

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ  
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## **ПОЯСНЮВАЛЬНА ЗАПИСКА**

до кваліфікаційної роботи освітнього ступеня «бакалавр»  
за спеціальністю 121 «Інженерія програмного забезпечення»  
(освітня програма «Інженерія програмного забезпечення»)

на тему:

**«Платформа для ведення персоналізованих списків  
перегляду та обговорення аніме»**

Виконав студент групи ІПЗ-21-3  
ЛОЄВСЬКИЙ Ярослав Дмитрович

Керівник роботи:  
ЧИЖМОТРА Олексій Володимирович

Рецензент:  
ЛОБАНЧИКОВА Надія Миколаївна

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ  
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ЗАТВЕРДЖУЮ

Зав. кафедри інженерії  
програмного забезпечення  
Тетяна ВАКАЛЮК  
«14» лютого 2025 р.

**ЗАВДАННЯ**

**на кваліфікаційну роботу**

Здобувач вищої освіти: **ЛОЄВСЬКИЙ Ярослав Дмитрович**

Керівник роботи: **ЧИЖМОТРЯ Олексій Володимирович**

Тема роботи: «Платформа для ведення персоналізованих списків перегляду та обговорення аніме»,

затверджена Наказом закладу вищої освіти від «14» лютого 2025 р., №61/с

Об'єктом дослідження є процеси ведення персоналізованих списків перегляду та обговорення аніме.

Предметом дослідження є застосування сучасних технологій розробки вебзастосунків для створення платформи, що забезпечує персоналізоване ведення списків перегляду аніме та можливості для його обговорення.

Консультанти з бакалаврської кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

| Розділ | Консультант   | Завдання видав | Завдання прийняв |
|--------|---------------|----------------|------------------|
| 1      | Чижмотря О.В. | 14.02.2025     | 14.02.2025       |
| 2      | Чижмотря О.В. | 14.02.2025     | 14.02.2025       |
| 3      | Чижмотря О.В. | 14.02.2025     | 14.02.2025       |

## РЕФЕРАТ

Кваліфікаційна робота освітнього ступеня «бакалавр» складається з веборієнтованої платформи для ведення персоналізованих списків перегляду та обговорення аніме та пояснювальної записки. Пояснювальна записка до кваліфікаційної роботи містить 71 сторінку, 41 ілюстрацію, 20 таблиць, 35 використаних джерел та 1 додаток.

Загальний обсяг – 86 сторінок.

Метою роботи є розробка платформи, яка забезпечить користувачам можливість створювати персоналізовані списки перегляду аніме, ділитися враженнями та брати участь в обговореннях, використовуючи сучасні веб-технології для досягнення зручності, ефективності й високої продуктивності.

У роботі поставлені основні завдання на розробку платформи. Це аналіз існуючих рішень для аніме, проєктування архітектури, реалізація веб-платформи, тестування та оцінка її ефективності. Проведено порівняльний аналіз аналогів, визначено їхні переваги та недоліки, а також окреслено конкурентні переваги системи. Розробка веб-платформи відбувалася із застосуванням сучасних веб-технологій, зокрема HTML, CSS, JavaScript, Node.js та бази даних MongoDB.

Описано основні сценарії використання, структуру бази даних, діаграми взаємодії та ключові алгоритми модулів, що ґрунтуються на методах моделювання та прототипування. Для перевірки працездатності проведено тестування, а оцінка ефективності здійснена шляхом збору відгуків користувачів.

**КЛЮЧОВІ СЛОВА:** АНІМЕ, ПЛАТФОРМА, СПИСКИ ПЕРЕГЛЯДУ, ОБГОВОРЕННЯ, ВЕБСАЙТ, ПЕРСОНАЛІЗАЦІЯ, КОРИСТУВАЧІ, РОЗРОБКА.

|              |      |                   |        |      |                                                                                                       |  |  |                                            |      |         |    |
|--------------|------|-------------------|--------|------|-------------------------------------------------------------------------------------------------------|--|--|--------------------------------------------|------|---------|----|
|              |      |                   |        |      | ІПЗ.КР.Б-121-25-ПЗ                                                                                    |  |  |                                            |      |         |    |
|              |      |                   |        |      |                                                                                                       |  |  |                                            |      |         |    |
| Змн.         | Арк. | № докум.          | Підпис | Дата |                                                                                                       |  |  |                                            |      |         |    |
| Розроб.      |      | Лосевський Я.Д.   |        |      | Платформа для ведення персоналізованих списків перегляду та обговорення аніме<br>Пояснювальна записка |  |  | Літ.                                       | Арк. | Аркушів |    |
| Керівник     |      | Чижмотря О.В.     |        |      |                                                                                                       |  |  |                                            |      | 3       | 86 |
|              |      |                   |        |      |                                                                                                       |  |  | Житомирська політехніка,<br>група ІПЗ-21-3 |      |         |    |
| Рецензент    |      | Лобанчикова М. Н. |        |      |                                                                                                       |  |  |                                            |      |         |    |
| Зав. кафедри |      | Вакалюк Т.А.      |        |      |                                                                                                       |  |  |                                            |      |         |    |

## ABSTRACT

The bachelor's qualification thesis consists of a web-oriented platform for maintaining personalized anime watchlists and discussions and an explanatory note. The explanatory note to the qualification thesis contains 71 pages, 41 illustrations, and 20 tables, 35 sources used and 1 appendix.

The total volume is 86 pages.

The aim of the thesis is to develop a platform that will enable users to create personalized anime watchlists, share impressions, and participate in discussions, using modern web technologies to achieve convenience, efficiency, and high performance.

The main objectives of the thesis are to design and develop the platform, conduct a comparative analysis of similar solutions, and determine the competitive advantages of the system. The main tasks for platform development include: analyzing existing anime solutions, designing the architecture, implementing the web platform, testing, and evaluating its effectiveness. The development of the web platform was carried out using modern web technologies, including HTML, CSS, JavaScript, Node.js, and MongoDB database.

The main usage scenarios, database structure, interaction diagrams, and key algorithms of the modules based on modeling and prototyping methods are described. Performance testing was conducted, and the effectiveness evaluation was carried out by collecting user feedback.

**KEYWORDS:** ANIME, PLATFORM, WATCHLISTS, DISCUSSION, WEBSITE, PERSONALIZATION, USERS, DEVELOPMENT.

|      |      |                 |        |      |                                            |      |
|------|------|-----------------|--------|------|--------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка.25.121.21.000 - ПЗ | Арк. |
|      |      | Чижмотря О.В.   |        |      |                                            | 4    |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                            |      |

## ЗМІСТ

|                                                                                                                                          |    |
|------------------------------------------------------------------------------------------------------------------------------------------|----|
| РЕФЕРАТ .....                                                                                                                            | 3  |
| ABSTRACT .....                                                                                                                           | 4  |
| ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....                                                                                                           | 7  |
| ВСТУП.....                                                                                                                               | 8  |
| РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ<br>ТЕХНОЛОГІЙ .....                                                                | 10 |
| 1.1. Постановка задачі.....                                                                                                              | 10 |
| 1.2. Аналіз аналогів програмного продукту .....                                                                                          | 11 |
| 1.3. Вибір архітектури платформи.....                                                                                                    | 16 |
| 1.4. Обґрунтування вибору інструментальних засобів.....                                                                                  | 18 |
| Висновки до першого розділу .....                                                                                                        | 20 |
| РОЗДІЛ 2. ПРОЄКТУВАННЯ ПЛАТФОРМИ ДЛЯ ВЕДЕННЯ<br>ПЕРСОНАЛІЗОВАНИХ СПИСКІВ ПЕРЕГЛЯДУ ТА ОБГОВОРЕННЯ АНІМЕ<br>.....                         | 21 |
| 2.1. Визначення варіантів використання та структури платформи.....                                                                       | 21 |
| 2.2. Розробка бази даних.....                                                                                                            | 30 |
| 2.3. Проєктування та реалізація алгоритмів роботи платформи .....                                                                        | 33 |
| 2.4. Реалізація платформи для ведення персоналізованих списків перегляду та<br>обговорення аніме .....                                   | 36 |
| Висновки до другого розділу .....                                                                                                        | 54 |
| РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ПЛАТФОРМОЮ ДЛЯ<br>ВЕДЕННЯ ПЕРСОНАЛІЗОВАНИХ СПИСКІВ ПЕРЕГЛЯДУ ТА<br>ОБГОВОРЕННЯ АНІМЕ.....        | 55 |
| 3.1 Порядок встановлення та налаштування параметрів додатку.....                                                                         | 55 |
| 3.2. Структура інтерфейсу та порядок взаємодії з платформою для ведення<br>персоналізованих списків перегляду та обговорення аніме ..... | 58 |
| 3.3. Тестування роботи програмного продукту.....                                                                                         | 66 |
| Висновки до третього розділу.....                                                                                                        | 69 |
| ВИСНОВКИ.....                                                                                                                            | 71 |

|      |      |                 |        |      |                                             |      |
|------|------|-----------------|--------|------|---------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка. 25.121.21.000 - ПЗ | Арк. |
|      |      | Чижевотря О.В.  |        |      |                                             | 5    |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                             |      |

|                                 |    |
|---------------------------------|----|
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ..... | 73 |
| ДОДАТКИ.....                    | 76 |

|      |      |                 |        |      |                                            |      |
|------|------|-----------------|--------|------|--------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка.25.121.21.000 - ПЗ | Арк. |
|      |      | Чижмотря О.В.   |        |      |                                            | 6    |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                            |      |

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

HTML – HyperText Markup Language

JS – JavaScript

URL – Uniform Resource Locator

Тайтл – це назва аніме-серіалу, фільму або манги

DOM – Document Object Model

JSON – JavaScript Object Notation

UI – User Interface

UX – User Experience

HTTP – HyperText Transfer Protocol

NPM – Node Package Manager

CLI – Command Line Interface

|      |      |                 |        |      |                                            |      |
|------|------|-----------------|--------|------|--------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка.25.121.21.000 - ПЗ | Арк. |
|      |      | Чижмотря О.В.   |        |      |                                            | 7    |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                            |      |

## ВСТУП

Темпи цифровізації та розвиток інформаційних технологій стрімко змінюють способи споживання культурного продукту та комунікації, яскравим прикладом чого є популяризація аніме. Зростає кількість українських шанувальників, які прагнуть не лише переглядати, а й вести власні списки, ділитися враженнями та обговорювати з однодумцями.

**Актуальність теми** полягає в необхідності створення інноваційних рішень, які забезпечать зручне ведення персоналізованих списків перегляду та обговорення аніме. Незважаючи на зростаючу популярність, український ринок досі не пропонує спеціалізованих веб-платформ, подібних до міжнародних MyAnimeList чи AniList, які б повною мірою відповідали потребам українських користувачів — зокрема, через відсутність української мови, локального контенту чи функцій, орієнтованих на вітчизняну спільноту. Існуючі рішення часто мають обмежений функціонал та складний інтерфейс. Створення сучасної веб-платформи для аніме є важливим і своєчасним завданням, що заповнить наявну нішу, об'єднає українську аніме-спільноту та сприятиме розвитку цифрової культури.

**Метою кваліфікаційної роботи** є розробка та впровадження веб-платформи, яка забезпечить зручне ведення персоналізованих списків перегляду аніме, а також ефективне обговорення аніме-контенту серед користувачів. Платформа має об'єднувати функції персональної бібліотеки аніме, інструментів для оцінювання та рецензування, а також можливостей для активної взаємодії між шанувальниками.

**Практичними завданнями** кваліфікаційної роботи є:

1. Проаналізувати існуючі міжнародні платформи для аніме (наприклад, MyAnimeList, AniList) та визначити їхні переваги й недоліки з точки зору потреб українських користувачів.
2. Розробити структуру веб-платформи та її функціональні можливості, орієнтовані на потреби вітчизняної аудиторії.

|      |      |                 |        |      |                                            |      |
|------|------|-----------------|--------|------|--------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка.25.121.21.000 - ПЗ | Арк. |
|      |      | Чижмотря О.В.   |        |      |                                            | 8    |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                            |      |

3. Реалізувати веб-платформу з використанням сучасних веб-технологій, зокрема HTML, CSS, JavaScript, Node.js та бази даних MongoDB, а також забезпечити зручний інтерфейс користувача.
4. Провести функціональне та Usability тестування веб-платформи для виявлення та усунення можливих помилок.
5. Оцінити ефективність та зручність використання платформи на основі збору відгуків і проведення опитувань серед цільової аудиторії.

**Об'єктом дослідження** є процес організації перегляду аніме та комунікації між шанувальниками аніме у цифровому середовищі.

**Предметом дослідження** є веб-платформа для ведення персоналізованих списків перегляду та обговорення аніме, її структура, функціональні можливості та ефективність у забезпеченні користувачів зручними інструментами для організації власної бібліотеки аніме й участі в обговореннях.

**Практична значимість** кваліфікаційної роботи полягає у створенні повноцінної веб-платформи, що задовольняє потреби української аніме-спільноти. Це буде перший вітчизняний спеціалізований ресурс для ведення персоналізованих списків перегляду та обговорення аніме українською мовою. Платформа забезпечить зручний інструмент для організації аніме-колекції, обміну враженнями та інтеграції в локальні тематичні спільноти, сприяючи їхньому розвитку та слугуючи прикладом реалізації сучасних веб-технологій у сфері культурного дозвілля.

|      |      |                 |        |      |                                            |      |
|------|------|-----------------|--------|------|--------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка.25.121.21.000 - ПЗ | Арк. |
|      |      | Чижмотря О.В.   |        |      |                                            | 9    |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                            |      |

# РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

## 1.1. Постановка задачі

Основна мета роботи — створення сучасного та зручного інструменту для шанувальників аніме, яким є веб-платформа для ведення персоналізованих списків перегляду та обговорення аніме. Вона має об'єднувати функціональність каталогізації переглянутого контенту, публікації відгуків і рецензій, участі в обговореннях та формування спільнот за інтересами. Така платформа має стати українським аналогом популярних міжнародних сервісів, з урахуванням локальних особливостей і потреб вітчизняних користувачів.

Ключовим елементом платформи є система реєстрації та авторизації, що забезпечує створення користувацьких профілів з можливістю входу через електронну пошту. В подальшому планується розширити функціонал, додавши вхід через популярні соціальні мережі, що спростить доступ для нових користувачів. Особливу увагу приділено інструментам для ведення власних списків перегляду, користувачі можуть додавати аніме до бібліотеки, позначати статус перегляду позначати статус перегляду, наприклад: переглядаю, переглянуто, планую подивитися, залишати оцінки та писати короткі замітки або розгорнуті рецензії до кожного тайтлу. Це забезпечує високий рівень персоналізації та зручності.

Наступним ключовим блоком є система обговорень і коментарів, що дає змогу користувачам ділитися своїми думками, обговорювати сюжетні лінії, персонажів чи художнє оформлення аніме. Для підвищення якості контенту передбачено можливість голосування за рецензії та коментарі, що дозволяє виділяти найбільш корисні та цікаві матеріали. Реалізована система пошуку надає швидкий доступ до потрібного контенту — як серед аніме-тайтлів, так і серед рецензій та обговорень — із можливістю фільтрації за жанрами, рейтингами, роками випуску та іншими параметрами.

|      |      |                 |        |      |                                            |      |
|------|------|-----------------|--------|------|--------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка.25.121.21.000 - ПЗ | Арк. |
|      |      | Чижемотря О.В.  |        |      |                                            | 10   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                            |      |

Важливою складовою платформи є модерація, що забезпечує контроль за дотриманням правил спільноти. Адміністратори й модератори мають змогу редагувати або видаляти невідповідний контент, блокувати користувачів, які порушують правила, а також керувати тегами й категоріями для підтримання порядку та зручності пошуку. Це створює комфортне й безпечне середовище для спілкування всередині української аніме-спільноти.

У сукупності всі ці компоненти створюють гнучку, масштабовану й безпечну платформу, здатну задовольнити потреби як індивідуальних користувачів, так і спільнот у сфері культурного дозвілля.

Процес розробки платформи включає кілька ключових етапів. Спочатку здійснюється проєктування та побудова архітектури, що охоплює як серверну, так і клієнтську частину, а також визначаються основні технічні компоненти. Після цього починається робота над інтерфейсом користувача, що включає створення дизайну вебсторінок платформи.

Наступним етапом є розробка основного функціоналу платформи. Зокрема, реалізується система авторизації та аутентифікації користувачів, інструменти для ведення списків перегляду, система обговорень та пошуку. Завершальним етапом є тестування розробленої платформи, яке передбачає перевірку її стабільності, безпеки, коректності роботи всіх функцій та відповідності поставленим вимогам.

Результатом реалізації поставленого завдання є веб-орієнтована платформа для ведення персоналізованих списків перегляду та обговорення аніме, яка забезпечує зручний доступ до контенту та можливості для взаємодії між шанувальниками.

## 1.2. Аналіз аналогів програмного продукту

Основна мета платформи для ведення персоналізованих списків перегляду та обговорення аніме полягає у створенні сучасного та зручного інструменту для шанувальників аніме. Вона об'єднує функціональність каталогізації переглянутого контенту, публікації відгуків і рецензій, участі в обговореннях та

|      |      |                 |        |      |                                            |      |
|------|------|-----------------|--------|------|--------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка.25.121.21.000 - ПЗ | Арк. |
|      |      | Чижемотря О.В.  |        |      |                                            | 11   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                            |      |

формування спільнот за інтересами. Така платформа має стати українським аналогом популярних міжнародних сервісів, на зразок MyAnimeList чи AniList, з урахуванням локальних особливостей і потреб вітчизняних користувачів. Відносно цих критеріїв розглянемо існуючі аналогічні платформи для аніме-спільнот.

MyAnimeList

MyAnimeList (рис 1.1) є однією з найбільш відомих і довгоіснуючих платформ для аніме-спільноти. Вона надає користувачам розширені можливості для ведення списків перегляду, детального оцінювання аніме, написання рецензій та активної участі у форумних обговореннях, що сприяє формуванню великої та лояльної аудиторії.

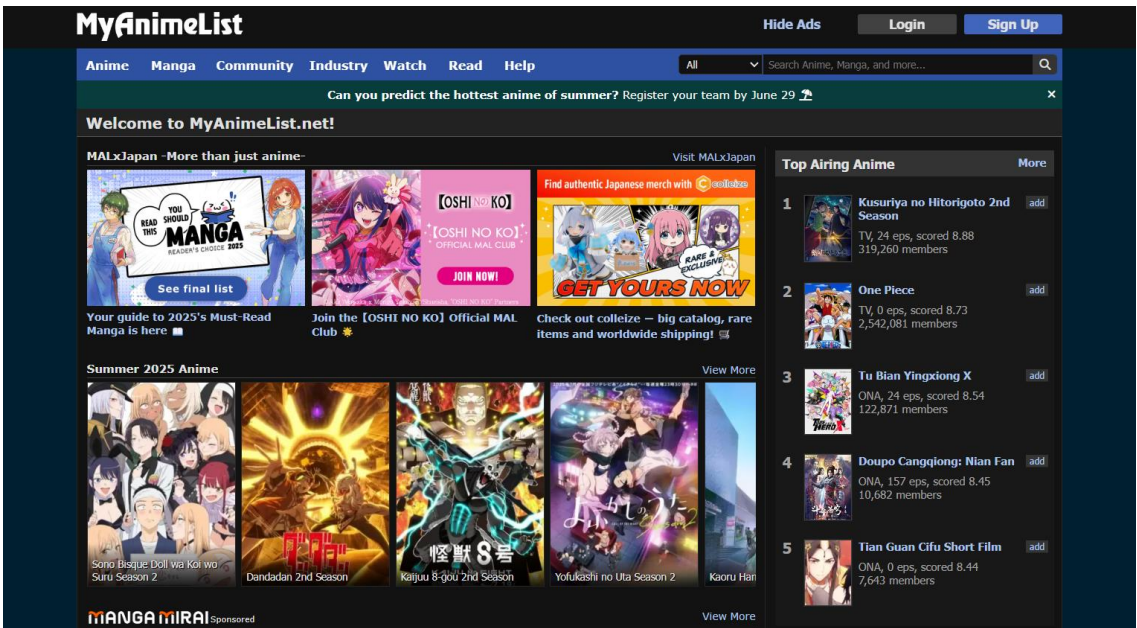


Рисунок 1.1 - MyAnimeList [1]

Переваги платформи включають зручний інтерфейс для ведення списків перегляду та всеосяжну систему рейтингів і рекомендацій, що допомагає користувачам відкривати новий контент. Крім того, є велика та активна спільнота користувачів, яка забезпечує живу взаємодію та обмін думками, а також глибока інтеграція з величезною базою даних аніме, що містить детальну інформацію про серіали, фільми, персонажів, режисерів та студії.

Водночас платформа має певні недоліки, що можуть обмежувати її використання. До них належить досить застарілий дизайн інтерфейсу, який може здатися неінтуїтивним для нових користувачів, та обмежений функціонал для нових користувачів з відсутністю деяких сучасних механізмів взаємодії.

## AniList

AniList (рис 1.2) позиціонує себе як сучасна та інноваційна платформа, що фокусується на глибокій персоналізації списків перегляду та наданні зручного й візуально привабливого інтерфейсу для користувачів. Платформа активно використовує передові технології, такі як GraphQL, для ефективної та швидкої роботи з даними.

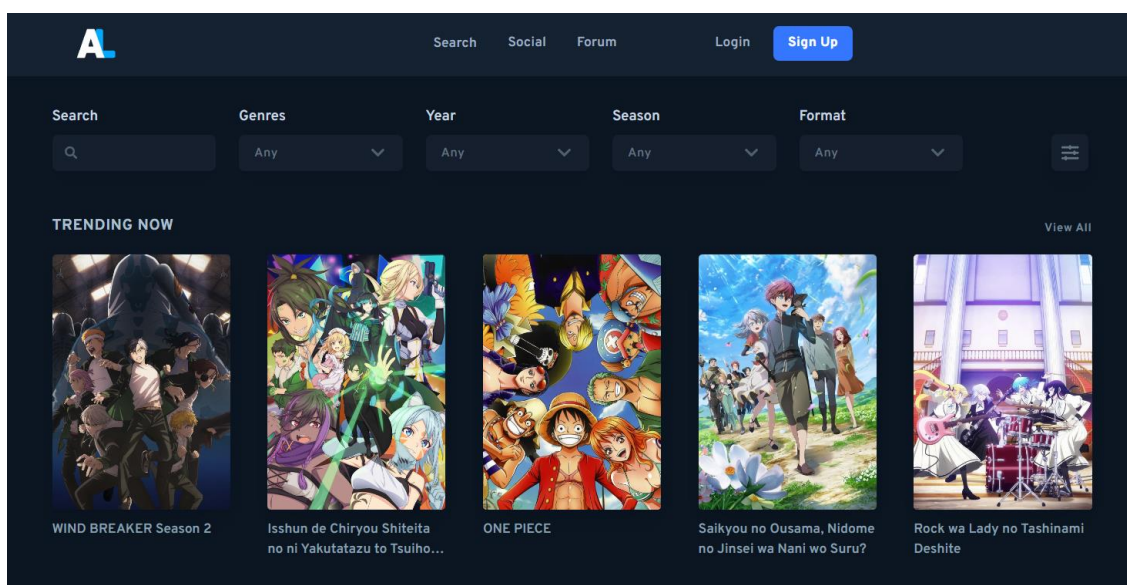


Рисунок 1.2 - AniList [2]

Серед переваг цієї платформи варто відзначити сучасний мінімалістичний та адаптивний дизайн, що забезпечує приємний користувацький досвід на різних пристроях, а також можливість глибокої персоналізації списків перегляду та налаштування інтерфейсу під індивідуальні вподобання користувача. Крім того, є широка інтеграція з соціальними мережами та сторонніми додатками, що розширює можливості взаємодії та обміну контентом, і покращена система

|      |      |                 |        |      |                                            |      |
|------|------|-----------------|--------|------|--------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка.25.121.21.000 - ПЗ | Арк. |
|      |      | Чижемотря О.В.  |        |      |                                            | 13   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                            |      |

рекомендацій, яка використовує складні алгоритми для пропонування аніме на основі інтересів та історії переглядів користувача.

До недоліків можна віднести відносно невелику спільноту порівняно з гігантами на кшталт MyAnimeList, що може обмежувати кількість доступних обговорень, а також той факт, що деякі розширені функції доступні лише для зареєстрованих користувачів, що може стати бар'єром для швидкого ознайомлення з усім функціоналом платформи.

## Kitsu

Kitsu (рис 1.3) — це унікальна платформа, яка успішно поєднує в собі функціонал соціальної мережі та комплексної системи для ведення списків перегляду аніме та манги. Вона активно використовує принципи соціальних взаємодій, таких як лайки, коментарі, спільні обговорення та можливість стежити за іншими користувачами, створюючи динамічне та інтерактивне середовище.

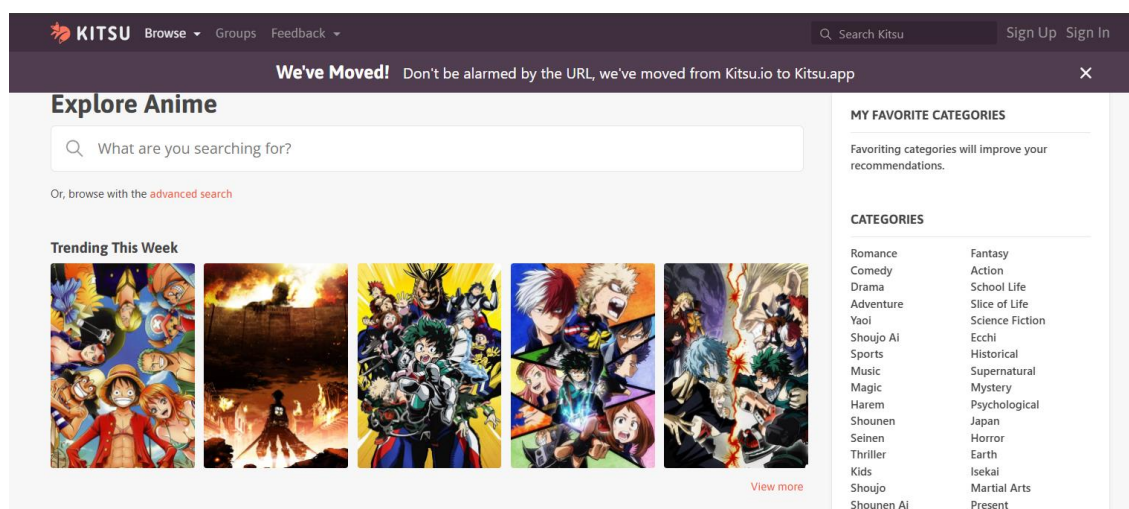


Рисунок 1.3 - Kitsu [3]

Серед переваг цієї платформи варто відзначити сильну соціальну складову платформи, що дозволяє користувачам активно взаємодіяти, слідкувати за друзями, ділитися своїми списками та враженнями. Також важливою перевагою є наявність мобільного додатку з гарним дизайном, що забезпечує зручний

|      |      |                 |        |      |                                             |      |
|------|------|-----------------|--------|------|---------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка. 25.121.21.000 - ПЗ | Арк. |
|      |      | Чижемотря О.В.  |        |      |                                             | 14   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                             |      |

доступ до функціоналу платформи на мобільних пристроях, та активна підтримка користувачів і регулярні оновлення.

До недоліків можна віднести дещо обмежений функціонал для модерації контенту, що іноді може призводити до появи невідповідних матеріалів або порушень правил спільноти, а також можлива нестабільність роботи через періодичні технічні проблеми або високе навантаження на сервери.

Аналіз існуючих платформ показав, що кожна з них має свої унікальні переваги та недоліки. На основі цього аналізу складемо порівняльну таблицю вищезгаданих програмних аналогів (табл. 1.1).

Таблиця 1.1

Порівняння функціональних можливостей платформ для аніме-спільнот

| Функціональні можливості | MyAnimeList | AniList | Kitsu    | Розроблена платформа |
|--------------------------|-------------|---------|----------|----------------------|
| Зручний інтерфейс        | Так         | Так     | Так      | Так                  |
| Персоналізація списків   | Так         | Так     | Так      | Так                  |
| Активна спільнота        | Так         | Так     | Так      | Так                  |
| Мобільний додаток        | Ні          | Ні      | Так      | Планується           |
| Модерація контенту       | Так         | Так     | Обмежено | Так                  |
| Українська локалізація   | Ні          | Ні      | Ні       | Так                  |
| Системні рекомендації    | Так         | Так     | Так      | Так                  |
| Соціальні механізми      | Обмежено    | Так     | Так      | Так                  |
| Інтеграція з БД аніме    | Так         | Так     | Так      | Так                  |
| Чат/Стрім-обговорення    | Ні          | Ні      | Ні       | Планується           |

Ключові напрямки для покращення майбутньої веб-платформи включають створення сучасного дизайну з мінімалістичним та інтуїтивно зрозумілим інтерфейсом, надання глибокої персоналізації можливості налаштування списків перегляду та рекомендацій, впровадження розширених соціальних функцій як-от чати та спільні обговорення, а також покращеної системи рекомендацій та аналітики. Надзвичайно важливою є повна локалізація для залучення

україномовної аудиторії, що вигідно відрізнятиме платформу від більшості існуючих аналогів.

Розроблена аніме-платформа об'єднує ключові переваги сучасних рішень та долає їхні недоліки. Вона забезпечує підтримку української мови, можливість глибокої персоналізації, розширені соціальні функції, а також сучасний дизайн та покращену систему рекомендацій. Саме це дозволяє зробити продукт максимально адаптованим до локальних потреб і зручним для широкого кола користувачів.

### 1.3. Вибір архітектури платформи

Для задоволення функціональних вимог до веб-платформи, яка має стати зручним та ефективним інструментом для аніме-спільноти, було проведено ретельний аналіз кількох сучасних технологій, що є основою для побудови масштабованих і продуктивних веб-додатків. Зокрема, глибокий аналіз охоплював популярні фреймворки для фронтенду, такі як React.js [4], Vue.js [5] та Angular [6], які відомі своєю здатністю створювати динамічні та інтерактивні користувацькі інтерфейси. Також були детально розглянуті рішення для серверної частини, як-от Express.js [7], Django [8] та Spring Boot [9], що забезпечують надійну обробку даних, реалізацію бізнес-логіки та ефективну взаємодію з базами даних. Окрему увагу приділено різним системам керування базами даних, серед яких MongoDB [10], PostgreSQL [11] та MySQL [12], кожна з яких має свої унікальні переваги для зберігання та управління різноманітними даними.

У результаті цього глибокого вивчення можливостей та характеристик зазначених технологій було обрано оптимальний стек для реалізації даного проєкту, що дозволить досягти не лише високої продуктивності та гнучкості, але й забезпечить необхідну масштабованість системи в майбутньому. Для клієнтської частини, яка безпосередньо відповідає за візуалізацію та взаємодію з користувачами, було обрано React.js. Цей фреймворк від Meta надає розробникам широкі можливості для створення гнучкого й динамічного

|      |      |                 |        |      |                                             |      |
|------|------|-----------------|--------|------|---------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка. 25.121.21.000 - ПЗ | Арк. |
|      |      | Чижомотря О.В.  |        |      |                                             | 16   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                             |      |

інтерфейсу з високою швидкістю відгуку [13]. Ключовою перевагою React.js є використання віртуального DOM, що дозволяє ефективно оновлювати лише ті окремі компоненти інтерфейсу, які змінилися, без потреби повного перезавантаження всієї сторінки. Цей механізм суттєво покращує користувацький досвід та є критично важливим для платформи з великою кількістю інтерактивних елементів, таких як динамічні списки перегляду, деталізовані рецензії, система рейтингів та активні коментарі. Крім того, модульна та компонентна архітектура React [14] значно спрощує процес розробки, тестування та подальшу підтримку коду, сприяючи його чистоті, читабельності та легкості модифікації. При реалізації клієнтської частини були активно застосовані принципи сучасної фронтенд-архітектури, що базуються на поєднанні React для побудови інтерфейсу, Redux [15] для ефективного управління станом програми та TypeScript [16] для забезпечення типобезпеки та підвищення надійності коду.

Для реалізації серверної частини, яка обробляє запити від клієнтів, реалізує основну бізнес-логіку та взаємодіє з базою даних, було обрано Express.js [17]. Це мінімалістичний, але водночас надзвичайно потужний та гнучкий фреймворк для Node.js [18]. Express.js забезпечує простоту створення ефективних REST API, пропонує широкі можливості для гнучкого налаштування маршрутизації та легку інтеграцію з різними типами баз даних. Важливою перевагою є його вбудована підтримка масштабованості [19], яка є абсолютно необхідною для роботи із зростаючою кількістю користувачів платформи та постійним збільшенням обсягів даних. При створенні архітектури застосунку були використані перевірені шаблони проєктування [20], що дозволило підвищити масштабованість, керованість, модульність та легкість підтримки коду. Для зберігання та ефективної роботи з даними в рамках Node.js середовища застосовувався Mongoose [21] — потужна бібліотека-обгортка над MongoDB, яка значно спрощує виконання запитів, забезпечує валідацію даних на рівні моделі та автоматизує роботу з колекціями.

|      |      |                 |        |      |                                             |      |
|------|------|-----------------|--------|------|---------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка. 25.121.21.000 - ПЗ | Арк. |
|      |      | Чижемотря О.В.  |        |      |                                             | 17   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                             |      |

У якості бази даних, що є центральним сховищем усієї інформації платформи, використовується MongoDB [22]. Її документно-орієнтована структура ідеально підходить для зберігання гнучких та різномірних даних, таких як детальна інформація про аніме-тайтли, персоналізовані списки перегляду користувачів, коментарі, обговорення та інші динамічні дані. MongoDB дозволяє швидко виконувати операції читання та запису [23], що забезпечує високу продуктивність системи навіть при значних навантаженнях. Вона легко масштабується в разі збільшення обсягів даних завдяки природному механізму горизонтального шардінгу [24], що дозволяє розподіляти дані між кількома серверами. Крім того, MongoDB безперешкодно інтегрується з Node.js та Express.js [25] завдяки великій кількості готових рішень, драйверів та бібліотек, що значно прискорює розробку та подальше розгортання системи.

#### 1.4. Обґрунтування вибору інструментальних засобів

Обрана комбінація технологій — React.js для клієнтської частини, Express.js для серверної логіки та MongoDB для зберігання даних — забезпечує високу продуктивність, швидкодію та надійність платформи. Віртуальний DOM у React дозволяє миттєво відображати зміни в інтерфейсі користувача, забезпечуючи плавний та чуйний досвід взаємодії [26]. Express.js обробляє запити від клієнтів швидко й ефективно, мінімізуючи затримки та забезпечуючи високу пропускну здатність серверної частини, що критично для великої кількості одночасних користувачів [27]. MongoDB гарантує швидкий доступ до даних навіть при значних навантаженнях та великих обсягах інформації, що є ключовим для інтерактивних веб-додатків, де швидкість доступу до даних безпосередньо впливає на користувацький досвід [28]. Цей технологічний стек дозволяє не тільки реалізувати всі функціональні вимоги, але й створити масштабований, гнучкий і сучасний продукт, повністю орієнтований на потреби української аніме-спільноти. Весь процес розробки проєкту супроводжувався активним використанням системи контролю версій Git [29], що забезпечило

|      |      |                 |        |      |                                             |      |
|------|------|-----------------|--------|------|---------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка. 25.121.21.000 - ПЗ | Арк. |
|      |      | Чижемотря О.В.  |        |      |                                             | 18   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                             |      |

ефективну співпрацю команди розробників, надійне збереження всіх змін у кодї та легкість повернення до попередніх версій у разі потреби.

React.js, Express.js та MongoDB є перевіреними часом технологіями, які здобули широке визнання та активно використовуються у багатьох успішних комерційних і некомерційних проєктах по всьому світу. Їхня популярність зумовлена низкою ключових факторів: високою продуктивністю, гнучкістю у розробці, відносною простотою у використанні та, що найважливіше, активною та постійно зростаючою підтримкою з боку великої спільноти розробників [30]. Це означає, що для цих технологій доступні численні навчальні ресурси, детальна документація, широкий спектр готових бібліотек, плагінів та рішень, що значно прискорює процес розробки та дозволяє ефективно розв'язувати можливі проблеми. Багато відомих компаній, від невеликих стартапів до масштабних онлайн-платформ з мільйонами користувачів, обрали саме ці технології для побудови власних веб-додатків, що є надійним свідченням їх стабільності, надійності та готовності до роботи в умовах значного навантаження [31]. Використання цього технологічного стеку гарантує можливість не тільки швидкої та ефективної розробки, але й подальшого легкого масштабування та довгострокового обслуговування платформи для української аніме-спільноти.

Обрані технології забезпечують надзвичайно високу масштабованість і підтримку системи. React.js як фронтенд-фреймворк дозволяє легко додавати нові функції, розширювати інтерфейс користувача та підтримувати його актуальність без значних переробок архітектури, завдяки своїй компонентній структурі та модульності [32]. Express.js підтримує архітектуру серверних додатків, яка легко масштабується, дозволяючи ефективно обробляти зростаючу кількість запитів та користувачів шляхом додавання додаткових серверів або оптимізації існуючих ресурсів. MongoDB забезпечує природне горизонтальне масштабування бази даних через механізм шардінгу, що дозволяє легко розподіляти дані між кількома серверами для експоненційного підвищення продуктивності, ємності та надійності. Ця можливість є критично важливою для проєктів з великими обсягами інформації та високим потенційним

|      |      |                 |        |      |                                            |      |
|------|------|-----------------|--------|------|--------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка.25.121.21.000 - ПЗ | Арк. |
|      |      | Чижемотря О.В.  |        |      |                                            | 19   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                            |      |

навантаженням, забезпечуючи безперебійну та ефективну роботу платформи навіть при значному зростанні користувацької бази.

### Висновки до першого розділу

Таким чином, вибір технологій для розробки веб-платформи персоналізованих списків перегляду та обговорення аніме було здійснено на основі глибокого аналізу вимог проєкту, вивчення сучасних технологій, оцінки їх продуктивності, швидкодії та надійності, а також досвіду успішного використання цих технологій в інших масштабних проєктах. Крім того, враховувався професійний досвід команди розробки й можливості для подальшого розширення й масштабування системи. Використання React.js для клієнтської частини, Express.js для серверної логіки та MongoDB для зберігання даних є найбільш доцільним і ефективним вибором для створення сучасного, стабільного й гнучкого програмного продукту, який відповідає всім функціональним і технічним вимогам платформи та потребам української аніме-спільноти.

|      |      |                 |        |      |                                            |      |
|------|------|-----------------|--------|------|--------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка.25.121.21.000 - ПЗ | Арк. |
|      |      | Чижмотря О.В.   |        |      |                                            | 20   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                            |      |

## РОЗДІЛ 2. ПРОЄКТУВАННЯ ПЛАТФОРМИ ДЛЯ ВЕДЕННЯ ПЕРСОНАЛІЗОВАНИХ СПИСКІВ ПЕРЕГЛЯДУ ТА ОБГОВОРЕННЯ АНІМЕ

### 2.1. Визначення варіантів використання та структури платформи

Платформа для шанувальників аніме надає веб інтерфейс, через який користувачі можуть переглядати інформацію про аніме-тайтли, керувати особистими списками перегляду, додавати рецензії та коментарі, а також взаємодіяти з іншими учасниками спільноти. Функціональність платформи залежить від активності користувачів. Користувачів поділено на дві основні групи, а саме звичайні користувачі та адміністратори. Звичайні користувачі можуть переглядати аніме-тайтли, створювати власні списки перегляду, додавати рецензії та коментарі, а також брати участь в обговореннях. Модератори відповідають за перевірку та управління контентом, модерацію коментарів та рецензій, а також оновлення інформації про аніме-тайтли. Адміністратори відповідають за загальне управління платформою, модерування користувачів, технічну підтримку та наповнення баз знань. Для наглядної взаємодії між елементами додатку було створено Діаграму варіантів використання (рисунок 2.1).

|      |      |                 |        |      |                                            |      |
|------|------|-----------------|--------|------|--------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка.25.121.21.000 - ПЗ | Арк. |
|      |      | Чижмотря О.В.   |        |      |                                            | 21   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                            |      |

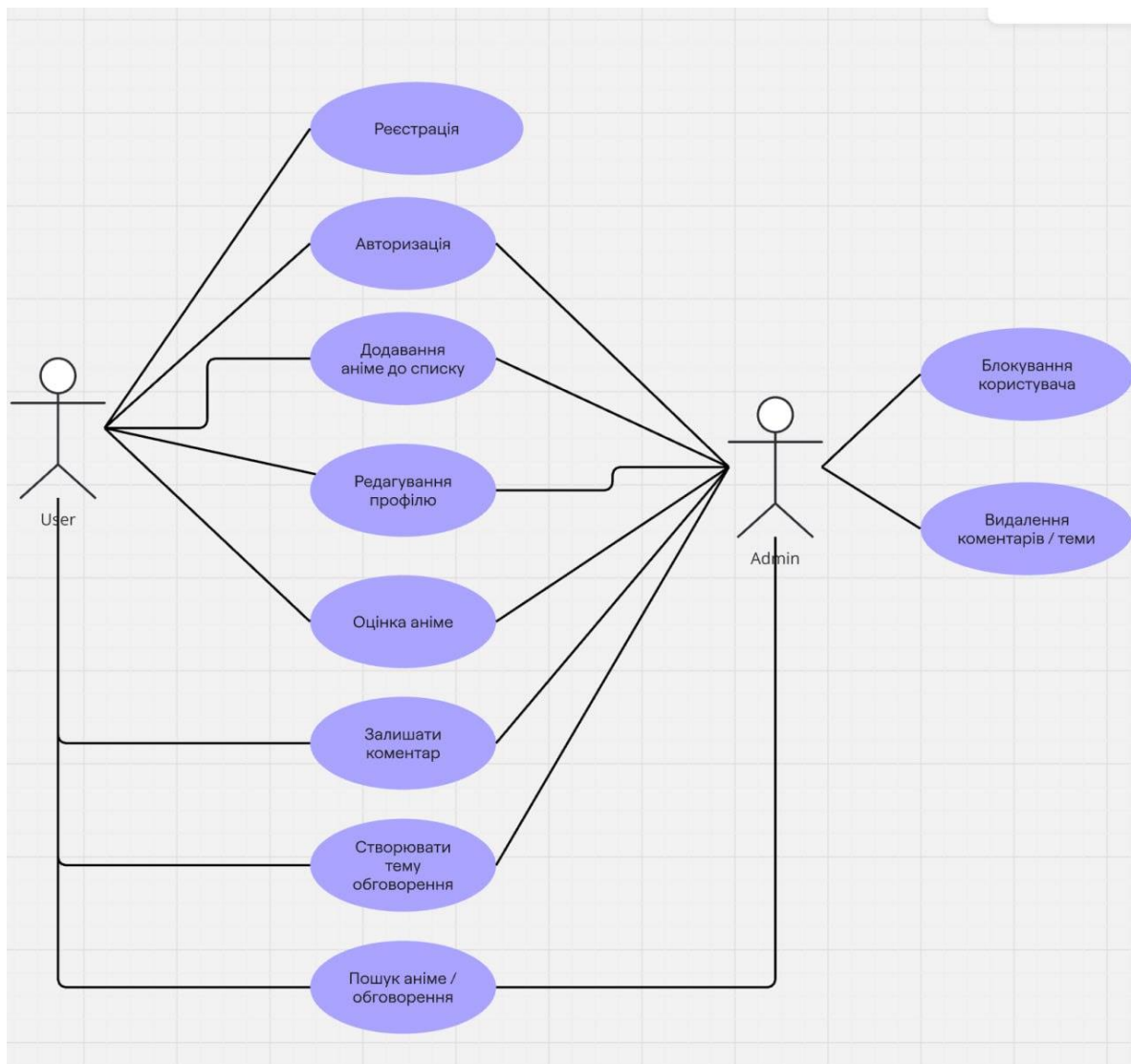


Рисунок. 2.1 - Діаграма варіантів використання

Основні варіанти використання описано нижче. (таблиця 2.1. – 2.12).

Таблиця 2.1

Опис прецеденту «Реєстрація»

| Поле                        | Опис                                                                                     |
|-----------------------------|------------------------------------------------------------------------------------------|
| Назва варіанту використання | Реєстрація                                                                               |
| Основні актори              | Користувач                                                                               |
| Короткий опис               | Користувач створює обліковий запис, надаючи необхідну інформацію для доступу до системи. |

Таблиця 2.2

## Опис прецеденту «Авторизація»

| Поле                        | Опис                                                     |
|-----------------------------|----------------------------------------------------------|
| Назва варіанту використання | Авторизація                                              |
| Основні актори              | Користувач, Адміністратор                                |
| Короткий опис               | Користувач входить у систему для доступу до функціоналу. |

Таблиця 2.3

## Опис прецеденту «Редагування профілю»

| Поле                        | Опис                                                                    |
|-----------------------------|-------------------------------------------------------------------------|
| Назва варіанту використання | Редагування профілю                                                     |
| Основні актори              | Користувач, Адміністратор                                               |
| Короткий опис               | Користувач оновлює власну інформацію та налаштування облікового запису. |

Таблиця 2.4

## Опис прецеденту «Додавання аніме до списку перегляду»

| Поле                        | Опис                                                         |
|-----------------------------|--------------------------------------------------------------|
| Назва варіанту використання | Додавання аніме до списку перегляду                          |
| Основні актори              | Користувач                                                   |
| Короткий опис               | Користувач формує персональний список аніме для відстеження. |

Таблиця 2.5

## Опис прецеденту «Оцінювання аніме»

| Поле                        | Опис                                                 |
|-----------------------------|------------------------------------------------------|
| Назва варіанту використання | Оцінювання аніме                                     |
| Основні актори              | Користувач                                           |
| Короткий опис               | Користувач залишає оцінку та відгук до аніме-тайтлу. |

Таблиця 2.6

## Опис прецеденту «Створення коментарів»

| Поле                        | Опис                                                    |
|-----------------------------|---------------------------------------------------------|
| Назва варіанту використання | Створення коментарів                                    |
| Основні актори              | Користувач                                              |
| Короткий опис               | Користувач додає коментарі до аніме або тем обговорень. |

Таблиця 2.7

## Опис прецеденту «Створення тем обговорення»

| Поле                        | Опис                                                  |
|-----------------------------|-------------------------------------------------------|
| Назва варіанту використання | Створення тем обговорення                             |
| Основні актори              | Користувач                                            |
| Короткий опис               | Користувач ініціює нові дискусії на форумі спільноти. |

Таблиця 2.8

## Опис прецеденту «Участь в обговореннях»

| Поле                        | Опис                  |
|-----------------------------|-----------------------|
| Назва варіанту використання | Участь в обговореннях |
| Основні актори              | Користувач            |

Продовження таблиці 2.8

|               |                                                                |
|---------------|----------------------------------------------------------------|
| Короткий опис | Користувач відповідає на коментарі та бере участь у дискусіях. |
|---------------|----------------------------------------------------------------|

Таблиця 2.9

Опис прецеденту «Пошук аніме або обговорень»

| Поле                        | Опис                                                                          |
|-----------------------------|-------------------------------------------------------------------------------|
| Назва варіанту використання | Пошук аніме або обговорень                                                    |
| Основні актори              | Користувач, Адміністратор                                                     |
| Короткий опис               | Актор використовує інструменти пошуку для знаходження контенту за критеріями. |

Таблиця 2.10

Опис прецеденту «Перегляд каталогу аніме»

| Поле                        | Опис                                                      |
|-----------------------------|-----------------------------------------------------------|
| Назва варіанту використання | Створення нових підписок                                  |
| Основні актори              | Користувач, Адміністратор                                 |
| Короткий опис               | Актор переглядає базу даних аніме та існуючі обговорення. |

Таблиця 2.11

Опис прецеденту «Адміністрування даних для ІІІ»

| Поле                        | Опис                   |
|-----------------------------|------------------------|
| Назва варіанту використання | Блокування користувача |

Продовження таблиці 2.11

|                |                                                                   |
|----------------|-------------------------------------------------------------------|
| Основні актори | Адміністратор                                                     |
| Короткий опис  | Адміністратор блокує користувача, який порушує правила платформи. |

Таблиця 2.12

## Опис прецеденту «Видалення коментарів або тем обговорень»

| Поле                        | Опис                                                             |
|-----------------------------|------------------------------------------------------------------|
| Назва варіанту використання | Видалення коментарів або тем обговорень                          |
| Основні актори              | Адміністратор                                                    |
| Короткий опис               | Адміністратор видаляє контент, що суперечить правилам платформи. |

Основною метою створення платформи для шанувальників аніме є формування сучасного онлайн-простору, що забезпечує зручний та інтерактивний спосіб для відстеження аніме, обміну думками та взаємодії у спільноті. Платформа об'єднує функції персоналізованого перегляду та активні обговорення, створюючи умови для залучення та утримання користувачів.

Її призначення полягає у підтримці шанувальників аніме шляхом надання можливості створювати та керувати списками перегляду, оцінювати та коментувати аніме, а також брати участь у тематичних обговореннях на форумі. Платформа також дозволяє здійснювати контроль якості контенту та забезпечувати безпечне середовище для спілкування через функції модерації. Таким чином, система сприяє розвитку аніме-спільноти, розширює доступ до інформації про аніме та забезпечує зручність спілкування незалежно від місцезнаходження користувача.

## Вимоги користувачів

Звичайний користувач має можливість реєструватися та авторизуватися на платформі, а також редагувати свій особистий профіль, включаючи зміну аватару, імені та опису. Він може додавати аніме до списку перегляду, оцінювати його та залишати відгуки. Користувач також здатен створювати коментарі до аніме та тем обговорень, започатковувати нові теми на форумі та брати участь в існуючих обговореннях, відповідаючи на коментарі. Для зручності передбачено пошук аніме або обговорень за критеріями та перегляд загального каталогу аніме та існуючих обговорень.

Адміністратор, після авторизації на платформі, має право редагувати власний профіль, здійснювати пошук аніме або обговорень та переглядати каталог аніме й існуючі обговорення. Додатково, адміністратор наділений повноваженнями блокувати користувачів, які порушують правила платформи, а також видаляти коментарі або теми обговорень, що суперечать встановленим правилам.

## Функціональні вимоги

Платформа повинна забезпечувати реєстрацію та авторизацію користувачів, а також можливість публікації питань та відповідей. Необхідно реалізувати систему голосування та рейтингу, можливість коментування питань та відповідей. Важливою функціональністю є пошук за тегами, авторами та контентом, а також ефективна модерація контенту.

## Нефункціональні вимоги

Система повинна демонструвати високу продуктивність та швидкодію. Вона має бути масштабованою для підтримки великої кількості користувачів. Окрім цього, критично важливою є безпека даних та захист від веб-уразливостей, а також зручність користування та інтуїтивно зрозумілий інтерфейс.

Аналіз функціональних вимог дозволив виділити наступні сутності, що забезпечать програмну реалізацію додатку. На рисунку 2.2. наведено діаграму класів, яка використовується для його роботи.

|      |      |                 |        |      |                                             |      |
|------|------|-----------------|--------|------|---------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка. 25.121.21.000 - ПЗ | Арк. |
|      |      | Чижемотря О.В.  |        |      |                                             | 27   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                             |      |

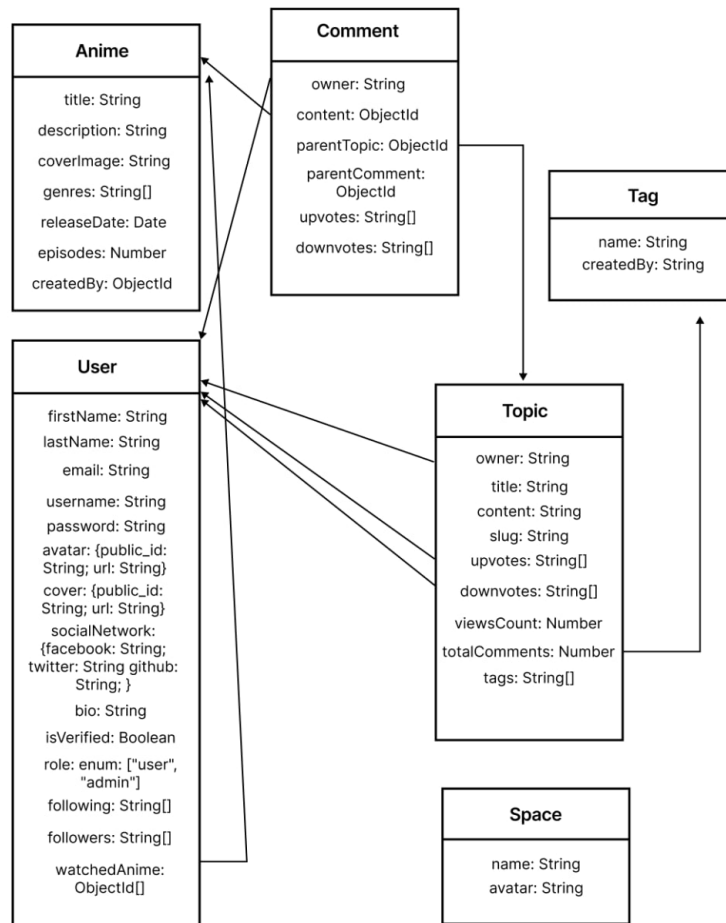


Рисунок 2.2 - Діаграма класів

У системі для аніме-спільноти визначено кілька ключових класів, кожен з яких відповідає за зберігання специфічної інформації та її взаємодію з іншими сутностями.

Клас Anime відповідає за сутність аніме-тайтлу та зберігання інформації про аніме-серіали чи фільми на платформі. Він містить основну інформацію про аніме, включаючи його назву, опис, візуальне представлення, а також дані для категоризації за жанрами та датою випуску. Клас також відстежує кількість епізодів і зберігає дані про користувача, який додав це аніме до системи. Ця модель використовується для створення колекції аніме, які будуть доступні користувачам для перегляду, обговорення та додавання до списку переглянутого.

Клас Comment призначений для збереження інформації про окремі коментарі, залишені користувачами, та реалізує систему коментування на платформі. Він містить текстовий зміст повідомлення, відомості про автора, а також посилання на тему, до якої належить коментар, і, за потреби, на батьківський коментар, якщо це відповідь. Додатково, цей клас відстежує реакції користувачів у вигляді голосів "за" та "проти". Вона дозволяє створювати гілки обговорень, формувати ланцюжки відповідей та організовувати рейтингування коментарів для підняття найкорисніших у верх списку.

Клас Space визначає сутність простору, спільноти чи групи, в межах якого створюються обговорення або об'єднуються користувачі за інтересами. Він містить назву простору, яка є ключовою для його ідентифікації, та візуальне представлення у вигляді аватара. Модель дозволяє будувати логічну ієрархію на форумі, наприклад, окремі простори для різних жанрів аніме або типів дискусій.

Клас Tag використовується для категоризації та позначення тем, обговорень або інших об'єктів у системі, а також для організації і класифікації тем. Він містить унікальну назву тега та інформацію про користувача, який його створив. Це дозволяє користувачам швидко знаходити теми за категоріями, створювати структуру і навігацію за інтересами, а також підтримувати порядок у великій кількості обговорень.

Клас Topic відповідає за створення та управління темами обговорень. Він містить інформацію про користувача-власника теми, її заголовок та основний зміст. Для зручності навігації та пошуку передбачено унікальний ідентифікатор для URL. Клас також відстежує реакції користувачів, лічильник переглядів, загальну кількість коментарів та список пов'язаних тегів. Модель також містить голоси за і проти, що дозволяє користувачам оцінювати якість і релевантність теми, і є основою для побудови форумної частини платформи, де розгортаються дискусії.

Клас User відповідає за сутність користувача платформи та зберігає всю його основну інформацію. Він містить ідентифікаційні дані, облікові дані, а також посилання на зображення профілю. Клас дозволяє підключати соціальні

|      |      |                 |        |      |                                            |      |
|------|------|-----------------|--------|------|--------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка.25.121.21.000 - ПЗ | Арк. |
|      |      | Чижемотря О.В.  |        |      |                                            | 29   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                            |      |

мережі, зберігати коротку біографію, статус підтвердження облікового запису та роль користувача в системі. Додатково, він підтримує функціонал відстеження інших користувачів та зберігає список переглянутих аніме для персоналізації. Ця модель є ключовою для аутентифікації, авторизації та взаємодії користувачів із контентом платформи.

## 2.2. Розробка бази даних

У якості бази даних використовується MongoDB. Її документно-орієнтована структура ідеально підходить для зберігання гнучких та різномірних даних, таких як інформація про тайтли, списки перегляду, коментарі та обговорення. MongoDB дозволяє швидко виконувати операції читання та запису, легко масштабуватися в разі збільшення обсягів даних і безперешкодно інтегрується з Node.js завдяки великій кількості готових рішень та бібліотек.

База даних системи складається з 6 таблиць:

1. Anime – таблиця, яка відповідає за сутність аніме-тайтлу та зберігає основну інформацію про нього.
2. Comment – таблиця, призначена для збереження інформації про коментарі користувачів.
3. Space – таблиця, яка визначає простір в межах платформи.
4. Tag – таблиця, яка використовується для категоризації та позначення контенту.
5. Topic – таблиця, яка відповідає за створення та управління темами обговорень.
6. User – таблиця, яка представляє користувача платформи та зберігає його основну інформацію.

Опис кожної з колекцій бази даних наведений у таблицях 2.13–2.18.

Таблиця 2.13

### Опис таблиці Anime

| Поле  | Тип даних | Опис         |
|-------|-----------|--------------|
| title | String    | Назва аніме. |

|      |      |                 |        |      |                                             |      |
|------|------|-----------------|--------|------|---------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка. 25.121.21.000 - ПЗ | Арк. |
|      |      | Чижевотря О.В.  |        |      |                                             | 30   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                             |      |

Продовження таблиці 2.13

|             |          |                           |
|-------------|----------|---------------------------|
| description | String   | Опис аніме                |
| coverImage  | String   | URL-адреса обкладинки     |
| genres      | String[] | Масив жанрів аніме        |
| releaseDate | Date     | Дата першого показу       |
| episodes    | Number   | Кількість серій           |
| createdBy   | ObjectId | Ідентифікатор користувача |

Таблиця 2.14

Опис таблиці Comment

| Поле          | Тип даних | Опис                           |
|---------------|-----------|--------------------------------|
| owner         | String    | Ім'я/ідентифікатор користувача |
| content       | ObjectId  | Текст коментаря                |
| parentTopic   | ObjectId  | Ідентифікатор теми             |
| parentComment | ObjectId  | Ідентифікатор для відповіді    |
| upvotes       | String[]  | Ідентифікатор лайків           |
| downvotes     | String[]  | Ідентифікатор дизлайків        |

Таблиця 2.15

Опис таблиці Space

| Поле   | Тип даних | Опис                   |
|--------|-----------|------------------------|
| name   | String    | Ідентифікатор простору |
| avatar | String    | URL/шлях зображення    |

Таблиця 2.16

Опис таблиці Tag

| Поле      | Тип даних | Опис                          |
|-----------|-----------|-------------------------------|
| name      | String    | Ключ для пошуку та фільтрації |
| createdBy | String    | Створювач тегу                |

Таблиця 2.17

## Опис таблиці Topic

| Поле          | Тип даних | Опис                          |
|---------------|-----------|-------------------------------|
| owner         | String    | Створювач теми                |
| title         | String    | Короткий опис суті            |
| content       | String    | Основний текст                |
| slug          | String    | URL-посилання                 |
| upvotes       | String[]  | ID користувачів для лайків    |
| downvotes     | String[]  | ID користувачів для дизлайків |
| viewsCount    | Number    | Лічильник популярності        |
| totalComments | Number    | Лічильник активності          |
| tags          | String[]  | Категоризація, пошук          |

Таблиця 2.18

## Опис таблиці User

| Поле                   | Тип даних | Опис                       |
|------------------------|-----------|----------------------------|
| firstName              | String    | Ім'я користувача           |
| lastName               | String    | Прізвище користувача       |
| email                  | String    | Email вхід                 |
| username               | String    | Унікальне ім'я користувача |
| password               | String    | Пароль користувача         |
| avatar.public_id       | String    | ID аватара                 |
| avatar.url             | String    | Посилання аватара          |
| cover.public_id        | String    | Ідентифікатор обкладинки   |
| cover.url              | String    | URL обкладинки профілю     |
| socialNetwork.facebook | String    | Посилання на Facebook      |

|                       |            |                         |
|-----------------------|------------|-------------------------|
| socialNetwork.twitter | String     | Посилання на Twitter    |
| socialNetwork.github  | String     | Посилання на GitHub     |
| bio                   | String     | Коротка біографія       |
| isVerified            | Boolean    | Статус верифікації      |
| role                  | String     | Роль користувача        |
| following             | String[]   | Підписки користувачів   |
| followers             | String[]   | Підписники користувачів |
| watchedAnime          | ObjectId[] | Переглянуте аніме       |

Таким чином, структура бази даних побудована на основі шести основних колекцій, що забезпечують повний цикл функціонування платформи: від реєстрації користувачів та зберігання інформації про аніме до організації коментування, обговорень, категоризації контенту та персоналізації взаємодії.

### 2.3. Проектування та реалізація алгоритмів роботи платформи

Оскільки архітектурною основою платформи обрано клієнт-серверну модель, застосунок побудований із чітким поділом на серверну та клієнтську частини. Для організації обміну даними між цими частинами використовується архітектурний стиль REST, який базується на концепції ресурсів і взаємодії через стандартні HTTP-запити. Кожен об'єкт у системі подається як ресурс з унікальною URL-адресою, а доступ до нього і керування ним реалізовані через відповідні HTTP-методи.

Зокрема, метод GET використовується для отримання даних із сервера, POST – для створення нових записів у системі, PUT – для повного оновлення існуючих ресурсів, PATCH – для часткових змін, а DELETE – для видалення ресурсів.

Серверна частина платформи реалізована за допомогою Express.js. Цей фреймворк забезпечує обробку HTTP-запитів, взаємодію з базою даних MongoDB, виконання бізнес-логіки та формування відповідей у форматі JSON. Такий підхід дозволяє забезпечити надійність роботи серверної частини, її масштабованість та легкість підтримки. Express.js надає гнучкий механізм побудови маршрутизації та підключення проміжних обробників, що дає змогу ефективно реалізувати механізми аутентифікації, авторизації, обробки помилок і логування запитів.

Клієнтська частина платформи розроблена на базі React. Ця бібліотека дозволяє створювати динамічний і швидкий інтерфейс користувача завдяки компонентному підходу та використанню віртуального DOM для оптимізації рендерингу. Після початкового завантаження сторінки користувач взаємодіє з односторінковим застосунком, де всі запити до серверного API здійснюються асинхронно за допомогою HTTP-запитів. Це забезпечує плавну роботу інтерфейсу без потреби повного перезавантаження сторінок. Таким чином, комунікація в системі побудована на уніфікованій моделі HTTP-запитів між React-клієнтом і Express-сервером. Така архітектура гарантує гнучкість, надійність та масштабованість платформи, що дозволяє легко розширювати її функціонал і підтримувати високий рівень продуктивності навіть при зростанні навантаження.

Процес проектування платформи базується на створенні чіткої архітектури, яка дозволяє ефективно обробляти запити користувачів і взаємодіяти з даними. Під час проектування були визначені основні алгоритмічні сценарії для обробки дій користувачів: реєстрації та авторизації, створення та перегляду контенту, голосування, управління зв'язками між об'єктами, а також реалізації системи пошуку та фільтрації.

Одним із ключових алгоритмів є алгоритм аутентифікації та авторизації користувачів. Він забезпечує перевірку правильності введених даних при вході в систему, створення і перевірку токенів доступу, а також керування правами доступу залежно від ролі користувача. Для цього використовуються проміжні

|      |      |                 |        |      |                                             |      |
|------|------|-----------------|--------|------|---------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка. 25.121.21.000 - ПЗ | Арк. |
|      |      | Чижемотря О.В.  |        |      |                                             | 34   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                             |      |

обробники, які аналізують заголовки запитів, перевіряють наявність і валідність токенів, згенерованих під час входу в систему. Це дозволяє забезпечити безпечний доступ до захищених ресурсів платформи.

Інший важливий алгоритм стосується роботи із контентом. Коли користувач створює новий об'єкт — тему, коментар чи тег — система здійснює валідацію даних, створення посилань між об'єктами в базі даних і автоматичне оновлення статистики, наприклад, підрахунок кількості коментарів у темі чи оновлення часу останньої активності. Ці алгоритми реалізовані через контролери, які обробляють логіку запитів, і моделі, що відповідають за зберігання й оновлення даних у MongoDB. Реалізація системи голосування вимагає розробки механізму запобігання подвійного голосу. При голосуванні користувачем перевіряється, чи існує його ідентифікатор у відповідному списку голосів. Якщо голосування вже було проведено, система може або скасувати попередній голос, або не дозволити повторного голосування, залежно від встановленої бізнес-логіки. Це забезпечує чесність рейтингової системи й мотивує користувачів створювати якісний контент.

Особливу увагу під час проєктування було приділено алгоритмам побудови ієрархічної структури коментарів. Кожен коментар може бути відповіддю як на тему, так і на інший коментар. Для цього під час додавання нового коментаря визначаються його зв'язки за допомогою полів. Таким чином формується дерево коментарів, яке потім використовується при побудові інтерфейсу для відображення вкладеності дискусій.

Алгоритм пошуку і фільтрації даних реалізований через створення індексів у базі даних за ключовими полями, такими як заголовок теми, теги, ім'я користувача або назва аніме. Запити на пошук оптимізовані для роботи з великими обсягами даних, що дозволяє забезпечити швидкий відгук системи навіть при значній кількості записів.

У частині проєктування платформи важливим є алгоритм керування просторами й тегами. При створенні нової теми або коментаря система дозволяє користувачам додавати теги, що потім використовуються для побудови системи

|      |      |                 |        |      |                                            |      |
|------|------|-----------------|--------|------|--------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка.25.121.21.000 - ПЗ | Арк. |
|      |      | Чижемотря О.В.  |        |      |                                            | 35   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                            |      |

навігації й рекомендацій. Простори забезпечують логічне групування тем для створення окремих розділів, що спрощує користувачам навігацію й участь у потрібних обговореннях.

Реалізація алгоритмів оновлення й видалення контенту передбачає перевірку прав доступу: тільки автор об'єкта або адміністратор має право змінювати або видаляти контент. Ці алгоритми інтегровані з системою авторизації й виконуються на рівні контролерів для кожного типу об'єкта.

Завдяки розділенню обов'язків між різними шарами платформи маршрутизацією, контролерами, моделями й проміжними обробниками забезпечується модульність коду і легкість його підтримки. Це дозволяє швидко розширювати функціонал системи й адаптувати її під нові потреби користувачів.

#### 2.4. Реалізація платформи для ведення персоналізованих списків перегляду та обговорення аніме

Розробка платформи для ведення персоналізованих списків перегляду та обговорення аніме є важливим етапом створення сучасного веб-застосунку, який поєднує можливості соціальної взаємодії та управління медіа контентом. На цьому етапі реалізовано технічні рішення, що забезпечують зручний інтерфейс для користувачів, швидку обробку даних і стабільну роботу системи під навантаженням. Основна увага приділялася створенню гнучкої архітектури, що дозволяє не лише зберігати списки перегляду аніме й ділитися ними з іншими користувачами, а й активно брати участь в обговореннях через коментарі, голосування та створення тематичних спільнот.

У процесі реалізації використовувались сучасні веб-технології, зокрема React для побудови клієнтської частини й Express.js у поєднанні з MongoDB для серверної логіки та збереження даних. Особливий акцент було зроблено на персоналізації: платформа дозволяє кожному користувачеві створювати власні списки аніме, відстежувати прогрес перегляду, додавати рецензії, брати участь у дискусіях і взаємодіяти з іншими учасниками спільноти. Реалізація функціоналу здійснювалася відповідно до попередньо розробленої моделі даних і діаграм

|      |      |                 |        |      |                                             |      |
|------|------|-----------------|--------|------|---------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка. 25.121.21.000 - ПЗ | Арк. |
|      |      | Чижемотря О.В.  |        |      |                                             | 36   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                             |      |

варіантів використання, що забезпечило послідовність і відповідність поставленим вимогам.

Цей розділ висвітлює основні аспекти реалізації платформи: вибір технологій, архітектурні рішення, особливості побудови клієнтської й серверної частин, а також основні алгоритми, які забезпечують коректну роботу застосунку.

Спочатку була розроблена архітектура БД і створені відповідні об'єкти:

На рисунку 2.3 представлено таблицю з прикладом збереженого документа моделі коментаря, який містить текст коментаря, ідентифікатор користувача, дату створення та інформацію про кількість голосів.

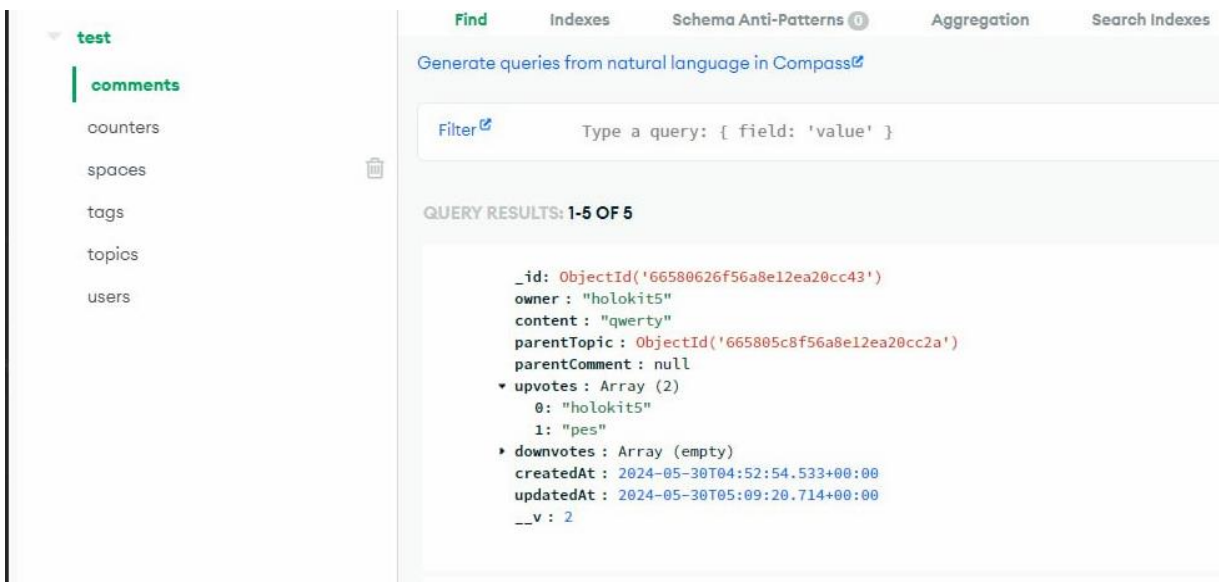


Рисунок 2.3 - Документ з коментарями

Рисунок 2.4 демонструє таблицю з прикладом збереженого документа моделі лічильника, що використовується для підрахунку кількості взаємодій із певними об'єктами.

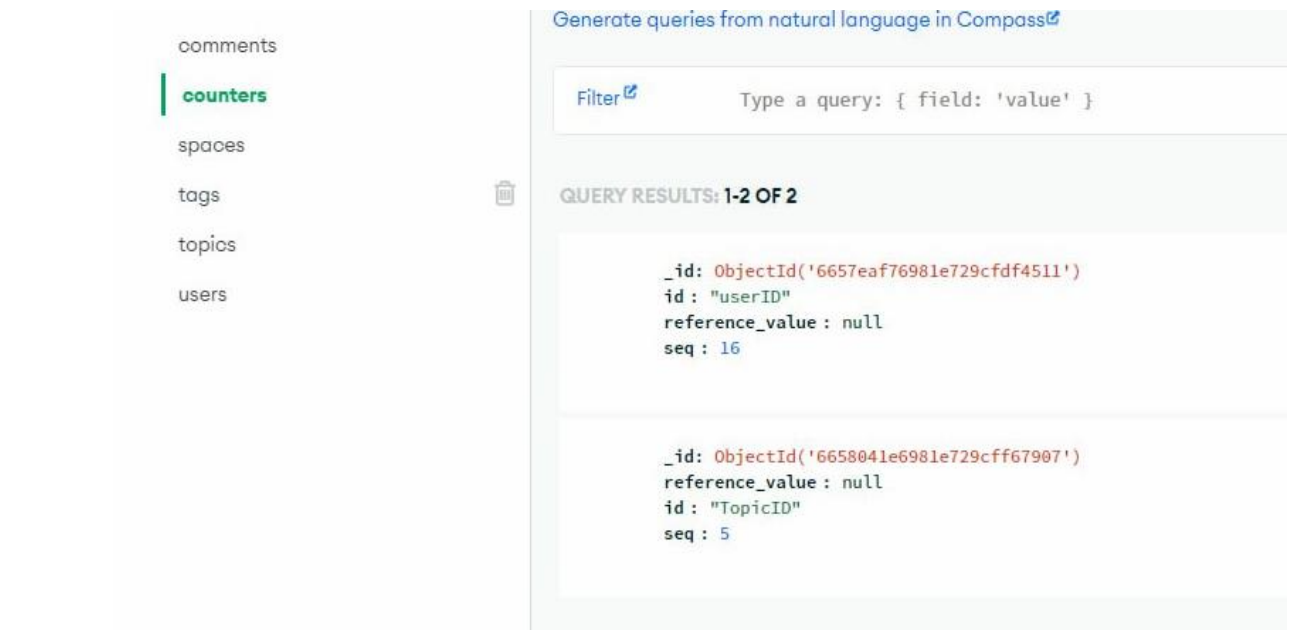


Рисунок 2.4 - Документ з лічильниками

Рисунок 2.5 показує таблицю з прикладом збереженого документа моделі простору, де зберігається інформація про окремий простір для обговорень.

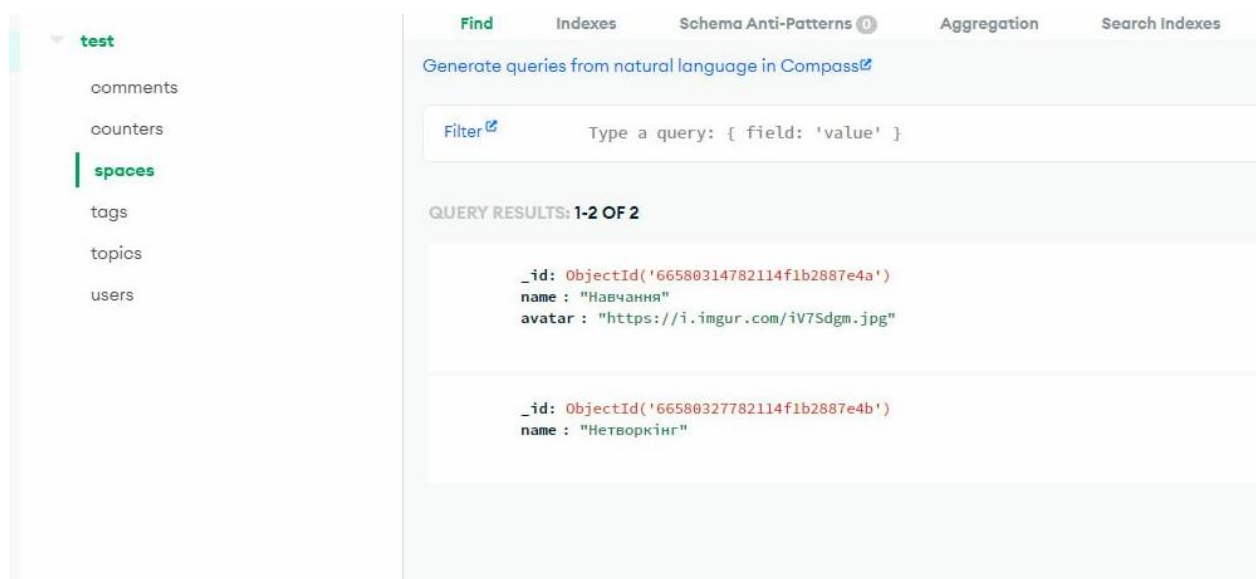


Рисунок 2.5 - Документ з просторами

На рисунку 2.6 наведено таблицю з прикладом документа моделі тегу, яка використовується для категоризації аніме та обговорень.

|      |      |                 |        |      |                                            |      |
|------|------|-----------------|--------|------|--------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка.25.121.21.000 - ПЗ | Арк. |
|      |      | Чижмотря О.В.   |        |      |                                            | 38   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                            |      |

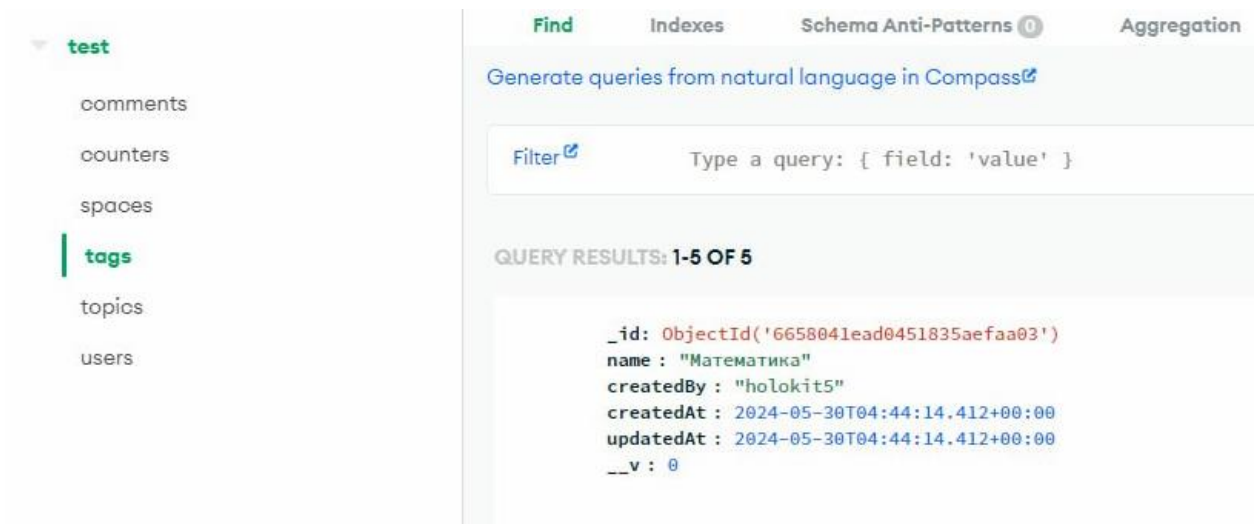


Рисунок 2.6 - Документ з тегами

Рисунок 2.7 містить таблицю з прикладом документа моделі топіка, що відображає тему обговорення та основну інформацію про неї.

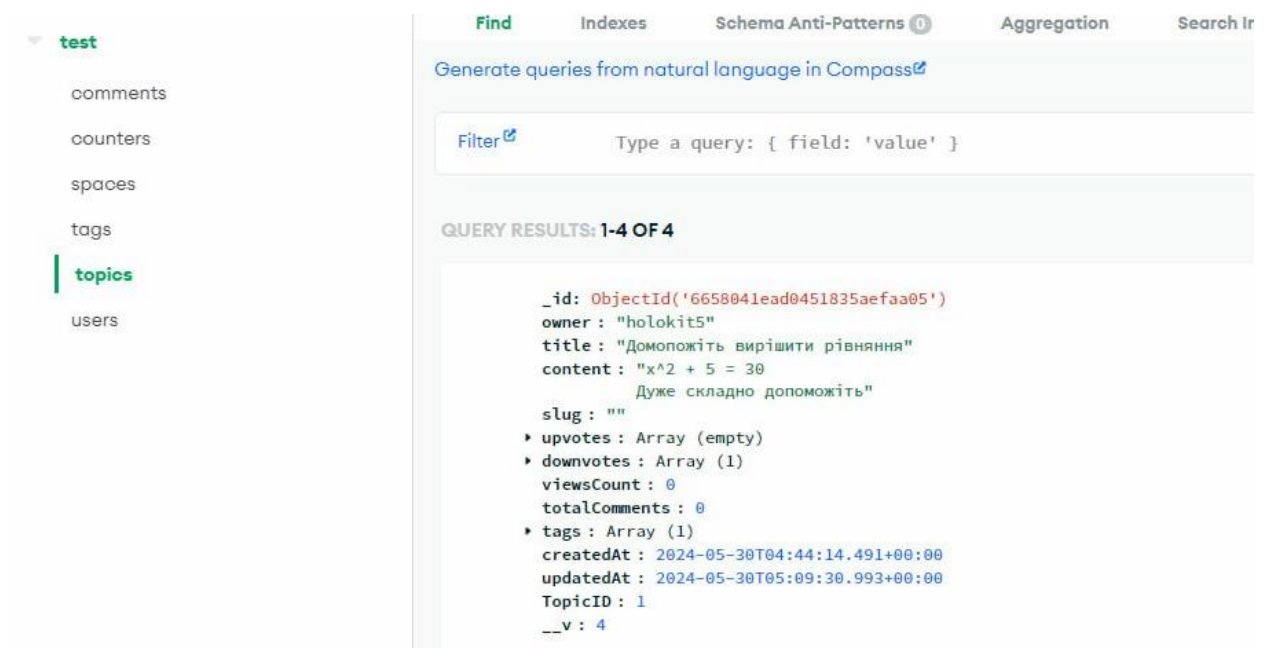


Рисунок 2.7 - Документ з топіками

Рисунок 2.8 демонструє таблицю з прикладом збереженого документа моделі користувача, де зберігаються персональні дані, такі як ім'я, електронна

|      |      |                 |        |      |                                            |      |
|------|------|-----------------|--------|------|--------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка.25.121.21.000 - ПЗ | Арк. |
|      |      | Чижмотря О.В.   |        |      |                                            | 39   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                            |      |

пошта, логін, роль користувача, аватар та інші відомості, необхідні для автентифікації й персоналізації досвіду користувача.

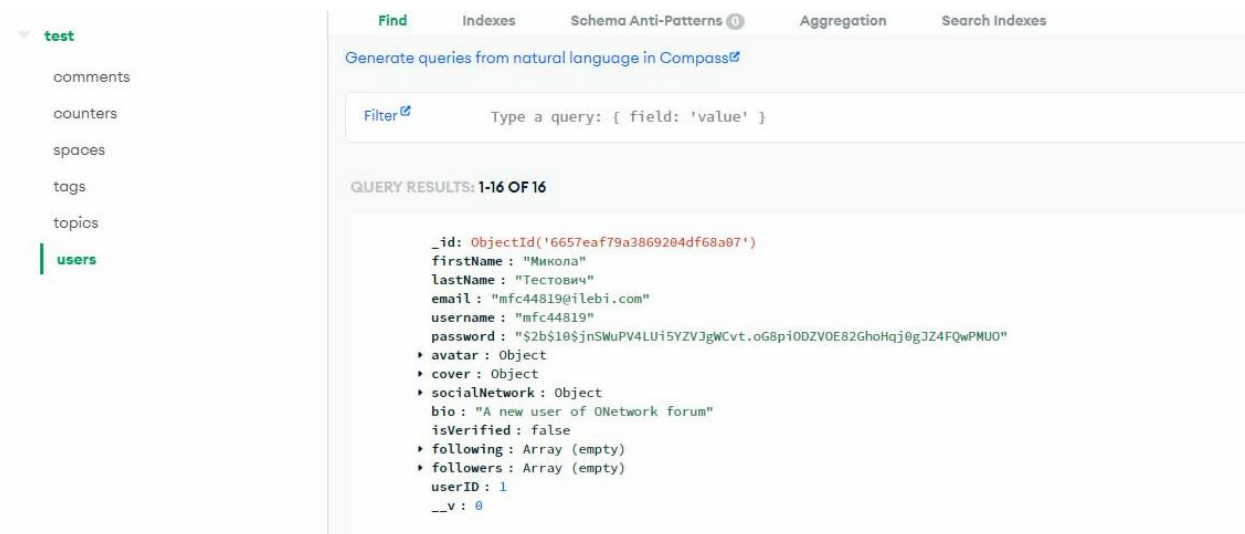


Рисунок 2.8 - Документ з топіками

Далі була створена клієнтська частина.

Рисунок 2.9 демонструє як була розроблена структура проєкту:

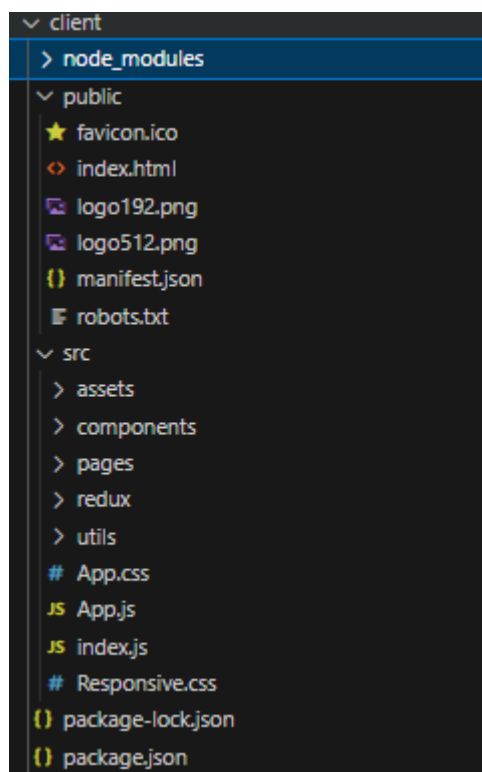


Рисунок 2.9 - Структура папок проєкту

|      |      |                 |        |      |                                            |      |
|------|------|-----------------|--------|------|--------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка.25.121.21.000 - ПЗ | Арк. |
|      |      | Чижмотря О.В.   |        |      |                                            | 40   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                            |      |

Потім були додані усі необхідні залежності:

```
"dependencies": {
  "@reduxjs/toolkit": "^1.9.1",
  "@testing-library/jest-dom": "^5.16.5",
  "@testing-library/react": "^13.4.0",
  "@testing-library/user-event": "^13.5.0",
  "@welldone-software/why-did-you-render": "^7.0.1",
  "axios": "^1.2.2",
  "bootstrap": "^5.2.3",
  "lottie-react": "^2.3.1",
  "moment": "^2.29.4",
  "react": "^18.2.0",
  "react-bootstrap": "^2.7.0",
  "react-dom": "^18.2.0",
  "react-icons": "^4.7.1",
  "react-loading-skeleton": "^3.1.0",
  "react-redux": "^8.0.5",
  "react-router-dom": "^6.6.2",
  "react-scripts": "5.0.1",
  "react-select": "^5.7.0",
  "reactstrap": "^9.1.5",
  "web-vitals": "^2.1.4"
},
```

Тут показані усі залежності які необхідні для Фронт-енда та встановлені всі модулі за допомогою пакетного менеджера npm [33].

Наступником кроком була верстка компонентів та сторінок (рисунок 2.10). Bootstrap застосовувався для швидкої розробки адаптивного дизайну інтерфейсу користувача [34].

Були зверстані такі сторінки:

- EditProfile
- EmailVerify
- ForgotPassword
- Home
- Login
- NewTopic
- NotFound
- Profile
- Register
- ResetPassword
- Topic

|      |      |                 |        |      |                                            |      |
|------|------|-----------------|--------|------|--------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка.25.121.21.000 - ПЗ | Арк. |
|      |      | Чижемотря О.В.  |        |      |                                            | 41   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                            |      |

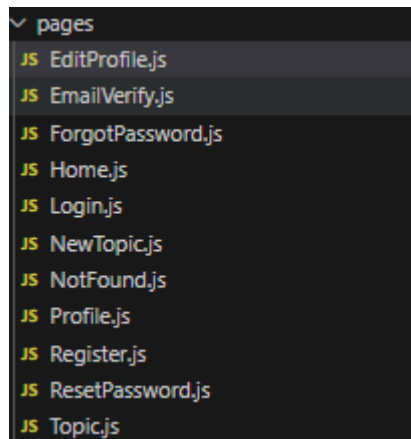


Рисунок 2.10 - Список сторінок

Для створення сторінок були обрані функціональні компоненти React.js

Підключення компонентів і бібліотек в сторінку

```
import { Col, Image, Container } from "react-bootstrap";
import { useParams } from "react-router-dom";
import { useDispatch, useSelector } from "react-redux";
import { useEffect, useMemo, useState } from "react";
import { getTopic } from "../redux/slices/topicSlice";
import CommentForm from "../components/Comment/CommentForm";
import { resetTopics } from "../redux/slices/topicSlice";
import CommentsContainer from "../components/Comment/CommentsContainer";
import TopicContent from "../components/Topic/TopicContent";
import RightSidebar from "../components/RightSidebar/RightSidebar";
import SkeletonTopicPage from "../components/Skeletons/SkeletonTopicPage";
```

Тут показано як ініціалізується компонент сторінки за допомогою функціонального компонента React.js та як беруться дані для відображення за допомогою Redux:

```
const Home = () => {
  const dispatch = useDispatch();
  const { topics, getAllTopicsIsLoading } = useSelector((state) => state.topic);
  const { sortOption, searchQuery } = useSelector((state) => state.topic);
  useEffect(() => {
    document.title = `Home | Anime Forum`;
  }, []);
  useEffect(() => {
    dispatch(resetUserProfile());
    dispatch(getAllTopics({ sortOption, searchQuery }));
  }, [dispatch, sortOption, searchQuery]);
  return (
    <div>
      <main>
      </main>
    </div>
  );
};
```

```
};
export default Home;
```

Сама jsx розмітка сторінки виглядає наступним чином:

```
<LeftSidebar />
  <Col lg={6} className="main-content">
    <div className="filter">
      <Form.Select
        name="topicsSort"
        className="custom-select"
        onChange={(e) => dispatch(setSortOption(e.target.value))}
      >
        <option value="latest">Останні обговорення</option>
      </Form.Select>
    </div>
    {!getAllTopicsIsLoading &&
      topics?.map((topic, idx) => (
        <TopicItem key={idx} topic={topic} />
      ))}
  </Col>
```

Також під час створення сторінки були розроблені компоненти для цих сторінок (рисунк 2.11).

Були створені такі компоненти:

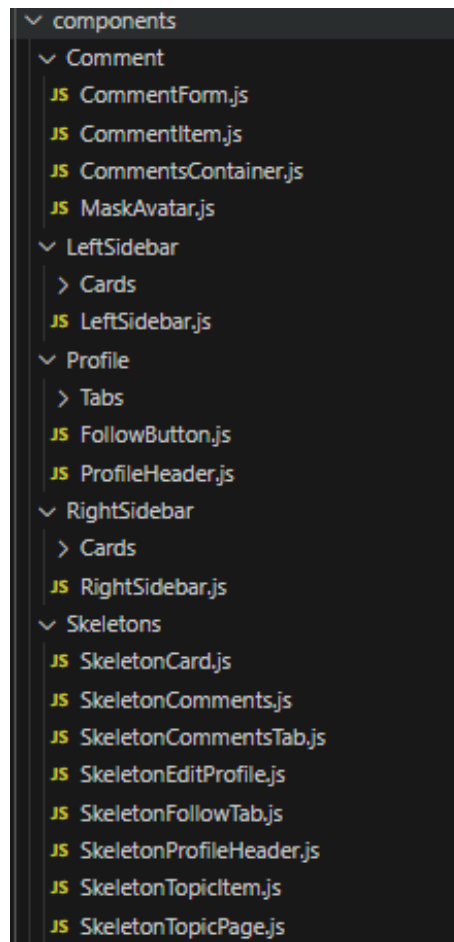


Рисунок 2.11 - Список jsx компонентів

|      |      |                 |        |      |                                            |      |
|------|------|-----------------|--------|------|--------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка.25.121.21.000 - ПЗ | Арк. |
|      |      | Чижмотря О.В.   |        |      |                                            | 43   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                            |      |

Також було створено два layout – header та footer (рисунок 2.12).

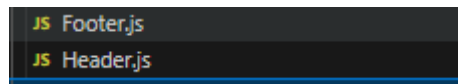


Рисунок 2.12 -Список layout

В компоненті header наявна обов'язкова перевірка на те, чи користувач залогінений для того, щоб відображати відповідно профіль або кнопки для логіну:

```
const { user, token, isHeaderLoading } = useSelector((state) => state.auth);
const isAuth = localStorage.getItem("isLoggedIn") ? true : false;
useMemo(() => {
  if (isAuth) {
    if (!token) {
      dispatch(refresh_token());
    }
  }
}, [dispatch, isAuth, token]);
{!isAuth &&
  !isHeaderLoading &&
  Object?.entries(user)?.length === 0 && (
    <>
      <Link className="login" to="/login">
        <Button className="login">Вхід</Button>
      </Link>
      <Link className="login" to="/register">
        <Button className="register">Реєстрація</Button>
      </Link>
    </>
  )}
```

Також в цьому компоненті наявний пошук по аніме та обговоренням:

```
<InputGroup>
  <span style={{ marginLeft: "8px", fontSize: "25px" }}>
    <CgSearch />
  </span>
  <FormControl
    type="text"
    placeholder="Пошук..."
    value={search}
    onChange={(e) => setSearch(e.target.value)}
    onKeyDown={handleSubmitSearch}
  />
</InputGroup>
</Form>
```

Вигляд компонент кнопки (рисунок 2.13):

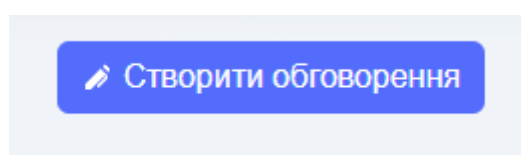


Рисунок 2.13 - Компонент кнопки

|      |      |                 |        |      |                                             |      |
|------|------|-----------------|--------|------|---------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка. 25.121.21.000 - ПЗ | Арк. |
|      |      | Чижомотря О.В.  |        |      |                                             | 44   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                             |      |

Вигляд компонента коментаря (рисунок 2.14):

Рисунок 2.14 - Компонент коментаря

Далі був доданий Store (Redux) для зв'язку бекенда і фронтенда

Redux — це контейнер стану для JS-застосунків, який забезпечує передбачуваність та централізованість керування даними в межах клієнтської частини проєкту [35]. Його використання дозволяє зручно зберігати, змінювати й відслідковувати стан застосунку в єдиному джерелі правди. Для платформи персоналізованих списків перегляду та обговорення аніме це особливо важливо, оскільки численні компоненти інтерфейсу мають доступ до спільних даних: інформації про користувача, списків перегляду, обговорень, повідомлень тощо. Redux дозволяє уніфікувати логіку керування станом, зробити її прозорою й такою, що легко тестується, а також значно спростити обробку складних сценаріїв взаємодії між компонентами. Це підвищує надійність і передбачуваність роботи веб-платформи, роблячи її зручною як для кінцевих користувачів, так і для команди розробки при супроводі й розвитку проєкту.

Ось ключові концепції та компоненти Redux:

Рисунок 2.15 містить файлову структуру стора, що відображає організацію стану додатку:

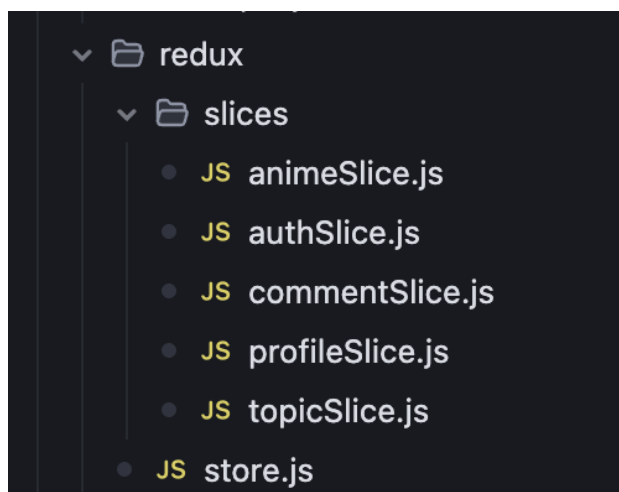


Рисунок 2.15 - Структура стора

|      |      |                 |        |      |                                             |      |
|------|------|-----------------|--------|------|---------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка. 25.121.21.000 - ПЗ | Арк. |
|      |      | Чижмотря О.В.   |        |      |                                             | 45   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                             |      |

В результаті було отримано UI. Для забезпечення зручності використання платформи було розроблено сучасний вебінтерфейс, що дозволяє користувачам легко взаємодіяти з основними функціональними можливостями системи. Інтерфейс створено з урахуванням принципів адаптивного дизайну та орієнтації на різні категорії користувачів.

На рисунку 2.16 представлено UI сторінку логіна, де реалізовано форму для введення електронної пошти та пароля.

Рисунок 2.16 - UI сторінка логіна

Рисунок 2.17 демонструє інтерфейс сторінки реєстрації нового користувача із заповненням основних персональних даних, таких як ім'я, прізвище, нікнейм та електронна пошта з паролем.

Рисунок 2.17 - Реєстрація

|      |      |                 |        |      |                                            |      |
|------|------|-----------------|--------|------|--------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка.25.121.21.000 - ПЗ | Арк. |
|      |      | Чижомотря О.В.  |        |      |                                            | 46   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                            |      |

На рисунку 2.18 зображено UI сторінки відновлення пароля, де користувач має можливість ввести адресу електронної пошти для отримання посилання на зміну пароля.

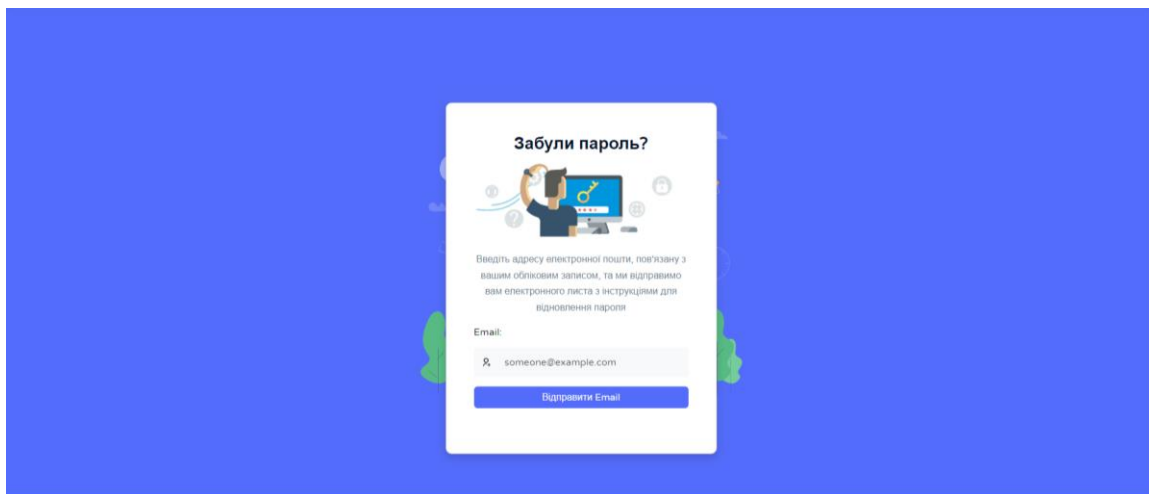


Рисунок 2.18 - UI сторінки відновлення паролю

Рисунок 2.19 ілюструє вигляд UI сторінки профілю користувача, яка містить інформацію про обліковий запис, список переглянутих аніме та інші персоналізовані дані.

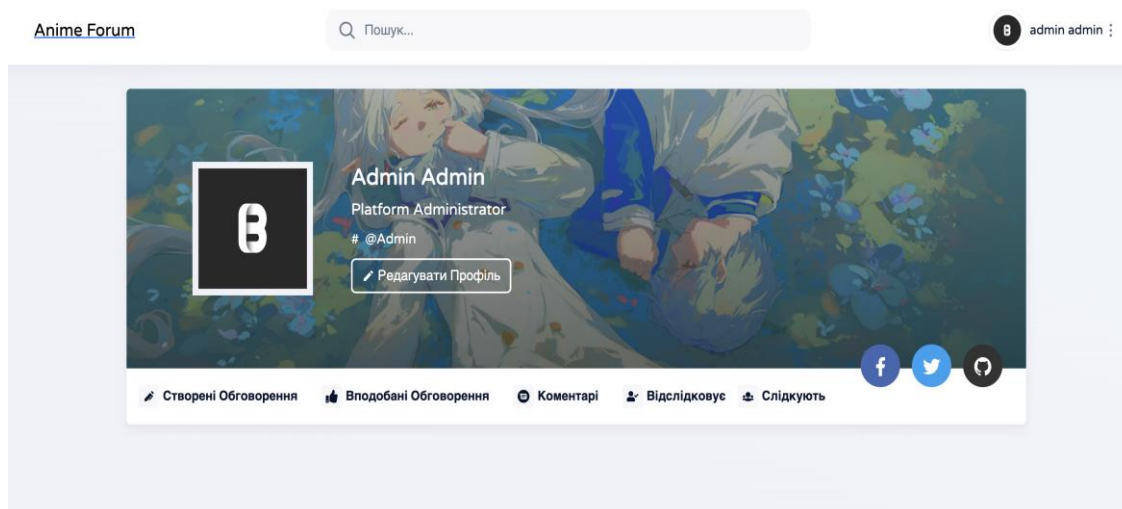


Рисунок 2.19 - UI сторінки профілю користувача

На рисунку 2.20 показано UI сторінку обговорень, де користувачі можуть переглядати наявні теми для дискусій, залишати коментарі та оцінювати думки інших учасників.

|      |      |                 |        |      |                                            |      |
|------|------|-----------------|--------|------|--------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка.25.121.21.000 - ПЗ | Арк. |
|      |      | Чижмотря О.В.   |        |      |                                            | 47   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                            |      |

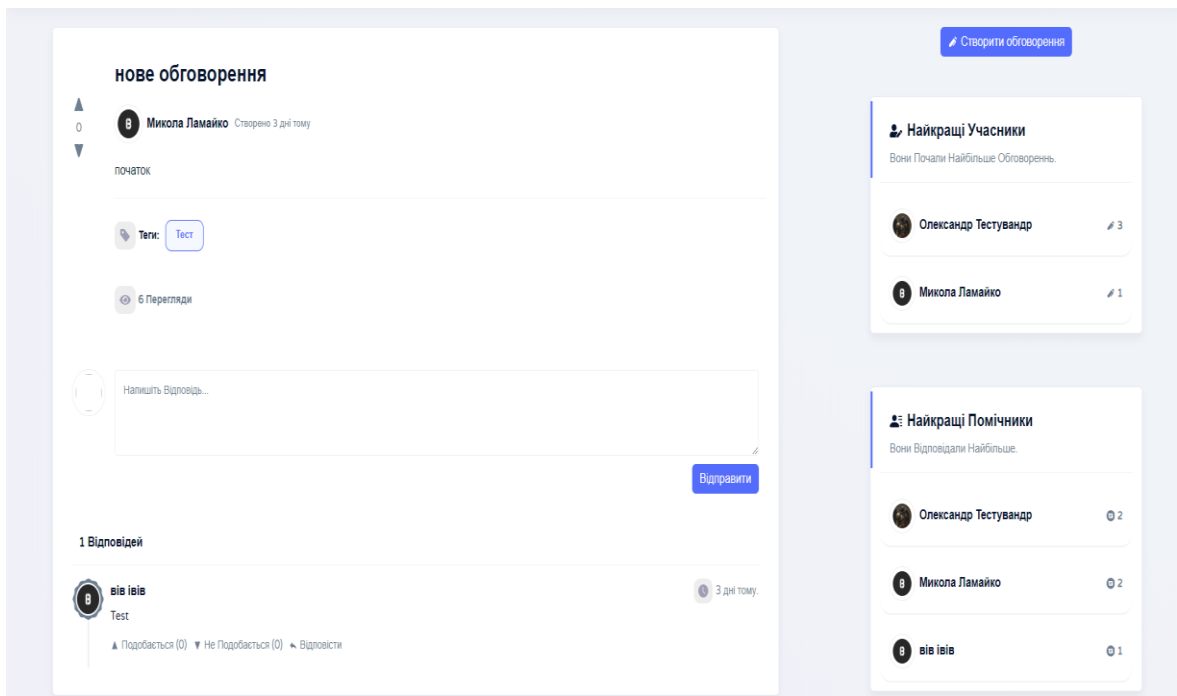


Рисунок 2.20 - UI сторінки обговорень

Рисунок 2.21 демонструє інтерфейс сторінки створення нового обговорення, що передбачає заповнення назви теми, опису та вибір пов'язаних тегів або простору.

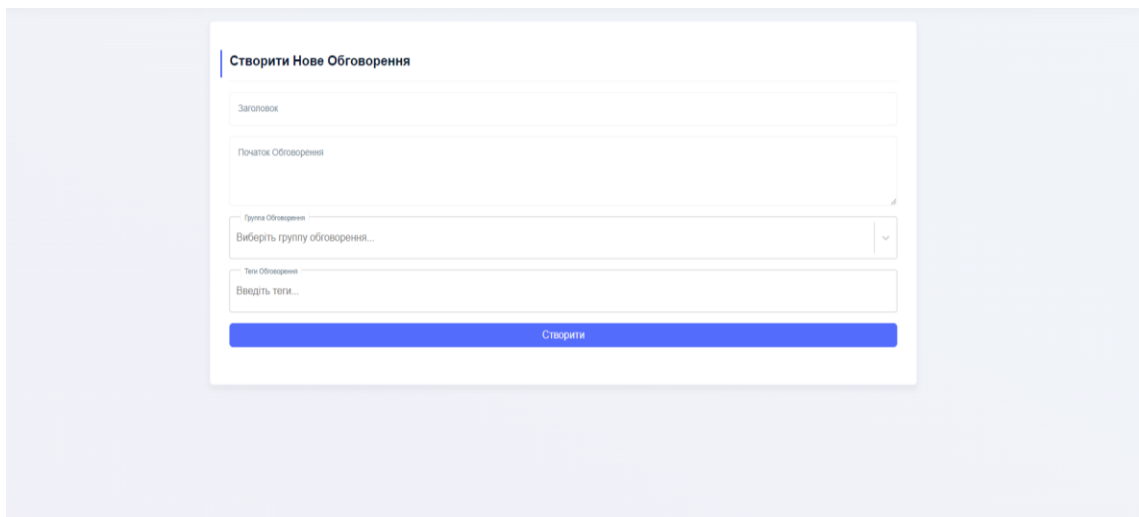


Рисунок 2.21 - UI сторінки створення обговорення

Рисунки 2.22 та 2.23 відображають UI сторінки редагування профілю користувача, де представлено дві частини форми: на першій частині реалізовано зміну основних персональних даних, на другій – зміну пароля та зовнішніх посилань.

|      |      |                 |        |      |                                            |      |
|------|------|-----------------|--------|------|--------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка.25.121.21.000 - ПЗ | Арк. |
|      |      | Чижомотря О.В.  |        |      |                                            | 48   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                            |      |

Інформація

Ім'я

Прізвище

Нікнейм

Email

Про Мене

Біографія

Зовнішні Посилання

Facebook URL

Рисунок 2.22 - UI сторінки редагування профіля (частина 1)

Зовнішні Посилання

Facebook URL

X URL

Github URL

Безпека

Поточний Пароль

Новий Пароль

Підтвердьте Пароль

Зберегти

Рисунок 2.23 - UI сторінки редагування профіля (частина 2)

|      |      |                 |        |      |                                            |      |
|------|------|-----------------|--------|------|--------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка.25.121.21.000 - ПЗ | Арк. |
|      |      | Чижмотря О.В.   |        |      |                                            | 49   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                            |      |

Після закінчення Фронт-енд частини було створено Бек-енд

Розробка бекенду для веб-платформи для ведення персоналізованих списків перегляду та обговорення аніме базувалася на принципах модульності, масштабованості та простоти супроводу. Архітектура програмного продукту реалізована за моделлю MVC (Model-View-Controller), де кожен елемент має чітко визначену роль у загальній структурі системи. Це забезпечує легкість додавання нових функцій, підвищує безпеку, а також спрощує налагодження й тестування.

У кореневій директорії проєкту розташовані основні папки, кожна з яких виконує свою роль у загальній архітектурі. У папці config зберігається файл db.js, що відповідає за налаштування з'єднання із базою даних MongoDB. Саме тут здійснюється підключення до бази даних через URI, які зберігаються у змінних середовища для підвищення безпеки системи. Налаштування підключення враховує можливість роботи як у режимі розробки, так і в режимі продуктивного використання.

Основну бізнес-логіку проєкту зосереджено у папці controllers. Кожен файл цієї папки відповідає за обробку запитів певного типу. Наприклад, animeController.js керує діями, пов'язаними із додаванням, редагуванням, переглядом та видаленням аніме зі списків користувачів. Файл authController.js реалізує процеси реєстрації, входу до системи та управління токенами доступу, забезпечуючи надійний захист даних користувачів. Коментарі до обговорень обробляються за допомогою commentController.js, де реалізовано створення, редагування й видалення коментарів. Логіка управління темами для дискусій зосереджена у topicController.js, а всі операції з профілями користувачів виконує userController.js. Завдяки такій побудові контролерів код залишається структурованим, легко зрозумілим і придатним для розширення в майбутньому.

Важливою частиною системи є проміжне програмне забезпечення, яке знаходиться у папці middlewares. Ці модулі відповідають за обробку запитів до того, як вони потрапляють у контролери. Тут реалізовано генерацію токенів для верифікації електронної пошти та скидання пароля, що підвищує безпеку

|      |      |                 |        |      |                                            |      |
|------|------|-----------------|--------|------|--------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка.25.121.21.000 - ПЗ | Арк. |
|      |      | Чижемотря О.В.  |        |      |                                            | 50   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                            |      |

платформи. Також middleware містить механізми валідації токенів доступу, що дозволяє ефективно керувати правами користувачів і запобігати несанкціонованому доступу. Використання проміжного програмного забезпечення дозволяє реалізувати додаткові рівні захисту та забезпечує гнучкість у налаштуванні правил доступу.

Приклад middleware для генерації токenu:

```
const generateAccessToken = (user) => {
  email = user.email;
  username = user.username;
  try {
    const accessToken = jwt.sign(
      { email, username },
      process.env.ACCESS_TOKEN_SECRET_KEY,
      {
        expiresIn: process.env.ACCESS_TOKEN_EXPIRATION,
      }
    );
    return accessToken;
  } catch (err) {
    console.log(err);
  }
};

const generateRefreshToken = (user, expiration) => {
  email = user.email;
  username = user.username;
  try {
    const refreshToken = jwt.sign(
      { email, username },
      process.env.REFRESH_TOKEN_SECRET_KEY,
      {
        expiresIn: expiration,
      }
    );
    return refreshToken;
  } catch (err) {
    console.log(err);
  }
};
```

У папці models містяться визначення всіх моделей бази даних, які створені з використанням бібліотеки Mongoose для роботи з MongoDB. Наприклад, модель animeModel.js описує структуру даних для аніме, що зберігається в базі, зокрема назви, жанри, рік випуску та інші атрибути. commentModel.js визначає формат збереження коментарів, що дозволяє ефективно зв'язувати їх із відповідними обговореннями та користувачами. У моделі userModel.js зосереджено структуру даних профілів користувачів, де зберігається інформація

|      |      |                 |        |      |                                            |      |
|------|------|-----------------|--------|------|--------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка.25.121.21.000 - ПЗ | Арк. |
|      |      | Чижемотря О.В.  |        |      |                                            | 51   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                            |      |

для автентифікації та персоналізації. Окрім того, у цій директорії містяться моделі для тем обговорень, тегів і просторів для спільнот, що дозволяє гнучко організовувати дискусії на платформі.

Приклад однієї з моделей:

```
const CommentSchema = new mongoose.Schema(
  {
    owner: String,
    content: String,
    parentTopic: {
      type: mongoose.Types.ObjectId,
      ref: "Topic",
    },
    parentComment: {
      type: mongoose.Types.ObjectId,
      ref: "Comment",
    },
    upvotes: [
      {
        type: String,
        ref: "User",
      },
    ],
    downvotes: [
      {
        type: String,
        ref: "User",
      },
    ],
  },
);
```

Маршрутизація запитів реалізована у папці routes, де кожен файл відповідає за обробку певної категорії запитів. animeRoutes.js обробляє всі запити, пов'язані з роботою із даними про аніме, authRoutes.js — запити автентифікації, а файли commentRoutes.js, topicRoutes.js та userRoutes.js відповідають за обробку запитів коментарів, тем і користувачів відповідно. Завдяки цьому архітектура залишається логічно поділеною і зручною для масштабування.

Приклад маршрутизації на основі animeRoutes:

```
router.get("/", animeController.getAllAnime);
router.post("/", validateAccessToken, animeController.addAnime);
router.put("/:id", validateAccessToken, animeController.updateAnime);
router.delete("/:id", validateAccessToken, animeController.deleteAnime);
router.put("/watch/:animeId", validateAccessToken, animeController.markAsWatched);
```

|      |      |                 |        |      |                                             |      |
|------|------|-----------------|--------|------|---------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка. 25.121.21.000 - ПЗ | Арк. |
|      |      | Чижемотря О.В.  |        |      |                                             | 52   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                             |      |

Розробка платформи передбачала поступове впровадження компонентів (рисунок 2.24). Спершу було спроектовано структуру бази даних та реалізовано базові моделі для користувачів, аніме та коментарів. Наступним кроком стало створення контролерів та роутів, що забезпечують можливість обробки запитів і взаємодії з даними. Значну увагу під час розробки приділено безпеці — впроваджено механізми автентифікації, авторизації, валідації даних та захисту від типових вразливостей. У процесі тестування використовувались інструменти для перевірки API, зокрема Postman, що дозволило перевірити коректність роботи всіх ендпоінтів.

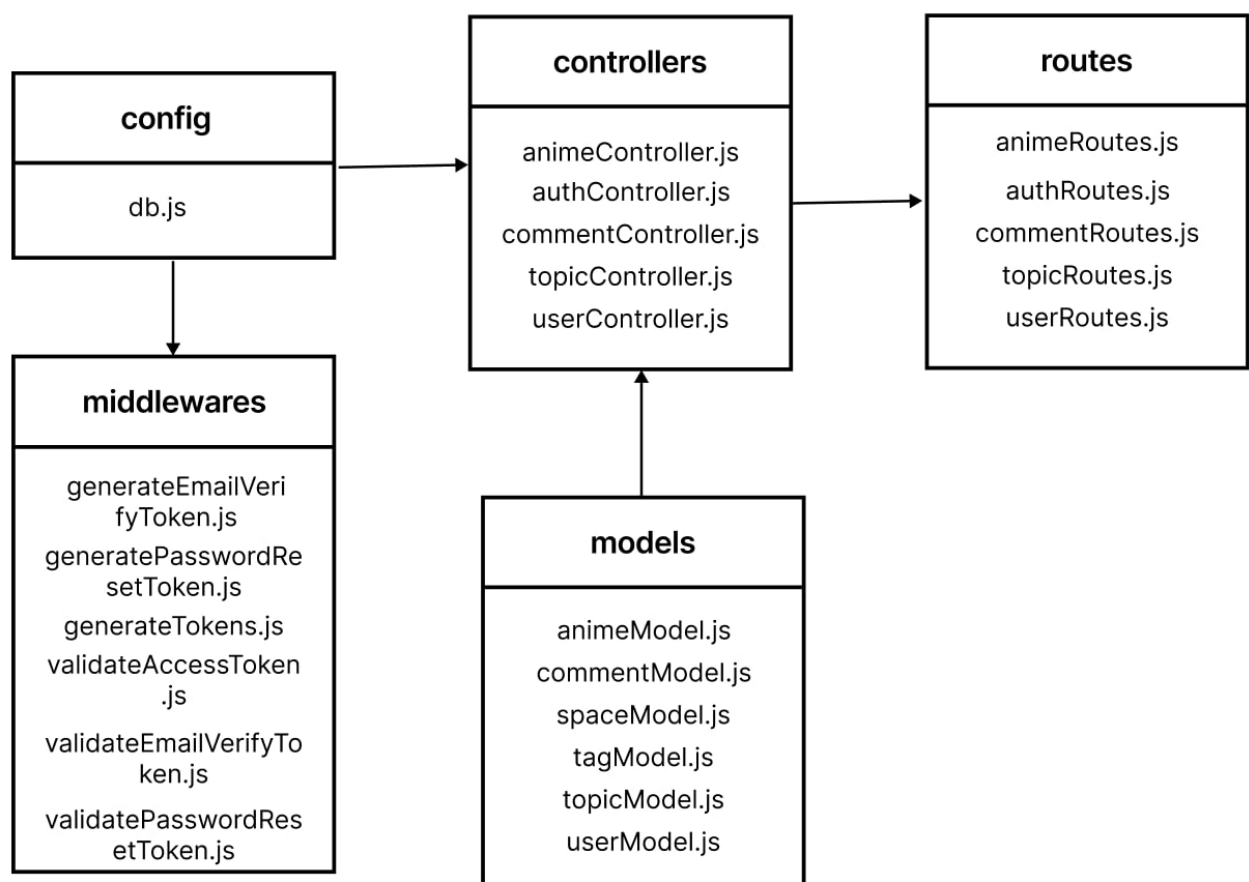


Рисунок 2.24 - UML-схема

Використана структура проекту забезпечує прозорість коду, дозволяє ефективно взаємодіяти з даними та реалізувати сучасні принципи RESTful-архітектури. Вона також дає змогу в майбутньому легко інтегрувати додаткові модулі, наприклад сервіси для аналітики, системи оповіщень або платіжні

шлюзи. Такий підхід дає змогу розвивати проєкт поступово, не порушуючи цілісності архітектури.

#### Висновки до другого розділу

У другому розділі було детально розглянуто етапи проєктування та побудови архітектури платформи для ведення персоналізованих списків перегляду та обговорення аніме з використанням сучасних інструментів і технологій. У результаті аналізу вимог було сформовано логічну структуру системи, яка забезпечує чітку взаємодію між клієнтською частиною, сервером і базою даних, а також дозволяє легко масштабувати платформу в майбутньому.

Особливу увагу приділено розробці моделі бази даних, яка оптимізована для зберігання інформації про користувачів, аніме, коментарі, теги, обговорення й історію взаємодії в системі. Проєктування структури даних і її візуалізація у вигляді діаграм дозволили чітко визначити основні сутності платформи та відобразити їх взаємозв'язки для забезпечення цілісності даних та ефективної роботи запитів.

У цьому розділі описано ключові програмні компоненти, алгоритми обробки запитів користувачів, принципи побудови системи коментарів, голосування та ведення списків перегляду, а також підходи до реалізації безпеки й захисту даних користувачів. Детально розглянуто процеси додавання аніме до списків, організації обговорень у спільнотах, управління профілями користувачів і інтеграції із зовнішніми сервісами (наприклад, для авторизації через соціальні мережі).

Впровадження обраного технологічного стеку та сучасних підходів до розробки дозволило реалізувати ефективну, масштабовану й безпечну платформу, яка відповідає заявленим вимогам і забезпечує зручність використання та активну взаємодію між учасниками спільноти.

|      |      |                 |        |      |                                            |      |
|------|------|-----------------|--------|------|--------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка.25.121.21.000 - ПЗ | Арк. |
|      |      | Чижемотря О.В.  |        |      |                                            | 54   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                            |      |

## РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ПЛАТФОРМОЮ ДЛЯ ВЕДЕННЯ ПЕРСОНАЛІЗОВАНИХ СПИСКІВ ПЕРЕГЛЯДУ ТА ОБГОВОРЕННЯ АНІМЕ

### 3.1 Порядок встановлення та налаштування параметрів додатку

Платформа для ведення персоналізованих списків перегляду та обговорення аніме побудована із кількох основних компонентів, кожен з яких виконує окрему функцію й потребує індивідуального налаштування та розгортання. Для зручності візуалізації архітектури системи та демонстрації процесу її впровадження була створена відповідна діаграма розгортання. Такий підхід дозволяє наочно відобразити взаємозв'язки між складовими платформи, забезпечити можливість масштабування та спростити підтримку й розвиток проекту.

На діаграмі розгортання (рисунок 3.1) зображено архітектуру платформи для обговорення та перегляду аніме, де ключову роль відіграє взаємодія клієнтської частини з серверною частиною та базою даних. Сервер бази даних реалізовано на основі MongoDB, що відповідає за зберігання інформації про користувачів, аніме, списки перегляду, коментарі, обговорення та інші дані. Передача даних між сервером бази даних та основним сервером здійснюється через захищений протокол, що гарантує безпечність і цілісність даних.

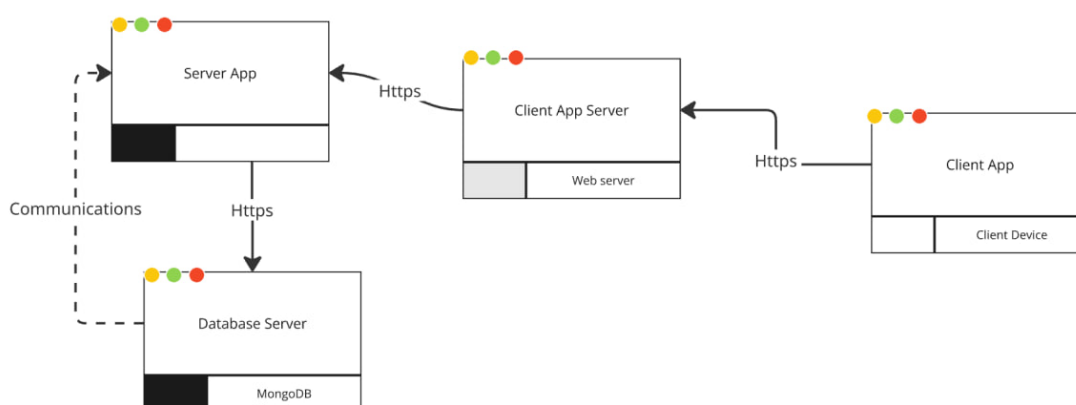


Рисунок 3.1 - Діаграма розгортання

|      |      |                 |        |      |                                            |      |
|------|------|-----------------|--------|------|--------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка.25.121.21.000 - ПЗ | Арк. |
|      |      | Чижомотря О.В.  |        |      |                                            | 55   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                            |      |

Основна логіка роботи платформи реалізована на вебсервері. Тут розгорнуто кілька основних компонентів: бекенд застосунку, що відповідає за обробку запитів, керування даними, авторизацію й автентифікацію користувачів, а також клієнтський застосунок, який забезпечує інтерфейс взаємодії користувача з платформою. Компоненти взаємодіють між собою через захищені API та протокол HTTPS для забезпечення безпеки передачі даних.

Користувачі взаємодіють із платформою через браузер або мобільний додаток, підключаючись до клієнтської частини через HTTPS для захисту персональних даних і забезпечення комфортної роботи.

Платформа використовує Node.js як основу серверної частини, з реалізацією основної бізнес-логіки на Express.js. Клієнтська частина реалізована на React.js. Для налаштування системи потрібно встановити MongoDB і налаштувати підключення до бази даних, розгорнути бекенд і клієнтську частину на сервері. Також налаштовуються доменне ім'я й HTTPS для безпечної взаємодії між модулями.

Усі важливі параметри середовища зберігаються у файлах конфігурації .env. Для клієнтської частини вони включають адресу бекенду та інші сервісні налаштування. Для серверної частини — параметри підключення до бази даних, ключі для автентифікації користувачів, токени для роботи зі сховищами та налаштування поштового сервера.

Файл .env для клієнта:

REACT\_APP\_BACKEND\_URL — це змінна середовища, яка містить адресу серверної частини (бекенду) вашої платформи. Вона використовується клієнтським застосунком (React) для того, щоб правильно надсилати запити API до сервера. Наприклад, якщо бекенд працює за адресою <https://api.myanimeapp.com>, то саме це значення буде вказано в REACT\_APP\_BACKEND\_URL. Це дозволяє React-додатку легко взаємодіяти з сервером, незалежно від середовища (локальна розробка, staging чи production).

Завдяки такій змінній легко перемикає середовище, не змінюючи код: достатньо вказати іншу адресу в .env файлі.

|      |      |                 |        |      |                                             |      |
|------|------|-----------------|--------|------|---------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка. 25.121.21.000 - ПЗ | Арк. |
|      |      | Чижемотря О.В.  |        |      |                                             | 56   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                             |      |

Файл .env для бекенду:

PORT — це порт, на якому запускається серверна частина застосунку. Наприклад, якщо встановити PORT=3000, сервер буде слухати підключення на цьому порту.

REACT\_APP\_URL — адреса клієнтської частини платформи, яка використовується для формування посилань у запрошеннях, листах підтвердження електронної пошти тощо. Це URL, за яким користувачі звертаються до інтерфейсу.

MONGODB\_URL\_CONNECTION — рядок підключення до бази даних MongoDB. Він містить адресу сервера бази даних, а також, за потреби, ім'я користувача і пароль для доступу. Використовується для зберігання всієї інформації про користувачів, списки аніме, коментарі, тощо.

ACCESS\_TOKEN\_SECRET\_KEY — секретний ключ для підписування та перевірки JWT access tokenів, які видаються користувачам при вході в систему для підтвердження їх автентичності при кожному запиті.

REFRESH\_TOKEN\_SECRET\_KEY — секретний ключ для створення й перевірки refresh tokenів. Ці токени використовуються для оновлення access tokenів без необхідності повторного входу користувача.

EMAIL\_VERIFY\_TOKEN\_SECRET\_KEY — секретний ключ, що застосовується для генерації tokenів підтвердження електронної пошти під час реєстрації чи зміни адреси користувача.

RESET\_PASSWORD\_TOKEN\_SECRET\_KEY — секретний ключ для створення tokenів скидання пароля, які відправляються користувачеві для відновлення доступу до облікового запису.

ACCESS\_TOKEN\_EXPIRATION — час життя access токена. Після закінчення цього часу користувачеві потрібно буде отримати новий токен через refresh токен.

REFRESH\_TOKEN\_EXPIRATION — тривалість життя refresh токена. Після закінчення його дії користувачеві доведеться повторно увійти в систему.

|      |      |                 |        |      |                                             |      |
|------|------|-----------------|--------|------|---------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка. 25.121.21.000 - ПЗ | Арк. |
|      |      | Чижемотря О.В.  |        |      |                                             | 57   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                             |      |

EMAIL\_VERIFY\_TOKEN\_EXPIRATION — час, протягом якого дійсний токен для підтвердження електронної пошти. Використовується для безпеки, щоб не дозволяти безстрокове підтвердження.

RESET\_PASSWORD\_TOKEN\_EXPIRATION — тривалість життя токена для скидання пароля. Він обмежує час, протягом якого користувач може скинути пароль після запиту.

CLOUDINARY\_NAME — ім'я облікового запису в Cloudinary, яке використовується для зберігання та обробки завантажених зображень (аватари, обкладинки аніме тощо).

CLOUDINARY\_API\_KEY — API-ключ для доступу до сервісу Cloudinary. Використовується для аутентифікації запитів до сервісу під час завантаження або видалення файлів.

CLOUDINARY\_API\_SECRET — секретний ключ для забезпечення захищеної роботи з Cloudinary API, використовується разом із API-ключем для підписування запитів.

SMTP\_SERVER — адреса поштового сервера, через який надсилаються службові листи (наприклад, підтвердження реєстрації, скидання пароля).

SMTP\_PORT — порт поштового сервера для надсилання повідомлень (зазвичай 465 або 587 залежно від конфігурації безпеки).

SMTP\_USERNAME — ім'я користувача (зазвичай це e-mail), яке використовується для авторизації на поштовому сервері.

SMTP\_PASSWORD — пароль до облікового запису поштового сервера, потрібний для аутентифікації при надсиланні листів.

### 3.2. Структура інтерфейсу та порядок взаємодії з платформою для ведення персоналізованих списків перегляду та обговорення аніме

Після переходу на платформу користувач одразу потрапляє на головну сторінку, яка є стартовою точкою взаємодії з сервісом для ведення персоналізованих списків перегляду та обговорення аніме. У верхній частині сторінки розташована навігаційна панель з назвою платформи, посиланнями на

|      |      |                 |        |      |                                             |      |
|------|------|-----------------|--------|------|---------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка. 25.121.21.000 - ПЗ | Арк. |
|      |      | Чижемотря О.В.  |        |      |                                             | 58   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                             |      |

основні розділи (наприклад, «Головна», «Обговорення», «Список аніме», «Мій список перегляду») та кнопками для входу або реєстрації. Інтерфейс платформи розроблений таким чином, щоб навіть неавторизовані користувачі могли ознайомитися зі списком обговорень та аніме, а також переглянути активність учасників спільноти. Детальніше з ним можна ознайомитись на рисунках 3.2 та 3.3.

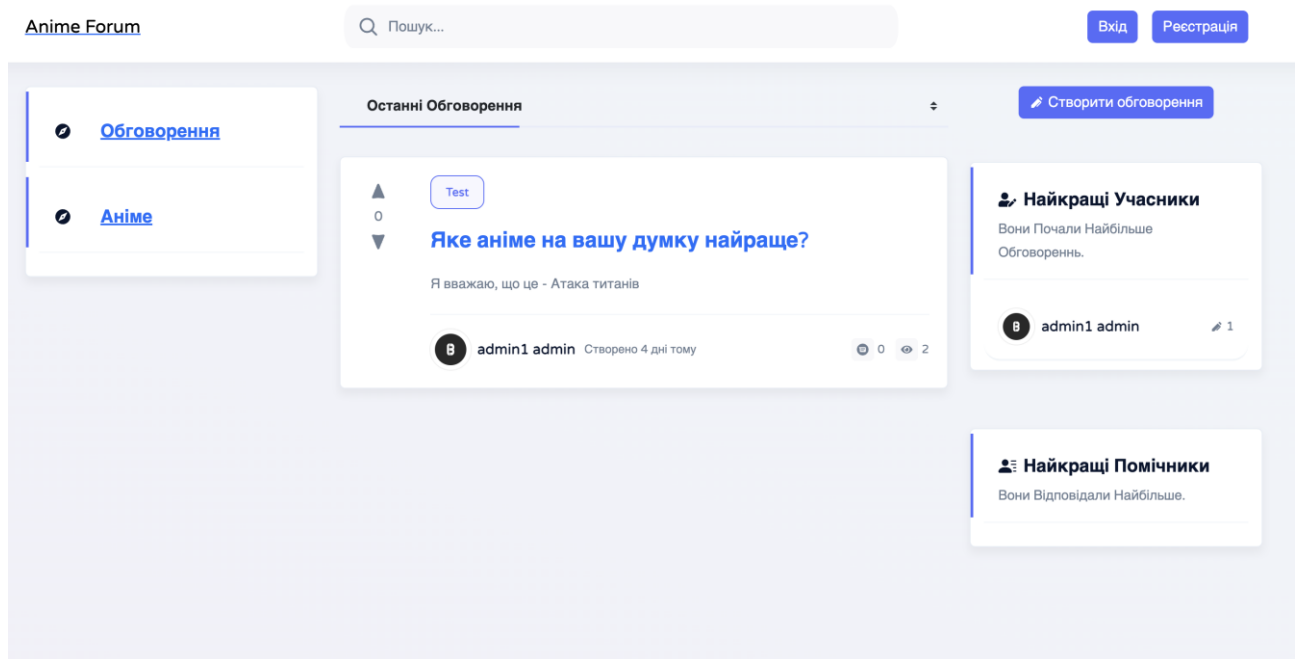


Рисунок 3.2 - Головна сторінка

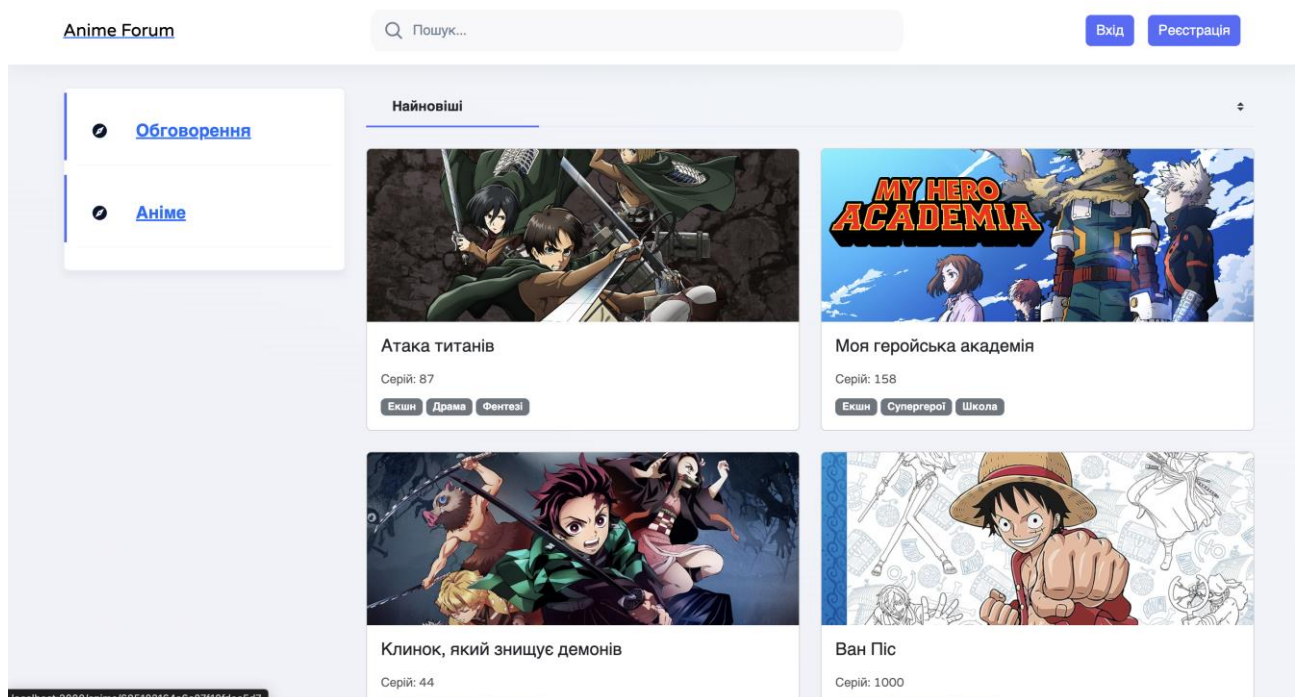


Рисунок 3.3 - Список аніме

|      |      |                 |        |      |                                            |      |
|------|------|-----------------|--------|------|--------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка.25.121.21.000 - ПЗ | Арк. |
|      |      | Чижмотря О.В.   |        |      |                                            | 59   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                            |      |

На головній сторінці користувач бачить список обговорень, які відображаються у вигляді карток із заголовками тем, короткими описами та інформацією про кількість коментарів. Поруч доступна вкладка, де представлений список аніме, що дозволяє швидко знайти потрібну назву та переглянути її основну інформацію. Ліва колонка містить блоки «Найкращі учасники» — користувачі, які почали найбільше обговорень, та «Найкращі помічники» — ті, хто залишив найбільше відповідей у коментарях. Ця інформація оновлюється динамічно на основі активності користувачів платформи, що стимулює їх участь в житті спільноти.

На рисунку 3.4 представлено фільтр обговорень, який дозволяє користувачам зручно здійснювати пошук та сортування тем за різними критеріями, такими як дата створення, кількість коментарів, популярність. Реалізація фільтрації сприяє підвищенню зручності навігації платформою та ефективнішому пошуку потрібних обговорень.

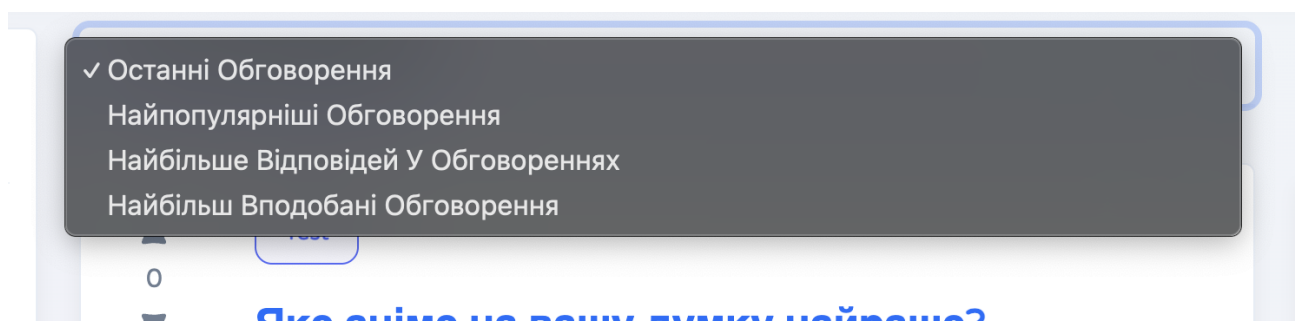


Рисунок 3.4 - Фільтр обговорень

На рисунку 3.5 зображено фільтр аніме, який реалізовано для спрощення пошуку та відбору аніме за заданими параметрами. Такий інструмент підвищує ефективність користувацької взаємодії з платформою та забезпечує швидкий доступ до релевантного контенту.

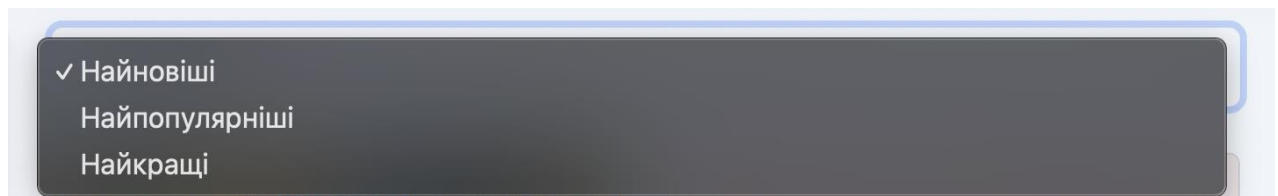


Рисунок 3.5 - Фільтр аніме

При натисканні на конкретне обговорення користувач переходить на сторінку самого обговорення (рисунки 3.6 та 3.7), де він може ознайомитися з деталями теми, прочитати наявні коментарі, залишити свій коментар чи лайкнути вже існуючий. Можливість коментувати та ставити оцінки коментарям доступна лише авторизованим користувачам, що сприяє безпеці та якості обговорень. Якщо користувач не залогінений, система запропонує йому увійти або створити обліковий запис перед публікацією коментаря чи взаємодією з контентом.

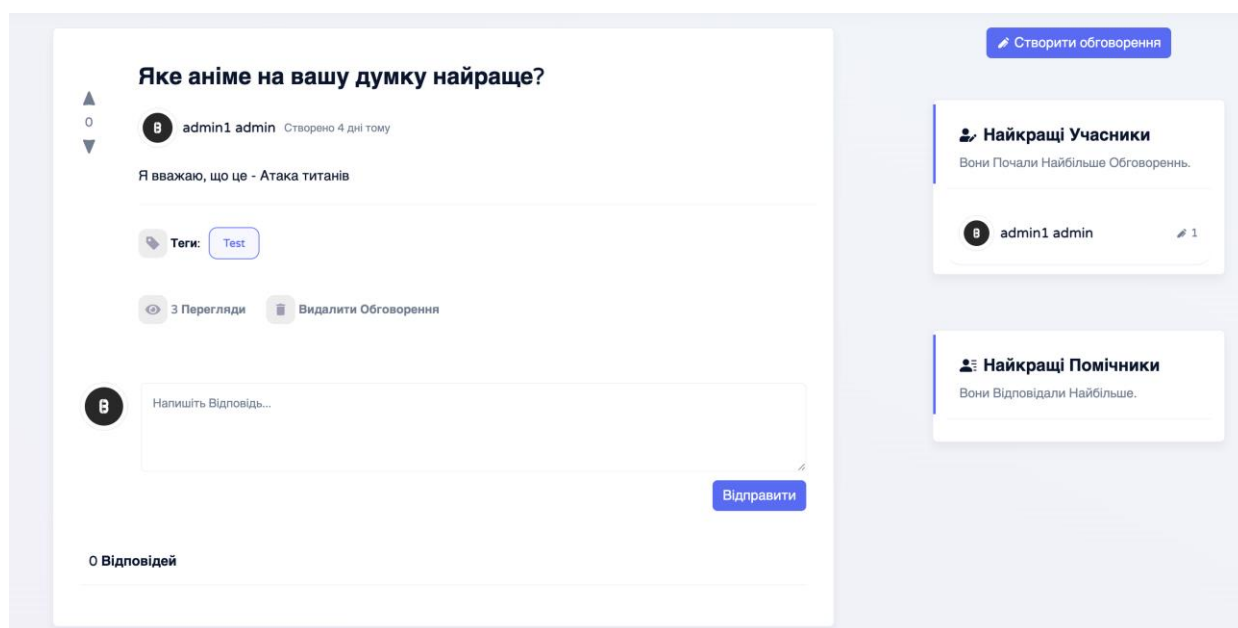


Рисунок 3.6 - Сторінка обговорення

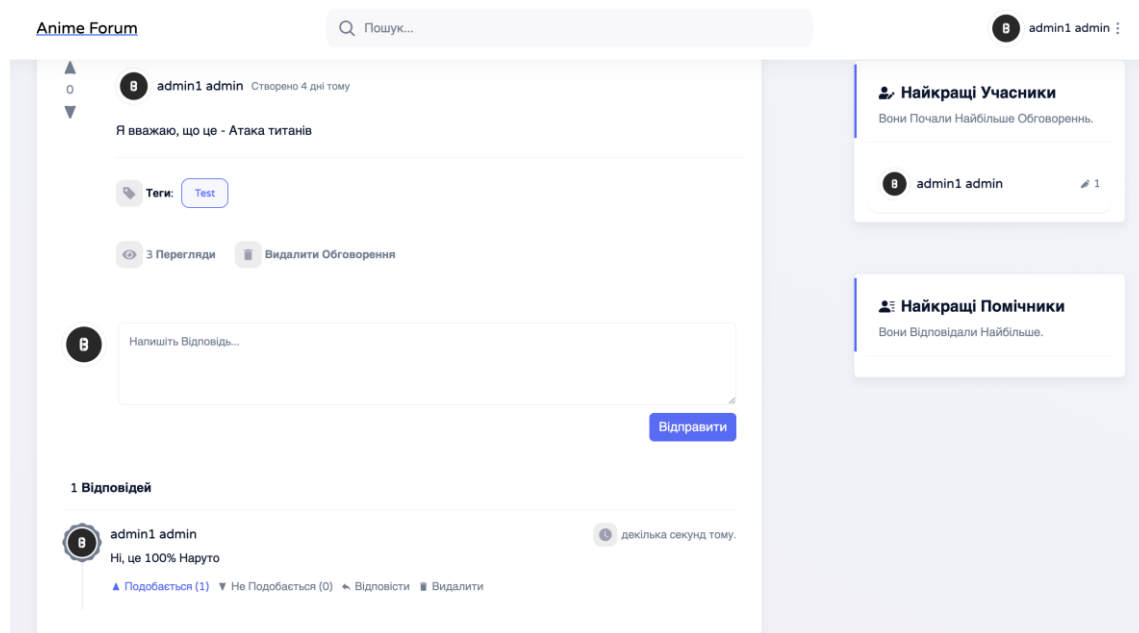


Рисунок 3.7 - Сторінка обговорення з відповідями

|      |      |                 |        |      |                                            |      |
|------|------|-----------------|--------|------|--------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка.25.121.21.000 - ПЗ | Арк. |
|      |      | Чижморя О.В.    |        |      |                                            | 61   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                            |      |

Процес реєстрації передбачає заповнення форми (рисунок 3.8) з електронною адресою та паролем. Після реєстрації на вказану пошту надходить лист із підтвердженням для активації облікового запису. Також при зміні пароля система надсилає відповідне повідомлення електронною поштою для додаткового захисту даних користувача.

Також представлена сторінка авторизація користувача (рисунок 3.9).

**Реєстрація**  
Ласкаво просимо!

Ім'я:  
Олександр

Прізвище:  
Лавріщев

Нікнейм:  
Нікнейм

E-mail:  
someone@example.com

Пароль:  
\*\*\*\*\*

[Зареєструватися](#)

Рисунок 3.8 - Сторінка реєстрації нового користувача

**Вхід**  
Ласкаво просимо!

Нікнейм або Email:  
someone@example.com

Ваш пароль:  
\*\*\*\*\*

☐ Запам'ятати пароль [Забули пароль?](#)

[Увійти](#)

Рисунок 3.9 - Сторінка авторизація користувача

|      |      |                 |        |      |                                            |      |
|------|------|-----------------|--------|------|--------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка.25.121.21.000 - ПЗ | Арк. |
|      |      | Чижомотря О.В.  |        |      |                                            | 62   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                            |      |

При переході на сторінку конкретного аніме користувач може переглянути його докладну інформацію: опис, жанри, студію, список головних персонажів, рейтинг та інші дані. Уі цієї сторінки показаний на рисунках 3.10 та 3.11 Авторизовані користувачі мають можливість додати це аніме до свого списку перегляду натисканням відповідної кнопки. Додане аніме зберігається в особистому профілі користувача, і він завжди може повернутися до нього з розділу «Мій список перегляду».

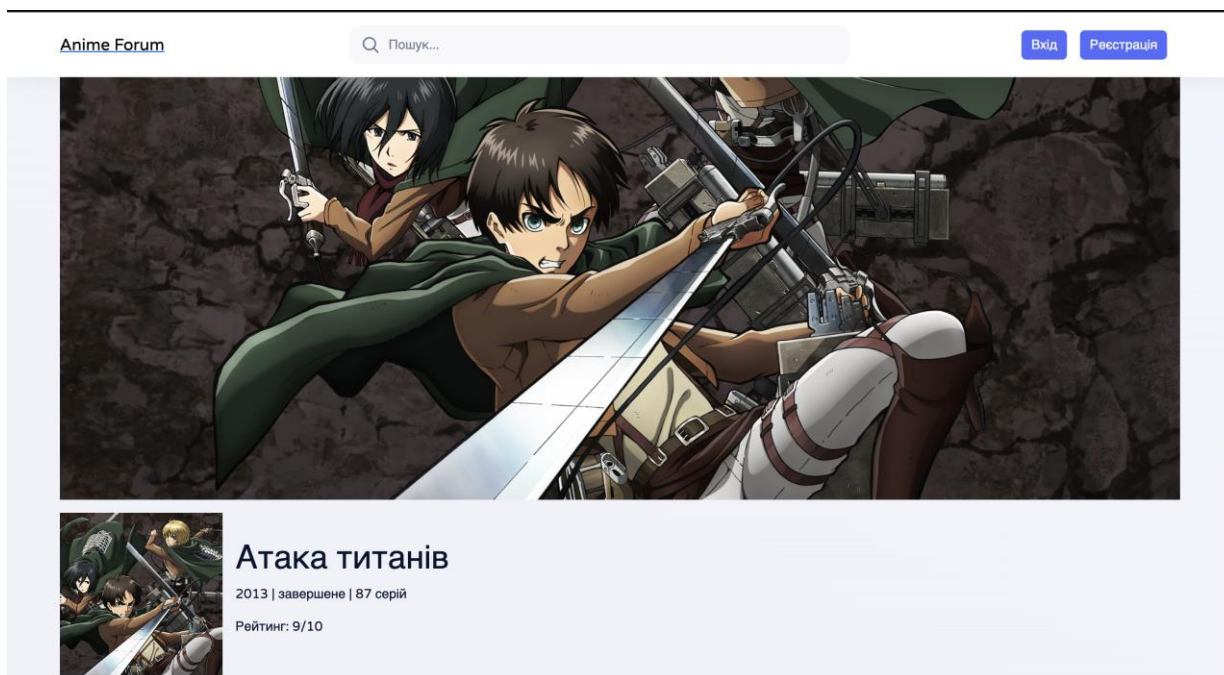


Рисунок 3.10 - Сторінка аніме

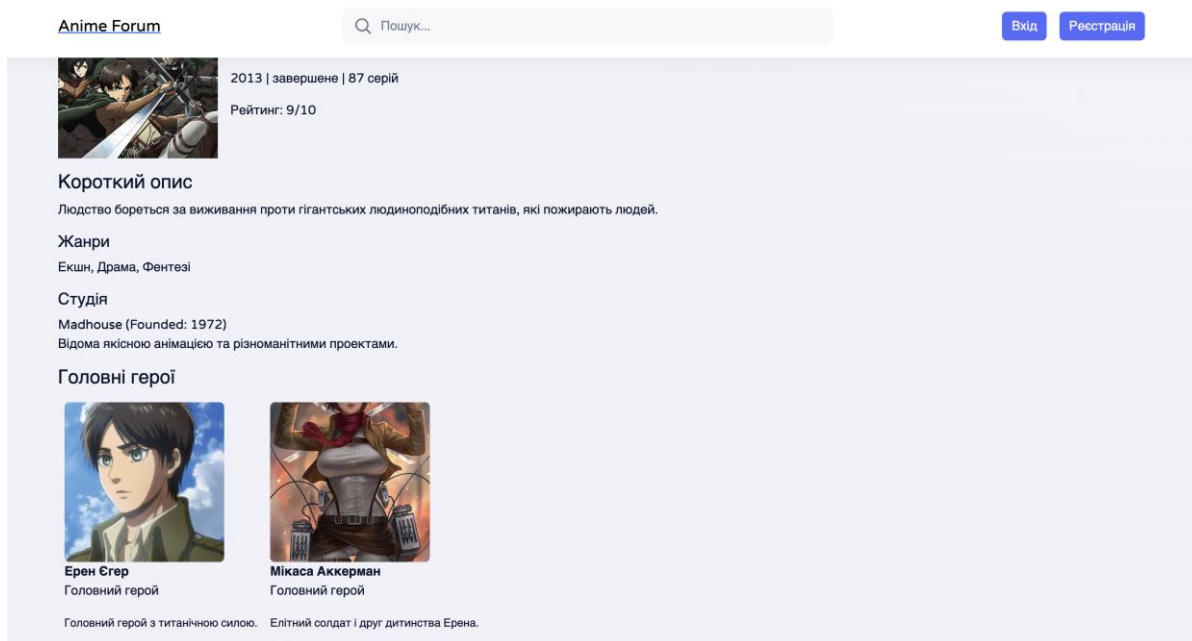


Рисунок 3.11 - Сторінка аніме

|      |      |                 |        |      |                                             |      |
|------|------|-----------------|--------|------|---------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка. 25.121.21.000 - ПЗ | Арк. |
|      |      | Чижемотря О.В.  |        |      |                                             | 63   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                             |      |

При вході на сторінку профілю користувач бачить свій аватар, банер, коротку біографію (якщо вона додана), кількість підписників і тих, на кого він підписаний. У верхній частині також розташовані кнопки для редагування профілю та налаштувань облікового запису. Більш детально можна побачити на рисунках 3.12 та 3.13.

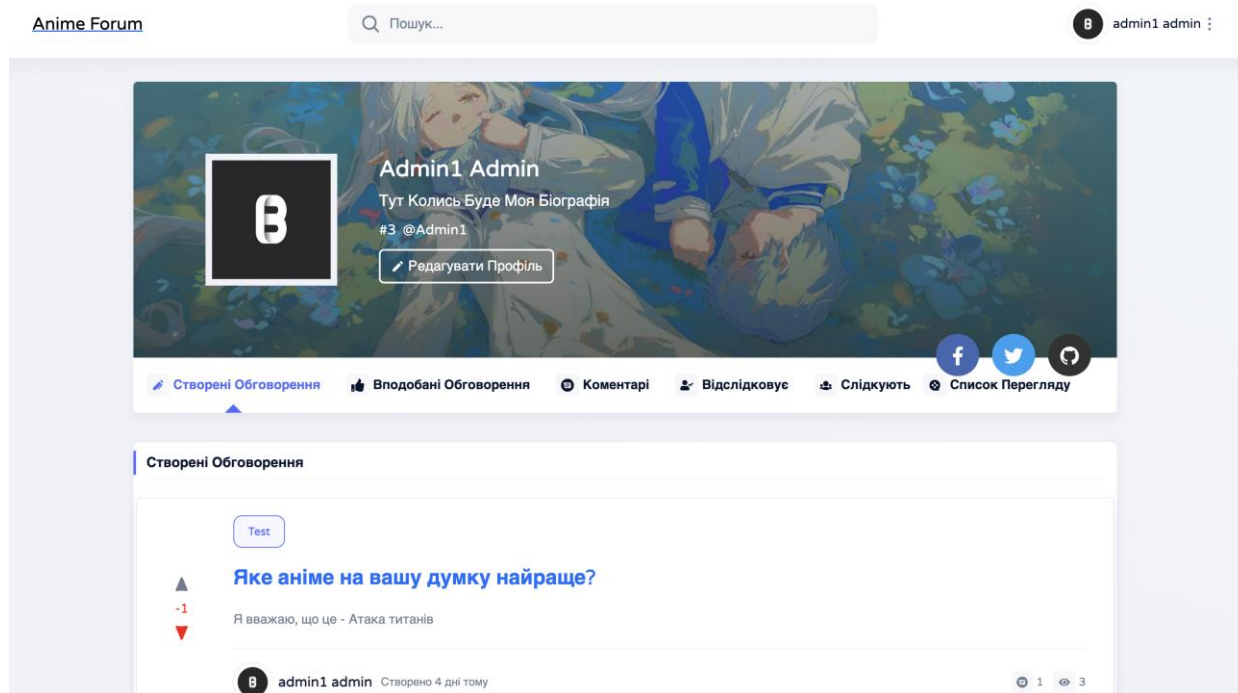


Рисунок 3.12 - Профіль

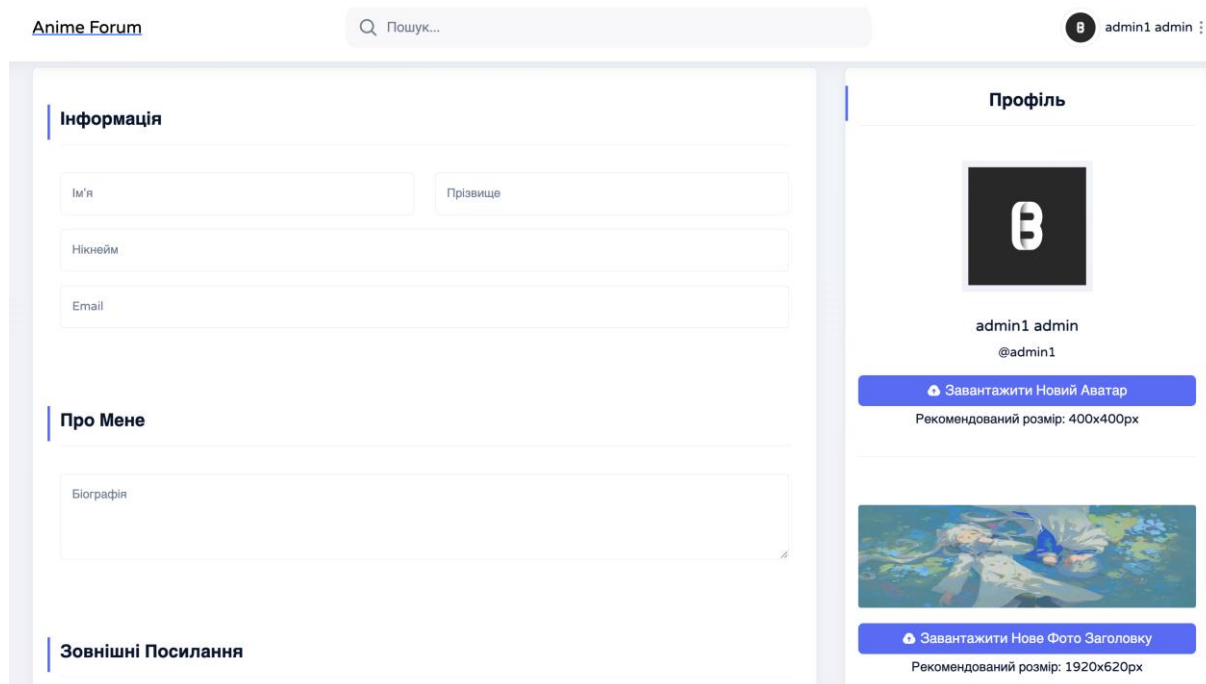


Рисунок 3.13 - Редагування профілю

|      |      |                 |        |      |                                            |      |
|------|------|-----------------|--------|------|--------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка.25.121.21.000 - ПЗ | Арк. |
|      |      | Чижмотря О.В.   |        |      |                                            | 64   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                            |      |

Сторінка профілю користувача на платформі для ведення персоналізованих списків перегляду та обговорення аніме є персональним простором кожного учасника спільноти. Вона призначена для зручного управління власною активністю, взаємодії з іншими користувачами та контролю за обраним контентом.

Основна частина профілю складається з кількох вкладок, кожна з яких відображає певний тип активності користувача:

Створені обговорення. У цій вкладці відображаються всі обговорення, які були створені користувачем. Кожне обговорення представлено у вигляді картки з назвою теми, коротким описом і статистикою (кількість коментарів, лайків). Це дозволяє користувачеві швидко знаходити власні теми й відслідковувати їхню активність.

Вподобані обговорення. Тут зібрані всі обговорення, яким користувач поставив лайк. Це дає можливість зручно повертатися до тем, що сподобалися, та брати участь у їх подальшому обговоренні.

Коментарі. Ця вкладка містить список усіх коментарів, залишених користувачем на різних обговореннях. Кожен коментар супроводжується посиланням на обговорення, в якому він був залишений, що дозволяє швидко перейти до теми й продовжити дискусію.

Відслідковує. У цьому розділі перелічені користувачі, на яких підписаний власник профілю. Це дозволяє легко знаходити цікаві профілі та стежити за їх активністю на платформі (створені обговорення, коментарі, нові пости).

Слідкують. Тут відображаються користувачі, які підписалися на профіль власника. Це мотивує користувача залишатися активним на платформі, створювати нові обговорення та коментувати, адже він бачить зацікавленість інших учасників спільноти.

Список перегляду. Ця вкладка містить персоналізований список аніме, які користувач додав до свого списку для перегляду. Кожен запис супроводжується обкладинкою та назвою аніме. Це допомагає користувачеві структурувати свій контент і планувати перегляд аніме.

|      |      |                 |        |      |                                            |      |
|------|------|-----------------|--------|------|--------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка.25.121.21.000 - ПЗ | Арк. |
|      |      | Чижевотря О.В.  |        |      |                                            | 65   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                            |      |

Отже, платформа поєднує функціонал обговорення та персоналізованого списку аніме, створюючи єдиний простір для спілкування, обміну думками та збереження вподобаного контенту. Користувачі мають можливість брати активну участь в обговореннях, коментувати, оцінювати повідомлення інших учасників, а також вести власний список перегляду аніме, що підвищує зацікавленість і залученість. Такий підхід стимулює розвиток спільноти однодумців, формує атмосферу підтримки та взаємодопомоги, де кожен може не лише ділитися враженнями про переглянуті аніме, а й отримувати рекомендації, знаходити нових друзів і однодумців. Інтеграція цих можливостей в єдину платформу сприяє побудові активної та стійкої онлайн-спільноти шанувальників аніме, яка з часом може розширюватися та вдосконалюватися завдяки зворотному зв'язку користувачів і впровадженню нових функцій.

### 3.3. Тестування роботи програмного продукту

Створення будь-якого програмного продукту супроводжується появою помилок на різних етапах розробки, адже неможливо передбачити всі можливі варіанти використання та несподівані ситуації. Тестування є невід'ємною складовою процесу розробки, оскільки воно дозволяє виявити недоліки, підвищити якість та стабільність системи перед її впровадженням у реальну експлуатацію.

У рамках розробки платформи для ведення персоналізованих списків перегляду аніме та організації обговорень було застосовано комплексний підхід до тестування, що включав модульні, інтеграційні та функціональні перевірки. Модульне тестування було спрямоване на перевірку коректності окремих функцій і методів, які реалізують логіку роботи з користувачами, списками перегляду, коментарями та голосуваннями. Інтеграційне тестування дозволило переконатися у правильній взаємодії клієнтської частини, серверної логіки та бази даних, а функціональне тестування підтвердило відповідність системи поставленим вимогам, таким як автентифікація через email, додавання аніме до персоналізованих списків, створення коментарів і механізми голосування.

|      |      |                 |        |      |                                            |      |
|------|------|-----------------|--------|------|--------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка.25.121.21.000 - ПЗ | Арк. |
|      |      | Чижмотря О.В.   |        |      |                                            | 66   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                            |      |

Для серверної частини, реалізованої на базі Node.js і MongoDB, було застосовано Jest, що дозволило протестувати основні контролери та сервіси, які відповідають за управління користувачами, списками перегляду, коментарями та перевірку прав доступу. Під час тестування особлива увага приділялася перевірці коректної обробки токенів доступу, а також валідації сценаріїв використання API, що дозволило виявити та усунути низку помилок, пов'язаних із неправомірним доступом і некоректною логікою роботи з голосуванням.

Проведене тестування (рисунок 3.14) сприяло виявленню багатьох важливих недоліків, серед яких некоректна обробка помилок при передачі неправильних або прострочених токенів, логічні помилки при оновленні рейтингу коментарів і недостатній контроль доступу до функцій видалення чи редагування. В результаті виправлень було значно покращено стабільність та безпеку системи, а також оптимізовано швидкодію API, що дозволило коректно обробляти запити навіть при великій кількості одночасних користувачів.

```
> server@1.0.0 test
> jest

PASS tests/commentRoutes.test.js
PASS tests/userRoutes.test.js
PASS tests/topicRoutes.test.js
PASS tests/animeRoutes.test.js

Test Suites: 4 passed, 4 total
Tests:       25 passed, 25 total
Snapshots:   0 total
Time:        0.596 s, estimated 1 s
Ran all test suites.
```

Рисунок 3.14 - Результати запуску unit-тестів для бекенд-частини програмного продукту

Клієнтська частина програмного продукту була протестована на відповідність усім ключовим сценаріям роботи платформи для обговорення та ведення персоналізованих списків перегляду аніме. Зокрема, проведено перевірку процесів реєстрації та авторизації користувачів, додавання та

відображення обговорень, коментування та оцінювання коментарів, а також роботи пошуку та сортування списку аніме. Окремо протестовано можливість додавання аніме до списку перегляду, взаємодію з профілем користувача (відображення створених обговорень, вподобаних обговорень, коментарів, підписок та підписників) і коректність роботи навігації між розділами платформи.

Узагальнені результати тестування клієнтської частини наведено у таблиці 3.1.

Таблиця 3.1.

| Назва тест-кейсу                               | Короткий опис дій                                                                        | Очікуваний результат                                                                                            | Фактичний результат                                         | Статус   |
|------------------------------------------------|------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------|----------|
| Відкриття головної сторінки                    | Відкрити головну сторінку платформи через браузер                                        | Головна сторінка відображається коректно, усі основні елементи доступні (список обговорень, список аніме)       | Головна сторінка завантажується коректно, елементи доступні | Пройдено |
| Перехід на сторінку аніме                      | Клікнути на картку аніме на головній сторінці                                            | Відкривається сторінка аніме з детальною інформацією (опис, жанри, персонажі, кнопка додати у список перегляду) | Відкривається сторінка аніме, всі елементи видно            | Пройдено |
| Додавання аніме у список перегляду             | Авторизуватись, перейти на сторінку аніме, натиснути кнопку «Додати до списку перегляду» | Аніме додається до списку перегляду користувача, з'являється повідомлення про успіх                             | Аніме додано, повідомлення з'явилося                        | Пройдено |
| Коментування обговорення                       | Авторизуватись, відкрити обговорення, додати коментар                                    | Коментар відображається під обговоренням                                                                        | Коментар доданий і відображений                             | Пройдено |
| Лайк коментаря                                 | Авторизуватись, відкрити обговорення, натиснути кнопку «лайк» під коментарем             | Кількість лайків збільшується на 1                                                                              | Кількість лайків збільшилася                                | Пройдено |
| Перевірка редиректу неавторизованого коментаря | Відкрити обговорення без авторизації                                                     | Користувача переадресовано на сторінку логіну                                                                   | Виконано редирект на логін                                  | Пройдено |

Продовження таблиці 3.1

|                               |                                                                                 |                                                                         |                                                  |          |
|-------------------------------|---------------------------------------------------------------------------------|-------------------------------------------------------------------------|--------------------------------------------------|----------|
| Реєстрація користувача        | Відкрити форму реєстрації, заповнити дані, надіслати форму                      | Користувач створений, повідомлення про підтвердження пошти              | Користувач створений, лист на пошту відправлений | Пройдено |
| Зміна паролю                  | Авторизуватись, відкрити форму зміни паролю, ввести новий пароль                | Пароль змінено                                                          | Пароль змінено                                   | Пройдено |
| Перехід на профіль            | Авторизуватись, натиснути на аватар або посилання профілю                       | Відкривається сторінка профілю з інформацією про користувача і списками | Сторінка профілю відкрилася                      | Пройдено |
| Відображення списку перегляду | Авторизуватись, відкрити сторінку профілю, перейти у вкладку «Список перегляду» | Відображається список аніме, доданих користувачем                       | Список перегляду відображається                  | Пройдено |

Таким чином, результати проведеного тестування підтверджують стабільність і надійність як серверної, так і клієнтської частин програмного продукту. Усі основні сценарії взаємодії користувачів працюють коректно, а функціонал відповідає поставленим вимогам. Комплексний підхід до тестування дозволив виявити й усунути потенційні недоліки ще на етапі розробки, що значно підвищує якість та експлуатаційну готовність системи.

### Висновки до третього розділу

У третьому розділі було представлено діаграму розгортання розробленої платформи для ведення персоналізованих списків перегляду аніме та обговорення серіалів і фільмів, а також детально описано основні інтерфейси та функціональні можливості системи для різних ролей користувачів. Зокрема, описано можливості для звичайних користувачів, які можуть створювати власні списки перегляду, додавати туди аніме, оновлювати статус перегляду та брати участь в обговореннях, а також для адміністраторів, які здійснюють модерацію коментарів і керують контентом платформи. Продемонстровано порядок взаємодії користувачів із ключовими модулями системи, включаючи механізми

|      |      |                 |        |      |                                             |      |
|------|------|-----------------|--------|------|---------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка. 25.121.21.000 - ПЗ | Арк. |
|      |      | Чижмотря О.В.   |        |      |                                             | 69   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                             |      |

автентифікації через email, створення і редагування персоналізованих списків, залишення коментарів, голосування за коментарі інших учасників обговорення та пошук аніме за різними критеріями.

Комплексне тестування дало змогу виявити й усунути ряд недоліків на ранніх етапах розробки, серед яких були некоректна обробка прострочених токенів доступу, помилки при оновленні статусів перегляду у списках, неточності у відображенні рейтингу коментарів та окремі логічні помилки при голосуванні за коментарі. За результатами тестування було підтверджено відповідність функціональних можливостей платформи встановленим вимогам, досягнуто належного рівня стабільності та надійності роботи системи навіть при значному навантаженні, що свідчить про готовність продукту до використання кінцевими користувачами.

|      |      |                |        |      |                                            |      |
|------|------|----------------|--------|------|--------------------------------------------|------|
|      |      | Лоєвський Я.Д. |        |      | Житомирська політехніка.25.121.21.000 - ПЗ | Арк. |
|      |      | Чижмотря О.В.  |        |      |                                            | 70   |
| Змн. | Арк. | № докум.       | Підпис | Дата |                                            |      |

## ВИСНОВКИ

У процесі виконання бакалаврської кваліфікаційної роботи було спроектовано та реалізовано сучасну веборієнтовану платформу для ведення персоналізованих списків перегляду та обговорення аніме. На основі аналізу предметної області та ринку аніме-платформ, а також порівняльної характеристики провідних аналогів, таких як MyAnimeList і AniList, було визначено основні вимоги до функціоналу системи, її архітектури та користувацького інтерфейсу.

У ході роботи було проведено ґрунтовне дослідження існуючих рішень, проаналізовано їх сильні та слабкі сторони, що дозволило сформулювати конкурентні переваги власного продукту — гнучку персоналізацію списків перегляду з можливістю додавання користувацьких тегів і статусів, інтеграцію обговорень із системою голосування за коментарі, підтримку соціальних функцій для взаємодії між користувачами, а також адаптацію інтерфейсу для української аудиторії. У результаті проектування було створено структуру бази даних для зберігання даних користувачів, аніме, коментарів та голосувань, розроблено сучасний вебінтерфейс із адаптивною версткою для зручного використання на різних пристроях, а також реалізовано механізми захисту облікових записів і персональних даних із використанням токенів доступу та сучасних засобів шифрування.

Детально описано основні сценарії використання системи, зокрема створення та редагування персоналізованих списків перегляду з можливістю зміни статусу перегляду аніме, залишення коментарів та участь у обговореннях окремих серіалів або фільмів, голосування за корисні коментарі, пошук і фільтрація аніме за жанрами, студіями та рейтингами. Крім того, користувачам надано можливість автентифікації за допомогою електронної пошти.

Розроблена платформа успішно пройшла комплексне тестування, у ході якого були перевірені всі основні функціональні модулі системи, зокрема реєстрація та автентифікація користувачів, робота з персоналізованими списками перегляду, створення та видалення коментарів, а також механізми

|      |      |                 |        |      |                                            |      |
|------|------|-----------------|--------|------|--------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка.25.121.21.000 - ПЗ | Арк. |
|      |      | Чижемотря О.В.  |        |      |                                            | 71   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                            |      |

голосування. У процесі тестування було виявлено та усунуто низку помилок, серед яких некоректна обробка невалідних токенів доступу, помилки при оновленні статусів у списках перегляду та окремі проблеми при синхронізації даних під час голосування. У результаті платформа була оптимізована для стабільної роботи під навантаженням і готова до впровадження для використання кінцевими користувачами.

Розроблена система відповідає сучасним вимогам до автоматизації процесів організації персонального перегляду та обговорення аніме, сприяє формуванню спільноти однодумців, підвищує зручність взаємодії з контентом і створює передумови для подальшого розвитку продукту відповідно до актуальних потреб ринку та побажань користувачів.

|      |      |                 |        |      |                                            |      |
|------|------|-----------------|--------|------|--------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка.25.121.21.000 - ПЗ | Арк. |
|      |      | Чижмотря О.В.   |        |      |                                            | 72   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                            |      |

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Myanimelist.net [Електронний ресурс] – Режим доступу до ресурсу:  
<https://myanimelist.net>
2. Anilist.co [Електронний ресурс] – Режим доступу до ресурсу:  
<https://anilist.co/>
3. Kitsu [Електронний ресурс] – Режим доступу до ресурсу:  
<https://kitsu.app/explore/anime>
4. Документація React [Електронний ресурс] – Режим доступу до ресурсу:  
<https://react.dev/>
5. Документація Vuejs [Електронний ресурс] – Режим доступу до ресурсу:  
<https://vuejs.org/>
6. Документація Angular [Електронний ресурс] – Режим доступу до ресурсу:  
<https://angular.io/>
7. Документація Expressjs [Електронний ресурс] – Режим доступу до ресурсу: <https://expressjs.com/>
8. Документація DjangoProject [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.djangoproject.com/>
9. Документація Spring [Електронний ресурс] – Режим доступу до ресурсу:  
<https://spring.io/>
10. Документація MongoDB [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mongodb.com/docs/>
11. Документація PostgreSQL [Електронний ресурс] – Режим доступу до ресурсу: <https://www.postgresql.org/docs/>
12. Документація MySQL [Електронний ресурс] – Режим доступу до ресурсу: <https://dev.mysql.com/doc/>
13. Документація React. "Introduction to React" [Електронний ресурс] – Режим доступу до ресурсу: <https://react.dev/learn>
14. Документація React. "Thinking in React" [Електронний ресурс] – Режим доступу до ресурсу: <https://react.dev/learn/thinking-in-react>

|      |      |                 |        |      |                                             |      |
|------|------|-----------------|--------|------|---------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка. 25.121.21.000 - ПЗ | Арк. |
|      |      | Чижемотря О.В.  |        |      |                                             | 73   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                             |      |

15. Документація Reduxjs [Електронний ресурс] – Режим доступу до ресурсу: <https://redux.js.org/>
16. Документація TypeScriptLang [Електронний ресурс] – Режим доступу до ресурсу: <https://www.typescriptlang.org/docs/>
17. Документація Expressjs [Електронний ресурс] – Режим доступу до ресурсу: <https://expressjs.com/>
18. Документація Nodejs [Електронний ресурс] – Режим доступу до ресурсу: <https://nodejs.org/docs/>
19. Документація Node.js Docs: Guides & Articles (Performance and Scalability) [Електронний ресурс] – Режим доступу до ресурсу: <https://nodejs.org/en/docs/guides/>
20. Документація Node.js Docs: Guides [Електронний ресурс] – Режим доступу до ресурсу: <https://nodejs.org/en/docs/guides/>
21. Документація Mongoosejs [Електронний ресурс] – Режим доступу до ресурсу: <https://mongoosejs.com/>
22. Документація MongoDB [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mongodb.com/docs/>
23. Документація MongoDB: Performance Tuning [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mongodb.com/docs/manual/administration/performance-tuning/>
24. Документація MongoDB: Sharding [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mongodb.com/docs/manual/sharding/>
25. Документація Expressjs. "Performance" [Електронний ресурс] – Режим доступу до ресурсу: <https://expressjs.com/en/advanced/best-practice-performance.html>
26. Документація MongoDB. "High Performance" [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mongodb.com/docs/manual/core/write-performance/>
27. Документація Git-scm [Електронний ресурс] – Режим доступу до ресурсу: <https://git-scm.com/docs>

|      |      |                 |        |      |                                            |      |
|------|------|-----------------|--------|------|--------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка.25.121.21.000 - ПЗ | Арк. |
|      |      | Чижемотря О.В.  |        |      |                                            | 74   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                            |      |

28. JetBrains. "The State of Developer Ecosystem 2024 Report" [Електронний ресурс] – Режим доступу до ресурсу:  
<https://www.jetbrains.com/lp/devecosystem-2024/>
29. Документація MongoDB. "Customer Stories" [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mongodb.com/customers>
30. Документація React. "Building with Components" [Електронний ресурс] – Режим доступу до ресурсу: <https://react.dev/learn/thinking-in-react#step-3-add-state-and-derive-ui>
31. Документація Node.js Docs: Guides & Articles (Scalability) [Електронний ресурс] – Режим доступу до ресурсу: <https://nodejs.org/en/docs/guides/>
32. Документація MongoDB: Sharding [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mongodb.com/docs/manual/sharding/>
33. Документація NPM [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.npmjs.com/>
34. Документація Getbootstrap [Електронний ресурс] – Режим доступу до ресурсу: <https://getbootstrap.com/>
35. Документація Redux [Електронний ресурс] – Режим доступу до ресурсу: <https://redux.js.org/>

# ДОДАТКИ

|      |      |                |        |      |                                            |      |
|------|------|----------------|--------|------|--------------------------------------------|------|
|      |      | Поєвський Я.Д. |        |      | Житомирська політехніка.25.121.21.000 - ПЗ | Арк. |
|      |      | Чижмотря О.В.  |        |      |                                            | 76   |
| Змн. | Арк. | № докум.       | Підпис | Дата |                                            |      |

## Ініціалізація бази даних:

```
const mongoose = require('mongoose');
const connectDB = async () => {
  try {
    await mongoose.set("strictQuery", false);
    await mongoose.connect(process.env.MONGODB_URL_CONNECTION, {
      useNewUrlParser: true,
      useUnifiedTopology: true,
    });
    console.log('MongoDB Connected!');
  } catch(err) {
    console.log(err.message);
  }
}
module.exports = connectDB;
```

## Роути для юзерів:

```
const router = express.Router();

router.get("/:username", userController.getUserProfile);
router.get("/:username/comments", userController.getUserComments);
router.get("/:username/following", userController.getUserFollowing);
router.get("/:username/followers", userController.getUserFollowers);
router.put(
  "/:username/follow",
  validateAccessToken,
  userController.toggleUserFollow
);
router.put("/:username", validateAccessToken, userController.updateUserProfile);
```

## Роути для топіків:

```
router.get("/", topicController.getAllTopics);
router.get("/spaces", topicController.getSpaces);
router.get("/contributors", topicController.getTopContributors);
router.get("/:id/:slug", topicController.getTopic);
router.post("/", validateAccessToken, topicController.addTopic);
router.post(
  "/:id/upvote",
  validateAccessToken,
  topicController.toggleUpvoteTopic
);
router.post(
  "/:id/downvote",
  validateAccessToken,
  topicController.toggleDownvoteTopic
);
```

|      |      |                 |        |      |                                             |      |
|------|------|-----------------|--------|------|---------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка. 25.121.21.000 - ПЗ | Арк. |
|      |      | Чижемотря О.В.  |        |      |                                             | 77   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                             |      |

```
);
router.delete("/:id", validateAccessToken, topicController.deleteTopic);
```

#### Роути для коментарів:

```
router.post("/", validateAccessToken, commentController.addComment);
router.get("/helpers", commentController.getTopHelpers);
router.get("/:id", commentController.getTopicComments);
router.post(
  "/:id/upvote",
  validateAccessToken,
  commentController.toggleUpvoteComment
);
router.post(
  "/:id/downvote",
  validateAccessToken,
  commentController.toggleDownvoteComment
);
router.delete("/:id", validateAccessToken, commentController.deleteComment);
```

#### Роути для авторизації:

```
router.get("/refresh_token", authController.refresh_token);
router.post("/register", authController.register);
router.post("/login", authController.login);
router.post("/logout", validateAccessToken, authController.logout);
router.post("/send-email-verification", authController.sendEmailVerification);
router.post(
  "/verify-email",
  validateEmailVerifyToken,
  authController.emailVerify
);
router.post("/forgot-password", authController.sendForgotPassword);
router.post(
  "/reset-password",
  validatePasswordResetToken,
  authController.resetPassword
);
```

#### Роути для аніме:

```
router.get("/", animeController.getAllAnime);
router.get(
  "/watchlist",
  validateAccessToken,
  animeController.getWatchedAnimeList
);
router.post(
  "/watchlist/:animeId",
  validateAccessToken,
```

|      |      |                 |        |      |                                             |      |
|------|------|-----------------|--------|------|---------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка. 25.121.21.000 - ПЗ | Арк. |
|      |      | Чижевотря О.В.  |        |      |                                             | 78   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                             |      |

```

    animeController.addToWatchList
  );
  router.get("/:id", animeController.getAnimeById);

  router.post("/", validateAccessToken, animeController.addAnime);
  router.put("/:id", validateAccessToken, animeController.updateAnime);
  router.delete("/:id", validateAccessToken, animeController.deleteAnime);

  router.put(
    "/watch/:animeId",
    validateAccessToken,
    animeController.markAsWatched
  );

```

### Модель тегів:

```

const TagSchema = new mongoose.Schema(
  {
    name: String,
    createdBy: String,
  },
  { timestamps: true }
);

```

```

TagSchema.virtual("author", {
  ref: "User",
  localField: "createdBy",
  foreignField: "username",
  justOne: true,
});

```

### Модель коментарів:

```

const CommentSchema = new mongoose.Schema(
  {
    owner: String,
    content: String,
    parentTopic: {
      type: mongoose.Types.ObjectId,
      ref: "Topic",
    },
    parentComment: {
      type: mongoose.Types.ObjectId,
      ref: "Comment",
      default: null,
    },
    upvotes: [
      {
        type: String,
        ref: "User",
      }
    ]
  }
);

```

|      |      |                 |        |      |                                            |      |
|------|------|-----------------|--------|------|--------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка.25.121.21.000 - ПЗ | Арк. |
|      |      | Чижмотря О.В.   |        |      |                                            | 79   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                            |      |

```

        default: [],
      },
    ],
    downvotes: [
      {
        type: String,
        ref: "User",
        default: [],
      },
    ],
  },
  { timestamps: true }
);
модель просторів
const SpaceSchema = new mongoose.Schema(
  {
    name: String,
    avatar: String,
  },
  { timestamps: true }
);

```

#### Модель аніме:

```

const AnimeSchema = new mongoose.Schema({
  title: {
    type: String,
    required: true,
    unique: true,
  },
  description: String,
  coverImage: String,
  genres: [String],
  releaseDate: Date,
  episodes: Number,
  createdBy: {
    type: mongoose.Schema.Types.ObjectId,
    ref: "User",
  },
});

```

#### Модель юзерів:

```

const UserSchema = new mongoose.Schema({
  firstName: String,
  lastName: String,
  email: {
    type: String,
    unique: true,
  },
});

```

|      |      |                 |        |      |                                            |      |
|------|------|-----------------|--------|------|--------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка.25.121.21.000 - ПЗ | Арк. |
|      |      | Чижемотря О.В.  |        |      |                                            | 80   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                            |      |

```

},
username: {
  type: String,
  unique: true,
},
password: String,
avatar: {
  public_id: {
    type: String,
    default: null,
  },
  url: {
    type: String,
    default: "https://i.imgur.com/iV7Sdgm.jpg",
  },
},
cover: {
  public_id: {
    type: String,
    default: null,
  },
  url: {
    type: String,
    default: "https://i.imgur.com/jJBhmVy.jpeg",
  },
},
socialNetwork: {
  facebook: {
    type: String,
    trim: true,
    match:

```

```

/(?:https?:\W)?(?:www\.|m\.|mobile\.|touch\.|mbasic\.)(?:facebook\.com|fb(?:\.me|\.com))\V(?:!$)(?:(?
:\w)*#\V)?(?:pages\|pg\|photo\.php\?fbid=)?(?:[w\
]*\V)*?(?:\V)?(?:profile\.php\?id=)?(?:^V?&\s*)(?:\V|&|\?)?.*/gm,

```

```

    default: "",
  },
  twitter: {
    type: String,
    trim: true,
    match: /^(?:https?:\W)?(?:www\.)?twitter\.com\V(?:#\V)?[a-zA-Z0-9_]+$/i,
    default: "",
  },
  github: {
    type: String,
    trim: true,
    match: /^(?:https?:\W)?(?:www\.)?github\.com\V(?:#\V)?[a-zA-Z0-9_]+$/i,
    default: "",
  },
},

```

|      |      |                 |        |      |                                            |      |
|------|------|-----------------|--------|------|--------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка.25.121.21.000 - ПЗ | Арк. |
|      |      | Чижемотря О.В.  |        |      |                                            | 81   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                            |      |

```

},
bio: {
  type: String,
  trim: true,
  maxlength: 500,
  default: "Тут колись буде моя біографія",
},
isVerified: {
  type: Boolean,
  default: false,
},
following: [
  {
    type: String,
    default: [],
  },
],
followers: [
  {
    type: String,
    default: [],
  },
],
watchedAnime: [
  {
    type: mongoose.Schema.Types.ObjectId,
    ref: "Anime",
  },
],
});

```

Структура контролера на прикладі commentContoller:

```

let repliesToDelete = [];
const nest = (comments, id) => {
  comments
    .filter((comment) => comment?.parentComment?.toString() === id?.toString())
    .map((comment) => {
      repliesToDelete?.push(comment._id);
      nest(comments, comment?._id.toString());
    });
};

```

```

module.exports = {
  getTopicComments: async (req, res) => {
    try {
      const { id: parentTopic } = req.params;
      const comments = await Comment.find({ parentTopic })
        .populate({ path: "author", select: { password: 0, __v: 0 } })
    }
  }
};

```

|      |      |                 |        |      |                                            |      |
|------|------|-----------------|--------|------|--------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка.25.121.21.000 - ПЗ | Арк. |
|      |      | Чижемотря О.В.  |        |      |                                            | 82   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                            |      |

```

        .lean()
        .exec();
    return res.json({
        comments: comments,
        message: "Comments Retrieved!",
    });
} catch (err) {
    console.log(err.message);
}
},
addComment: async (req, res) => {
    let { id, comment, parentComment } = req.body;
    if (!id || !comment || comment?.trim()?.length === 0) {
        return res.status(400).json({
            message: "Comment field have not to be empty, Please fill it!",
        });
    }
    parentComment = parentComment || null;
    try {
        let createdComment = await Comment.create({
            owner: req.user.username,
            parentTopic: id,
            parentComment: parentComment,
            content: comment.trim(),
        });
        if (createdComment) {
            await Topic.findByIdAndUpdate(id, {
                $inc: { totalComments: 1 },
            });
        }
        createdComment = await createdComment.populate({
            path: "author",
            select: { password: 0, __v: 0 },
        });
        return res.status(201).json({
            comment: createdComment,
            message: "Comment Created Successfully!",
        });
    } catch (err) {
        console.log(err.message);
    }
},
deleteComment: async (req, res) => {
    let { id } = req.params;
    try {
        repliesToDelete = [];
        const rootComment = await Comment.findById(id);
        if (req.user.username !== rootComment.owner) {
            return res.status(403).json({

```

|      |      |                 |        |      |                                            |      |
|------|------|-----------------|--------|------|--------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка.25.121.21.000 - ПЗ | Арк. |
|      |      | Чижемотря О.В.  |        |      |                                            | 83   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                            |      |

```

        message: "You are not allowed to delete this comment",
    });
}
const comments = await Comment.find();
nest(comments, rootComment._id.toString());
repliesToDelete.push(rootComment._id);
await Comment.deleteMany({ _id: repliesToDelete });
await Topic.findByIdAndUpdate(rootComment.parentTopic, {
    $inc: { totalComments: -repliesToDelete.length },
});
return res.status(200).json({
    deletedComments: repliesToDelete,
    message: "Comment Successfully Deleted!",
});
} catch (err) {
    console.log(err.message);
}
},
toggleUpvoteComment: async (req, res) => {
    const { id } = req.params;
    try {
        const comment = await Comment.findById(id);
        if (!comment) {
            return res.status(404).json({
                message: "Comment not found!",
            });
        }
        if (comment.upvotes.includes(req.user.username)) {
            comment.upvotes.pull(req.user.username);
            await comment.save();
            return res.status(200).json({
                commentId: id,
                username: req.user.username,
                message: "Comment was upvoted successfully.",
            });
        } else {
            comment.upvotes.push(req.user.username);
            comment.downvotes.pull(req.user.username);
            await comment.save();
            return res.status(200).json({
                commentId: id,
                username: req.user.username,
                message: "Comment was upvoted successfully.",
            });
        }
    } catch (err) {
        return res.status(403).json({
            Error: err.message,
        });
    }
}

```

|      |      |                 |        |      |                                             |      |
|------|------|-----------------|--------|------|---------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка. 25.121.21.000 - ПЗ | Арк. |
|      |      | Чижемотря О.В.  |        |      |                                             | 84   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                             |      |

```

    }
  },
  toggleDownvoteComment: async (req, res) => {
    const { id } = req.params;
    try {
      const comment = await Comment.findById(id);
      if (!comment) {
        return res.status(404).json({
          message: "Comment not found!",
        });
      }
      if (comment.downvotes.includes(req.user.username)) {
        comment.downvotes.pull(req.user.username);
        await comment.save();
        return res.status(200).json({
          commentId: id,
          username: req.user.username,
          message: "Comment was downvoted successfully.",
        });
      } else {
        comment.downvotes.push(req.user.username);
        comment.upvotes.pull(req.user.username);
        await comment.save();
        return res.status(200).json({
          commentId: id,
          username: req.user.username,
          message: "Comment was downvoted successfully.",
        });
      }
    } catch (err) {
      return res.status(403).json({
        Error: err.message,
      });
    }
  },
  getTopHelpers: async (req, res) => {
    try {
      let topHelpers = await Comment.aggregate([
        { $group: { _id: "$owner", count: { $sum: 1 } } },
        { $sort: { count: -1 } },
        {
          $lookup: {
            from: "users",
            localField: "_id",
            foreignField: "username",
            as: "author",
            pipeline: [{ $project: { password: 0, __v: 0 } }],
          },
        },
      ],
    ),
  },

```

|      |      |                 |        |      |                                             |      |
|------|------|-----------------|--------|------|---------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка. 25.121.21.000 - ПЗ | Арк. |
|      |      | Чижемотря О.В.  |        |      |                                             | 85   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                             |      |

```

    { $unwind: "$author" },
    { $limit: 3 },
  ]);
  return res.status(200).json(topHelpers);
} catch (err) {
  console.log(err.message);
}
},
};

```

Middleware для валідації токена:

```

const validateAccessToken = async (req, res, next) => {
  const bearerToken = req?.headers?.authorization;

  if (!bearerToken || !bearerToken.startsWith("Bearer ")) {
    return res.status(401).json({
      message: "You are not logged-in!",
    });
  }

  const accessToken = bearerToken.split(" ")[1];

  jwt.verify(
    accessToken,
    process.env.ACCESS_TOKEN_SECRET_KEY,
    async (err, decoded) => {
      if (err) {
        if (err instanceof TokenExpiredError) {
          return res
            .status(401)
            .json({ message: "Unauthorized! Access Token was expired!" });
        }
      }
      if (!decoded) {
        return res.status(401).json({
          message: "Unauthorized!",
        });
      }
      const { username, email } = decoded;
      req.user = { username, email };
      next();
    }
  );
};

module.exports = validateAccessToken;

```

|      |      |                 |        |      |                                             |      |
|------|------|-----------------|--------|------|---------------------------------------------|------|
|      |      | Лосєвський Я.Д. |        |      | Житомирська політехніка. 25.121.21.000 - ПЗ | Арк. |
|      |      | Чижемотря О.В.  |        |      |                                             | 86   |
| Змн. | Арк. | № докум.        | Підпис | Дата |                                             |      |