Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Системотехніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Розробка підсистеми аналітики для вебзастосунку керування проєктами та завданнями
(тема)

Виконав:
студент *2* курсу, групи *ІТПм-22-1*
Базарбаєв О.Ш.
(прізвище, ініціали)

Спеціальність *122 Комп'ютерні науки*
(код і повна назва спеціальності)
Тип програми освітньо-професійна

Освітня програма «*Інформаційні технології проектування*»
(повна назва освітньої програми)

Керівник *проф. Калита Н.І.*
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри СТ _____ *Гребеннік І.В.*
(підпис) (прізвище, ініціали)

2024 р.

Я, як студент ХНУРЕ, розумію і підтримую політику закладу із академічної доброчесності. Я не надавав і не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

Дата: 15.01.2024 

Кваліфікаційна робота не містить відомостей заборонених до відкритого опублікування.

Кваліфікаційна робота виконана у відповідності до стандартів, що діють в Україні.

Попередній захист проведено 15 січня 2024 р.

Керівник кваліфікаційної роботи  *проф. Калита Н.І.*

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Системотехніки

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Інформаційні технології проектування
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Комп'ютерні науки
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри СТ проф. Гребеннік І.В.
(підпис)

«_____»_____2024 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Базарбаєву Олегу Шухратовичу
(прізвище, ім'я, по батькові)

1. Тема роботи *Розробка підсистеми аналітики для вебзастосунку керування проєктами та завданнями*
затверджена наказом по університету від *20.11 2023* р. *№ 1373Ст*
2. Термін подання студентом роботи до екзаменаційної комісії *10 січня 2024 р.*
3. Вихідні дані до роботи *Методи та засоби управління проєктами та завданнями. Функція: аналіз даних по проєктах з подальшою візуалізацією.* Перелік використовуваних програмних засобів*: ОС Windows 10, СУБД Microsoft SQL Server, IDE Visual Studio 2022. Мови програмування: C#.*

4. Перелік питань, що потрібно опрацювати в роботі *4.1 Вступ. 4.2 Аналіз предметної області. 4.2.1 Опис сучасного стану розвитку інформаційних систем та технологій. 4.2.2 Аналіз застосування досліджуваних технологій в існуючих системах. 4.3 Постановка задачі на дослідження. 4.3.1 Мета дослідження. 4.3.2 Постановка задачі. 4.4 Дослідження методів вирішення задачі. 4.4.1 Обґрунтування вибору методології розробки програмного забезпечення. 4.4.1.1 Waterfall методологія. 4.4.1.2 Agile методологія. 4.4.1.3 Пояснення обраної методології. 4.4.2 Важливість візуалізації даних. 4.4.3 Алгоритми прогнозування. 4.4.3.1 Метод ковзної середньої. 4.4.3.2 Експоненційне згладжування. 4.4.3.3 Модель ARIMA. 4.4.3.4 Методи машинного навчання. 4.4.3.4 Обґрунтування вибору методу прогнозування. 4.5 Опис прийнятих проектних рішень. 4.5.1 Обґрунтування вибору мови програмування. 4.5.2 Обґрунтування вибору платформи СУБД. 4.5.3 Опис архітектури розробленої системи. 4.5.4 Розробка інтерфейсу системи. 4.6 Висновки. 4.7 Перелік джерел посилання. 4.8 Додаток А Графічний матеріал. 4.8 Додаток Б Текст програми.*

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) waterfall методологія, статистика обробки та засвоєння інформації, вплив візуалізації інформації у бізнесі, порівняння .net 5 з іншими мовами програмування, форма логіну, форма реєстрації, форма створення нової компанії, налаштування проєкту, форма створення нового завдання, список завдань та детальна інформація про завдання, редагування завдання, канбан-дошка, діаграма ганта, діаграма виконаних завдань, діаграма прогнозування, табличне представлення

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1.	*Отримання завдання атестаційної роботи*	*16.10.2023*	
2.	*Огляд завдання та літератури*	*16.11 – 17.11.2023*	
3.	*Аналіз досліджуваних технологій в існуючих системах*	*18.11 – 21.11.2023*	
4.	*Постановка задачі на дослідження*	*22.11 – 23.11.2023*	
5.	*Дослідження методологій розробки програмного забезпечення*	*24.11 – 26.11.2023*	
6.	*Дослідження методів прогнозування*	*27.11 – 29.11.2023*	
7.	*Вибір технологій для розробки вебзастосунку*	*30.11.2023*	
8.	*Розробка вебзастосунку*	*01.12 – 30.12.2023*	
9.	*Оформлення пояснювальної записки та програмної документації*	*01.01 – 09.01.2024*	
10.	*Представлення на рецензування*	*10.01.2024*	
11.	*Представлення роботи в ЕК*	*12.01.2024*	

Дата видачі завдання *16.10.2023* р.

Студент _____ Базарбаєв О.Ш.
(підпис)

Керівник роботи _____ проф. Калита Н.І.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до магістерської кваліфікаційної роботи: 81 с., 29 рис., 2 додатки, 26 джерел інформації.

УПРАВЛІННЯ ПРОЄКТАМИ, МЕТОДОЛОГІЇ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ, AGILE, KANBAN, ПІДСИСТЕМА АНАЛІТИКИ, ДОШКА ПРОЄКТУ, МЕТОДИ ПРОНОЗУВАННЯ, ІНТЕРФЕЙС ДОСТУПУ

Об’єктом дослідження є підсистема аналітики вебзастосунку управління проєктами та завданнями. Особлива увага приділяється аспектам збору даних, їх аналізу та візуалізації для забезпечення ефективного моніторингу і прийняття управлінських рішень.

Предметом дослідження є алгоритми обробки даних, методи аналітики та програмні засоби для створення інтегрованої підсистеми аналітики в контексті вебзастосунку управління проєктами.

Мета дослідження полягає у розробці та інтеграції підсистеми аналітики, яка надаватиме змогу користувачам вебзастосунку аналізувати великі обсяги інформації про хід виконання проєктів та ефективність командної роботи.

Методи дослідження – системний аналіз, управління проєктами, методи прогнозування, методи аналізу даних, мова програмування C# платформа ASP.NET Core, фреймворки для розробки вебінтерфейсів, СУБД MS SQL Server, технології зберігання даних, засоби візуалізації даних.

Результатом роботи є підсистема аналітики для вебзастосунку управління проєктами та завданнями.

ABSTRACT

Master’s Thesis: 81 pages, 29 figures, 2 appendices, 26 title.

PROJECT MANAGEMENT, SOFTWARE DEVELOPMENT METHODOLOGY, AGILE, KANBAN, ANALYTICS SUBSYSTEM, PROJECT BOARD, PROPOSITION METHODS, ACCESS INTERFACE

The research object is the analytics subsystem of the web application for project and task management. Special attention is paid to aspects of data collection, analysis, and visualization to ensure effective monitoring and management decision-making.

The subject of the research are data processing algorithms, analytics methods, and software tools for creating an integrated analytics subsystem in the context of a web application for project management.

The purpose of the research is to develop and integrate an analytics subsystem that will allow users of the web application to analyze large volumes of information about the progress of projects and the effectiveness of team work.

Research methods include system analysis, project management, forecasting methods, data analysis methods, the C# programming language, ASP.NET Core platform, web interface development frameworks, MS SQL Server database management system, data storage technologies, and data visualization tools.

The result of the work is an analytics subsystem for the web application for project and task management.The object of study is the analytics subsystem of the web application for project and task management. Special attention is paid to aspects of data collection, analysis, and visualization to ensure effective monitoring and management decision-making.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД – база даних;

ІС – інформаційна система;

СУБД – система управління базами даних.

ARIMA – Авторегресійне Інтегроване Ковзне Середнє

EF Core – Entity Framework Core

ВСТУП

Сучасний світ переживає швидкі технологічні зміни, що суттєво впливають на спосіб, яким ми працюємо та управляємо проектами. Інформаційні технології та системи стали необхідними інструментами в управлінні завданнями та проектами. Все це призвело до активного розвитку систем управління завданнями та проектами, які надають організаціям ефективні засоби для досягнення своїх цілей.

Актуальність теми роботи полягає в тому, що сучасні організації повинні ефективно управляти своїми проектами та завданнями для досягнення успіху в умовах постійних змін та конкуренції. В сучасному світі, де технології стрімко розвиваються, системи управління проектами стають необхідною ланкою для успішної реалізації завдань. Проте, деякі існуючі технології вже демонструють ознаки застаріння, навіть у порівнянні з відносно новими рішеннями.

Однією з найпоширеніших систем управління проектами є Jira [1], яка довгий час служила як стандарт для багатьох організацій. Однак, з плином часу, деякі користувачі відзначають її складність та велику кількість налаштувань, що може ускладнити роботу невеликим командам. Крім того, Jira нерідко використовується великими підприємствами, де вона може виявитися занадто гнучкою для деяких проектів.

З іншого боку, інструмент Trello [2] визначається своєю простотою та інтуїтивністю, але він може виявитися недостатньо ефективним для складних завдань чи великих команд. Його надмірна простота може обмежувати можливості аналізу та деталізації завдань, особливо в умовах, коли необхідна більш глибока аналітика та інструменти для аналізу.

Зокрема, традиційні методології управління проектами, такі як Waterfall, можуть зазнавати критики через їхню жорсткість та відсутність гнучкості у змінних умовах. З іншого боку, Agile, зокрема методології Scrum і Kanban, визначаються гнучкістю та ітеративністю, що робить їх більш адаптованими до змін і вимог ринку.

Все це свідчить про те, що швидкий розвиток технологій та зміни в управлінських підходах підштовхують до переосмислення сучасних систем управління проектами, наголошуючи на потребі більш гнучких та інтегрованих рішень.

Темою цієї роботи є «Розробка підсистеми аналітики для вебзастосунку керування проєктами та завданнями». Головна проблема, яка вирішується, полягає в необхідності надати організаціям засіб, який дозволить ефективно планувати, виконувати та контролювати різноманітні проекти та завдання.

Об’єктом дослідження є сам процес управління завданнями та проектами в організаціях.

Предмет дослідження – підсистема аналітики для вебзастосунку, яка надає можливості для виконання цих процесів.

Метою даної роботи є розробка підсистеми аналітики для вебзастосунку, який буде забезпечувати користувачів засобами для управління завданнями та проектами, спрощуючи процеси планування, моніторингу та аналізу.

Впровадження такої системи сприяє покращенню методів управління завданнями та проектами у сучасному бізнес-середовищі, надавши організаціям ефективний інструмент для досягнення їх цілей та оптимізації бізнес-процесів.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Опис сучасного стану розвитку інформаційних систем та технологій

Сучасний стан розвитку інформаційних систем та технологій у сфері управління завданнями та проектами є динамічним і швидко зростає. Завдяки швидкому технологічному прогресу, розробники і фахівці в цій галузі мають доступ до різноманітних інструментів та платформ для ефективного керування завданнями та проектами. Існують як універсальні, так і спеціалізовані програмні рішення, які пропонують широкий спектр функцій, включаючи планування, контроль, моніторинг, звітність та аналітику.

Такі інформаційні системи повинні бути спрямовані на широкий спектр користувачів і підприємств, забезпечуючи їм можливість ефективно керувати різноманітними проектами, незалежно від їх обсягу та складності. А також надавати можливість спільної роботи для команд, що дозволить співробітникам спілкуватися, обмінюватися інформацією та спільно працювати над проектами в режимі реального часу.

Одним з ключових напрямків розвитку є вдосконалення інтерфейсів користувача та забезпечення їх інтуїтивно зрозумілим дизайном, що сприяє зручності в роботі з системою для широкого кола користувачів. Також звертається увага на забезпечення безпеки даних і захисту конфіденційної інформації, оскільки це стає дедалі більш важливим аспектом для організацій у сучасному цифровому середовищі.

Актуальність даного напрямку розвитку технологій управління завданнями та проектами надзвичайно важлива в контексті сучасного бізнесу. Завдяки надходженню великої кількості даних і великому обсягу завдань та проектів, організаціям потрібні ефективні інструменти для їх керування та моніторингу. Технології управління завданнями надають підприємствам можливість бути більш організованими, ефективними і конкурентоздатними,

допомагаючи вчасно завершувати проекти і досягати поставлених цілей. До цього додається необхідність забезпечення співпраці між командами та комунікації на відстані, що стає дедалі важливішою через глобалізацію та роботу з віддаленими співробітниками. Тому, розвиток та впровадження сучасних систем управління завданнями та проектами залишається актуальним завданням для організацій, які прагнуть підтримувати конкурентоспроможність та ефективність у сучасному бізнес-середовищі.

1.2 Аналіз застосування досліджуваних технологій в існуючих системах

В рамках дослідження були обрані та вивчені дві популярні платформи для управління завданнями та проектами: Jira [1] і Trello [2]. Для кращого розуміння їх застосування та функціоналу, були проведені докладні аналізи та вивчення можливостей, які ці системи пропонують.

Jira є високопродуктивною системою, розробленою для керування проектами та завданнями. Ця платформа надає широкий спектр можливостей для організації та контролю проектів у реальному часі. Jira пропонує різноманітні функції для полегшення організації та виконання проектних завдань. Ось перелік деяких з них:

- створення завдань: можливість швидкого створення нових завдань для проектів або задач, які потребують уваги;
- відстеження завдань: механізм для відстеження стану виконання завдань та їхнього прогресу;
- керування проектами: можливість створення та організація проектів з урахуванням різноманітних вимог та завдань;
- призначення завдань: функціонал для призначення завдань конкретним користувачам чи командам для відповідальності;
- спільна робота: засоби спільної роботи над завданнями та можливість обміну коментарями між користувачами;
- призначення пріоритетів: можливість встановлення пріоритетів для

завдань, що дозволяє виділити найважливіші завдання;

- інструменти аналізу: Jira дозволяє візуалізувати та аналізувати дані, включаючи створення графіків відстеження прогресу завдань та графіків звітності.

Користувачі можуть достатньо легко створювати нові завдання, визначати пріоритети, надавати їм опис та призначати відповідальних осіб (рис. 1.1).

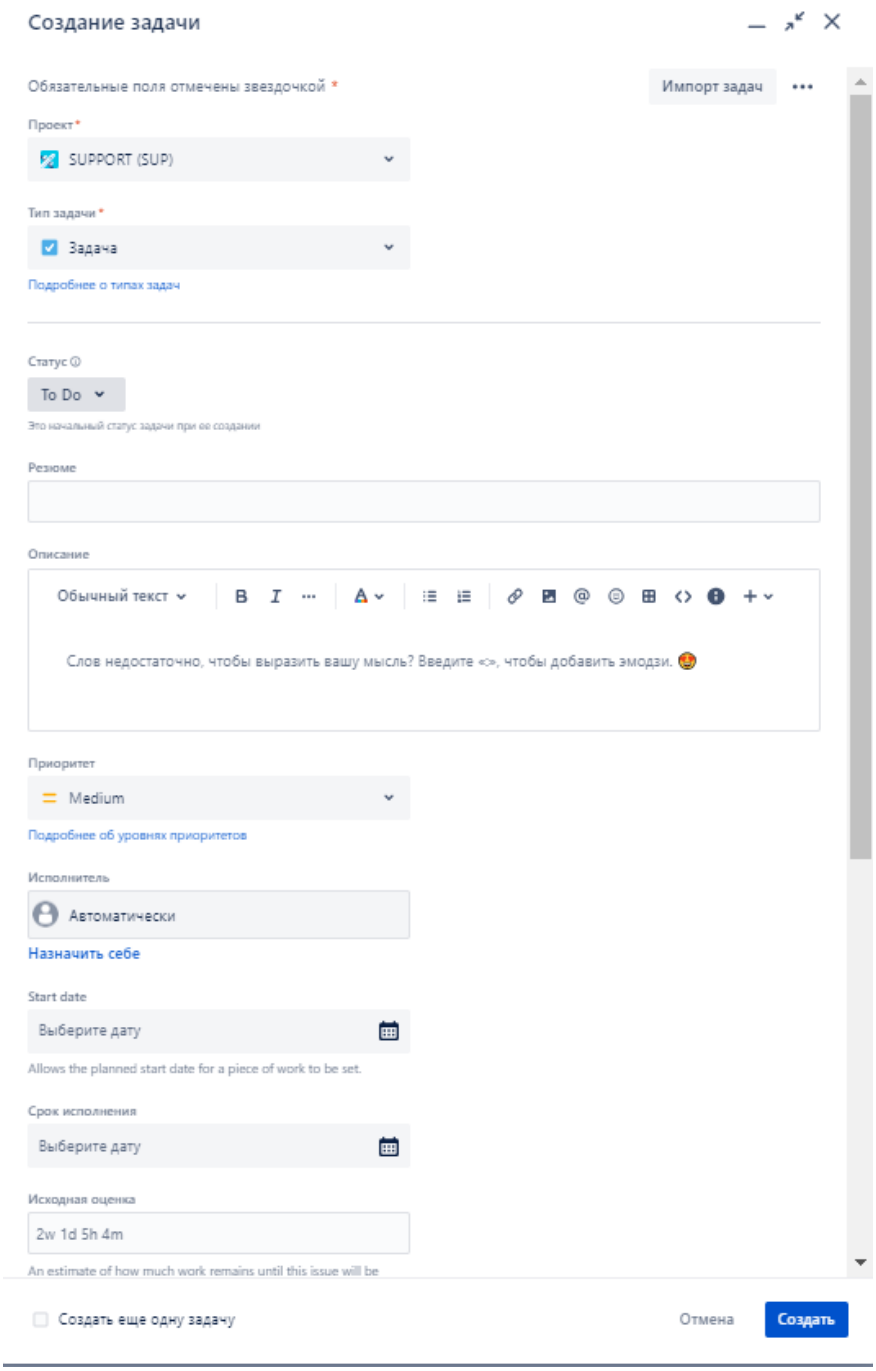


Рисунок 1.1 – Створення задачі в Jira

Додатково до цього, Jira пропонує зручну систему керування ресурсами, що дозволяє ефективно розподіляти завдання між учасниками команди. Завдяки інтерактивним дошкам, спеціально налаштованим під кожен проект, користувачі можуть візуально відстежувати статус та прогрес кожного завдання. Приклад дошки проекту показаний на рисунку 1.2.

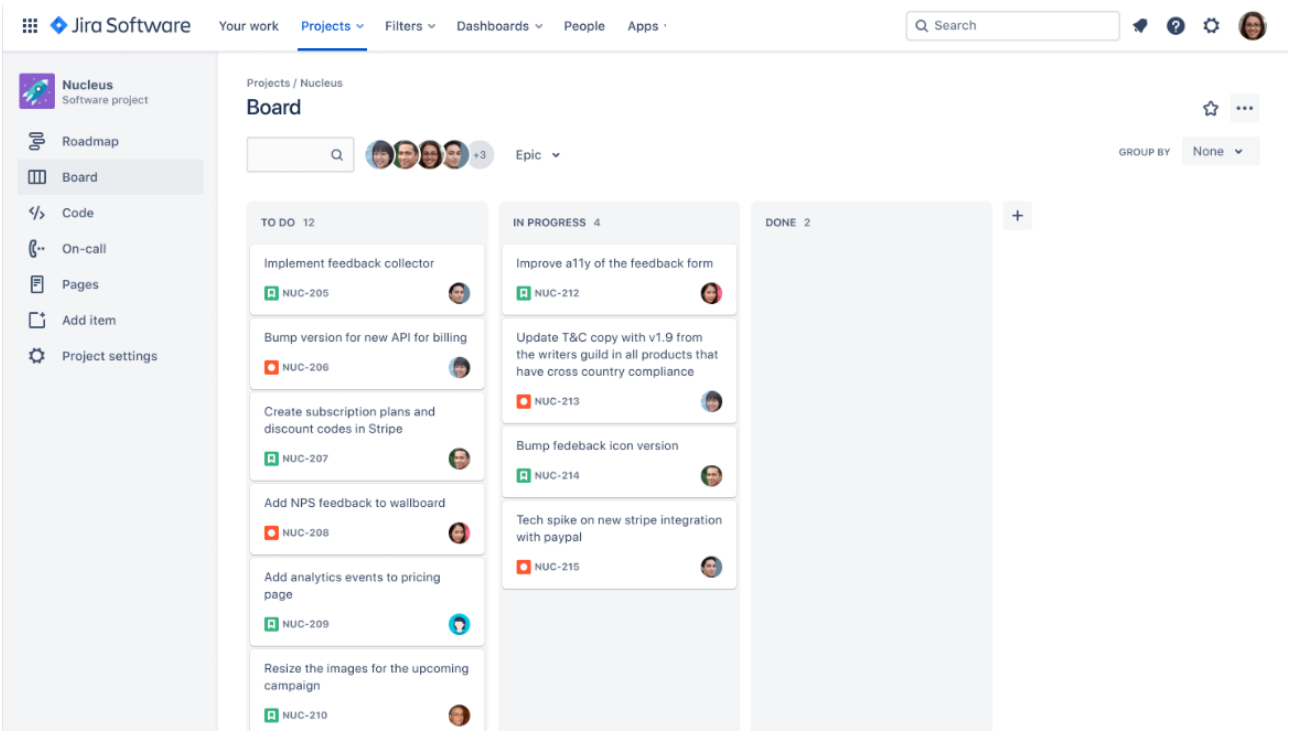


Рисунок 1.2 – Дошка проекту в Jira

Ще однією корисною функцією є можливість створення діаграм Ганта для моніторингу та планування проектів. Діаграми Ганта дозволяють візуалізувати послідовність завдань та зв'язки між ними, що сприяє кращому управлінню проектами та вчасному виявленню можливих затримок. Візуалізація діаграми Ганта представлена на рисунку 1.3.

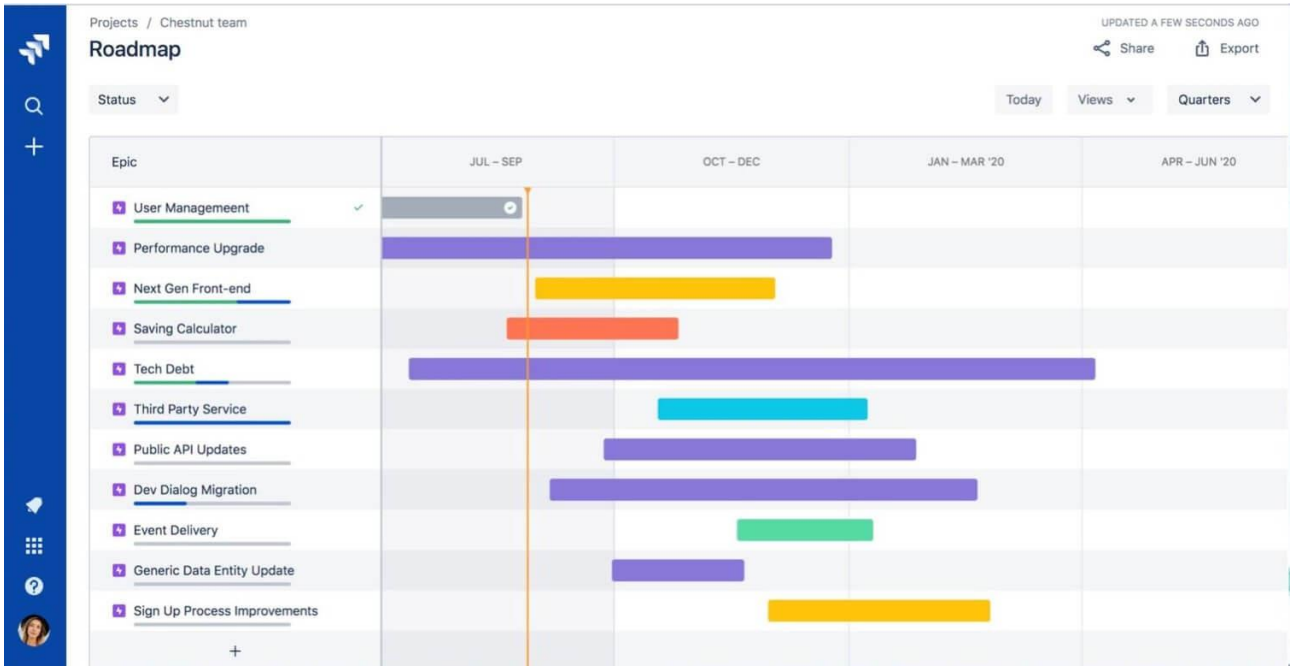


Рисунок 1.3 – Діаграма Ганта в Jira

Проте, важливо відзначити недоліки детальних діаграм Ганта в Jira (рис. 1.4), серед яких може бути нагромадження інформації у великих проектах. З великою кількістю завдань та складними залежностями діаграма може стати складною для читання та перевантаженою, що ускладнює її використання та зрозуміння. На практиці, для менших проектів чи для команд, що використовують більш гнучкі методології управління, детальні діаграми Ганта можуть виявитися зайвими, оскільки їхній рівень деталізації може перевищувати необхідність та витрати часу на їхнє створення та підтримку. У таких випадках, використання інших інструментів, таких як інтерактивні дошки чи розширені аналітичні інструменти, може бути більш ефективним підходом до керування проектами.

Рисунок 1.4 – Приклад перенавантаженої детальної діаграми Ганта

На рисунку 1.5 відображено процес адміністрування користувачів та ролей. Jira надає адміністраторам широкий спектр можливостей для налаштування доступу користувачів до системи та визначення їх ролей та повноважень. Адміністратори можуть додавати нових користувачів, встановлювати їх ролі, призначати права доступу та надавати дозволи на виконання конкретних дій в системі. Цей процес дозволяє ефективно управляти робочими процесами та забезпечувати безпеку даних, забезпечуючи кожному

користувачу відповідність з його обов'язками та функціональними потребами в межах проекту.

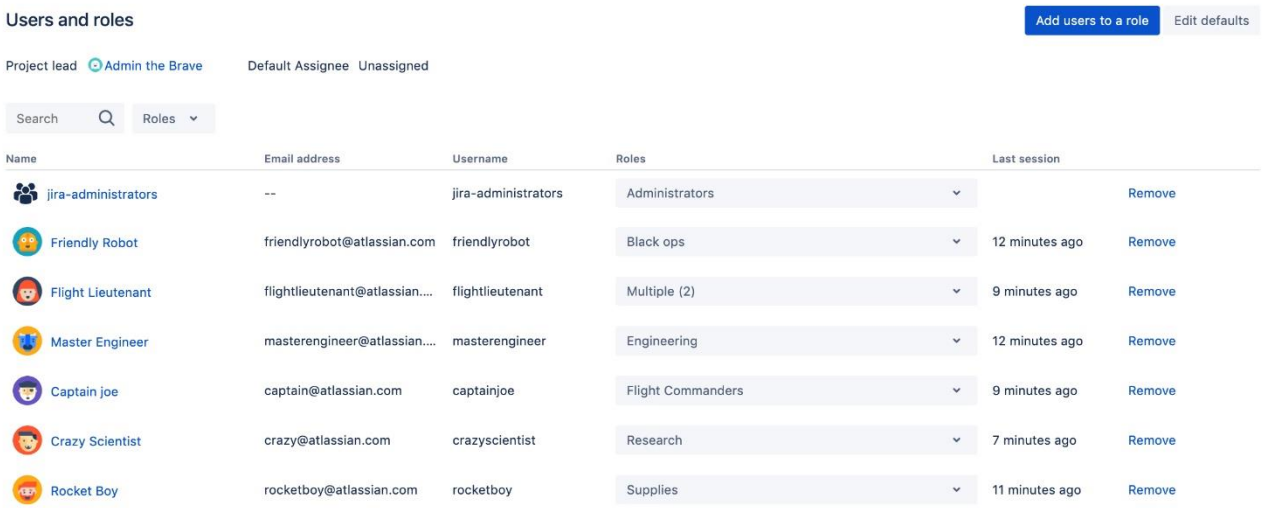


Рисунок 1.5 – Адміністрування користувачів в Jira

Загалом, Jira є потужним інструментом для управління проектами, що надає користувачам можливість ефективно організовувати свою роботу, відстежувати прогрес та досягати поставлених цілей. Вона підходить як для невеликих команд, так і для великих підприємств, що потребують комплексного управління різноманітними проектами та завданнями.

Trello - це інтуїтивно зрозуміла платформа для управління завданнями та проектами, що надає користувачам зручний інструмент для спільної роботи та організації завдань. Trello базується на концепції "дошок", де кожна дошка представляє собою окремий проект або завдання, а картки відображають конкретні завдання або етапи. Картки можна легко переміщувати між колонками, відображаючи різні стадії їх виконання. На рисунку 1.6 представлений приклад дошки, її колонок та завдань, а також продемонстровано переміщення карточки з одного статусу в інший.

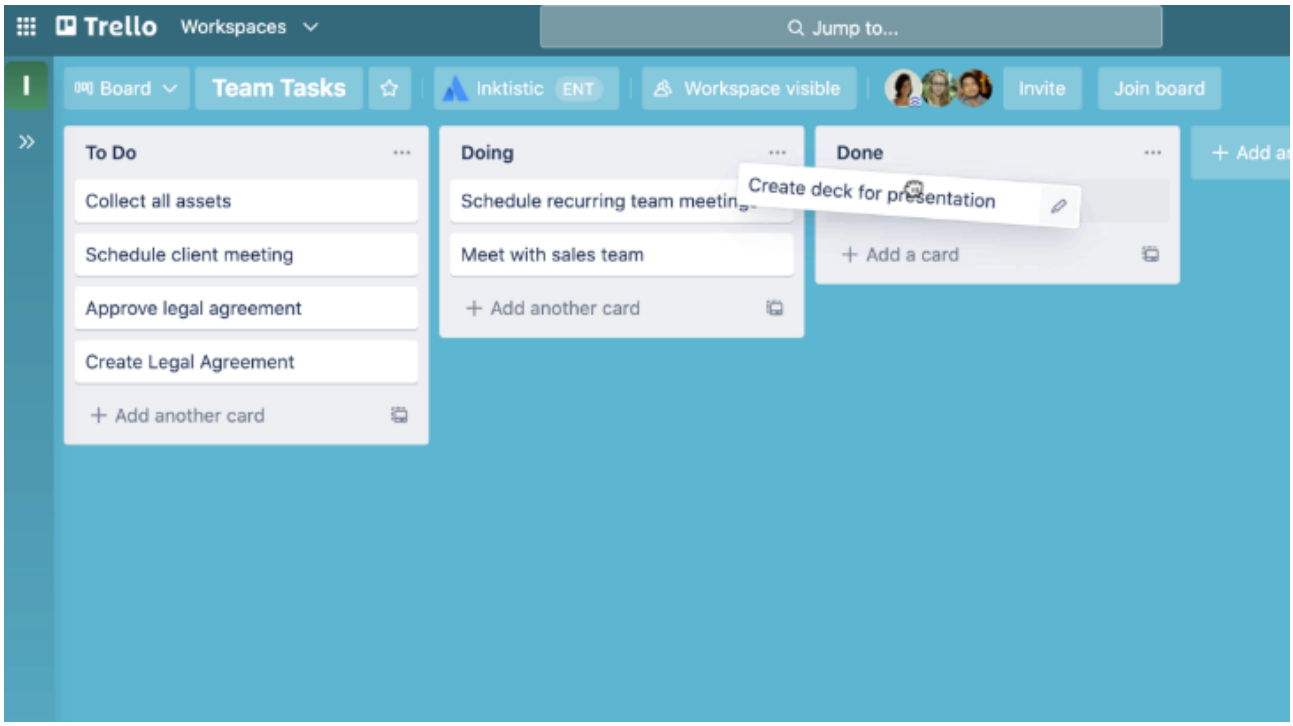


Рисунок 1.6 – Дошка проекту в Trello

Далі на рисунку 1.7 видно процес створення нового завдання на дошці в Trello. Після того, як користувач увійшов до свого проекту, він може натиснути на кнопку «Створити нову картку». Потім відкриється вікно для додавання назви завдання та додаткової інформації, як опис, додаткові файли, теги тощо. Це дозволяє користувачеві докладно описати завдання та визначити його деталі.

Користувач може визначити, які учасники команди відповідають за конкретні завдання та проекти. Він може призначати користувачам ролі, визначати їх рівень доступу, а також відповідальності (рис. 1.8). Це важливо для організації та координації робочих процесів.

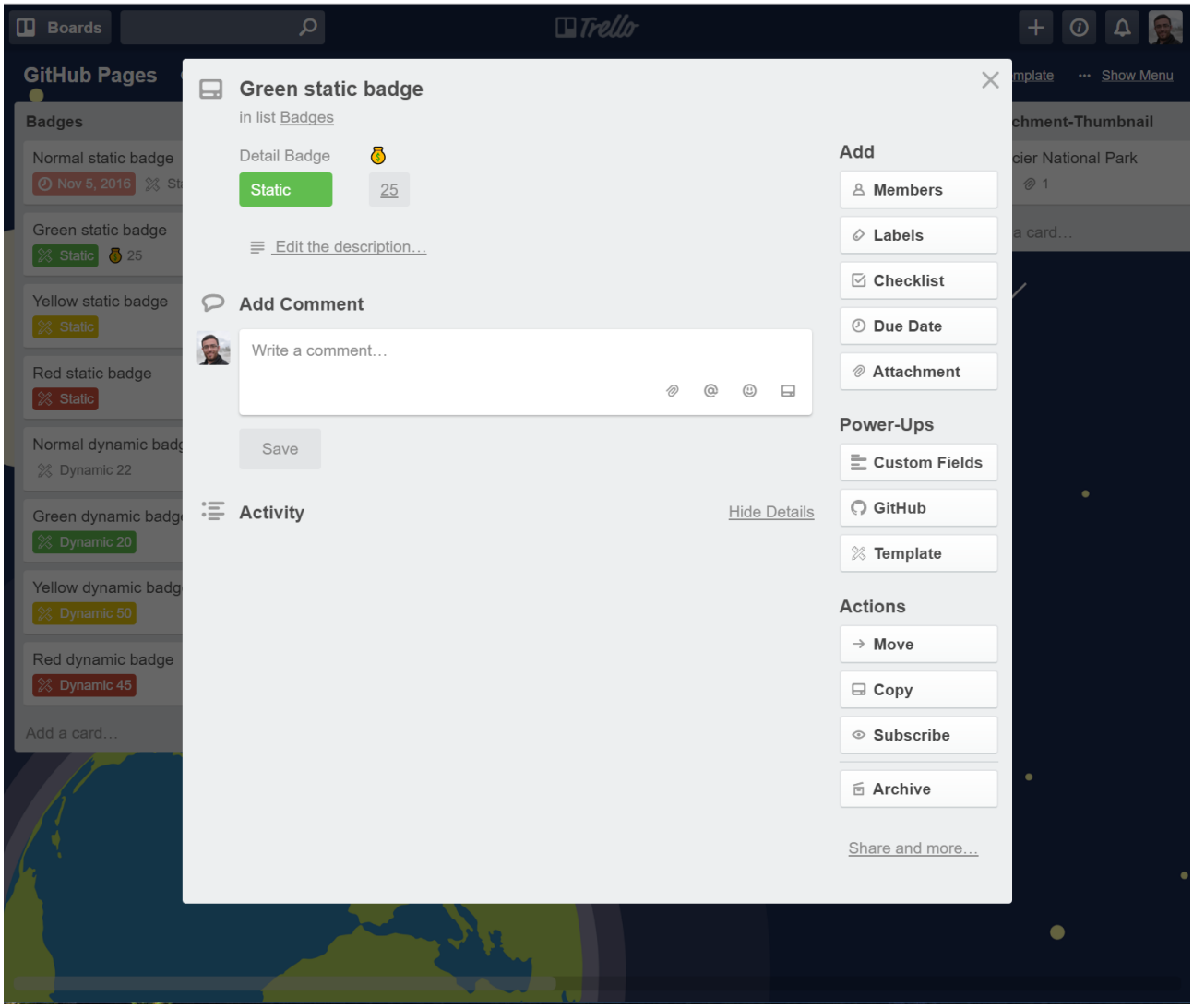


Рисунок 1.7 – Створення завдань на дошці в Trello

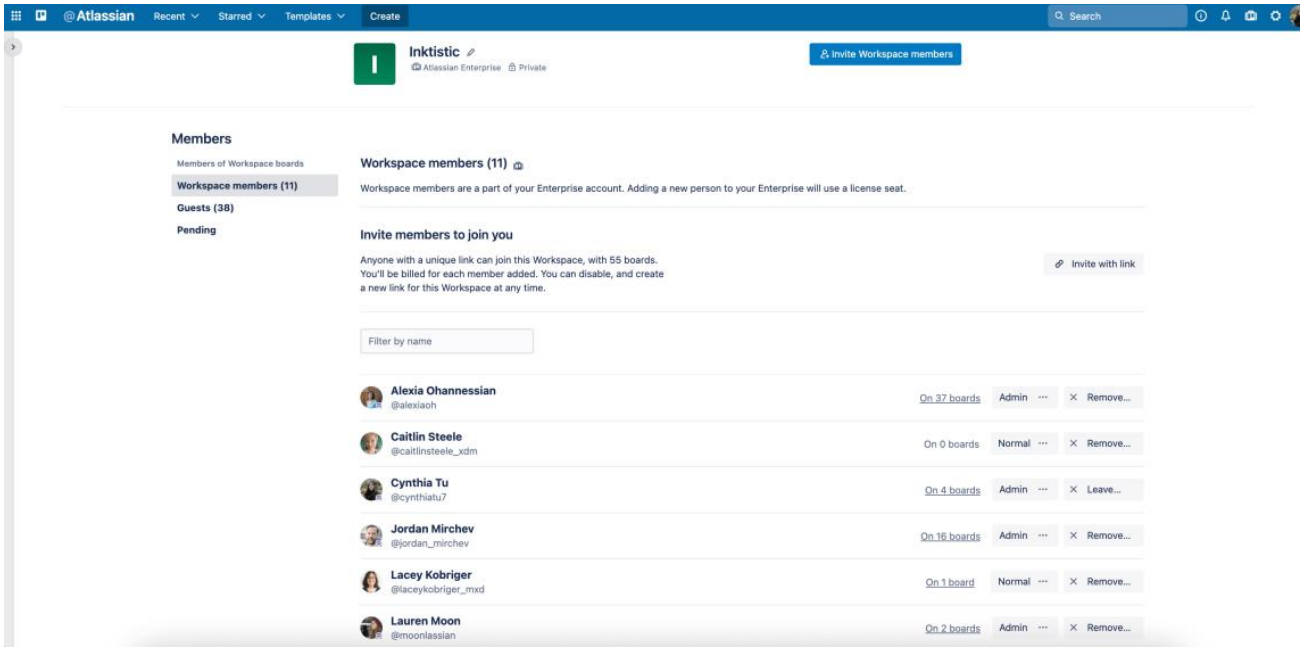


Рисунок 1.8 – Адміністрування користувачів та ролей в Trello

Отже, підсумуючи, актуальність використання Trello полягає в її здатності забезпечувати просте та ефективне управління завданнями і проектами в різних галузях та для різних видів проектів. Trello підходить для невеликих команд і проектів, які потребують зручного інструмента для спільної роботи та організації завдань. Вона дозволяє користувачам використовувати простий інструмент для керування робочими процесами та досягнення результатів.

Переваги Trello:

- простота та інтуїтивний інтерфейс: Trello славиться своєю легкістю використання та інтуїтивно зрозумілим інтерфейсом. Користувачі швидко можуть оволодіти системою та отримати зручний і швидкий доступ до своїх завдань;
- гнучкість в організації: система дошок, списків і карток у Trello дозволяє користувачам гнучко налаштовувати та адаптувати свій робочий простір, адаптуючи його до конкретних потреб команди чи проекту;
- легкість спільної роботи: Trello пропонує простий механізм для спільної роботи команд, дозволяючи додавати коментарі, визначати відповідальних і використовувати картки для обговорення завдань;
- зручність для невеликих проектів: для невеликих команд та проектів, Trello може виявитися ефективним, не завантажуючи зайвою функціональністю.

Недоліки Trello:

- відсутність вбудованих інструментів аналізу: однією з головних недоліків Trello є відсутність вбудованих інструментів для складнішого аналізу та звітності. Для докладного вивчення прогресу проекту чи робочих процесів, користувачам може доводитися використовувати зовнішні інструменти;
- обмеженість в робочих процесах: для більш складних проектів, особливо тих, які вимагають строгого керування часом та завданнями, може відчуватися обмеженість можливостей Trello;
- несумісність для великих проектів: для великих команд та складних проектів, Trello може виявитися недостатньою для задоволення потреб у складному керуванні та аналізі.

У свою чергу Jira - це потужний інструмент для керування проектами і задачами, який надає широкий функціонал для команд розробників та проектних груп. Ось кілька переваг та недоліків Jira, враховуючи різні аспекти користування:

Переваги Jira:

- розширені можливості керування проектами: Jira надає розширені інструменти для створення, відстеження та оцінювання завдань і проектів;
- гнучкість налаштування: Jira дозволяє налаштовувати робочі процеси та відстежувати проекти відповідно до унікальних потреб команди;
- широкі можливості аналізу: інтегровані інструменти аналітики дозволяють вам стежити за прогресом проектів, виділяти тенденції та приймати рішення на основі даних.

Недоліки Jira:

- складність в освоєнні: у зв'язку з великою кількістю функцій Jira може виявитися складним для користувачів, члени команди можуть відчувати необхідність в додатковому навчанні;
- перевантаженість інтерфейсу: інтерфейс Jira може бути перевантаженим через обширну кількість опцій та меню, що може призвести до меншої ефективності користувача;
- складність адміністрування: налаштування та адміністрування Jira може вимагати певних навичок, особливо в разі великих проектів;
- перенавантаженість інструментів аналізу: інструменти аналізу та звітності в Jira можуть бути перенавантаженими, особливо для користувачів, які не є експертами в області управління проектами. Спроби створити складні звіти чи діаграми можуть потребувати додаткового часу та зусиль, іноді ускладнюючи взаємодію з інтерфейсом та затримуючи процес прийняття рішень.

Отже, попри наявність потужних інструментів для контролю та аналізу проектів і завдань, існує очевидна потреба у їх удосконаленні. Актуальним завданням є надання IT-командам вдосконаленого інструменту для відстеження прогресу проектів, який би також оснащував їх функціями аналітики.

Враховуючи ці вимоги, потрібно зосередити зусилля на тому, щоб максимально використати переваги існуючих систем, одночасно пом'якшуючи їхні недоліки та недосконалості, щоб забезпечити ефективне та результативне управління проектами.

2 ПОСТАНОВКА ЗАДАЧІ НА ДОСЛІДЖЕННЯ

2.1 Мета дослідження

Мета даної роботи полягає в дослідженні інструментів аналітики та подання аналітичної інформації в системах керування проєктами та завданнями з метою поліпшення їх управління. Головною метою є надання ефективного та гнучкого інструмента для організації, відстеження та виконання завдань, а також керування проектами в реальному часі.

Ця розробка має наступні ключові цілі та завдання:

- забезпечення ефективного управління завданнями: додаток дозволить користувачам з легкістю створювати, відстежувати та оновлювати завдання. Він спростить процес призначення завдань та розподілу роботи між членами команди;
- покращення координації проектів: додаток надасть зручні інструменти для створення, керування та відстеження проектів. Він спростить планування та ресурсне управління;
- підвищення ефективності робочого процесу: вебзастосунок дозволить команді зосередитися на важливих завданнях, уникнути перевантаження та підвищити продуктивність завдяки оптимізації робочого процесу;
- забезпечення доступності та зручного користування: метою дослідження є створення додатку з інтуїтивним інтерфейсом, який буде доступний для широкого кола користувачів;
- вирішення проблеми неефективного управління завданнями та проектами: вебзастосунок створюється для вирішення проблем та недоліків існуючих методів управління завданнями та проектами.

В цілому, мета дослідження полягає в створенні інструменту, який допоможе організаціям та командам управляти завданнями та проектами більш ефективно, забезпечуючи гнучкість, зручність та оптимізацію робочого процесу.

2.2 Постановка задачі

Потрібно описати необхідні вхідні дані для розробки підсистеми аналітики для вебзастосунку керування проєктами та завданнями. Ці дані визначають контекст та специфікації проекту, а також відображають вимоги та потреби користувачів.

Функціональні вимоги до вебзастосунку:

– створення та відстеження завдань, а саме: створення нових завдань, встановленння їх статусів, термінів виконання та пріоритетів;

– управління проектами, а саме: створення проектів, призначення завдань до проектів та керування ресурсами;

– звітність та аналітика: генерація звітів, аналіз статистики завдань та проектів, а також моніторинг продуктивністі команди;

– керування користувачами та ролями: функціонал управління правами доступу користувачів з різними ролями.

Отже результатом дослідження методів та технологій управління проектами та завданнями буде підсистеми аналітики для вебзастосунку керування проєктами та завданнями. Ця підсистема призначена для використання як малими, так і більшими командами. Також вона повинна виконувати декілька додаткових умов:

– вимоги до безпеки: забезпечить систему автентифікації та авторизації для обмеження доступу до конфіденційної інформації. Забезпечить шифрування даних для захисту їх від несанкціонованого доступу;

– масштабованість: забезпечить можливість розширення функціональності та впровадження нових функцій для відповіді на зростання потреб користувачів. Оптимізує продуктивність для ефективної роботи з

великою кількістю даних та користувачів.

Отже, в рамках розробки підсистеми аналітики для вебзастосунку керування проєктами та завданнями, основним результатом має стати реалізація інструменту візуалізації даних, який надаватиме можливість представляти інформацію у вигляді зрозумілих і зручних для сприйняття діаграм. Це дозволить користувачам легко інтерпретувати складні дані та здійснювати швидкий аналіз продуктивності завдань та проектів. Також необхідно розробити алгоритм прогнозування для вирішення задачі передбачення тенденцій та результативності команд в майбутньому на основі попередніх даних, забезпечуючи можливість планування та оптимізації ресурсів. Візуалізація цього алгоритму надасть користувачам інтуїтивно зрозумілі інструменти для підтримки рішень у сфері управління проєктами.

3 ДОСЛІДЖЕННЯ МЕТОДІВ ВИРІШЕННЯ ЗАДАЧІ

3.1 Обґрунтування вибору методології розробки програмного забезпечення

Методологія розробки програмного забезпечення визначає стратегію та підхід, які використовуються для організації та виконання процесу розробки програмного продукту. У цьому розділі ми проводимо аналіз різних методологій розробки, зосереджуючись на традиційних методах, таких як Waterfall, та більш гнучким Agile підходом, що є широко використовуваними у сучасній індустрії програмного забезпечення.

Актуальність дослідження полягає в тому, що вибір відповідної методології впливає на ефективність, продуктивність та якість розробки програмного забезпечення. У контексті нашого проекту, який передбачає створення вебзастосунку на платформі ASP.NET Core для управління завданнями та проектами, правильний вибір методології має критичне значення для успішної реалізації цієї системи.

У цьому розділі пропонується огляд кожної методології, що включає її основні принципи, переваги та недоліки, а також сферу застосування. Детальний аналіз кожного підходу допоможе визначити найбільш підходящий для наших потреб в рамках розробки вебзастосунку для управління завданнями та проектами.

Важливо глибоко розуміти особливості різних методологій розробки програмного забезпечення та їх відповідність конкретним вимогам проекту. Порівняння та аналіз кожної методології допоможуть зрозуміти, як кожен підхід може вплинути на успішність нашого проекту.

3.1.1 Waterfall методологія

Waterfall методологія, також відома як «лінійний метод» розробки, є класичним підходом до управління проектами розробки програмного забезпечення. Ця методологія передбачає послідовний підхід до розробки, де кожна фаза проекту розглядається окремо та повинна бути завершена, перш ніж розпочнеться наступна. Waterfall методологія визначає фіксований порядок фаз, що допомагає впорядкувати та структурувати процес розробки [14]. На рисунку 2.1 представлено порядок фаз у методології Waterfall:

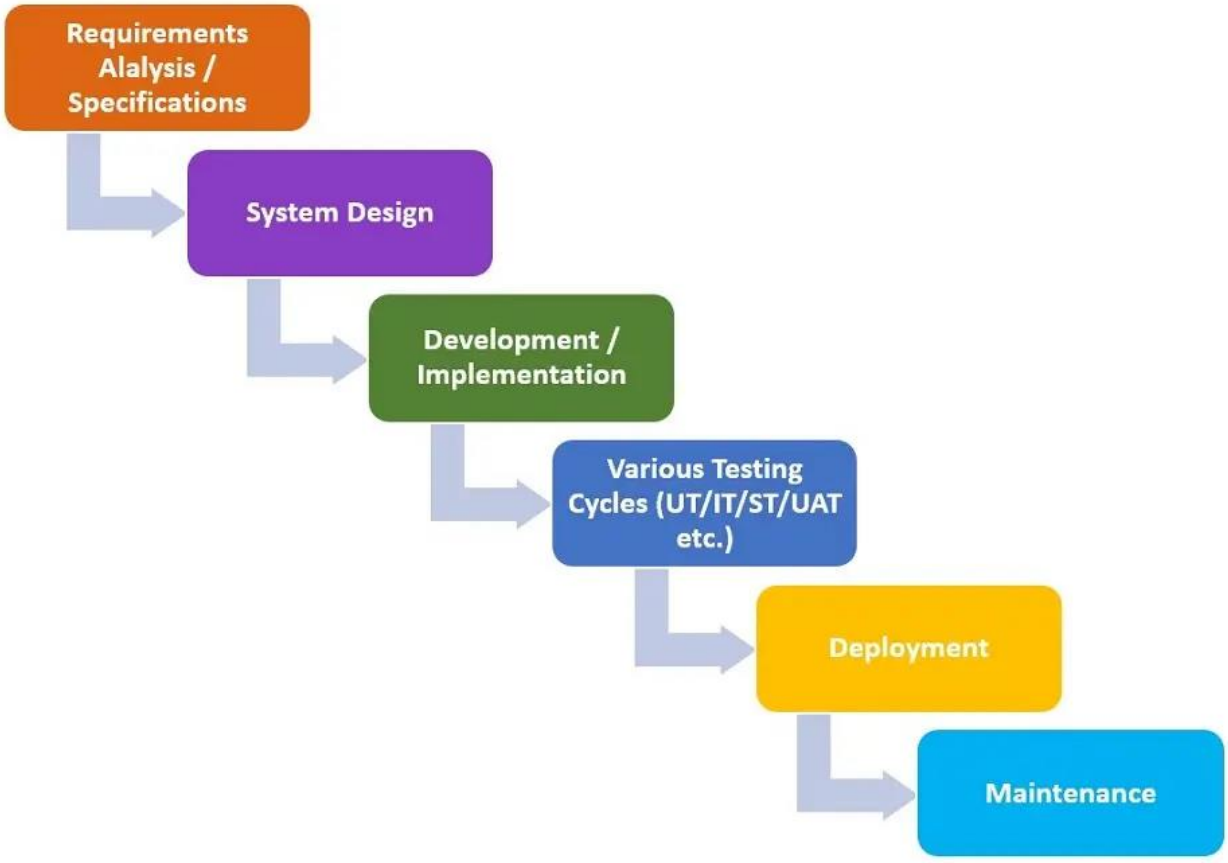


Рисунок 3.1 – Схематичне зображення методології Waterfall

Опис основних фаз даної методології:

- фаза аналізу (Requirements Analysis): збираються та аналізуються всі вимоги до програмного продукту. Це включає в себе вимоги користувачів, функціональні вимоги, технічні вимоги і т.д.;

- фаза проектування (System Design): визначається архітектура системи і розробляються деталізовані плани для реалізації. Визначення технічних деталей, проектування бази даних, створення діаграм, наприклад, UML;
- фаза реалізації (Implementation): розробляється програмний код відповідно до специфікацій та дизайну, розробленого на попередній стадії;
- фаза тестування (Testing): програма піддається вичерпному тестуванню для виявлення помилок та дефектів. Виконання різних видів тестів, включаючи модульні, інтеграційні, системні та приймальні тести;
- фаза впровадження (Deployment): програмний продукт встановлюється на реальному обладнанні чи серверах та готується до використання. Також може включати в себе навчання користувачів, підготовка до впровадження;
- фаза підтримки (Maintenance): після впровадження підтримка та покращення продукту забезпечуються протягом його життєвого циклу. Виправлення помилок, оновлення, забезпечення безпеки та розвиток продукту.

Waterfall методологія найбільше підходить для проектів з чітко визначеними та стабільними вимогами, де зміни під час розробки обмежені. Тепер розглянемо переваги і недоліки цього підходу.

Переваги даної методології:

- простота та структурованість: Waterfall надає чітку та структуровану послідовність фаз розробки, що полегшує планування та управління проектом;
- чіткість вимог: вимоги визначаються на ранніх стадіях, і вони залишаються стабільними протягом усього процесу розробки;
- легка контрольованість: через послідовний характер роботи, кожна фаза може бути легко керована та вимірювана;
- сфокусований тестування: тестування проводиться на завершальних стадіях, що дозволяє виявити помилки відразу після завершення розробки.

Недоліки методології:

- важкість змін: зміни вимог під час розробки можуть бути важко

реалізовані, оскільки кожна фаза повинна бути завершена перед переходом до наступної;

- високий ризик: якщо вимоги неправильно збагачуються на ранніх стадіях, це може призвести до великих проблем на пізніших етапах;
- відсутність реальних результатів рано в проекті: клієнт отримує продукт лише після завершення всіх фаз, що може призвести до незадоволеності чи недорозумінь.

3.1.2 Agile методологія

Agile методологія є гнучким та ітеративним підходом до розробки програмного забезпечення, що дозволяє забезпечити адаптивність та реагувати на зміни під час розробки проекту. Основною ідеєю Agile є співпраця, комунікація та ітеративна розробка, що спрямована на максимальне задоволення потреб клієнта [15]. Agile передбачає, що вимоги можуть змінюватися протягом процесу розробки, і реагування на ці зміни є ключовою властивістю методології.

Основні переваги Agile методології:

- гнучкість та адаптивність: Agile дозволяє легко вносити зміни в вимоги та пристосовуватися до змін в оточенні;
- постійна готовність до видачі функціонального продукту після кожної ітерації: Agile методологія сприяє постійному вдосконаленню та розвитку продукту, готового до використання після кожної ітерації;
- активне залучення клієнта: Agile сприяє регулярному взаємодія з клієнтом та отриманню зворотнього зв'язку;
- вдосконалення якості продукту: заснований на тестуванні та інтеграції процес дозволяє підвищити якість програмного забезпечення.

Agile методологія також має свої недоліки:

- потребує високої комунікації: Agile вимагає ефективної комунікації та взаємодії між командою розробки та клієнтом, що може бути вимогливим завданням;

- наявність досвіду та дисципліни: реалізація Agile може бути складною без досвіду та дисципліни в команді розробки;
- недостатня документація: Agile спрямований на результат та співпрацю, і може бути менше акценту на детальній документації.

Scrum є одним з найпопулярніших фреймворків Agile методології, який використовується для управління процесами розробки програмного забезпечення. Цей фреймворк дозволяє командам ефективно працювати над складними проектами, забезпечуючи гнучкий підхід до управління змінами та швидку адаптацію до нових вимог. Scrum покликаний сприяти збільшенню продуктивності та вдосконаленню комунікації в команді, а також забезпечити прозорість та гнучкість під час розробки програмного забезпечення.

Scrum передбачає роботу в рамках чітко визначених часових рамок, відомих як «спринти», що зазвичай тривають від одного до чотирьох тижнів. Кожен спринт починається плануванням, під час якого визначаються цілі спринту та список завдань, які мають бути виконані [15]. Протягом спринту команда зустрічається на щоденних «стендапах», де обговорюються прогрес та можливі перешкоди. На завершення спринту проводиться огляд, під час якого демонструється досягнутий результат, а також ретроспектива, під час якої команда аналізує свою роботу та визначає можливі шляхи вдосконалення.

Scrum базується на чітко визначених ролях, таких як Scrum Master, Product Owner та Розробник, які мають свої відповідальності та обов'язки у процесі розробки. Цей фреймворк дозволяє командам швидко реагувати на зміни, вдосконалювати продукт протягом кожного спринту та забезпечити ефективне управління проектом.

Kanban - це фреймворк Agile методології, який спрямований на вдосконалення управління робочим процесом та оптимізацію продуктивності команди. Головна ідея Kanban полягає в візуалізації робочого процесу за допомогою дошки (зазвичай фізичної або електронної), де кожне завдання або завдання представлені картками [15]. Цей фреймворк дозволяє команді бачити,

які завдання в робочому процесі, як вони рухаються від початку до завершення та які можливі перешкоди.

Kanban надає акцент на обмеженні робочого потоку, що допомагає збалансувати навантаження команди та уникнути перевантажень. У фреймворку Kanban немає фіксованих часових рамок, як у Scrum, і завдання обробляються в порядку їхнього надходження, але з дотриманням обмеження потоку.

Основні принципи Kanban включають:

- візуалізація робочого процесу: всі завдання відображаються на дошці, що дозволяє команді бачити, як просувається кожне завдання в процесі роботи;
- обмеження робочого потоку: встановлення лімітів на кількість завдань, які можуть знаходитися в конкретних стадіях робочого процесу, що допомагає збалансувати робоче навантаження;
- регуляція потоку роботи: завдання рухаються від одного етапу до іншого відповідно до лімітів, і команда працює над завданнями, які найбільше пріорітетні;
- зосередження на покращенні процесу: Kanban підтримує постійний аналіз та вдосконалення робочого процесу для забезпечення оптимальної продуктивності та якості;
- Kanban може бути використаний в різних галузях та для різних типів проектів, де контроль над робочим процесом і оптимізація продуктивності є ключовими завданнями.

3.2 Обгрунтування вибору методології проєктування

Обираючи Agile методологію з Kanban фреймворком для розробки програмного забезпечення, було виділено кілька важливих причин. Перш за все, варіант з використанням Scrum та спринтами не завжди відповідає реальним потребам багатьох проектів. Часто виявляється, що цей підхід стає формальністю, а не інструментом для забезпечення реальної користі для

проекту. Великі спринти та строго регламентований графік можуть обмежити гнучкість та адаптивність команди, що часто не є найкращим підходом для складних і змінних завдань.

З іншого боку, Kanban дозволяє команді зосередитися на конкретних завданнях, забезпечуючи більшу гнучкість та адаптивність. З використанням Kanban команда може працювати над завданнями у порядку їх пріорітетності, не обмежуючи себе строгими часовими обмеженнями спринтів. Це дозволяє швидше реагувати на зміни в вимогах та ринковому середовищі, що особливо важливо для проектів, де потрібна максимальна гнучкість.

Крім того, використання Kanban сприяє вдосконаленню робочого процесу та покращенню продуктивності команди. Картки на дошці візуалізують робочий потік та дозволяють ефективно балансувати завдання між членами команди.

Отже, обравши Agile з Kanban, враховуються потреби проекту в гнучкому та адаптивному підході, який сприяє максимізації продуктивності та забезпечує швидку реакцію на зміни.

3.3 Роль візуалізації даних у прийнятті рішень

У сучасному світі, керованим даними, підприємства і організації стикаються з проблемою вилучення значної інформації з великих обсягів даних. Аналіз та візуалізація даних відіграють ключову роль у перетворенні необроблених даних у практичні знання.

Візуалізація даних має значний вплив на продуктивність та процеси прийняття рішень у різноманітних бізнес-контекстах. Ефективність візуалізації даних є очевидною в декількох ключових аспектах:

– обробка та засвоєння інформації: людський мозок обробляє візуальну інформацію набагато швидше та ефективніше, ніж текст. Дослідження показують, що мозок може обробляти візуали в 60,000 разів швидше, ніж текст, і візуальна інформація становить близько 90% даних, що передаються нашому мозку. Люди зберігають 65% інформації через три дні після перегляду

зображення з даними порівняно з лише 10% інформації, яку вони чують (рис. 3.2). Це вказує на те, що візуалізація даних значно покращує засвоєння та розуміння інформації [6];

– ділові зустрічі та прийняття рішень: використання візуалізації даних може значно скоротити тривалість ділових зустрічей. Школа бізнесу Вортон з'ясувала, що організації можуть скоротити тривалість своїх зустрічей на 24%, використовуючи техніки візуалізації даних. Менеджери на 28% частіше знаходять своєчасну інформацію з інструментами візуального відновлення даних, ніж ті, хто покладається лише на звіти та інформаційні панелі (рис. 3.3). Крім того, компанії, що інтегрують візуалізацію даних у свою діяльність, в п'ять разів частіше здатні швидко приймати рішення [6];

– продажі та маркетинг: у контексті продажів та маркетингу візуалізація даних відіграє ключову роль. Наприклад, візуалізація даних соціальних мереж може поліпшити сегментацію аудиторії, генерацію потенційних клієнтів, вихід на зв'язок з продажами та залучення до контенту. Візуалізація даних допомагає маркетологам адаптувати свої стратегії та краще розуміти переваги та поведінку своєї аудиторії. Частота читання контенту зростає на 80%, коли він включає візуальні елементи, що підкреслює важливість візуальних елементів у залученні аудиторії [7];

– загальний вплив на продуктивність: дослідження вказують, що більше двох третин респондентів вважають, що візуалізація даних має "дуже високий" вплив на продуктивність користувачів, з додатковими 30%, які описують її вплив як "помірний". Це підкреслює ефективність візуальних засобів у підвищенні бізнес-інтуїції та прискоренні часу до отримання інсайтів, тим самим підвищуючи продуктивність [8].

Підсумовуючи, візуалізація даних відіграє ключову роль у покращенні ефективності обробки інформації, прийняття рішень та продуктивності у різних сферах бізнесу. Здатність швидко розуміти та запам'ятовувати візуальну інформацію, а також ефективність, яку вона вносить у зустрічі та стратегічне планування, робить її цінним інструментом у сучасному бізнес-середовищі.

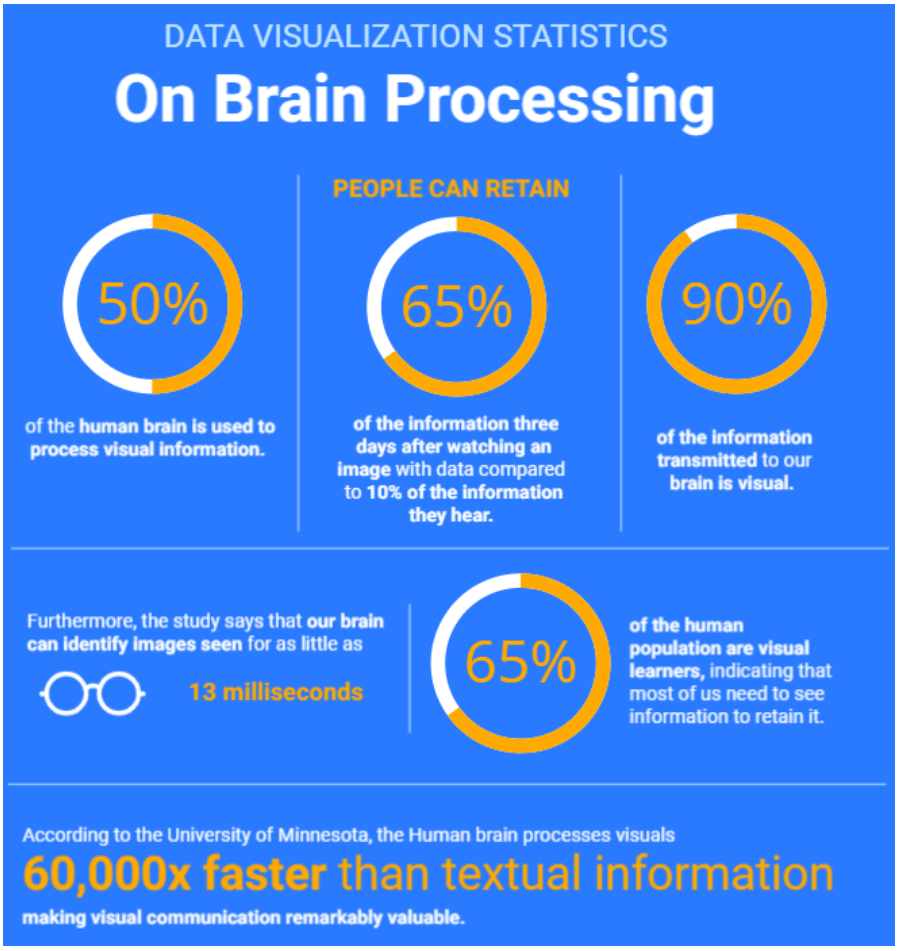


Рисунок 3.2 – Статистика обробки та засвоєння інформації

Рисунок 3.3 – Вплив візуалізації інформації у бізнесі

3.4 Алгоритми прогнозування

Методи прогнозування є ключовими інструментами в аналітиці та плануванні, які використовуються для оцінки майбутніх подій на основі аналізу існуючих даних. Вони відіграють важливу роль у різноманітних областях - від економічного прогнозування до оптимізації ланцюгів поставок, від стратегічного планування до управління ризиками. Суть прогнозування полягає у використанні історичних даних для виявлення закономірностей, що можуть бути проекцією на майбутнє. Це дозволяє організаціям бути більш підготовленими до майбутніх змін, адаптуватися до можливих ринкових коливань та використовувати ресурси більш ефективно.

Існує багато різних методів прогнозування, кожен з яких має свої особливості, переваги та обмеження. Деякі методи прості і зрозумілі, тоді як інші - значно складніші та вимагають спеціальних знань у галузі статистики чи машинного навчання. Вибір методу залежить від специфіки даних, цілей прогнозування, доступних ресурсів та необхідної точності.

Важливо підкреслити, що прогнозування не завжди є точним, оскільки воно засноване на припущенні, що минулі тренди та закономірності продовжать існувати у майбутньому. Тому, крім самого методу прогнозування, важливе значення має також розуміння можливих змін у зовнішніх умовах, які можуть вплинути на точність прогнозів.

Далі будуть описані конкретні методи прогнозування, які часто використовуються в аналізі часових рядів.

3.4.1 Метод ковзної середньої

Метод ковзної середньої є одним з найпростіших і найбільш популярних інструментів для аналізу часових рядів. Він використовується для виявлення тенденцій у даних, зменшення випадкових коливань та прогнозування майбутніх

значень. Далі будуть описані конкретні методи прогнозування, які часто використовуються в аналізі часових рядів

Метод ковзної середньої згладжує часові ряди, розраховуючи середні значення певної кількості послідовних спостережень. Це середнє значення MA_t вираховується для "вікна" певного розміру, яке "ковзає" по часовому ряду з кожним новим періодом [16]:

$$MA_t = \frac{1}{n}(X_{t-n+1} + X_{t-n+2} + \cdots + X_t), \quad (3.1)$$

де MA_t — ковзна середня у момент часу t,

X_t — значення часового ряду у момент часу t,

n — кількість періодів у "вікні" ковзної середньої.

Метод ковзної середньої має декілька варіацій:

– проста ковзна середня (SMA): розраховується як простий середній арифметичний попередніх n значень;

– зважена ковзна середня (WMA): кожне значення в "вікні" має власну вагу, яка зменшується для більш ранніх значень;

– експоненційно зважена ковзна середня (EWMA): найсвіжіші дані мають найбільшу вагу, яка експоненційно зменшується для більш ранніх значень.

Метод ковзної середньої може використовуватись для зменшення випадкових коливань та виявлення основних трендів, визначення майбутнього рівня серії на основі недавніх спостережень, аналізу перетину короткотермінових та довготермінових ковзних середніх для визначення змін у тренді.

Переваги методу:

– простота використання та розуміння;

– хороше згладжування для даних без сильних коливань.

Обмеження:

– запізнювання (lag): ковзна середня відстає від поточних трендів через

затримку в "вікні";

– не адаптується до сезонних або циклічних коливань;

– не підходить для даних з сильними трендами.

3.4.2 Експоненційне згладжування

Експоненційне згладжування (Exponential Smoothing, ES) — це метод прогнозування часових рядів, який дозволяє зменшити випадкові коливання і виділити основний тренд чи шаблон у даних. Він є розширенням методу ковзної середньої і призначений для того, щоб приділити більше уваги недавнім спостереженням, тим самим роблячи прогноз більш реагуючим на останні зміни в даних [17].

Експоненційне згладжування вагомо знижує вплив старих спостережень з часом, надаючи більше ваги новітнім даним. Це досягається за допомогою застосування коефіцієнта згладжування α, який визначає, як швидко вплив старих спостережень зменшується.

Формула простого експоненційного згладжування:

$$S_t = \alpha X_t + (1 - \alpha) S_{t-1} , \tag{3.2}$$

де S_t — прогнозоване (зглажене) значення в момент часу t,

X_t — фактичне значення часового ряду в момент часу t,

α — коефіцієнт згладжування, який варіюється від 0 до 1,

S_{t-1} — прогнозоване значення на попередньому кроці.

Варіації методу:

– просте експоненційне згладжування: використовується для даних без тренду та сезонності;

– подвійне експоненційне згладжування (метод Хольта): додає компонент для врахування тренду в даних;

– тройне експоненційне згладжування (метод Хольта-Вінтерса): додає

третій компонент для врахування сезонності.

Експоненційне згладжування застосовується, коли необхідно швидко реагувати на зміни в найновіших даних, особливо при прогнозуванні в короткостроковому періоді. Це робить метод особливо корисним у ситуаціях, де тренди чи шаблони швидко змінюються.

Переваги методу:

– здатність швидко реагувати на останні зміни у даних;

– гнучкість через вибір коефіцієнта згладжування.

Обмеження:

– може бути чутливим до викидів у даних, особливо при високому коефіцієнті згладжування;

– вибір невірного коефіцієнта згладжування може призвести до пере- або недореагування на зміни в даних.

3.4.3 Модель ARIMA

ARIMA – Авторегресійне Інтегроване Ковзне Середнє (AutoRegressive Integrated Moving Average), є складним статистичним підходом до аналізу і прогнозування часових рядів [18]. Модель ARIMA здатна описувати різні стаціонарні та нестаціонарні часові ряди через використання диференціювання для досягнення стаціонарності та авторегресії з ковзним середнім для моделювання кореляцій.

Основні компоненти ARIMA:

– AR (p), авторегресія – модель використовує залежність між спостереженням і декількома попередніми спостереженнями (затримками або лагами);

– I (d), інтегроване – це означає, що дані потрібно один чи декілька разів диференціювати, щоб зробити часовий ряд стаціонарним;

– MA (q): ковзне середнє – модель використовує залежність між спостереженням і залишковими помилками попередніх ковзних середніх.

Модель ARIMA(p, d, q) описує часовий ряд за допомогою рівняння [18]:

$$(1 - \Sigma_{i=1}^{p} \phi_i L^i)(1 - L)^d X_t = (1 + \Sigma_{i=1}^{q} \theta_i L^i)\epsilon_t \ , \qquad (3.3)$$

де L – оператор затримки (лагу),

ϕ_i – параметри авторегресії,

θ_i – параметри ковзного середнього,

ϵ_t – білий шум,

X_t – часовий ряд,

d – порядок диференціювання.

Етапи роботи з ARIMA:

1. Перевірка на стаціонарність: перш ніж застосовувати модель ARIMA, необхідно перевірити часовий ряд на стаціонарність, оскільки ARIMA працює зі стаціонарними даними. Якщо ряд не є стаціонарним, використовують диференціювання.

2. Ідентифікація моделі: вибір відповідних значень p, d, і q, які найкраще описують часовий ряд. Це можна зробити за допомогою аналізу автокореляційної (ACF) та часткової автокореляційної (PACF) функцій.

3. Оцінювання параметрів: після ідентифікації моделі оцінюють параметри за допомогою статистичних методів, наприклад, методу найменших квадратів або максимальної правдоподібності.

4. Перевірка моделі: перевірка адекватності моделі через аналіз залишків, які повинні виглядати як білий шум.

5. Прогнозування: використання моделі для прогнозування майбутніх значень часового ряду.

Переваги методу ARIMA:

- здатність моделювати різноманітні часові ряди;
- гнучкість у виборі параметрів для різних типів даних;
- сильна теоретична основа, яка дозволяє детальне розуміння моделі.

Обмеження ARIMA:

– потребує стаціонарних даних для ефективного моделювання;

– може бути складним у виборі та налаштуванні параметрів моделі;

– часто не враховує вплив зовнішніх чи пояснювальних змінних на часовий ряд.

3.4.4 Методи машинного навчання

Методи машинного навчання використовуються для прогнозування та аналізу часових рядів, де вони забезпечують більш глибоке розуміння даних та можливість виявлення складних зв'язків і шаблонів. Ці методи особливо ефективні в ситуаціях, де традиційні статистичні методи не в змозі впоратися з великою кількістю даних або з даними, що містять складні нелінійні зв'язки [19].

Основні види методів машинного навчання для прогнозування часових рядів:

1. Регресійні моделі:

– лінійна регресія використовується для моделювання лінійних зв'язків між залежною змінною та однією чи кількома незалежними змінними;

– логістична регресія застосовується для прогнозування ймовірності наступу певної події на основі однієї або декількох незалежних змінних.

2. Нейронні мережі:

– штучні нейронні мережі є потужними інструментами для моделювання складних нелінійних зв'язків;

– мережі зворотного поширення помилки (backpropagation networks) та рекурентні нейронні мережі (RNN) особливо корисні для аналізу часових рядів.

3. Ансамблеві методи:

– випадковий ліс (Random Forest) - це метод, який використовує кілька дерев рішень для вироблення більш стабільного та надійного прогнозу;

– градієнтний бустинг - це техніка, що поступово покращує прогнози, використовуючи послідовність слабких моделей, зазвичай дерев рішень.

4. Методи підтримуючих векторів (Support Vector Machines, SVM): SVM

можуть бути використані для класифікації або регресії та ефективні в випадках, де спостерігається велика кількість ознак.

Переваги методів машинного навчання:

– здатність виявляти складні нелінійні зв'язки в даних;

– можливість ефективної роботи з великими об'ємами даних;

– гнучкість у виборі методів та підходів залежно від конкретних потреб і задач.

Обмеження:

– вимагають великої кількості даних для навчання;

– можуть бути складними у розумінні та інтерпретації результатів (особливо глибокі нейронні мережі);

– ризик перенавчання (overfitting), коли модель занадто точно адаптується до тренувальних даних і втрачає здатність до узагальнення.

Використання методів машинного навчання для прогнозування часових рядів може вимагати значних обчислювальних ресурсів і спеціальних знань у сфері обробки даних, статистики та програмування.

3.4.5 Обґрунтування вибору методу прогнозування

Вибір методу експоненціального згладжування полягає в його адаптивності та простоті застосування, особливо коли має місце обмежена кількість даних, як це часто трапляється у нових вебзастосунках. Проста модель експоненціального згладжування є найбільш підходящою, коли відсутні чітко виражені тренди та сезонні коливання, що робить її ідеальною для стартапів або нових проєктів із обмеженим набором історичних даних.

У міру зростання даних та еволюції застосунку може з'явитися потреба в більш складних методах прогнозування, таких як експоненціальне згладжування з урахуванням тренду та сезонності. Планування розширення функціональності прогнозування передбачає можливість включення цих методів у майбутньому, дозволяючи користувачам обирати найбільш підходящий алгоритм на основі

обсягу та характеру їх даних.

До традиційного алгоритму експоненціального згладжування було додано шум для моделювання випадковості та непередбачуваності реальних даних. Це дозволяє підвищити реалістичність моделі, надаючи можливість прогнозування з урахуванням потенційних відхилень та коливань у поведінці даних.

Алгоритм простого експоненціального згладжування включає кроки:

1. Збір історичних даних: використовуються історичні дані $D = \{d_1, d_2, \dots, d_n\}$ про кількість виконаних завдань за минулі періоди часу, де d_i - кількість завдань, виконаних у і-й період.

2. Розрахунок середнього значення: обчислюється середнє значення μ історичних даних: $\mu = \frac{1}{n}(\Sigma_{i=1}^{n} d_i)$. Це значення представляє середню кількість завдань, виконаних у кожен з минулих періодів.

3. Розрахунок стандартного відхилення: визначається стандартне відхилення σ історичних даних, яке показує ступінь розкиду даних відносно середнього значення: $\sigma = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}(d_i - \mu)^2}$.

4. Генерація випадкового шуму: створюється послідовність випадкових чисел з фіксованим початковим значенням (seed), щоб забезпечити відтворюваність результатів. Для кожного прогнозованого періоду генерується випадкове число R_i, яке представляє шум: $R_i = (rand() - 0.5) * 2 * \sigma$, де rand() генерує рівномірно розподілене число від 0 до 1.

5. Додавання шуму до середнього: для кожного прогнозованого періоду до середнього значення додається випадковий шум, щоб отримати прогнозоване значення $P_i = max(\mu + R_i, 0)$. Функція max() гарантує, що прогнозоване значення не буде від'ємним, оскільки кількість завдань не може бути меншою за нуль.

6. Прогнозування: послідовність прогнозованих значень $\{P_1, P_2, \dots, P_{12}\}$ представляє очікувану кількість завдань на наступні 12 місяців.

Цей підхід дозволяє отримати більш адаптивну модель, яка краще відображає потенційні ризики та невизначеності.

4 ОПИС ПРИЙНЯТИХ ПРОЕКТНИХ РІШЕНЬ

4.1 Обґрунтування вибору мови програмування

Для розробки вебзастосунку керування завданнями та проєктами була обрана мова програмування C# в поєднанні з платформою ASP.NET Core та фреймворком Blazor. ASP.NET Core [4] — це високопродуктивний фреймворк для створення сучасних вебзастосунків та API з відкритим вихідним кодом. Він підтримує розробку на C# з використанням моделі .NET, що дозволяє розробникам використовувати повний спектр можливостей цієї мови та платформи [9].

Blazor дозволяє створювати інтерактивні веб-інтерфейси за допомогою C# замість JavaScript, що відкриває двері для тих розробників, які вже знайомі з екосистемою .NET. Завдяки підтримці розподіленої обробки та можливості використання одного коду для клієнтських та серверних частин застосунку, Blazor забезпечує зменшення зусиль на розробку та підтримку.

Entity Framework Core (EF Core) є сучасним, високопродуктивним та модульним ORM-фреймворком для .NET, який дозволяє розробникам працювати з базою даних за допомогою об'єктів .NET, забезпечуючи абстракцію від конкретної бази даних. Використання EF Core значно спрощує процес розробки, оскільки зменшує кількість необхідного коду для реалізації операцій з даними. Також, EF Core підтримує лінійку баз даних та пропонує функціональні можливості, такі як кешування запитів, оптимізовану роботу з транзакціями та асинхронне виконання запитів, що є важливим для підвищення продуктивності вебзастосунків [10].

ASP.NET Core з Blazor вирізняється високою продуктивністю, безпекою, масштабованістю та сумісністю з різними операційними системами. Це робить його ідеальним вибором для створення корпоративних вебзастосунків. Фреймворк підтримує сучасні веб-стандарти та технології і забезпечує

розробникам можливість створення настільки ж багатих та потужних веб-інтерфейсів, як у традиційних додатків.

Для побудови графіків було обрано бібліотеку Blazor-ApexCharts [11], яка є обгорткою для бібліотеки ApexCharts, яка дозволяє інтегрувати гнучкі та візуально привабливі графіки безпосередньо в Blazor-додатки. Це дає можливість легко створювати інтерактивні графіки та діаграми, що можуть бути повністю налаштовані відповідно до потреб користувачів. Blazor-ApexCharts підтримує ряд стандартних типів діаграм, включаючи лінійні, стовпчикові, кругові та багато інших, забезпечуючи тим самим потужний інструментарій для візуалізації даних.

Розробники можуть скористатися всією екосистемою .NET, включаючи бібліотеки класів, інструменти для тестування, пакети NuGet та інші ресурси, що значно прискорює процес розробки та допомагає утримувати високу якість коду.

На рисунку 4.1 показано порівняння продуктивності .NET 5 з іншими технологіями. .NET 5 обробляє 7.01 мільйонів запитів за секунду, що на порядок швидше, ніж Node.js, і швидше, Go, C++ та Java для серверної продуктивності gRPC. З випуском .NET 6 продуктивність Entity Framework Core зросла на 70%, де .NET 6 обробляє 251,992 запитів за секунду порівняно з 147,852 у .NET 5 (рис. 4.2) [12]. Ці дані вказують на значне поліпшення продуктивності .NET з новими версіями та лідерство серед інших веб-фреймворків.

У результаті, вибір ASP.NET Core та Blazor для розробки вебзастосунку керування завданнями та проєктами був зумовлений потребою у надійній, гнучкій та продуктивній платформі, яка дозволяє використовувати переваги C# та .NET у повній мірі, одночасно забезпечуючи легкість у впровадженні сучасних веб-технологій.

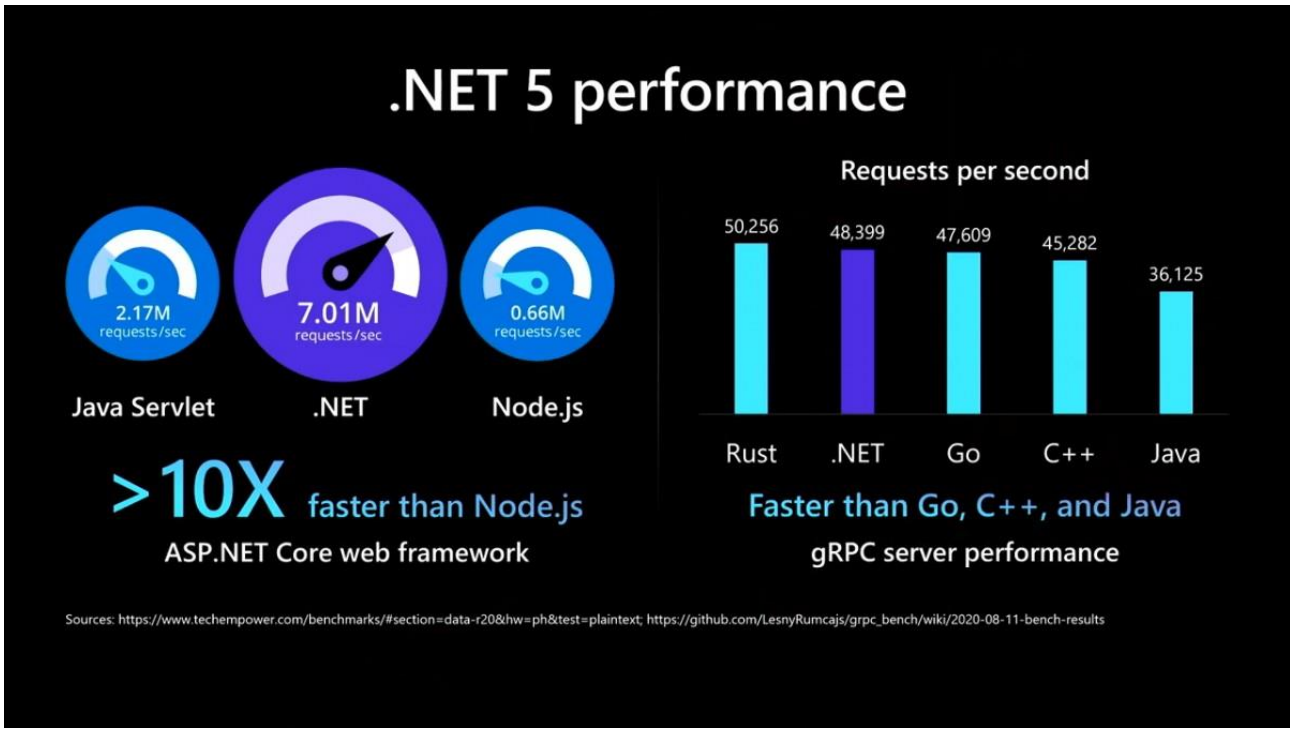


Рисунок 4.1 – Порівняння .NET 5 з іншими мовами програмування

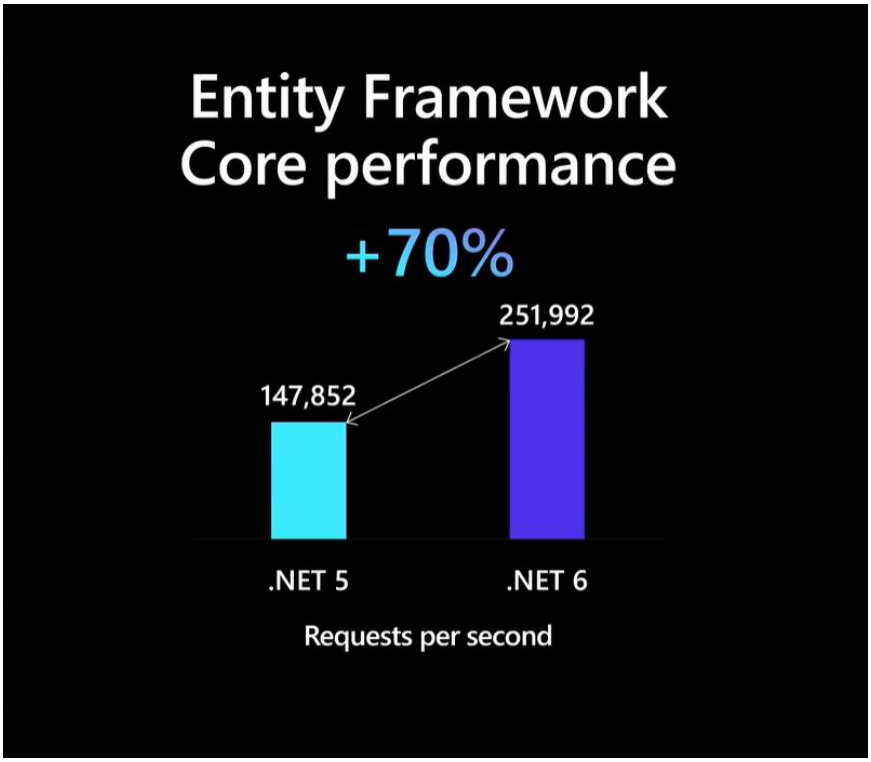


Рисунок 4.2 – Порівняння нової версії .NET 6 з .NET 5

4.2 Обґрунтування вибору платформи СУБД

Система управління базами даних — це комплексне програмне забезпечення, яке дозволяє користувачам створювати, редагувати, управляти та організовувати дані в структурованому форматі. Загалом, СУБД поділяються на два основних типи: реляційні та нереляційні. Реляційні СУБД, такі як Microsoft SQL Server, використовують таблиці для зберігання інформації, що дозволяє виконувати складні запити та аналізи.

Microsoft SQL Server — це реляційна СУБД, її вибір для вебзастосунку можна обґрунтувати її масштабованістю, високою продуктивністю, інтегрованістю з іншими продуктами Microsoft, вбудованими засобами безпеки та підтримкою транзакцій. Крім того, SQL Server пропонує глибоку інтеграцію з інструментами розробки та системами бізнес-аналітики, що робить її ідеальною для корпоративних середовищ, де потрібна надійність та розширені аналітичні можливості [13].

Переваги SQL Server включають:

– надійність та стабільність: висока стабільність та підтримка критично важливих систем;

– інтегровані рішення для бізнес-аналітики: інструменти, такі як SQL Server Reporting Services та SQL Server Analysis Services;

– безпека: розширені функції безпеки, включаючи шифрування, аудит та авторизацію на рівні даних;

– масштабованість: підтримка великих обсягів даних та одночасної роботи великої кількості користувачів;

– інтеграція з .NET: оптимізована робота з .NET-додатками, що забезпечує швидкий доступ та обробку даних.

Огляд цих переваг демонструє, чому Microsoft SQL Server є оптимальним вибором для розробки надійних, безпечних та високопродуктивних корпоративних вебзастосунків.

4.3 Опис архітектури розробленої системи

Архітектура вебзастосунку на Blazor використовує сучасний підхід до побудови вебзастосунків, об'єднуючи клієнтську та серверну частини в єдиному додатку. Вона є модульною та масштабованою, з чітким розділенням відповідальності між різними рівнями системи, що забезпечує високу продуктивність та легкість у підтримці та розвитку.

Архітектура, реалізована з використанням Razor компонентів, представляє собою модель, яка зосереджена на взаємодії користувача з системою та управлінні даними. Разом із MS SQL Server базою даних та Entity Framework Core для доступу до даних, ця архітектура дозволяє створювати швидкі та ефективні вебзастосунки з багатим користувацьким інтерфейсом.

У такому підході, логіка застосунку розподілена між компонентами Razor, які не лише формують UI, але й виконують функції бізнес-логіки, обробляючи події користувача та забезпечуючи взаємодію з репозиторіями. Репозиторії відповідають за ізоляцію та капсулювання запитів до бази даних, що дозволяє зменшити залежність UI від конкретної схеми даних і сприяє більшій гнучкості та масштабованості застосунку.

Архітектурні складові:

– компоненти Razor: виконують роль презентаційного шару та бізнес-логіки, дозволяючи розробникам вбудовувати логіку безпосередньо всередину компонентів для реагування на користувацькі дії, валідації, управління станом та інших аспектів взаємодії з користувачем;

– репозиторії: цей шар забезпечує абстракцію доступу до даних, реалізуючи взаємодію з контекстом бази даних через EF Core. Він дозволяє стандартизувати запити до бази даних, забезпечує можливість їх перевикористання та спрощує тестування та підтримку коду;

– EF Core контекст: слугує шаром доступу до даних, керуючи взаємодією з базою даних MS SQL Server. EF Core ормує об'єкти в коді до таблиць бази даних, забезпечуючи гладку та ефективну роботу з даними.

– MS SQL Server: використовується як реляційна система управління базами даних для зберігання, пошуку та управління даними. Пропонує розширені функції для високопродуктивних операцій з даними та їх безпеки.

Така архітектура забезпечує чітке розділення відповідальностей між різними частинами застосунку та сприяє більшій організованості та легкості розширення його функціональності.

4.4 Розробка інтерфейсу системи

Вебзастосунок для керування завданнями та проєктами забезпечує зручний та інтуїтивно зрозумілий інтерфейс, який дозволяє користувачам ефективно управляти своїми проєктами та завданнями. Головна мета інтерфейсу - забезпечити легкий доступ до всіх функцій застосунку, мінімізуючи кількість кліків та переходів між різними екранами.

Основною точкою входу в застосунок є форма логіну, яка містить поля для введення імені користувача та пароля, а також кнопку «Увійти», що ініціює процес автентифікації (рис 4.3).

Для нових користувачів, які ще не мають облікового запису, передбачена можливість реєстрації. Натиснувши на кнопку «Реєстрація», користувач переходить на сторінку реєстрації, де йому необхідно ввести такі дані, як електронна пошта, ім'я користувача, пароль та підтвердження пароля (рис 4.4). Усі поля форми реєстрації супроводжуються валідацією, що відображає повідомлення про помилки, якщо введені дані не відповідають вимогам системи. Такий підхід допомагає запобігти найпоширенішим помилкам під час реєстрації, таким як некоректні дані або недостатньо міцний пароль.

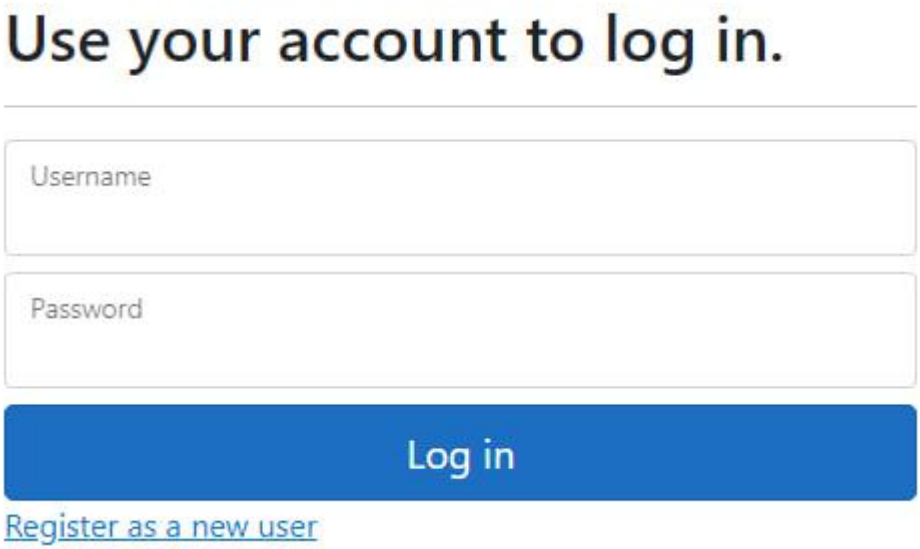


Рисунок 4.3 – Форма логіну

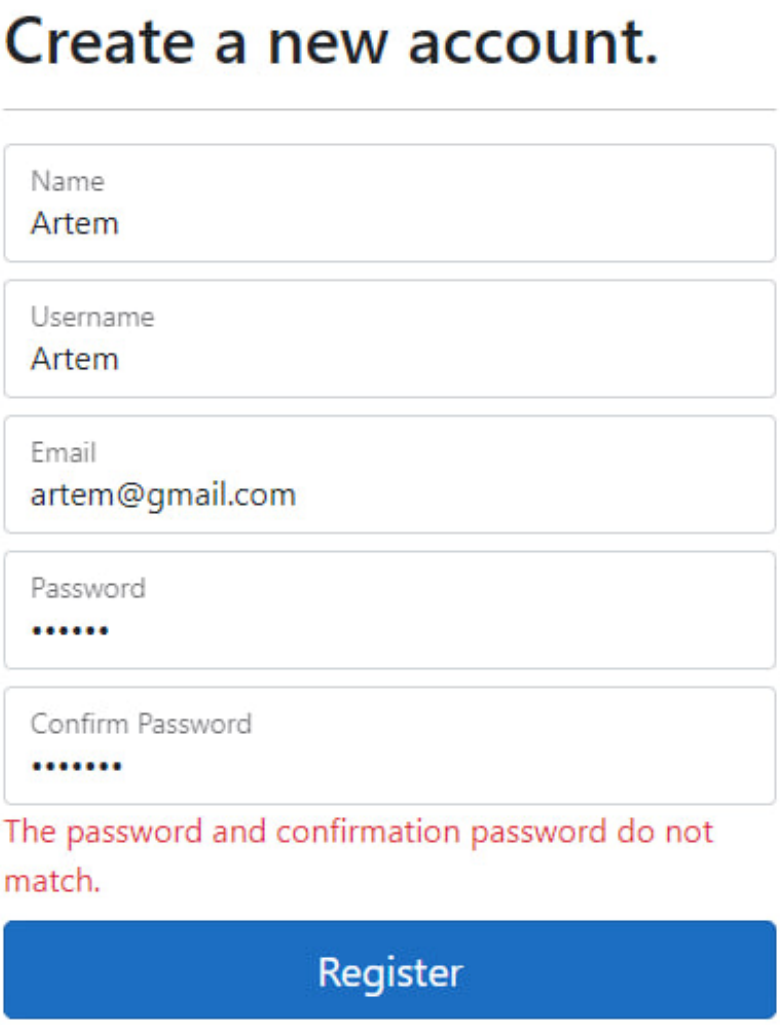


Рисунок 4.4 – Форма реєстрації

У верхній частині головної сторінки вебзастосунку після реєстрації або входу в систему розташовані вітальне повідомлення та іконка користувача, які відображають ім'я залогіненого користувача, створюючи відчуття персоналізації та привітності інтерфейсу (рис. 4.5).

Справа від вітального повідомлення знаходяться два випадаючі списки, які дають можливість користувачеві швидко перемикатися між контекстами в межах застосунку. Перший випадаючий список дозволяє вибрати компанію зі списку всіх компаній, до яких користувач має доступ. При зміні вибраної компанії, випадаючий список проектів автоматично оновлюється, показуючи лише проекти, що належать обраній компанії. Це забезпечує легкість навігації та ефективність управління проектами в рамках вибраної компанії.

Крім того, в кожному з цих списків передбачені опції для швидкого доступу до форм створення нових компаній та проектів. Це дозволяє користувачам легко розширювати свій портфель проєктів та компаній без необхідності переходу на інші сторінки.

На скріншоті також видно форму для створення нової компанії. Для створення компанії користувачеві потрібно ввести назву та веб-адресу, а потім натиснути кнопку «Create» (рис. 4.5). Ця форма розташована на зручному для користувача місці й інтуїтивно зрозуміла, що забезпечує легкість у використанні та ефективність створення нових бізнес-структур в системі.

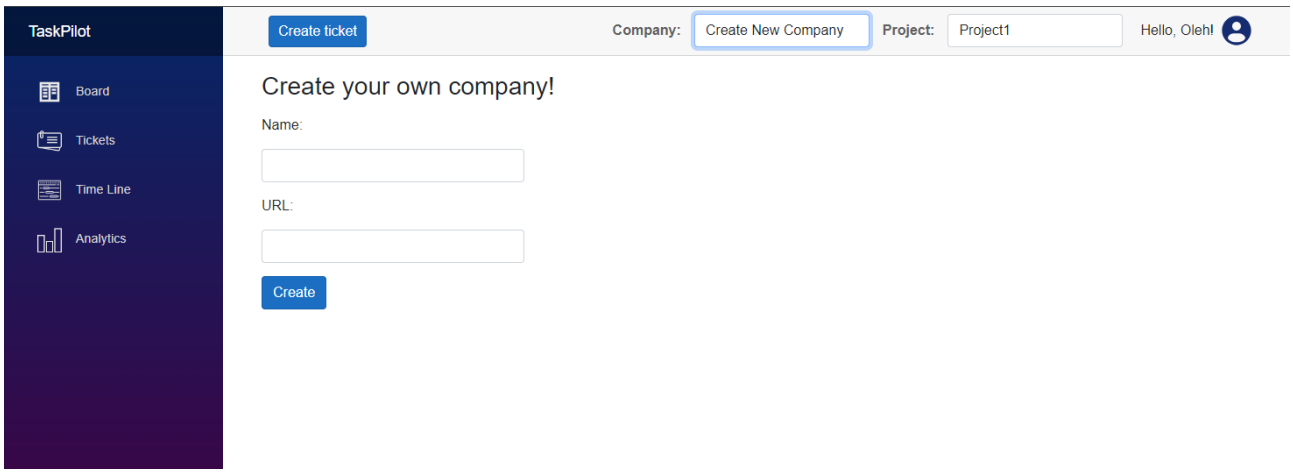


Рисунок 4.5 – Форма створення компанії

На рисунку 4.6 зображено верхню панель користувацького інтерфейсу вебзастосунку. При натисканні на іконку користувача з'являється спливаюче меню з трьома опціями: «Company Settings», «Project Settings» та «Log out». Опції «Company Settings» та «Project Settings» доступні тільки користувачам з роллю «Менеджер», що дозволяє їм здійснювати управління налаштуваннями компанії та проектів відповідно. Такий підхід забезпечує додатковий рівень контролю та безпеки, дозволяючи лише авторизованим особам вносити зміни у важливі сегменти застосунку.

Пункт «Log out» дає можливість користувачам швидко та безпечно вийти з системи, що є важливою функцією з точки зору безпеки, особливо коли доступ до застосунку здійснюється з публічних або спільних пристроїв.

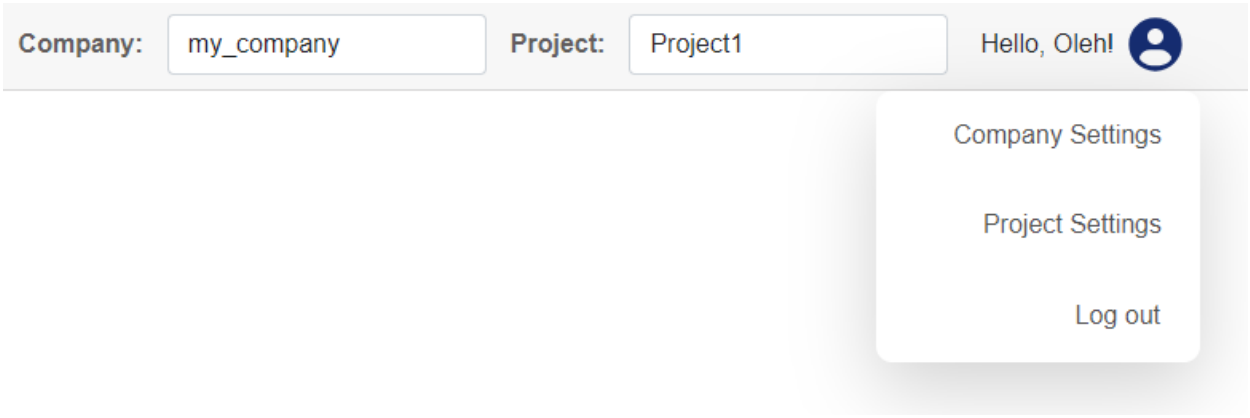


Рисунок 4.6 – Спливаюче меню

Рисунок 4.7 демонструє сторінку налаштувань проекту у вебзастосунку, де користувачі можуть оновлювати інформацію про проект та керувати учасниками проекту. У верхній частині сторінки знаходяться поля для оновлення назви проекту та його URL, що дозволяє користувачам з легкістю змінювати ці параметри. Після внесення змін необхідно натиснути кнопку «Save» для їх збереження.

Нижня частина сторінки містить розділ «Manage Users», де можна встановлювати ролі для користувачів, що вже додані до проекту, чи прибирати їх із проекту за допомогою кнопки «Remove».

Також існує можливість додавати нових користувачів до проекту за допомогою випадаючого списку з користувачами в системі та кнопки «Add User», що спрощує процес нарощування команди для ефективної роботи над проектом. Ця функція забезпечує гнучке управління ресурсами та сприяє легкому розподілу завдань та обов'язків серед членів проектної команди.

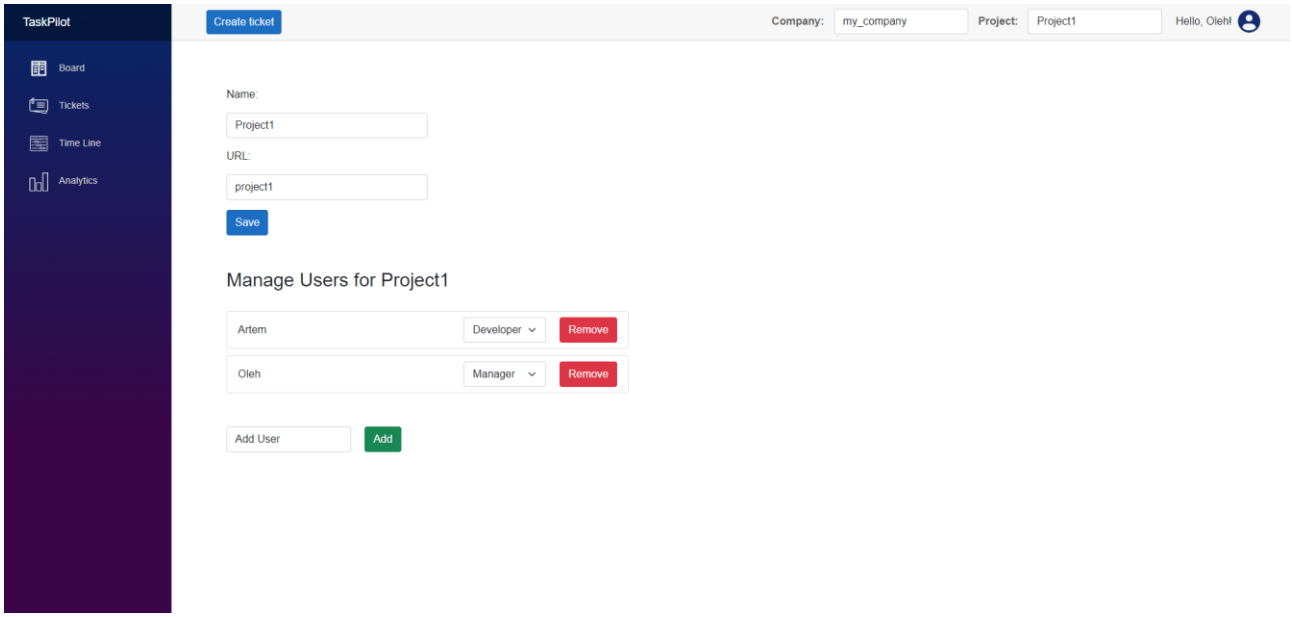


Рисунок 4.7 – Налаштування проекту

При натисканні у верхній частині сторінці кнопки «Create ticket» , яка доступна з будь-якого місця у вебзастосунку, користувача перенаправить на форму створення нового завдання (рис. 4.8). Користувачам пропонується інтуїтивно зрозуміла форма з полями для введення основної інформації про завдання. Поля форми включають «Name» для назви завдання, «Description» для детального опису, «Assignee» для призначення відповідальної особи, яка буде виконувати завдання, та «Project» для вибору проекту, до якого належить завдання.

Користувачі також можуть встановити пріорітетність завдання, обравши один з доступних варіантів, наприклад «Low», «Medium» або «High», в залежності від його важливості. Статус завдання визначається у полі «Status». Поле «Due Date» дозволяє встановити кінцевий термін виконання завдання, а

«Estimated Time» – очікувану кількість годин на виконання завдання.

Після заповнення всіх полів форми користувач має натиснути на кнопку «Create» для створення нового завдання. Ця дія додасть завдання до системи керування завданнями та проєктами, де воно буде доступне для подальшого відстеження та управління.

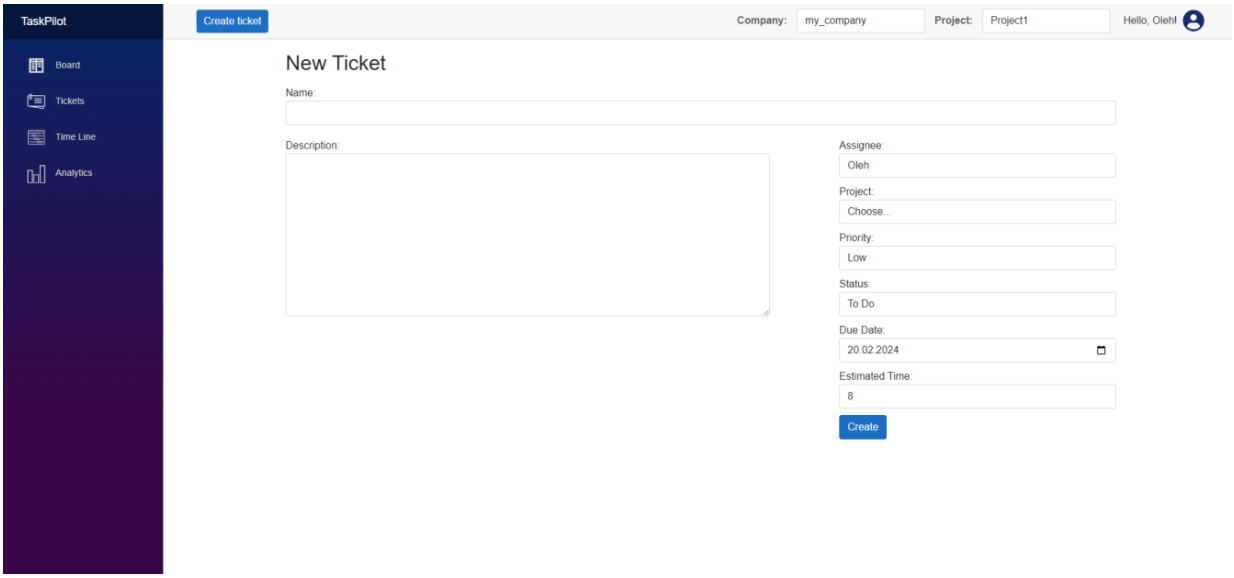


Рисунок 4.8 – Форма створення нового завдання

Після створення нового завдання, користувач може перейти до розділу «Tickets» через ліве меню (рис. 4.9). Ліва частина сторінки представляє список завдань, де кожен елемент списку містить назву завдання, ім'я виконавця, посилання на завдання, а також індикатор пріоритетності у вигляді кольорового кружечка (червоний для високого, жовтий для середнього, зелений для низького пріоритету).

Над списком завдань знаходиться панель фільтрації, де користувачі можуть виконати пошук завдань за різними критеріями: назвою завдання, іменем виконавця, проектом, пріоритетом. Для активації фільтрації потрібно вибрати необхідні параметри і натиснути кнопку «Apply». Для скидання усіх фільтрів використовується кнопка «Reset filters».

Клік по конкретному завданню відкриває детальну інформацію про нього в основній частині сторінки. Ця інформація включає назву завдання, опис,

виконавця, проект, до якого воно належить, пріоритет, статус, крайній термін виконання, оцінку часу, необхідного для виконання, та фактичний час, витрачений на завдання. При натисканні на кнопку «Edit», поля форми стають доступними для редагування, і користувач може внести зміни в інформацію про завдання, а потім зберегти їх, натиснувши кнопку «Save» (рис. 4.10). Це дозволяє аналітикам зручно редагувати створенні завдання, без зайвих переходів на окремі форми.

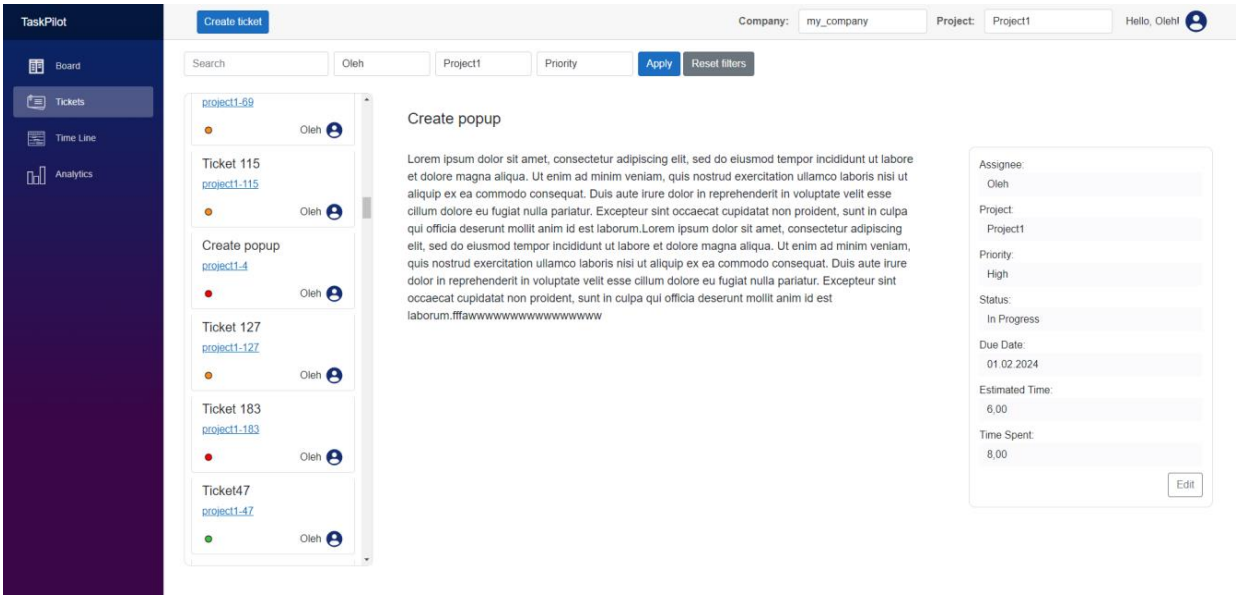


Рисунок 4.9 – Список завдань та детальна інформація про завдання

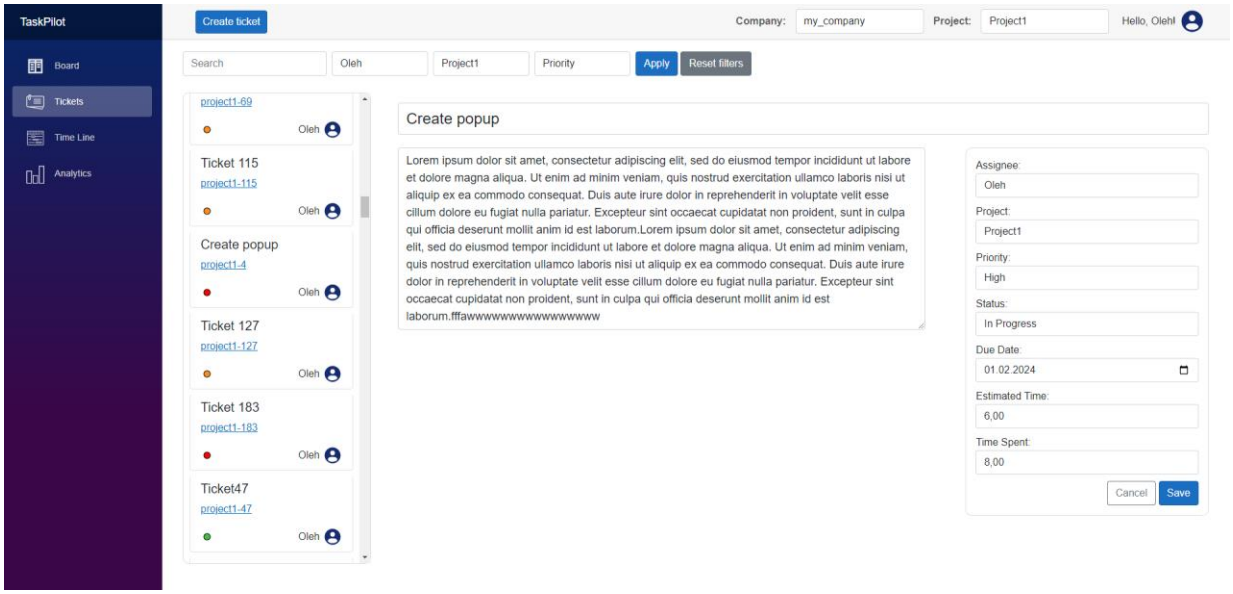


Рисунок 4.10 – Редагування завдання

На рисунку 4.11 представлено інтерфейс канбан-дошки вебзастосунку для управління проектами та завданнями. Вона розділена на колонки, які відповідають різним статусам завдань: «To Do», «In Progress», «Testing», «On Hold» та «Completed». Кожен статус містить картки завдань з іменем виконавця, назвою проекту та індикатором пріоритету у вигляді кольорового кружечка.

Над дошкою розташована панель фільтрації, де можна виконати пошук завдань за різними параметрами, такими як користувач, проект чи пріоритет.

Користувачі можуть легко перетягувати картки завдань між колонками, що дозволяє візуально відстежувати прогрес їх виконання. При перетягуванні завдання в колонку «In Progress», автоматично встановлюється дата початку виконання у поле «Start Date». Аналогічно, коли завдання переміщається до колонки «Completed», для нього фіксується дата завершення в поле «EndDate», що сприяє зручному управлінню проектами та ефективному моніторингу виконання завдань.

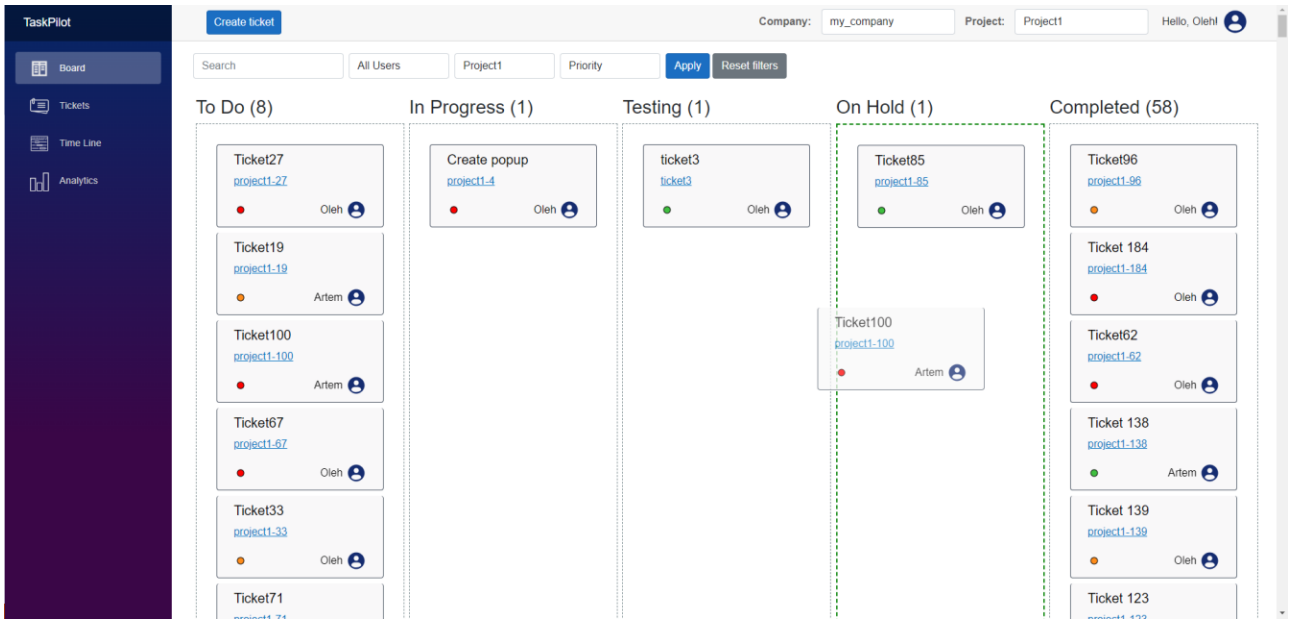


Рисунок 4.11 – Канбан-дошка

При виборі пункту «Time Line» в основному меню користувач перейде до сторінки з діаграмою Ганта вебзастосунку для управління завданнями та

проектами (рис. 4.12). Діаграма Ганта є популярним інструментом у проектному менеджменті для відображення графіку проекту. Вона дозволяє аналітикам та менеджерам проектів візуалізувати часові рамки та прогрес виконання окремих завдань в рамках проекту [23].

На діаграмі кожне завдання представлене у вигляді горизонтальної смуги, довжина якої відповідає часовому проміжку його виконання. Ліва колонка містить список завдань з можливістю вертикальної прокрутки для легкого доступу до будь-якого завдання в списку. Хедер діаграми включає місяці, що дозволяє користувачам бачити розподіл завдань за часом.

Завдання, що перевищили встановлений термін завершення, підсвічуються червоним кольором, це вказує на потенційні затримки. Завдання, які виконуються в рамках терміну або не мають жорстко визначеного дедлайну, показані фіолетовим кольором.

Хоча на цій діаграмі не представлені зв'язки між завданнями, і не можна змінювати терміни безпосередньо, що зроблено для обходу перенавантаження інтерфейсу, вона слугує як інформаційний інструмент для аналітиків, які вивчають загальний прогрес проекту. При наведенні курсору на елемент діаграми відображається спливаюче вікно з детальною інформацією про завдання (рис. 4.13), включаючи його назву, виконавця та заплановані терміни виконання, що дозволяє користувачам швидко отримати перегляд статусу без потреби переходу на інші сторінки або відкриття додаткових меню. При натисненні ж на завдання ми перейдемо на сторінку з його детальною інформацією (рис. 4.9).

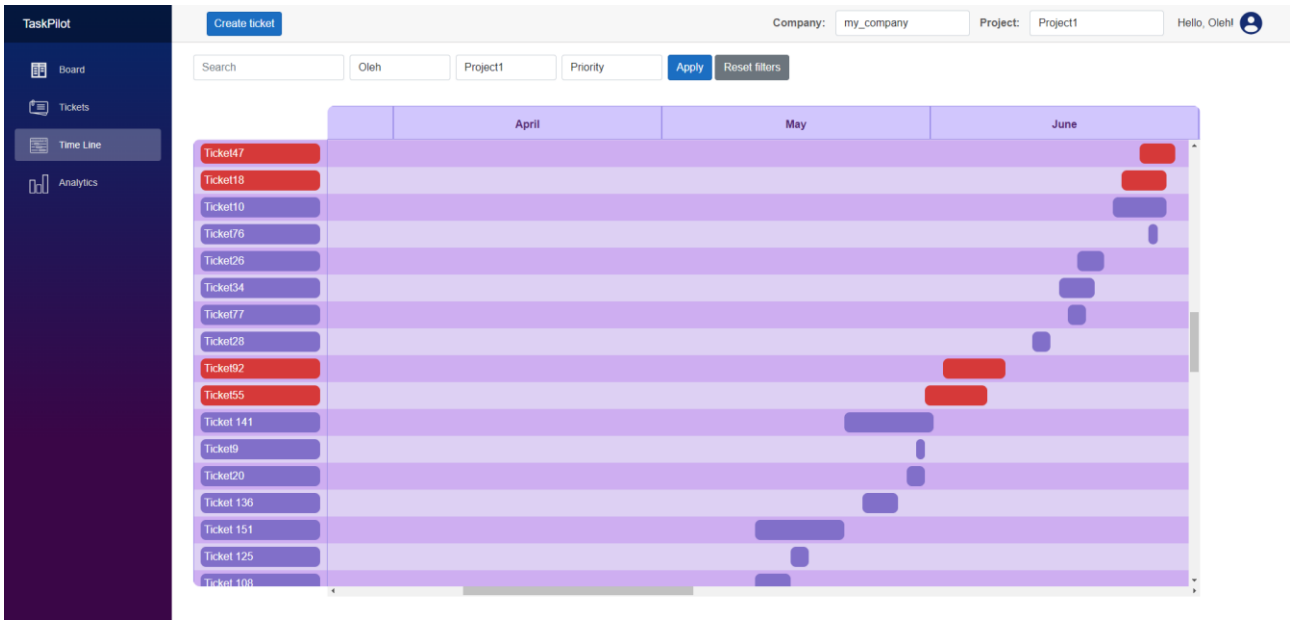


Рисунок 4.12 – Діаграма Ганта

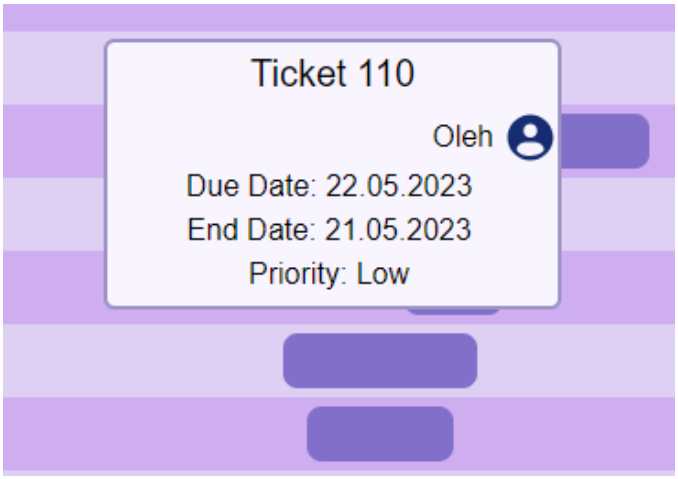


Рисунок 4.13 – Спливаюче інформаційне вікно на діаграмі

Обравши в основному меню пункт «Analytics», відкриються його підпункти. На рисунку 4.14 представлено аналітичну сторінку вебзастосунку з управління проектами та завданнями, яка містить стовпчаті діаграми. Ліва діаграма - це місячний графік, що відображає кількість завдань, виконаних у кожному місяці протягом минулого року. При наведенні курсора на будь-яку смугу діаграми, з'являється вспливаюче вікно, яке показує точну кількість завдань, виконаних в цьому місяці.

Коли користувач обирає конкретний місяць, на правій діаграмі з'являється

статистика за кожним учасником проекту, показуючи кількість завдань, які він виконав у вибраному місяці. Це дозволяє керівникам проектів та аналітикам оцінити внесок кожного члена команди в проект та визначити лідерів за продуктивністю.

Крім того, обидві діаграми можуть бути збережені на локальний пристрій користувача у форматах PNG або SVG для подальшої звітності чи аналізу. Ця функція забезпечує зручність подальшого використання даних діаграм у презентаціях або звітах.

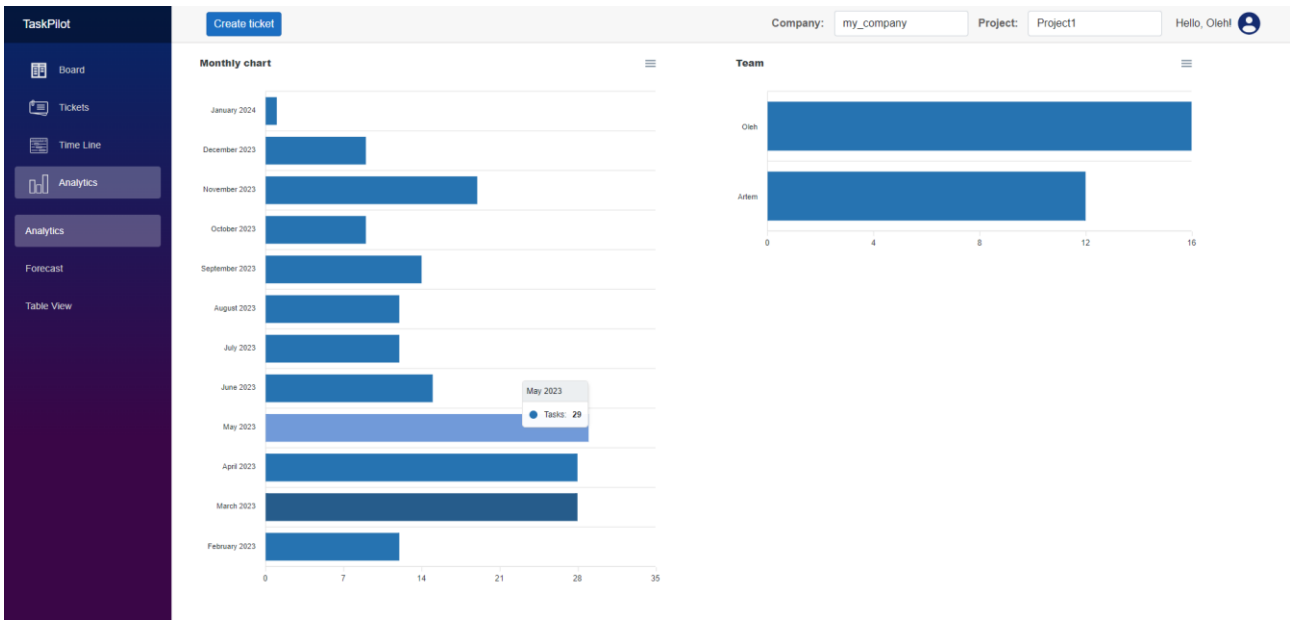


Рисунок 4.14 – Діаграми виконаних завдань

При переході в пункт «Forecast» відкриється сторінка вебзастосунку з прогнозною діаграмою (рис. 4.15), яка використовує алгоритм експоненціального згладжування з введенням шумів для аналізу даних про виконання завдань. Діаграма представляє минулу та прогнозовану кількість завдань за місяцями.

Сині стовпці на діаграмі представляють фактичну кількість завдань, виконаних в минулі місяці. Оранжеві стовпці відображають прогнозовані значення кількості завдань на наступні 12 місяців. Ця візуалізація дозволяє командам візуально оцінити продуктивність праці в минулому та підготуватися

до майбутніх викликів, враховуючи можливі тенденції та зміни в обсязі роботи.

Прогноз формується на основі історичних даних з урахуванням введеного шуму для відображення потенційної варіативності в майбутніх даних. Це може бути корисним для аналітиків та керівників проектів для планування ресурсів, розподілу навантаження та встановлення реалістичних термінів для завдань.

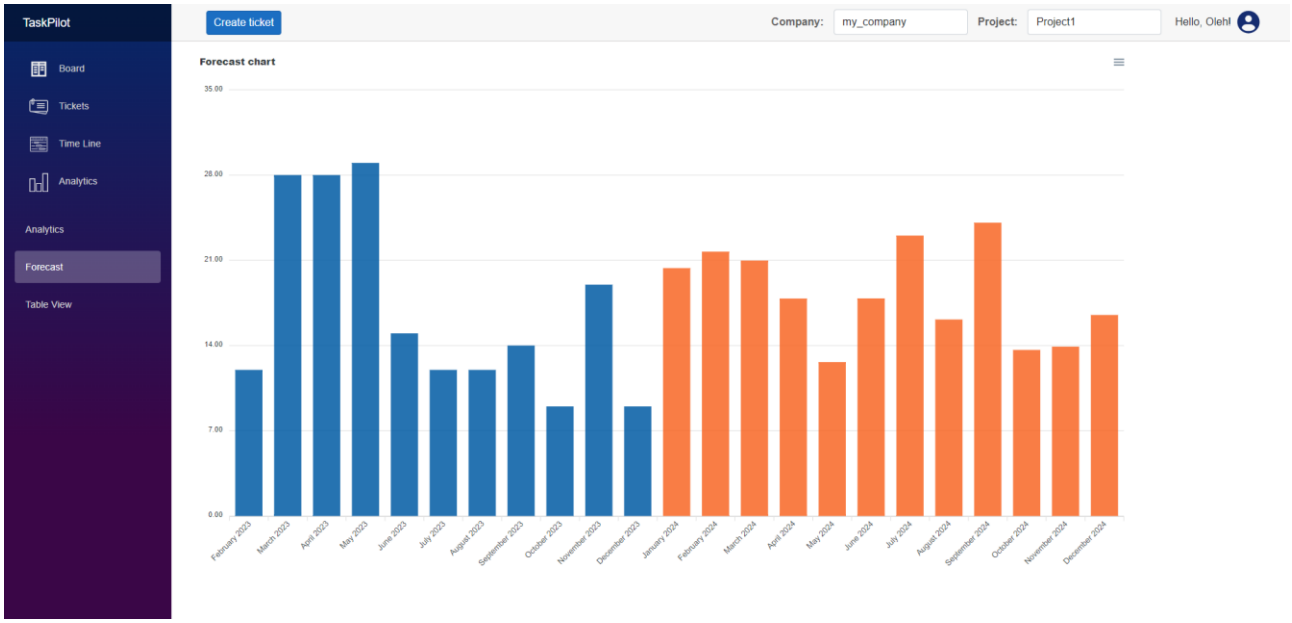


Рисунок 4.15 – Діаграма прогнозування

На сторінці «Table View» бачимо таблицю вебзастосунку для керування завданнями та проєктами (рис. 4.16). Ця таблиця дає користувачам можливість огляду всіх завдань з детальною інформацією.

Колонки таблиці включають:

- назву завдання;
- URL завдання;
- проект, до якого належить завдання;
- виконавець завдання;
- пріоритет;
- статус;
- термін виконання;
- дату початку та закінчення;

– оцінений час та фактичний час витрачений на завдання;

Внизу таблиці відображено сумарний оцінений час та фактичний час, витрачений на всі завдання, що дозволяє аналізувати продуктивність роботи.

В верхній частині сторінки розташований фільтр, який дозволяє користувачам швидко знайти завдання за конкретними параметрами, такими як виконавець, проект або пріоритет. Також додано можливість обрати період, за який потрібно отримати дані, через календарик (рис 4.17).

Стрілочки у заголовках колонок дозволяють сортувати завдання за кожним з параметрів у порядку зростання або спадання, що робить процес управління завданнями більш зручним і ефективним.

Крім того, є можливість експортувати ці дані у формат Excel для подальшого аналізу або звітності, що робить цей інструмент важливим для менеджерів проектів та аналітиків. На рисунку 4.18 представлено експортований файл Excel.

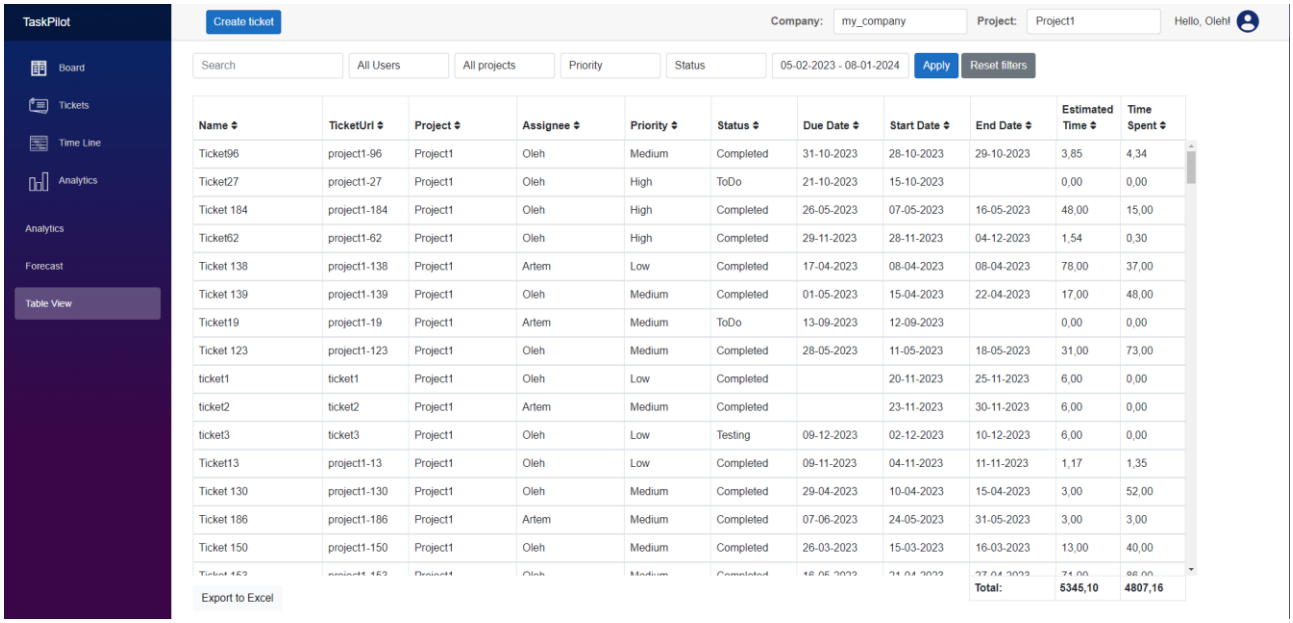


Рисунок 4.16 – Табличне представлення

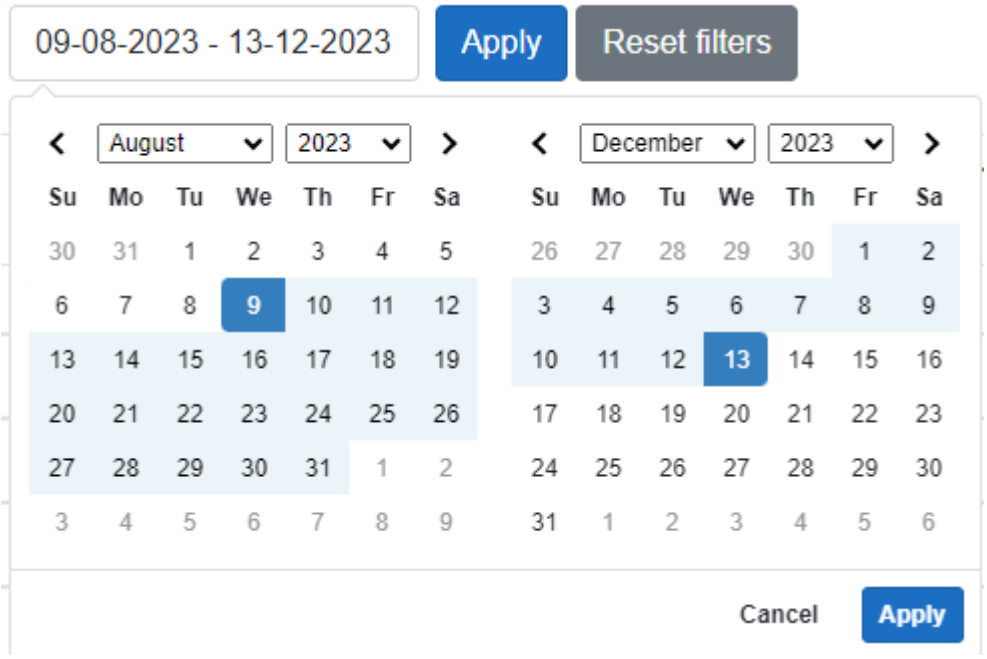
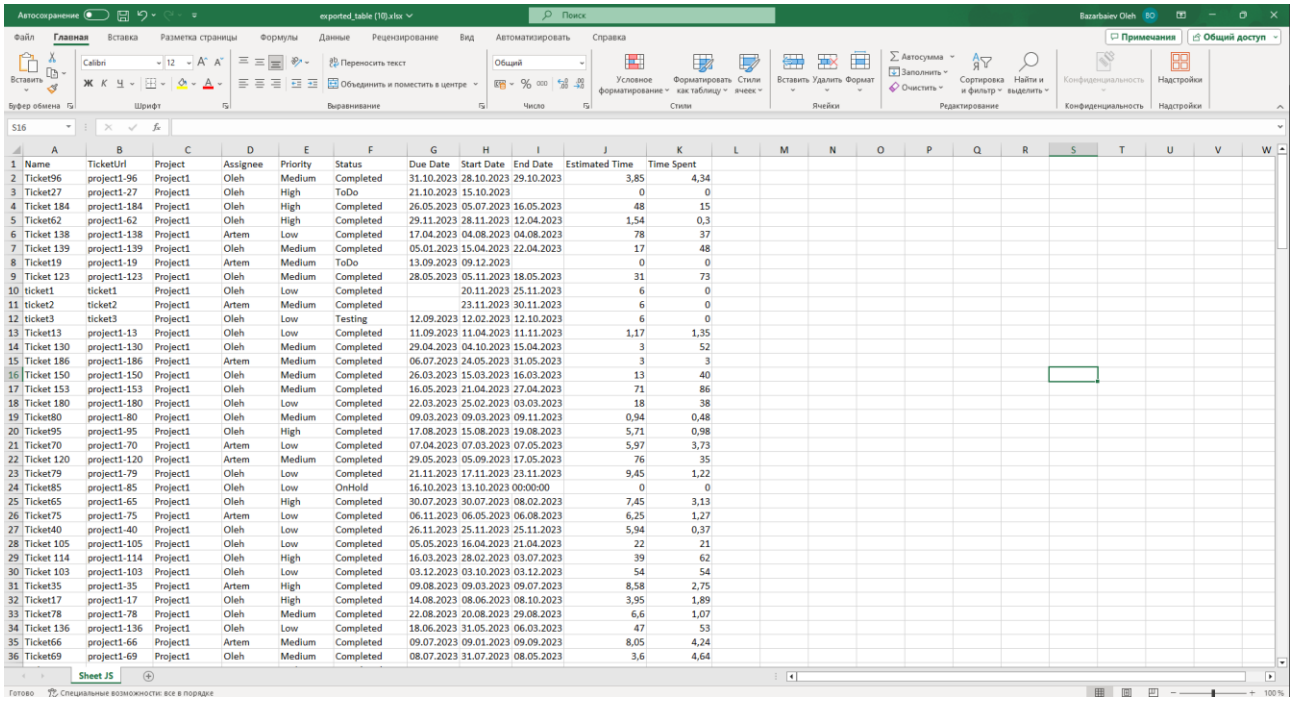


Рисунок 4.17 – Фільтр по даті



Name	TicketUrl	Project	Assignee	Priority	Status	Due Date	Start Date	End Date	Estimated Time	Time Spent
Ticket96	project1-96	Project1	Oleh	Medium	Completed	31.10.2023	28.10.2023	29.10.2023	3,85	4,34
Ticket27	project1-27	Project1	Oleh	High	ToDo	21.10.2023	15.10.2023		0	0
Ticket 184	project1-184	Project1	Oleh	High	Completed	26.05.2023	05.07.2023	16.05.2023	48	15
Ticket62	project1-62	Project1	Oleh	High	Completed	29.11.2023	28.11.2023	12.04.2023	1,54	0,3
Ticket 138	project1-138	Project1	Artem	Low	Completed	17.04.2023	04.08.2023	04.08.2023	78	37
Ticket 139	project1-139	Project1	Oleh	Medium	Completed	05.01.2023	15.04.2023	22.04.2023	17	48
Ticket19	project1-19	Project1	Artem	Medium	ToDo	13.09.2023	09.12.2023		0	0
Ticket 123	project1-123	Project1	Oleh	Medium	Completed	28.05.2023	05.11.2023	18.05.2023	31	73
ticket1	ticket1	Project1	Oleh	Low	Completed		20.11.2023	25.11.2023	6	0
ticket2	ticket2	Project1	Artem	Medium	Completed		23.11.2023	30.11.2023	6	0
ticket3	ticket3	Project1	Oleh	Low	Testing	12.09.2023	12.02.2023	12.10.2023	6	0
Ticket13	project1-13	Project1	Oleh	Low	Completed	11.09.2023	11.04.2023	11.11.2023	1,17	1,35
Ticket 130	project1-130	Project1	Oleh	Medium	Completed	29.04.2023	04.10.2023	15.04.2023	3	52
Ticket 186	project1-186	Project1	Artem	Medium	Completed	06.07.2023	24.05.2023	31.05.2023	3	3
Ticket 150	project1-150	Project1	Oleh	Medium	Completed	26.03.2023	15.03.2023	16.03.2023	13	40
Ticket 153	project1-153	Project1	Oleh	Medium	Completed	16.05.2023	21.04.2023	27.04.2023	71	86
Ticket 180	project1-180	Project1	Oleh	Low	Completed	22.03.2023	25.02.2023	03.03.2023	18	38
Ticket80	project1-80	Project1	Oleh	Medium	Completed	09.03.2023	09.03.2023	09.11.2023	0,94	0,48
Ticket95	project1-95	Project1	Oleh	High	Completed	17.08.2023	15.08.2023	19.08.2023	5,71	0,98
Ticket70	project1-70	Project1	Artem	Low	Completed	07.04.2023	07.03.2023	07.05.2023	5,97	3,73
Ticket 120	project1-120	Project1	Artem	Medium	Completed	29.05.2023	05.09.2023	17.05.2023	76	35
Ticket79	project1-79	Project1	Oleh	Low	Completed	21.11.2023	17.11.2023	23.11.2023	9,45	1,22
Ticket85	project1-85	Project1	Oleh	Low	OnHold	16.10.2023	13.10.2023	00:00:00	0	0
Ticket65	project1-65	Project1	Oleh	High	Completed	30.07.2023	30.07.2023	08.02.2023	7,45	3,13
Ticket75	project1-75	Project1	Artem	Low	Completed	06.11.2023	06.05.2023	06.08.2023	6,25	1,27
Ticket40	project1-40	Project1	Oleh	Low	Completed	26.11.2023	25.11.2023	25.11.2023	5,94	0,37
Ticket 105	project1-105	Project1	Oleh	Low	Completed	05.05.2023	16.04.2023	21.04.2023	22	21
Ticket 114	project1-114	Project1	Oleh	High	Completed	16.03.2023	28.02.2023	03.07.2023	39	62
Ticket 103	project1-103	Project1	Oleh	Low	Completed	03.12.2023	03.10.2023	03.12.2023	54	54
Ticket35	project1-35	Project1	Artem	High	Completed	09.08.2023	09.03.2023	09.07.2023	8,58	2,75
Ticket17	project1-17	Project1	Oleh	High	Completed	14.08.2023	08.06.2023	08.10.2023	3,95	1,89
Ticket78	project1-78	Project1	Oleh	Medium	Completed	22.08.2023	20.08.2023	29.08.2023	6,6	1,07
Ticket 136	project1-136	Project1	Oleh	Low	Completed	18.06.2023	31.05.2023	06.03.2023	47	53
Ticket66	project1-66	Project1	Artem	Medium	Completed	09.07.2023	09.01.2023	09.09.2023	8,05	4,24
Ticket69	project1-69	Project1	Oleh	Medium	Completed	08.07.2023	31.07.2023	08.05.2023	3,6	4,64

Рисунок 4.18 – Експортований файл

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи було розроблено підсистему аналітики для вебзастосунку керування проєктами та завданнями.

Було проведено аналіз предметної області підсистеми аналізу для вебзастосунку керування проєктами та завданнями. Описано сучасний стан розвитку інформаційних систем та технологій та актуальність подальшого їх вдосконалення та підтримки. Під час аналізу реалізованих систем-аналогів, розглянуто їх функціонал та можливості аналітики. Підкреслені як сильні сторони вебзастосунків так і виявлено низку недоліків, такі як: складність в освоєнні та недружній інтерфейс (Jira), перенавантаженість інструментів аналізу (Jira) або навпаки відсутність вбудованих інструментів аналізу (Trello), обмеженість в робочих процесах (Trello).

Були виділені ключові цілі та завдання дослідження підсистеми аналізу для вебзастосунку керування проєктами та завданнями. Проведений аналіз методологій розробки програмного забезпечення показав, що методологія Waterfall хоч і є простою та структурованою, але страждає практично відсутністю гнучкості. Тому для розробки підсистеми аналітики було обрано адаптивну методологію Agile з фреймворком Kanban для кращої візуалізації робочого процесу та зосередження на покращенні робочого процесу.

В рамках дослідження розглянуто різноманітні методи прогнозування для аналітичної підсистеми, що включають класичні статистичні моделі та сучасні машинно-навчальні алгоритми. Серед них було обрано метод експоненційного згладжування, який відрізняється своєю ефективністю у та здатністю адаптуватися до змін у даних. Цей метод виявився найбільш придатним для прогнозування тенденцій у виконанні завдань та управлінні проектами завдяки своїй гнучкості та здатності швидко реагувати на непередбачувані коливання.

У процесі розробки аналітичної підсистеми особлива увага приділена вибору технологій. Для розробки використано Blazor, сучасний фреймворк для створення інтерактивних веб-інтерфейсів з використанням .NET та C#.

Використання Blazor дозволило забезпечити ефективну взаємодію з користувачем, що сприяє поліпшенню продуктивності роботи з системою.

В якості системи управління базами даних обрано Microsoft SQL Server через його високу продуктивність, надійність, широкі можливості аналізу та забезпечення безпеки даних. Застосування Microsoft SQL Server дало можливість ефективно організувати зберігання, обробку та отримання великих обсягів даних, необхідних для аналітичної підсистеми, а також забезпечити швидкий доступ до інформації та її високу доступність.

У майбутньому розроблена підсистема аналітики може бути розширена за рахунок додаткових функціональних елементів, інших алгоритмів прогнозування та адаптації під специфічні потреби різних проектів і організацій, що зробить її незамінним інструментом для сучасного управління проектами.

Результати дослідження з теми кваліфікаційної роботи доповідались на XVI міжнародній науково-практичній конференції [26].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Atlassian JIRA Software, Atlassian Corp., 2023. [Online]. Available: https://atlassian.com/software/jira [Accessed: October 14, 2023].

2. “Trello – це безкоштовний та гнучкий спосіб організувати, що завгодно з ким завгодно”, Atlassian Corp., 2023. [Online]. Available: https://trello.com/ [Accessed: October 14, 2023].

3. “10 найкращих інструментів управління проєктами у 2023 році ”, М. Ложко, 18.07.2023. [Online]. Available: https://blog.depositphotos.com/ua/instrumenti-upravlinnya-proyektamy.html [Accessed: October 14, 2023].

4. Documentation ASP.NET Core [Online]. Available: https://learn.microsoft.com/en-us/aspnet/core [Accessed: October 14, 2023].

5. An introduction to NuGet [Online]. Available: https://learn.microsoft.com/en-us/nuget/what-is-nuget [Accessed: October 14, 2023].

6. 50 Data Visualization Statistics That Prove Its Importance [Online]. Available: https://visme.co/blog/data-visualization-statistics/ [Accessed: January 04, 2024].

7. 80+ Data Visualization Statistics: Sales, Business & More [Online]. Available: https://marketsplash.com/data-visualization-statistics/ [Accessed: January 04, 2024].

8. Evaluating the Impact of Data Visualization [Online]. Available: https://tdwi.org/articles/2011/02/02/Impact-of-Data-Visualization.aspx [Accessed: January 04, 2024].

9. ASP.NET Core Blazor [Online]. Available: https://learn.microsoft.com/ru-ru/aspnet/core/blazor/?view=aspnetcore-8.0 [Accessed: January 09, 2024].

10. Entity Framework Core [Online]. Availablehttps://learn.microsoft.com/ru-ru/ef/core/ [Accessed: January 09, 2024].

11. Apexcharts [Online]. Available: https://apexcharts.com/docs [Accessed: January 09, 2024].

12. The future of modern application development with .NET [Online]. Available: https://youtu.be/2Ky28Et3gy0?si=SpTviFNOAJx9XPWD [Accessed: January 09, 2024].

13. SQL Server technical documentation [Online]. Available: SQL Server technical documentation [Accessed: January 09, 2024].

14. The Waterfall Methodology for Software Development Project Management [Online]. Available: https://www.ejable.com/tech-corner/methodologies/waterfall-methodology/ [Accessed: January 09, 2024].

15. Agile, Scrum, and Kanban [Online]. Available: https://www.toptal.com/project-managers/technical/agile-scrum-kanban-what-do-they-mean [Accessed: January 09, 2024].

16. Simple Moving Average (SMA) [Online]. Available: https://www.investopedia.com/terms/s/sma.asp [Accessed: January 09, 2024].

17. Exponential Smoothing: Definition of Simple, Double and Triple [Online]. Available: https://www.statisticshowto.com/exponential-smoothing/ [Accessed: January 09, 2024].

18. ARIMA models [Online]. Available: https://otexts.com/fpp2/non-seasonal-arima.html [Accessed: January 09, 2024].

19. Understanding machine learning-based forecasting methods [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169207021001771 [Accessed: January 09, 2024].

20. Blazor Basics: A Beginner’s Guide to Blazor Architecture [Online]. Available: https://medium.com/simform-engineering/blazor-basics-a-beginners-guide-to-blazor-architecture-962113daac0c [Accessed: January 09, 2024].

21. “Український веб-довідник” [Online]. Available: https://css.in.ua/ [Accessed: January 09, 2024].

22. Build fast, responsive sites with Bootstrap [Online]. Available: https://getbootstrap.com/ [Accessed: January 09, 2024].

23. Gantt chart [Online]. Available: https://www.techtarget.com/searchsoftwarequality/definition/Gantt-chart [Accessed: January 09, 2024].

24. Web Framework Benchmarks [Online]. Available: https://www.techempower.com/benchmarks/#hw=ph&test=fortune&section=data-r22 [Accessed: January 09, 2024].

25. Design patterns in C# [Online]. Available: https://refactoring.guru/design-patterns/csharp [Accessed: January 09, 2024].

26. Базарбаєв О.Ш., Калита Н.І. Веб-застосунок керування завданнями та проєктами // Інформаційні технології і автоматизація – 2023 / Матеріали XVI міжнародної науково-практичної конференції. Одеса, 19-20 жовтня 2023 р. - Одеса, Видавництво ОНТУ, 2023 р. – с. 160-161