прогресу, призначення термінів, комунікацію між учасниками проекту та інші важливі функції для ефективного управління проектами.
2. **Надійність:** Система повинна бути стабільною та надійною, забезпечуючи безперебійну роботу навіть при великому навантаженні та інтенсивному використанні. Вона повинна мати механізми для виявлення та відновлення від помилок, а також резервне копіювання та відновлення даних.
3. **Ефективність:** Система повинна працювати швидко та ефективно, забезпечуючи швидку відповідь на дії користувачів. Вона повинна оптимізувати використання ресурсів, таких як процесор, пам'ять та мережевий трафік, для забезпечення оптимальної продуктивності.
4. **Легкість використання:** Система повинна мати інтуїтивний та зручний інтерфейс, що дозволяє користувачам легко орієнтуватися та використовувати її функціонал. Вона повинна бути легко навчаною та доступною для користувачів з різним рівнем технічної підготовки.
5. **Сумісність:** Система повинна бути сумісною з різними операційними системами, веб-браузерами та мобільними пристроями. Вона повинна підтримувати стандарти та протоколи, що дозволяють інтеграцію з іншими системами та сервісами.
6. **Безпека:** Система повинна забезпечувати захист конфіденційної та важливої інформації. Вона повинна мати механізми аутентифікації, авторизації та шифрування даних, а також контроль доступу до функцій та ресурсів системи.
7. **Розширюваність:** Система повинна мати гнучку архітектуру, що дозволяє легко розширювати її функціонал та додавати нові модулі та інтеграції.

Вона повинна підтримувати розширення функцій, налаштування та модифікацію з мінімальними зусиллями.

2.11. Мінімальні вимоги до ресурсів технічних засобів

Операційна система – Windows, Linux, Ios. (Табл. 2.1)

Вимоги до ресурсів технічних засобів включають набір характеристик, необхідних для ефективної роботи та задоволення потреб користувачів. Ці вимоги варіюються в залежності від типу технічного засобу та його призначення.

У разі мережевих пристроїв, таких як маршрутизатори або комутатори, важливими є мережеві можливості, такі як швидкість передачі даних та підтримка протоколів зв'язку. Крім того, ресурси технічних засобів повинні бути сумісними зі зв'язаним програмним забезпеченням, щоб забезпечити його оптимальну роботу.

Вимоги до ресурсів включають:

1. Процесор: Мінімум двоядерний процесор з тактовою частотою 2.0 ГГц або еквівалент. Рекомендовано використовувати більш потужні процесори для оптимальної продуктивності системи.
2. Пам'ять: Мінімум 4 ГБ оперативної пам'яті. Рекомендовано мати більшу кількість пам'яті для обробки великих обсягів даних та більшої швидкодії роботи системи.
3. Дисковий простір: Мінімум 10 ГБ вільного місця на жорсткому диску для встановлення системи та зберігання даних. Рекомендовано мати більший обсяг дискового простору для забезпечення зручності та можливості розширення системи.

Монітор: Мінімум роздільна здатність екрану 1280x800 пікселів. Рекомендовано використовувати більш високу роздільну здатність для зручного відображення інтерфейсу та деталей проектів.

1. Інтернет-з'єднання: Доступ до стабільного та швидкого Інтернет-з'єднання для забезпечення безперебійної роботи системи та обміну даними між користувачами.

Таблиця 2.1 - Вимоги до технічного забезпечення системи

| Характеристика | Вимога |
|-----------------------|---|
| Операційна система | Windows, macOS, Linux |
| Процесор | Мінімум 1,0 ГГц |
| Оперативна пам'ять | Мінімум 2 ГБ |
| Веб-браузер | Google Chrome, Mozilla Firefox, Safari, Microsoft Edge та інші останні версії |
| Інтернет-з'єднання | Стабільне інтернет-з'єднання |

РОЗДІЛ 3 ОПИС КОМПЛЕКСУ ЗАДАЧ АВТОМАТИЗАЦІЇ

3.1. Основні задачі автоматизації

Комплекс задач автоматизації для управління проектами в ІТ-компанії має на меті полегшити та оптимізувати процеси планування, виконання та моніторингу проектів. Він надає зручні та ефективні інструменти для менеджера проекту, які сприяють кращому управлінню завданнями та працівниками, а також покращують комунікацію та звітність.

Основні функції та можливості комплексу задач автоматизації включають:

1. Створення та керування проектами: Система дозволяє менеджеру створювати нові проекти, встановлювати їх параметри (назва, опис, терміни), призначати відповідальних працівників та розподіляти завдання.
2. Розподіл завдань та контроль виконання: Менеджер може створювати завдання в межах проекту, призначати їх відповідальним працівникам та встановлювати терміни виконання. Система надає засоби для моніторингу прогресу виконання завдань.
3. Моніторинг та звітність: Система збирає дані про прогрес виконання проектів, часові розклади, витрати ресурсів, виконання завдань тощо.

Інформаційне забезпечення є важливою складовою комплексу задач автоматизації для управління проектами в ІТ-компанії. Його основна мета - забезпечити надійний та зручний доступ до інформації, необхідної для ефективного управління проектами та прийняття обґрунтованих рішень.

Інформаційне забезпечення включає наступні аспекти:

Зберігання та організація даних: Інформаційна система повинна забезпечувати безпечне зберігання всіх даних, пов'язаних з проектами, включаючи документацію, історію змін та інше.

Керування доступом до інформації: Забезпечення безпеки та обмеження доступу до конфіденційної інформації є важливим аспектом інформаційного забезпечення. Інформаційна система повинна мати механізми аутентифікації та авторизації, що дозволяють керувати правами доступу різних користувачів до різних частин системи та типів інформації.

3.2. Алгоритмізація та реалізація комплексу задач автоматизації

Алгоритмізація та реалізація комплексу задач автоматизації відіграють важливу роль у розробці веб-сайту для управління проектами в ІТ-компанії. Основні етапи алгоритмізації та реалізації можуть включати наступні кроки:

1. **Аналіз та збір вимог:** Починаючи з аналізу потреб та вимог користувачів, необхідно визначити функціональні та нефункціональні вимоги до системи управління проектами. Цей етап включає вивчення бізнес-процесів, виявлення ключових функціональностей та створення прототипу системи.
2. **Проектування архітектури:** На основі зібраних вимог, необхідно розробити архітектурну концепцію системи. Це включає визначення компонентів, модулів, бази даних, інтерфейсів та зв'язків між ними. Важливо забезпечити масштабованість, надійність та ефективність системи.
3. **Реалізація:** На цьому етапі розробляються програмний код, бази даних та інші необхідні компоненти системи. Використовуються мови програмування, такі як HTML, CSS, JavaScript, а також фреймворки та інструменти, які допомагають у швидкій та ефективній розробці.

4. Тестування та налагодження: Після реалізації системи проводиться тестування, яке включає функціональне тестування, тестування безпеки та навантаження. Під час тестування виявляються та виправляються помилки та недоліки. Після успішного тестування проводиться налагодження системи.
5. Підтримка: Після успішного завершення тестування система впроваджується у виробниче середовище. Після впровадження забезпечується підтримка та обслуговування системи, включаючи виправлення помилок, оновлення та розширення функціональності.
6. Алгоритмізація та реалізація комплексу задач автоматизації допомагають покращити ефективність управління проектами в ІТ-компанії, забезпечуючи автоматизацію процесів, зниження ризиків та поліпшення якості роботи.

3.3. Інструкція користувача

Інформаційна система є веб-сайтом під назвою ProjectPR.

- 1) При вході на сайт користувач попадає на сторінку із загальною інформацією про сайт. (Рис. 3.1.)

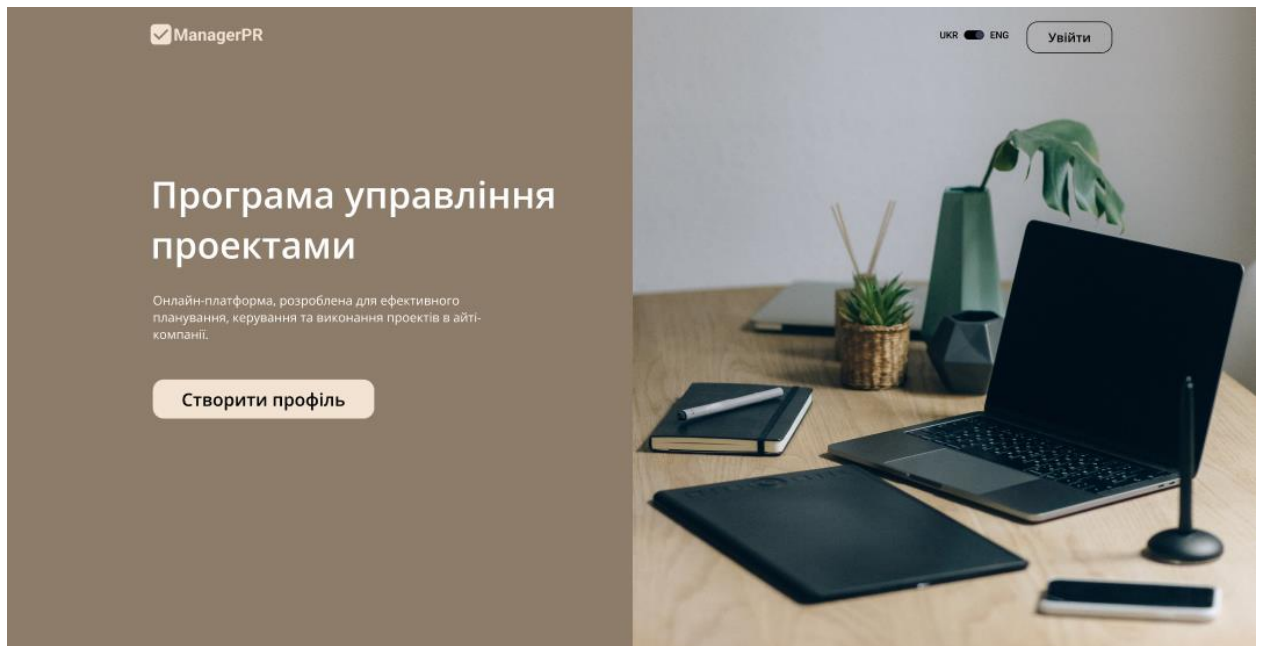


Рисунок 3.1 - Лендінг сайту

На сторінці можна ознайомитися з назвою та інформацією про веб-сайт та перейти на сторінку реєстрації або входу. (Рис.3.2, 3.3)

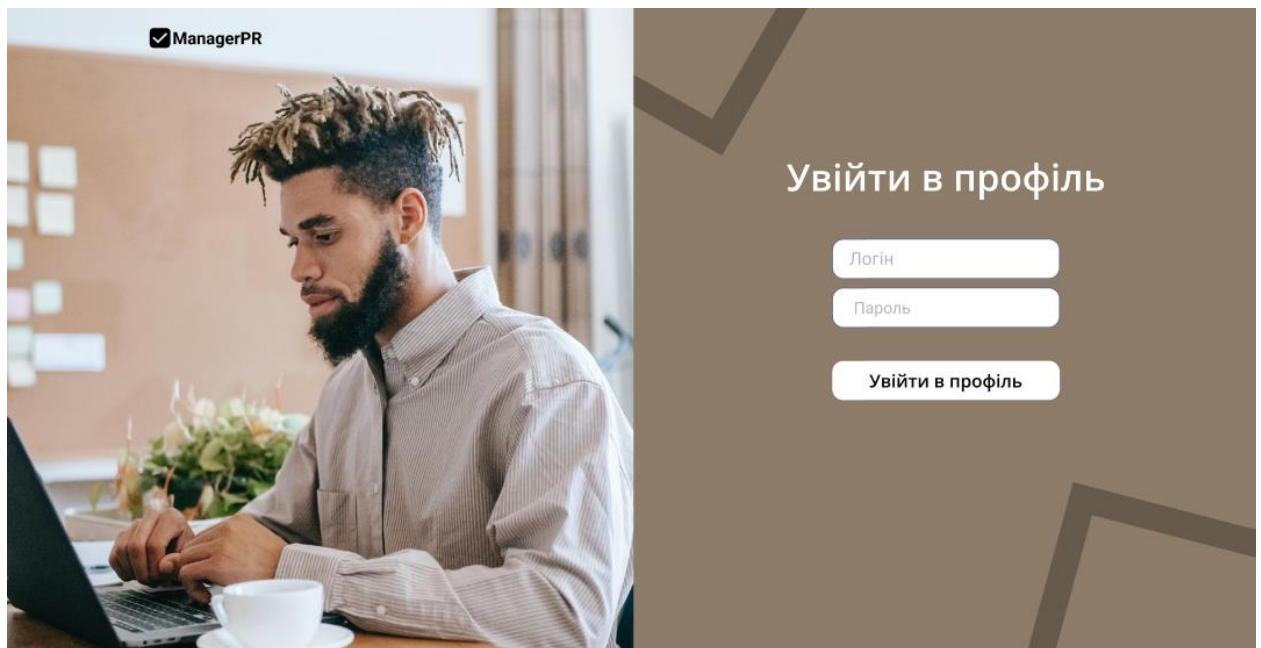


Рисунок 3.2 – Сторінка входу

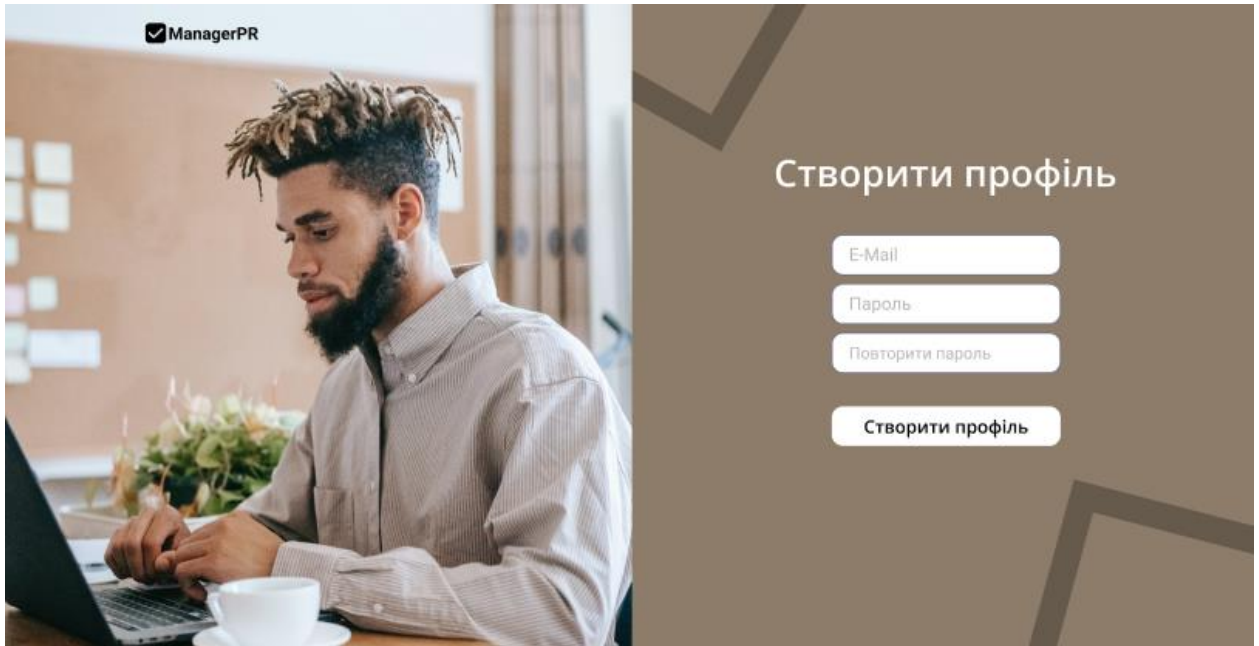


Рисунок 3.3 – Сторінка реєстрації

2) Після входу або реєстрації користувач попадає на головну сторінку сайта, де є список усіх проєктів. (Рис. 3.4)

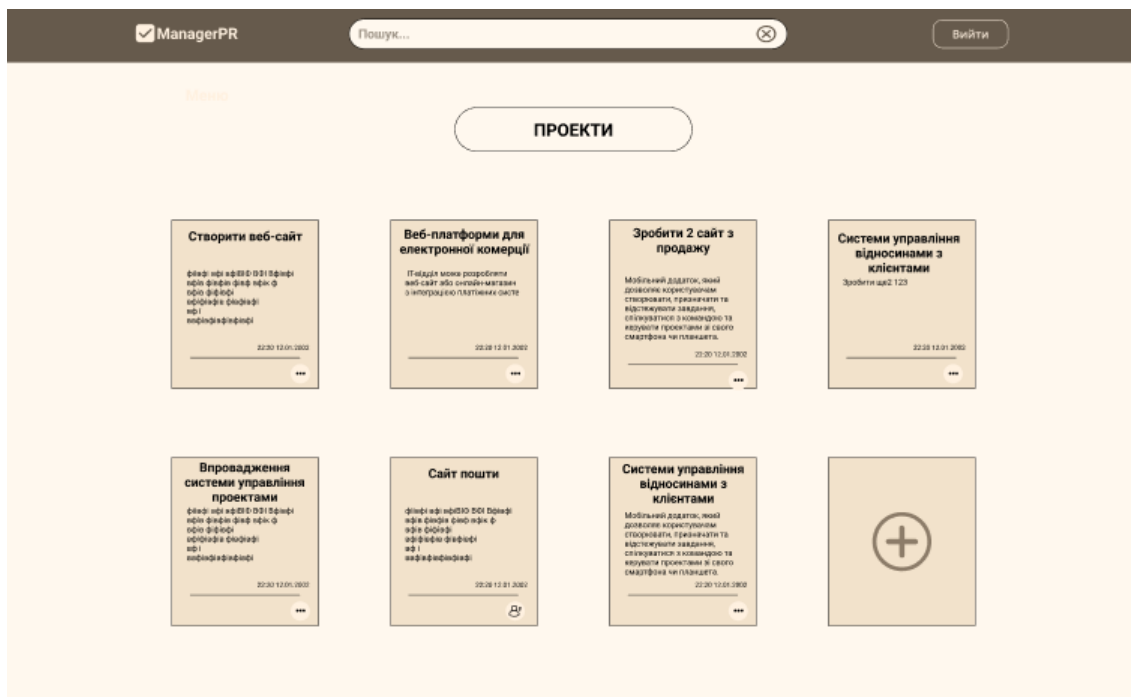


Рисунок 3.4 – Головна сторінка

3) На головній сторінці користувач може користуватися пошуком, створювати нові проекти, та взаємодіяти із вже створеними проектами. (Рис. 3.5, 3.6, 3.7)

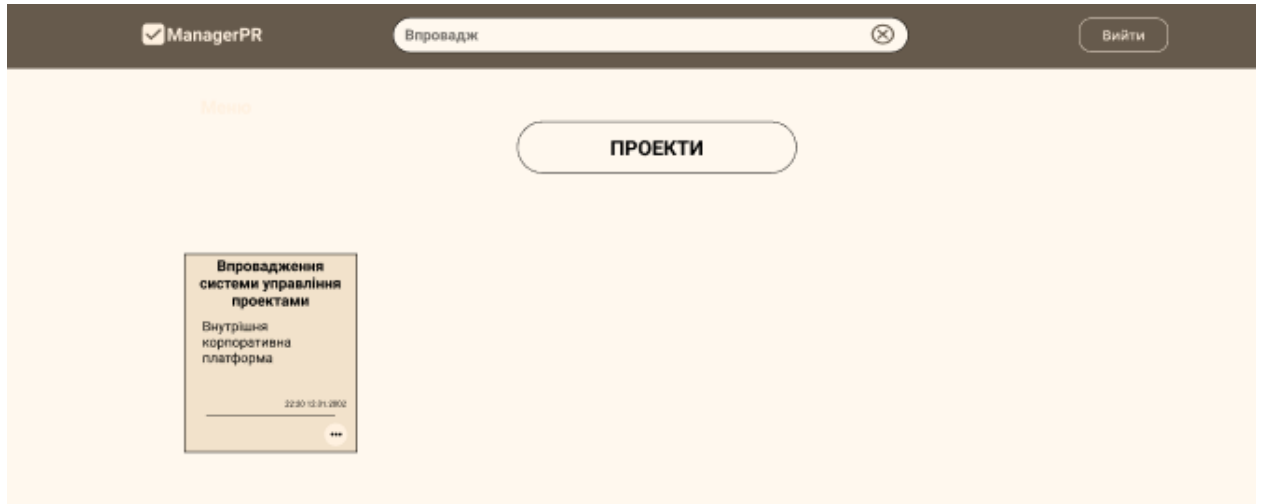


Рисунок 3.5 – Функція пошуку

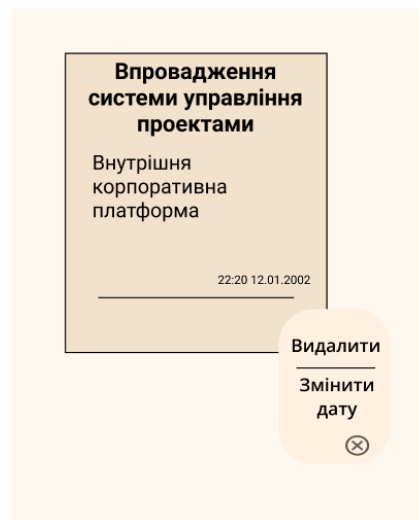


Рисунок 3.6 – Додаткове меню

The screenshot shows a mobile application interface for creating a new project. The window is titled "Створення нового проекту" and has a close button in the top right corner. The form includes the following fields:

- Назва проекту:** A text input field.
- Дата завершення:** A date selection field.
- Опис проекту:** A large text area for project description.
- Список учасників:** A search input field containing "ПІБ" and a plus icon below it.
- Задачі:** A search input field and a plus icon below it.

At the bottom of the form, there are two buttons: "Створити проект" (Create project) and "Назад" (Back).

Рисунок 3.7 – Вікно створення нового проекту

4) Після успішного створення нового проекту користувач отримує можливість відкрити меню проекту, де доступна повна інформація про даний проект. Це меню надає зручний спосіб організації та керування проектом. В ньому міститься наступна інформація:

1. Дати: В меню проекту користувач зможе переглянути важливі дати, пов'язані з проектом.
2. Задачі: Меню проекту також надає можливість перегляду всіх задач, пов'язаних з проектом. Користувач зможе побачити список задач, описи,
3. Список учасників: В меню проекту буде доступний список учасників, які беруть участь у проекті. Користувач зможе побачити імена (Рис.3.8)

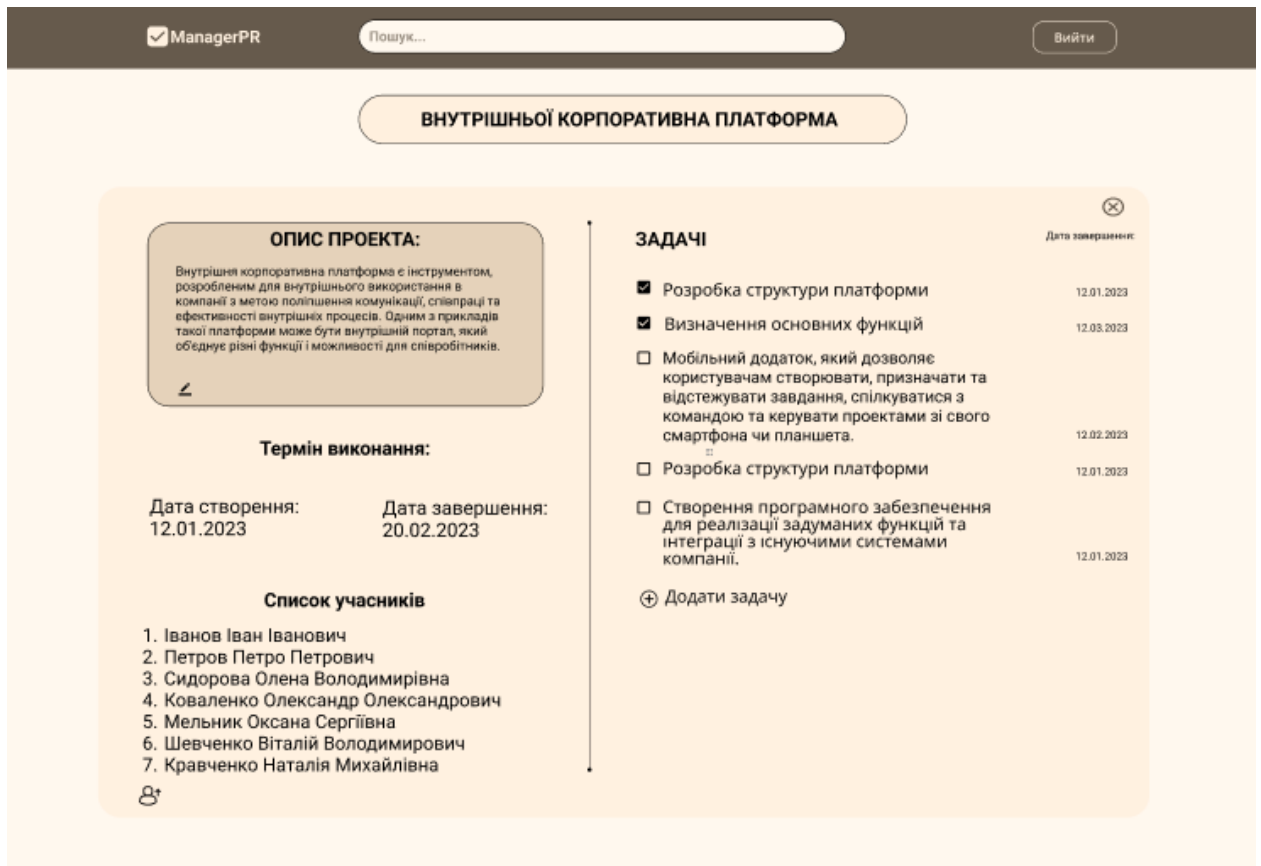


Рисунок 3.8 – Меню проекту

5) В меню проекту користувач може додавати нових учасників, змінювати опис та назву проекту, та задачі.

Навівши курсор на задачу, користувач отримує додаткові функції для взаємодії із задачею, такі як: видалення задачі та зміна назви. (Рис.3.9)

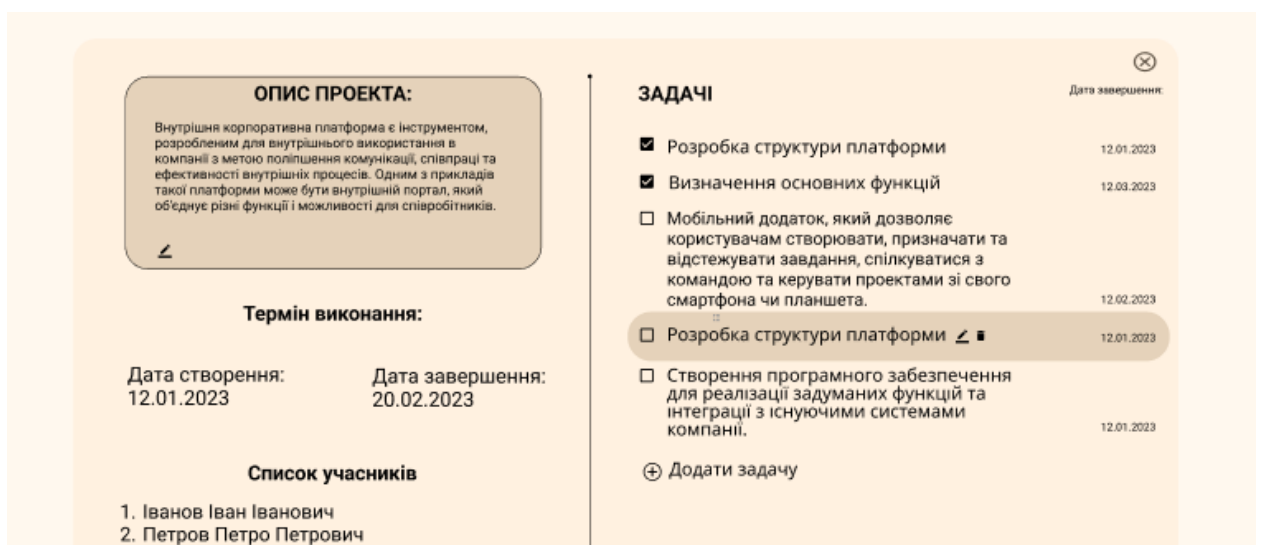


Рисунок 3.9 – Взаємодія із задачею

6) Користувач може натиснути на конкретну задачу для отримання повної інформації про задачу(Рис.3.10)

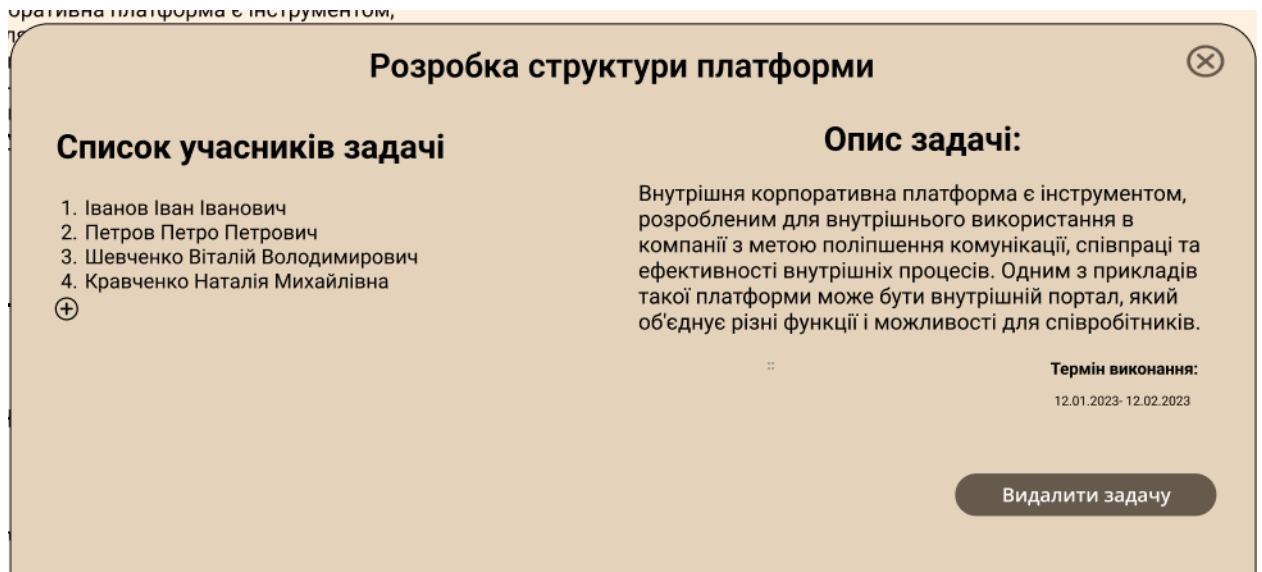


Рисунок 3.10 – Меню задачі

7) Після внесення всіх потрібних змін, користувач може повернутися до меню проекту або до списку проектів натиснувши кнопку праворуч-зверху.

3.4. Технічне та системне забезпечення розробки

Технічне та системне забезпечення сайту включає в себе використання різних технологій та інструментів для розробки та функціонування веб-сайту. Основні компоненти, що використовуються, включають HTML5, CSS (Less), JavaScript Framework Angular 16 та плагіни для роботи з JSON.

HTML5: HTML5 є останньою версією стандарту мови розмітки HTML. Вона надає розширені можливості для створення структури та вмісту веб-сторінок. HTML5 дозволяє використовувати нові теги, аудіо- та відеоелементи, графіку та багато іншого, що сприяє поліпшенню функціональності та візуального вигляду сайту.

CSS (Less): CSS (Less) є мовою стилізації, що використовується для оформлення та визначення вигляду елементів HTML. За допомогою CSS

можна встановлювати кольори, шрифти, розташування елементів та багато іншого. Мова Less є розширенням CSS, яке додає більше можливостей, таких як змінні, міксини та імпорт, що полегшують створення та керування стилями.

JavaScript Framework Angular 16: Angular 16 є одним з найпопулярніших фреймворків JavaScript для розробки веб-додатків. Він надає інструменти та компоненти для створення динамічного та інтерактивного веб-інтерфейсу. Angular 16 базується на TypeScript і має потужні можливості для створення односторінкових додатків (SPA) та розширення функціональності сайту.

Плагін json: JSON (JavaScript Object Notation) є легким форматом обміну даними, що використовується для передачі та збереження структурованих інформаційних об'єктів.

3.5. Обґрунтування вибору технічних засобів

Дані технології були обрані через такі моменти:

1. HTML5 (HyperText Markup Language 5): HTML5 є останньою версією стандарту мови розмітки HTML. Вона має багато нових елементів, атрибутів та функцій, що полегшують створення більш сучасних та потужних веб-сторінок. HTML5 дозволяє розробникам використовувати багато медіаелементів, графіку, анімації, форм та інших елементів, що покращують користувацький досвід.
2. CSS (Less) (Cascading Style Sheets): CSS використовується для стилізації та оформлення веб-сторінок. Використання CSS дозволяє розробникам контролювати вигляд, кольори, шрифти, розміри та розташування елементів на сторінці. Less є розширенням CSS, яке надає додаткові функції, такі як змінні, міксини та можливість вкладення стилів.

Використання CSS (Less) дозволяє писати більш структурований, підтримуваний та повторно використовуваний код стилів.

3. JavaScript Framework Angular 16: Angular 16 є потужним JavaScript-фреймворком для розробки веб-додатків. Він пропонує широкі можливості для створення високопродуктивних та масштабованих додатків. Angular 16 використовується для побудови клієнтської частини додатків, забезпечуючи розширюваність, модульність та швидку реакцію на зміни. Він пропонує розумний двосторонній зв'язок даних, компонентну архітектуру, маршрутизацію та багато інших функцій, які полегшують розробку веб-додатків.
4. Плагін JSON: JSON (JavaScript Object Notation) є легким форматом обміну даними, який використовується для представлення структурованої інформації. JSON дозволяє зручно зберігати та передавати дані у вигляді об'єктів та масивів. Плагін JSON використовується для роботи з цим форматом даних в рамках веб-сайту. Він дозволяє легко серіалізувати та десеріалізувати дані, здійснювати запити до зовнішніх API, передавати дані між клієнтом та сервером та забезпечує зручні методи для роботи з даними у форматі JSON.

Використання цих технологій дозволяє створити потужний та сучасний веб-сайт зі зручним інтерфейсом, ефектним дизайном та високою продуктивністю. Комбінація HTML5, CSS (Less), Angular 16 та плагіна JSON дозволяє розробникам створювати динамічні, інтерактивні та швидкі веб-додатки з багатьма функціями та можливостями.

3.6. Топологія комп'ютерної мережі

В даному випадку, сайт розміщений на хостингу і користувач заходить на нього через браузер, для цього використана топологію "зірка" (star topology). У цій топології, хостинговий сервер, на якому розміщений сайт, є

центральним вузлом, до якого підключені всі клієнтські пристрої (браузери) через Інтернет.

Клієнтські пристрої, такі як комп'ютери, смартфони або планшети, використовують мережу Інтернет для доступу до хостингового сервера, де розташований сайт. Кожен клієнтський пристрій здійснює окреме з'єднання з сервером за допомогою протоколу HTTP або HTTPS.

Ця топологія надає простоту управління та керування, оскільки всі клієнтські пристрої звертаються до одного центрального сервера. Хостинговий сервер забезпечує обробку запитів від користувачів та передає відповіді назад. Також, в разі необхідності, можливо використовувати додаткові сервіси або бази даних, які також знаходяться на хостинговому сервері.

Важливим аспектом цієї топології є надійність та доступність хостингового сервера, оскільки він є центральною точкою доступу до вашого сайту для всіх користувачів.

3.7. Обґрунтування вибору ОС та протоколу обміну даними

Одним з ключових аспектів веб-розробки є вибір оптимальної операційної системи (ОС) та протоколу обміну даними для забезпечення надійності та безпеки функціонування веб-сайту. З урахуванням сучасних вимог до веб-систем, різноманітних операційних середовищ та технологій, важливо проаналізувати відповідність цих компонентів проекту.

Щодо операційної системи, важливо враховувати широкий спектр підтримуваних ОС, таких як Windows, Linux та macOS. Це дає змогу підібрати оптимальну ОС відповідно до ваших унікальних вимог, компетенцій та обмежень. Такий вибір забезпечує гнучкість інфраструктури і дозволяє розгорнути веб-сайт на будь-якому хостинговому середовищі.

Однак, важливим аспектом є вибір протоколу обміну даними. У цьому контексті рекомендованим протоколом є HTTPS - захищений протокол передачі гіпертексту. Використання протоколу HTTPS гарантує конфіденційність, цілісність та аутентичність даних, що передаються між клієнтом та сервером. Це досягається шляхом шифрування даних за допомогою протоколу SSL/TLS та використанням сертифікатів безпеки.

Однією з основних переваг використання протоколу HTTPS є забезпечення безпеки користувачів під час взаємодії з веб-сайтом. Враховуючи розширення функціоналу веб-сайтів та збільшення обсягу передаваної конфіденційної інформації, важливо забезпечити захист персональних даних, аутентифікацію користувачів та запобігання можливим атакам.

Таким чином, вибір оптимальної ОС та протоколу обміну даними для веб-сайту є критичним завданням. Вибір будь-якої операційної системи дозволяє адаптувати веб-сайт до різних хостингових середовищ, забезпечуючи гнучкість та комфорт при розгортанні та підтримці проекту. Водночас, використання протоколу HTTPS дозволяє забезпечити безпеку та конфіденційність передачі даних, покращуючи довіру користувачів та виконання сучасних стандартів безпеки веб-додатків.

РОЗДІЛ 4. ОХОРОНА ПРАЦІ

Охорона праці є невід'ємною складовою успішного функціонування будь-якої організації. Вона орієнтована на забезпечення безпеки та здоров'я працівників, запобігання ризикам, негативним наслідкам та постійному вдосконаленню умов праці. Охорона праці включає в себе визначення та оцінку потенційних небезпек, встановлення профілактичних заходів, навчання працівників правилам безпеки, Адекватна організація охорони праці включає ідентифікацію потенційних ризиків та небезпек, що можуть виникнути в робочому середовищі, а також розробку та впровадження ефективних заходів для їх запобігання та зниження. Це можуть бути технічні заходи, які передбачають встановлення безпечного обладнання, дотримання норм та стандартів безпеки, а також організаційні заходи, спрямовані на раціоналізацію робочих процесів та впровадження правильних методів праці. Регулярне навчання працівників правилам безпеки, надання інформації щодо ризиків та небезпек, а також впровадження системи контролю за дотриманням вимог охорони праці є необхідними складовими успішної системи охорони праці.

Окрім цього, необхідно передбачити запобіжні заходи та процедури в разі виникнення надзвичайних ситуацій, таких як пожежа, аварія чи інші небезпечні події. Відповідні навички та знання дозволять працівникам ефективно реагувати на небезпеку, зменшити можливі наслідки та забезпечити швидку евакуацію в разі потреби. контроль за виконанням норм та стандартів, а також організацію ефективної системи управління охороною праці. Раціональне використання ресурсів, усунення причин виникнення небезпек та створення безпечного та здорового робочого середовища сприяє підвищенню продуктивності, задоволеності працівників та досягненню успіху в діяльності організації.

1. Загальна характеристика робочого середовища:

- Вимоги до освітлення: Забезпечення належного освітлення на робочих місцях, враховуючи вимоги до рівня освітленості, уникання сліпоти, рівномірного розподілу світла тощо.
 - Вимоги до вентиляції: Забезпечення достатнього обміну повітря в приміщенні, враховуючи норми щодо концентрації шкідливих речовин, регулювання температури та вологості повітря.
 - Вимоги до температури: Забезпечення комфортного рівня температури в робочому середовищі, уникання перегріву або переохолодження працівників.
2. Ідентифікація ризиків:
- Фізичні ризики: Виявлення потенційних загроз, таких як можливість травмування, пошкодження обладнання, несправність електричних систем та інші фізичні небезпеки.
 - Ергономічні ризики: Визначення факторів, що можуть призводити до неправильної позиції тіла, напруження м'язів, повторних рухів, і розробка заходів для попередження таких проблем.
3. Заходи щодо запобігання та зниження ризиків:
- Організація робочого місця: Забезпечення належного розташування обладнання, правильного налаштування робочого меблів, використання ергономічних пристроїв для запобігання неправильної позиції тіла.

Охорона праці робочого місця робітника є важливою складовою забезпечення безпеки і здоров'я працюючих. Нижче наведено загальні правила охорони праці, які робітники повинні дотримуватися:

1. Знання і дотримання правил: Робітники повинні бути ознайомлені з правилами та процедурами охорони праці, пов'язаними з їхнім конкретним робочим місцем. Вони повинні розуміти ці правила та дотримуватися їх усіх часів.

2. Використання особистого захисного обладнання (ОЗО): Робітники повинні носити та правильно використовувати необхідне ОЗО, таке як захисні каски, окуляри, вушні затички, респіратори тощо, залежно від ризиків, пов'язаних з їхньою роботою.
3. Запобігання пожежам та вибухам: Робітники повинні дотримуватися правил пожежної безпеки, включаючи правильне зберігання та використання легкозаймистих матеріалів, регулярну перевірку пожежних пристроїв та знання процедур евакуації в разі пожежі.
4. Попередження про небезпеку: Робітники повинні бути уважними до потенційних небезпек, які можуть виникнути на їхньому робочому місці, і повідомляти свого керівника або відповідний відділ про будь-які небезпеки, що потребують усунення.
5. Зберігання та обробка матеріалів: Робітники повинні правильно зберігати та обробляти матеріали, зокрема хімікати, токсичні речовини, гострі предмети тощо. Вони повинні знати процедури безпечної роботи з цими матеріалами та уникати потенційних небезпек.

Правила пожежної безпеки в ІТ-відділі є важливою складовою забезпечення безпеки та захисту працюючих, а також захисту обладнання та даних. Нижче наведено загальні правила пожежної безпеки, які повинні дотримуватися в ІТ-відділі:

1. Знання процедур евакуації: Усі працівники ІТ-відділу повинні бути ознайомлені з процедурами евакуації в разі виникнення пожежі. Це включає знання виходів, плану евакуації та місця збору після евакуації. Працівники повинні бути готові швидко та безпечно покинути приміщення.
2. Перевірка та обслуговування пожежних пристроїв: ІТ-відділ повинен мати правильно встановлені та функціонуючі пожежні пристрої, такі як вогнегасники та пожежні тривоги. Регулярна перевірка та обслуговування цих пристроїв повинні бути проведені згідно з вимогами безпеки.

3. Запобігання пожежам в електроустановках: ІТ-відділ повинен використовувати електроустановки, які відповідають стандартам безпеки. Електропроводка повинна бути правильно встановлена та регулярно перевірятися на наявність пошкоджень або неправильного підключення.
4. Зберігання та обробка запалювальних матеріалів: Зберігання запалювальних матеріалів, таких як легкозаймисті речовини, повинно відбуватися відповідно до правил безпеки. Вони повинні бути зберігані в спеціальних контейнерах та далеко від електричних апаратів або джерел вогню. Обробка цих матеріалів повинна відбуватися з дотриманням відповідних заходів безпеки.
5. Організація кабельного прокладання: Кабелі та проводи в ІТ-відділі повинні бути організовані таким чином, щоб уникнути перевантаження, перегрівання та короткого замикання. Кабелі повинні бути правильно прокладені та захищені від пошкоджень або небезпеки пожежі.

Важливо, щоб всі працівники ІТ-відділу були ознайомлені з цими правилами пожежної безпеки та дотримувалися їх для забезпечення безпеки на робочому місці та запобігання пожежам.

Охорона праці є надзвичайно важливим аспектом будь-якої організації, особливо в ІТ-відділах, де співробітники працюють з комп'ютерами та іншою складною технікою. Забезпечення безпеки та здоров'я працівників має вирішальне значення для їхньої продуктивності, добробуту та загального успіху компанії. Ось кілька причин, чому охорона праці є настільки важливою в ІТ-відділах:

Забезпечення безпеки працівників: У ІТ-відділах працівники можуть зіштовхнутися з різноманітними ризиковими ситуаціями, такими як електричний удар, пожежа, радіація, напруженість очей та розлади зі сторони м'язів через тривале сидіння перед комп'ютером. Охорона праці допомагає ідентифікувати потенційні небезпеки та вживати необхідні заходи для запобігання нещасним випадкам і захворюванням.

Зниження втрат часу та ресурсів: Нещасні випадки, травми та захворювання можуть призвести до відсутності працівників, зниження продуктивності та збитків для компанії. Відповідно до законодавства, компанії зобов'язані забезпечити безпечне робоче середовище для своїх працівників. Інвестування в охорону праці дозволяє зменшити ризики травм та захворювань, що в свою чергу сприяє збільшенню продуктивності та ефективності праці.

Законодавча вимога: Багато країн мають законодавство, яке регулює охорону праці та встановлює мінімальні стандарти безпеки та здоров'я для робочих місць. Недотримання цих вимог може призвести до юридичних проблем, штрафів та негативного впливу на репутацію компанії. Тому важливо, щоб ІТ-відділи дотримувалися вимог охорони праці, щоб уникнути правових проблем.

Покращення робочого клімату: Охорона праці сприяє покращенню загального робочого клімату в ІТ-відділах. Застосування політик та процедур, спрямованих на забезпечення безпеки та здоров'я працівників, демонструє їхню важливість та дбайливість з боку керівництва. Це може позитивно позначитися на мотивації працівників, їхній відданості компанії та зниженні показників втрати персоналу.

Інновації та продуктивність: Безпека та здоров'я працівників мають прямий вплив на їхню продуктивність та здатність до інновацій. Забезпечення комфортних робочих умов, відповідного освітлення, належної організації робочих місць та відпочинку сприяє збереженню енергії та підвищенню концентрації працівників. Це може сприяти зростанню творчості та ефективності в ІТ-відділах.

У підсумку, охорона праці є невід'ємною частиною успішного функціонування ІТ-відділів. Забезпечення безпеки та здоров'я працівників не

тільки сприяє їхньому благополуччю, але й позитивно впливає на продуктивність, ефективність та репутацію компанії.

Охорона праці в ІТ-відділах також є ключовим елементом створення здорової організаційної культури. Працівники, які відчують, що їхнє благополуччя та безпека є пріоритетом для компанії, більш схильні до співпраці, творчого мислення та інновацій. Підтримка та імплементація ефективних програм охорони праці сприяють створенню сприятливого середовища, де працівники можуть розвиватися, досягати свого потенціалу та сприяти успіху компанії в цифровій епохі.

ВИСНОВОК

В данній кваліфікаційній роботі була розроблена інформаційна система для керування проектами в ІТ-відділі компанії ВЕНКОН. Метою роботи було створення ефективного і зручного інструменту для керування проектами, що дозволить полегшити роботу менеджера, розподілити завдання між працівниками та забезпечити успішну реалізацію проектів.

Для досягнення цієї мети були визначені та розроблені різноманітні вимоги до системи. Вимоги до функціональності включали можливість створення проектів, розподіл завдань, моніторинг виконання, та багато іншого. Вимоги до структури передбачали чітку ієрархію проектів, завдань та користувачів системи. Вимоги до збереження даних включали використання формату JSON для зручного збереження та обміну даними. Вимоги до надійності передбачали належне резервне копіювання даних та захист інформації. Вимоги до лінгвістичного забезпечення передбачали підтримку різних мов та локалізацію інтерфейсу. Вимоги до програмного забезпечення включали підтримку різних операційних систем та забезпечення мінімального використання ресурсів комп'ютера.

В результаті реалізації дипломного проекту була створена інформаційна система, що відповідає поставленим вимогам. Система забезпечує зручний і ефективний процес управління проектами, полегшує роботу менеджера та підвищує продуктивність всієї команди. Застосування такої системи у веб-розробці компанії ВЕНКОН допоможе забезпечити успішну реалізацію проектів та досягнення поставлених цілей.

Отже, розробка інформаційної системи для керування проектами в ІТ-відділі компанії ВЕНКОН є актуальною та важливою завданням, яке сприятиме покращенню ефективності та якості веб-розробки. Виконання цієї кваліфікаційної роботи відповідає сучасним вимогам та впровадження такої системи може стати вагомим кроком до досягнення успіху в сфері ІТ-управління проектами.

Імплементація розробленої інформаційної системи для управління проектами в ІТ-відділі компанії ВЕНКОН має потенціал забезпечити позитивний вплив на робочий процес, сприяючи покращенню координації, комунікації та ефективності роботи в команді. Застосування такої системи дозволить зменшити ризики та небезпеки, пов'язані з управлінням проектами, забезпечити належний контроль та моніторинг, а також підвищити якість та своєчасність виконання проектів. В цілому, впровадження розробленої системи сприятиме підвищенню продуктивності та досягненню більш високих результатів у сфері управління проектами в ІТ-відділі компанії ВЕНКОН.

ВИКОРИСТАНІ ДЖЕРЕЛА:

1. JavaScript Docs [Список використаних джерел] – Режим доступу до ресурсу: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
Дата звернення (29.05.2023)
2. CSS(Less) Documentation [Список використаних джерел] – Режим доступу до ресурсу: <https://lesscss.org/> Дата звернення (25.05.2023)
3. М'якшило О.М. CASE-технології у проектуванні інформаційних систем:
Проектування інформаційних систем : лабораторний практикум для студ. Частина 2 "Проектування клієнтського додатку" / уклад. : О. М. М'якшило, О.В. Харкянен; Нац. ун-т харч. технол. - Київ : НУХТ, 2017. - 33 с.
4. ДСТУ 3321:2003. Системи конструкторської документації. Терміни та визначення основних понять.
5. Documentation for the npm registry, website, and command-line interface[Список використаних джерел] – Режим доступу до ресурсу: <https://docs.npmjs.com/> Дата звернення (02.06.2023)
6. ДСТУ ISO 6309:2007. Пожежна безпека. Загальні вимоги.
7. TypeScript Documentation. o [Список використаних джерел] – Режим доступу до ресурсу: <https://www.typescriptlang.org/docs/> Дата звернення (29.06.2023)
8. Angular docs [Список використаних джерел] – Режим доступу до ресурсу: <https://angular.io/docs> Дата звернення (05.06.2023)
9. ДСТУ ISO/IEC 12207:2014. Інженерія систем і програмного забезпечення. Процеси життєвого циклу програмного забезпечення.

10. Angular Material [Список використаних джерел] – Режим доступу до ресурсу: <https://material.angular.io>
11. ДСТУ ISO/IEC 29155-1:2015. Розроблення систем і програмного забезпечення. Платформи для тестування проєктів з розроблення інформаційних систем. Частина 1. Концепції та визначення.
12. ДСТУ 3008:2015 — Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання. – К.: ДП «УкрНДНЦ», 2015. – 32 с.
13. Системний аналіз [Список використаних джерел]: Методичні рекомендації до практичних занять для студентів напряму підготовки 6.050101 «Комп’ютерні науки» денної та заочної форм навчання/ уклад. В. В. Самсонов. – К.: НУХТ, 2013. – 28 с
14. ДСТУ 2226:1993. Автоматизовані системи.
15. ДСТУ ISO/IEC TR 15504. Інформаційні технології. Оцінювання процесів життєвого циклу програмних засобів. – 315 с.
16. М’якшило О.М. CASE-технології у проєктуванні інформаційних систем: електронний навчальний посібник для студ. вищих навч. закладів / О.М. М’якшило, Л.Г. Загоровська,– К.: НУХТ, 2017. – 190 с.
17. Методичні рекомендації до виконання кваліфікаційної роботи на здобуття освітнього ступеня «Бакалавр» спеціальності 122 «Комп’ютерні науки» освітньо-професійної програми «Комп’ютерні науки» денної та заочної форм навчання / уклад. : Л. Г. Загоровська, О. М. М’якшило, М. П. Костіков. – К. : НУХТ, 2020. – 30 с
18. Теорія прийняття рішень [Список використаних джерел]: лаборат. практикум для студентів освітнього ступеня «бакалавр» спеціальності 122 «Комп’ютерні науки та інформаційні технології» денної та заочної форм навчання / уклад. Л.Г. Загоровська, С.В. Грибков, Т.В. Ярова – К.: НУХТ, 2016. – 43 с.

19. CASE-технології для інжинірингу бізнес-процесів підприємств [Електронний ресурс]: лаборат. практикум для студ. спец. 8.05010101 «Інформаційні управляючі системи і технології», 8.05010105 «Комп'ютерний еколого-економічний моніторинг» денної форми навчання / уклад. Л.Г. Загоровська, С.В. Грибков – К.:НУХТ, 2015. – 40с.
20. Структурне моделювання інформаційних систем [Текст] : метод. вказівки до викон. лаборатор. робіт. для студ. спец. 8.080400 "Інформаційні управляючі системи і технології" напряму 0804 "Комп'ютерні науки" ден. і заоч. форм навч. Ч. 1 / О. М. М'якшило, Л. Г. Загоровська, О. В. Харкянєн ; Нац. ун-т харч. технол. — К. : НУХТ, 2010. — 26 с.
21. Управління ІТ-проектами: методичні рекомендації до виконання курсового проекту для студентів освітнього ступеня «Бакалавр» спеціальності 122 «Комп'ютерні науки» денної та заочної форм навчання. Уклад: М. В. Гладка, О. А. Хлобистова: К. НУХТ, 2014.– 91 с.
22. ДСТУ 3008-2015. Документація. Звіти у сфері науки і техніки. Структура та правила оформлювання
23. ДСТУ EN 12464-1:2016. Світло та освітлення. Освітлення робочих місць.
24. JSON Documentation Advanced [Електронний ресурс] – Режим доступу до ресурсу: <https://www.json.org/json-en.html> Дата звернення (31.05.2023)
25. CSS Advanced [Електронний ресурс] – Режим доступу до ресурсу: <https://www.w3schools.com/css> Дата звернення (02.06.2023)

ДОДАТКИ

Додаток А. Скріншоти програми

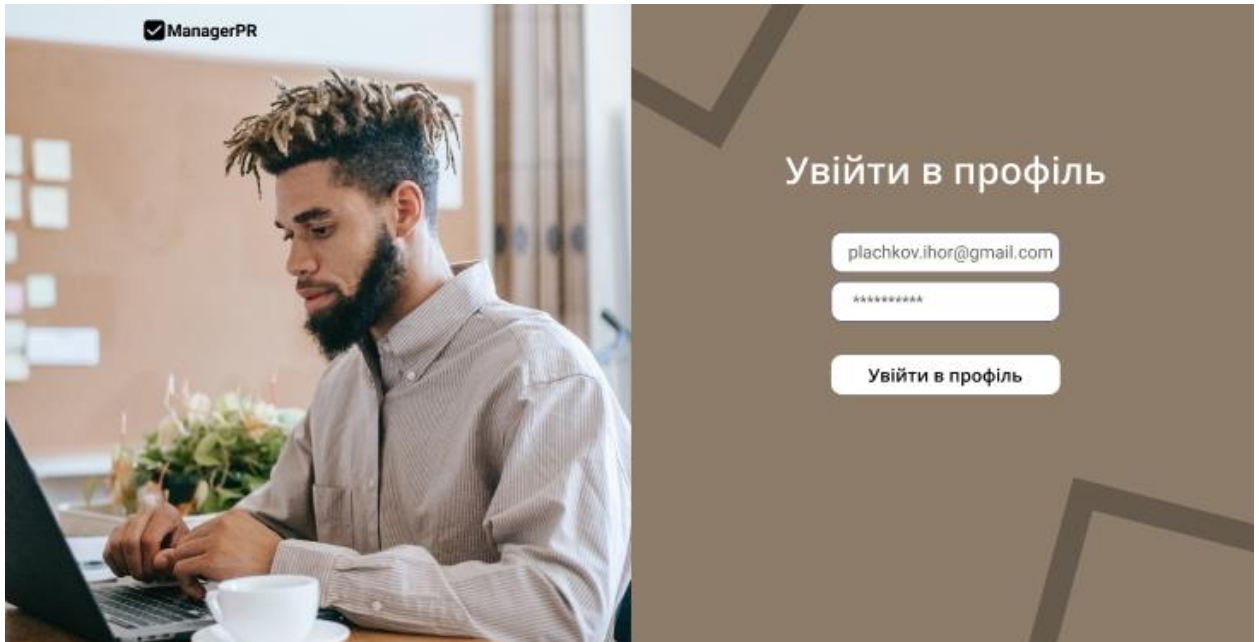


Рисунок А.1 – Заповнена форма

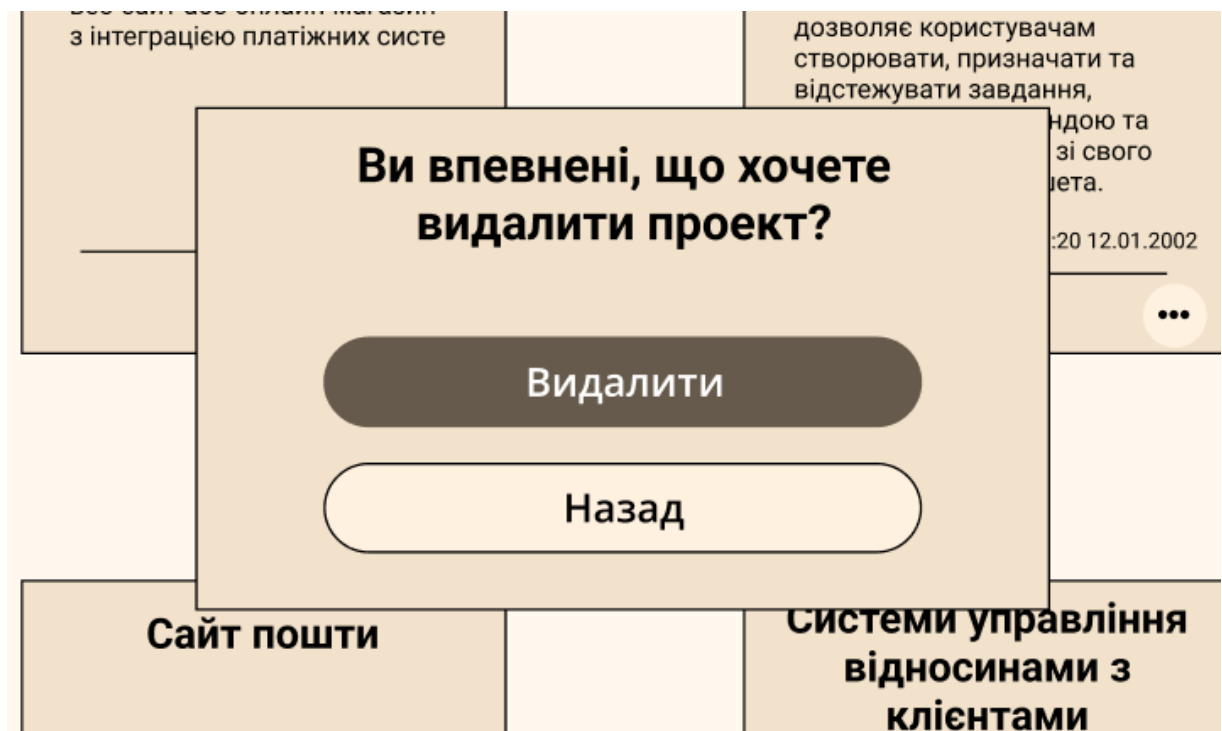


Рисунок А.2 – Вікно видалення

The screenshot shows a web application interface for creating a new project. The header includes the 'ManagerPR' logo and a 'Вийти' (Logout) button. The main form is titled 'Створення нового проекту' and is divided into several sections:

- Назва проекту:** A text input field containing 'Сайт з продажу'.
- Дата завершення:** A date input field containing '12.01.2023'.
- Опис проекту:** A text area containing the text: 'Сайт дозволяє користувачам створювати, призначати та відстежувати завдання, спілкуватися з командою та керувати проектами з'.
- Список учасників:** A list of participants, currently containing 'Іванов Іван Іванович' and 'ПІБ', with a '+' button to add more.
- Задачі:** A list of tasks, currently containing 'Зробити головну сторінку' with a completion date of '12.01.2023', and an empty task field with a completion date field, plus a '+' button to add more.

At the bottom of the form, there are two buttons: 'Створити проект' (Create project) and 'Назад' (Back).

Рисунок А.3 – Заповнена форма створення проекту

Додаток В. Код системи:**Код системи:**

```
"version": "0.2.0",
"configurations": [
  {
    "name": "ng serve",
    "type": "chrome",
    "request": "launch",
    "preLaunchTask": "npm: start",
    "url": "http://localhost:4200/"
  },
  {
    "name": "ng test",
    "type": "chrome",
    "request": "launch",
    "preLaunchTask": "npm: test",
    "url": "http://localhost:9876/debug.html"
  }
]
}
"tasks": [
  {
    "type": "npm",
    "script": "start",
    "isBackground": true,
    "problemMatcher": {
      "owner": "typescript",
      "pattern": "$tsc",
      "background": {
```

```
"activeOnStart": true,
"beginsPattern": {
  "regexp": "(.*?)"
},
"endsPattern": {
  "regexp": "bundle generation complete"
}
}
},
{
  "type": "npm",
  "script": "test",
  "isBackground": true,
  "problemMatcher": {
    "owner": "typescript",
    "pattern": "$tsc",
    "background": {
      "activeOnStart": true,
      "beginsPattern": {
        "regexp": "(.*?)"
      },
      "endsPattern": {
        "regexp": "bundle generation complete"
      }
    }
  }
}
}
]
```

```
}  
import { NgModule } from '@angular/core';  
import { RouterModule, Routes } from '@angular/router';  
  
const routes: Routes = [];  
  
@NgModule({  
  imports: [RouterModule.forRoot(routes)],  
  exports: [RouterModule]  
})  
export class AppRoutingModule { }  
import { TestBed } from '@angular/core/testing';  
import { RouterTestingModule } from '@angular/router/testing';  
import { AppComponent } from './app.component';  
  
describe('AppComponent', () => {  
  beforeEach(() => TestBed.configureTestingModule({  
    imports: [RouterTestingModule],  
    declarations: [AppComponent]  
  }));  
  
  it('should create the app', () => {  
    const fixture = TestBed.createComponent(AppComponent);  
    const app = fixture.componentInstance;  
    expect(app).toBeTruthy();  
  });  
  
  it('should have as title \'ManagerPR\'', () => {  
    const fixture = TestBed.createComponent(AppComponent);
```

```
const app = fixture.componentInstance;
expect(app.title).toEqual('ManagerPR');
});

it('should render title', () => {
  const fixture = TestBed.createComponent(AppComponent);
  fixture.detectChanges();
  const compiled = fixture.nativeElement as HTMLElement;
  expect(compiled.querySelector('.content
span)?.textContent).toContain('ManagerPR app is running!');
});
});
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';

@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule
  ],
  providers: [],
  bootstrap: [AppComponent]
```