**МІНІСТЕРСТВООСВІТИІНАУКИУКРАЇНИ**

**ЛЬВІВСЬКИЙНАЦІОНАЛЬНИЙУНІВЕРСИТЕТІМЕНІІВАНА**

**ФРАНКА**

Факультет прикладної математики та інформатики

(повне найменування назва факультету)

Кафедра інформаційних систем

(повна назва кафедри)

**ДИПЛОМНА РОБОТА на тему:**

РОЗРОБКА ПЛАТФОРМИ ДЛЯ СПІЛЬНОТВОРУ ІДЕЙ

Студента ІV курсу, групи ПМі-42 ,

спеціальності 122 – комп’ютерні науки

Янішевського Д. І.

(прізвище та ініціали)

Керівник доцент кафедри інформаційних

систем, кандидат фіз.-мат. наук Стельмащук В. В.

(посада, прізвище та ініціали)

Національна шкала \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_\_\_\_Оцінка: ECTS

\_\_\_\_\_

Львів – 2025

**Зміст**

[**ВСТУП 4**](#_Toc197775467)

[**РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ, ОГЛЯД ВІДОМИХ РІШЕНЬ ТА 5**](#_Toc197775468)

[**ОПИС ВИКОРИСТАНИХ ТЕХНОЛОГІЙ 5**](#_Toc197775469)

[**1.1 Постановка задачі 5**](#_Toc197775470)

[**1.2 Огляд відомих рішень 6**](#_Toc197775471)

[**1.3 Методики реалізації та опис використаних технологій 7**](#_Toc197775472)

[**розробки програм 7**](#_Toc197775473)

[**1.3.1 FastAPI 8**](#_Toc197775474)

[**1.3.2 FastAPI-Users 9**](#_Toc197775475)

[**1.3.3 Vue.js 9**](#_Toc197775476)

[**1.3.4 PostgreSQL 10**](#_Toc197775477)

[**1.3.5 GitHub 11**](#_Toc197775478)

[**1.3.6 Axios 11**](#_Toc197775479)

[**1.3.7 SQLAlchemy 12**](#_Toc197775480)

[**1.3.8 Pydantic 12**](#_Toc197775481)

[**1.3.9 Шарова архітектура 12**](#_Toc197775482)

[**РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ 14**](#_Toc197775483)

[**ДОДАТКУ ВІДПОВІДНО ДО ВИМОГ 14**](#_Toc197775484)

[**2.1 Загальний опис та характеристика продукту 14**](#_Toc197775485)

[**2.2 Діаграми проекту 15**](#_Toc197775486)

[**2.3 База даних проекту 17**](#_Toc197775487)

[**РОЗДІЛ 3. СЦЕНАРІЙ РОБОТИ ВЕБЗАСТОСУНКУ 20**](#_Toc197775488)

[**ВИСНОВКИ 31**](#_Toc197775489)

[**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 32**](#_Toc197775490)

# ВСТУП

У сучасному світі, який постійно змінюється, інновації стали одним з ключових факторів успішного розвитку бізнесу, технологій та науки. Ефективний спосіб генерації та реалізації нових ідей полягає не лише в креативності, а й в колективності, швидкий зворотній зв’язок та спільна робота над удосконаленням ідей. Саме ці умови забезпечує підхід спільнотвору — залучення великого кола учасників до створення, оцінки та удосконалення ідей.

Метою цієї дипломної роботи є розробка вебзастосунку, який стане зручним інструментом для спільнотворення ідей стартапів. Цей застосунок забезпечить зручний інтерфейс для взаємодії користувачів, можливість створення стартапів, додавання завдань для них, а також хорошу комунікацію за допомогою коментарів та чату. В рамках проекту також передбачена система рейтингу, що дозволяє оцінювати внесок та якість виконання завдань користувачем.

Розробка цієї платформи спрямована на підтримку нової діяльності, полегшення співпраці між учасниками різного рівня підготовки та створення середовища для народження нових ідей, що мають потенціал для подальшого розвитку в реальні стартапи.

# РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ, ОГЛЯД ВІДОМИХ РІШЕНЬ ТА

# ОПИС ВИКОРИСТАНИХ ТЕХНОЛОГІЙ

## 1.1 Постановка задачі

Завданням дипломної роботи є проектування вебаплікації, для спільнотвору ідей стартапів. Метою проєкту створення інтуїтивно зрозумілої онлайн-платформи, яка дозволяє користувачам ділитися стартап-ідеями, додавання завдань до проєктів, пропонувати нові ідеї та цікаві рішення у коментарях, брати участь у виконанні завдань, мати можливість особистого діалогу з власником завдання. Система має сприяти спільному розвитку проєктів за допомогою інтерактивної взаємодії між користувачами платформи.  
Таким чином, вебзастосунок має виконувати роль середовища для генерації, обговорення та розвитку інновацій, залуча.чи широке коло зацікавлених користувачів до спільної роботи над проєктами цікавими для них.

Специфікація ставить наступні вимоги перед програмним продуктом:

а) створення, редагування та перегляд стартапів

б) додавання та редагування завдань до стартапів

в) реалізація системи коментарів до стартапів

г) реалізація чату для обговорення в межах завдання

ґ) Авторизація та реєстрація користувачів

д) система оцінювання виконаних завдань

е) обмеження функціоналу для неавторизованих користувачів

є) пошук стартапів за назвою

## 1.2 Огляд відомих рішень

*IdeaScale*

**Домашня сторінка:** <https://ideascale.com>

IdeaScale — це хмарна платформа для управління ідеями, яка дозволяє компаніям, урядам та організаціям збирати, аналізувати й реалізовувати ідеї зсередини чи ззовні. Користувачі можуть додавати ідеї, коментувати, голосувати, а також проходити через етапи фільтрації та реалізації.

Недоліки:

а) система більше орієнтована на корпоративне або державне використання, має складну структуру;

б) відсутня можливість брати завдання на виконання чи вести особисте спілкування між учасниками;

в) закритість системи — більшість прикладів проєктів недоступна широкій спільноті.

*OpenIDEO*

Домашня сторінка: https://openideo.com

OpenIDEO — глобальна спільнота, що вирішує соціальні виклики за допомогою спільновірного підходу. Платформа пропонує «виклики», на які учасники подають ідеї, а потім разом удосконалюють їх у відкритому форматі.

Недоліки:

а) акцент переважно на соціальних і глобальних проблемах, а не стартапах;

б) обмежена можливість структурованого розвитку проєктів (наприклад, немає завдань та їх виконання);

в) процес є більш ідеологічним, ніж техніко-реалізаційним

*HeroX*

Домашня сторінка: https://www.herox.com

HeroX — платформа, на якій організації створюють змагання для вирішення складних проблем. Учасники змагаються за грошові призи, подаючи свої ідеї чи прототипи.

Недоліки:

а) орієнтація на конкурси, а не на сталу співпрацю та розвиток ідей;

б) немає структури взаємодії в рамках одного проєкту після завершення конкурсу;

в) обмежена соціальна складова: відсутній фокус на спільноту та постійну участь.

## 1.3 Методики реалізації та опис використаних технологій

## розробки програм

Для розробки програми були використані сучасні технології та методики, що дозволили створити ефективну та зручну вебзастосунок.

Однією з основних технологій бекенду було використано FastAPI, який забезпечує швидку та ефективну розробку REST API на мові програмування Python. FastAPI відзначається високою продуктивністю та автоматичною генерацією документації за стандартом OpenAPI.

Для автентифікації користувачів та управління ролями була використана бібліотека FastAPI-Users, яка надає готові рішення для автентифікації, реєстрації, зміни паролю тощо, що значно спрощує розробку системи користувачів.

Фронтенд програми було реалізовано з використанням Vue.js - сучасного JavaScript-фреймворка для побудови інтерфейсів користувача. Vue.js надає простоту та гнучкість у розробці динамічних вебсторінок та додатків, а також дозволяє створювати компоненти для реюзу в різних частинах додатку.

Для зберігання та управління даними була обрана PostgreSQL - потужна та надійна система керування базами даних з високою продуктивністю та розширюваністю.

Використання таких технологій дозволило створити програму, яка відповідає сучасним вимогам щодо швидкості, безпеки та зручності використання.

### 1.3.1 FastAPI

FastAPI - це високопродуктивний фреймворк для створення вебзастосунків на мові програмування Python. Його особливість полягає в швидкості розробки та автоматичній генерації документації за стандартом OpenAPI. FastAPI дозволяє створювати REST API з мінімальними зусиллями завдяки декларативному підходу до опису маршрутів та валідації даних.

Цей фреймворк відрізняється високою продуктивністю завдяки використанню асинхронного програмування та вбудованої підтримки для корутин. Він підтримує автоматичну генерацію документації на основі анотацій функцій, що дозволяє автоматично створювати та оновлювати документацію для API.

FastAPI забезпечує інтеграцію з різними інструментами, такими як бази даних, аутентифікація користувачів та системи керування сесіями. Він також має вбудовану підтримку аутентифікації через стандартні механізми, такі як OAuth2 та JWT.

FastAPI дозволяє створювати ефективні вебзастосунки з мінімальними зусиллями, що робить його популярним вибором для розробки сучасних та швидких API.

### 1.3.2 FastAPI-Users

FastAPI-Users - це бібліотека для FastAPI, яка надає готові рішення для автентифікації, авторизації та управління користувачами в вебзастосунках. Ця бібліотека спрощує процес розробки системи користувачів, забезпечуючи гнучкість та безпеку.

FastAPI-Users забезпечує можливість налаштування різних методів автентифікації, включаючи аутентифікацію за допомогою електронної пошти та пароля, аутентифікацію через соціальні мережі, такі як Google або Facebook, а також інші способи. Вона також підтримує ролі користувачів та можливість обмеження доступу до ресурсів залежно від цих ролей.

Ця бібліотека має вбудовану підтримку зберігання хешованих паролів, що забезпечує високий рівень безпеки для користувачів. Крім того, вона надає готові рішення для відновлення пароля та зміни електронної адреси користувача.

FastAPI-Users дозволяє легко і швидко інтегрувати систему користувачів у вебзастосунок на базі FastAPI, забезпечуючи високий рівень безпеки та зручності для розробників.

### 1.3.3 Vue.js

Vue.js - це прогресивний JavaScript-фреймворк для побудови користувацьких інтерфейсів. Цей фреймворк відомий своєю простотою використання, легкістю вивчення та високою продуктивністю. Vue.js дозволяє створювати складні вебзастосунки швидко та ефективно.

Однією з ключових особливостей Vue.js є його реактивність. Фреймворк автоматично відстежує зміни даних та оновлює відповідний DOM, що робить розробку вебзастосунків більш простою та продуктивною. Vue.js також має модульну структуру, що дозволяє легко організувати та управляти кодом великих проектів.

Цей фреймворк має широкий набір інструментів та бібліотек для створення різноманітних компонентів та додатків. Він підтримує використання шаблонів, компонентів та директив для побудови інтерфейсу вебзастосунків. Vue.js також інтегрується з різними інструментами та бібліотеками, такими як Vuex для керування станом додатку та Vue Router для реалізації маршрутизації.

Завдяки своїм простоті та потужності, Vue.js став популярним вибором для розробників вебзастосунків у всьому світі.

### 1.3.4 PostgreSQL

PostgreSQL - це потужна та надійна об'єктно-реляційна система управління базами даних (СКБД), яка використовується для зберігання та управління даними в різних вебзастосунках. Вона відома своєю високою продуктивністю, розширюваністю та розширеними можливостями.

PostgreSQL підтримує багато різних типів даних, включаючи числа, рядки, дати, географічні дані, JSON та багато іншого. Вона також має вбудовану підтримку для транзакцій, що забезпечує цілісність даних та консистентність операцій.

Ця база даних має розширені можливості управління даними, включаючи індексацію, оптимізацію запитів та забезпечення високої доступності даних. Вона також підтримує реплікацію, що дозволяє створювати дублікати баз даних для забезпечення надійності та швидкості доступу до даних.

PostgreSQL є відкритим джерелом і має активну спільноту розробників, яка постійно вдосконалює та розширює його функціональні можливості. Ця база даних є популярним вибором для вебзастосунків, які потребують надійного та швидкого зберігання даних.

### 1.3.5 GitHub

GitHub — це онлайн-платформа розробки програмного забезпечення, яка використовується для зберігання, відстеження та співпраці над програмними проектами. Це дозволяє розробникам завантажувати власні файли коду та співпрацювати з іншими розробниками над проектами з відкритим кодом.

GitHub також служить сайтом соціальної мережі, на якому розробники можуть відкрито спілкуватися, співпрацювати та презентувати свою роботу. GitHub дозволяє розробникам програмного забезпечення та інженерам безкоштовно створювати віддалені загальнодоступні сховища в хмарі. Після того, як ви налаштували репозиторій на GitHub, ви можете скопіювати його на свій пристрій, локально додавати та змінювати файли, а потім «відправляти» свої зміни назад у сховище, де ваші зміни відображаються для всіх.

### 1.3.6 Axios

Axios — це популярна JavaScript-бібліотека для здійснення HTTP-запитів, яка широко використовується на фронтенді. У цьому проєкті Axios використовується у зв’язці з Vue.js для взаємодії з REST API, реалізованим на FastAPI. Вона забезпечує зручний інтерфейс для надсилання GET, POST, PUT, DELETE запитів та обробки відповідей сервера. Axios підтримує асинхронні запити, перехоплення запитів і відповідей, обробку помилок, що робить її ефективним інструментом для побудови клієнт-серверної архітектури.

### 1.3.7 SQLAlchemy

SQLAlchemy — це популярна бібліотека ORM (Object-Relational Mapping) для Python, яка використовується в бекенд-частині застосунку для роботи з базою даних PostgreSQL. Вона дозволяє розробникам працювати з базою даних, використовуючи об’єктно-орієнтований підхід, що підвищує читабельність коду та знижує ймовірність помилок при виконанні SQL-запитів. SQLAlchemy підтримує як декларативний стиль моделювання даних, так і роботу з сирими SQL-запитами, надаючи гнучкість у реалізації логіки взаємодії з базою.

### 1.3.8 Pydantic

Pydantic — це бібліотека для Python, що використовується для валідації та серіалізації даних. Вона є важливою складовою FastAPI, оскільки саме з її допомогою описуються схеми вхідних та вихідних даних у вигляді Python-класів. Pydantic автоматично перевіряє відповідність даних заданим типам і повертає зрозумілі повідомлення про помилки. Це значно полегшує налагодження API та гарантує, що дані, отримані від клієнта, відповідають очікуваному формату.

### 1.3.9 Шарова архітектура

У проєкті реалізовано шарову (багаторівневу) архітектуру, яка забезпечує чіткий поділ відповідальності між різними компонентами системи. Такий підхід сприяє масштабованості, повторному використанню коду та полегшенню тестування. Презентаційний шар, реалізований за допомогою Vue.js, відповідає за взаємодію з користувачем, відображення даних та ініціювання дій, зокрема запитів до API через бібліотеку Axios. Шар бізнес-логіки реалізовано на основі FastAPI — він включає обробку запитів, автентифікацію, авторизацію, а також реалізацію основної логіки обробки даних. Шар доступу до даних побудований з використанням SQLAlchemy та відповідає за взаємодію з базою даних PostgreSQL, зокрема за зберігання, вибірку й оновлення інформації. Окрему роль відіграють моделі даних, представлені у вигляді Pydantic-схем для валідації та взаємодії з API, а також ORM-моделей, призначених для відображення структури бази даних. Така архітектура дозволяє гнучко керувати залежностями, оновлювати окремі компоненти без впливу на інші та забезпечує ефективну підтримку проєкту в довгостроковій перспективі.

# РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ

# ДОДАТКУ ВІДПОВІДНО ДО ВИМОГ

Для зручного написання коду та управління проектом використовувалися інтегровані середовища розробки PyCharm та WebStorm вебсервіс контролю версій GitHub. Як базу даних для аплікації було обрано PostgreSQL. Усі вони були описані у попередніх розділах.

## 2.1 Загальний опис та характеристика продукту

Дана вебаплікація була створена для вирішення наступного ряду вимог:

а) створення профілю користувача

б) можливість переглядати стартапи

в) створення/редагування/видалення та виконання завдань у рамках стартапів

г) обмін пропозиціями для виконання завдань у персональному чаті

ґ) створення/редагування/видалення стартапів

д) пошук стартапів за назвою

е) оцінювання роботи користувачів

є) можливість коментувати стартапи

ж) блокування певних функцій для неавторизованого користувача

Платформа передбачає наявність одного типу користувача — звичайного користувача. Користувач може переглядати список стартапів, отримувати додаткову інформацію про кожен з них, а також здійснювати пошук конкретних стартапів. Крім того, він має доступ до свого особистого профілю, де може побачити свої стартапи та активні завдання, які він взяв на виконання. Користувач також має можливість додавати, редагувати або видаляти стартапи, а також додавати, редагувати або видаляти завдання для цих стартапів. Коли користувач бере завдання, він може створити персональний чат із замовником, тобто власником завдання, для ефективної комунікації та обговорення деталей.

## 2.2 Діаграми проекту

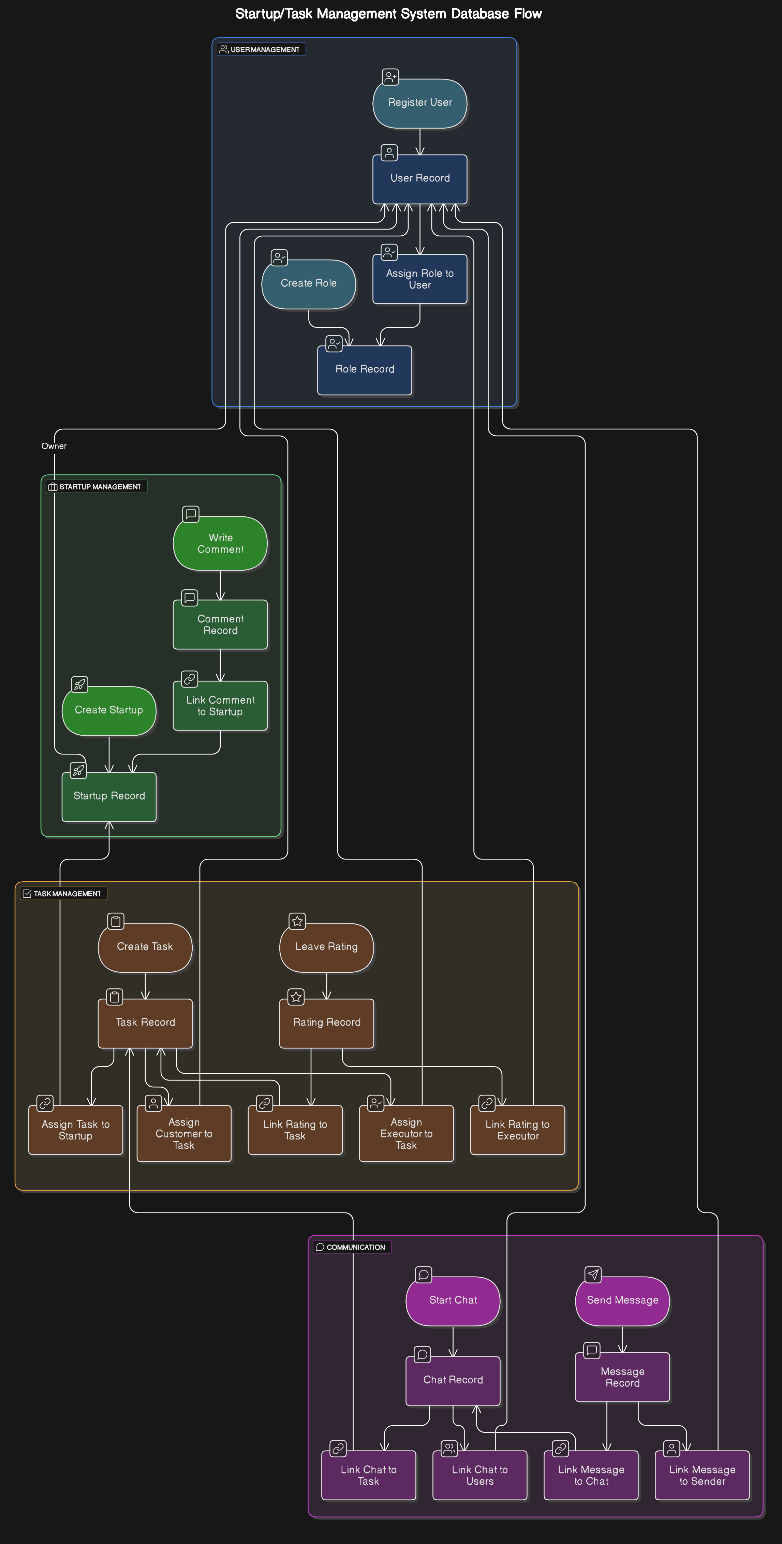


Рисунок 1 – Use-case діаграма

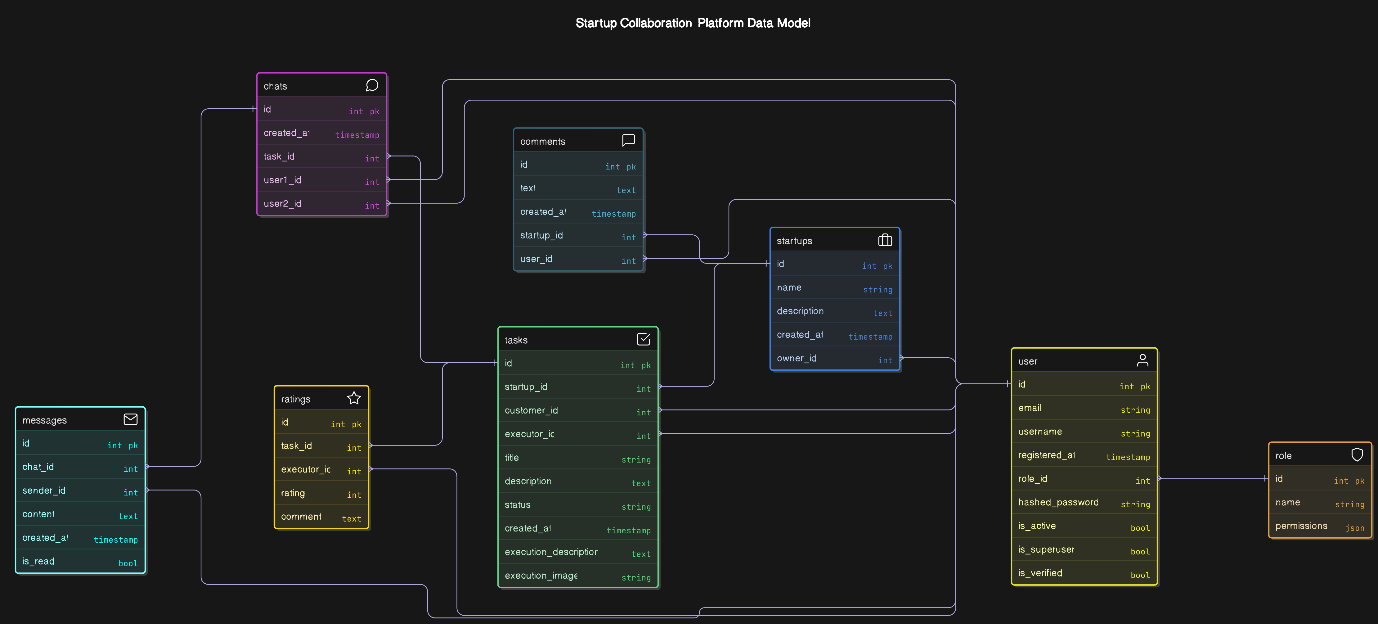


Рисунок 2 - ER-діаграма

## 2.3 База даних проекту

Під час запуску програми, система автоматично створює базу даних на локальному пристрої, і тут же встановлює зв'язок між цією базою даних і самою програмою. Таким чином, користувачеві не потрібно вручну налаштовувати або підключати базу даних до програми - це відбувається автоматично під час запуску програми.

Схема бази даних:

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| Id | Integer | User's id |
| Email | String | User’s email |
| Username | String | User’s username |
| Registered\_at | Timestamp | Automatically created timestamp |
| Role\_id | Integer | User's role id |
| Hashed\_password | String | secured user's password |
| Is\_active | Boolean | User’s activity |
| Is\_superuser | Boolean | User’s opportunities |
| Is\_vetified | Boolean | User’s verification |

Таблиця 1 - Users table

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| Id | Integer | Startup's id |
| Name | String | Startup’s name |
| Description | Text | Startup's description |
| Created\_at | Timestamp | Automatically created timestamp |
| Owner\_id | Integer | User's id (startup owner) |

Таблиця 2 - Startups table

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| Id | Integer | Task's id |
| Title | String | Task's title |
| Description | Text | Task's description |
| Status | String | Task's current status |
| Created\_at | Timestamp | Automatically created timestamp |
| Startup\_id | Integer | |  | | --- | |  |  |  | | --- | | Startup’s id | |
| Customer\_id | Integer | User’s id (task’s customer) |
| Executor\_id | Integer | User’s id (task’s executor) |
| Execution\_description | Text | Task execution description |
| Execution\_image | String | Task execution image (optional) |

Таблиця 3 - Tasks table

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| Id | Integer | Chat's id |
| Task\_id | Integer | Task’s id |
| User1\_id | Integer | User’s id (customer) |
| User2\_id | Integer | User’s id (executor) |
| Created\_at | Timestamp | Automatically created timestamp |

Таблиця 4 – Chats table

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| Id | Integer | Message's id |
| Chat\_id | Integer | Chat's id |
| Sender\_id | Integer | User’s id (message sender) |
| Content | Text | Message content |
| Created\_at | Timestamp | Automatically created timestamp |
| Is\_read | Boolean | Whether the message has been read |

Таблиця 5 - Messages table

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| Id | Integer | Rating's id |
| Rating | Integer | Rating value (1-5) |
| Comment | Text | Comment on the task |
| Task\_id | Integer | Task’s id |
| Executor\_id | Integer | User’s id (task’s executor) |

Таблиця 6 - Ratings table

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| Id | Integer | Comment's id |
| Text | Text | Text of the comment |
| Created\_at | Timestamp | Automatically created timestamp |
| User\_id | Integer | User’s id (comment author) |
| Startup\_id | Integer | Startup’s id |

Таблиця 7 – Comments table

# РОЗДІЛ 3. СЦЕНАРІЙ РОБОТИ ВЕБЗАСТОСУНКУ

Увійшовши в програму, користувача зустріне форма для входу в персональний аккаунт, якщо такого немає, то є можливсіть створити його при кліку на кнопку “Зареєструватися”

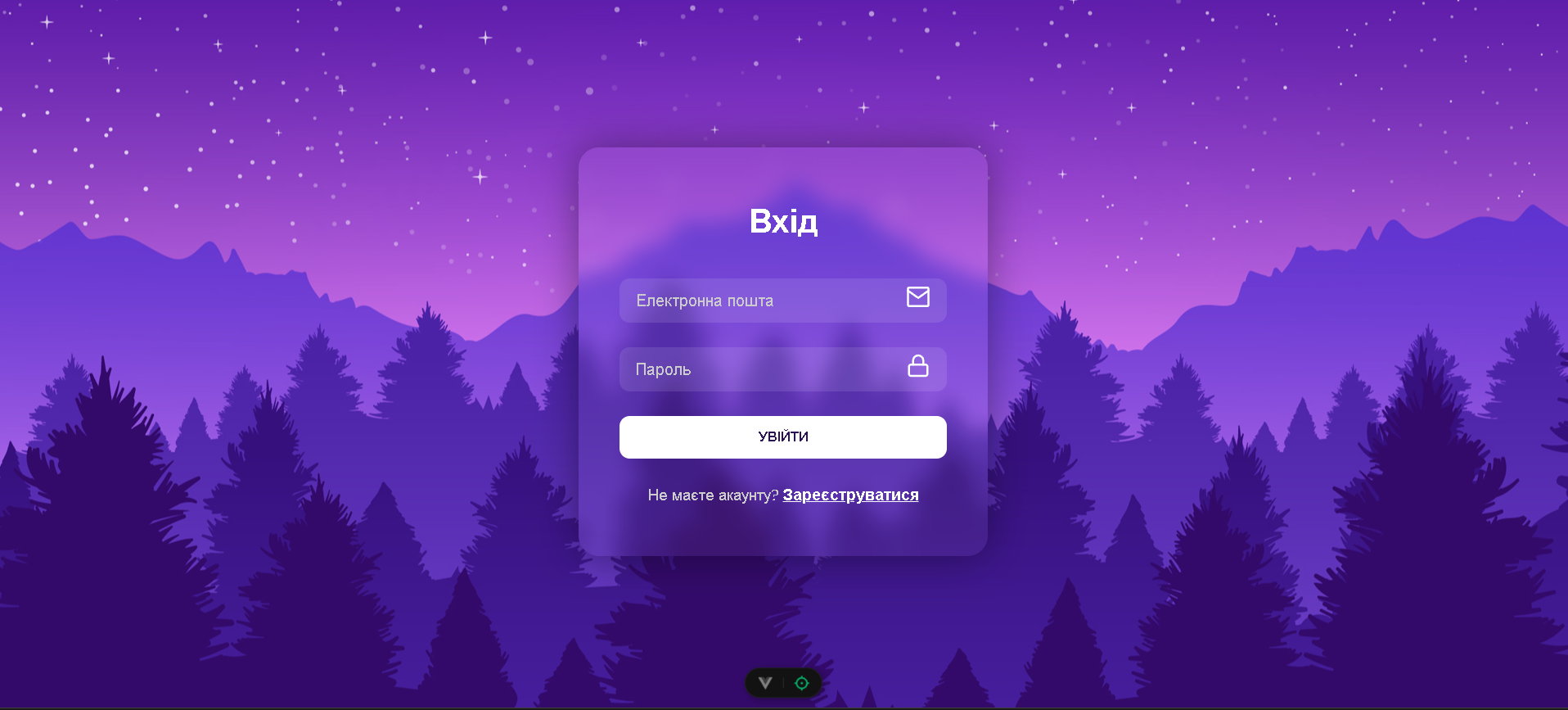


Рисунок 3.1 – Сторінка входу

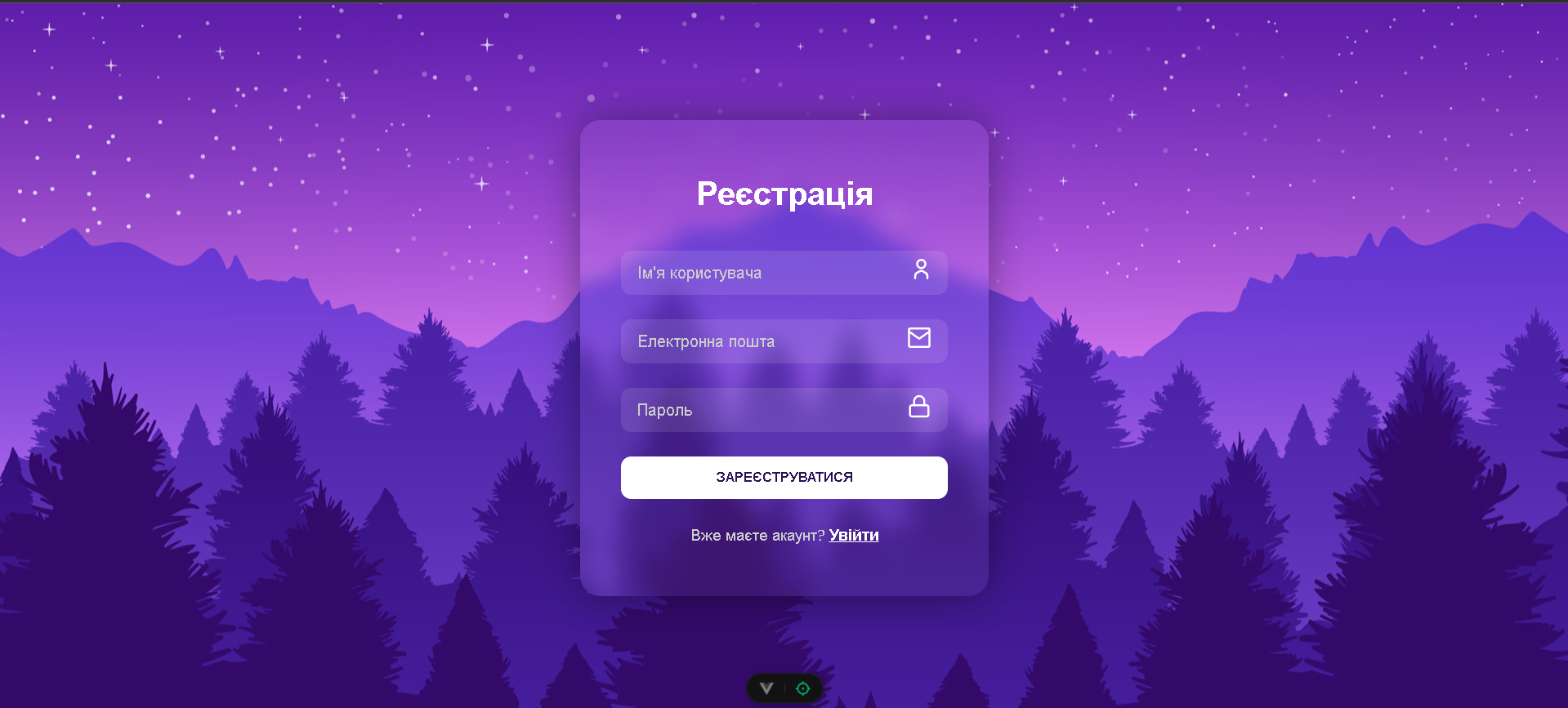


Рисунок 3.2 – Сторінка реєстрації

Після успішної реєстрації, користувача перенаправляє на сторінку входу, після успішного входу він потрапляє на головну сторінку.

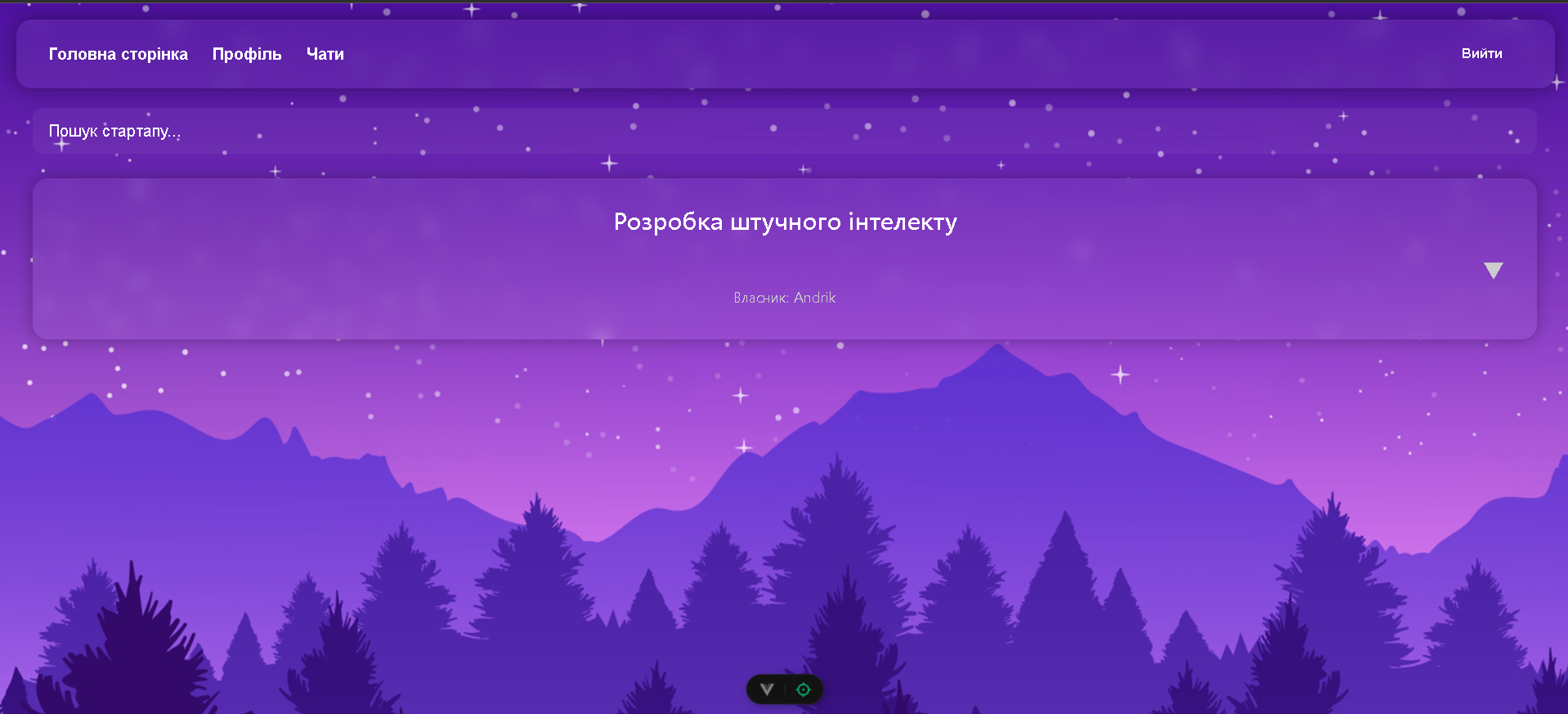


Рисунок 3.3 – Головна сторінка

На головній сторінці відображаються стартапи які зараз є в активному розвитку, також є можливість шукати стартапи за назвою, розгортати для детальнішої інформації.

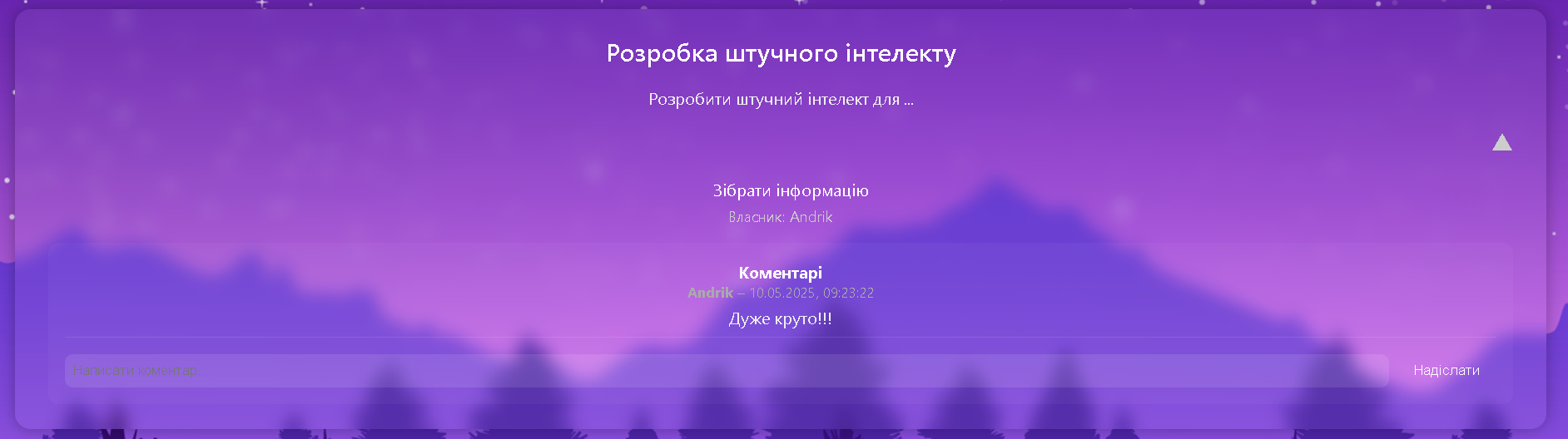


Рисунок 3.4 – Розгорнутий вигляд стартапу

В розгорнутому вигляді ми можемо почитати опис стартапу, побачити перелік завдань, список коментарів та залишити свій власний коментар. При кліку на завдання, користувача перенаправляє на сторінку з інформацією про завдання.

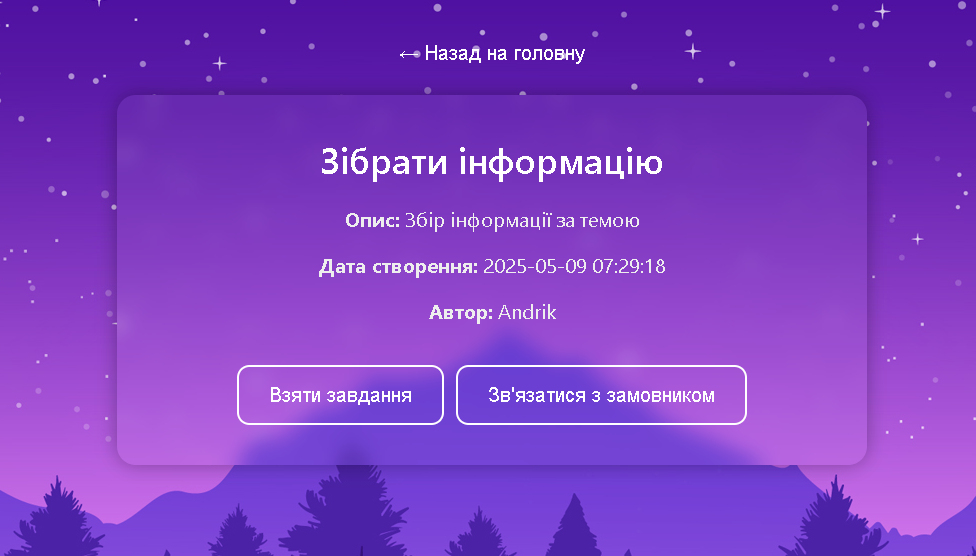


Рисунок 3.5 – Сторінка інформації про завдання

На цій сторінці користувач може детально ознайомитися з завданням, взяти його на виконання та зв’язатися з замовником, при натисненні на цю кнопку створиться персональний чат між користувачем та власником.

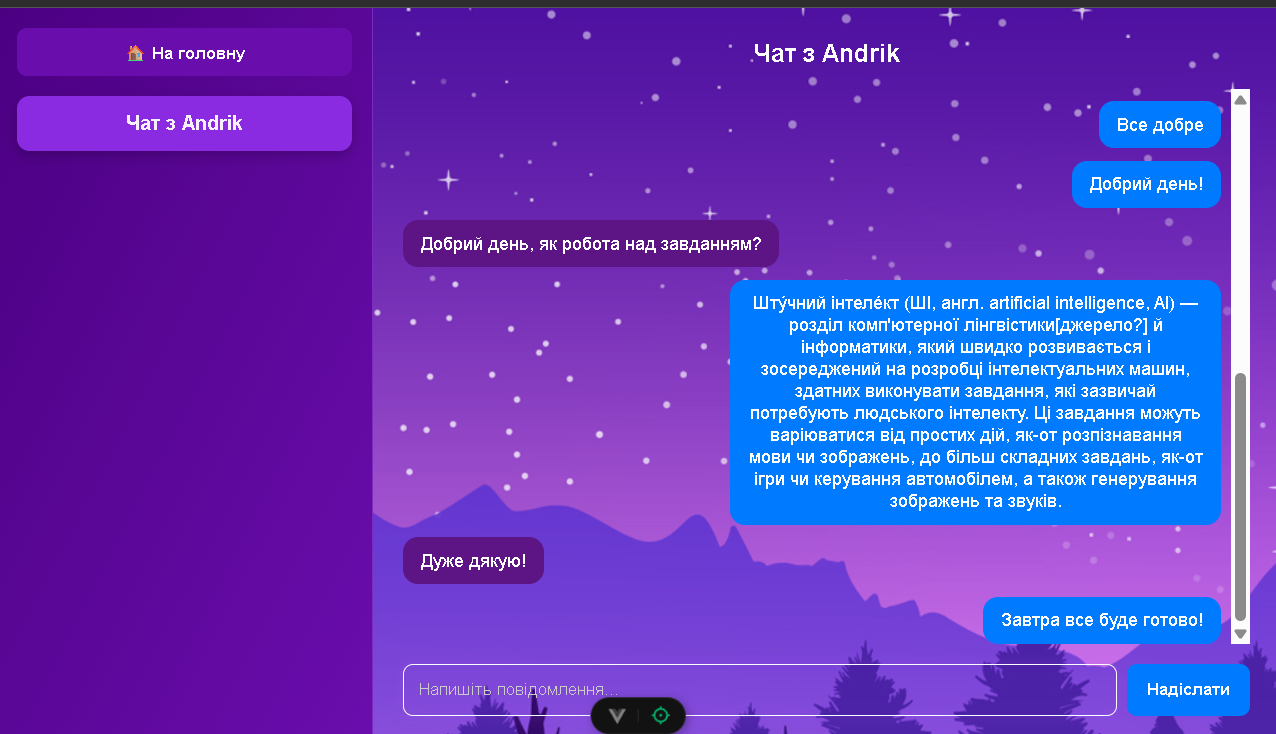


Рисунок 3.6 – Сторінка персонального чату

Також на цій сторінці є можливість повернутися на головну сторінку. В навігації застосунку є пункт профіль, при натиснені на нього, ми потрапимо на сторінку профілю.

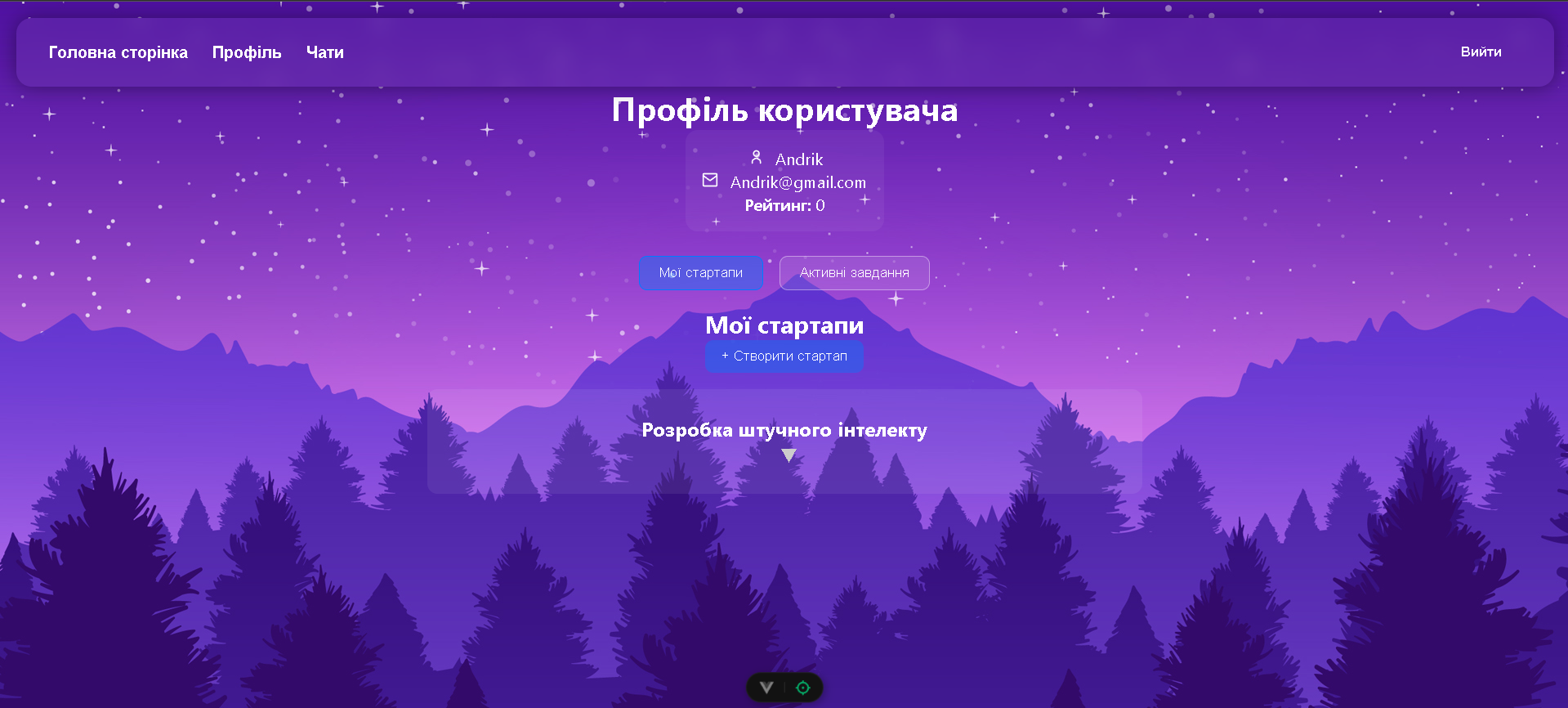


Рисунок 3.7 – Сторінка профілю

На ній ми можемо побачити особисту інформацію, рейтинг тобто середню оцінку по виконаних завданнях. Також тут є 2 вкладки “Мої стартапи“ та “Активні завдання“. Як можна побачити на рисунку, є можливість додати новий стартап.

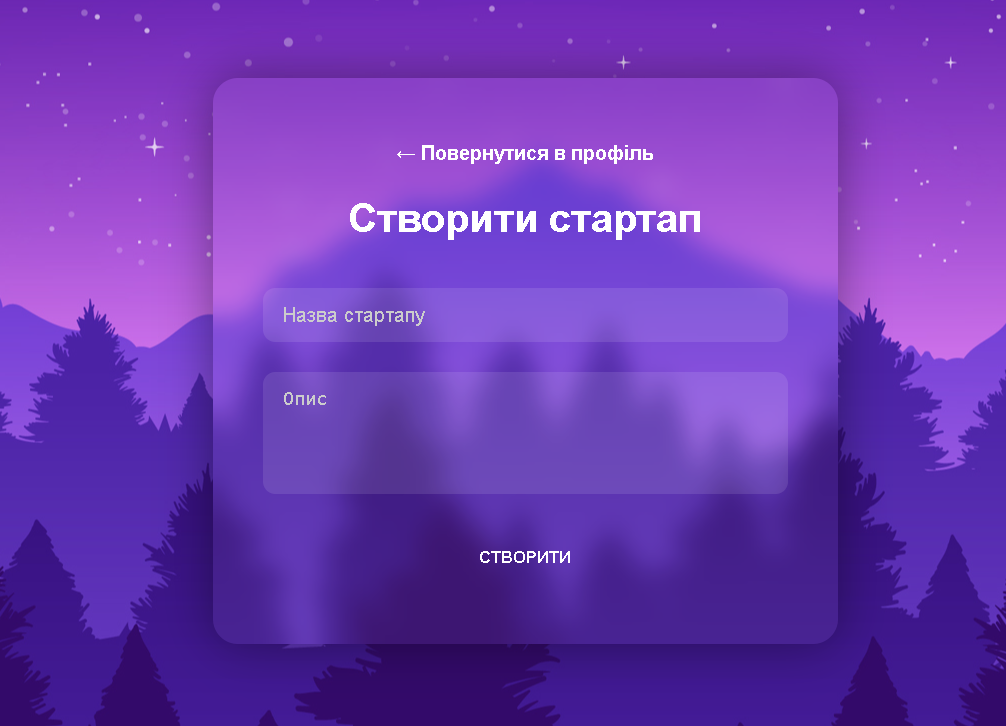


Рисунок 3.8 – Сторінка створення стартапу

Також у нас відображаються вже наявні стартапи, які в розгортці мають наступний вигляд

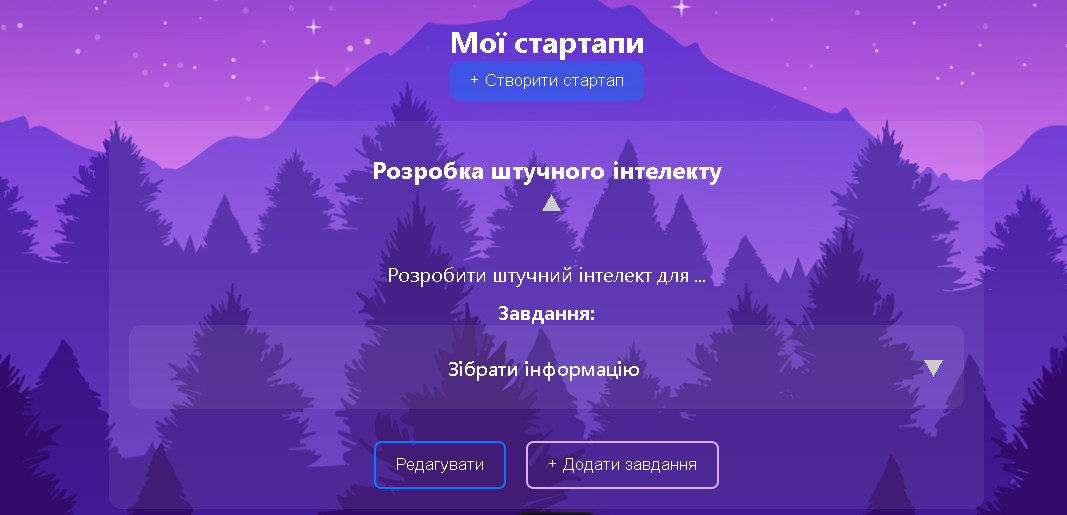


Рисунок 3.9 – Розгорнутий вигляд стартапу

Як ми можемо побачити в нас тут є можливість редагувати стартап, нам автоматично підтягується вся інформація про нього в поля для вводу і виглядає це все наступним чином

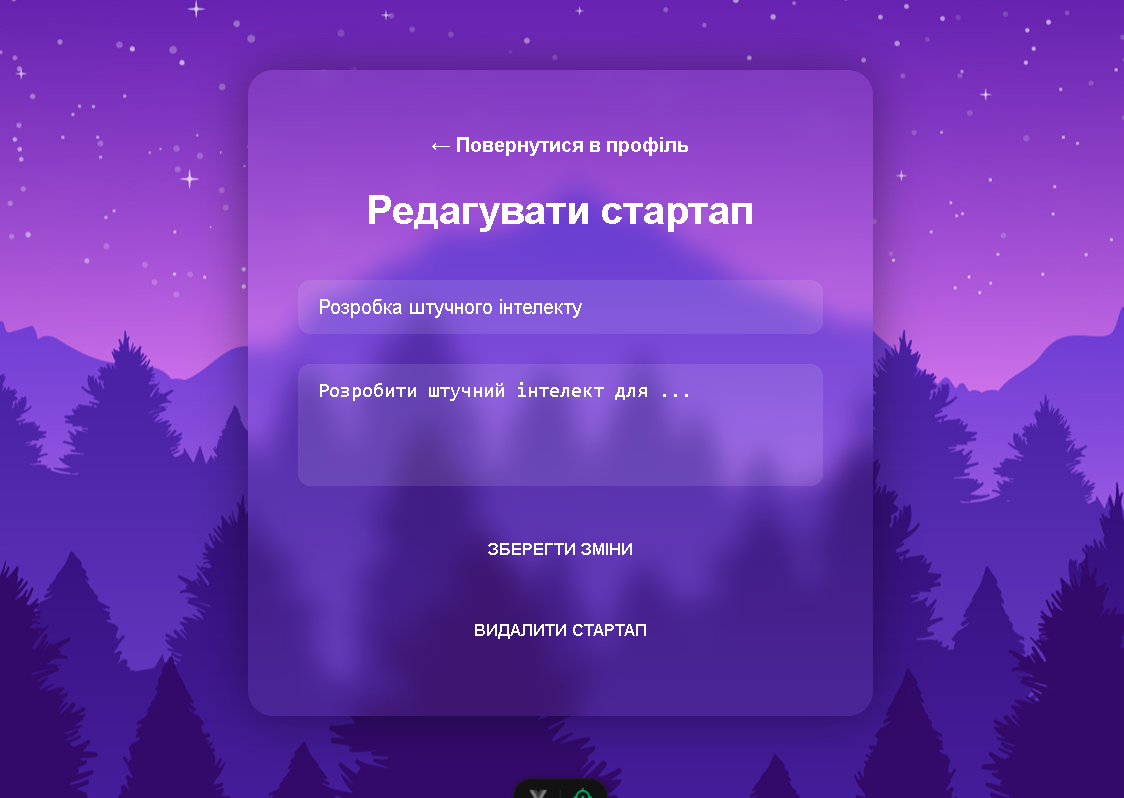


Рисунок 3.10 – Сторінка редагування стартапу

Як і в інших випадках ми маємо кнопку повернення в профіль, бачимо наші заповнені поля, які можемо змінювати і потім зберегти зміни, також є можливість видалення стартапу.  
 Також в розгортці стартапу можна було побачити кнопку “+ Додати завдання”, вона додає завдання до нашого стартапу і сторінка з додаванням завдання виглядає наступним чином

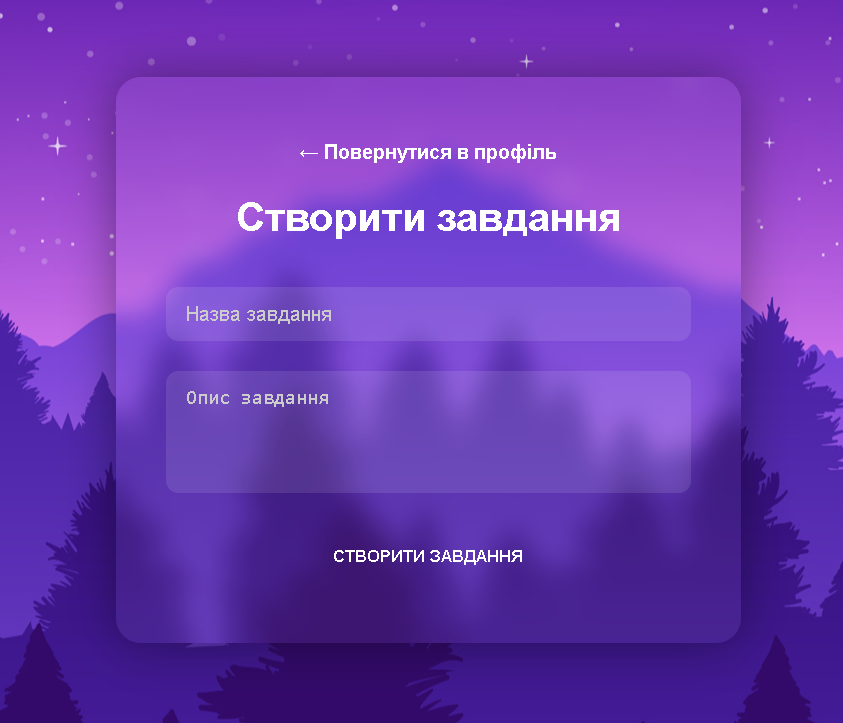


Рисунок 3.11 – Сторінка створення завдання

На цій сторінці ми можемо надати всі потрібні дані та натиснути кнопку створити завдання, після чого воно буде додано до нашого стартапу.

Завдання може перебувати в різних статусах, поки його ще ніхто не взяв, то є можливість редагування нижче приклад, як виглядають завдання в різних статусах:

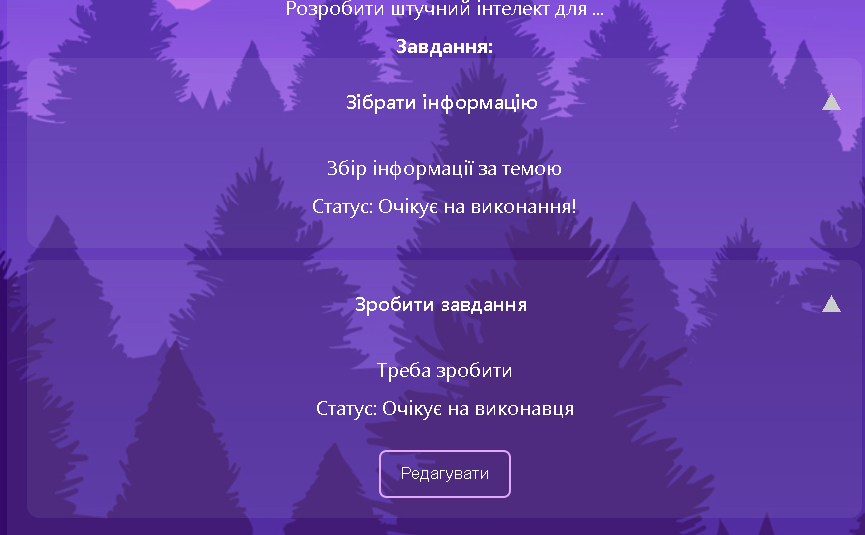


Рисунок 3.12 – Завдання в різних статусах

При натиску на кнопку редагування для завдання, нас переносить на сторінку редагування завдань, де знову таки вся наявна інформація підтягується в відповідні поля і є можливість зберегти зміни чи видалити завдання.

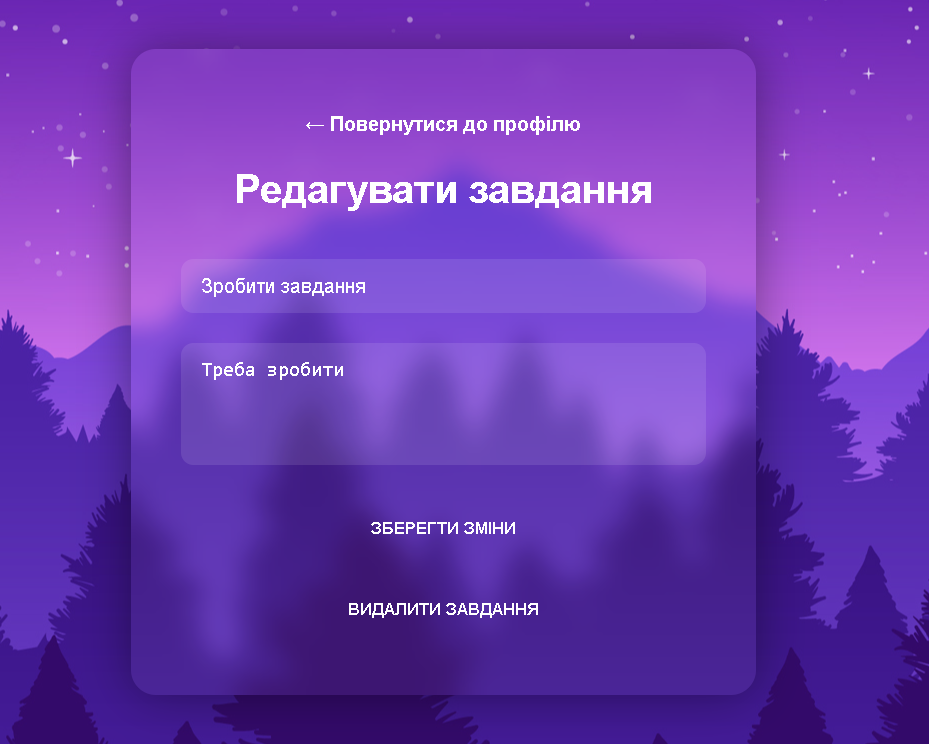


Рисунок 3.12 – Сторінка редагування завдання

Також є вкладка з активними завданнями, там відображаються завдання, які ми взяли на виконання, виглядає наступним чином:

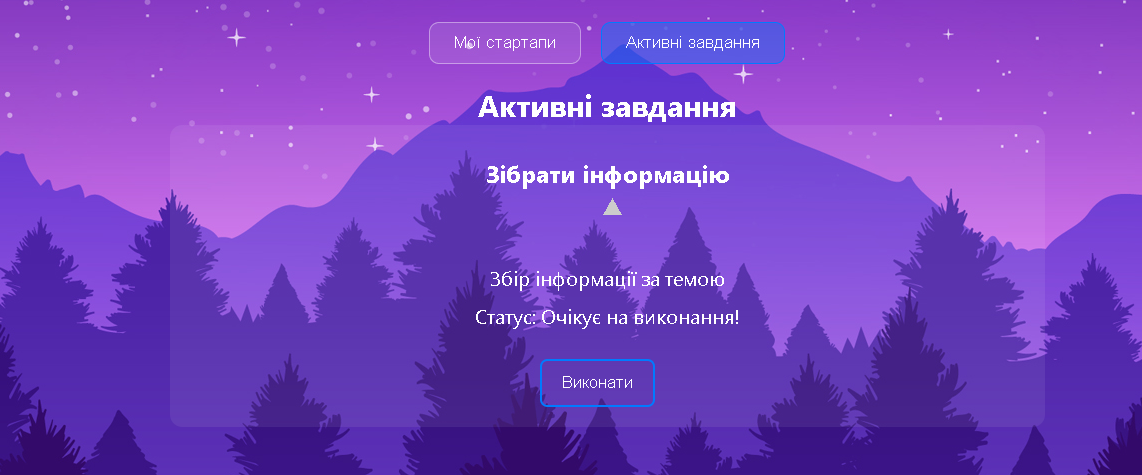


Рисунок 3.13 – Вкладка активних завдань

Користувач тут може виконати завдання, яке він взяв, при натискані на кнопку виконати його перекине на наступну сторінку

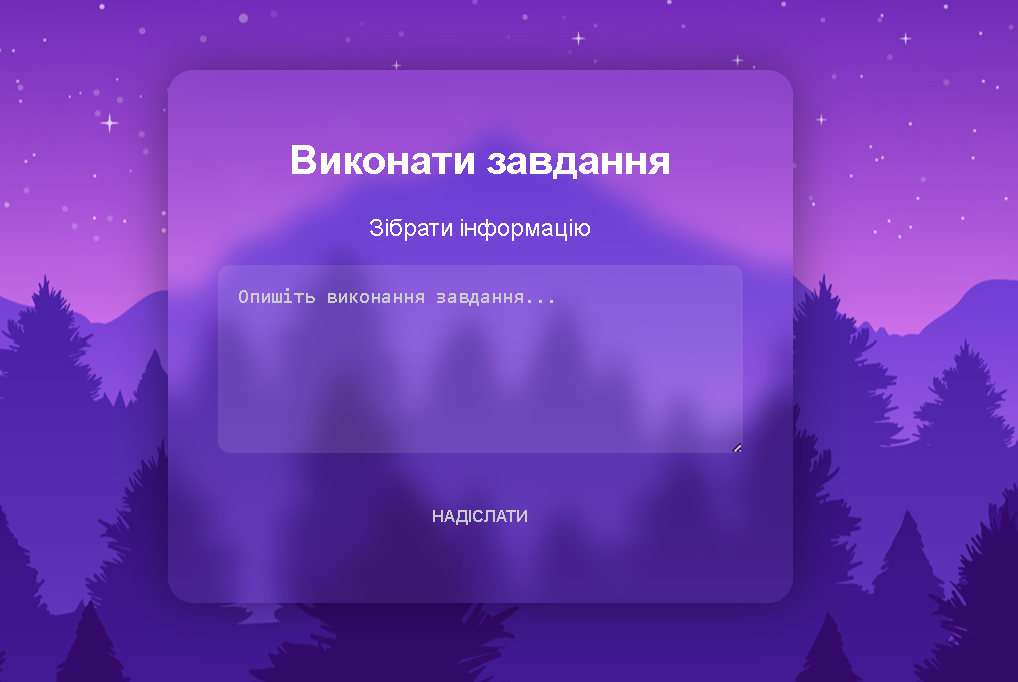


Рисунок 3.14 – Сторінка для виконання завдання

Тут користувач може вказати потрібну інформацію, посилання на якийсь ресурс, все що потрібно для виконання завдання, та натиснути кнопку надіслати. Після чого користувач буде мати наступне відображення в завданнях

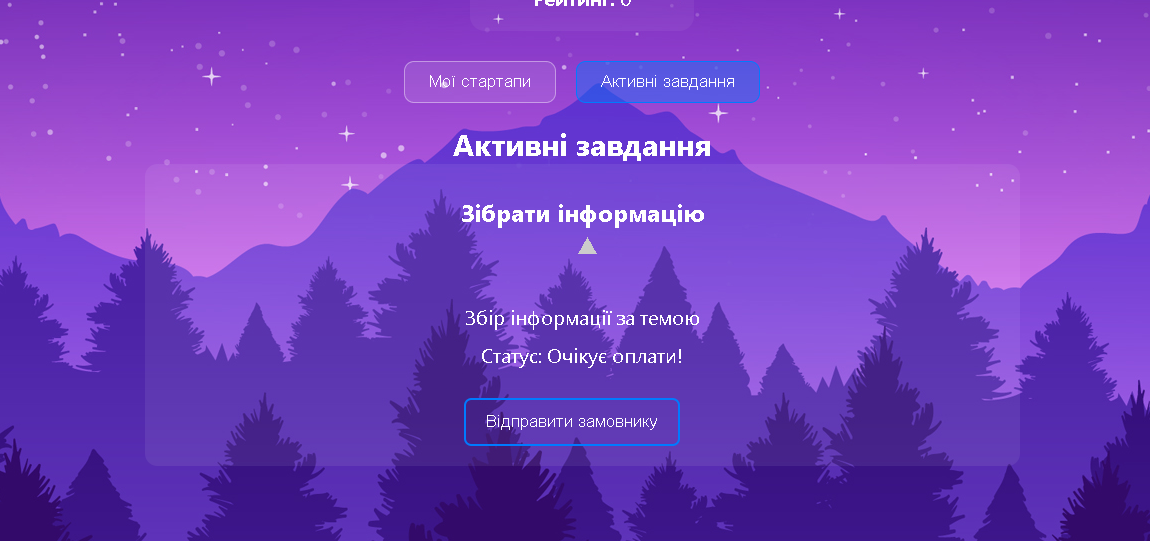


Рисунок 3.15 – Вкладка активних завдань при виконаному завданні

Кнопка відправити замовнику дасть доступ до виконаного завдання власнику цього завдання, тобто платформа дає можливість домовитися в чаті про оплату, після чого користувач виконує завдання відправляє його на платформу і при отриманні коштів дає доступ замовнику.

Отримане завдання виглядає наступним чином

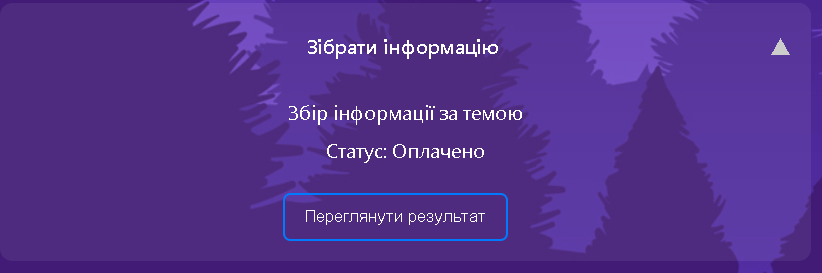


Рисунок 3.16 – Отримане власником завдання

При натиснені кнопки переглянути результат, нас перенесе на сторінку з виконаним завданням, та можливістю оціти його за шкалою 1-5

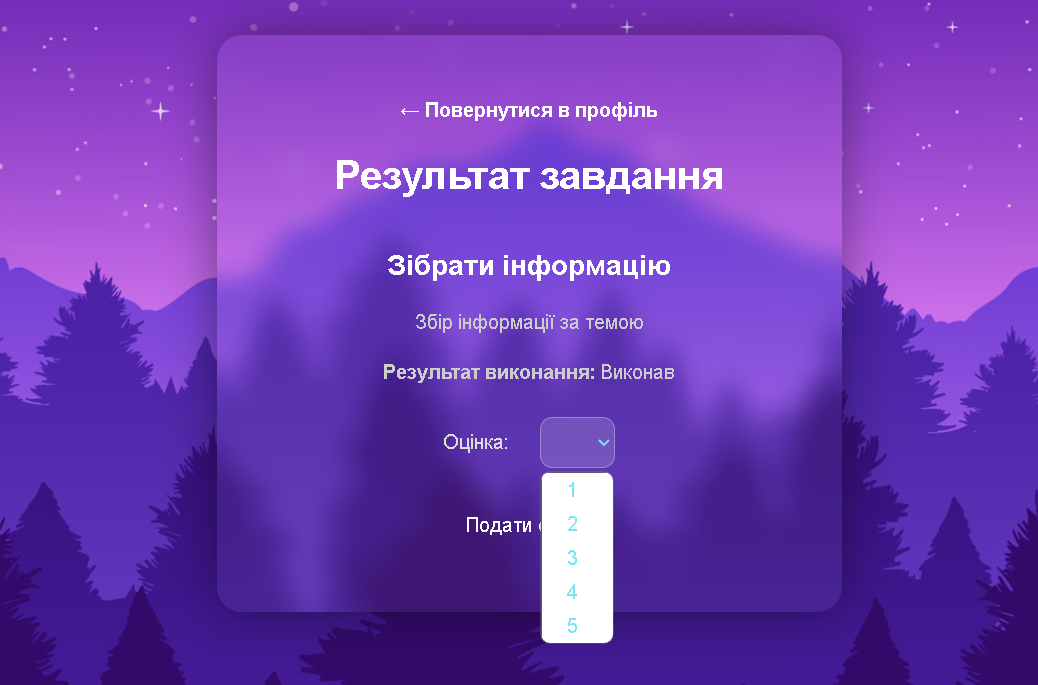


Рисунок 3.17 – Сторінка виконаного завдання

# ВИСНОВКИ

У цій дипломній роботі було розроблено вебплатформу для краудсорсингу ідей та завдань у стартапах із використанням мови програмування Python і фреймворку FastAPI. Основна мета полягала у створенні зручного середовища, де користувачі можуть створювати стартапи, додавати до них завдання, залишати коментарі-пропозиції, брати завдання у виконання та взаємодіяти через систему чатів.

У процесі реалізації було спроєктовано та створено моделі для користувачів, стартапів, завдань, оцінювання, коментарів і чатів. Було використано реляційну базу даних для зберігання інформації та реалізовано API-інтерфейси для зручної взаємодії з фронтендом. Особливу увагу приділено організації логіки доступу, системі ролей та безпеці облікових записів користувачів.

Для розробки застосунку були використані сучасні технології: Python, FastAPI, SQLAlchemy, PostgreSQL. Вихідний код проєкту розміщено на GitHub і доступний за посиланням: https://github.com/c1own-depressed/Dyploma

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. FastAPI Cookbook авторства Пола Гловацького (Paul Głowacz): Ця книга містить практичні приклади та поради щодо розробки вебдодатків з використанням FastAPI. Видавництво: Packt Publishing, Рік: 2020.
2. Vue.js in Action авторства Еріка Раховіца (Erik Hanchett): Ця книга детально описує використання фреймворку Vue.js для розробки вебдодатків. Видавництво: Manning Publications, Рік: 2018. PostgreSQL: Up and Running авторства Регіни О’Бейрн (Regina Obe) та Лео Хсю (Leo S. Hsu): Ця книга пропонує огляд та практичні приклади використання
3. PostgreSQL для розробки та управління базами даних. Видавництво:

O'Reilly Media, Рік: 2017.

1. Python for Data Analysis авторства Веса МакКінні (Wes McKinney): Ця книга допоможе вам зрозуміти основи роботи з даними в Python, що може бути корисним при розробці вебзастосунків з використанням FastAPI та PostgreSQL. Видавництво: O'Reilly Media, Рік: 2017.
2. FastAPI Web Development Cookbook авторства Річарда Лоузі (Richard Lowe): Ця книга містить практичні рецепти та приклади для розробки вебзастосунків з використанням FastAPI. Вона надає читачам інсайти щодо різних аспектів розробки, включаючи взаємодію з базами даних, автентифікацію та авторизацію, тестування та багато іншого.

Видавництво: Packt Publishing, Рік: 2020.

1. Learning PostgreSQL 12: A beginner's guide to building high-performance

PostgreSQL database solutions авторства Абхішек Роя (Ahsanul Bari, Andrey Volkov, Jayadevan Maymala): Ця книга зосереджена на PostgreSQL 12 і допоможе вам зрозуміти основні концепції та методи роботи з цією базою даних. Видавництво: Packt Publishing, Рік: 2019.

1. Vue.js: Up and Running авторства Чарльза Джука (Charles David Crawford): Ця книга пропонує практичний підхід до вивчення Vue.js і надає ряд корисних порад та прикладів. Видавництво: O'Reilly Media, Рік: 2018.