

БАКАЛАВРСЬКА РОБОТА

БР. ІІІ - 45.00.00.000 ІІЗ

Група ІІІ-21-4

Варинський Арсен

2025

Івано-Франківський національний технічний університет нафти і газу

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

Варинський Арсен Захарович

(прізвище, ім'я, по батькові)

УДК 004
(індекс)

БАКАЛАВРСЬКА РОБОТА

Реалізація планувальника персональних завдань

(назва роботи)

Інженерія програмного забезпечення

(назва освітньої програми)

121 - Інженерія програмного забезпечення

(шифр і назва спеціальності)

Робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Здобувач освітнього рівня Варинський А.З.
(підпис, ініціали та прізвище здобувача)

Науковий керівник Романишин Юлія Любомирівна, д.п.н., професор
(підпис, прізвище, ім'я, по батькові, науковий ступінь, вчене звання керівника)

Допущено до захисту
Завідувач кафедри

доц. Бандура В.В.
(посада) (підпис) (дата) (ініціали та прізвище)

Івано-Франківськ – 2025

6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 28 квітня 2025 р.

Керівник _____

(підпис)

Завдання прийняв до виконання _____

(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту	Примітка
1	Аналіз застосування інформаційних технологій до побудови планувальника завдань	04.05.2025	виконано
2	Аналіз вимог та реалізація архітектури планувальника завдань та управління проектами	15.05.2025	виконано
3	Представлення випадків використання планувальника завдань	21.05.2025	виконано
4	Програмна розробка системи управління і планування завдань	28.05.2025	виконано
5	Функціональні можливості інтерфейсу адміністратора, менеджера, співробітника	03.06.2025	виконано
6	Оформлення пояснювальної записки дипломної роботи завідувачем кафедри	10.06.2025	виконано

Студент – дипломник _____

(підпис)

Керівник роботи _____

(підпис)

АНОТАЦІЯ

Бакалаврська робота містить 75 сторінок, 31 рисунок, список використаних джерел із 41 найменуванням, 1 додаток.

Мета роботи - розробити веб-платформу для планування персональних завдань та управління проектами з реалізацією багаторольової моделі, сучасного інтерфейсу, забезпеченням зручної взаємодії користувачів і надійного зберігання даних.

Об'єкт дослідження - процес управління персональними та командними завданнями за допомогою цифрових засобів.

Предмет дослідження - методи, моделі та засоби реалізації веб-застосунку для планування завдань із багаторівневою рольовою структурою та інтерактивним інтерфейсом.

В першому розділі проведено огляд функціоналу планувальників, обґрунтовано розробку власного рішення та проаналізовано сучасні платформи.

В другому розділі сформульовано вимоги до системи, обрано технології реалізації та представлено архітектуру платформи.

В третьому розділі реалізовано веб-застосунок з урахуванням усіх функціональних ролей, проведено тестування та запропоновано шляхи розвитку

Висновок: реалізований планувальник персональних завдань є ефективним, масштабованим інструментом для організації командної роботи та управління проектною діяльністю, який відповідає сучасним вимогам цифрового середовища.

КЛЮЧОВІ СЛОВА: ПЛАНУВАЛЬНИК ЗАВДАНЬ, УПРАВЛІННЯ ПРОЄКТАМИ, ВЕБ-ДОДАТОК, РОЛЬОВА МОДЕЛЬ, SPRING BOOT, REACT, MYSQL, АВТЕНТИФІКАЦІЯ, ІНТЕРФЕЙС КОРИСТУВАЧА, ЦИФРОВЕ ПЛАНУВАННЯ.

ANNOTATION

The Bachelor's thesis contains 75 pages, 31 figures, a list of 41 references, and 1 appendix.

The purpose of the work is to develop a web platform for personal task planning and project management, featuring a multi-role model, a modern interface, convenient user interaction, and reliable data storage.

The object of research is the process of managing personal and team tasks using digital tools.

The subject of research is the methods, models, and tools for implementing a web application for task planning with a multi-level role structure and an interactive interface.

The first chapter reviews the functionality of planners, substantiates the development of a proprietary solution, and analyzes modern platforms.

The second chapter formulates system requirements, selects implementation technologies, and presents the platform's architecture.

The third chapter implements the web application, taking into account all functional roles, conducts testing, and proposes avenues for future development.

Conclusion: The implemented personal task planner is an effective, scalable tool for organizing teamwork and managing project activities, meeting the modern requirements of the digital environment.

KEYWORDS: TASK PLANNER, PROJECT MANAGEMENT, WEB APPLICATION, ROLE MODEL, SPRING BOOT, REACT, MYSQL, AUTHENTICATION, USER INTERFACE, DIGITAL PLANNING.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	9
ВСТУП.....	10
РОЗДІЛ 1. АНАЛІЗ ТА ДОСЛІДЖЕННЯ ЗАСТОСУВАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДО ПОБУДОВИ ПЛАНУВАЛЬНИКА ЗАВДАНЬ	14
1.1. Пропонована функціональність планувальника персональних завдань та управління проектами	14
1.1.1. Рольова модель та функціональні повноваження	14
1.1.2. Інтерфейс користувача та засоби безпеки	15
1.2. Обґрунтування дослідження та розробки веб-додатка для управління завданнями та проектами	16
1.2.1. Передумови	16
1.2.2. Значимість проекту.....	16
1.2.3. Мета проекту.....	18
1.3. Аналіз існуючих платформ планувальника персональних завдань та управління проектами.....	19
1.3.1. Платформа для управління робочими процесами Monday.com.....	19
1.3.2. Інструмент для управління завданнями та проектами MeisterTask	22
1.3.3. Платформа для управління робочими процесами та проектами Wrike.....	25
1.3.4. Веб-платформа Notion	28
РОЗДІЛ 2. АНАЛІЗ ВИМОГ ТА РЕАЛІЗАЦІЯ АРХІТЕКТУРИ ПЛАНУВАЛЬНИКА ПЕРСОНАЛЬНИХ ЗАВДАНЬ ТА УПРАВЛІННЯ ПРОЕКТАМИ.....	32

					БР.ІІ – 45.00.00.000 ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	Реалізація планувальника персональних завдань ПОЯСНЮВАЛЬНА ЗАПИСКА	Літ.	Арк.	Акрушіє
Розроб.		Варинський А.						
Перевір.		Романишин Ю.					6	
Реценз.						ІФНТУНГ ІІ-21-4		
Н. Контр.		Піх М.М.						
Затверд.		Бандура В.В.						

2.1. Збір та аналіз вимог до веб-платформи управління завданнями.....	32
2.1.1. Функціональні вимоги.....	32
2.1.2. Функціональність за ролями.....	32
2.1.3. Загальна функціональність.....	33
2.1.4. Нефункціональні вимоги.....	33
2.2. Технічні вимоги до середовища розробки та експлуатації.....	34
2.2.1. Апаратні вимоги.....	34
2.2.2. Програмні вимоги.....	34
2.3. Інструментарій та технології реалізації.....	35
2.3.1. Spring Boot.....	35
2.3.2. React.....	37
2.3.3. MySQL.....	38
2.3.4. MySQL Workbench.....	39
2.3.5. Maven.....	40
2.3.6. Bootstrap.....	41
2.4. Представлення діаграми робочих процесів.....	42
2.5. Представлення випадків використання планувальника завдань.....	45

РОЗДІЛ 3. ПРОГРАМНА РОЗРОБКА СИСТЕМИ УПРАВЛІННЯ І

ПЛАНУВАННЯ ЗАВДАНЬ.....	49
3.1. Запропонована архітектура системи.....	49
3.1.1. Функціональні повноваження ролей.....	49
3.1.2. Забезпечення ефективності системи.....	51
3.2. Основна реалізація системи.....	51
3.2.1. Налаштування середовища розробки.....	51
3.2.2. Розробка фронтенду (UI).....	51
3.2.3. Розробка бекенду (API).....	52
3.2.4. Підключення до бази даних.....	52
3.2.5. Взаємодія фронтенду та бекенду.....	53
3.2.6. Локальне розгортання та тестування.....	53

3.3. Реалізація інтерфейсу головної сторінки, реєстрації та автентифікації	53
3.3.1. Головна сторінка	53
3.3.2. Процеси реєстрації та автентифікації	54
3.4. Функціональність ролі адміністратора	56
3.4.1. Управління менеджерами проектів	57
3.4.2. Управління проектами	58
3.4.3. Управління співробітниками	60
3.4.4. Загальні функції користувача	61
3.5. Функціональні можливості інтерфейсу менеджера	62
3.6. Функціональні можливості інтерфейсу співробітника	65
3.7. Перспективи розвитку та потенційні вдосконалення системи планування та управління завданнями	68
 ВИСНОВКИ	 70
ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	72
 ДОДАТКИ	
БІБЛІОГРАФІЧНА ДОВІДКА	

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

UX - User Experience - Користувацький досвід

DBMS - DataBase Management System - Система керування базами даних

MVC - Model-View-Controller - Модель-Вид-Контролер

JSON - JavaScript Object Notation - Нотація об'єктів JavaScript

REST - Representational State Transfer - Передавання репрезентативного стану

JWT - JSON Web Token - Веб-токен на основі формату JSON

CI/CD - Continuous Integration / Continuous Delivery - Безперервна інтеграція / безперервне постачання

					БР.ІІІ – 45.00.00.000 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

У сучасному динамічному цифровому середовищі зростає потреба в ефективному управлінні особистими та командними завданнями. Планувальники завдань відіграють ключову роль у підвищенні продуктивності, організації робочого часу, контролі над дедлайнами та координації проєктів. Більшість популярних інструментів зосереджені або на великих корпоративних рішеннях, або на індивідуальному тайм-менеджменті, проте не завжди враховують потреби гнучкої багаторівневої рольової моделі та зручної взаємодії між учасниками проєкту. Саме тому розробка веб-платформи, яка дозволяє поєднати інтуїтивність, масштабованість та ефективність, є актуальною та практично значущою задачею.

У період цифрової трансформації дедалі більшої актуальності набуває проблема ефективної організації робочого процесу, особливо у сфері управління проєктами та особистою продуктивністю. Нестача інструментів, які забезпечують просту, інтуїтивно зрозумілу і водночас функціонально гнучку взаємодію між учасниками проєкту, зумовлює необхідність створення нових програмних рішень. Особливої уваги потребують системи, що дозволяють об'єднати переваги персонального планування та колективного керування задачами в межах єдиного середовища.

Більшість доступних сьогодні платформ (наприклад, Trello, Asana, Wrike, Notion) мають потужний набір інструментів, однак часто виявляються надлишковими, складними в освоєнні або обмеженими в безкоштовних версіях. У багатьох із них реалізація рольової моделі є недостатньо гнучкою або вимагає інтеграції сторонніх модулів. Це створює потребу в розробці більш доступних та адаптивних рішень, зокрема для малого та середнього бізнесу, навчальних команд, студентських проєктів, фрилансерів тощо.

					БР.ІІІ – 45.00.00.000 ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

Запропонована в дипломній роботі система управління персональними завданнями дозволяє вирішити ці проблеми завдяки поєднанню сучасного технологічного стеку (Spring Boot, React, MySQL), продуманої архітектури, логічно реалізованої рольової структури та зручного веб-інтерфейсу. Особливу увагу було приділено простоті використання, безпеці автентифікації, розмежуванню повноважень і можливості масштабування.

В межах роботи було реалізовано веб-додаток, який охоплює процеси постановки, контролю, редагування завдань, розподілу ролей у команді, створення проєктів і ведення обліку. Система дозволяє не лише ефективно планувати особисту роботу, а й здійснювати централізоване управління діяльністю кількох користувачів з різними правами доступу.

Результати дослідження мають як теоретичну, так і практичну цінність. Теоретично було здійснено аналіз вимог, архітектурних підходів та існуючих систем, а практично — розроблено і протестовано працездатну програмну систему. Отримані результати можуть бути адаптовані під реальні потреби різних користувацьких груп, а також слугувати базою для подальших досліджень і вдосконалення.

Актуальність теми

Потреба в цифрових рішеннях для ефективного управління часом, ресурсами та завданнями є однією з ключових у професійному та освітньому середовищі. Існуючі системи (як-от Trello, Asana, Monday.com) мають обмеження у безкоштовних версіях, складні інтерфейси для новачків або недостатню гнучкість в управлінні ролями користувачів. Крім того, на тлі широкого впровадження гнучких методологій управління проєктами (Scrum, Kanban) виникає запит на кастомізовані інструменти, що дозволяють легко адаптувати функціонал до конкретних потреб. Запропонована в роботі система поєднує сучасні технології (Spring Boot, React, MySQL) та враховує потреби різних ролей – адміністратора, менеджера і співробітника, що підвищує її актуальність.

					БР.ІІІ – 45.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

Мета роботи - розробити веб-платформу для планування персональних завдань та управління проєктами з реалізацією багаторольової моделі, сучасного інтерфейсу, забезпеченням зручної взаємодії користувачів і надійного зберігання даних.

Завдання дослідження:

1. Проаналізувати сучасні інформаційні технології, що застосовуються для побудови систем управління завданнями.
2. Визначити функціональні та нефункціональні вимоги до веб-додатка.
3. Розробити архітектуру системи та обґрунтувати вибір технологічного стеку.
4. Реалізувати інтерфейс користувача та логіку взаємодії з бекендом.
5. Забезпечити реалізацію ролей і механізмів автентифікації.
6. Провести тестування системи в локальному середовищі.
7. Запропонувати напрями подальшого розвитку програмного продукту.

Об'єкт дослідження - процес управління персональними та командними завданнями за допомогою цифрових засобів.

Предмет дослідження - методи, моделі та засоби реалізації веб-застосунку для планування завдань із багаторівневою рольовою структурою та інтерактивним інтерфейсом.

Методи дослідження

- порівняльний аналіз існуючих систем;
- структурний аналіз функціональних вимог;
- проектування програмної архітектури;
- об'єктно-орієнтоване програмування (Java, React);
- моделювання робочих процесів (діаграми BPMN);
- тестування програмного забезпечення.

					БР.ІІІ – 45.00.00.000 ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

Наукова новизна

Розроблено структуру веб-застосунку, яка поєднує багаторівневу рольову модель, адаптивний інтерфейс, гнучку логіку взаємодії з базою даних, а також спрощує впровадження у командну роботу завдяки простоті використання та розширюваності.

Практичне застосування

Система може бути впроваджена в малих та середніх підприємствах, навчальних закладах і неформальних командах для планування роботи, розподілу відповідальностей, контролю за виконанням завдань і ведення статистики продуктивності.

Бакалаврська робота містить 75 сторінок, 31 рисунок, 3 розділи список використаних джерел із 41 найменуванням, 1 додаток.

					БР.ІІІ – 45.00.00.000 ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1. АНАЛІЗ ТА ДОСЛІДЖЕННЯ ЗАСТОСУВАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДО ПОБУДОВИ ПЛАНУВАЛЬНИКА ЗАВДАНЬ

1.1. Пропонована функціональність планувальника персональних завдань та управління проектами

Цей документ описує архітектуру та функціональні можливості веб-платформи, розробленої для оптимізації управління завданнями та проектами на індивідуальному рівні. Система реалізує три дискретні рівні доступу користувачів: адміністратор, менеджер проектів та співробітник, кожен з яких наділений чітко визначеним набором повноважень та функціональних можливостей для забезпечення ефективного робочого процесу.

1.1.1. Рольова модель та функціональні повноваження

Система управління завданнями базується на ролевій моделі доступу, що забезпечує диференційований контроль над функціоналом та даними:

1. Адміністратор.

Ця роль має найвищий рівень повноважень, що дозволяє здійснювати повний нагляд за системою. Функції адміністратора включають:

- Управління обліковими записами менеджерів проектів (додавання, перегляд, редагування).
- Моніторинг всіх активних та завершених проектів.
- Доступ до детальної інформації про співробітників.
- Забезпечення загального контролю та конфігурації системи.

2. Менеджер проектів.

Ця роль відповідає за оперативне управління проектами та командою.

Основні функції менеджера проектів:

- Управління обліковими записами співробітників.

					БР.ІІІ – 45.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

- Призначення завдань та делегування повноважень.
- Відстеження прогресу виконання проектів та окремих завдань.
- Взаємодія зі співробітниками щодо статусу та змісту завдань.

3. Співробітник.

Ця роль має сфокусований на завданнях функціонал. Співробітники можуть:

- Переглядати призначені їм завдання.
- Оновлювати статус виконання завдань.
- Керувати власним портфоліо призначених завдань.

1.1.2. Інтерфейс користувача та засоби безпеки

Платформа спроектована з акцентом на зручний та інтуїтивно зрозумілий інтерфейс користувача (UI), що забезпечує безперебійну взаємодію при управлінні проектами та завданнями. Це включає оптимізовані навігаційні елементи та чітке представлення інформації для всіх ролей.

Окрім того, для підвищення безпеки даних, всі користувачі системи, незалежно від їхньої ролі, мають можливість змінювати свої паролі після входу в систему. Ця функція забезпечує конфіденційність облікових даних та запобігає несанкціонованому доступу.

1.1.3. Мета та очікувані результати

Основна мета реалізації цієї веб-платформи полягає у підвищенні індивідуальної та командної продуктивності шляхом забезпечення ефективного планування, розподілу та моніторингу персональних завдань та проектів. Очікується, що надані спеціалізовані функції та ергономічний дизайн сприятимуть оптимізації робочих процесів та підвищенню ефективності виконання проектів.

					БР.ІІІ – 45.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

1.2. Обґрунтування дослідження та розробки веб-додатка для управління завданнями та проектами

1.2.1. Передумови

На тлі зростаючої складності сучасних проектів та необхідності ефективного управління ресурсами, була розроблена веб-платформа для управління завданнями та проектами. Ця система інтегрує функціонал для планування проектів, розподілу завдань та моніторингу прогресу виконання.

Архітектура системи базується на ролевій моделі доступу, що забезпечує диференційоване управління функціональними можливостями та даними. Визначено три основні ролі:

1. Адміністратор. Ця роль має повноваження для глобального управління системою, що включає контроль над обліковими записами менеджерів проектів, моніторинг усіх активних проектів та доступ до детальної інформації про співробітників.

2. Менеджер проектів. Ця роль відповідає за оперативне управління проектами, що передбачає управління обліковими записами співробітників та ефективне призначення завдань.

3. Співробітник. Ця роль орієнтована на виконання призначених завдань, забезпечуючи можливість їх перегляду та управління статусом.

З метою підвищення безпеки даних, всі користувачі системи мають можливість змінювати свої паролі після аутентифікації. Поєднання ролево-специфічного функціоналу з інтуїтивно зрозумілим інтерфейсом спрямоване на оптимізацію робочих процесів, підвищення продуктивності та забезпечення ефективного виконання проектів.

1.2.2. Значимість проекту

Розроблений веб-додаток має значну цінність для організацій та індивідуальних користувачів, які прагнуть оптимізувати управління

					БР.ІІІ – 45.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

завданнями та проектами. Його значимість визначається кількома ключовими аспектами:

1. Оптимізація управління та відстеження.

Додаток надає структуровану платформу для ефективного управління та відстеження завдань та проектів, призначаючи специфічні ролі з чітко визначеними функціями.

2. Покращений нагляд та контроль.

Адміністратор може здійснювати всебічний нагляд за всіма проектами та ефективно керувати Менеджерами Проектів, що сприяє безперебійній операційній діяльності.

3. Ефективне делегування та моніторинг.

Менеджери проектів отримують інструменти для легкого призначення завдань співробітникам та моніторингу прогресу, що є критично важливим для своєчасного та ефективного виконання проектів.

4. Збільшення відповідальності співробітників.

Співробітники можуть переглядати та керувати своїми призначеними завданнями, що забезпечує дотримання термінів та своєчасне завершення проектів.

5. Проактивне управління термінами.

Система передбачає функціонал сповіщень про майбутні терміни та важливі завдання, що мінімізує ризик пропуску дедлайнів та сприяє своєчасному завершенню проектів.

6. Розвиток професійних навичок.

Планується інтеграція навчальних модулів, що надаватимуть користувачам доступ до найкращих практик управління проектами та пріоритезації завдань. Це сприятиме підвищенню індивідуальної продуктивності, покращенню командної співпраці та ефективнішому досягненню проектних цілей.

					БР.ІІІ – 45.00.00.000 ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

Таким чином, надаючи структурований підхід до управління завданнями з ролево-специфічним функціоналом, наш додаток спрямований на підвищення продуктивності, забезпечення ефективного виконання проектів та сприяння кращій командній співпраці.

1.2.3. Мета проекту

Головна мета розробки цього веб-додатка полягає у наданні користувачам комплексного інструменту для ефективного управління та відстеження своїх завдань та проектів. Додаток спроектований для сприяння у:

- Плануванні проектів - надання інструментів для структурованого планування проектних ініціатив.
- Призначенні завдань - забезпечення функціоналу для ефективного розподілу завдань серед членів команди.
- Моніторингу прогресу - надання механізмів для відстеження поточного стану виконання завдань та проектів.

Інтеграція диференційованих ролей для адміністраторів, менеджерів проектів та співробітників забезпечує безперебійний робочий процес та значне підвищення продуктивності.

Зокрема:

- Адміністратори здійснюють загальний нагляд за проектами.
- Менеджери проектів ефективно керують завданнями та ресурсами проекту.
- Співробітники відповідально виконують призначені завдання.

Шляхом забезпечення структурованого підходу до управління завданнями, наш додаток прагне підвищити продуктивність та забезпечити ефективне виконання проектів та планування персональних завдань співробітників та членів команд.

					БР.ІІІ – 45.00.00.000 ПЗ	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

1.3. Аналіз існуючих платформ планувальника персональних завдань та управління проектами

Існує широкий спектр існуючих платформ для планування персональних завдань та управління проектами, які відрізняються функціоналом, складністю, цільовою аудиторією та методологіями, які вони підтримують.

1.3.1. Платформа для управління робочими процесами Monday.com

Monday.com — це гнучка та візуально орієнтована платформа для управління робочими процесами та проектами, що позиціонується як "Work OS" (операційна система для роботи). Вона розроблена для команд будь-якого розміру та різних галузей, дозволяючи їм організовувати, планувати та відстежувати свою роботу в одному централізованому місці.

Дошки є основою Monday.com. Вони являють собою настроювані візуальні робочі простори, де команди можуть організовувати свої завдання, проекти, клієнтів, процеси тощо. Користувачі можуть додавати різні типи стовпців до своїх дошок (статус, дати, числа, текст, файли, люди, формули, відстеження часу тощо), що дозволяє їм адаптувати дошку до будь-якого робочого процесу. Завдання можна групувати за різними критеріями (наприклад, за фазами проекту, за відповідальними, за пріоритетом), що покращує візуальну організацію.

Monday.com пропонує різноманітні види відображення даних, що дозволяє користувачам переглядати інформацію у форматі, який найкраще відповідає їхнім потребам:

- Стандартний вигляд, схожий на електронну таблицю.
- Для візуалізації робочих процесів та відстеження прогресу за допомогою карток (канбан-дошка).

					БР.ІП – 45.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

- Використання календаря для планування та перегляду завдань у часовому форматі.
- Застосування діаграм Ганта для візуалізації термінів проекту, залежностей між завданнями та загального прогресу.
- Настроювані панелі, що дозволяють отримувати високорівневі огляди проектів, показників ефективності та ключових даних з різних дошок.

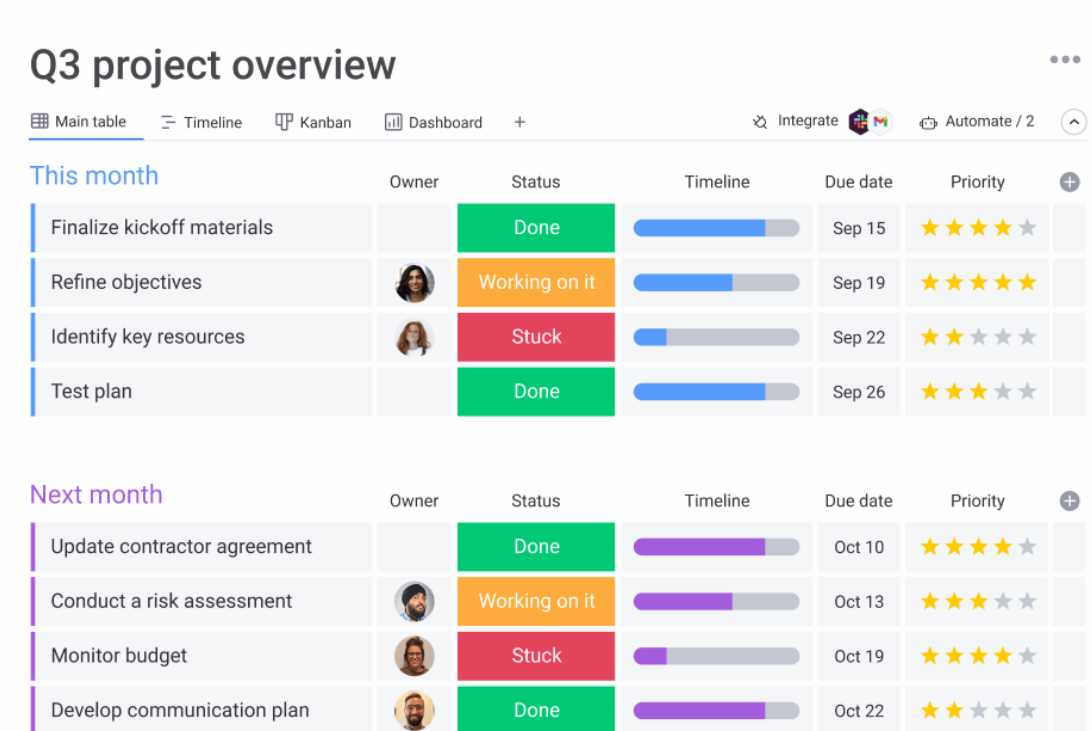


Рисунок 1.1 - Платформа для управління робочими процесами Monday.com

Monday.com дозволяє створювати автоматизовані робочі процеси без написання коду. Це допомагає економити час та мінімізувати ручну роботу.

Приклади автоматизації:

- Автоматична зміна статусу завдання при його переміщенні на дошці.
- Надсилання сповіщень команді при настанні певних подій (наприклад, дедлайн, зміна статусу).
- Призначення завдань на основі певних умов.

Платформа підтримує інтеграцію з великою кількістю популярних інструментів, таких як:

- Комунікаційні платформи (Slack, Microsoft Teams).
- Інструменти для керування файлами (Google Drive, Dropbox, OneDrive).
- CRM-системи (Salesforce, HubSpot).
- Інструменти розробки (GitHub, Jira).
- Електронна пошта (Gmail, Outlook).

В системі є можливість коментувати завдання, додавати файли, згадувати колег, що сприяє ефективній комунікації в контексті роботи. Також присутня можливість створювати настроювані звіти та панелі інструментів для аналізу даних та відстеження ключових показників ефективності (KPIs).

Monday.com може використовуватися для широкого спектру завдань та проектів, включаючи:

- Управління проектами та завданнями
- Планування маркетингових кампаній
- Управління клієнтами та продажами (як CRM)
- Відстеження розробки продуктів
- Управління HR-процесами (набір, адаптація співробітників)
- Організація IT-операцій та служби підтримки
- Керування подіями
- Створення контент-календарів
- Будь-який інший робочий процес, який потребує організації та візуалізації

Переваги Monday.com:

1. Дозволяє адаптувати платформу під майже будь-який робочий процес.
2. Інтуїтивно зрозумілий та візуально привабливий інтерфейс, легкий у навчанні та використанні.

					БР.ІІІ – 45.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		21

3. Широкі можливості автоматизації, дозволяє підключити Monday.com до існуючих інструментів.

Недоліки Monday.com

1. Може бути дорогим для великих команд, особливо з урахуванням мінімальної кількості користувачів у планах. Цінова політика може здаватися складною.

2. Для дуже простих сценаріїв використання може бути надмірним.

3. Хоча базовий інтерфейс інтуїтивно зрозумілий, освоєння всіх можливостей автоматизації та кастомізації може вимагати часу.

Загалом, Monday.com є потужним та універсальним інструментом для управління робочими процесами, який допомагає командам покращити співпрацю, підвищити прозорість та ефективність у виконанні завдань та проектів.

1.3.2. Інструмент для управління завданнями та проектами MeisterTask

MeisterTask — це візуально орієнтований онлайн-інструмент для управління завданнями та проектами, який широко використовує методологію Канбан. Він позиціонується як простий, але потужний інструмент, що допомагає командам та окремим користувачам організувати свою роботу, відстежувати прогрес та ефективно співпрацювати. MeisterTask розроблений компанією MeisterLabs, яка також відома своїм інструментом для створення ментальних карт MindMeister, і ці два продукти тісно інтегровані.

Основний інтерфейс MeisterTask побудований на Канбан-дошках, де завдання (картки) переміщуються між колонками (секціями), що відображають різні етапи робочого процесу (наприклад, "Що робити", "В процесі", "Завершено", "На перевірці"). Колонки дошки можуть бути

					БР.ІІІ – 45.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		22

налаштовані відповідно до унікальних потреб будь-якого проекту чи команди. Користувачі можуть додавати та перейменовувати секції.

Кожна картка завдання є міні-центром інформації. Вона може містити:

- Детальний опис завдання.
- Контрольні списки (чек-листи) для розбиття завдання на менші кроки.
- Вкладення файлів (документи, зображення, медіа).
- Терміни виконання.
- Призначених виконавців.
- Мітки (теги) для категоризації та фільтрації.
- Коментарі та обговорення.

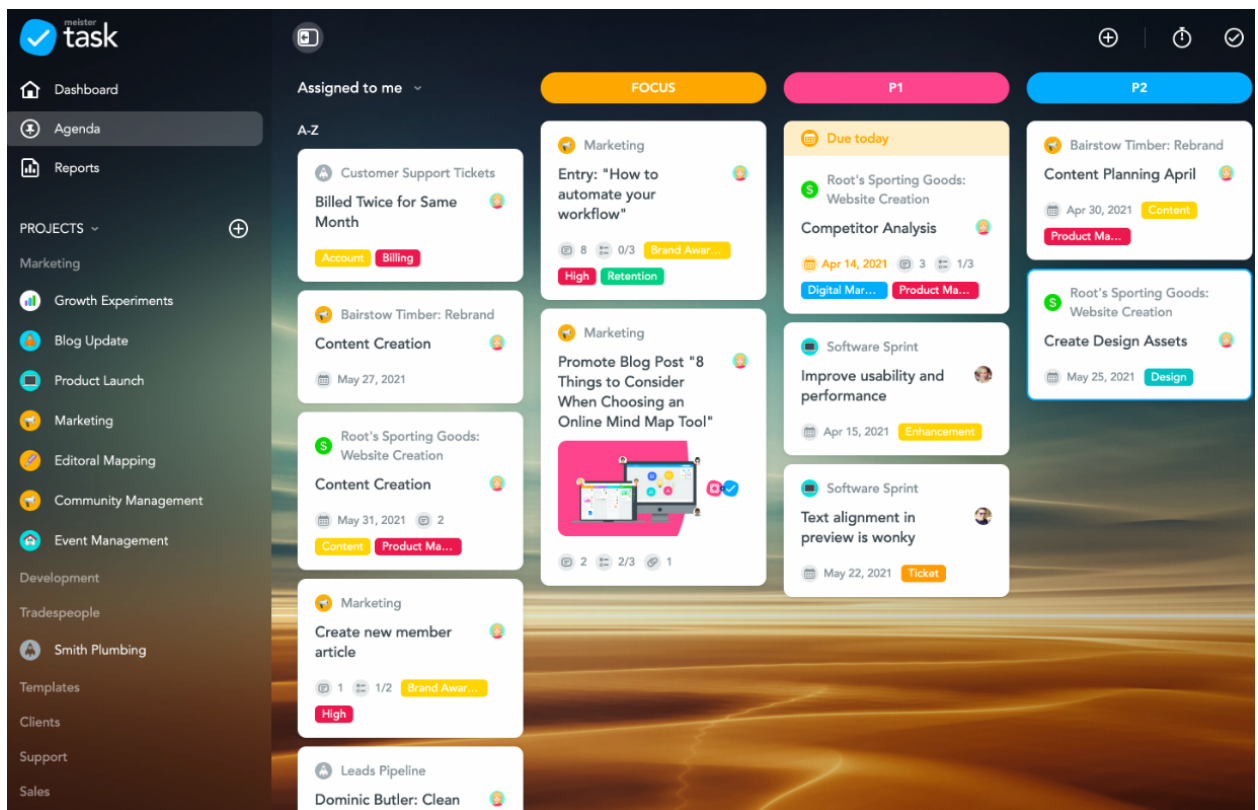


Рисунок 1.2 - Інструмент для управління завданнями та проектами MeisterTask

MeisterTask дозволяє налаштовувати автоматичні дії, які спрацьовують, коли завдання переміщується в певну секцію. Це дозволяє оптимізувати рутинні операції. Приклади автоматизації:

					БР.ІІІ – 45.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

- Автоматичне призначення завдання певній особі.
- Зміна статусу завдання.
- Додавання заздальгідь визначеного коментаря.
- Перенесення завдання до архіву або іншого проекту.
- Автоматичний запуск або зупинка таймера відстеження часу.

Система сповіщень інформує користувачів про важливі події (зміни в завданнях, нові коментарі, наближення термінів). Вбудований таймер дозволяє користувачам легко відстежувати час, витрачений на кожне завдання. Це корисно для обліку робочого часу, виставлення рахунків клієнтам та аналізу продуктивності.

MeisterTask надає огляди та звіти про прогрес проектів, завантаженість команди та індивідуальну продуктивність. Це допомагає менеджерам отримувати уявлення про стан проектів та ефективність команди. Персональна панель завдань, яка збирає всі завдання користувача з різних проектів в одному місці, дозволяючи йому краще організувати свою роботу.

MeisterTask інтегрується з безліччю сторонніх сервісів, покращуючи робочий процес:

- Slack, Microsoft Teams: Для комунікації та сповіщень.
- Google Drive, Dropbox, OneDrive: Для вкладень файлів.
- Google Calendar, Outlook Calendar: Для синхронізації термінів виконання.
- GitHub, Bitbucket, Jira: Для команд розробників.
- Zapier: Для зв'язку з тисячами інших веб-додатків.

Функція MeisterNote - інтегрований інструмент для створення та управління нотатками та документацією проекту, що дозволяє зберігати всю необхідну інформацію в одному місці.

MeisterTask розміщує дані в дата-центрах ЄС, сертифікованих за ISO 27001, і відповідає вимогам GDPR, що важливо для захисту даних.

					БР.ІІІ – 45.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		24

MeisterTask пропонує безкоштовний (Basic) план з обмеженим функціоналом (наприклад, до 3 проектів). Платні плани (Pro, Business, Enterprise) надають розширені можливості, включаючи необмежену кількість проектів, розширену автоматизацію, детальні звіти, розширені інтеграції, ролі та дозволи, підтримку пріоритетних запитів та інші корпоративні функції.

MeisterTask підходить:

1. Малим та середнім командам, завдяки простоті та візуалізації, він ідеальний для невеликих груп, яким потрібен чіткий огляд роботи.

2. Командам, що використовують Канбан, для Agile-методологій та візуального відстеження прогресу.

3. Для особистого планування та управління завданнями.

MeisterTask є привабливим вибором для тих, хто шукає інтуїтивно зрозумілий, візуально приємний та ефективний інструмент для управління завданнями, особливо якщо візуалізація прогресу та простота використання є пріоритетом.

1.3.3. Платформа для управління робочими процесами та проектами Wrike

Wrike — це потужна та масштабована платформа для управління робочими процесами та проектами, розроблена для команд будь-якого розміру, від малих підприємств до великих корпорацій. Вона надає комплексний набір інструментів, що дозволяють організувати роботу, співпрацювати, відстежувати прогрес та оптимізувати робочі процеси. Wrike позиціонується як універсальне рішення для управління всіма аспектами роботи.

Wrike дозволяє створювати проекти, папки та завдання, організовуючи всю роботу в ієрархічній структурі. Користувачі можуть переглядати свої завдання та проекти в різних форматах, що відповідають їхнім потребам.

					БР.ІІІ – 45.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

Є можливість створювати власні робочі процеси та статуси завдань, що дозволяє Wrike адаптуватися до унікальних потреб будь-якої команди чи індустрії. Команди можуть спільно працювати над документами та завданнями, залишати коментарі, згадувати колег (@mentions). Функція для спрощення процесу перегляду та затвердження файлів, документів та візуальних матеріалів з можливістю додавання коментарів безпосередньо до вмісту. Це особливо корисно для маркетингових та креативних команд. Присутня можливість прикріплювати файли до завдань та проектів, що забезпечує централізоване зберігання та доступ до інформації.

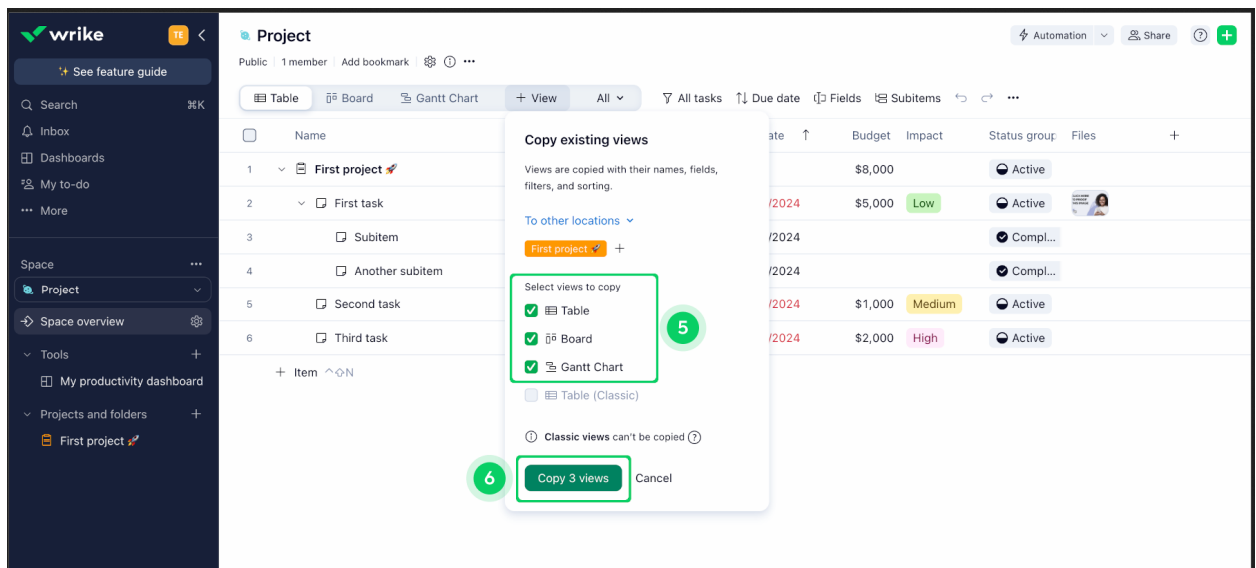


Рисунок 1.3 - Платформа для управління робочими процесами та проектами Wrike

Wrike дозволяє автоматизувати рутинні дії та процеси (наприклад, автоматичне призначення завдань, зміна статусів, надсилання сповіщень) на основі заданих тригерів та умов, що значно економить час та знижує кількість помилок. Також є можливість створювати настроювані форми запитів, які автоматично перетворюються на завдання або проекти з усією необхідною інформацією, забезпечуючи стандартизований потік вхідних запитів.

Настроювані панелі інструментів (Dashboards) дозволяють користувачам та менеджерам отримувати швидкий огляд ключових показників, прогресу проектів та завантаженості команди. Wrike надає гнучкі інструменти для створення детальних звітів про прогрес проекту, використання ресурсів, час виконання завдань та інші важливі показники. Наявність AI-функцій - інтеграція штучного інтелекту для генерації звітів, створення описів та інших автоматизованих функцій, що прискорюють процеси.

Присутня можливість налаштовувати права доступу для різних користувачів та груп на рівні папок, проектів та завдань. Високий рівень безпеки даних, включаючи шифрування, двофакторну автентифікацію та відповідність міжнародним стандартам (GDPR, SOC 2, HIPAA).

Завдяки своїй масштабованості та широкому функціоналу, Wrike є відмінним вибором для великих організацій з комплексними проектами та різними відділами (маркетинг, IT, розробка продуктів, креативні команди). Wrike дозволяє адаптувати робочі процеси та інтерфейс під дуже специфічні вимоги. Функції перевірки та спільної роботи роблять його ідеальним для команд, які працюють з контентом та дизайном.

Командам, що потребують детальної звітності та аналітики: Потужні інструменти звітів допомагають керівництву приймати обґрунтовані рішення.

Недоліки Wrike:

1. Через велику кількість функцій, Wrike може здатися складним для нових користувачів, особливо для невеликих команд.
2. Може бути дорогим, особливо для малих команд, оскільки часто продається пакетами користувачів (наприклад, по 5, 10, 25).
3. Деякі користувачі відзначають, що платформа може сповільнюватися при роботі з дуже великими проектами або великою кількістю даних.

Загалом, Wrike є одним з лідерів у галузі програмного забезпечення для управління проектами, що надає гнучке, масштабоване та

					БР.ІІІ – 45.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		27

багатофункціональне рішення для підвищення продуктивності та оптимізації робочих процесів у складних корпоративних середовищах.

1.3.4. Веб-платформа Notion

Notion — це унікальна та надзвичайно гнучка веб-платформа, яка позиціонується як "все-в-одному робочий простір" або "операційна система для вашого життя та роботи". Вона поєднує в собі функції нотатника, бази даних, інструменту для управління проектами, створення вікі-сторінок та багато іншого, дозволяючи користувачам створювати власні кастомізовані робочі середовища без написання коду.

Основна концепція Notion полягає в тому, що все є блоком, а блоки можуть бути організовані в сторінки, які, в свою чергу, можуть бути вкладені в інші сторінки. Це створює безмежні можливості для ієрархічної та нелінійної організації інформації.

Система побудована як універсальний робочий простір на основі блоків . Це фундаментальні одиниці контенту в Notion. Вони можуть бути текстовими параграфами, заголовками, списками (маркованими, нумерованими, списками перевірки), зображеннями, відео, аудіо, кодом, цитатами, таблицями, кнопками, вбудованими файлами та багато чим іншим. Блоки об'єднуються в сторінки. Сторінки можуть містити інші сторінки, що дозволяє створювати складні ієрархічні структури, такі як вікі, бази знань або цілі відділи компанії.

Одна з найпотужніших функцій Notion — це його система баз даних. Вони не схожі на традиційні електронні таблиці. Кожен рядок у базі даних є окремою сторінкою Notion, яку можна відкрити та додати будь-який вміст (текст, зображення, інші блоки). Бази даних мають настроювані властивості (стовпці), які можуть бути різних типів: текст, числа, дати, люди (для призначення завдань), URL, чекбокси, мультиселекти, формули, відстеження часу тощо.

					БР.ІІІ – 45.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		28

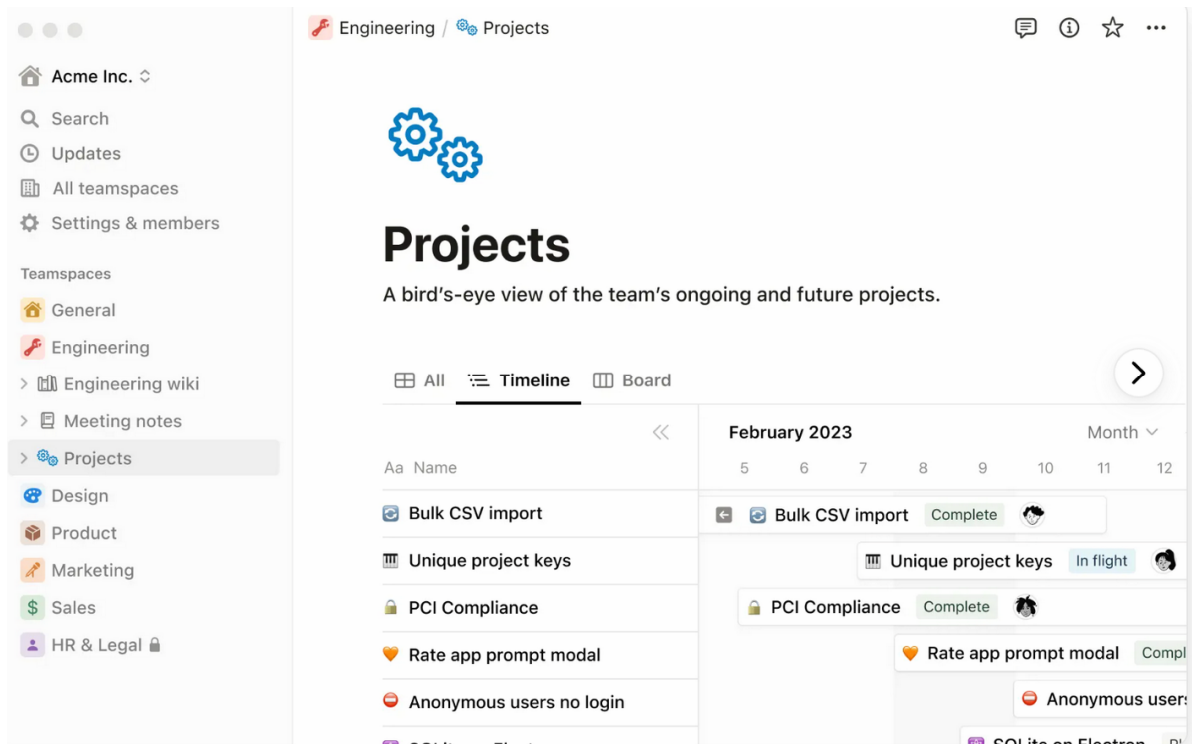


Рисунок 1.4 - Веб-платформа Notion

Користувачі можуть ділитися будь-якою сторінкою або базою даних з іншими учасниками команди, зовнішніми гостями або навіть публікувати їх в Інтернеті як веб-сайти. Присутня можливість додавати коментарі до будь-якого блоку або сторінки, а також згадувати колег для привернення їхньої уваги. Notion зберігає історію змін, дозволяючи відновлювати попередні версії сторінок.

Notion має величезну бібліотеку готових шаблонів для різних сценаріїв використання (управління проектами, особисте планування, ведення нотаток, CRM, HR, освітні потреби тощо). Користувачі також можуть створювати власні шаблони.

Інтегрований штучний інтелект, який може виконувати безліч завдань: писати тексти, резюмувати документи, перекладати, генерувати ідеї, відповідати на запитання на основі вмісту вашого робочого простору, а також інтегрованих інструментів (Slack, Google Drive, GitHub, Jira та ін.). Notion пропонує власні інтеграції (Slack, Google Drive, GitHub, Jira) та API

					БР.ІІ – 45.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		29

для створення користувацьких інтеграцій, що дозволяє підключати його до інших сервісів.

Основні сценарії використання Notion:

1. Від простих списків завдань до складних систем управління проектами з Канбан-дошками, діаграмами Ганта (через Chronology view) та відстеженням прогресу.

2. Гнучкий інструмент для створення нотаток, що дозволяє організувати їх за будь-якою структурою, додавати медіа та посилання.

3. Ідеально підходить для створення корпоративних вікі, особистих баз знань або бібліотек інформації завдяки можливості глибокої ієрархії та посилань між сторінками.

4. Планування дня, відстеження звичок, ведення журналів, управління фінансами, планування подорожей тощо.

5. Календарі контенту, бази даних для блогів, сценаріїв, ідей.

Переваги Notion:

- Надзвичайна гнучкість, можна адаптувати під майже будь-яку потребу та робочий процес.

- Універсальність, замінює кілька окремих інструментів (нотатник, таск-менеджер, база даних, вікі).

- Візуальна привабливість та інтуїтивність, чистий та мінімалістичний інтерфейс, легкий у навігації.

- Потужні бази даних з різними видами відображення.

- Спільна робота в реальному часі.

- Велика спільнота та багато шаблонів.

Недоліки Notion:

- Хоча базові функції прості, освоєння всіх можливостей та створення складних систем може зайняти значний час.

- Відсутність деяких специфічних функцій. Оскільки це універсальний інструмент, йому може не вистачати глибоких спеціалізованих функцій, які є

					БР.ІІІ – 45.00.00.000 ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

в окремих рішеннях (наприклад, для складного управління бюджетом проекту чи дуже детальної звітності).

- Можливості офлайн-роботи обмежені в порівнянні з настільними додатками.

Notion є хорошим вибором для тих, хто шукає універсальний інструмент, який можна налаштувати під свої унікальні потреби, і готовий витратити час на його освоєння. Він дозволяє об'єднати всю інформацію та робочі процеси в одному місці, значно підвищуючи організацію та продуктивність.

					БР.ІІІ – 45.00.00.000 ПЗ	Арк.
						31
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2. АНАЛІЗ ВИМОГ ТА РЕАЛІЗАЦІЯ АРХІТЕКТУРИ ПЛАНУВАЛЬНИКА ПЕРСОНАЛЬНИХ ЗАВДАНЬ ТА УПРАВЛІННЯ ПРОЕКТАМИ

2.1. Збір та аналіз вимог до веб-платформи управління завданнями

2.1.1. Функціональні вимоги

Розроблена веб-платформа для управління завданнями та проектами базується на трирівневій ролевій моделі, що визначає диференційований доступ до функціоналу.

Система передбачає три основні ролі з чітко визначеними повноваженнями:

1. Адміністратор (Administrator)
2. Менеджер проекту (Project Manager)
3. Співробітник (Employee)

2.1.2. Функціональність за ролями

Адміністратор:

- Створення та управління обліковими записами менеджерів проектів.
- Створення та управління проектами у відповідності до корпоративних потреб.
- Перегляд детальної інформації про всіх співробітників.

Менеджер проекту:

- Додавання та управління обліковими записами співробітників.
- Перегляд проектів, ініційованих адміністраторами.
- Призначення проектів співробітникам.

Співробітник:

- Управління проектами, призначеними безпосередньо співробітнику.

					БР.ІІІ – 45.00.00.000 ПЗ	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

- Надання оновлень щодо статусу та прогресу виконання призначених проектів.

2.1.3. Загальна функціональність

Забезпечення безперешкодного додавання та модифікації інформації для всіх ролей користувачів, гарантуючи інтуїтивність взаємодії користувачів між собою.

Реалізація можливості оновлення облікових даних (паролів) для всіх ролей користувачів після успішної автентифікації, що підвищує безпеку системи.

2.1.4. Нефункціональні вимоги

Для забезпечення ефективності та надійності функціонування системи, визначені наступні нефункціональні вимоги:

1. Масштабованість.

Система повинна демонструвати здатність до ефективної обробки великої кількості одночасних користувацьких сесій. Зокрема, архітектура повинна бути оптимізована для обслуговування приблизно однієї тисячі одночасних користувачів без суттєвого зниження продуктивності.

2. Зручність використання (Usability).

Інтерфейс користувача має бути максимально інтуїтивно зрозумілим та доступним навіть для користувачів без попереднього досвіду роботи з подібними системами, мінімізуючи потребу в додатковому навчанні користувачів.

3. Швидкодія (Performance).

Система повинна забезпечувати високу швидкість відгуку на дії користувача. Час відповіді на типові операції має бути мінімальним, забезпечуючи плавну та ефективну взаємодію.

					БР.ІІІ – 45.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		33

2.2. Технічні вимоги до середовища розробки та експлуатації

Для успішної реалізації та подальшого функціонування веб-платформи визначено наступні апаратні та програмні вимоги.

2.2.1. Апаратні вимоги

Операційна Система:

- Windows 10/8 (64-біт)
- macOS 10.10 (Yosemite) або новіша версія
- Дистрибутиви Linux (наприклад, Ubuntu)

Оперативна Пам'ять (RAM): Мінімум 8 ГБ.

Дисковий Простір: Мінімум 4 ГБ вільного місця.

Процесор: Еквівалент Intel i5 або вище, з підтримкою технологій Intel VT-x, Intel EM64T (Intel 64) та функції Execute Disable (XD) Bit.

Роздільна Здатність Екрану: Мінімальна роздільна здатність 1280x800 пікселів.

2.2.2. Програмні вимоги

Середовища Розробки (IDE):

- Visual Studio Code
- IntelliJ IDEA

Основні технології:

- Фронтенд: React, JavaScript, HTML, CSS, Bootstrap.
- Бекенд: Spring Boot.

База Даних: MySQL.

Інструменти:

- MySQL Workbench (для управління базою даних)
- Maven (для управління залежностями проекту)
- NPM (Node Package Manager, для управління пакетами JavaScript)

					БР.ІІІ – 45.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		34

- GitHub (для контролю версій)
- Nodemon (для автоматичного перезапуску сервера під час розробки)

Сервер додатків: Apache Tomcat.

2.3. Інструментарій та технології реалізації

Цей підрозділ описує ключові інструменти та технології, використані в процесі розробки веб-платформи для управління завданнями, обґрунтовуючи їхній вибір та роль у архітектурі системи.

2.3.1. Spring Boot

Spring Boot є провідним фреймворком з відкритим вихідним кодом для розробки додатків на платформі Java. Його використання значно спрощує процес створення автономних, готових до виробництва додатків завдяки реалізації принципу автоконфігурації, вбудованим серверам (наприклад, Apache Tomcat) та ефективному управлінню залежностями. Фреймворк підтримує архітектуру мікросервісів, що сприяє модульній та масштабованій розробці. Додатково, Spring Boot інтегрує такі інструменти, як Spring Initializr для швидкої ініціалізації проектів та Spring Boot Actuator для моніторингу та управління додатками.

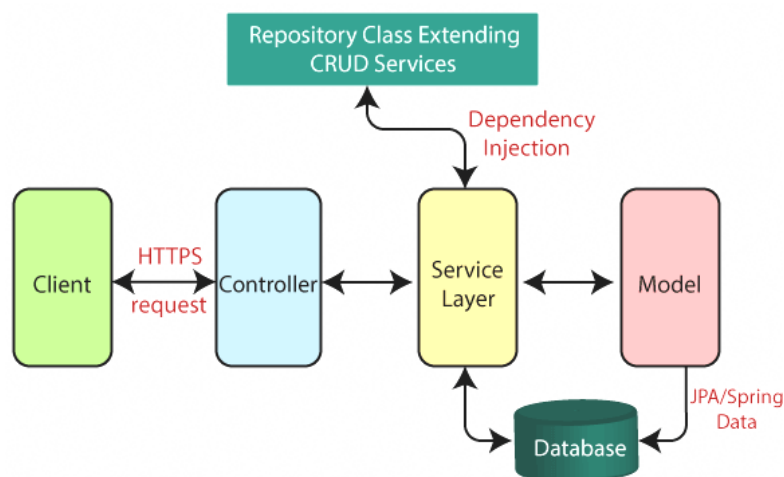


Рисунок 2.1 - Архітектура веб-додатку на базі Spring Boot

					БР.ІІІ – 45.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		35

Потік виконання запиту (рис. 2.1):

1. Client надсилає HTTP-запит (наприклад, GET, POST) до сервера.
2. Controller приймає цей запит.
3. Controller викликає відповідний метод у Service Layer, передаючи необхідні дані.
4. Service Layer виконує бізнес-логіку, яка може включати валідацію, обчислення та, за необхідності, взаємодію з даними.
5. Service Layer через механізм Dependency Injection викликає методи Repository Class.
6. Repository Class, використовуючи JPA/Spring Data, взаємодіє з Database для виконання операцій з даними (отримання, збереження, оновлення, видалення).
7. Дані з Database відображаються в Model об'єкти через JPA/Spring Data.
8. Результат операції повертається від Repository Class до Service Layer.
9. Service Layer повертає оброблений результат до Controller.
10. Controller формує HTTP-відповідь (наприклад, JSON, XML) і надсилає її назад Client.

Ця архітектура сприяє високій керованості, тестуванню (кожен шар можна тестувати ізольовано) та еволюції системи, оскільки зміни в одному шарі мінімально впливають на інші.

Для управління залежностями проекту використовувався Maven. Конфігурація залежностей здійснювалася через файл pom.xml, що забезпечувало автоматичне завантаження та управління необхідними бібліотеками. Зокрема, для встановлення з'єднання з базою даних MySQL, до pom.xml була додана залежність MySQL Connector. Параметри з'єднання, такі як JDBC URL, ім'я користувача та пароль, конфігурувалися у файлі application.properties. Таке налаштування забезпечило надійну та ефективну

					БР.ІІІ – 45.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		36

взаємодію між бекенд-додатком на Spring Boot та реляційною базою даних MySQL.

2.3.2. React

React — це декларативна, компонентно-орієнтована бібліотека JavaScript, призначена для розробки інтерфейсів користувача (UI). Вона дозволяє створювати складні UI шляхом композиції невеликих, незалежних та повторно використовуваних компонентів (наприклад, кнопки, текстові поля, графічні елементи). В рамках даного проекту використовувалася React версії 18.2.0. Ініціалізація розробки фронтенду виконувалася за допомогою npm (Node Package Executor), а управління необхідними бібліотеками здійснювалося за допомогою npm (Node Package Manager).

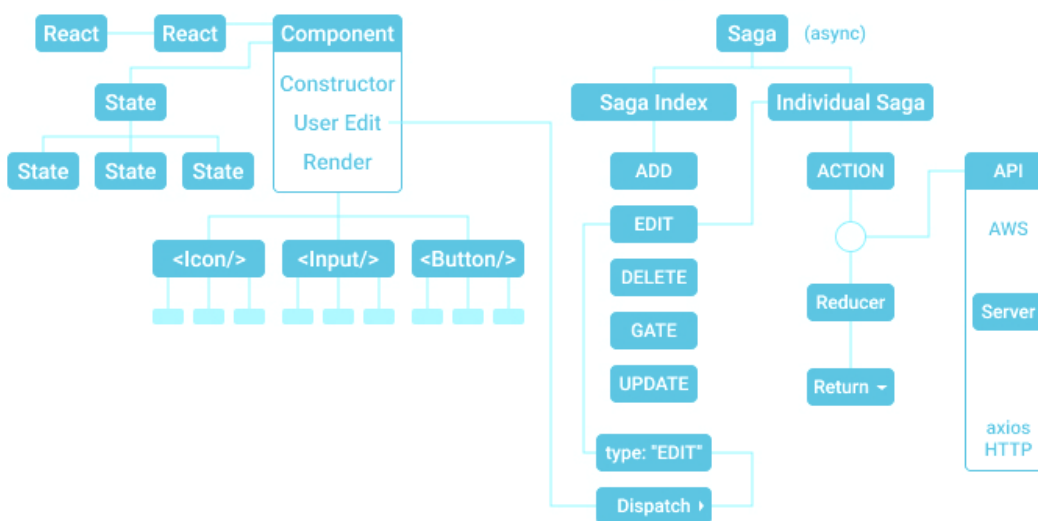


Рисунок 2.2 - Робочий процес (workflow) веб-додатків, розроблених з використанням React

Ліва частина (рис. 2.2) - компонентна архітектура React та управління станом. Ця частина схеми показує, як React будує інтерфейс користувача з компонентів і як вони взаємодіють зі станом. Права частина - управління асинхронними операціями та станом за допомогою Redux Saga. Ця частина

схеми демонструє архітектуру для управління складним станом та асинхронними сайд-ефектами, що є типовим для великих React-додатків, які використовують Redux для управління станом та Redux Saga для асинхронності.

Взаємозв'язок між частинами:

- Користувач взаємодіє з компонентами React на UI.
- Ці взаємодії викликають відправлення (dispatch) дій.
- Саги перехоплюють ці дії, виконують асинхронні операції (наприклад, запити до API/сервера через axios HTTP).
- Після завершення асинхронних операцій, саги відправляють нові дії.
- Ці дії обробляються редюсерами, які оновлюють стан додатка.
- Зміна стану призводить до перерендеру компонентів React, оновлюючи UI для користувача.

Ця архітектура дозволяє відокремити бізнес-логіку (особливо асинхронну) від компонентів UI, роблячи додаток більш структурованим, легшим для тестування та масштабованим.

Розробка фронтенду реалізована як односторінковий веб-додаток (SPA), що базується на компонентній архітектурі. React використовує концепцію віртуального DOM (Document Object Model), що забезпечує високоефективний рендеринг UI шляхом мінімізації прямих маніпуляцій з реальним DOM. Для забезпечення безперебійної та інтуїтивно зрозумілої навігації між різними розділами додатку була інтегрована бібліотека react-router-dom.

2.3.3. MySQL

MySQL є високопродуктивною, багатопотоковою, багато-користувацькою системою управління реляційними базами даних (СУБД), що використовує SQL (Structured Query Language). Вона спроектована для інтеграції в широко розповсюджене програмне забезпечення та ефективного

									Арк.
									38
Змн.	Арк.	№ докум.	Підпис	Дата	БР.ІІІ – 45.00.00.000 ПЗ				

функціонування в умовах високих навантажень виробничих систем. В даному проєкті MySQL обрано як основну СУБД завдяки її надійності та здатності забезпечувати цілісність даних. MySQL повністю відповідає властивостям ACID (Атомарність, Узгодженість, Ізольованість та Стійкість), що гарантує цілісність та консистентність даних навіть у випадку системних збоїв.

2.3.4. MySQL Workbench

MySQL Workbench — це уніфікований візуальний інструмент, що надає комплексний набір функцій для проєктування, розробки та адміністрування баз даних MySQL. Він підтримує ключові операції, включаючи візуальне проєктування баз даних (Entity-Relationship Diagrams), розробку SQL-запитів та конфігурацію серверів MySQL.

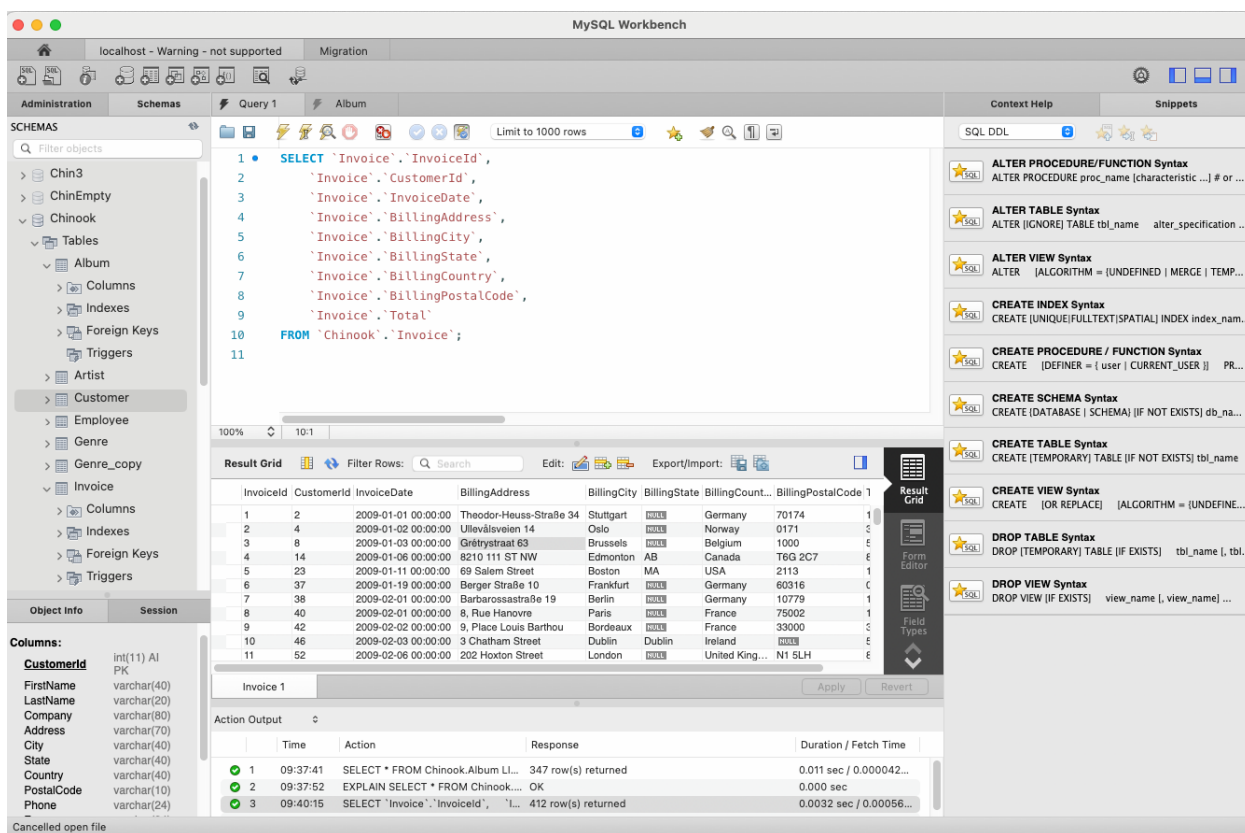


Рисунок 2.3 – Інструмент MySQL Workbench

Завдяки його графічному інтерфейсу, будь-який SQL-запит може бути легко виконаний, а управління базами даних, таблицями, стовпцями, зв'язками, індексами, користувачами та дозволами значно спрощується. Використання MySQL Workbench дозволило ефективно проектувати та адмініструвати схему бази даних для додатку управління завданнями, а також візуалізувати дані та їхні взаємозв'язки.

2.3.5. Maven

Maven є потужним інструментом автоматизації збірки, переважно використовуваним для проектів на Java. Він відіграє ключову роль у спрощенні управління життєвим циклом проекту шляхом автоматизації таких процесів, як управління залежностями, компіляція вихідного коду, виконання автоматизованих тестів та пакування додатків. Maven використовує Project Object Model (POM), який описується у файлі pom.xml, для управління конфігураціями проекту та всіма його залежностями. Це забезпечує високу узгодженість процесів збірки в різних середовищах розробки та розгортання.

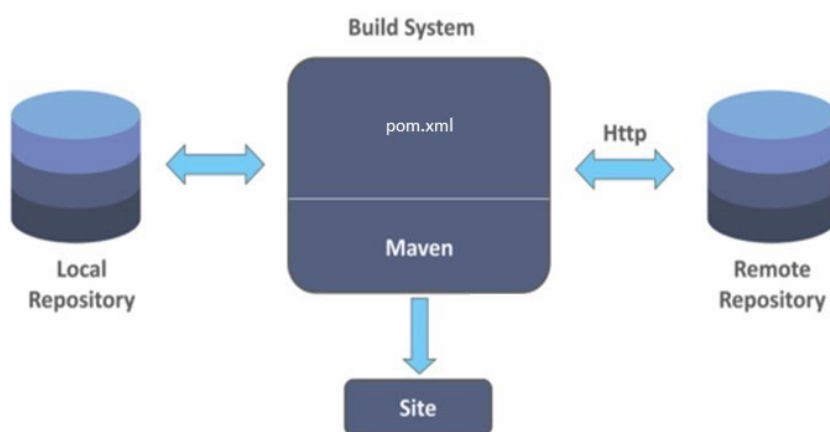


Рисунок 2.4 - Архітектура Maven як система збірки (Build System) та її взаємодія з репозиторіями

Короткий робочий процес (рис. 2.4):

					БР.ІІІ – 45.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		40

1. Розробник запускає команду Maven для збірки проекту.
 2. Maven читає файл pom.xml, щоб зрозуміти структуру проекту та його залежності.
 3. Для кожної залежності Maven спочатку шукає її у Local Repository.
 4. Якщо залежність не знайдена локально, Maven завантажує її з Remote Repository (через HTTP) і зберігає копію в локальному репозиторії.
 5. Після вирішення всіх залежностей Maven виконує інші кроки збірки (компіляція, тестування, пакування) відповідно до конфігурації в pom.xml.
 6. За бажанням, Maven також може генерувати Site проекту.
- Ця архітектура дозволяє Maven ефективно керувати залежностями, забезпечувати відтворювані збірки та спрощувати управління проектами на Java

2.3.6. Bootstrap

Bootstrap — це відкритий, популярний CSS-фреймворк, що забезпечує швидку розробку адаптивних та мобільно-орієнтованих веб-додатків. Він надає розробникам обширну колекцію готових до використання компонентів CSS та JavaScript, включаючи адаптивні сітки, навігаційні панелі, модальні вікна, кнопки та типографічні стилі.

Переваги:

- Швидкий старт розробки
- Велика спільнота та документація
- Підтримка сучасних браузерів
- Сумісність з багатьма іншими інструментами

На момент 2025 року Bootstrap 5 — це основна версія, яка:

- не потребує jQuery,
- краще підтримує CSS-змінні,
- має нові компоненти та оновлений дизайн.

					БР.ІІІ – 45.00.00.000 ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

Використання Bootstrap дозволяє значно прискорити процес проектування та налаштування інтерфейсів, одночасно гарантуючи узгоджений стиль та функціональність на різних веб-браузерах та мобільних пристроях.

2.4. Представлення діаграми робочих процесів

Діаграма (рис. 2.5) є діаграмою робочих процесів (Workflow Diagram) або діаграмою потоку даних (Data Flow Diagram), що ілюструє функціональні можливості та взаємодію між різними ролями користувачів у системі управління завданнями (Task Management System).

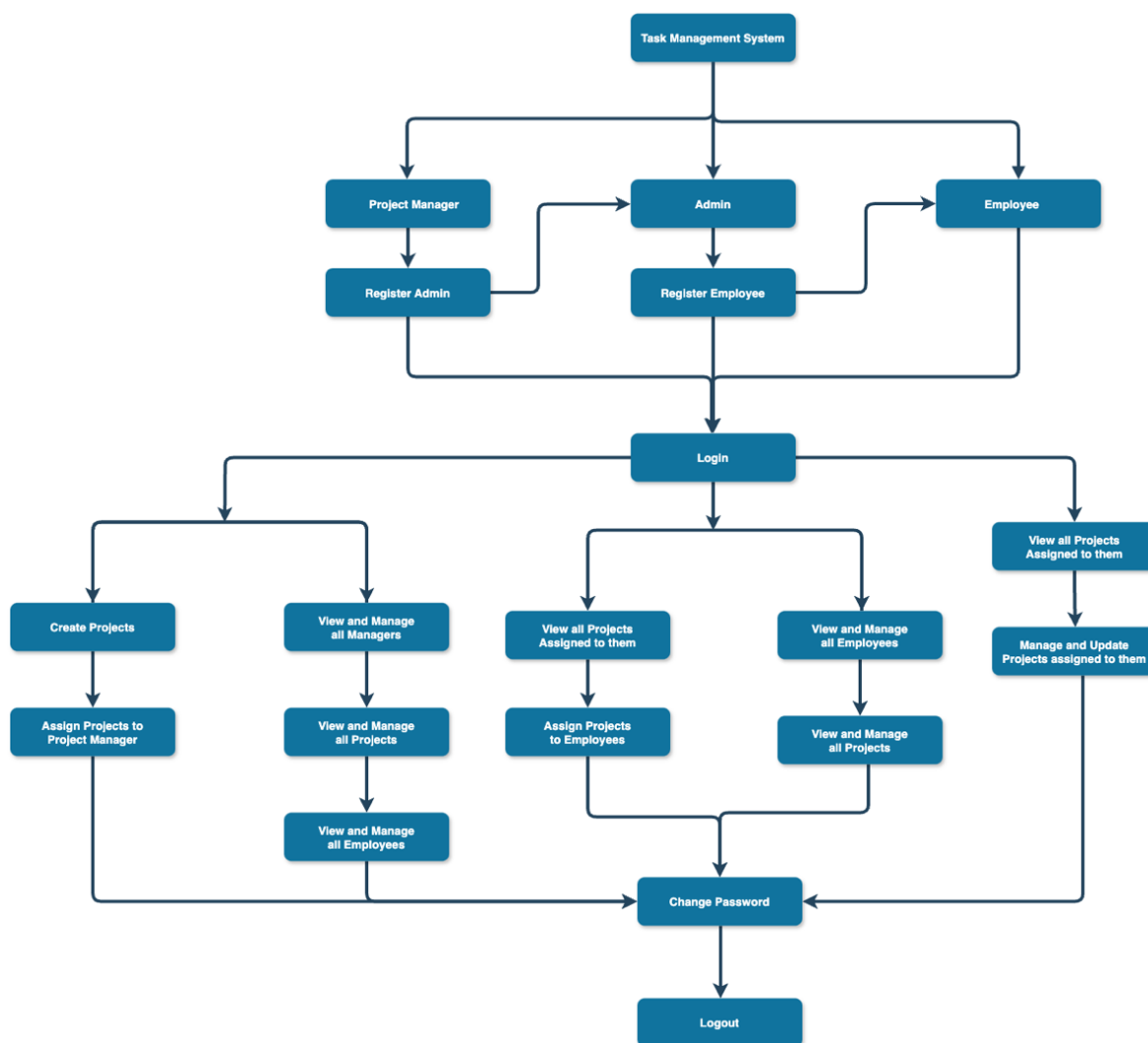


Рисунок 2.5 - Діаграма активності

Давайте розберемо діаграму по частинах, описуючи кожну роль та їхні повноваження:

1. Головний вузол: Task Management System - це корінь системи, з якого виходять всі основні ролі користувачів.

2. Основні ролі користувачів:

Від головного вузла відходять три основні ролі:

- Project Manager (Менеджер проекту)

- Admin (Адміністратор)

- Employee (Співробітник)

3. Процес реєстрації (перед входом в систему):

- Project Manager може перейти до Register Admin. Це означає, що роль менеджера проекту є вихідною точкою для ініціалізації системи і менеджер проекту має повноваження створювати облікові записи адміністраторів.

- Admin може перейти до Register Employee. Це вказує на те, що адміністратор відповідає за створення облікових записів для співробітників у системі.

4. Процес автентифікації: Login (Вхід)

Усі три ролі (Project Manager, Admin, Employee) після реєстрації (або ініціалізації) переходять до процесу Login (Вхід), що є єдиною точкою входу для доступу до функціоналу системи.

5. Функціонал після входу для кожної ролі:

5.1. Функціонал для Project Manager:

Після входу, менеджер проекту може:

- Create Projects: Ініціювати нові проекти в системі.

- Assign Projects to Project Manager: Цей потік означає, що вони можуть брати на себе відповідальність за створені проекти або їх може призначити адміністратор.

- View and Manage all Managers: Це вказує на повноваження менеджера проекту щодо інших менеджерів.

					БР.ІІІ – 45.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		43

- View and Manage all Projects: Повний контроль над проектами.
- View and Manage all Employees: Доступ до інформації про Співробітників та управління ними.

5.2. Функціонал для Admin:

Після входу, адміністратор може:

- View all Projects Assigned to them: Адміністратор може мати призначені йому проекти.
- Assign Projects to Employees: Безпосередньо призначати проекти співробітникам.
- View and Manage all Projects: Повний огляд і контроль над усіма проектами в системі.

5.3. Функціонал для Employee:

Після входу, співробітник може:

- View all Projects Assigned to them: Доступ до своїх призначених завдань/проектів.
- Manage and Update Projects assigned to them: Здійснювати зміни та оновлювати статус своїх призначених проектів.

6. Загальні функції після входу:

Після виконання специфічних для ролі функцій (наприклад, "Create Projects" для Project Manager), всі ролі мають спільні шляхи, що ведуть до:

- Change Password (Змінити Пароль): Ця функція доступна для всіх користувачів після входу, що підвищує безпеку системи.
- Logout (Вихід): Після завершення роботи користувачі можуть вийти з системи.

Отже, діаграма демонструє ієрархічну структуру системи управління завданнями з чітко визначеними ролями та їхніми повноваженнями. Вона показує, як користувачі реєструються, входять у систему та виконують специфічні для своєї ролі операції, а також спільні функції, такі як зміна пароля та вихід. Діаграма підкреслює центральну роль адміністратора в

					БР.ІІІ – 45.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		44

управлінні користувачами та проектами, а також функції менеджерів проектів у координації та призначенні завдань.

2.5. Представлення випадків використання планувальника завдань

Взаємодії між користувачем і системою можуть бути представлені за допомогою діаграми випадків використання. Рисунок 2.6 показує взаємодії трьох основних ролей у додатку для управління завданнями.

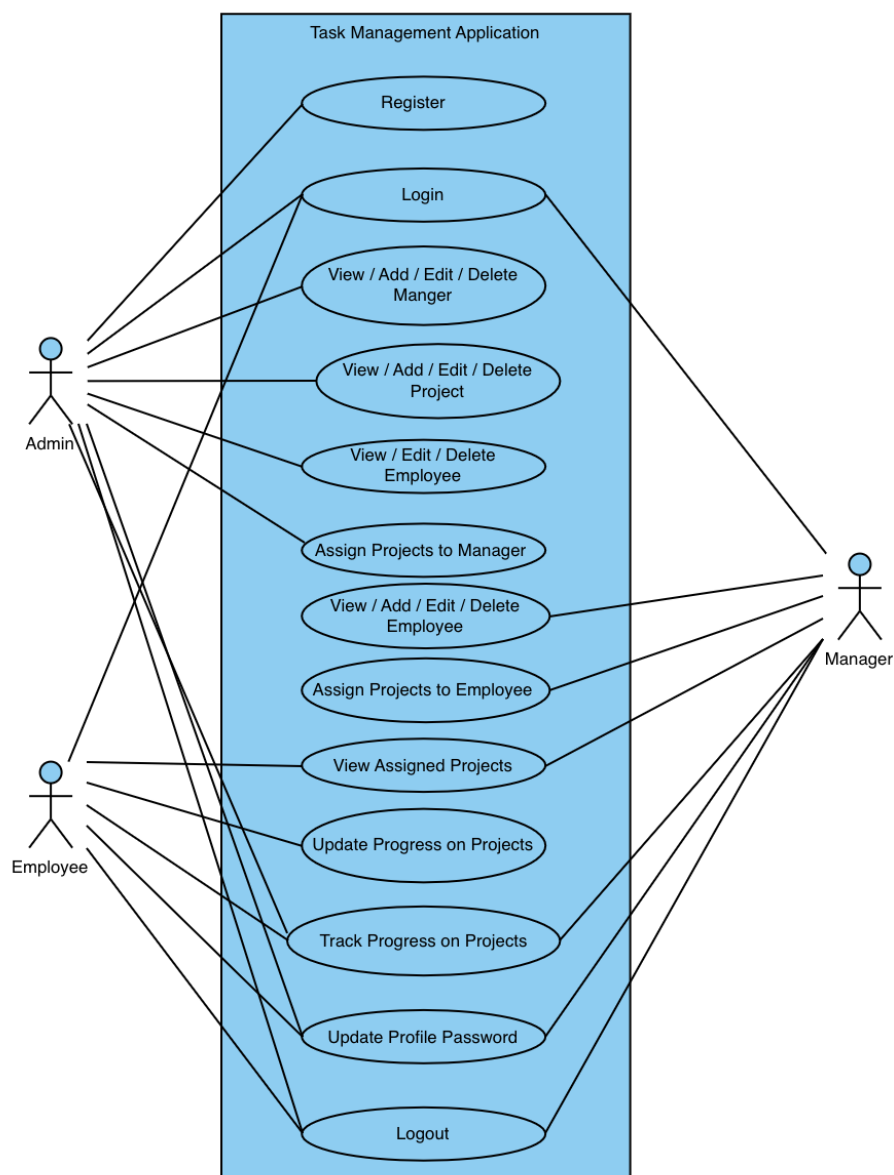


Рисунок 2.6 - Діаграма випадків використання

Діаграма (рис. 2.6) ілюструє взаємодію між користувачами (акторами) та системою, показуючи, які дії (варіанти використання) може виконувати кожен тип користувача.

Actors (Актори): Зовнішні сутності, які взаємодіють із системою. На цій діаграмі представлені три актори:

- Admin (Адміністратор)
- Manager (Менеджер)
- Employee (Співробітник)

Проведемо опис варіантів використання за акторами.

1. Admin (Адміністратор)

Адміністратор має найширші повноваження у системі. Він може виконувати наступні варіанти використання:

- Register (Реєстрація): Реєструвати нових користувачів у системі.
- Login (Вхід): Входити в систему.
- View / Add / Edit / Delete Manger (Перегляд / Додавання / Редагування / Видалення Менеджерів): Повний контроль над обліковими записами менеджерів.
- View / Add / Edit / Delete Project (Перегляд / Додавання / Редагування / Видалення Проектів): Повний контроль над проектами.
- View / Add / Edit / Delete Employee (Перегляд / Додавання / Редагування / Видалення Співробітників): Повний контроль над обліковими записами співробітників.
- Assign Projects to Manager (призначення проектів менеджеру): Делегування проектів менеджерам.
- View / Add / Edit / Delete Employee (Перегляд / Додавання / Редагування / Видалення Співробітників): (Дублюється з попереднім, але підкреслює повний контроль)

					БР.ІІІ – 45.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		46

- Assign Projects to Employee (призначення проектів співробітнику):
Безпосереднє призначення проектів співробітникам.

- Track Progress on Projects (відстеження прогресу по проектах):
Моніторинг виконання всіх проектів.

- Update Profile Password (оновлення пароля профілю): Зміна свого пароля.

- Logout (Вихід): Вихід із системи.

2. Manager (Менеджер)

Менеджер має повноваження, орієнтовані на управління проектами та командами. Він може виконувати наступні варіанти використання:

- Login (Вхід): Входити в систему.

- View / Add / Edit / Delete Employee (Перегляд / Додавання / Редагування / Видалення Співробітників): Контроль над обліковими записами співробітників у своїй команді або під своїм управлінням.

- Assign Projects to Employee (призначення проектів співробітнику):
Делегування завдань/проектів своїм співробітникам.

- Track Progress on Projects (відстеження прогресу по проектах):
Моніторинг виконання проектів, якими він керує.

- Update Profile Password (оновлення пароля профілю): Зміна свого пароля.

- Logout (Вихід): Вихід із системи.

3. Employee (Співробітник)

Співробітник має найменші повноваження, зосереджені на його власних завданнях. Він може виконувати наступні варіанти використання:

- Login (Вхід): Входити в систему.

- View Assigned Projects (перегляд призначених проектів): Переглядати список проектів, які йому призначено.

					БР.ІІІ – 45.00.00.000 ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

- Update Progress on Projects (оновлення прогресу по проектах):
Оновлювати статус або прогрес виконання своїх призначених проектів/завдань.

- Update Profile Password (оновлення пароля профілю): Зміна свого пароля.

- Logout (Вихід): Вихід із системи.

Діаграма чітко показує ієрархію повноважень: адміністратор має повний контроль, менеджер керує командами та проектами, а співробітник виконує та оновлює свої завдання.

Спільні функції: `Login`, `Update Profile Password` та `Logout` є спільними варіантами використання для всіх акторів.

В цілому, діаграма варіантів використання відображає основні функціональні вимоги до додатку управління і планування завдань та взаємодію між різними типами користувачів.

					БР.ІІІ – 45.00.00.000 ПЗ	Арк.
						48
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3. ПРОГРАМНА РОЗРОБКА СИСТЕМИ УПРАВЛІННЯ І ПЛАНУВАННЯ ЗАВДАНЬ

3.1. Запропонована архітектура системи

Запропонована система реалізує три дискретні рівні доступу користувачів, кожен з яких наділений чітко визначеним набором повноважень, адаптованих до функціональних обов'язків у контексті управління проектами та завданнями. Ці ролі включають:

- Адміністратор (Administrator)
- Менеджер проекту (Project Manager)
- Співробітник (Employee)

Кожна роль розроблена для оптимізації робочих процесів та забезпечення ефективної координації в межах організації.

3.1.1. Функціональні повноваження ролей

Роль адміністратора

Адміністратор виконує функції центрального управління та стратегічного нагляду за системою. Його повноваження охоплюють:

- Управління менеджерами проектів: Створення, редагування та видалення облікових записів менеджерів проектів.
- Управління проектами: Повний контроль над життєвим циклом проектів у системі, включаючи їхнє створення, модифікацію та архівацію.
- Управління даними співробітників: Доступ до детальної інформації про всіх Співробітників.
- Організаційний нагляд: Забезпечення коректного розподілу проектів та загального моніторингу системних операцій для підтримки організаційної ефективності.

					БР.ІП – 45.00.00.000 ПЗ	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

Роль менеджера проекту

Менеджери проектів відповідають за оперативне управління проектами та координацію діяльності команд. Їхні ключові функції включають:

- Управління завданнями та командами - делегування завдань Співробітникам та контроль за їхнім виконанням.
- Моніторинг прогресу проектів - відстеження статусу виконання завдань та загального прогресу проектів.
- Оновлення статусів проектів - можливість змінювати статуси проектів відповідно до їхнього поточного стану.
- Перегляд деталей проектів - доступ до повної інформації про проекти, якими вони керують.
- Колаборація - ефективна взаємодія з членами команди для вирішення робочих питань.

Роль співробітника

Співробітники є кінцевими виконавцями завдань, призначених менеджерами проектів. Їхні основні функціональні можливості включають:

- Управління призначеними завданнями - огляд та організація завдань, делегованих їм.
- Оновлення статусів завдань - внесення змін до статусу виконання призначених завдань.
- Подання звітів про прогрес - надання інформації про хід роботи над проектами.
- Комунікація з менеджерами проектів - взаємодія з керівництвом для обговорення питань, що стосуються проектів.
- Доступ до деталей проектів - перегляд інформації, що відноситься до їхніх призначених завдань.

Колаборація - співпраця з іншими членами команди у межах спільного проекту.

					БР.ІІІ – 45.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		50

3.1.2. Забезпечення ефективності системи

Даний структурований підхід до організації ролей та їхніх повноважень спрямований на досягнення наступних ключових цілей:

1. Ефективне управління завданнями: чітке розподілення обов'язків забезпечує впорядкований процес виконання завдань.
2. Чітка комунікація: визначені канали та повноваження сприяють прозорій та ефективній взаємодії між усіма рівнями користувачів.
3. Ефективний нагляд за проектами: забезпечення комплексного моніторингу та контролю за ходом проектів для всіх ролей користувачів у додатку.

3.2. Основна реалізація системи

3.2.1. Налаштування середовища розробки

Початковий етап реалізації проекту передбачав налаштування необхідного середовища розробки. Були встановлені Node.js та Java Development Kit (JDK) з їхніх офіційних дистрибутивів. Для ініціалізації фронтенд-частини на базі React було виконано команду `prx create-react-app task-management-application` у командному рядку. Паралельно здійснювалося конфігурування середовищ Apache, MySQL та MySQL Workbench для забезпечення функціональності веб-сервера та системи управління базами даних.

3.2.2. Розробка фронтенду (UI)

Інтерфейс користувача (UI) системи розроблявся з використанням компонентного підходу, характерного для бібліотеки React. Для стилізації та забезпечення адаптивності застосовувалися HTML та SCSS. Навігація в односторінковому додатку реалізована за допомогою бібліотеки React Router Dom. Управління станом та асинхронними операціями забезпечувалося за

					БР.ІІІ – 45.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		51

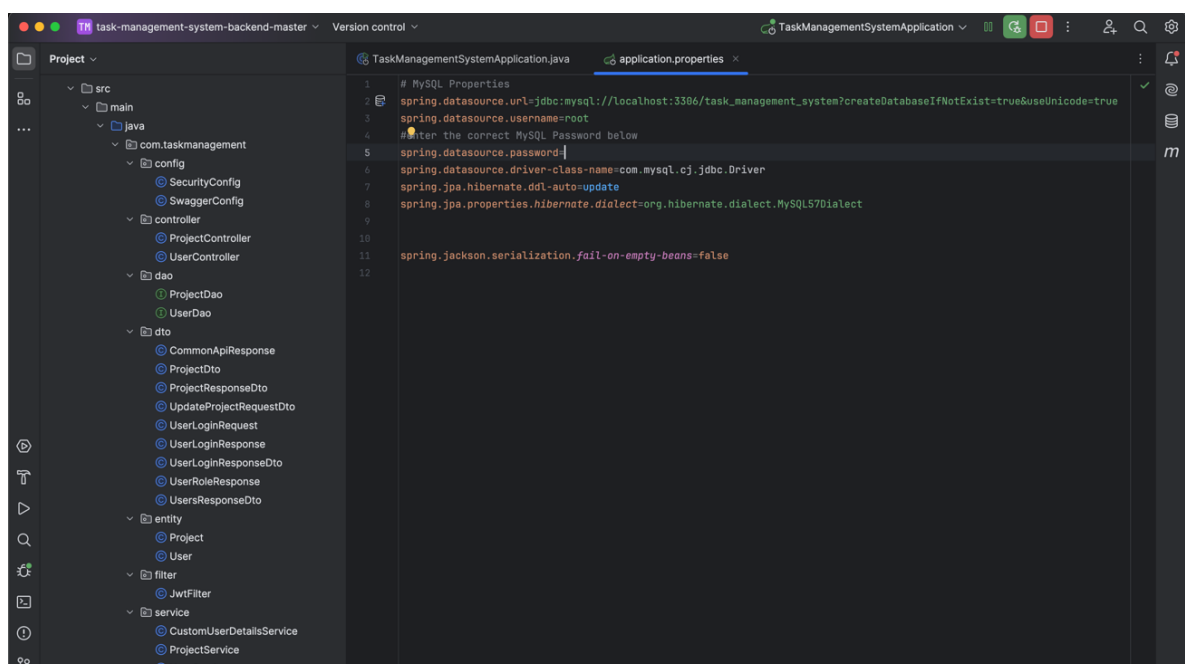
допомогою Redux та Axios відповідно. Фреймворк React Bootstrap використовувався для швидкої розробки привабливого та зручного інтерфейсу, що забезпечило відповідність бізнес-логіки та вимогам до зручності використання.

3.3.3. Розробка бекенду (API)

Для функціональності бекенду було розроблено RESTful API на базі фреймворку Spring Boot. Ці API відповідають за обробку ключових операцій, включаючи управління користувачами (Адміністратори, Менеджери, Співробітники), управління проектами та механізми автентифікації. Особливу увагу було приділено забезпеченню належної авторизації та автентифікації за допомогою JSON Web Tokens (JWT), що гарантує безпеку доступу до ресурсів системи.

3.3.4. Підключення до бази даних

Зв'язок між бекенд-сервером на Spring Boot та базою даних MySQL був встановлений для підтримки операцій CRUD (Create, Read, Update, Delete).



```
1 # MySQL Properties
2 spring.datasource.url=jdbc:mysql://localhost:3306/task_management_system?createDatabaseIfNotExist=true&useUnicode=true
3 spring.datasource.username=root
4 #Enter the correct MySQL Password below
5 spring.datasource.password=
6 spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
7 spring.jpa.hibernate.ddl-auto=update
8 spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL57Dialect
9
10
11 spring.jackson.serialization.fail-on-empty-beans=false
12
```

Рисунок 3.1 – Схема підключення до бази даних

Конфігурація підключення до бази даних здійснювалася через файл `application.properties` у проєкті Spring Boot, що є стандартною практикою для налаштування параметрів доступу до бази даних (як показано на рисунку 3.1, що демонструє схему підключення до бази даних).

3.2.5. Взаємодія фронтенду та бекенду

Взаємодія між фронтом та бекенд-API реалізована за допомогою HTTP-запитів. Для відправки цих запитів з клієнтської частини до RESTful сервісів бекенду використовувався пакет Axios. Цей вибір дозволив ефективно виконувати асинхронні дії в межах викликів API, забезпечуючи плавну та інтерактивну взаємодію користувача із системою.

3.2.6. Локальне розгортання та тестування

Для цілей розробки та тестування було виконано налаштування локальних серверів Apache та MySQL. Це дозволило розробляти та тестувати функціонал веб-додатку в контрольованому середовищі локального хосту, забезпечуючи безперебійну інтеграцію компонентів та належну функціональність бази даних до етапу розгортання.

3.3. Реалізація інтерфейсу головної сторінки, реєстрації та автентифікації

Цей підрозділ описує початкові інтерфейси користувача та механізми доступу до системи управління завданнями.

3.3.1. Головна сторінка

При доступі до додатку для управління та планування завданнями через веб-браузер, користувачу відображається головна сторінка. Візуальне представлення цієї сторінки наведено на рисунку 3.2. Дизайн інтерфейсу був

					БР.ІІІ – 45.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		53

автентифікації користувачів відображено на рисунку 3.4. Ця сторінка вимагає введення облікових даних для отримання доступу до функціоналу додатку.

The screenshot shows a web interface for 'Task Management System' with a navigation bar containing 'About Us' and 'Contact Us' links, and a 'Login' link on the right. The main content is a 'Register admin' form with the following fields:

- First Name
- Last Name
- Email Id
- Password
- User Gender (with a 'Select Sex' dropdown)
- Contact No
- Age
- Street
- City
- Pincode
- State
- Country

A 'Register User' button is located at the bottom of the form.

Рисунок 3.3 - Інтерфейс сторінки реєстрації адміністратора

The screenshot shows a web interface for 'Task Management System' with a navigation bar containing 'About Us' and 'Contact Us' links, and a 'Login' link on the right. The main content is a 'Login' form with the following fields:

- User Role (with a 'Select Role' dropdown)
- Email Id
- Password

A 'Login' button is located at the bottom of the form.

Рисунок 3.4 - Інтерфейс сторінки входу до системи

Ці сторінки є критично важливими для забезпечення контрольованого доступу до ресурсів та функціоналу системи управління завданнями.

3.4.1. Управління менеджерами проектів

Адміністратор наділений повноваженнями щодо управління обліковими записами менеджерів проектів:

- Реєстрація менеджерів.

Адміністратор може ініціювати створення нових облікових записів для менеджерів проектів розроблюваної системи, як продемонстровано на рисунку 3.6.

Для реєстрації менеджера необхідно заповнити наступні поля - First Name (Ім'я), Last Name (Прізвище), Email Id (Електронна пошта), Password (Пароль), User Gender (Стать користувача) з випадającym списком "Select Sex" (Вибрати стать), Contact No (Контактний номер), Age (Вік), Street (Вулиця), City (Місто), Pincod (Поштовий індекс), State (Область/Штат), Country (Країна)

The screenshot shows a web interface for a 'Task Management System'. At the top, there is a navigation bar with links: 'About Us', 'Contact Us', 'Register Manager', 'Add Project', 'Projects', 'Managers', 'Employees', 'Change Password', and 'Logout'. The main content area features a 'Register manager' form with the following fields:

- First Name
- Last Name
- Email Id
- Password
- User Gender (with a dropdown menu labeled 'Select Sex')
- Contact No
- Age
- Street
- City
- Pincod
- State
- Country

A 'Register User' button is located at the bottom of the form.

Рисунок 3.6 - Інтерфейс реєстрації менеджера адміністратором

- Перегляд та управління менеджерами

Доступ до повного списку зареєстрованих менеджерів проектів з можливістю їхнього перегляду, редагування та видалення. Ця функціональність ілюструється на рисунку 3.7.

All Managers					
First Name	Last Name	Email Id	Phone No	Address	Action
Emily	Johnson Manager	EmilyJohnsonManager@gmail.com	9081636681	5500 University	Remove
Michael	Brown Manager	MichaelBrownManager@gmail.com	9091275631	1930 W College	Remove
Jessica	Davis Manager	JessicaDavisManager@gmail.com	9091472774	1830 Sierra	Remove
William	Jones Manager	WilliamJonesManager@gmail.com	7748927747	4444 University	Remove
Ashley	Miller Manager	AshleyMillerManager@gmail.com	9573728849	4972 Cambridge	Remove
James	Wilson Manager	JamesWilsonManager@gmail.com	8374088817	2065 W College	Remove
Amanda	Moore Manager	AmandaMooreManager@gmail.com	8375729918	2065 W College	Remove

Рисунок 3.7 - Інтерфейс перегляду та управління менеджерами адміністратором

3.4.2. Управління проектами

Адміністратор має повний контроль над проектами в системі:

- Додавання проектів.

Можливість додавання нових проектів до системи, інтерфейс якої показано на рисунку 3.8.

- Перегляд та управління проектами.

Доступ до повного огляду всіх проектів, зареєстрованих у системі, з функціями їхнього перегляду, редагування та архівації (як показано на рисунку 3.9).

- Призначення проектів менеджерам:

Функціонал для делегування проектів конкретним менеджерам проектів (зображено на рисунку 3.10).

The screenshot shows a web interface for adding a project. At the top, there is a navigation bar with the logo 'Task Management System' and links for 'About Us', 'Contact Us', 'Register Manager', 'Add Project', 'Projects', 'Managers', 'Employees', 'Change Password', and 'Logout'. The main content is a modal form titled 'Add Project'. It contains four sections: 'Project Name' with a text input field containing 'enter name..'; 'Project Description' with a large text area; 'Project Requirement' with another large text area; and 'Project Deadline' with a date input field showing 'mm/dd/yyyy' and a calendar icon. An 'Add Project' button is located at the bottom of the form.

Рисунок 3.8 - Інтерфейс додавання проекту адміністратором

The screenshot shows the 'All Projects' dashboard. It features a search bar with two input fields: 'Enter Project Name...' and 'Enter Project Id...', each with a 'Search' button. Below the search bar is a table with the following data:

Project Name	Project Description	Project Requirement	Manager Assign Status	Manager Name	Employee Assign Status	Employee Name	Project Created Date	Project Assign Date	Project Deadline	Project Status	Action
Revise User Guide	Revise the user guide for the Online Food Ordering System	Ensure all sections are updated with current features and functionalities.	Not Assigned	Not Assigned	Not Assigned	Not Assigned	2024-07-10	Not Assigned	2024-07-26	Not Assigned to Manager	Assign To Manager
Enhance UI Design	Enhance the user interface design for the Student Attendance Tracker.	Implement modern design principles and improve usability.	Assigned to Manager	Emily Johnson Manager	Assigned to Employee	José Martínez	2024-07-10	2024-07-10	2024-08-02	Working	
Implement Search Functionality	Implement search functionality for the Library Management System.	Ensure search feature supports multiple criteria and is efficient.	Not Assigned	Not Assigned	Not Assigned	Not Assigned	2024-07-10	Not Assigned	2024-07-29	Not Assigned to Manager	Assign To Manager
Optimize Database Queries	Optimize database queries for the E-commerce Platform.	Improve query performance to reduce page load times.	Assigned to Manager	Michael Brown Manager	Not Assigned	Not Assigned	2024-07-10	2024-07-10	2024-09-12	Not Assigned to Employee	
Develop Mobile App Prototype	Develop a prototype for the Health Tracking Dashboard mobile app.	Include basic functionality for tracking health metrics.	Not Assigned	Not Assigned	Not Assigned	Not Assigned	2024-07-10	Not Assigned	2024-08-10	Not Assigned to Manager	Assign To Manager
Integrate Payment Gateway	Integrate a payment gateway into the Tourism	Ensure secure and reliable payment processing.	Assigned to Manager	James Wilson	Not Assigned	Not Assigned	2024-07-10	2024-07-10	2024-09-13	Not Assigned to Employee	

Рисунок 3.9 - Інтерфейс перегляду та управління всіма проектами адміністратором

Task Management System | About Us | Contact Us | Register Manager | Add Project | Projects | Managers | Employees | Change Password | Logout

Assign Project To Manager

Project Name

Project Description

Project Created Date

Project Deadline Date

Assign Project To Manager

Рисунок 3.10 - Інтерфейс призначення проекту менеджеру адміністратором

3.4.3. Управління співробітниками

Адміністратор також здійснює нагляд за обліковими записами співробітників:

- Перегляд та управління співробітниками

Доступ до списку всіх співробітників із можливостями їхнього перегляду та управління (як показано на рисунку 3.11).

Task Management System | About Us | Contact Us | Register Manager | Add Project | Projects | Managers | Employees | Change Password | Logout

All Employee

First Name	Last Name	Email Id	Phone No	Address	Action
José	Martínez	JoseMartinez@gmail.com	9091747772	1831 Northpark	<input type="button" value="Remove"/>
María	García	MariaGarcia@gmail.com	7361759284	1505 Northpark	<input type="button" value="Remove"/>
Juan	Rodríguez	JuanRodriguez@gmail.com	7164679992	777 San Manuel	<input type="button" value="Remove"/>
Carmen	López	CarmenLopez@gmail.com	6175727771	1251 Research	<input type="button" value="Remove"/>
Luis	Hernández	LuisHernandez@gmail.com	9914777713	2351 W Lugonia	<input type="button" value="Remove"/>
Ana	Pérez	AnaPerez@gmail.com	9816723882	26200 Redlands	<input type="button" value="Remove"/>
Carlos	González	CarlosGonzalez@gmail.com	8172877782	25948 Business	<input type="button" value="Remove"/>
Isabel	Sánchez	IsabelSanchez@gmail.com	8881937771	25828 Redlands	<input type="button" value="Remove"/>
Miguel	Rivera	MiguelRivera@gmail.com	6657269917	130 E San	<input type="button" value="Remove"/>
Sofia	Flores	SofiaFlores@gmail.com	7264791928	10951 Cedar	<input type="button" value="Remove"/>

Рисунок 3.11 - Інтерфейс перегляду та управління всіма співробітниками адміністратором

3.4.4. Загальні функції користувача

Крім специфічних адміністративних функцій, адміністратор, подібно до інших типів користувачів, має доступ до загальних функціональних можливостей:

- Зміна пароля облікового запису

Можливість оновлення власного пароля для підвищення безпеки, що є універсальною функцією для всіх ролей (рис. 3.12).

Рисунок 3.12 - Інтерфейс зміни пароля для всіх користувачів

- Вихід із системи

Адміністратор може безпечно завершити сесію в додатку для управління завданнями, скориставшись кнопкою виходу (рис. 3.13).

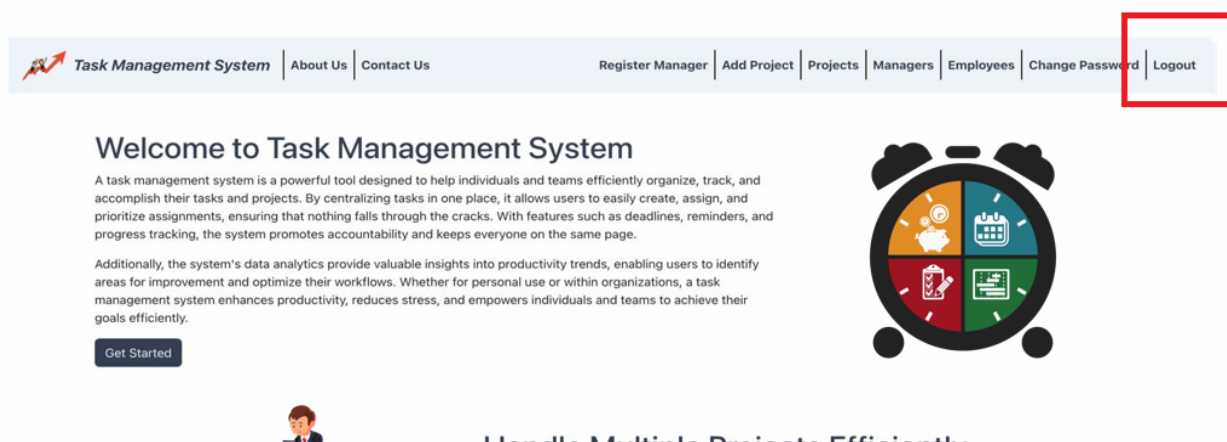


Рисунок 3.13 - Компонент виходу з системи для всіх користувачів

					БР.ІІІ – 45.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		61

Таким чином, функціональність адміністратора забезпечує централізований контроль та нагляд за всіма ключовими аспектами системи управління завданнями, від управління користувачами до повного життєвого циклу проєктів.

3.5. Функціональні можливості інтерфейсу менеджера

Після успішної автентифікації користувача з роллю "менеджер" відбувається перенаправлення на головний екран системи (рис. 3.14).

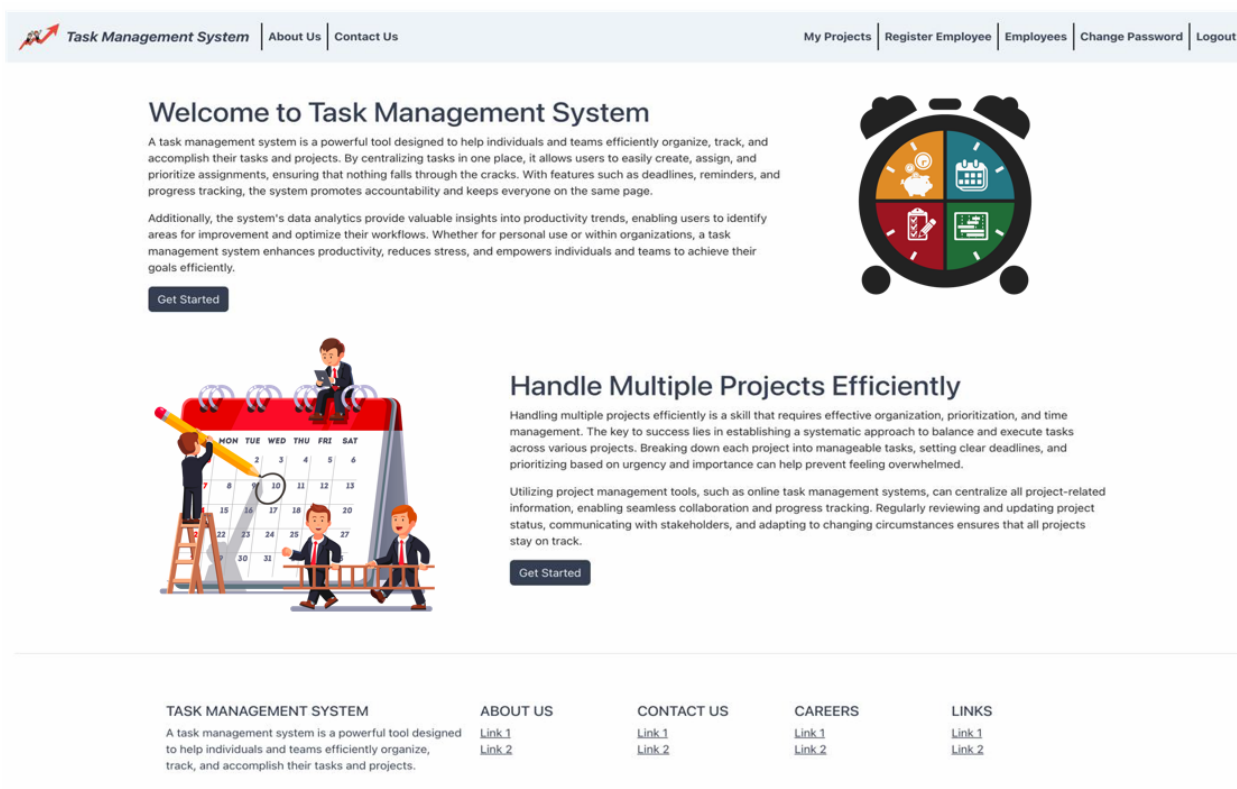


Рисунок 3.14 - Головний екран інтерфейсу менеджера

Інтерфейс менеджера надає доступ до розширеного набору функціональних можливостей, реалізованих через навігаційну панель, розташовану у верхній частині екрану. Доступні опції включають: перегляд та управління призначеними проєктами, призначення проєктів

					БР.ІП – 45.00.00.000 ПЗ	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

співробітникам, реєстрацію нових співробітників, а також перегляд та адміністрування переліку всіх співробітників.

Система дозволяє менеджеру здійснювати реєстрацію нового співробітника (рис. 3.15) та переглядати й редагувати дані наявних співробітників (рис. 3.16).

The screenshot shows a web interface for a 'Task Management System'. At the top, there is a navigation bar with links: 'Task Management System', 'About Us', 'Contact Us', 'My Projects', 'Register Employee', 'Employees', 'Change Password', and 'Logout'. The main content area is a form titled 'Register employee'. The form contains the following fields:

- First Name
- Last Name
- Email Id
- Password
- User Gender (with a dropdown menu labeled 'Select Sex')
- Contact No
- Age
- Street
- City
- Pincode
- State
- Country

A 'Register User' button is located at the bottom center of the form.

Рисунок 3.15 - Форма реєстрації співробітника менеджером

The screenshot shows the 'All Employee' interface in the 'Task Management System'. The navigation bar is identical to the previous screenshot. The main content area is a table titled 'All Employee' with the following columns: First Name, Last Name, Email Id, Phone No, Address, and Action. The table contains ten rows of employee data:

First Name	Last Name	Email Id	Phone No	Address	Action
José	Martínez	JoseMartinez@gmail.com	9091747772	1831 Northpark	Remove
María	García	MariaGarcia@gmail.com	7361759284	1505 Northpark	Remove
Juan	Rodríguez	JuanRodriguez@gmail.com	7164679992	777 San Manuel	Remove
Carmen	López	CarmenLopez@gmail.com	6175727771	1251 Research	Remove
Luis	Hernández	LuisHernandez@gmail.com	9914777713	2351 W Lugonia	Remove
Ana	Pérez	AnaPerez@gmail.com	9816723882	26200 Redlands	Remove
Carlos	González	CarlosGonzalez@gmail.com	8172877782	25948 Business	Remove
Isabel	Sánchez	IsabelSanchez@gmail.com	8881937771	25828 Redlands	Remove
Miguel	Rivera	MiguelRivera@gmail.com	6657269917	130 E San	Remove
Sofia	Flores	SofiaFlores@gmail.com	7264791928	10951 Cedar	Remove

Рисунок 3.16 - Інтерфейс перегляду та управління даними співробітників

Аналогічним чином, менеджер має можливість переглядати та керувати всіма призначеними проектами (рис. 3.17), а також розподіляти їх між співробітниками (рис. 3.18).

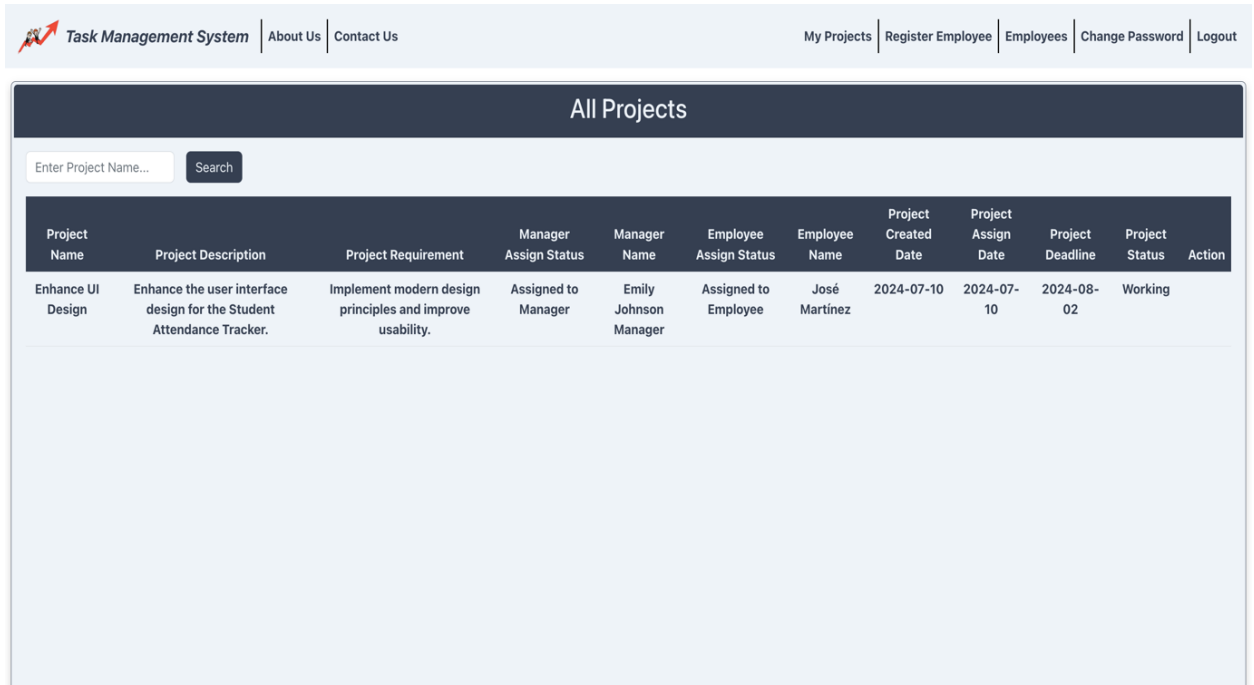


Рисунок 3.17 - Інтерфейс перегляду та управління призначеними проектами

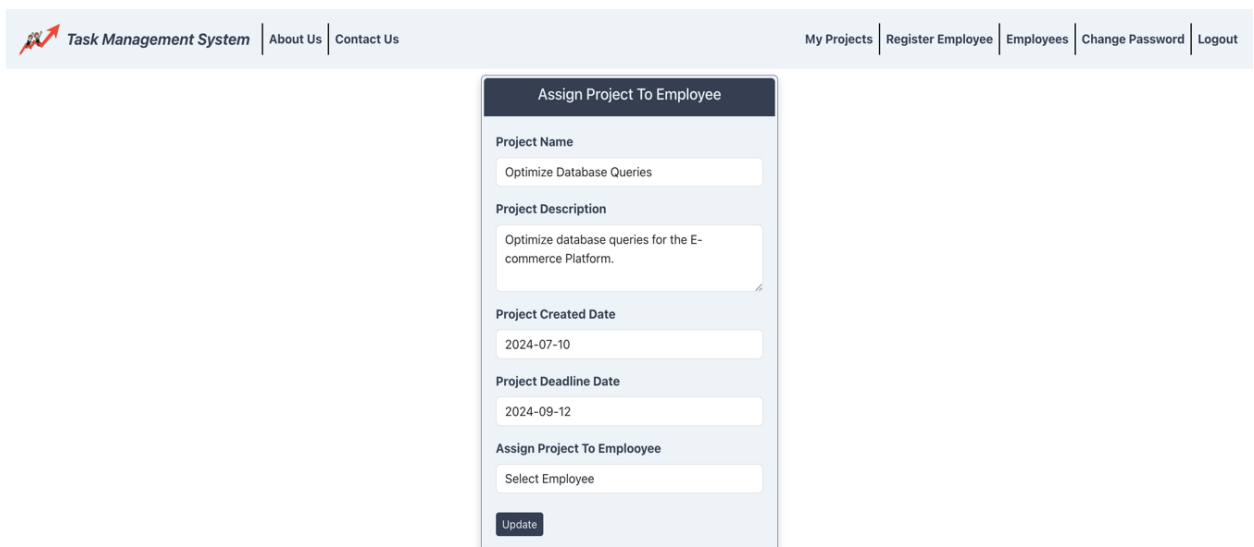


Рисунок 3.18 - Інтерфейс призначення проекту співробітнику

3.6. Функціональні можливості інтерфейсу співробітника

Після успішної автентифікації користувача з роллю "співробітник" здійснюється автоматичне перенаправлення на головний екран системи (рис. 3.19).

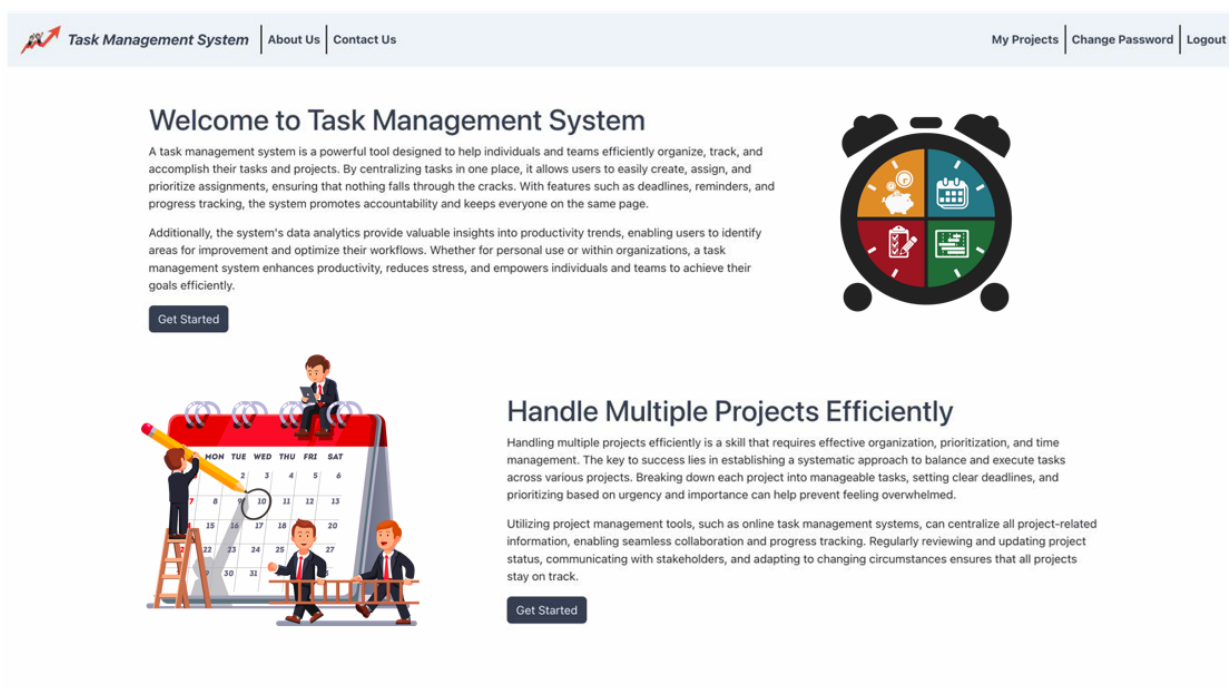


Рисунок 3.19 - Головний екран інтерфейсу співробітника

Інтерфейс співробітника забезпечує доступ до призначених йому проектів за допомогою навігаційної панелі, розташованої у верхній частині екрану (рис. 3.20). Співробітник, на відміну від менеджера і адміністратора має обмежені права, а тому в меню в нього представлено тільки три пункти:

- мої проекти;
- змінити пароль;
- вихід.

					БР.ІП – 45.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		65

Співробітники мають можливість оновлювати статус та прогрес виконання проєктів (рис. 3.21). На рисунку 3.22 продемонстровано екран, що відображається після успішного завершення проєкту співробітником.

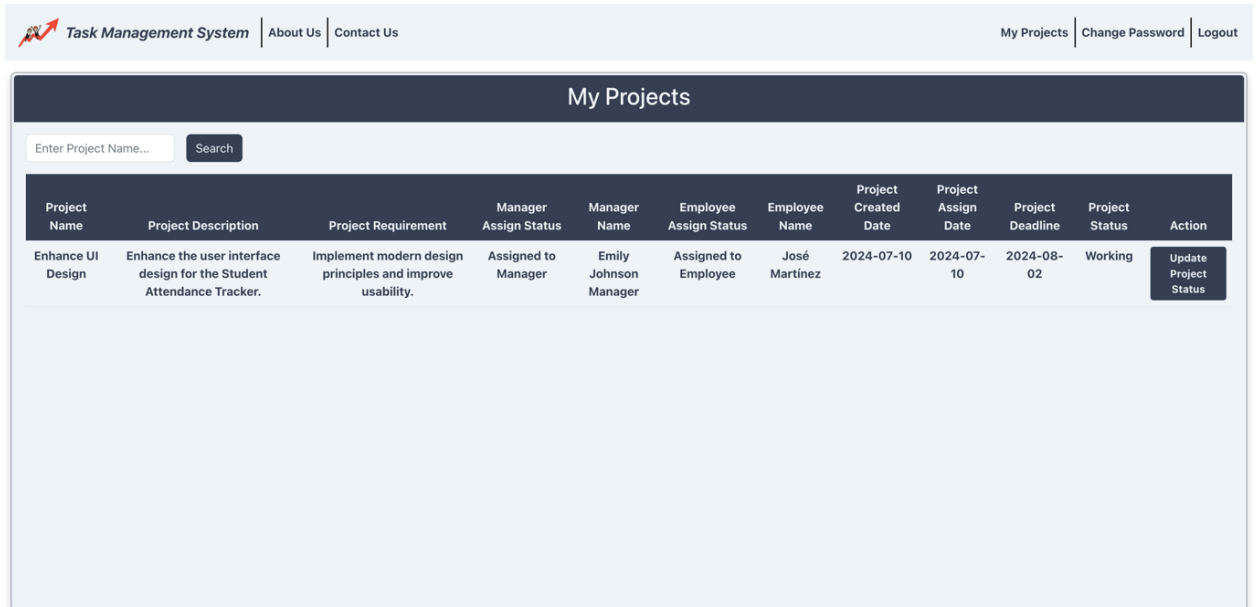


Рисунок 3.20 - Інтерфейс перегляду та управління призначеними проєктами для співробітника

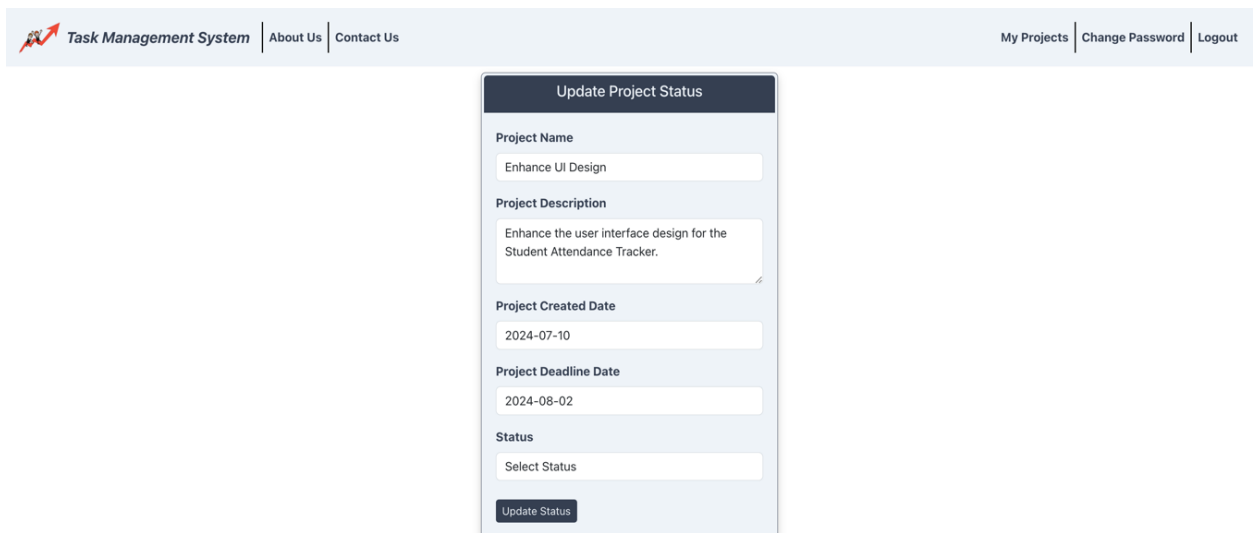


Рисунок 3.21 - Інтерфейс оновлення прогресу виконання проєкту співробітником

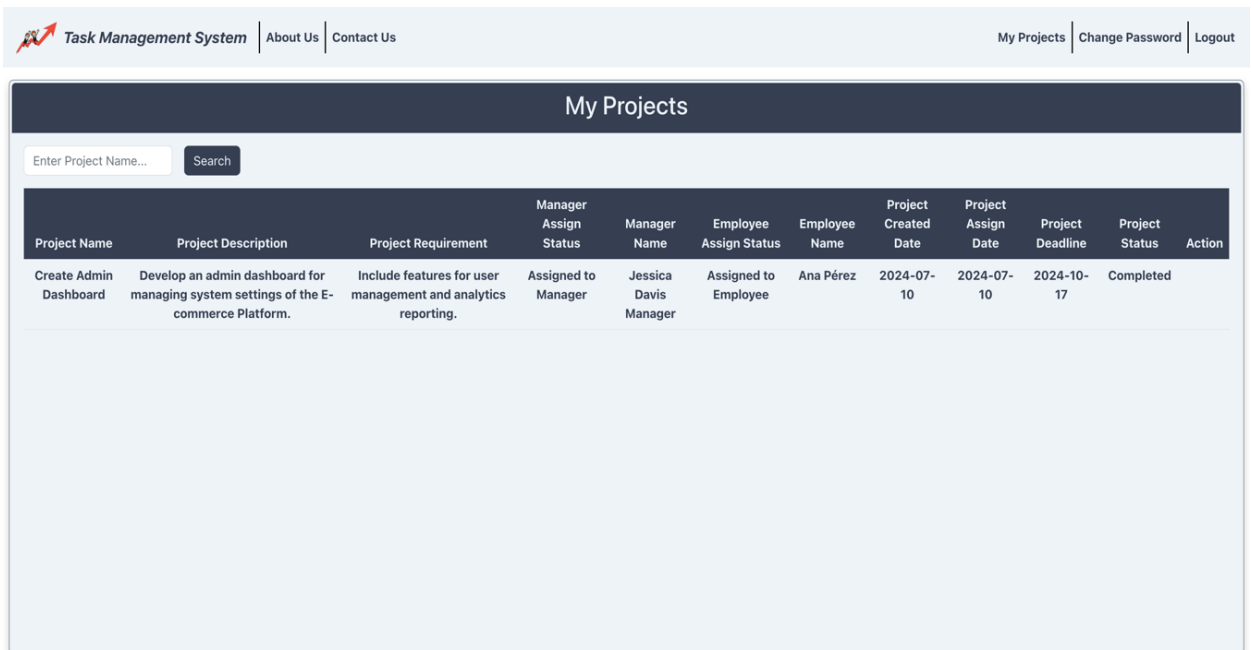


Рисунок 3.22 - Інтерфейс перегляду завершених проєктів співробітником

Представлений програмний додаток для управління завданнями є комплексним інструментальним засобом, розробленим для ефективного адміністрування та моніторингу завдань і проєктів на різних ієрархічних рівнях організації. Застосування надійної та масштабованої серверної архітектури (бек-енду) забезпечує стабільне та продуктивне управління проєктною діяльністю. Це дозволяє адміністраторам системи здійснювати загальний нагляд за операційними процесами, керівникам проєктів – делегувати завдання та контролювати їх виконання, а співробітникам – результативно виконувати покладені на них обов'язки.

Додаток сприяє підвищенню продуктивності та оптимізації комунікаційних процесів завдяки чітко визначеній структурі та набору спеціалізованих функцій для кожної визначеної ролі користувача. Інтуїтивно зрозумілий користувацький інтерфейс забезпечує ефективну взаємодію з системою для всіх категорій користувачів, незалежно від їхньої ролі та рівня технічної підготовки.

Використання даної технологічної платформи дозволяє організаціям оптимізувати робочі процеси, вдосконалити механізми управління

завданнями та посилити командну взаємодію. Таким чином, система управління завданнями є цінним інструментальним рішенням для підприємств, що прагнуть раціоналізувати процеси управління проєктами та підвищити ефективність досягнення поставлених цілей.

3.7. Перспективи розвитку та потенційні вдосконалення системи планування та управління завданнями

Незважаючи на наявний широкий функціонал системи управління завданнями, що забезпечує ефективне адміністрування та моніторинг проєктів і завдань, існує низка потенційних напрямків для її подальшого вдосконалення з метою підвищення продуктивності та оптимізації користувацького досвіду.

Пропоновані напрямки модернізації:

1. Декомпозиція завдань.

Реалізація функціональної можливості поділу комплексного проєкту на ієрархію підзавдань дозволить забезпечити більш гранульований контроль та моніторинг окремих компонентів проєкту. Такий підхід сприятиме деталізованому відстеженню прогресу та більш точному розподілу відповідальності.

2. Система нотифікацій.

Інтеграція системи сповіщень для керівників проєктів, що інформуватиме їх в режимі реального часу про оновлення статусу або прогресу виконання завдань співробітниками, забезпечить оперативне інформування про поточний стан розробки. Це, в свою чергу, сприятиме своєчасному прийняттю управлінських рішень та підтримці динаміки реалізації проєкту.

3. Розширені можливості форматування та документообігу.

					БР.ІІІ – 45.00.00.000 ПЗ	Арк.
						68
Змн.	Арк.	№ докум.	Підпис	Дата		

Впровадження інструментів форматування текстового опису проєктів, а також функції додавання супровідних документів, збагатить інформаційний контент та підвищить рівень деталізації проєктної документації. Це дозволить сформувавши більш повне уявлення про вимоги та етапи виконання проєкту, оптимізуючи загальний процес управління.

4. Управління командами.

Розробка модуля для створення та адміністрування робочих груп в рамках системи надасть керівникам проєктів інструментарій для ефективного управління кількома командами та відстеження їхньої сукупної продуктивності. Окрім того, співробітники отримають можливість моніторити прогрес колег у межах команди, що стимулюватиме покращення співпраці та командної взаємодії.

Імплементация зазначених удосконалень дозволить підвищити гнучкість та розширити функціональні можливості системи управління завданнями, утверджуючи її як незамінний інструмент для ефективного управління проєктами та організації командної роботи.

					БР.ІІІ – 45.00.00.000 ПЗ	Арк.
						69
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У процесі виконання дипломної роботи було реалізовано повнофункціональний веб-додаток для управління персональними завданнями та проектами, що охоплює сучасні підходи до проектування, розробки та тестування інформаційних систем. Робота включає всебічне дослідження предметної області, аналіз існуючих рішень, постановку вимог до функціоналу та архітектури системи, вибір відповідного інструментарію, а також безпосередню реалізацію і тестування програмного забезпечення.

На першому етапі роботи було проведено глибокий аналіз функціональних потреб користувачів, визначено рольову модель системи (адміністратор, менеджер проекту, співробітник), сформовано перелік вимог до функціональності, інтерфейсу та безпеки додатка. Було обґрунтовано актуальність тематики розробки, враховуючи зростаючу потребу в інструментах цифрового планування та підвищення продуктивності командної роботи.

Дослідження ринку сучасних інструментів управління завданнями (Monday.com, MeisterTask, Wrike, Notion) дозволило виявити переваги та недоліки наявних рішень, що, у свою чергу, стало підґрунтям для формування функціонального ядра власного програмного продукту. Було зроблено акцент на інтуїтивному інтерфейсі, адаптованості під різні категорії користувачів та можливості гнучкої інтеграції в існуючі робочі процеси.

На етапі технічного проектування та реалізації було визначено вимоги до програмного та апаратного забезпечення, обґрунтовано вибір технологічного стеку: Spring Boot як серверна платформа, React для реалізації фронтенду, MySQL для збереження даних, а також додаткові інструменти (Maven, Bootstrap), які забезпечили ефективну організацію розробки та привабливий інтерфейс користувача.

					БР.ІІІ – 45.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		70

У третьому розділі представлено повну реалізацію програмної системи: розроблено архітектуру, налагоджено взаємодію між фронтендом і бекендом, реалізовано ключові функції — реєстрацію, автентифікацію, управління проектами, завданнями, користувачами. Особливу увагу приділено розробці рольового доступу та безпеки взаємодії з даними. Кожен інтерфейс — адміністратора, менеджера проектів, співробітника — містить набір інструментів, адаптований до його обов’язків і повноважень.

Проведене тестування системи на локальному середовищі засвідчило її стабільність, зручність використання та відповідність поставленим вимогам. У роботі також окреслено перспективи подальшого розвитку програмного продукту, серед яких — розширення аналітичних функцій, інтеграція з календарями, підтримка мобільної версії, впровадження системи нагадувань та використання засобів штучного інтелекту для прогнозування завантаження ресурсів.

					БР.ІІІ – 45.00.00.000 ПЗ	Арк.
						71
Змн.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. monday.com Work Platform | Made For Work, Designed To Love - <https://monday.com/>
2. The ultimate guide to Notion templates - <https://www.notion.com/help/guides/the-ultimate-guide-to-notion-templates>
3. Wrike Views – Wrike Help Center - <https://help.wrike.com/hc/en-us/articles/360056639873-Wrike-Views>
4. Explore MeisterTask Pro Features – MeisterTask - <https://support.meistertask.com/hc/en-us/articles/360015778520-Explore-MeisterTask-Pro-Features>
5. 5 Best Free MySQL Database Clients in 2025 | DbGate - <https://dbgate.org/blog/2025-02-23-free-mysql-clients/>
6. Spring Boot Architecture. Several application development... | by Uday Patil | Medium - <https://medium.com/@udaypatil318/spring-boot-architecture-39935654ce5c>
7. Maven Architecture – YouTube - <https://www.youtube.com/watch?v=KUiYqCQq6wQ>
8. Schwaber, K., & Sutherland, J. (2020). The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game. Scrum.org.
9. Sommerville, I. (2016). Інженерія програмного забезпечення (10-е вид.). Pearson.
10. Martin, R. C. (2008). Чистий код: Посібник з гнучкої майстерності програмного забезпечення. Prentice Hall. [Англ.: Clean Code: A Handbook of Agile Software Craftsmanship]
11. Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley.
12. Fowler, M. (2002). Patterns of Enterprise Application Architecture. Addison-Wesley.

					БР.ІІІ – 45.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		72

13. Pressman, R. S., & Maxim, B. R. (2020). Інженерія програмного забезпечення: підхід практика (9-е вид.). McGraw-Hill Education. [Англ.: Software Engineering: A Practitioner's Approach]
14. Walls, C. (2022). Spring Boot: Up and Running: Building Cloud Native Java and Kotlin Applications. O'Reilly Media.
15. Deinum, M., et al. (2021). Pro Spring 5: An In-Depth Guide to the Spring Framework and Its Tools (5th ed.). Apress.
16. Spring Framework Committers. (2024). Spring Boot Reference Documentation (Version 3.3.0). <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>
17. Banks, A., & Porcello, E. (2020). Learning React: Modern Patterns for Developing React Apps (2nd ed.). O'Reilly Media.
18. DuBois, P. (2014). MySQL Cookbook: Solutions for Database Developers and Administrators (3rd ed.). O'Reilly Media.
19. Oracle Corporation. (2024). MySQL 8.0 Reference Manual. Отримано з <https://dev.mysql.com/doc/refman/8.0/en/>
20. Elmasri, R., & Navathe, S. B. (2017). Fundamentals of Database Systems (7th ed.). Pearson.
21. Apache Maven Committers. (2024). Apache Maven Project – Documentation. <https://maven.apache.org/guides/>
22. The Bootstrap Authors. (2024). Bootstrap Documentation (Version 5.3). <https://getbootstrap.com/docs/5.3/getting-started/introduction/>
23. Krug, S. (2014). Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability (3rd ed.). New Riders.
24. Norman, D. A. (2013). The Design of Everyday Things: Revised and Expanded Edition. Basic Books.
25. Shneiderman, B., et al. (2016). Designing the User Interface: Strategies for Effective Human-Computer Interaction (6th ed.). Pearson.

- 26.OWASP Foundation. (2021). OWASP Top 10 - 2021: The Ten Most Critical Web Application Security Risks. <https://owasp.org/www-project-top-ten/>
- 27.Sandhu, R. S., Coyne, E. J., Feinstein, H. L., & Youman, C. E. (1996). Role-based access control models. *IEEE Computer*, 29(2), 38-47.
- 28.Ferraiolo, D. F., Sandhu, R., Gavrila, S., Kuhn, D. R., & Chandramouli, R. (2001). Proposed NIST standard for role-based access control. *ACM Transactions on Information and System Security (TISSEC)*, 4(3), 224-274.
- 29.Wiegers, K., & Beatty, J. (2013). *Software Requirements* (3rd ed.). Microsoft Press.
- 30.Pohl, K. (2010). *Requirements Engineering: Fundamentals, Principles, and Techniques*. Springer.
- 31.Myers, G. J., Sandler, C., & Badgett, T. (2011). *The Art of Software Testing* (3rd ed.). Wiley.
- 32.Ammann, P., & Offutt, J. (2016). *Introduction to Software Testing* (2nd ed.). Cambridge University Press.
- 33.Kerzner, H. (2017). *Project management: a systems approach to planning, scheduling, and controlling*. John Wiley & Sons.
- 34.Brooks Jr, F. P. (1995). *The mythical man-month: Essays on software engineering, anniversary edition*. Addison-Wesley Professional.
- 35.Monday.com. (2024). Product Features. <https://monday.com/features>
- 36.Wrike, Inc. (2024). Wrike Features. <https://www.wrike.com/features/>
- 37.MeisterLabs GmbH. (2024). MeisterTask Features. <https://www.meistertask.com/features>
- 38.Notion Labs, Inc. (2024). Notion Product Features. <https://www.notion.so/product>
- 39.Bass, L., Clements, P., & Kazman, R. (2021). *Software Architecture in Practice* (4th ed.). Addison-Wesley Professional.

					БР.ІІІ – 45.00.00.000 ІІЗ	Арк. 74
Змн.	Арк.	№ докум.	Підпис	Дата		

40.Larman, C. (2004). Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd ed.). Prentice Hall.

41.Object Management Group. (2011). Business Process Model and Notation (BPMN) Version 2.0. OMG Document Number formal/2011-01-03.

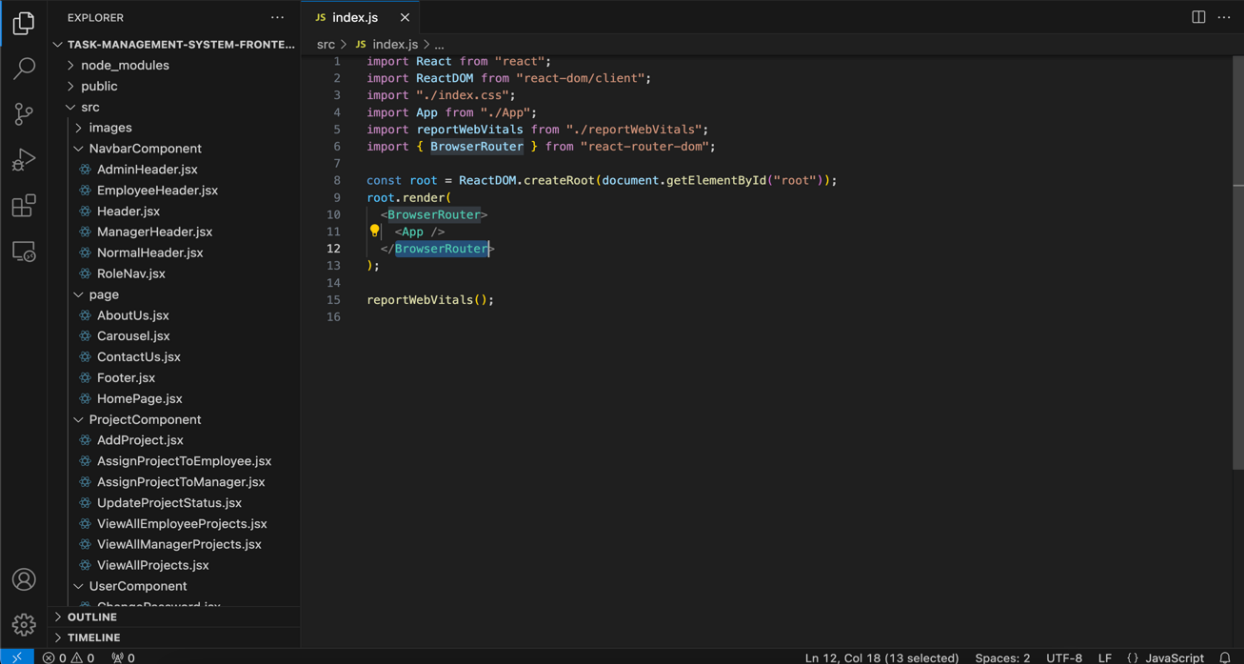
Отримано з <https://www.omg.org/spec/BPMN/2.0/>

					БР.ІІІ – 45.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		75

ДОДАТКИ

Додаток А

Рисунок А.1 відображає файл Index.js додатку для управління завданнями, який служить як кореневий файл у фронтенд-проекті, побудованому з React. Його структура на основі компонентів полегшує виконання всіх дочірніх компонентів відповідно до конкретних вимог користувача.



```
1 import React from "react";
2 import ReactDOM from "react-dom/client";
3 import "./index.css";
4 import App from "./App";
5 import reportWebVitals from "./reportWebVitals";
6 import { BrowserRouter } from "react-router-dom";
7
8 const root = ReactDOM.createRoot(document.getElementById("root"));
9 root.render(
10   <BrowserRouter>
11     <App />
12   </BrowserRouter>
13 );
14
15 reportWebVitals();
16
```

Рисунок А.1 - Файл Index.js фронтенду

Рисунок А. 2 демонструє файл TaskManagementSystemApplication.java додатку для управління завданнями, який служить як кореневий файл у бекенд-проекті, розробленому з Spring Boot. Цей файл ініціалізує та запускає сервер для додатку.

```
1 package com.taskmanagement;
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5 import org.springframework.boot.web.servlet.support.SpringBootServletInitializer;
6
7 @SpringBootApplication
8 public class TaskManagementSystemApplication extends SpringBootServletInitializer {
9
10
11     public static void main(String[] args) {
12         SpringApplication.run(TaskManagementSystemApplication.class, args);
13     }
14 }
15
```

Рисунок А.2 – Файл бекенду TaskManagementSystemApplication.java

БІБЛІОГРАФІЧНА ДОВІДКА

Тема дипломної роботи: “ Реалізація планувальника персональних завдань ”

Обсяг пояснювальної записки: 75 аркушів.

Дата закінчення роботи: 9 червня 2025 р.

Підпис студента _____