МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНІЙ ТЕХНІЧНИЙ УНІВЕРИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра стратегiчного управлiння

КУРСОВА РОБОТА

на тему: «Розробка веб-застосунку гри на логіку«Running Wild»

Керівник роботи:

| ас. каф. СУ | Ткаченко А. Є. |
| --- | --- |

Виконавець:

| студент гр. КН-318а | Рєдіна Г. М. |
| --- | --- |

Харків – 2021

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ | | | | | | | | | | | | | |
| НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  «ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ» | | | | | | | | | | | | | |
| Кафедра стратегічного управління | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | |
| **КУРСОВА РОБОТА** | | | | | | | | | | | | | |
| з | Програмування веб-застосунків Javascript | | | | | | | | | | | | |
| (назва дисципліни) | | | | | | | | | | | | | |
| на тему: | | Розробка веб-застосунку гри на логіку «Running Wild» | | | | | | | | | | | |
|  | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | |
| Студента (ки) | | 3 | | курсу | | | КН-318а | | групи |
| напряму підготовки | | | 6.050101 Комп’ютерні науки | | | | | | | | |
|  | | | | | | | | | |
| Рєдіна Г. М. | | | | | | | | | |
| (прізвище та ініціали) | | | | | | | | | |
| Керівник | | ас. каф. СУ Луценко С. Ю. | | | | | | | |
|  | | | | | | | | | |
| (посада, вчене звання, науковий ступінь, прізвище та ініціали) | | | | | | | | | |
|  | | | | | | | | | | | |
| Національна шкала | | |  | | | | | | | |
| Кількість балів | |  | | | Оцінка ECTS | | |  | | | |
|  | | | | | | | | | | | | | | |
| Члени комісії | |  | | |  | Мельник К. В. | | | | | |
|  | | (підпис) | | |  | (прізвище та ініціали) | | | | | |
|  | |  | | |  | Луценко С. Ю. | | | | | |
|  | | (підпис) | | |  | (прізвище та ініціали) | | | | | |
|  | |  | | |  |  | | | | | |
|  | | (підпис) | | |  | (прізвище та ініціали) | | | | | |
|  |  | |  |  | | | | | | | |
|  | | | | | | | | | | | | | |
| м. Харків – 2021 рік | | | | | | | | | | | | | |

Національний технічний університет «Харківський політехнічний інститут»

Кафедра програмної інженерії та інформаційних технологій управління

Студент Група КН – 318а Курс 3

З А В Д А Н Н Я

на науково-дослідну курсову роботу

з курсу «Програмування веб-застосунків Javascript»

Тема:

Розробка веб-застосунку гри на логіку «Running Wild» з використанням Javascript – технологій.

Постановка задачі:

Необхідно реалізувати наступні можливості веб-застосунку:

1 Клієнт-серверний RESTful веб-застосунок.

2 Дані повинні зберігатися у документо-орієнтованій базі даних MongoDB.

3 Реалізація REST API бекенду за допомогою Express.

4 Реалізація авторизації користувача.

5 Створити меню гри: нова гра, вибір рівня складності, вихід з гри, таблиця лідерів, Help – опис використання.

6 При досягненні певного результату необхідно реалізувати перехід на інший рівень складності.

7 Розробити презентацію результатів виконання курсової роботи.

Короткий зміст роботи:

а) реферативна частина

Дослідження і аналіз обраної предметної області. Огляд існуючих аналогів для вирішення поставленої задачі. Обґрунтування та вибір технологій для розробки веб-застосунку. Постановка задачі.

б) теоретична частина

Розробка специфікації вимог щодо програмного забезпечення. Побудова необхідних UML-діаграм. Проектування архітектури та інтерфейсу користувача розроблюваної RESTfullсистеми.

в) програмна частина

Реалізація тестування REST full системи. Документування програмного продукту.

Термін захисту курсової роботи \_\_.12.2019

Керівник курсової роботи ас. каф. СУ Луценко С.Ю.

ВІДГУК

на курсову роботу

студентки групи КН-318а Рєдіної Г. М.

«Розробка веб-застосунку гри на логіку «Running Wild» з використанням Javascript – технологій»

В результаті виконання курсової роботи студент Гарбуз С. І. розробив гру на логіку «Running Wild» з використанням Javascript – технологій.

У ході роботи студент проаналізував предметну область індивідуального завдання, розробив модель даних у вигляді схем, здійснив процедуру перетворення отриманої моделі даних в схему документо-орієнтованої бази даних MongoDB, спроектував архітектуру гри з використанням технології "клієнт-сервер".

Із застосуванням принципів об'єктно-орієнтованого аналізу, документно-орієнтованого моделювання даних, а також використання програмних засібів СУБД MongoDB і MS Visual Studio Code для програмування веб-додатків на серверної платформі NodeJS в операційній системі MS Windows 10, студентом була розроблена, реалізована і задокументована браузерна гра-платформер «Running Wild».

В ході виконання курсової роботи Гарбуз С.І. виявив самостійність, обізнаність, наполегливість та ініціативність. Робота виконана самостійно, з використанням сучасних програмних технологій, текст викладено чітко та лаконічно. Обсяг роботи та зміст відповідають вимогам щодо курсових робіт, робота заслуговує оцінки «Відмінно А».

|  |  |  |
| --- | --- | --- |
| ас каф. СУ |  | /Ткаченко А. Є./ |

РЕФЕРАТ

*Ключові слова*: JavaScript, NodeJS, MongoDB, Клієнт-Сервер, Веб-Програмування, Бази Даних

Об'єктом даної курсової роботи є розробка браузерної гри.

Метою роботи є закріплення знань по курсу "Програмування веб-додатків Javascript" і отримання навичок розробки веб-додатків з клієнт-серверною архітектурою. Отримання навичок роботи з серверної частиною і її взаємодія з клієнтською частиною.

Було проаналізовано предметну область індивідуального завдання курсової роботи, з використанням основних понять клієнт-серверної архітектури. Розроблено модель даних у вигляді схем. Проведена процедура перетворення отриманої моделі даних в схему документо-орієнтованої бази даних MongoDB. Проведено проектування архітектури гри з використанням технології "клієнт-сервер".

В ході виконання курсової роботи, послідовно застосовуючи принципи об'єктно-орієнтованого аналізу, документно-орієнтованого моделювання даних, а також використовуючи програмний засіб СУБД MongoDB і MS Visual Studio Code для програмування веб-додатків на серверній платформі NodeJS в операційній системі MS Windows 10, була розроблена, реалізована і задокументована браузерна гра-платформер «Running Wild».

РЕФЕРАТ

Ключевые слова: JavaScript, NodeJS, MongoDB, Клиент-Сервер, Веб-Программирование, Базы Данных

Объектом данной курсовой работы является разработка браузерной игры.

Целью работы является закрепление знаний по курсу "Программирование веб-приложений Javascript" и получения навыков разработки веб-приложений с клиент-серверной архитектурой. Получение навыков работы с серверной частью и ее взаимодействие с клиентской частью.

Была проанализирована предметная область индивидуального задания курсовой работы, с использованием основных понятий клиент-серверной архитектуры. Разработана модель данных в виде схем. Проведена процедура преобразования полученной модели данных в схему документо-ориентированной базы данных MongoDB. Проведено проектирование архитектуры игры с использованием технологии "клиент-сервер".

В ходе выполнения курсовой работы, последовательно применяя принципы объектно-ориентированного анализа, документо-ориентированного моделирования данных, а также используя программное средство СУБД MongoDB и MS Visual Studio Code для программирования веб-приложений на серверной платформе NodeJS в операционной системе MS Windows 10 была разработана, реализована и задокументирована браузерная игра-платформер «Running Wild».

ABSTRACT

Keywords: JavaScript, NodeJS, MongoDB, Client Server, Web Development, Databases

The object of this course work is the development of a browser game.

The purpose of this work is to consolidate the knowledge in the "Programming Web Applications Javascript" course and to acquire the skills of developing web applications with client-server architecture. Obtaining skills to work with the server part and its interaction with the client part.

The subject area of the individual coursework assignment has been analyzed, using the basic concepts of the client-server architecture. The data model in the diagrams form has been developed. The procedure of converting the obtained data model into a schema of a MongoDB document-oriented database has been performed. The architecture of the information system using client-server technology has been designed.

During the course work, consistently applying the principles of the object-oriented analysis, document-oriented data modeling, as well as using the software DBMS MongoDB and MS Visual Studio Code to program web applications on the NodeJS server platform in MS Windows 10, the "Running Wild" browser platformer has been developed, implemented and documented.

Зміст

[**КУРСОВА РОБОТА** 2](#_Toc72472024)

[**ВСТУП** 10](#_Toc72472025)

[**1 ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ** 11](#_Toc72472026)

[**1.1 Аналіз предметної області** 11](#_Toc72472027)

[**1.2** **Огляд існуючих рішень за обраною темою** 11](#_Toc72472028)

[**1.2.1 Todoist** 11](#_Toc72472029)

[**1.2.2 Microsoft To-Do** 12](#_Toc72472030)

[**1.2.3 Weeek** 13](#_Toc72472031)

[**1.3 Постановка задачі** 13](#_Toc72472032)

[**2 СПЕЦИФІКАЦІЯ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ (SRS – SOFTWARE REQUIREMENT SPECIFICATION)** 15](#_Toc72472033)

[**2.1 Загальна інформація щодо веб-застосунку** 15](#_Toc72472034)

[**2.2 Діаграми** 15](#_Toc72472035)

[**2.3 Вимоги до веб-застосунку** 17](#_Toc72472036)

[**2.2.1 Функціональні вимоги** 17](#_Toc72472037)

[**2.2.2 Вимоги до зовнішнього інтерфейсу** 18](#_Toc72472038)

[**2.2.3 Нефункціональні вимоги** 19](#_Toc72472039)

[**2.3 Опис обраних технологій для розробки веб-застосунку** 19](#_Toc72472040)

[**2.3.1 Мова гіперрозмітки HTML5** 19](#_Toc72472041)

[**2.3.2 Каскадні стилі CSS3** 20](#_Toc72472042)

[**2.3.3 Мова програмування JavaScript** 22](#_Toc72472043)

[**2.3.4 Серверна платформа Node.js** 24](#_Toc72472044)

[**2.3.5Фреймворк Express** 26](#_Toc72472045)

[**2.3.6 База даних MongoDB** 26](#_Toc72472046)

[**2.4 Опис архітектури веб-застосунку** 27](#_Toc72472047)

[**2.4.1 Діаграма розміщення** 27](#_Toc72472048)

[**2.4.2 Діаграма компонентів** 28](#_Toc72472049)

[**2.5 Модель бази даних** 29](#_Toc72472050)

[**3 ОПИС ВЕБ-ЗАСТОСУНКУ** 31](#_Toc72472051)

[**3.1 Тестування програмного забезпечення** 31](#_Toc72472052)

[**3.1.1 Валідація програмного продукту** 31](#_Toc72472053)

[**3.1.2 Навантажене тестування** 32](#_Toc72472054)

[**3.2 Керівництво користувача** 33](#_Toc72472055)

[**3.3 Використання веб-застосунку** 34](#_Toc72472056)

[**ВИСНОВКИ** 36](#_Toc72472057)

[**СПИСОК ІНФОРМАЦІЙНИХ ДЖЕРЕЛ** 36](#_Toc72472058)

# **ВСТУП**

Тайм-менеджмент допомагає швидко приймати рішення. Це дуже важливо в сучасному світі, де успіх залежить іноді від того, наскільки швидко ви відреагували на ту чи іншу ситуацію. Управління часом дозволяє контролювати практично всі відбуваються навколо вас процеси, без чого неможливо уявити ефективність діяльності організації. Щоб у вас вистачало часу на сприйняття великої кількості інформації, необхідно раціонально розподіляти свій час. До появ технологій в цьому людям допомагали звичайний листок паперу і ручка. Але зараз є можливість тримати список справ і завдань прямо у себе в смартфоні.

Зараз є величезний вибір програм і сайтів для тайм-менеджменту і списку справ, покупок і завдань. Наприклад, найпопулярнішими є програми Todoist, Microsoft To-Do та Week. Всі ці застосунки допомагають правильно спланувати свій час. Ця організації часу та списку важливих справ буде актуальна в будь-який час.

Метою виконання даної роботи є розробка програми-помічника для програміста. Основне завдання веб-додатку – створення своїх проектів і додавання туди списків завдань / цілей, які потрібно виконати.

На етапі проектування будуть описані методи та етапи проектування, надані UML діаграми. Технології, що будуть використовуватися включають мову Java, сервер NodeJS, React, Reduce, MySQL.

Результатом стане додаток «Project Manager Tool», що являє собою планер завдань проекту для програміста.

# **1 ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ**

## **1.1 Аналіз предметної області**

Предметною областю обраного проекту є застосунок з категорії «Продуктивність» – планер завдань, задач та цілей для проектів.

Актуальність обраної тематики, а саме «продуктивних» застосунків є безперечною. Такі застосунки можна знайти на будь-яку операційну систему із будь-яким дизайном. Функціонал таких застосунків майже однаковий – список завдань із можливістю помічати їх як «виконані» або «в процесі виконання».

Такі прості застосунки не потребують багато часу для розробки. В інтерфейсі програм також легко розібратись – все працює на інтуїтивному рівні.

## **Огляд існуючих рішень за обраною темою**

Огляд існуючих програм є важливою складовою роботи. Аналіз сильних та слабих сторін існуючих аналогів дозволить врахувати їх недоліки та використати переваги при розробці власного застосунку.

При оцінюванні аналогів будемо керуватись наступними критеріями: дизайн,простота і зрозумілість, реєстрація.

### **1.2.1 Todoist**

Простий і красивий онлайн планувальник, який підтримує практично всі мобільні і настільні платформи. В т.ч. працює в поштових програмах і сервісах. Дозволяє спільно працювати над завданнями, отримувати нагадування та оповіщення.

Дизайн застосунку приведено на рисунку 1.1.

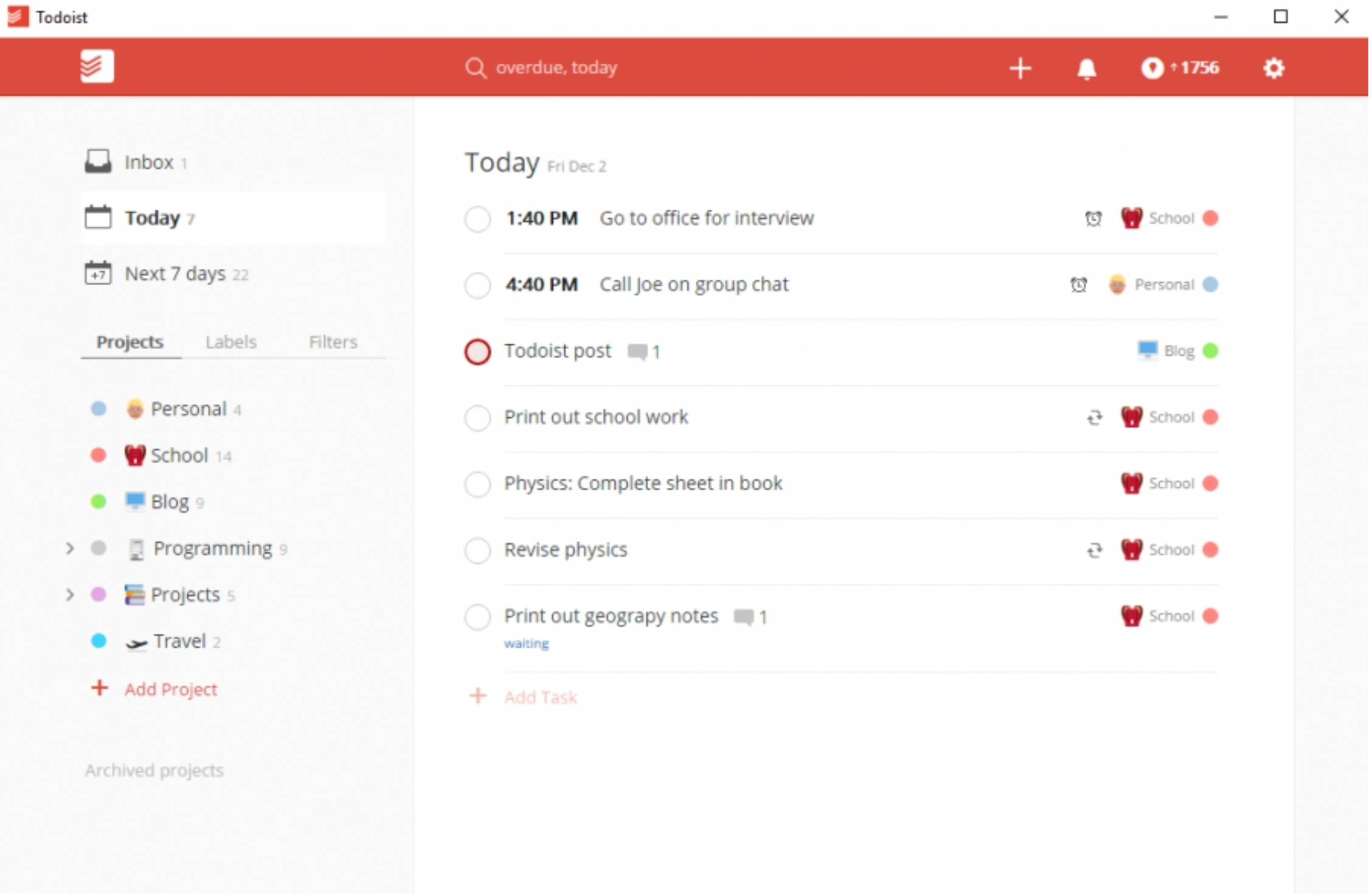


Рис. 1.1 – Застосунок Todoist

### **1.2.2 Microsoft To-Do**

Таск менеджер від Майкрософт. Інтеграція з Office 365. Повний шифрування даних. Інтелектуальні функції для створення завдань. Мобільні додатки.

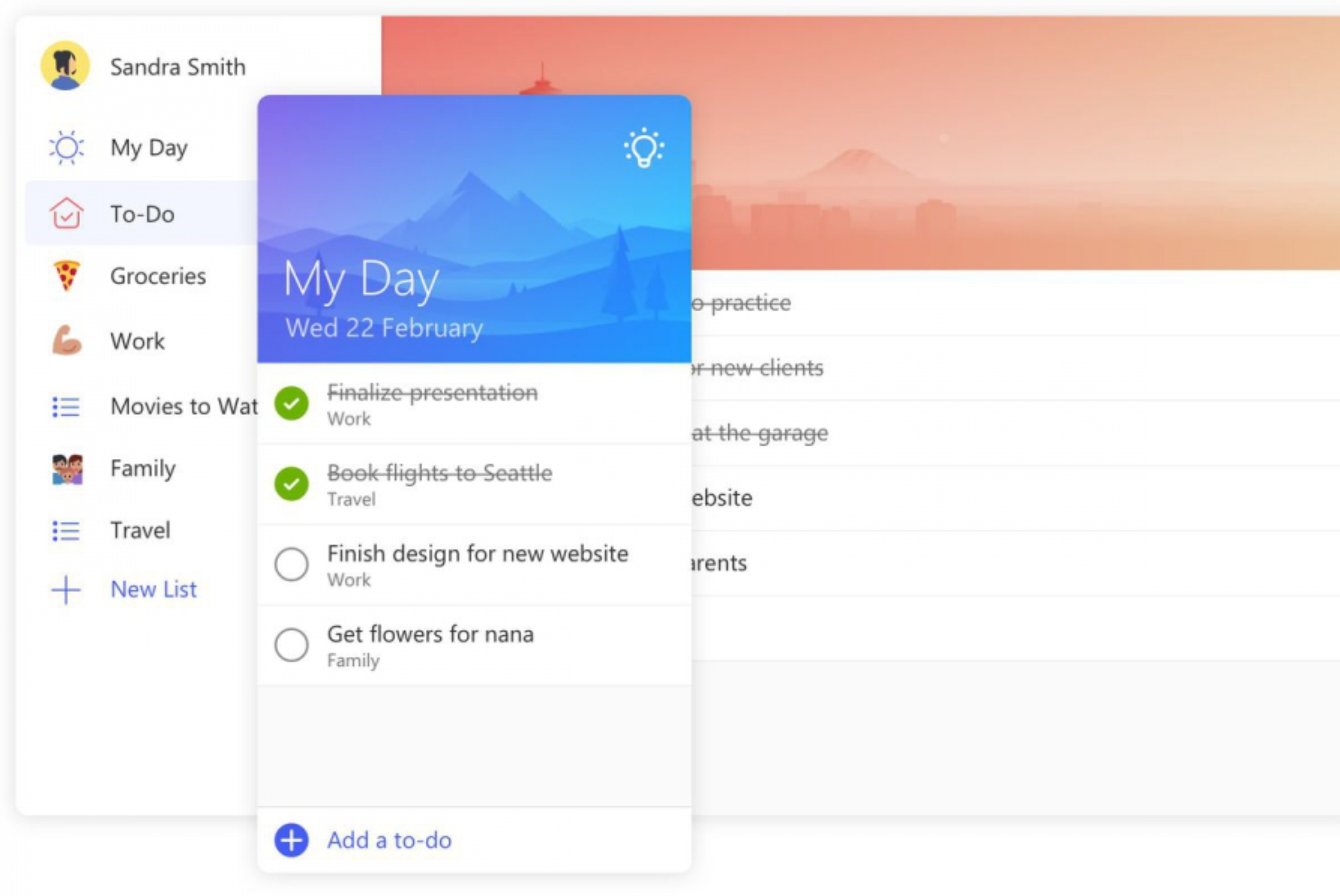


Рис. 1.2. – Застосунок Microsoft To-Do

### **1.2.3 Weeek**

Продумана система управління завданнями і проектами, заснована на канбан-методології. Гнучка структура (Воркспейс → проекти → дошки). Багатоступінчасті завдання. Вбудований тайм-трекер і Pomodoro-таймер. Мобільні додатки на iOS і Android. Є безкоштовна версія з невеликими обмеженнями за кількістю проектів та учасників команди. Російська підтримка.

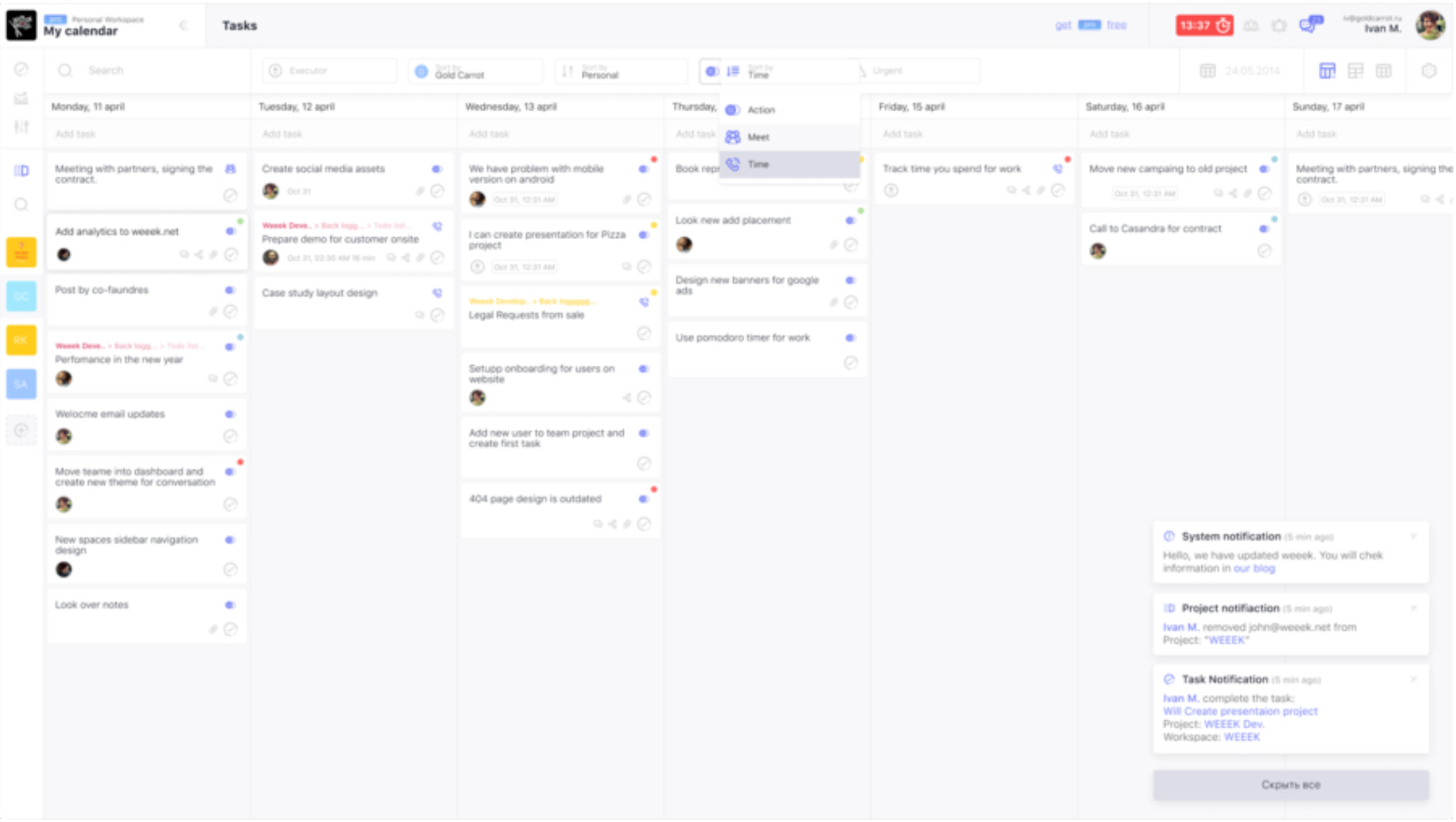


Рис. 1.3 – Застосунок Weeek

## **1.3 Постановка задачі**

Метою виконання курсової роботи є розробка планера на мові Java «Project Manager Tool», що буде доступною з звичайного браузера кінцевому користувачу. На сервері повинні зберігатися база даних користувачів (ім’я, пошта та паролі).

Для досягнення поставленої мети необхідно виконати наступні задачі:

* клієнт-серверний веб-застосунок;
* дані повинні зберігатися у документо-орієнтованій базі даних MySQL;
* реалізація бекенду за допомогою Java;
* реалізація клієнтського додатку до за допомогою розмітки HTML, таблиць стилів CSS, Reduce та React;
* реалізація авторизації користувача – клієнта з обмеженим доступом до бази даних;
* створити основний інтерфейс: реєстрація/логін, додати/видалити проект, додати задачу до проекту, змінити статус задачі;
* розробити документацію виконання курсової роботи.

# **2 СПЕЦИФІКАЦІЯ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ (SRS – SOFTWARE REQUIREMENT SPECIFICATION)**

## **2.1 Загальна інформація щодо веб-застосунку**

У данному разі веб-застосунок – це планер на мові Java. Його суть полягає в організації задач в рамках одного або декількох проектів. Стандартне керування за допомогою клавіатури та миші (або тачпаду).

Мета – розробити простий планер-органайзер задач для проектів. Створення подібної програми повинно закріпити знання ООП, посприяти кращому засвоєнню мови Java у виконавця курсової, стати відправним пунктом на шляху створення та відлаження відносно середніх та великих проектів.

## **2.2 Контексна діаграма**

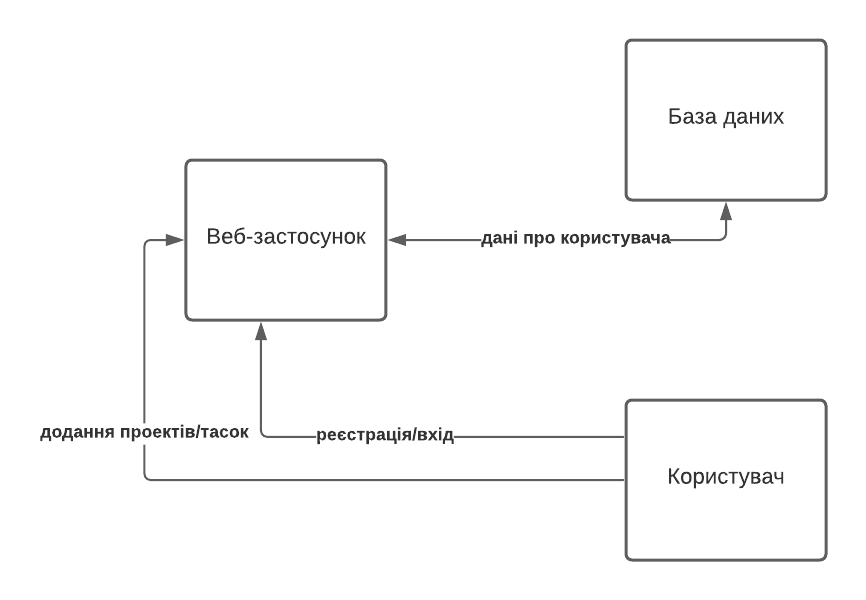


Рис. 2.1 – Контексна діаграма веб-застосунку

## **2.3 Вимоги до веб-застосунку**

### **2.3.1 Функціональні вимоги**

Функціональні вимоги наведені в таблиці 2.1.

Таблиця 2.1 – Опис функціональних вимог до веб-застосунку.

Вимога Опис

Реєстрація.Клієнт Якщо користувач хоче пограти, то він має бути зареєстрованим.

Реєстрація.Клієнт.Дані Для реєстрації клієнт має ввести необхідні дані в реєстраційну форму.

Реєстрація.Клієнт.Завершення Після того як клієнт натисне кнопку «Зареєструватися», система завантажує доступні рівні для гри

Реєстрація.Клієнт.Перевірка Після подання форми реєстрації клієнтом, система має провірити чи не існує даний клієнт в базі даних.

Реєстрація.Клієнт.Підтвердження При невиявлені даного клієнта в базі даних, додати нового клієнта до бази даних з статусом «Реєстрація пройшла успішно».

Реєстрація.Клієнт.Відмова При виявлені даного клієнта в базі даних, завершити транзакцію зі статусом «Даний клієнт уже зареєстрований».

Вхід.Система Для входу до системи, користувач має ввести логін і пароль в форму входу.

Вхід.Система.Підтвердження

При збігу даних завершити транзакцію зі статусом «Вхід виконано успішно».

Вхід.Система.Відмова При розбіжності даних завершити транзакцію зі статусом «Невірні дані»

ОбліковийЗапис.Користувач Клієнт має можливість вийти з власного облікового запису.

Рівень.ЗбереженняЧасу Система повинна дозволяти клієнту, зареєстрованому в системі, розміщувати час затрачений на пройденя кожного рівня гри

Рівень.ЗавантаженняФайліс Якщо користувач виконав вхід до системи та відправив запит на рівень передати потрібні для гри файли (картинки, jsonтощо)

Діаграму варіантів використання наведено на рисунку 2.4

Рисунок 2.4 – Діаграма варіантів використання

### **2.3.2 Вимоги до зовнішнього інтерфейсу**

Застосунок забезпечує доступ до певного функціоналу гри при умові успішної авторизації. Доступ виконується завдяки мережі Інтернет.

В грі повинне бути присутнім основне меню, та меню вибору рівня. В основному меню потрібно розмістити корисну для користувача інформацію, таку як таблиця лідерів, допомога.

### **2.3.3 Нефункціональні вимоги**

Користувачі повинні обов'язково реєструватися для можливості виконання операцій перегляду статистики та гри.

Тільки користувачі, позначені як адміністратори повинні мати змогу змінювати інформацію про інших користувачів (не адміністраторів).

Читабельність коду можна розглядати як важливу складову супроводжуваності. Код повинний бути зручним для аналізу, змін та тестування.

Гра не потребує інсталяції, так як вона буде доступна користувачеві за допомогою мережі інтернет і тому буде мати відмінну швидкість доступу до бази даних та завантаження файлів, не буде сильно завантажувати процесор та оперативну пам'ять.

## **2.3 Опис обраних технологій для розробки веб-застосунку**

### **2.3.1 Об’єктно-орієнтовна мова Java**

Java - об'єктно-орієнтована мова програмування, що розробляється компанією Sun Microsystems з 1991 року і офіційно випущений 23 травня 1995 року. Спочатку нову мову програмування називався Oak (James Gosling) і розроблявся для побутової електроніки, але згодом був перейменований в Java і став використовуватися для написання аплетів, додатків і серверного програмного забезпечення. Три ключові елементи об'єдналися в технології мови Java:

Java надає для широкого використання свої аплети (applets) - невеликі, надійні, динамічні, які не залежать від платформи активні мережеві додатки, що вбудовуються в сторінки Web. Аплети Java можуть налаштовуватися і поширюватися споживачам з такою ж легкістю, як будь-які документи HTML.

Java вивільняє міць об'єктно-орієнтованої розробки додатків, поєднуючи простий і знайомий синтаксис з надійним і зручним в роботі середовищем розробки. Це дозволяє широкому колу програмістів швидко створювати нові програми і нові аплети.

Java надає програмісту багатий набір класів об'єктів для ясного абстрагування багатьох системних функцій, використовуваних при роботі з вікнами, мережею і для введення-виведення. Ключова риса цих класів полягає в тому, що вони забезпечують створення незалежних від використовуваної платформи абстракцій для широкого спектра системних інтерфейсів.

### **2.3.2 База даних MySQL**

MySQL – це реляційна система управління базами даних з відкритим вихідним кодом. В даний час ця СУБД одна з найбільш популярних в веб-додатках - переважна більшість CMS використовує саме MySQL (часто тільки її, без альтернатив), а майже всі веб-фреймворки підтримують MySQL вже на рівні базової конфігурації (без додаткових модулів).

З переваг СУБД MySQL варто відзначити простоту використання, гнучкість, низьку вартість володіння (щодо платних СУБД), а також масштабованість і продуктивність.

MySQL дозволяє зберігати цілочисельні значення зі знаком і беззнакові, довжиною в 1, 2, 3, 4 і 8 байтів, працює із строковими і текстовими даними фіксованої і змінної довжини, дозволяє здійснювати SQL-команди SELECT, DELETE, INSERT, REPLACE і UPDATE, забезпечує повну підтримку операторів і функцій в SELECT- і WHERE- частинах запитів, працює з GROUP BY і ORDER BY, підтримує групові функції COUNT (), AVG (), STD (), SUM (), MAX () і MIN (), дозволяє використовувати JOIN в запитах, в т.ч. LEFT OUTER JOIN і RIGHT OUTER JOIN, підтримує реплікацію, транзакції, роботу з зовнішніми ключами і каскадні зміни на їх основі, а також забезпечує багато інших функціональні можливості.

Гнучкість СУБД MySQL забезпечується підтримкою великої кількості типів таблиць: користувачі можуть вибрати як таблиці типу MyISAM, що підтримують повнотекстовий пошук, так і таблиці InnoDB, що підтримують транзакції на рівні окремих записів. Є й інші типи таблиць, розроблені спільнотою.

### **2.3.3 Мова програмування JavaScript**

JavaScript (JS) – це динамічна, об'єктно-орієнтована прототипна мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується для створення сценаріїв веб-сторінок, що надає можливість на стороні клієнта (пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінки.

JavaScript класифікують як прототипну (підмножина об'єктно-орієнтованої), скриптову мову програмування з динамічною типізацією. Окрім прототипної, JavaScript також частково підтримує інші парадигми програмування (імперативну та частково функціональну) і деякі відповідні архітектурні властивості, зокрема: динамічна та слабка типізація, автоматичне керування пам'яттю, прототипне наслідування, функції як об'єкти першого класу.

### **2.3.4 Серверна платформа Node.js**

Node.js – платформа з відкритим кодом для виконання високопродуктивних мережевих застосунків, написаних мовою JavaScript[8]. Засновником платформи є РаянДал (RyanDahl). Якщо раніше Javascript застосовувався для обробки даних в браузері на сторонні користувача, то node.js надав можливість виконувати JavaScript-скрипти на сервері та відправляти користувачеві результат їх виконання. Платформа Node.js перетворила JavaScript на мову загального використання з великою спільнотою розробників.

Платформа Node.js призначена для виконання високопродуктивних мережевих додатків, написаних мовою програмування JavaScript. Платформа окрім роботи із серверними скриптами для веб-запитів, також використовується для створення клієнтських та серверних програм.

### **2.3.5 Java-Script библиотека React**

React – це декларативна, ефективна і гнучка JavaScript бібліотека для створення користувацьких інтерфейсів. Вона дозволяє вам збирати складний UI з маленьких ізольованих шматочків коду, званих «компонентами».

### **2.3.6 Таблиця стилів CSS**

CSS (англ. Cascading Style Sheets - каскадні таблиці стилів) – технологія опису зовнішнього вигляду документа, оформленого мовою розмітки.

Переважно використовується як засіб оформлення веб-сторінок в форматі HTML і XHTML, але може застосовуватися з будь-якими видами документів в форматі XML, включаючи SVG і XUL.

Каскадні таблиці стилів використовуються творцями веб-сторінок для завдання кольорів, шрифтів, розташування і інших аспектів представлення веб-документа. Основною метою розробки CSS було розділення вмісту (написаного на HTML або іншій мові розмітки) і оформлення документа (написаного на CSS). Це поділ може збільшити доступність документа, надати велику гнучкість і можливість управління його поданням, а також зменшити складність і повторюваність в структурному вмісті. Крім того, CSS дозволяє представляти один і той же документ в різних стилях або методах виведення, таких як екранне уявлення, друк, читання голосом (спеціальним голосовим браузером або програмою читання з екрану), або при виведенні пристроями, що використовують шрифт Брайля.

## **2.4 Опис архітектури веб-застосунку**

Діаграми архитектури проекту були створені за допомогою застосунків IntellijIDEA, WebStorm та DataGrip.

### **2.4.1 Діаграма backend**

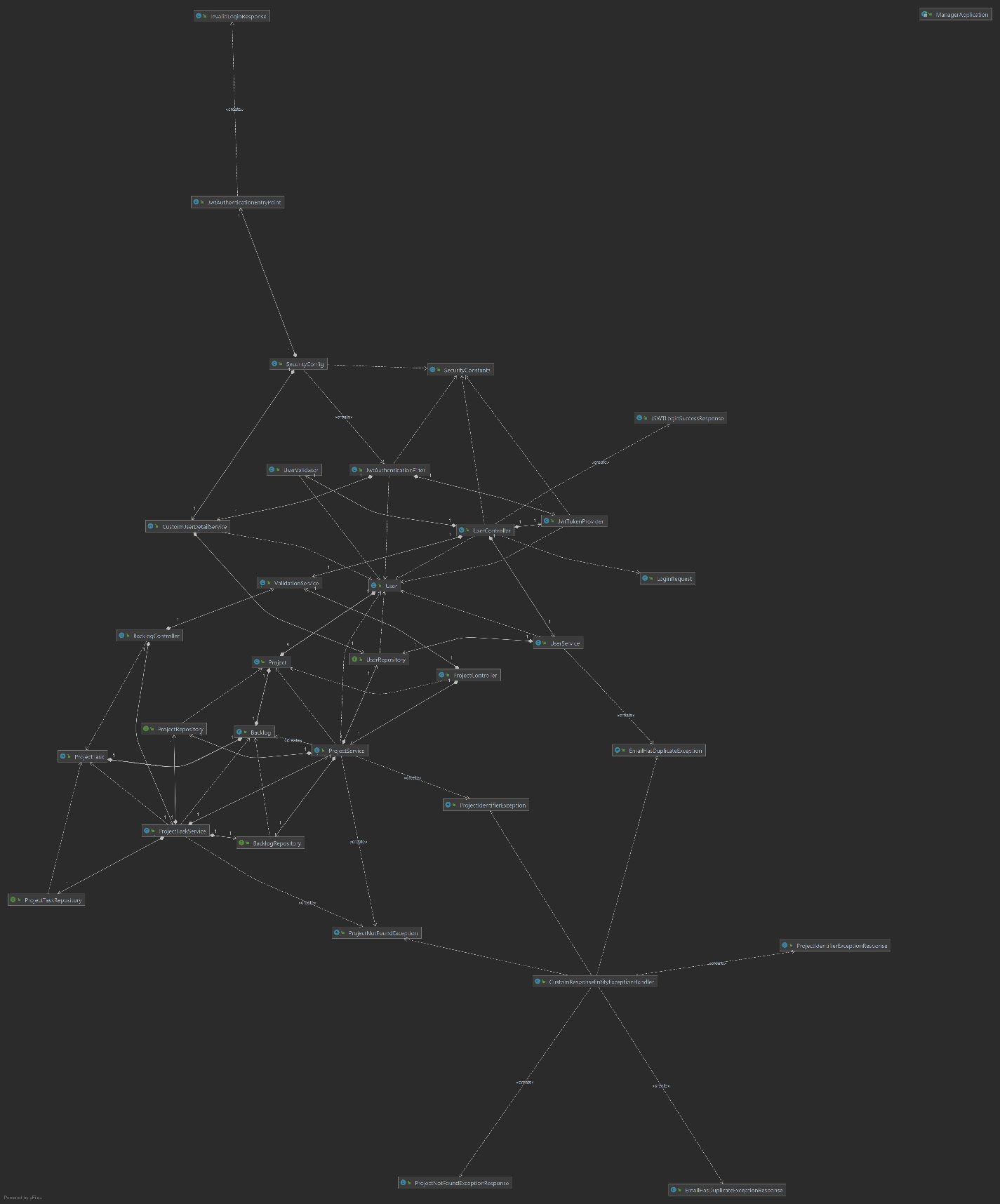


Рис. 2.\* – Бек-енд діаграма

### **2.4.2 Діаграма front-end**

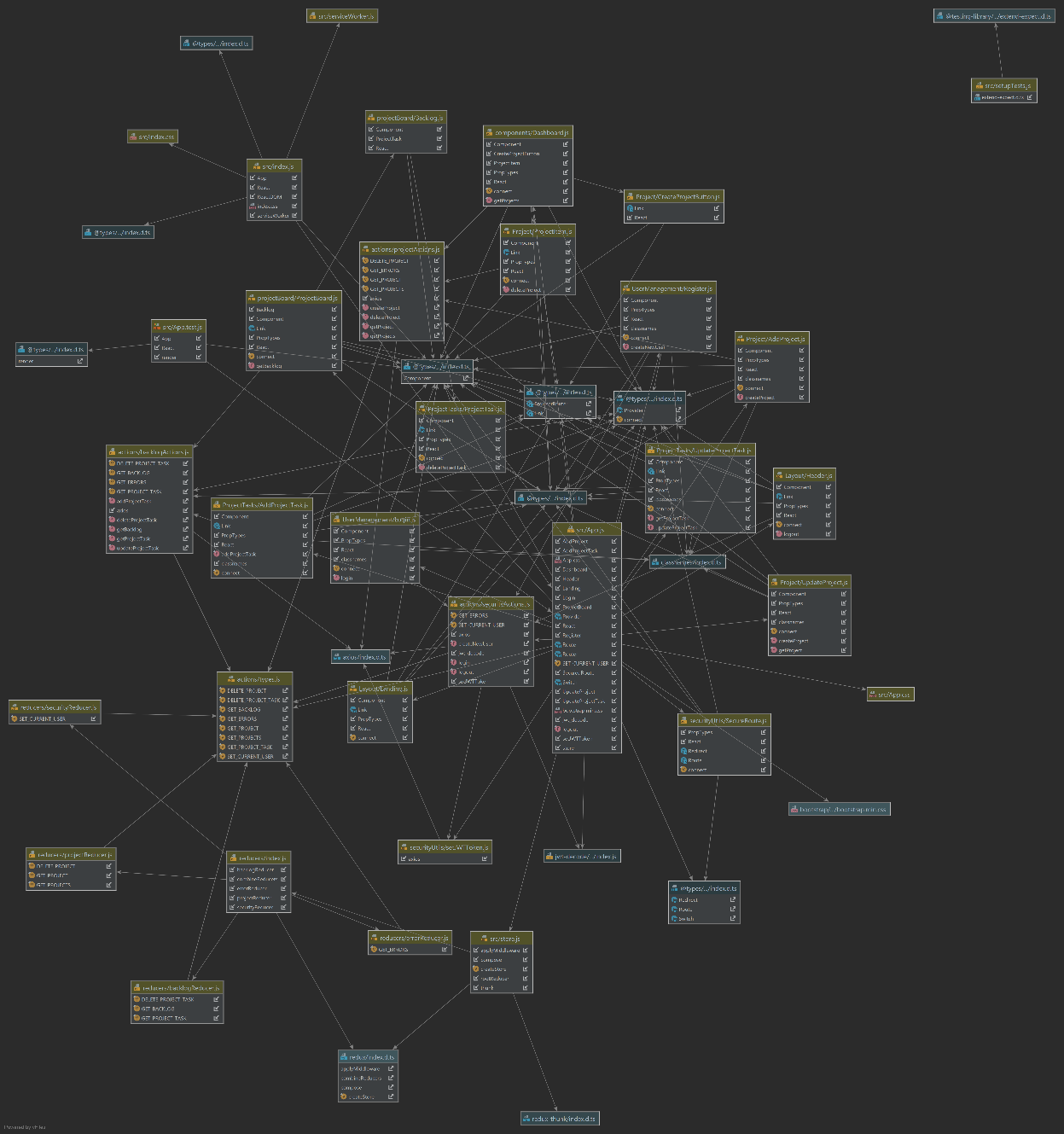


Рис. 2.\* – Фронт-енд діаграма

### **2.5 Модель бази даних**

Нижче представлені структурні таблиці бази даних.

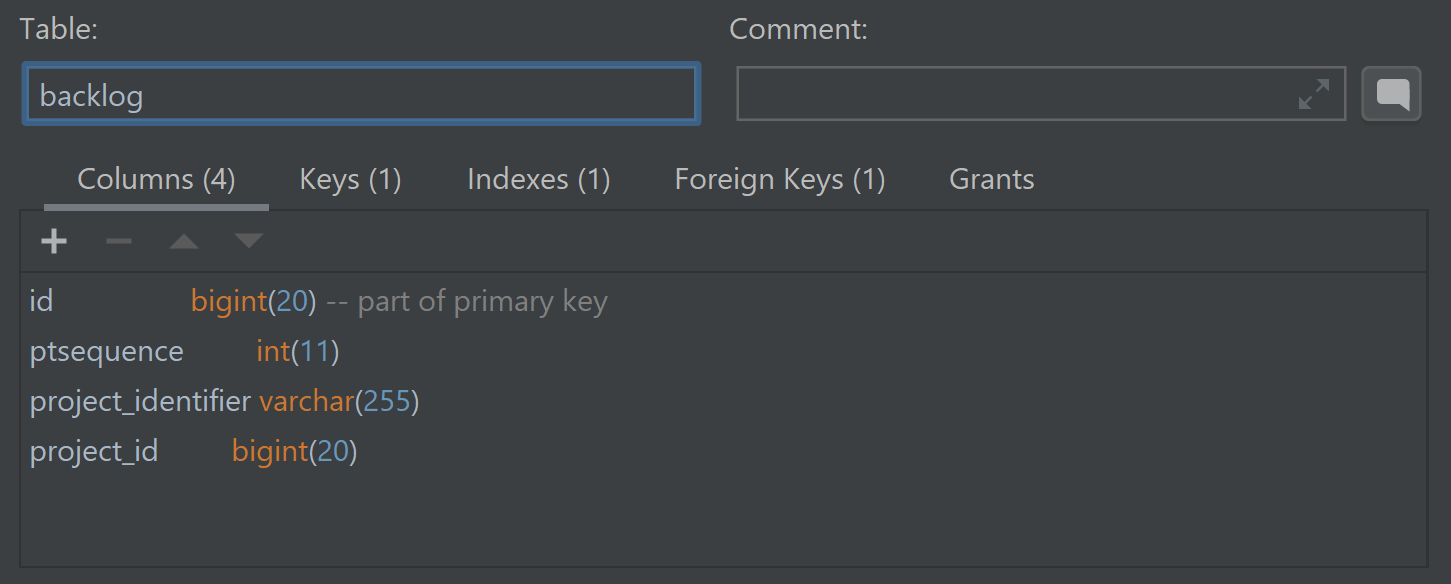


Рис. 2.\* ­– Таблиця backlog

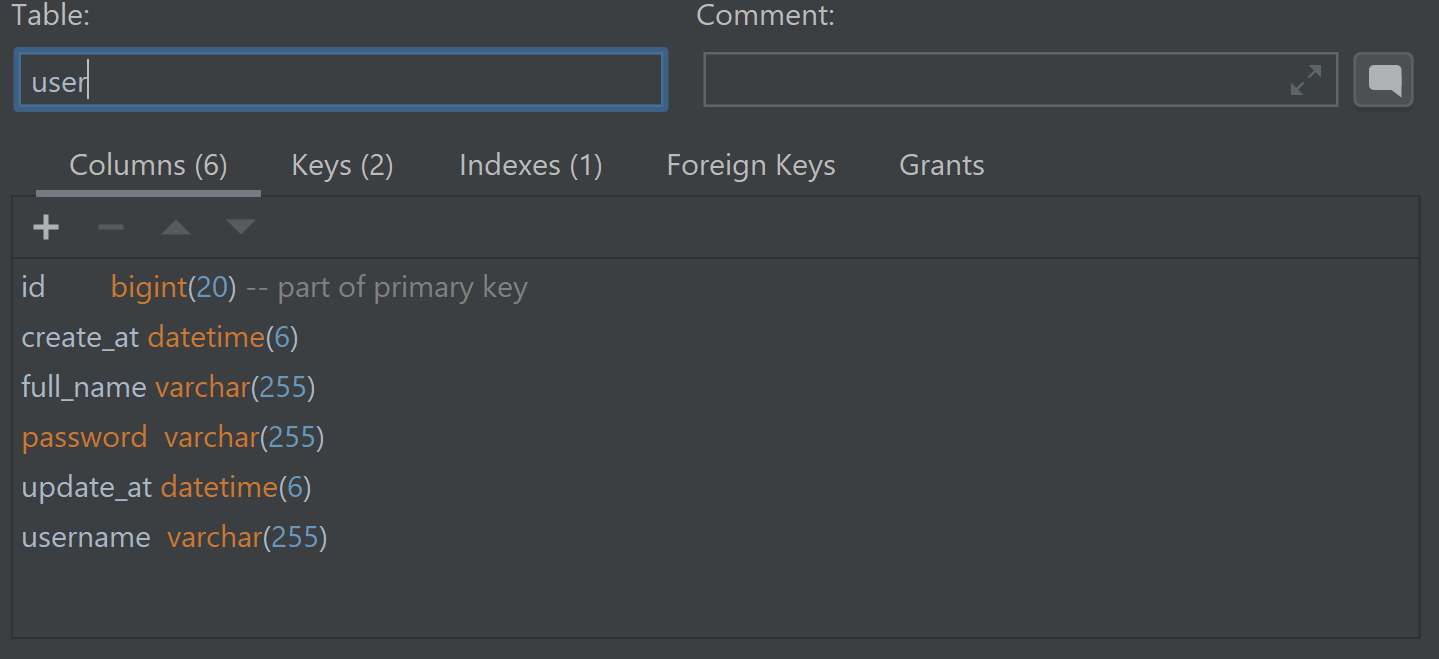


Рис. 2.\* ­– Таблиця user

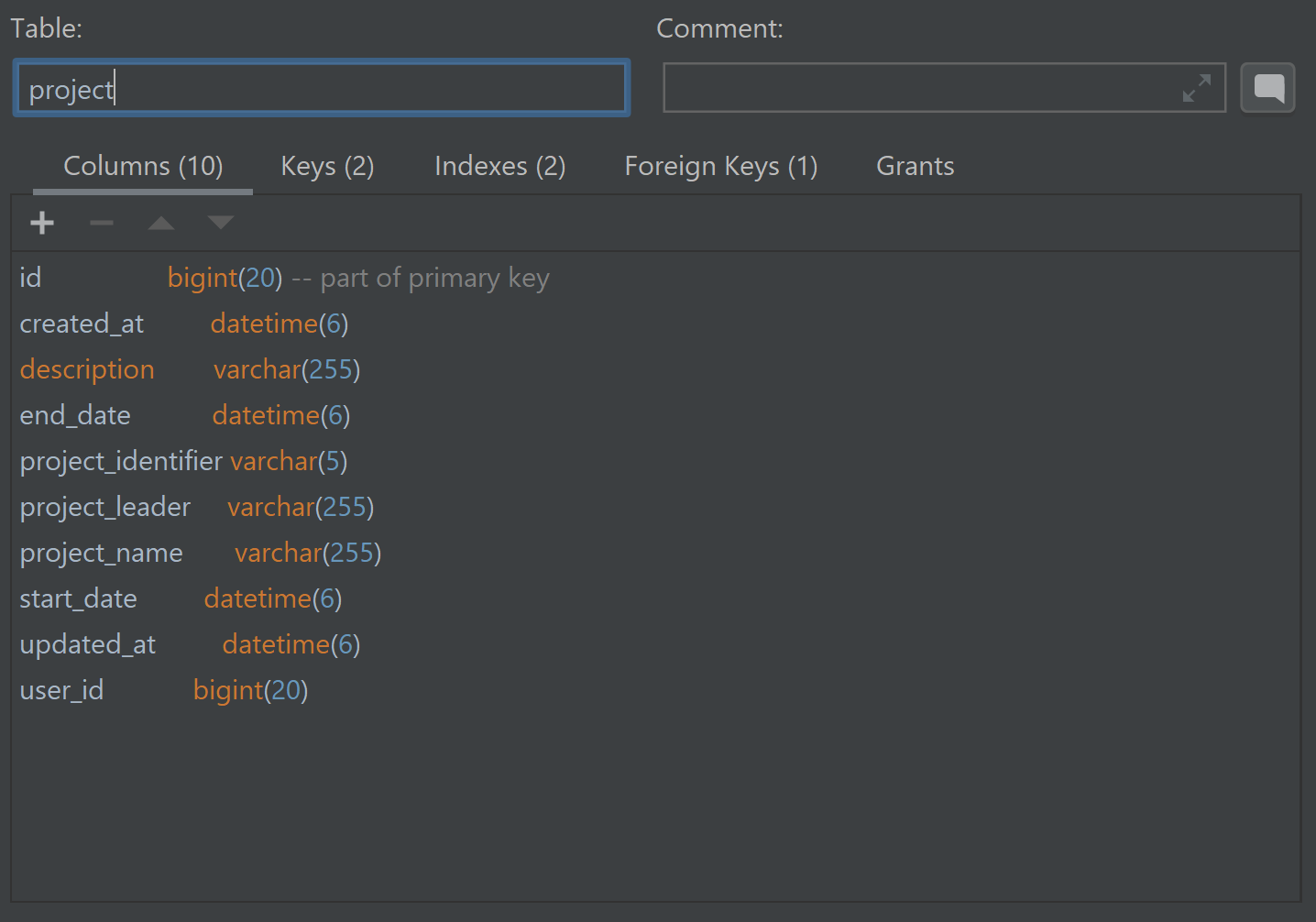


Рис. 2.\* ­– Таблиця project

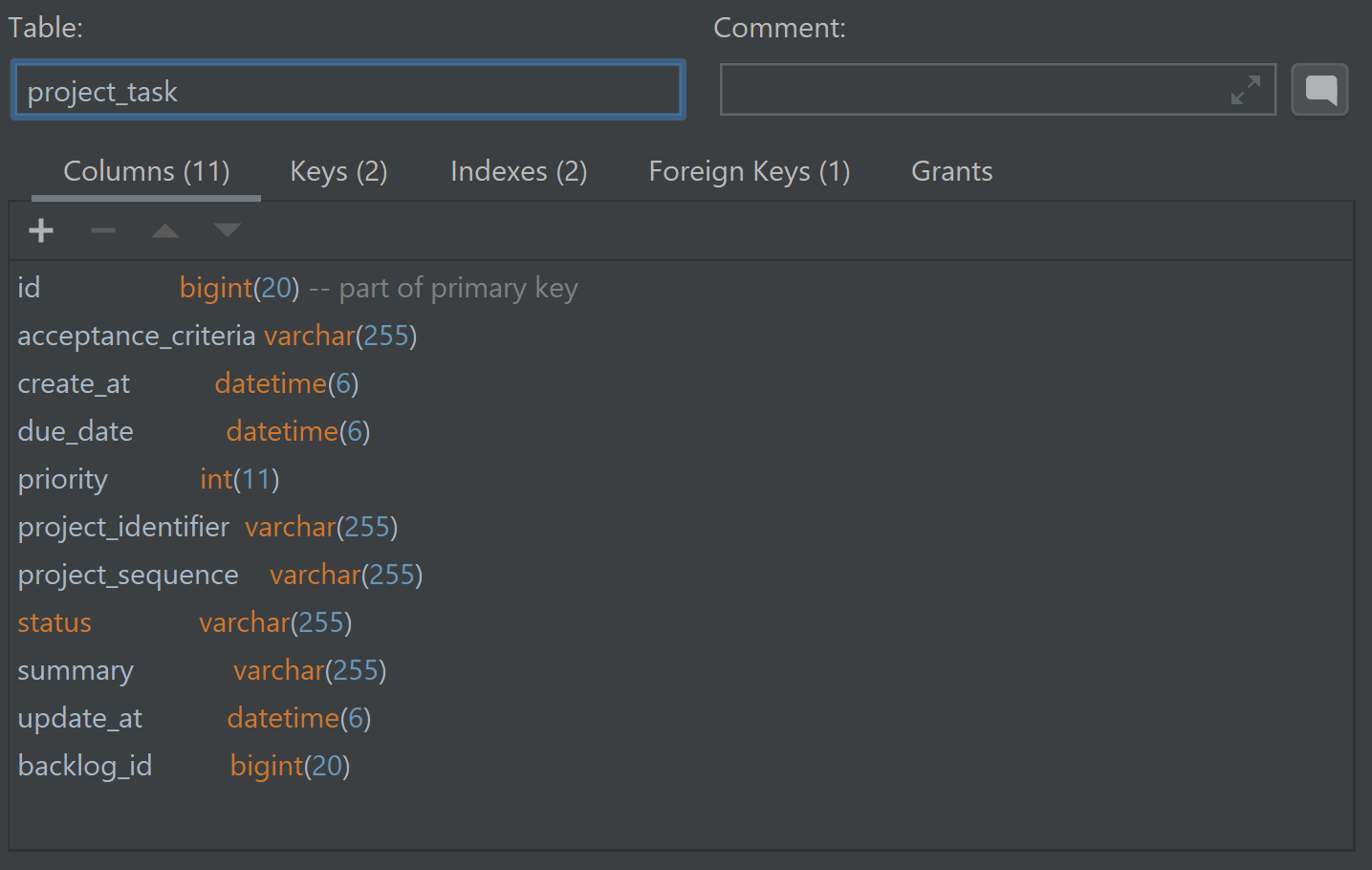


Рис. 2.\* ­– Таблиця project\_task

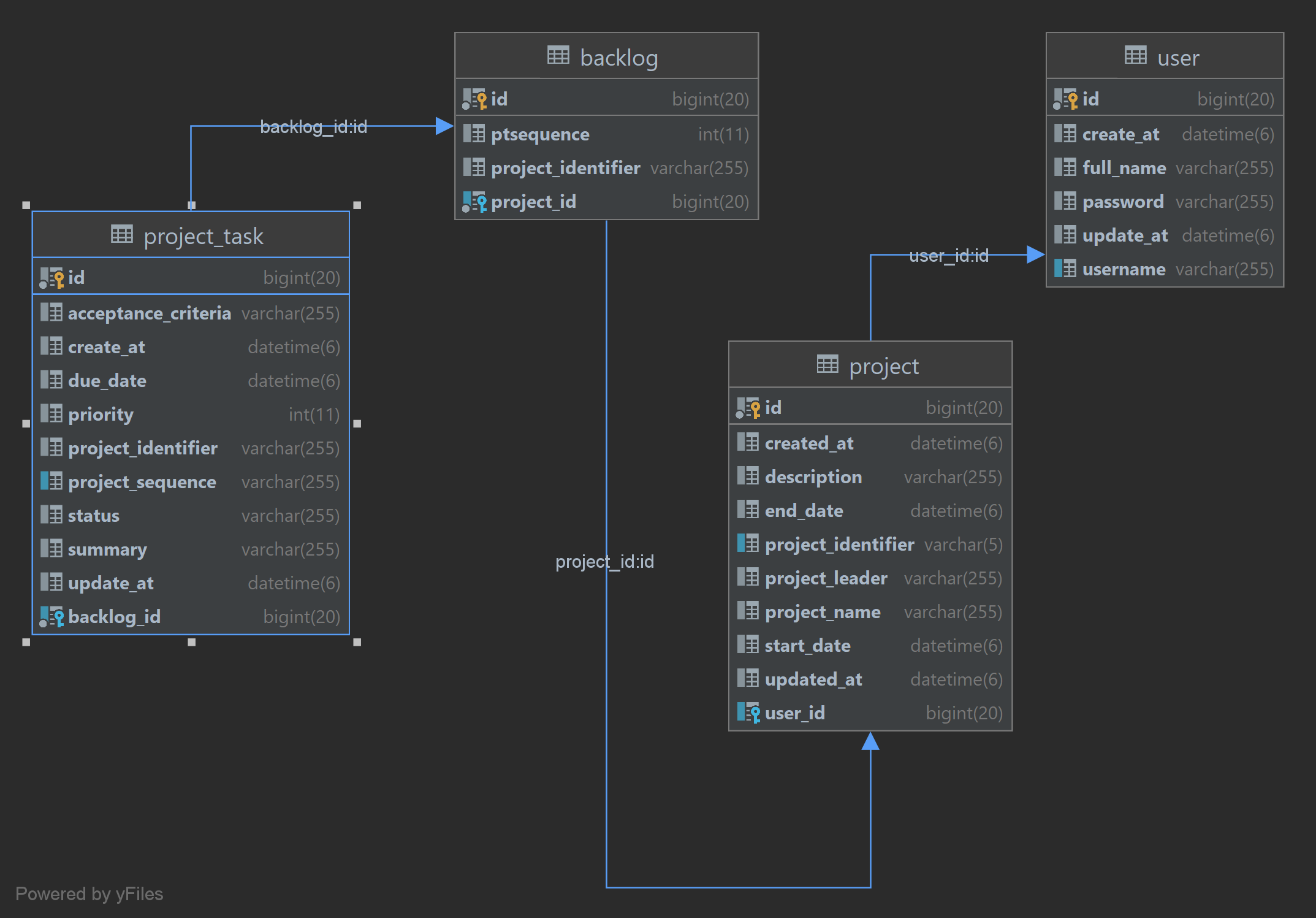


Рис. 2.\* – Діаграма бази даних

# **3 ОПИС ВЕБ-ЗАСТОСУНКУ**

## **3.1 UI застосунку**

Нижче наведений інтерфейс застосунку для користувача.

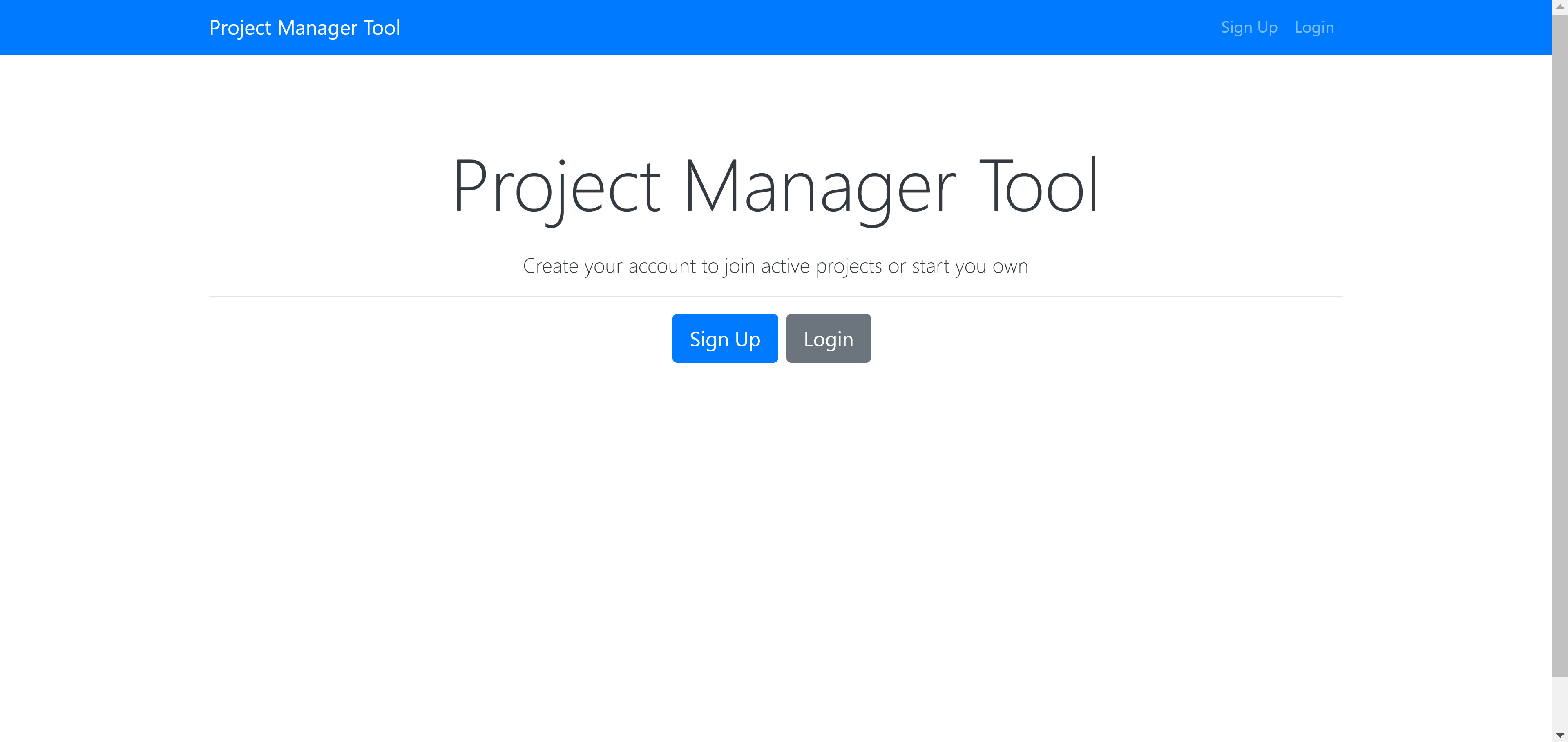


Рис. 3.1 – Головна сторінка веб-застосунку

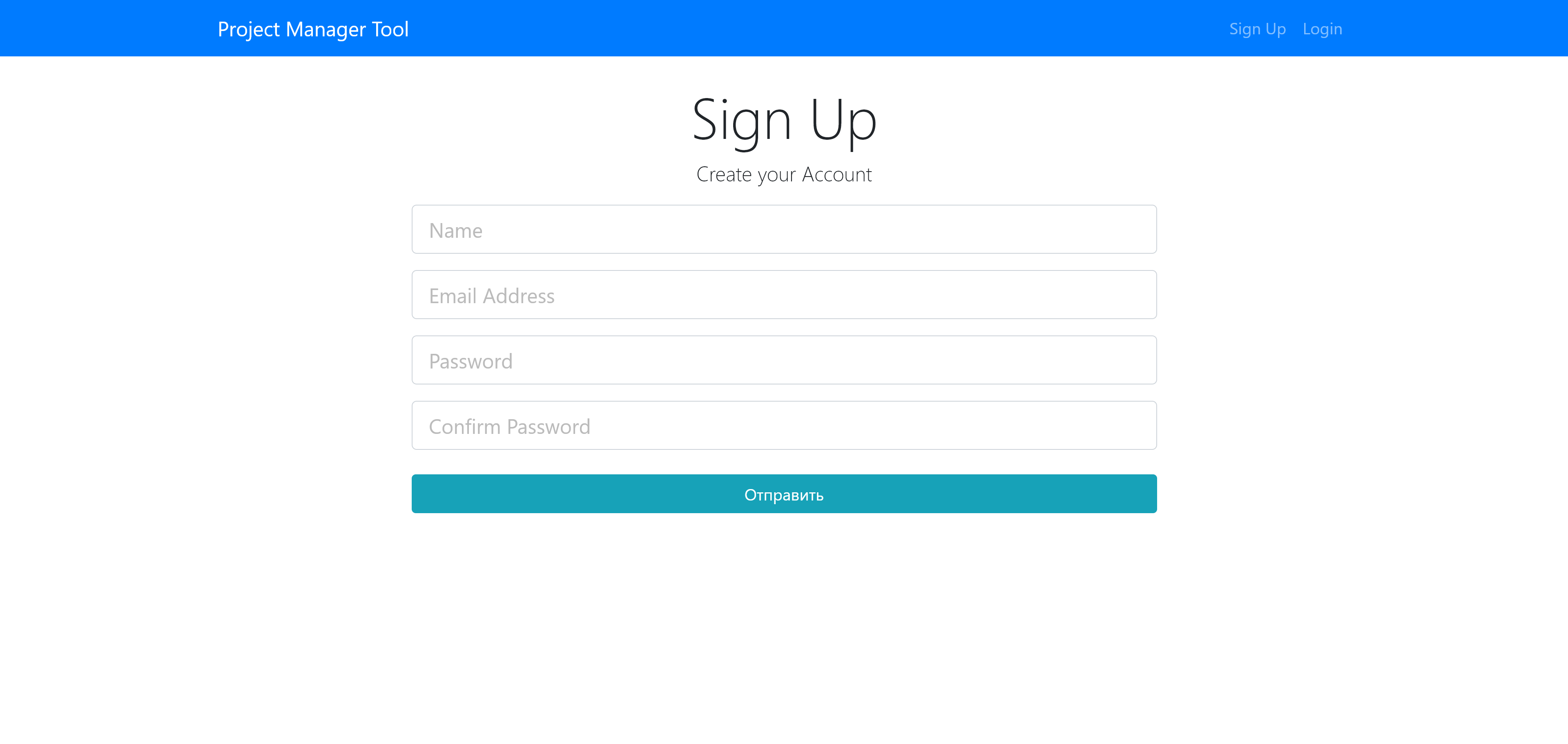


Рис. 3.2 – Сторінка реєстрації

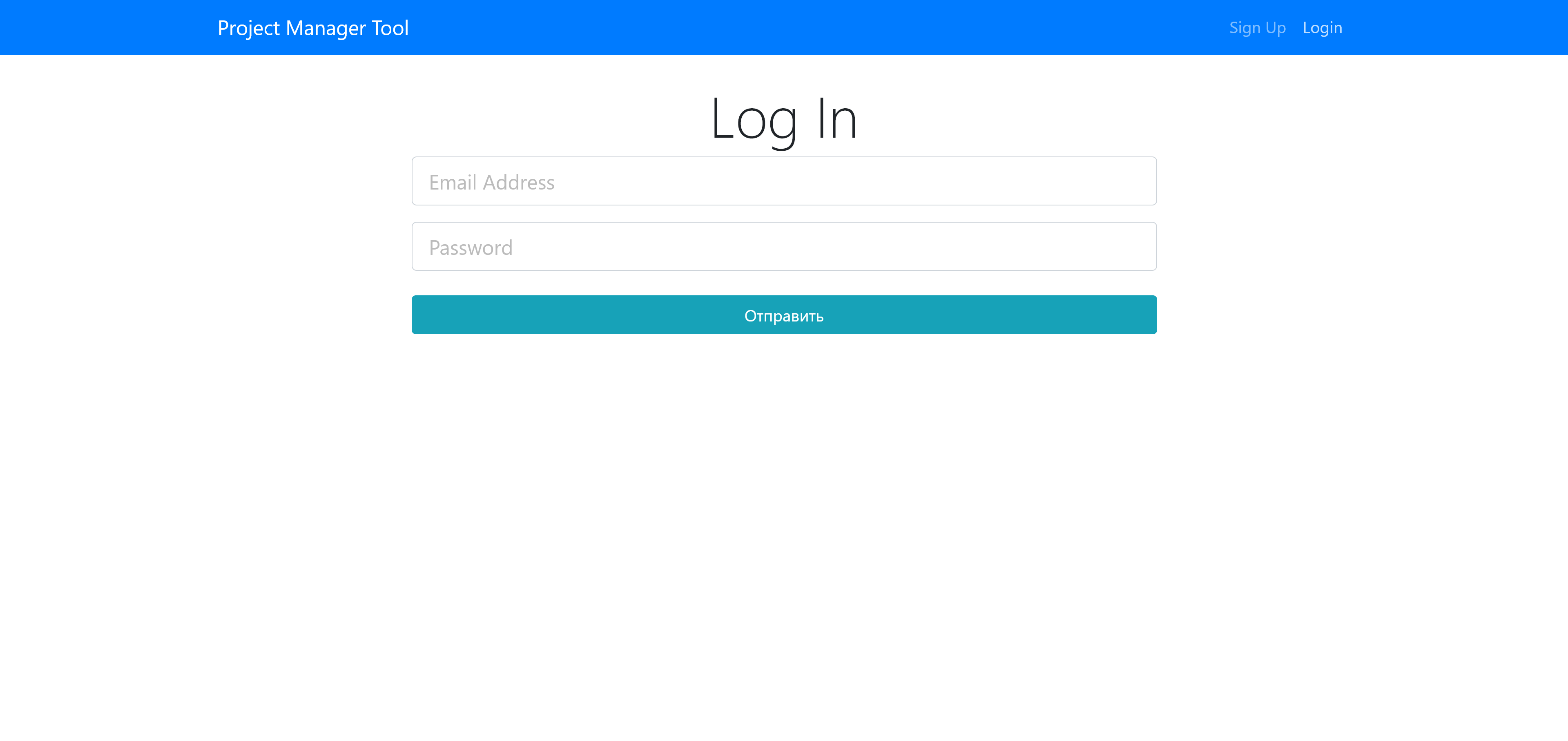


Рис.3.3 – Сторінка входу

## **3.2 Тестування застосунку**

Для тестування роботи застосунку створимо тестовий акаунт із логіном test.



Рис.3.4 – Вхід у тестовий акаунт

Після входу отримуємо сторінку з панеллю проектів, яка зображена на рисунку 3.5. Створюємо новий тестовий проект (рисунки 3.6, 3.7, 3.8) та додаємо до нього задачі із різнем рівнем пріоритету (рис.3.9, 3.10).

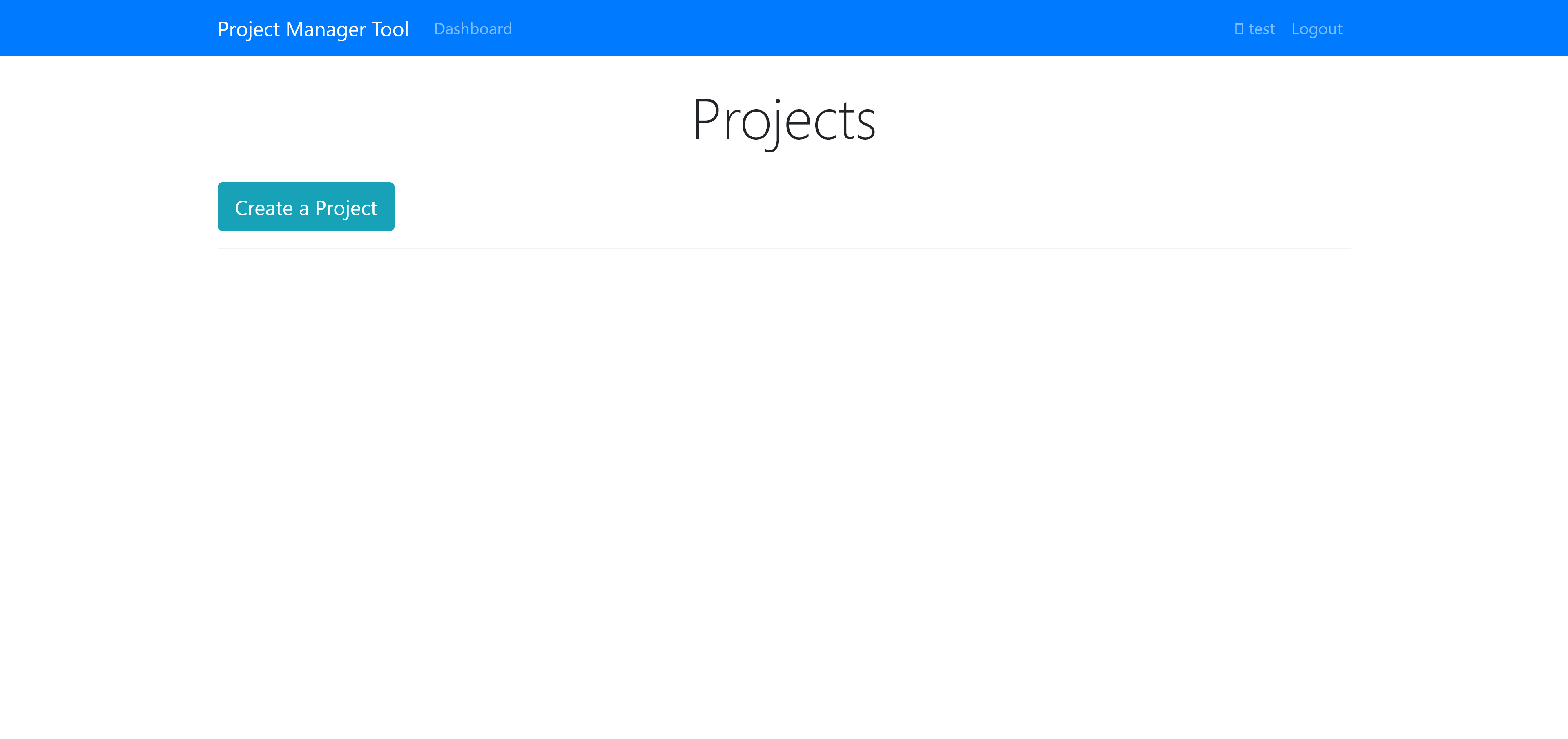


Рис.3.5 – Сторінка панелі проектів

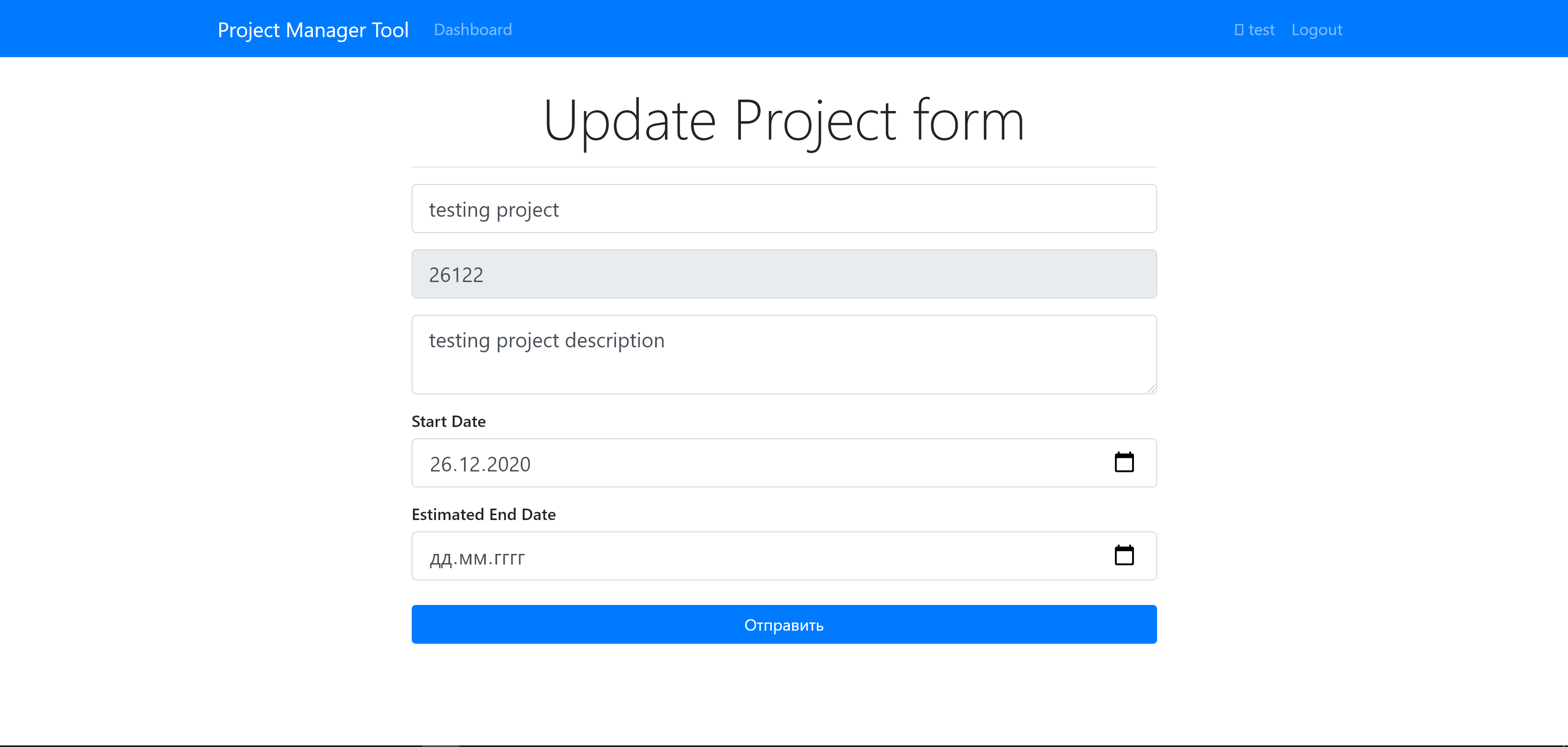


Рис. 3.6 – Створення нового проекту

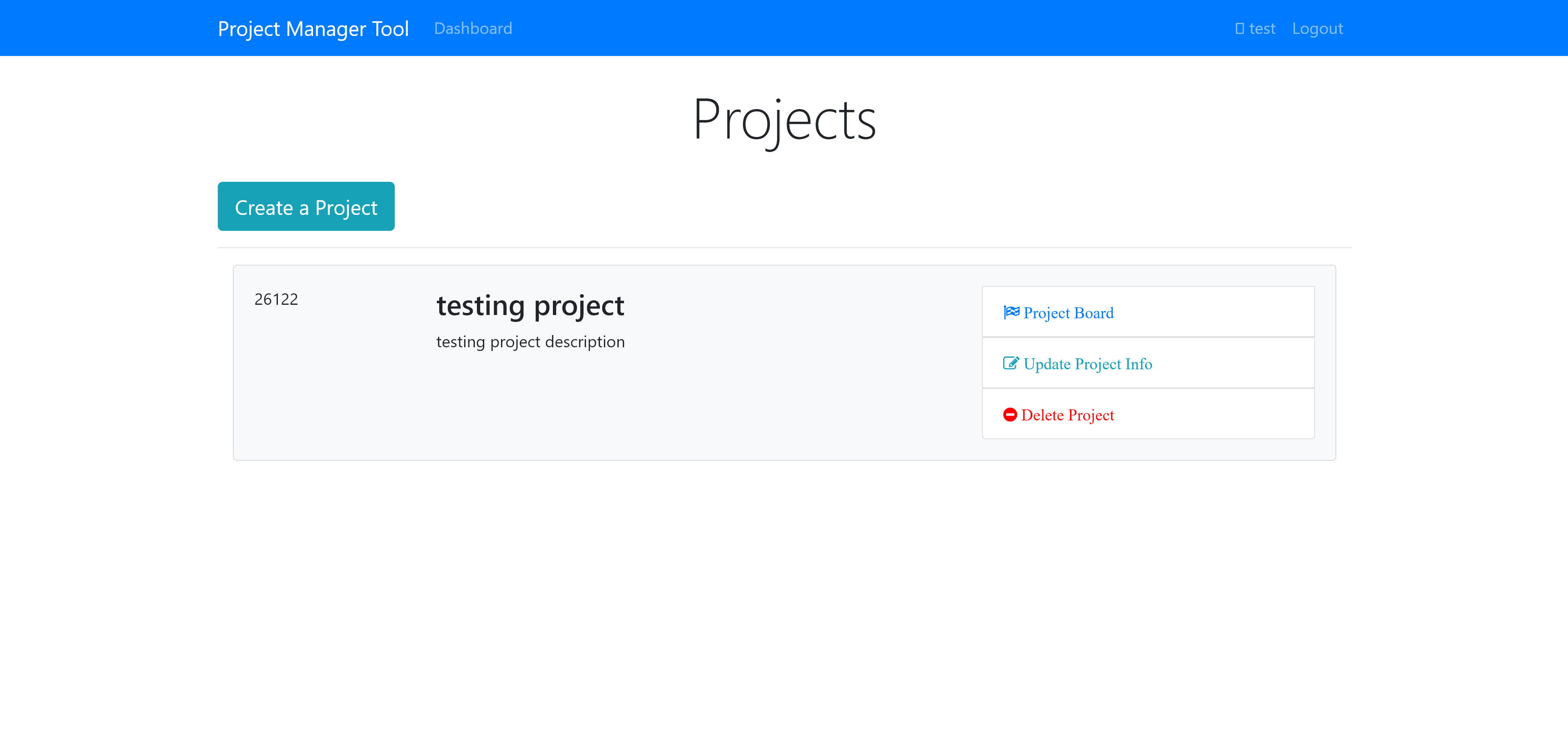


Рис. 3.7. – Панель із тестовим проектом

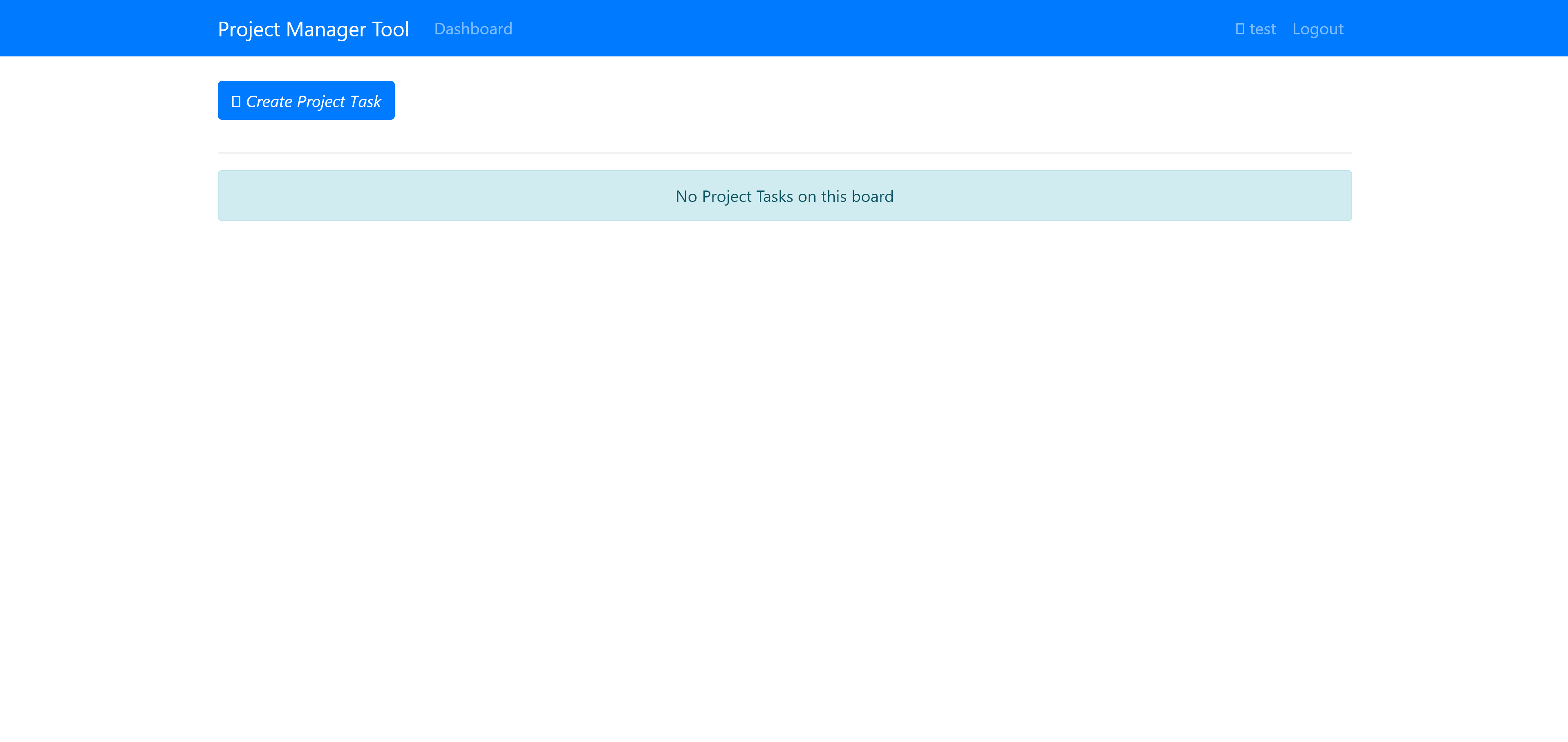


Рис. 3.8. – Проект без доданих задач до нього

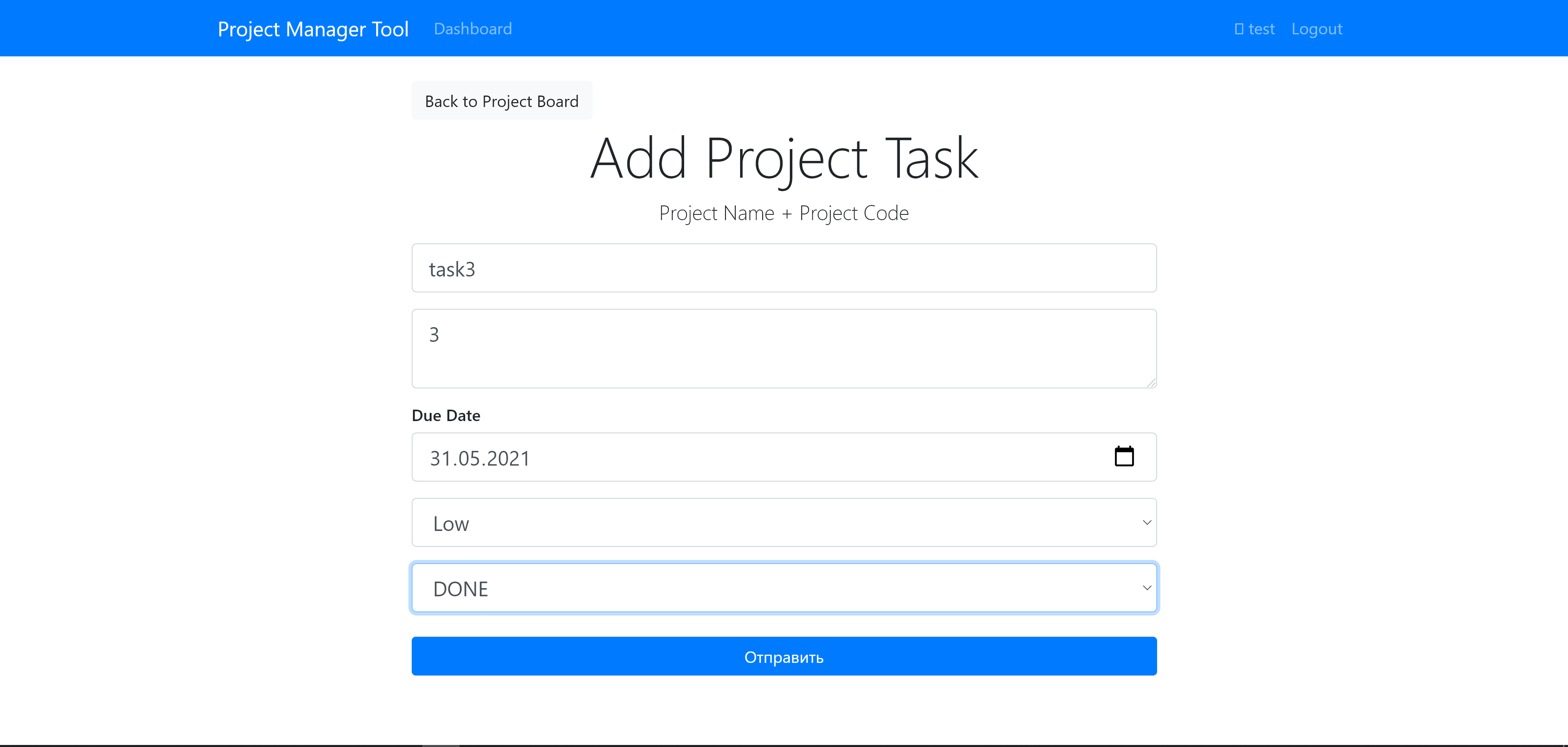


Рис. 3.9 – Створення таски

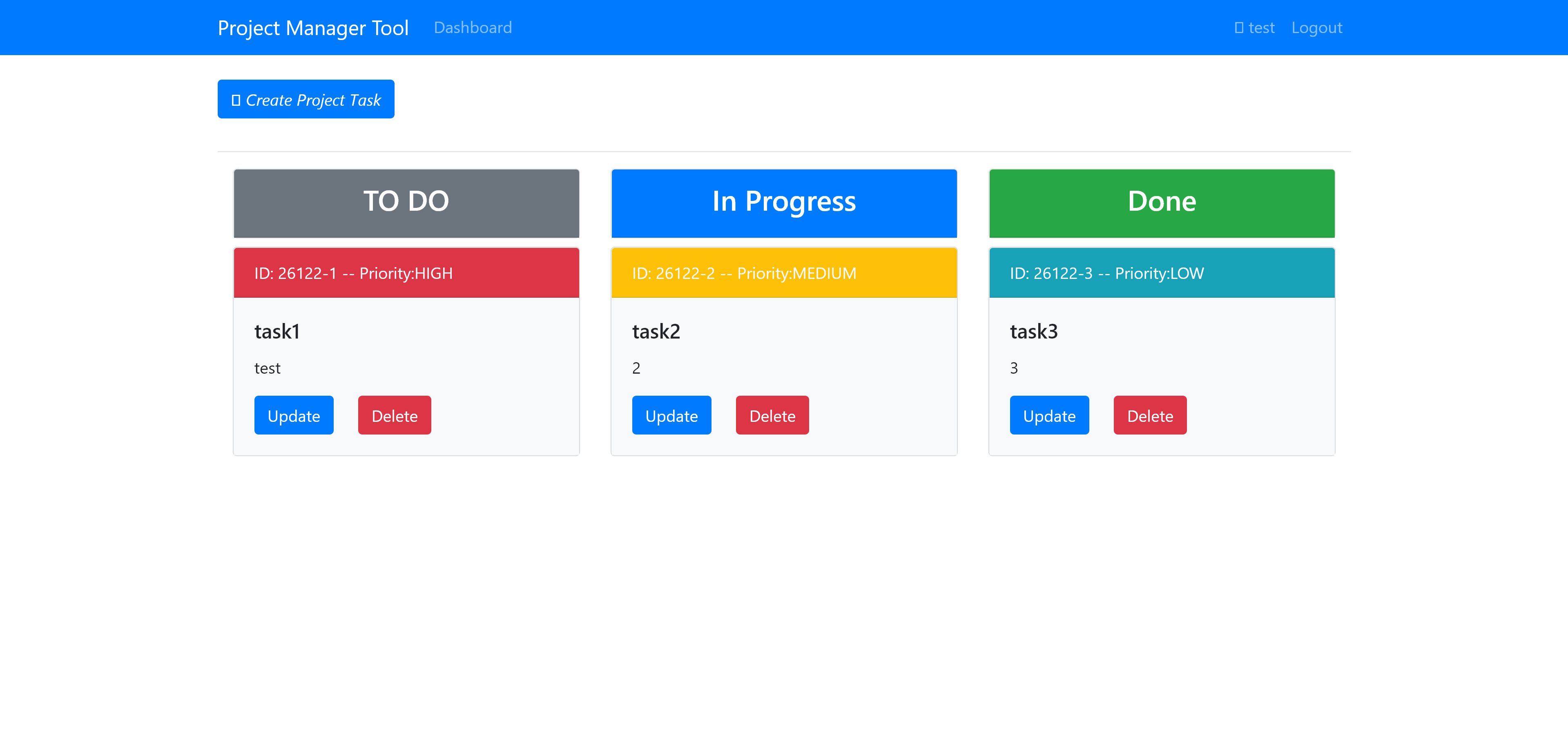


Рис. 3.10 – Тестовий проект із задачами різного рівня пріоритету

# **ВИСНОВКИ**

В ході виконання курсової роботи було виконано наступні етапи.

1) Було проаналізовано предметну область індивідуального завдання курсової роботи.

2) Розроблено модель даних у виді схем у документо-орієнтованій базі даних.

3) Проведено первинну розробку серверної архітектури додатку.

4) Проведено проектування архітектури інформаційної системи з використанням технології «клієнт-сервер».

5) Проведено програмну реалізацію та налагодження розробленої інформаційної системи.

Отже, в ході виконання курсової роботи, послідовно застосовуючи принципи об'єктно-орієнтованого аналізу, документо-орієнтованого моделювання даних, а також використовуючи програмний засіб СУБД MongoDB, середу розробки VisualStudioCode для програмування веб-додатку на серверній платформі NodeJS в операційній системі MS Windows 10, було розроблено, реалізовано і задокументовано гру-платформер на мові програмування JavaSript.

# **СПИСОК ІНФОРМАЦІЙНИХ ДЖЕРЕЛ**

1 MySQL [Електронний ресурс]. – Режим доступу: https://www.mysql.com/ (21.05.2021)

2 Документация по Веб-программированию 0.0.0 [Електронний ресурс]. – Режим доступу: https://iit-web-lectures.readthedocs.io/ru/latest/www/css.html (21.05.2021)

3 MySQL — система управления базами данных [Електронний ресурс]. – Режим доступу: https://web-creator.ru/articles/mysql (21.05.2021)

4 Энциклопедия языков программирования [Електронний ресурс]. – Режим доступу: http://progopedia.ru/language/java/ (21.05.2021)

5 Бизнес-клас [Електронний ресурс]. – Режим доступу: http://www.classs.ru/career/articles1/article4.html (21.05.2021)

6 LiveBusiness [Електронний ресурс]. – Режим доступу: https://www.livebusiness.ru/tool/1483/ (21.05.2021)