

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:

**«Застосунок для створення мультимедійних  
історій»**

Виконав студент групи ПЗк-23-1  
РОМАНШИНА Валерія Вікторівна

Керівник роботи:  
ГОДЛЕВСЬКИЙ Юрій Олександрович

Рецензент:  
ЧИЖМОТРЯ Олексій Володимирович

Житомир – 2025

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: **РОМАНІШИНА Валерія Вікторівна**

Керівник роботи: **ГОДЛЕВСЬКИЙ Юрій Олександрович**

Тема роботи: **«Застосунок для створення мультимедійних історій»**,  
затверджена Наказом закладу вищої освіти від **«14» лютого 2025 р., №61/с**  
Вихідні дані для роботи: **технології та засоби розробки застосунку для створення мультимедійних історій та програмні інструменти та системи, які дозволяють створювати інтерактивні мультимедійні історії.**

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

| Розділ | Консультант      | Завдання видав | Завдання прийняв |
|--------|------------------|----------------|------------------|
| 1      | Годлевський Ю.О. | 14.02.2025р.   | 14.02.2025р.     |
| 2      | Годлевський Ю.О. | 16.03.2025р.   | 16.03.2025р.     |
| 3      | Годлевський Ю.О. | 06.06.2025р.   | 06.06.2025р.     |

## РЕФЕРАТ

Випускна кваліфікаційна робота бакалавра складається з застосунку для створення мультимедійних історій та пояснювальної записки. Пояснювальна записка до випускної роботи містить 83 сторінок, 69 ілюстрацій та 14 таблиць.

Метою є розробка інтуїтивно зрозумілого застосунку для створення мультимедійних історій є надання користувачу середовища для створення інтерактивних історій за допомогою різноманітних мультимедійних елементів, таких як: картинки, відео, аудіо, створення розв'язок у історіях та їх публікація більш широкому колу користувачів.

Архітектура застосунку для створення мультимедійних історій базується на компонентно-орієнтованій архітектурі з використанням React. Інтерфейс складається з незалежних, повторно використовуваних компонентів, що полегшує підтримку, тестування та масштабування. Структура проєкту організована за функціональними директоріями (components, styles, server, utilities, controller, routing) для чіткого розподілу відповідальностей.

Додаток має зручний інтерфейс, який дозволяє користувачам легко створювати, переглядати та редагувати мультимедійні історії, взаємодіяти з контентом та має свою систему модерації контенту та адміністрування застосунку. Також веб-застосунок має приємний користувацький досвід, завдяки інтуїтивно зрозумілим компонентам управління та зручній навігації.

**КЛЮЧОВІ СЛОВА:** ЗАСТОСУНОК, ІНТУЇТИВНІСТЬ, КОРИСТУВАЦЬКИЙ ДОСВІД, МУЛЬТИМЕДІА, БАЗА ДАНИХ, ІНТЕРФЕЙС, ЦИФРОВИЙ КОНТЕНТ.

|           |      |                  |        |      |                                                 |  |  |                     |      |         |    |  |
|-----------|------|------------------|--------|------|-------------------------------------------------|--|--|---------------------|------|---------|----|--|
|           |      |                  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ |  |  |                     |      |         |    |  |
| Змн.      | Арк. | № докум.         | Підпис | Дата | Застосунок для створення мультимедійних історій |  |  | Літ.                | Арк. | Аркушів |    |  |
| Розроб.   |      | Романішина В.В.  |        |      |                                                 |  |  |                     |      |         |    |  |
| Керівник  |      | Годлевський Ю.О. |        |      |                                                 |  |  |                     |      | 4       | 61 |  |
|           |      |                  |        |      |                                                 |  |  | ФІКТ, ар. ІПЗк-23-1 |      |         |    |  |
| Н. контр. |      |                  |        |      |                                                 |  |  |                     |      |         |    |  |
| Затвердив |      |                  |        |      |                                                 |  |  |                     |      |         |    |  |
|           |      |                  |        |      | Пояснювальна записка                            |  |  |                     |      |         |    |  |

## ABSTRACT

The bachelor's qualification thesis consists of an application for creating multimedia stories and an explanatory note. The explanatory note to the thesis comprises 83 pages, 69 illustrations, and 14 tables. The goal of the work is to develop an intuitive application for creating multimedia stories that provides users with an environment to build interactive narratives using various multimedia elements such as images, videos, audio, branching storylines, and publishing them for a wider audience. The architecture of the application for creating multimedia stories is based on a component-oriented approach using React. The interface consists of independent, reusable components, which simplifies maintenance, testing, and scalability. The project structure is organized into functional directories (components, styles, server, utilities, controller, routing) to ensure a clear separation of concerns. The application features a user-friendly interface that allows users to easily create, view, and edit multimedia stories, interact with content, and includes its own content moderation and application administration system. The web application also provides a pleasant user experience, thanks to intuitive control components and easy navigation.

KEYWORDS: APPLICATION, INTUITIVENESS, USER EXPERIENCE, MULTIMEDIA, DATABASE, INTERFACE, DIGITAL CONTENT.

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 |      |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 | 4    |

## ЗМІСТ

|                                                                                                        |           |
|--------------------------------------------------------------------------------------------------------|-----------|
| <b>РЕФЕРАТ .....</b>                                                                                   | <b>3</b>  |
| <b>ABSTRACT .....</b>                                                                                  | <b>4</b>  |
| <b>ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ .....</b>                                                                 | <b>6</b>  |
| <b>ВСТУП .....</b>                                                                                     | <b>7</b>  |
| <b>РОЗДІЛ 1. АНАЛІЗ ВИМОГ КОРИСТУВАЧА ТА КОНЦЕПТУАЛЬНЕ<br/>МОДЕЛЮВАННЯ .....</b>                       | <b>9</b>  |
| 1.1. Опис проєкту та постановка задачі. ....                                                           | 9         |
| 1.2. Аналіз аналогів програмного продукту. ....                                                        | 11        |
| 1.3. Вибір архітектури застосунків для створення мультимедійних історій.<br>.....                      | 15        |
| 1.4. Вибір інструментальних засобів розробки. ....                                                     | 17        |
| Висновки до першого розділу. ....                                                                      | 22        |
| <b>РОЗДІЛ 2. ПРОЕКТУВАННЯ ЗАСТОСУНКУ ДЛЯ СТВОРЕННЯ<br/>МУЛЬТИМЕДІЙНИХ .....</b>                        | <b>23</b> |
| 2.1. Визначення варіантів використання та об'єктно-орієнтованої структури<br>системи .....             | 23        |
| 2.2. Розробка бази даних системи .....                                                                 | 39        |
| 2.3. Проектування та реалізація алгоритмів роботи системи .....                                        | 43        |
| 2.4. Реалізація функціоналу застосунку для створення мультимедійних<br>історій .....                   | 46        |
| Висновки до другого розділу .....                                                                      | 54        |
| <b>РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ ЗАСТОСУНКУ ДЛЯ<br/>СТВОРЕННЯ МУЛЬТИМЕДІЙНИХ ІСТОРІЙ .....</b> | <b>55</b> |
| 3.1. Порядок встановлення та налаштування параметрів застосунку .....                                  | 55        |
| 3.2 Структура інтерфейсу застосунку для створення інтерактивних історій<br>.....                       | 56        |
| 3.3. Взаємодія з інтерфейсом та тестування застосунку .....                                            | 65        |
| Висновки до третього розділу .....                                                                     | 79        |
| <b>ВИСНОВКИ .....</b>                                                                                  | <b>80</b> |
| <b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....</b>                                                                 | <b>81</b> |
| <b>ДОДАТКИ .....</b>                                                                                   | <b>83</b> |

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 5    |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

RESTful API — це тип API, який дотримується принципів REST. Він працює через HTTP і дозволяє програмам взаємодіяти одна з одною, обмінюючись даними у зрозумілому та структурованому вигляді.

HTTP — це протокол, який визначає, як передавати інформацію між клієнтом і сервером у мережі. Він використовується для запитів і відповідей між сторонами.

CRUD — це аббревіатура для чотирьох основних дій з даними в додатках: створення, читання, оновлення та видалення інформації.

API — це інтерфейс, який дозволяє одній програмі взаємодіяти з іншою за встановленими правилами, не розкриваючи внутрішню реалізацію.

UI (User Interface) — це все, що користувач бачить і з чим взаємодіє у додатку або на сайті.

UX (User Experience) — це загальне враження користувача від роботи з інтерфейсом: наскільки все зрозуміло, зручно та ефективно.

HTML (HyperText Markup Language) — це мова розмітки, яка створює структуру веб-сторінки. Вона визначає, де буде текст, зображення, заголовки, посилання тощо.

CSS (Cascading Style Sheets) — це мова стилів, яка відповідає за зовнішній вигляд веб-сторінки: кольори, шрифти, розміри, розташування елементів та анімації.

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 |      |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 | 6    |

## ВСТУП

**Актуальність** теми розробки застосунку для створення мультимедійних історій є необхідність задоволення зростаючого попиту на такі продукти. Існуючі рішення, як правило, мають обмежені можливості або високу вартість і також, подібні рішення часто не мають української локалізації, що робить їх недоступними та незручним для широкого кола українських користувачів. Крім того, розробка застосунку, що дозволяє легко та швидко створювати інтерактивні мультимедійні історії без необхідності глибоких технічних знань, сприятиме демократизації цього виду творчості, залучаючи більше людей до створення україномовного цифрового контенту.

З огляду на вищезазначене, розробка застосунку для створення інтерактивних мультимедійних історій має велике значення для розвитку сучасного цифрового суспільства. Вона сприятиме появі нових форм вираження, підвищенню якості навчальних матеріалів, покращенню маркетингових комунікацій та збагаченню культурного ландшафту українського письменництва.

**Метою** є розробка інтуїтивно зрозумілого застосунку для створення інтерактивних мультимедійних історій є надання користувачу середовища для створення інтерактивних історій за допомогою різноманітних мультимедійних елементів, таких як: картинки, відео, аудіо, створення розв'язок у історіях та їх публікація більш широкому колу користувачів.

**Основними завданнями** дипломної роботи є створення зручного та інтуїтивного інтерфейсу для користувачів, надання інструментів для реалізації їхніх ідей у сфері написання мультимедійних історій, забезпечення обміну текстовими матеріалами між користувачами з можливістю їх публікації на сторінці додатку. Додаток також дозволяє зберігати історії на особистому акаунті для подальшої публікації, а ще забезпечує доступ до вже опублікованих історій для їх читання на сторінці застосунку.

**Предметом дослідження** є методи, технології та засоби розробки застосунку для створення мультимедійних історій.

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 7    |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |

Це включає вивчення теоретичних основ мультимедіа, аналіз існуючих програмних рішень у цій сфері, а також розробку нових підходів та алгоритмів, що дозволяють створювати історії з використанням різних видів мультимедійного контентує основні поняття, визначення та класифікація мультимедійних елементів, різні рівні та види інтерактивності з їх значенням та впливом на користувацький досвід.

Також розглядаються методи інтеграції мультимедійного контенту, включаючи текст, зображення, відео та аудіо в інтерактивні історії. Увага приділяється технічним засобам розробки, таким як мови програмування, бібліотеки, платформи та середовища розробки, які використовуються для створення інтерактивних мультимедійних історій, а також методам оцінки та покращення взаємодії користувача з мультимедійними історіями.

**Об'єктом дослідження** є програмні інструменти та системи, які дозволяють створювати інтерактивні мультимедійні історії. Це включає як існуючі рішення, такі як редактори мультимедійного контенту, так і нові інструменти, що розробляються у рамках даного дипломного проекту. До основних компонентів об'єкта дослідження належать редактори та платформи, які дають можливість створювати й редагувати мультимедійний контент, системи для організації, зберігання та забезпечення доступу до мультимедійних елементів, інтерфейси користувача, алгоритми для ефективної обробки й інтеграції різноманітного мультимедійного контенту, а також технології, що використовуються у застосунку.

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 |      |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 | 8    |

# РОЗДІЛ 1. АНАЛІЗ ВИМОГ КОРИСТУВАЧА ТА КОНЦЕПТУАЛЬНЕ МОДЕЛЮВАННЯ

## 1.1. Опис проєкту та постановка задачі.

Метою кваліфікаційної роботи бакалавра є створення сучасного застосунку для створення мультимедійних історій, який надаватиме користувачам можливість комбінувати різноманітні типи контенту, створюючи інтерактивні та візуально привабливі оповіді. Такий застосунок має бути зручним у використанні, адаптивним до різних пристроїв і здатним масштабуватись із часом.

Мультимедійні історії є потужним засобом комунікації, що поєднує текст, зображення, аудіо, відео та інтерактивні елементи у єдину наративну структуру. Вони широко застосовуються в освіті, журналістиці, рекламі, культурних проєктах і соціальних медіа. Завдяки своїй гнучкості та емоційному впливу, мультимедійний формат стає дедалі більш популярним як спосіб донесення ідей, навчальних матеріалів або особистих історій. У зв'язку з цим виникає потреба у платформах, які забезпечують користувачам прості та ефективні інструменти для створення такого контенту.

Ідея розробки полягає у створенні універсального інструменту для генерації історій із можливістю додавання різних типів мультимедійних елементів. Особливу увагу приділено інтуїтивному користувацькому інтерфейсу, який дозволяє навіть недосвідченим користувачам без знань у сфері програмування легко створювати повноцінні мультимедійні продукти.

Проектований застосунок передбачає архітектуру, що базується на розділенні клієнтської та серверної частин. Клієнтська частина відповідатиме за взаємодію з користувачем, відображення контенту та обробку дій у реальному часі, тоді як серверна частина забезпечуватиме обробку даних, зберігання інформації та зв'язок з базою даних.

Вибір сучасних фреймворків та бібліотек дозволить реалізувати застосунок із дотриманням принципів модульності, повторного використання коду та високої продуктивності.

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 9    |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |

Розробка передбачає послідовну реалізацію кількох ключових етапів, кожен з яких виконує важливу функцію у створенні якісного програмного продукту. Спочатку здійснюється аналіз предметної області, що охоплює дослідження потреб користувачів та вивчення існуючих аналогів, аби визначити найкращі підходи і рішення. На етапі проектування створюються прототипи інтерфейсів та структури даних, що допомагає закласти міцний фундамент для подальшого кодування.

Реалізація включає розробку клієнтської частини, що забезпечує взаємодію користувача з додатком, і серверної логіки, яка відповідає за обробку запитів, управління даними та забезпечення продуктивності системи. Інтеграція між компонентами дозволяє синхронізувати всі частини програмного комплексу, забезпечуючи їх безперебійну взаємодію.

Тестування функціоналу є критичним етапом, під час якого проводиться перевірка системи на наявність помилок, оптимізація продуктивності та оцінка зручності використання. Завершальним етапом є підготовка детальної документації, що включає опис функціоналу, інструкції користувача і рекомендації для подальшого розвитку продукту. Це забезпечує прозорість проєкту та підтримує його довготривале функціонування.

Очікуваним результатом є створення повноцінного застосунку, що дозволяє створювати, зберігати, поширювати та переглядати мультимедійні історії.

Платформа повинна відповідати актуальним вимогам до веб-сервісів — забезпечувати стабільну роботу, високу швидкодію, безпеку збереження даних та зручність користування.

Таким чином, створення такого застосунку дозволить задовольнити потреби великої кількості користувачів, які шукають інструменти для творчого самовираження, навчання або професійної роботи з цифровим контентом.

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 10   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |

## 1.2. Аналіз аналогів програмного продукту.

Для початку, потрібно проаналізувати, який основний функціонал повинен мати застосунок. Для цього розглянемо кілька найбільш відомих та популярних аналогів, щоб краще зрозуміти ринок і визначити ключові особливості, які можуть бути враховані при розробці веб додатку.

Першим на черзі буде Twine.

Зовнішній вигляд головної сторінки у Twine є такий:

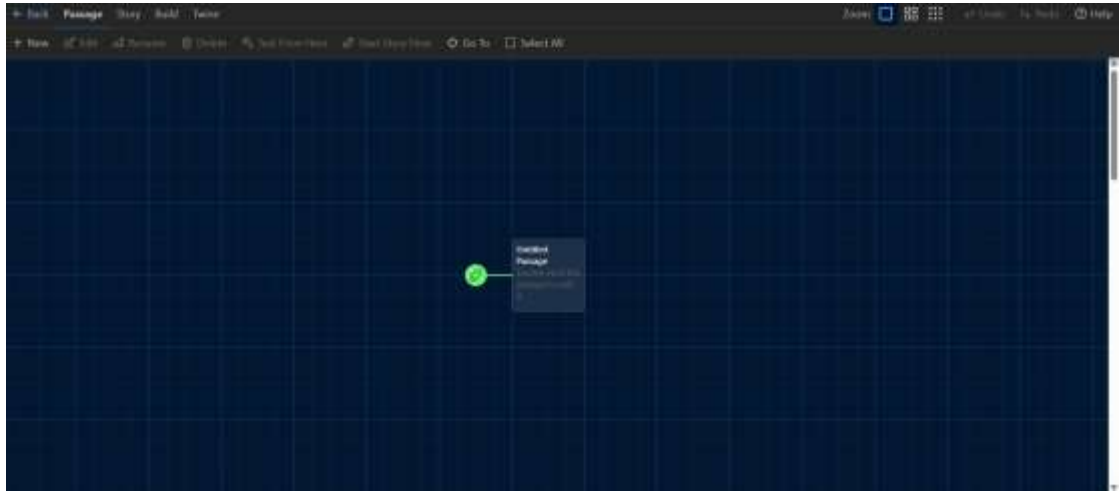


Рис. 1.1 Зовнішній вигляд головної сторінки у Twine

Twine – це безкоштовний інструмент з відкритим вихідним кодом, який дозволяє створювати інтерактивні оповідання, переважно у вигляді нелінійних текстових пригод. Це означає, що історії, створені у Twine, не обмежуються послідовним розвитком подій — користувач може сам обирати напрям сюжету через різні варіанти вибору. Такий формат дозволяє створювати багатогалузеві сценарії, де рішення гравця безпосередньо впливають на подальший розвиток подій.

Основні особливості Twine, є такими, а саме: інтерфейс користувача інтуїтивно зрозумілий, що дозволяє швидко створювати історії без знання програмування, є можливість використання CSS для стилізації та JavaScript для додавання складнішої логіки. Також Twine є безкоштовним і має відкритий код, що дозволяє розробникам вносити зміни та доповнення та історії можуть бути експортовані у форматі HTML, що робить їх легкодоступними через веб-браузери.

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 11   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |

Отже, Twine ідеально підходить для створення текстових пригод та інтерактивних романів, але має обмежені можливості у сфері мультимедіа. Але це лише конструктор, тому користувач зможе опублікувати свою історію тільки на інших платформах. А продукт, який буде розроблений, буде давати можливість публікувати готові історії на сайті, де інші користувачі зможуть їх прочитати. Також Twine не має української локалізації.

Наступним ми розглянемо веб додаток StoryWeaver.

Зовнішній вигляд головної сторінки у StoryWeaver є такий:

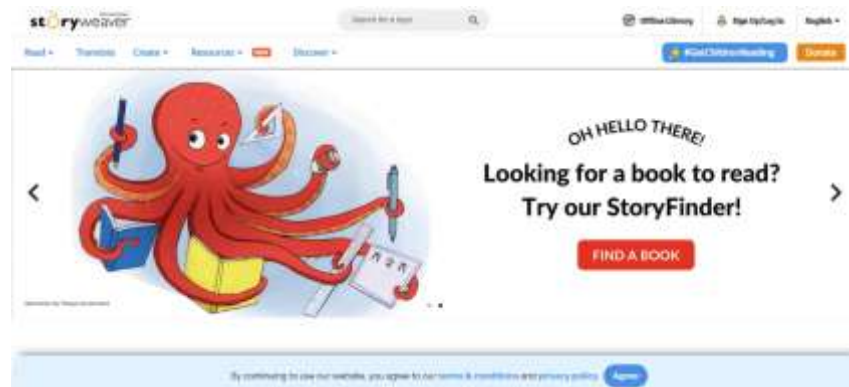


Рис. 1.2 Зовнішній вигляд головної сторінки у StoryWeaver

StoryWeaver – це потужний веб-додаток, створений для того, щоб допомогти користувачам створювати захопливі мультимедійні історії без необхідності володіння технічними навичками. Зручний інтерфейс дозволяє легко додавати та редагувати текст, зображення, відео та аудіо, створюючи інтерактивний контент.

Функції StoryWeaver включають такі можливості, як: інтерактивний редактор з інтуїтивно зрозумілим інтерфейсом, бібліотеку шаблонів для різних типів історій, інтеграцію з соціальними мережами для легкого поширення контенту, а також можливість експорту історій у різні формати, включаючи PDF та відео.

StoryWeaver підходить для освітніх проектів, маркетингових кампаній, художніх наративів та особистих проектів. Але він є дуже не оптимізованим, не має інтерактивності та української локалізації.

Останнім розглянемо веб додаток Medium

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 12   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |

Зовнішній вигляд головної сторінки у Medium є такий:

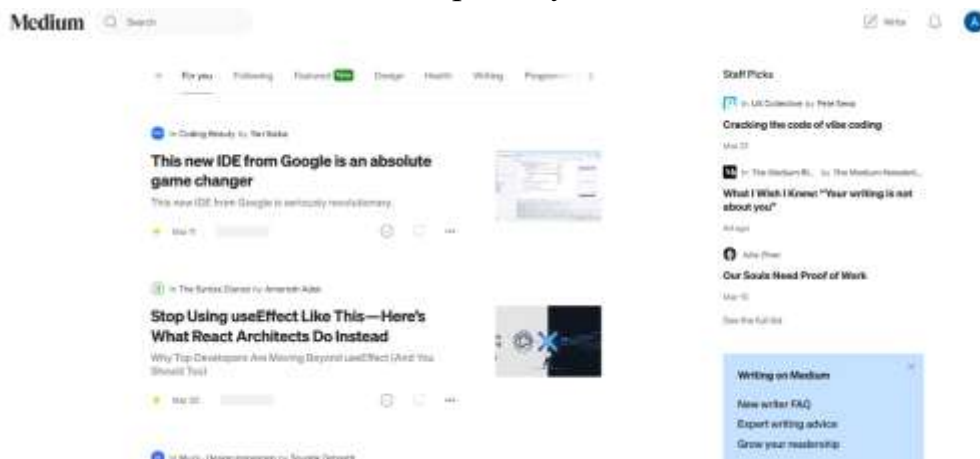


Рис. 1.3 Зовнішній вигляд головної сторінки у Medium

Medium - це веб-додаток, що дозволяє користувачам читати та публікувати статті, блоги та есе. Він створений як платформа для обміну думками, досвідом і знаннями, де кожен може стати автором і знайти аудиторію для своїх ідей. Тут можна знайти матеріали на різні теми, від технологій і бізнесу до культури і здоров'я.

Розглянемо основні особливості Medium, які представляють з себе, що користувачі можуть публікувати статті, есе, блоги та інші форми контенту на різні теми. Платформа дозволяє авторам знайти аудиторію для своїх публікацій, залучаючи читачів з усього світу. Також Medium має зручний інтерфейс з текстовими редакторами, що допомагає створювати контент, який виглядає чисто і професійно. Контент можна організовувати за категоріями, що спрощує пошук інформації. Користувачі мають можливість коментувати статті та брати участь в обговореннях, сприяючи обміну думками. Також можна підписуватися на улюблених авторів, щоб отримувати повідомлення про нові публікації. Medium надає інструменти для відстеження популярності статей, кількості переглядів та інші аналітичні дані. Ці функції роблять Medium відмінною платформою для авторів та читачів. Але ця платформа не надає інтерактивності у історіях. Всі ці застосунки мають свої стильні сторони, проте кожен з них має свої сильні та слабкі сторони. Twine більше підходить для текстових пригод та має розвилки в історіях, StoryWeaver орієнтований на навчальні матеріали і не має розвилки, а Medium спеціалізується на статтях та статичному тексті і також не має ніякої інтерактивності.

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 13   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |

Таблиця 1.1

Таблиця порівняння застосунків для створення мультимедійних історій

| Характеристика                    | Twine | Story Weaver | Medium | BlobStory<br>(додаток, що розробляється) |
|-----------------------------------|-------|--------------|--------|------------------------------------------|
| Інтерактивність                   | +     | -            | -      | +                                        |
| Підтримка мультимедіа             | +/-   | +            | +      | +                                        |
| Зручність створення контенту      | +     | +/-          | +      | +                                        |
| Публікація історій на платформі   | -     | -            | +      | +                                        |
| Наявність української локалізації | -     | -            | -      | +                                        |
| Наявність спільноти / коментарів  | -     | -            | +      | +                                        |
| Можливість створення розгалужень  | +     | -            | -      | +                                        |
| Аналітика переглядів              | -     | -            | +      | +                                        |
| Доступність                       | +     | +            | +      | +                                        |
| Безкоштовність                    | +     | +            | +/-    | +                                        |

Отже, наш додаток буде мати кращі риси цих інструментів, запропонувавши простоту використання, підтримку різноманітних мультимедійних елементів та гнучкість у створенні нелінійних наративів. Це дозволить задовольнити потреби як новачків, так і досвідчених розробників інтерактивних історій. Тому буде створено зручний додаток, в якому буде вбудований конструктор, для створення історій та можливість їх публікації на сам сайт, для обміну вражень, коментування та розвитку українського писемницького ком'юніті, бо не в одному з додатків не було знайдено української локалізації. А наш додаток не тільки буде зручним інструментом для написання історій, але і буде мати українську локалізацію, що стане гарною підтримкою для українських письменників.

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 14   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |

### 1.3. Вибір архітектури застосунків для створення мультимедійних історій.

Для реалізації даного проекту було використано клієнт-серверну архітектуру. Такий підхід дозволяє ефективно розподіляти навантаження між сервером і клієнтом, що особливо важливо для інтерактивних мультимедійних історій, де необхідно забезпечити швидку обробку даних та мінімальний час затримки.

Основними перевагами використання клієнт-серверної архітектури на клієнтській стороні є масштабованість, яка забезпечується збереженням логіки застосунку на сервері, що дозволяє легко оновлювати та модифікувати функціонал без необхідності внесення змін у клієнтську частину. Завдяки цьому архітектура спрощує підтримку та розвиток застосунків.

Швидкість доступу також є важливим аспектом: всі дані, необхідні для відображення мультимедійних елементів, завантажуються на клієнт, що забезпечує високу продуктивність і знижує затрати на обробку запитів. Гнучкість дизайну дозволяє адаптувати інтерфейс для різних пристроїв і екранів, завдяки використанню фреймворків для розробки інтерфейсу. Ці переваги роблять клієнт-серверну архітектуру ефективною та зручною для сучасних застосунків. Під час виконання дипломної роботи, була визначена така архітектура веб-додатку. У рамках цього проекту буде використовуватися компонентний підхід для розробки користувацького інтерфейсу. Для реалізації компонентів буде обрано React, який дозволяє створювати компоненти, що можуть бути повторно використані в різних частинах програми. Це значно спрощує підтримку та розвиток додатку, забезпечуючи чистоту коду та можливість масштабування проекту. Компонентний підхід дозволяє розбивати інтерфейс на невеликі, незалежні частини, кожна з яких відповідає за певну функціональність або відображення даних. Кожен компонент може мати власний стан та логіку, що робить код більш модульним, легким для тестування і повторного використання.

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 15   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |

Наприклад, окремо можуть бути розроблені компоненти для відображення мультимедійного контенту, футерів та хедерів, вкладок з інформацією, логіка яких буде ідентична на всіх сторінках, а також компоненти для управління користувацькими взаємодіями, такі як кнопки, меню або форми введення. Це дозволяє зручно оновлювати або замінювати окремі частини інтерфейсу без зміни решти системи. Для підтримки логічної організації проекту і зручності масштабування, структура директорій проекту буде мати такий вигляд: components – буде містити всі React-компоненти; styles – буде зберігати CSS-файли для стилізації додатка; server – для зберігання функцій для взаємодії з бекендом, такі як HTTP-запити; utilities – зберігання утилітних функцій та хелперів; controllers – для створення функцій роботи з базами даних та реалізації логіки взаємодії з системою через бекенд; route – буде використовуватися для управління навігацією всередині сервера.

У бекенд структурі буде застосовано такі технології, як фреймворк Express.js, який буде слугувати основою для створення RESTful API, що забезпечить швидкість та гнучкість розробки. Для логіки запитів буде використано контролери, які виконуватимуть обробку даних, а маршрути визначатимуть шляхи для запитів, забезпечуючи чітку організацію коду. Моделі даних будуть реалізовані з використанням PostgreSQL.

Цей підхід сприятиме ефективності та масштабованості серверної частини додатку. Наступним для опису буде база даних, яка буде представлена таким чином:

Схема даних – буде визначати структуру таблиць, поля з відповідними типами даних, обмеженнями у вигляді полів з унікальними значеннями та необхідними зв'язками між ними.

Міграції – буде забезпечувати можливість міграцій для оновлення схеми бази даних без втрати даних.

У якості основи для конструктора історій планується використання готової бібліотеки TuesdayJS, яка буде інтегрована у проєкт як ключовий елемент редактора історій. Ця бібліотека надає інструменти для побудови

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 16   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |

інтерактивних блокових структур, що ідеально підходить для створення сценаріїв, діалогів та логіки переходів між сценами. Функціональність TuesdayJS буде розширено та модифіковано відповідно до специфічних вимог застосунку. Зокрема, планується реалізація підтримки мультимедійних елементів (зображення, відео, аудіо), настроюваних типів виборів, а також інтеграція з бекендом для збереження прогресу створення історій. Бібліотека буде використана для побудови інтуїтивного інтерфейсу для авторів, що дозволить створювати та редагувати власні історії за допомогою зручного графічного редактора. Це дозволить користувачам без глибоких технічних знань реалізувати свої ідеї, структурувати сцени, налаштовувати переходи між ними та переглядати результат у режимі попереднього перегляду.

Інтеграція TuesdayJS забезпечить основу для створення історій на сайті, що значно покращить користувацький досвід і зробить платформу більш доступною та зручною для широкого кола авторів.

#### **1.4. Вибір інструментальних засобів розробки.**

Для створення програмного комплексу використано сучасні технології програмування. HTML5 забезпечує підтримку мультимедійного контенту, необхідного для реалізації інтерактивних візуальних історій, тоді як CSS3 дозволяє створювати адаптивний і привабливий інтерфейс для різних пристроїв. JavaScript разом із бібліотекою React реалізують компонентний підхід до розробки інтерфейсу, ефективно керують станом додатку та забезпечують високу швидкодію завдяки використанню віртуального DOM.

Для серверної частини застосовується Node.js, яка забезпечує асинхронну обробку запитів, високу швидкість та масштабованість розробки.

Як система керування базами даних використовується PostgreSQL — реляційна СУБД, що підтримує структуроване зберігання даних, складні SQL-запити, транзакції та забезпечує високу надійність.

PostgreSQL дозволяє ефективно працювати з взаємозв'язаними даними, масштабувати систему та забезпечує сувору типізацію, що сприяє цілісності та передбачуваності збережених даних.

|      |      |                  |        |      |                                                  |      |
|------|------|------------------|--------|------|--------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка». 24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                  | 17   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                  |      |

Окремо варто згадати бібліотеку TuesdayJS, яка є основою для створення інтерактивних візуальних новел. Вона забезпечує сценарний механізм, діалоги, переходи та анімації, що значно збагачує користувацький досвід. Ці технології створюють міцну базу для реалізації проєкту та його адаптації до потреб користувачів.

Для реалізації проєкту було використано різноманітні програмні продукти та інтегровані середовища розробки.

Visual Studio Code слугувало універсальним середовищем розробки для роботи з JavaScript, React, Node.js та іншими необхідними інструментами, забезпечуючи зручність написання, форматування та налагодження коду.

pgAdmin використовувався для керування базою даних PostgreSQL, включаючи перегляд, редагування та аналіз структури таблиць і вмісту даних у зручному графічному інтерфейсі.

Postman став корисним інструментом для тестування REST API, забезпечуючи ефективність та контроль коректної роботи серверної логіки.

Git та GitLab забезпечували контроль версій, збереження коду в хмарному репозиторії та можливості спільної роботи.

Для дизайну інтерфейсу було використано Figma, а Trello слугував зручним інструментом для планування і координації завдань за методологією Kanban.

Ці засоби стали ключовими для ефективної розробки та організації роботи над проєктом. Застосування зазначених інструментів дозволило ефективно організувати процес розробки, забезпечити гнучкість архітектури, зручність масштабування та високу якість кінцевого продукту.

### **Вибір клієнтських технологій застосунків.**

Для створення інтерактивного інтерфейсу застосунку для мультимедійних історій вибір клієнтських технологій базується на забезпеченні високої продуктивності, гнучкості у розробці, а також зручності для кінцевого користувача. Враховуючи потреби проєкту, для реалізації фронтенду були обрані наступні технології:

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 18   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |

HTML5 — це сучасний стандарт для створення веб-сторінок, що забезпечує підтримку мультимедійного контенту, необхідного для інтерактивних історій. HTML5 дозволяє вбудовувати відео, аудіо, анімації та інші медіафайли без потреби у сторонніх плагінах, що робить інтерактивні елементи доступними та зручними для користувача. Також він підтримує нові API, такі як Canvas, для малювання на веб-сторінках, що дозволяє створювати динамічний контент прямо в браузері.

CSS3 — надає потужні інструменти для стилізації веб-сторінок і створення адаптивного дизайну, який коректно відображатиметься на різних пристроях і екранах. З допомогою CSS3 можна реалізувати складні анімації, трансформації та ефекти, які дозволяють зробити інтерфейс інтерактивним та привабливим для користувачів. Завдяки медіа-запитам, CSS3 підтримує адаптивний дизайн, що автоматично налаштовує вигляд інтерфейсу залежно від розміру екрана, дозволяючи забезпечити зручність роботи на мобільних пристроях, планшетах та десктопах.

React — бібліотека для розробки користувацьких інтерфейсів, яка забезпечує високу продуктивність завдяки компонентному підходу та використанню віртуального DOM (Document Object Model). Віртуальний DOM дозволяє значно скоротити витрати на рендеринг інтерфейсу, обробляючи тільки ті частини сторінки, що змінилися, замість перерендерингу всієї сторінки. Це забезпечує високу швидкість роботи додатку, що особливо важливо для додатків з великою кількістю інтерактивних елементів і частими змінами стану, як у випадку з мультимедійними історіями.

Вибір цих клієнтських технологій дозволяє забезпечити високу швидкість розробки, продуктивність та зручність у підтримці додатку.

Вони ідеально підходять для реалізації складних і інтерактивних інтерфейсів, необхідних для створення мультимедійних історій, і дозволяють без труднощів працювати з великим обсягом контенту та взаємодіями користувача.

|      |      |                  |        |      |                                                  |      |
|------|------|------------------|--------|------|--------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка». 24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                  | 19   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                  |      |

## Вибір серверних технологій застосунків.

Для створення серверної частини веб-додатку, що дозволяє ефективно обробляти запити, зберігати та обробляти дані, було обрано наступні серверні технології:

### Бекенд технології

Node.js — забезпечує асинхронну обробку запитів, що дозволяє створювати високо продуктивні серверні додатки.

### База даних

PostgreSQL — це надійна реляційна база даних, яка зберігає дані у вигляді таблиць зі зв'язками між ними. Вона підтримує складні SQL-запити, транзакції, реплікацію для відмовостійкості та масштабування при зростанні навантаження. PostgreSQL дозволяє зберігати також JSON-дані, поєднуючи гнучкість з чіткою структурою. Завдяки стабільності та можливості роботи зі складними структурами, вона добре підходить для проєктів, які потребують надійності, контролю над даними та масштабованості.

### Бібліотека

TuesdayJS - це бібліотека для створення інтерактивних візуальних новел, яка надає прості інструменти для реалізації складних історій, діалогів та сценаріїв. Вона дозволяє створювати інтерактивний контент без необхідності глибоких знань програмування, що є важливим для прискорення розробки. Однак для реалізації специфічних функціональних вимог проєкту бібліотеку необхідно вбудовувати та адаптувати під потреби конкретного застосунку. Завдяки своїй гнучкості, TuesdayJS дозволяє швидко інтегрувати елементи мультимедійних історій у додаток, забезпечуючи тим самим високу інтерактивність та користувацьку привабливість.

Цей стек обрано через його здатність забезпечити ефективну розробку, високу продуктивність та надійність додатку. Використання React для фронтенду дозволить створити динамічний та легко масштабований інтерфейс. Node.js забезпечить швидку розробку серверної логіки. PostgreSQL гарантує надійне і структуроване зберігання даних, що особливо корисно для

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 20   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |

додатків з складними зв'язками між інформацією та потребою у точному контролі цілісності даних. Цей стек краще підходить для розробки інтерактивних мультимедійних історій через свою гнучкість, швидкість та можливість масштабування. Кожна технологія в цьому стеці вибрана з урахуванням її переваг і відповідності вимогам проекту, що дозволяє створити надійний, ефективний і користувацький орієнтований продукт.

### **Вибір середовища розробки.**

Для ефективної розробки веб-додатку, орієнтованого на створення інтерактивних мультимедійних історій, було обрано кілька потужних середовищ розробки та інструментів, які забезпечують зручність, ефективність і швидкість роботи над проєктом:

Visual Studio Code — це популярне, легке та багатифункціональне середовище для розробки програмного забезпечення, яке підтримує роботу з різними мовами програмування, включаючи JavaScript, React, Node.js та інші. Воно має великий набір розширень і плагінів, які значно покращують досвід розробника, зокрема для роботи з кодом, тестування, налагодження та інтеграцією з іншими інструментами.

PGAdmin — це графічний інтерфейс для роботи з базою даних PostgreSQL, який дозволяє зручно керувати даними, переглядати та редагувати структуру таблиць, виконувати SQL-запити та аналізувати інформацію. Він дає змогу швидко перевіряти дані, виконувати операції CRUD, а також забезпечує доступ до функцій реплікації, налаштувань безпеки PostgreSQL.

Postman — це потужний інструмент для тестування та розробки RESTful API. Він дозволяє створювати, тестувати та автоматизувати запити до серверних API, що є важливим етапом при розробці серверної логіки та інтеграції між фронтендом і бекендом. Postman дозволяє легко перевіряти правильність роботи API, що обробляє запити для інтерактивних історій, а також тестувати різні методи HTTP (GET, POST, PUT, DELETE).

Trello — це онлайн-інструмент для управління проєктами та командною роботою, який дозволяє ефективно організовувати завдання, етапи розробки

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 21   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |

та відстежувати прогрес проекту. Для керування процесом розробки інтерактивних мультимедійних історій Trello використовується для планування завдань за методологією Kanban, що дозволяє візуально відстежувати статус кожного етапу робіт.

Git — це система контролю версій, яка дозволяє відслідковувати зміни в коді, зберігати різні версії програми і працювати з гілками для паралельної розробки. GitLab є хмарним сервісом для зберігання та спільної роботи над репозиторіями Git, який дозволяє кільком розробникам працювати над проектом одночасно, а також забезпечує безпеку і резервне зберігання коду.

Figma — це популярний інструмент для дизайну інтерфейсів та прототипування, який дозволяє створювати макети веб-додатків і тестувати їх у реальному часі. Використання Figma дозволяє дизайнерам і розробникам ефективно співпрацювати над створенням UI/UX інтерфейсу та взаємодією користувача з додатком.

### **Висновки до першого розділу.**

У результаті проведеного аналізу предметної області були визначені основні вимоги до веб-застосунку для створення мультимедійних історій. Розглянуто декілька аналогів, серед яких Twine, StoryWeaver та Medium, що дозволило виявити їх сильні й слабкі сторони, зокрема відсутність інтерактивності, публікаційної платформи та української локалізації.

На основі порівняння сформовано перелік ключових вимог до власного застосунку, що поєднує зручність, мультимедійність і підтримку нелінійних наративів. Також обрано відповідні технології для реалізації майбутнього продукту.

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 22   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |

## РОЗДІЛ 2. ПРОЕКТУВАННЯ ЗАСТОСУНКУ ДЛЯ СТВОРЕННЯ МУЛЬТИМЕДІЙНИХ

### 2.1. Визначення варіантів використання та об'єктно-орієнтованої структури системи

Для забезпечення роботи та відповідності встановленим вимогам і стандартам, розроблено високорівневі вимоги для застосунку для створення інтерактивних мультимедійних історій. Ці вимоги допоможуть створити зручний і функціональний інструмент для користувачів, які бажають створювати й ділитися інтерактивними історіями.

#### Бізнес вимоги

Основні цілі - додаток розробляється для того, щоб полегшити процес створення мультимедійних історій для різних категорій користувачів, таких як письменники, художники, медіа-виробники, освітяни та новачки, які хочуть почати писати та публікувати мультимедійні історії.

Представлення проекту - застосунку буде представлений у вигляді веб-інтерфейсу з можливістю додавання медіаконтенту (тексти, зображення, відео, аудіо), інтерактивних елементів та сценаріїв для різних сюжетних гілок.

#### Вимоги користувачів

Можливість створення та редагування облікових записів користувачів;

Інтуїтивний інтерфейс для додавання мультимедійного контенту (текст, відео, зображення, аудіо);

Можливість створення нелінійних сюжетних гілок із варіантами вибору для користувачів, що читають історії;

Підтримка збереження історії у особистий кабінет для особистого використання та публікації історій (інтерактивні вебсторінки);

Можливість додавання інтерактивних елементів, таких як опитування, тести, рішення на основі вибору;

Система управління правами доступу для різних учасників проекту (користувачі, модератори, адміністратори).

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 |      |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 | 23   |

### **Вимоги персоналу веб-додатку (модератор):**

Наявність інструментів модерації для перевірки, цензурування та схвалення опублікованих історій.

Наявність інструментів для схвалення історії на публікацію, або несхвалення з можливістю написати причину відмови.

### **Вимоги персоналу веб-додатку (адміністратор):**

Наявність інструментів для отримання даних про користувачів та історії, які потребують модерації.

Наявність інструментів для зміни ролі користувача у застосунку на модератора або відміни цього статусу.

Наявність інструментів для назначення модератор, який буде проводити перевірку історії на відповідність правилам застосунку.

### **Характеристика об'єкта комп'ютеризації**

Користувач застосунку матиме можливість створювати, редагувати та переглядати мультимедійні історії. Інтерфейс додатку буде простим і зручним, дозволяючи легко додавати та організовувати контент, створювати сюжетні гілки та впроваджувати інтерактивні елементи.

Користувач буде мати можливість зберігати свої історії в архіві без подальшої публікації, або відправити запит на публікацію на сторінках застосунку.

Користувач буде мати можливість оцінювати історії та відмічати їх як ті, що сподобалися, залишати коментарі та оцінювати коментарі інших користувачів, шляхом лайків та дизлайків.

### **Функціональні вимоги:**

Авторизація у системі - користувачі повинні мати можливість реєструватися, створювати профілі, зберігати свої історії у профілі до та після публікації.

Редагування контенту - система повинна надавати інструменти для редагування тексту, зображень, аудіо та відео, а також створення інтерактивних елементів.

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 |      |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 | 24   |

Збереження та управління даними - історії та контент мають зберігатися в базі даних з можливістю їх редагування і публікації.

Мультимедійна підтримка - застосунок має підтримувати різні формати медіа для інтеграції в історії.

Взаємодія з користувачами - можливість читачам взаємодіяти з історіями, обираючи варіанти розвитку сюжету або залишаючи коментарі.

Оцінювання історій - застосунок повинен давати можливість користувачу оцінити історію по шкалі від 0 до 10, поставити чорну мітку(дизлайк). Також користувач повинен мати змогу змінити свої оцінки до конкретної історії.

Функціонал коментарів – застосунок повинен надавати користувачу залишати свою рецензію та коментарі, а також мати можливість оцінити коментарі інших користувачів. Також користувач повинен мати змогу видаляти або редагувати коментарі, та змінювати та знімати свої оцінки з коментарів інших користувачів.

#### **Нефункціональні вимоги:**

**Зручність користування** - час, необхідний для навчання нових користувачів, не повинен перевищувати 2-3 години; відгук інтерфейсу для основних операцій має бути меншим ніж 10 секунд.

**Надійність** - доступність додатку повинна становити не менше 95% часу роботи; середній час безвідмовної роботи — 30 днів.

**Продуктивність** - система повинна підтримувати одночасно не менше 100 активних користувачів.

**Експлуатація** - система має бути масштабованою, щоб забезпечити безперебійне функціонування при зростанні кількості користувачів.

Аналіз вимог користувачів надав підстави для визначення основних варіантів використання додатку для створення мультимедійних історій.

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 25   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |

На рис. 2.1 наведено діаграму використання програмної системи.

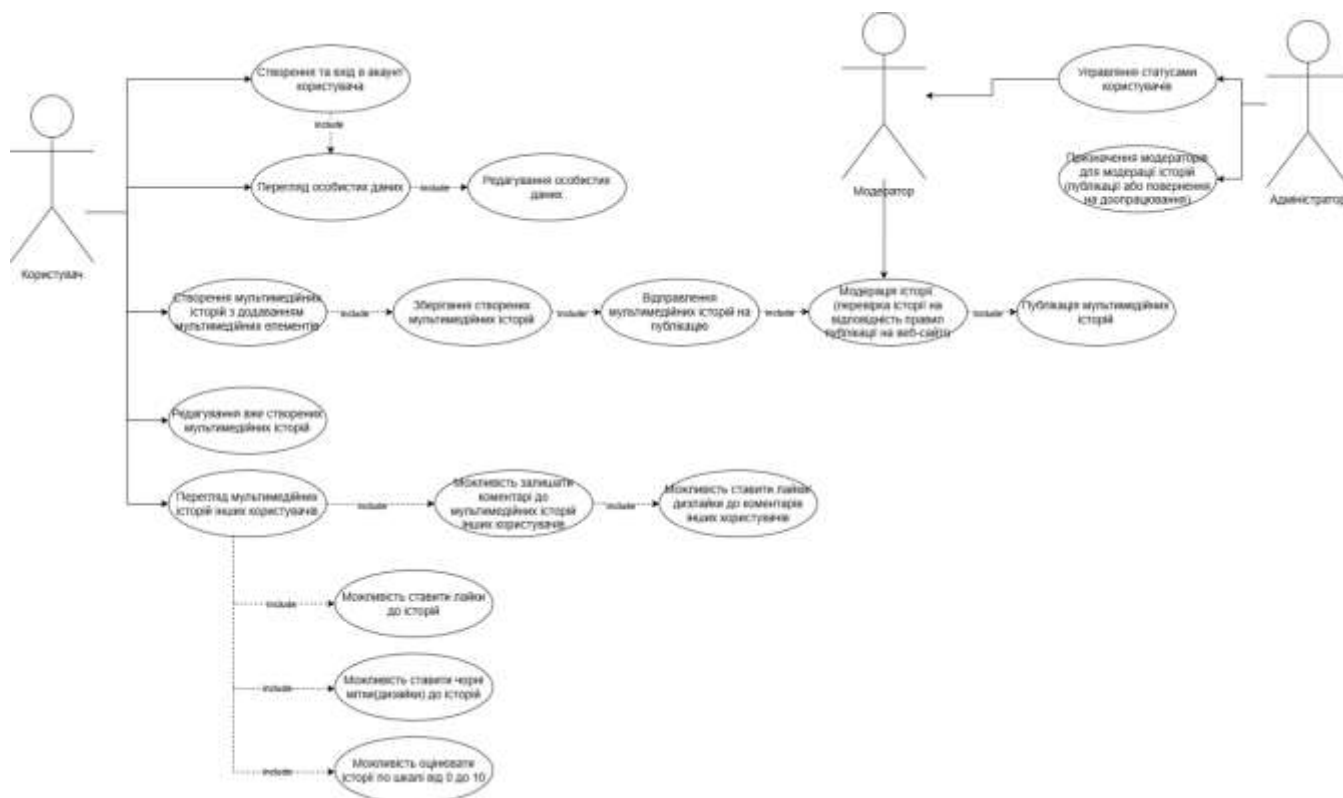


Рис. 2.1. Діаграма варіантів використання додатку для створення мультимедійних історій

Визначивши функціональність системи, ми можемо перейти до оформлення основних варіанті використання системи, які виглядають так:

1. Користувач створює нову історію;
2. Користувач переглядає історію;
3. Користувач(автор історії) редагує існуючу історію;
4. Користувач залишає відгук на історію;
5. Користувач створює акаунт.

І першим, що повинен зробити користувач, це зареєструватися у нашому застосунку, опис цього процесу представлений у таблиці 2.1.

Таблиця 2.1

#### Опис прецеденту «Користувач створює акаунт»

|                             |                                                                    |
|-----------------------------|--------------------------------------------------------------------|
| Назва варіанту використання | Користувач створює акаунт                                          |
| Основні актори              | Користувач/адміністратор/модератор                                 |
|                             | Коли користувач відкриває форму реєстрації, система її завантажує. |

|  |                                                                                                                                                                                                                                                                                                |
|--|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Після введення імені, логіна, електронної пошти та пароля, система перевіряє правильність даних. У разі помилок запитує їх виправлення. Якщо все вірно — створює обліковий запис у базі даних і підтверджує успішну реєстрацію. Далі система очікує подальших дій, наприклад, входу в систему. |
|--|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Для кращого розуміння цього процесу, нижче, на рис. 2.2., представлена діаграма активності.

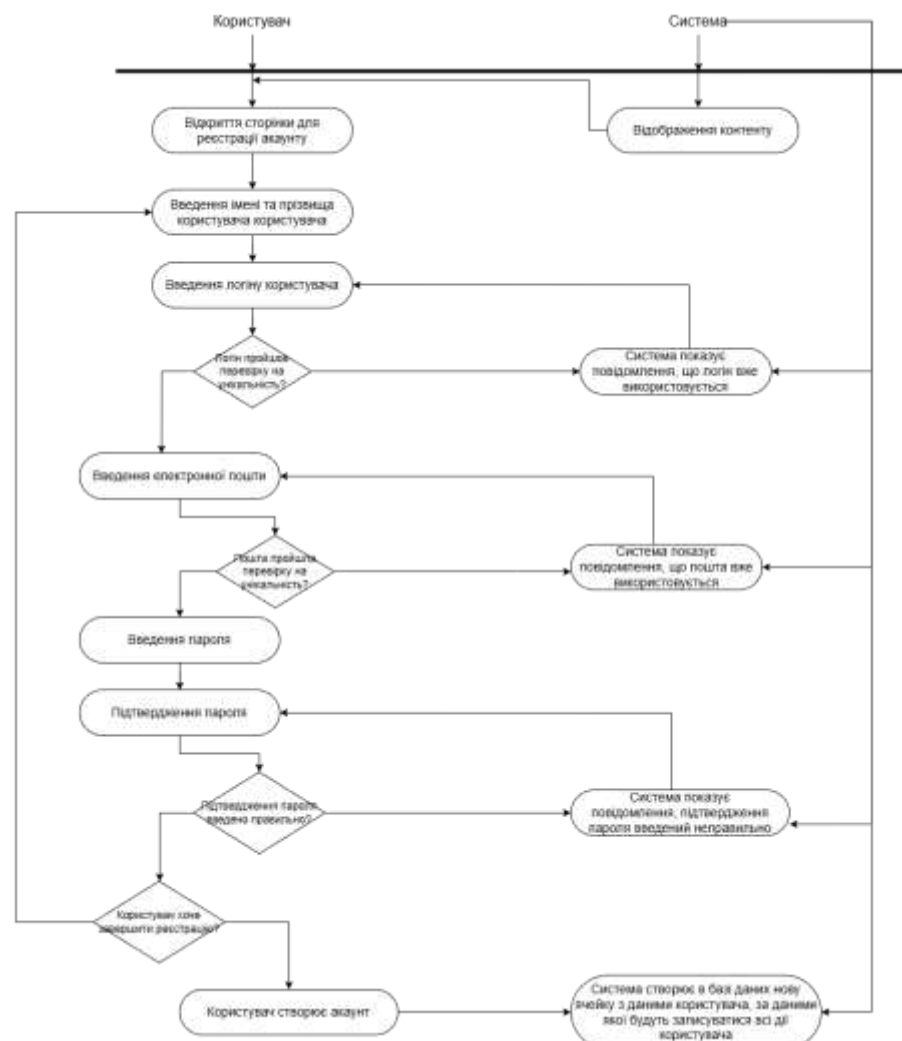


Рис. 2.2. Діаграма активності "Створення акаунта користувача"

Наступним, що скоріше за все, почне виконувати користувач, буде перегляд історій наявних у застосунку. Ця діаграма представлена нижче на рис. 2.3. та рис. 2.4. Опис цього процесу представлений у таблиці 2.2.

Таблиця 2.2

Опис прецеденту «Користувач переглядає історію»

|                             |                                                                                                                                                                                                                                                                                                                                                |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Назва варіанту використання | Користувач переглядає історію                                                                                                                                                                                                                                                                                                                  |
| Основні актори              | Користувач                                                                                                                                                                                                                                                                                                                                     |
| Короткий опис               | Користувач відкриває додаток, де система показує доступні історії. Після вибору історії завантажується мультимедійний контент і починається показ першої сцени. Користувач взаємодіє з інтерактивними елементами, а система відтворює контент згідно з вибором. Після завершення або закриття історії система повертається в режим очікування. |

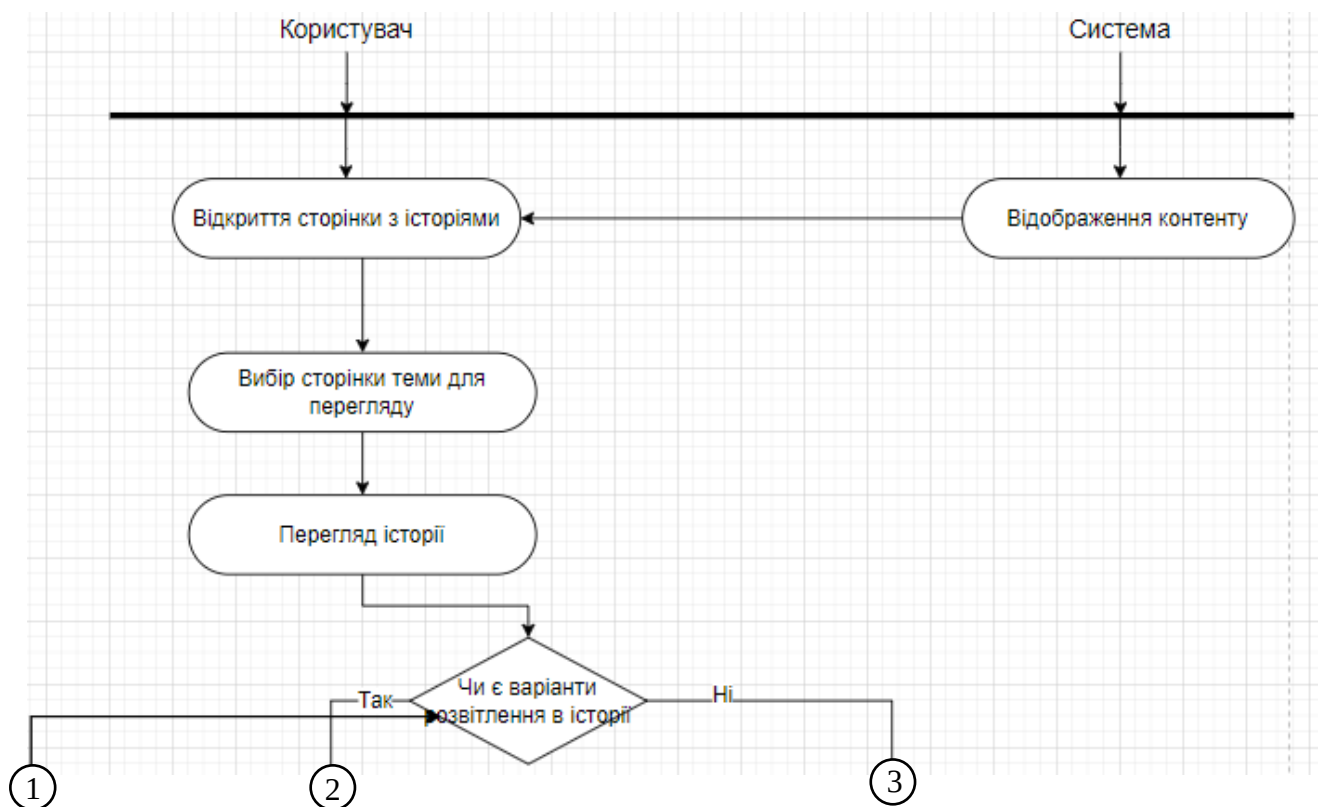


Рис. 2.3. Діаграма активності "Користувач переглядає історію"

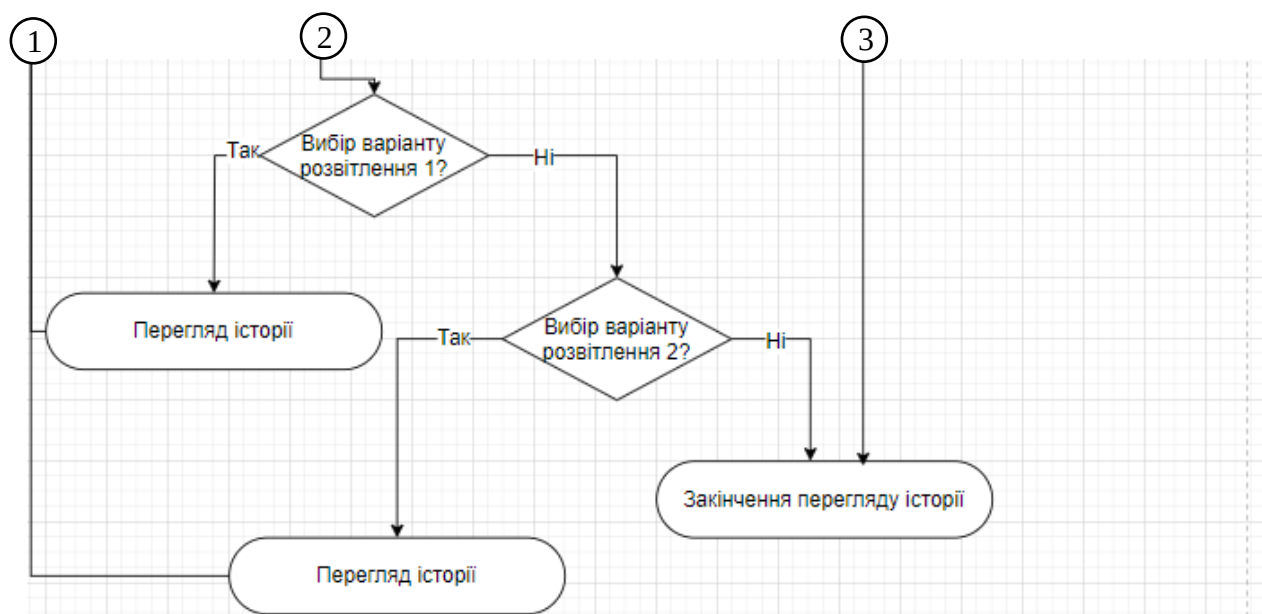


Рис. 2.4. Продовження діаграми активності "Користувач переглядає історію"

Далі розглянемо можливість створення власної історії користувачем. Ця діаграма представлена нижче на рис. 2.5. Опис цього процесу представлений у таблиці 2.3.

Таблиця 2.3

Опис прецеденту «Користувач переглядає історію»

|                    |          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|--------------------|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Назва використання | варіанту | Користувач створює нову історію                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Основні актори     |          | Користувач                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Короткий опис      |          | Користувач відкриває додаток для створення інтерактивних мультимедійних історій, а система очікує вибору дії. Після вибору опції "Створити нову історію" система створює проект і переходить у режим редагування. Користувач додає мультимедійні файли, які система завантажує і зберігає. Завершивши редагування, користувач зберігає історію, яку система зберігає на сервері. Після публікації, яка включає модерацію, історія стає доступною для інших користувачів. Закінчивши створення, система готова до нового проекту або редагування існуючого. |

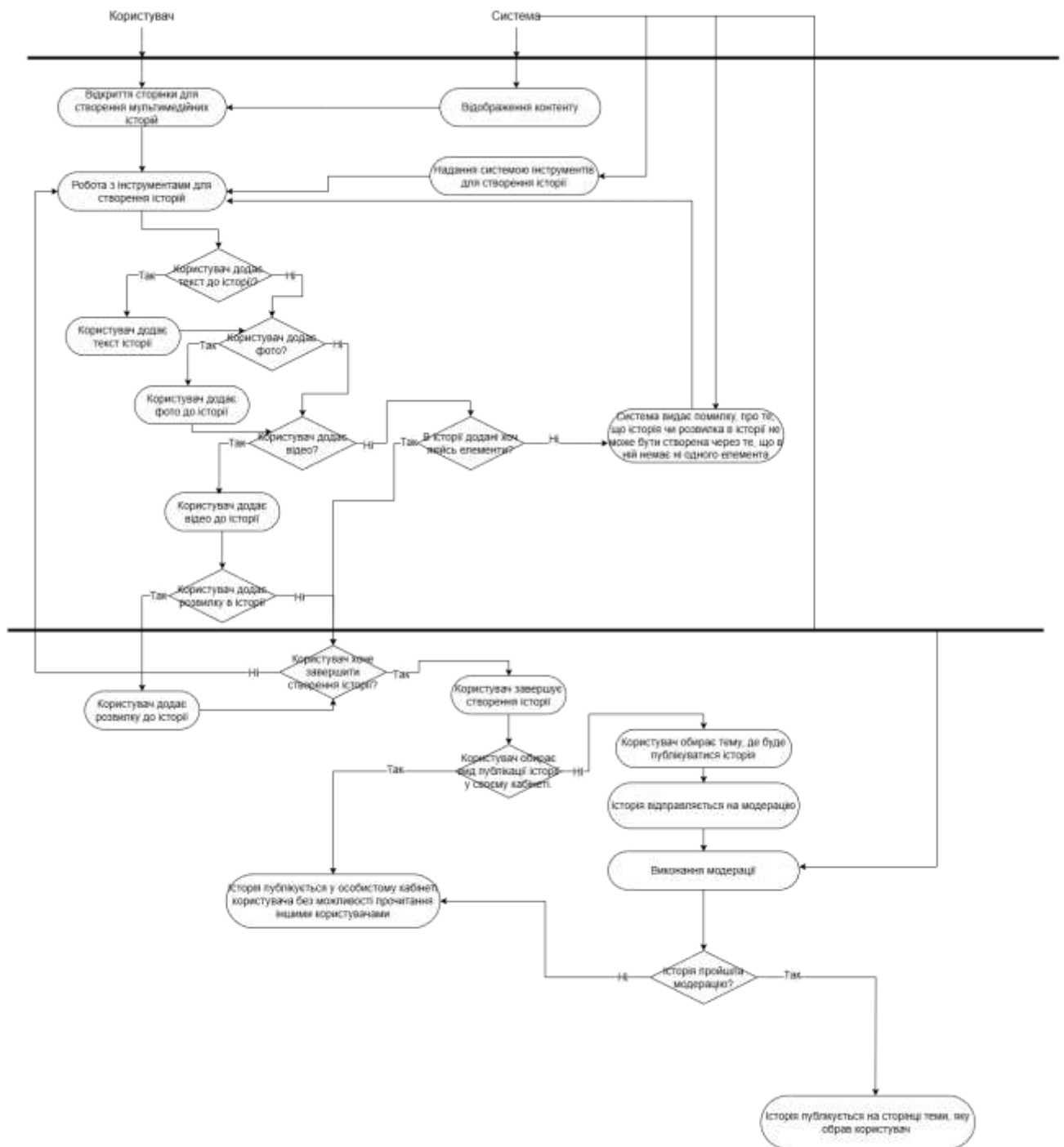


Рис. 2.5. Діаграма активності "Користувач створює нову історію"

Наступним розглянемо, як користувач міг би змінити існуючу історію, яку створив. Ця діаграма представлена нижче на рис. 2.6. Опис цього процесу представлений у таблиці 2.4.

## Опис прецеденту «Користувач редагує історію»

|                    |          |                                                                                                                                                                                                                                                                                                                                                                |
|--------------------|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Назва використання | варіанту | Користувач редагує історію                                                                                                                                                                                                                                                                                                                                     |
| Основні актори     |          | Користувач                                                                                                                                                                                                                                                                                                                                                     |
| Короткий опис      |          | Користувач обирає історію зі списку — система завантажує сцени та мультимедіа і переходить у режим редагування. Під час редагування можна додавати елементи, сцени та налаштовувати інтерактивність, зміни тимчасово зберігаються. Після натискання "Зберегти" система перевіряє зміни, зберігає їх і пропонує публікацію, а потім повертає до списку історій. |

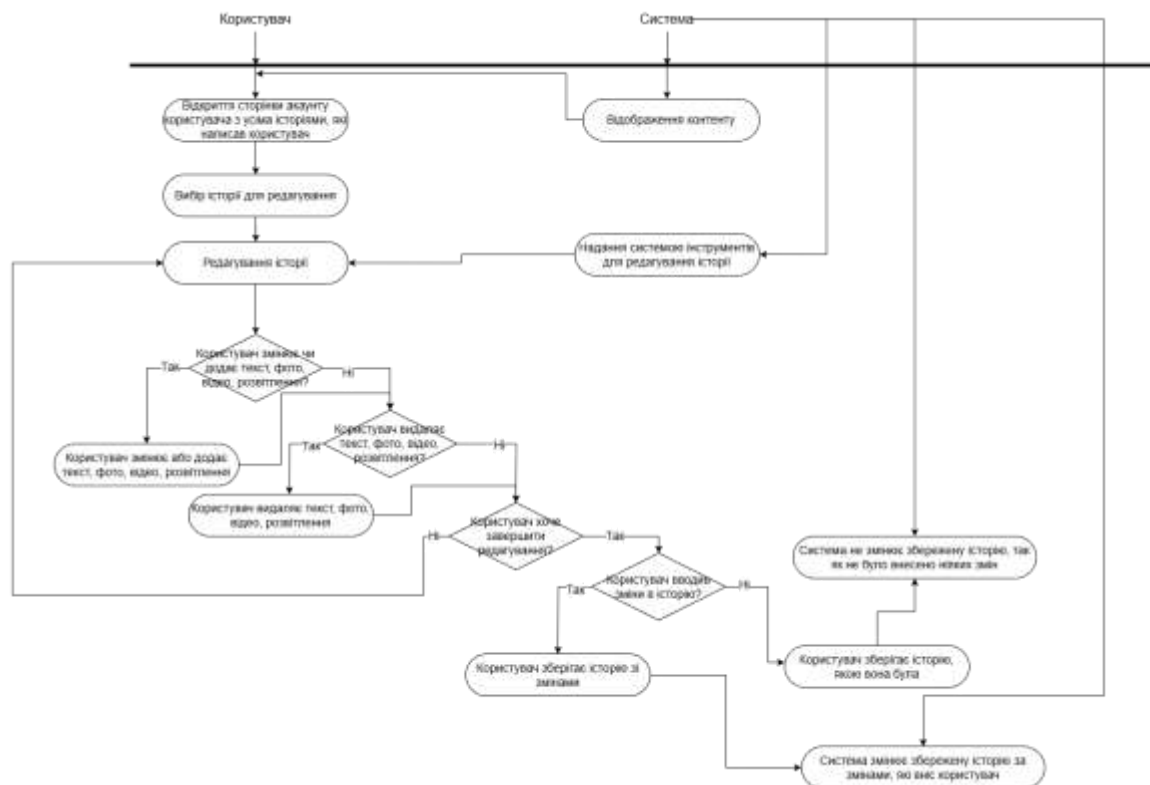


Рис. 2.6. Діаграма активності "Користувач змінює історію"

Останнім розглянемо додавання коментаря до історії. Ця діаграма представлена нижче на рис. 2.7. Опис цього процесу представлений у таблиці 2.5.

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 31   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |

## Опис прецеденту «Користувач додає коментар»

|                             |                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Назва варіанту використання | Користувач додає коментар                                                                                                                                                                                                                                                                                                                                                                                                              |
| Основні актори              | Користувач                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Короткий опис               | Система очікує, поки користувач вибере історію для перегляду чи коментування, після чого завантажує її дані. Натиснувши кнопку "Додати коментар", користувач отримує форму для введення текстового чи відео коментаря, який система перевіряє та обробляє. Після збереження коментар додається до історії, а користувач отримує підтвердження. Завершивши дію, система повертається в режим перегляду або очікування наступних кроків. |

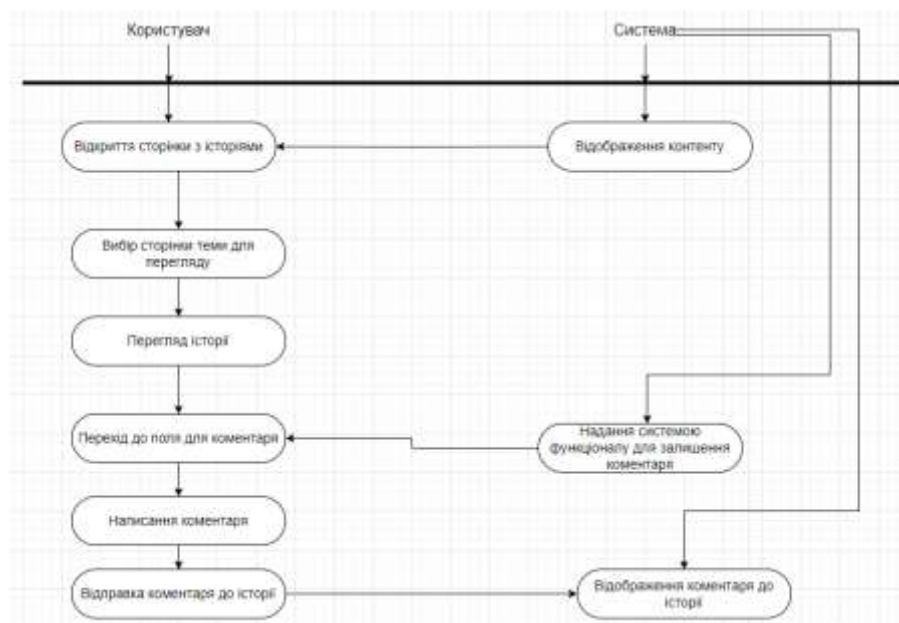


Рис. 2.7. Діаграма активності "Користувач додає коментар"

Визначивши функціональні вимоги та сценарії використання додатку, можливо виділити наступні сутності, що забезпечать програмну реалізацію додатку

Розпочнемо зі створення діаграму класів, визначивши основні класи та їх взаємозв'язки в системі.

Основними класами є: Користувач (User) - цей клас необхідний для управління додатком. Він містить основну інформацію про кожного користувача, а також функції для аутентифікації та управління взаємодією користувача з системою; Історія (Story) - цей клас представляє основний контент додатка. Історії є центральним елементом, навколо якого будується взаємодія користувачів. Користувачі можуть створювати, редагувати та переглядати різні історії, що робить цей клас критично важливим для функціонування додатка; Коментар (Comment) - коментарі дозволяють користувачам взаємодіяти з контентом та один з одним. Це підвищує залученість та інтерактивність платформи, надаючи можливість залишати відгуки та рецензії, обговорювати сцени або ділитися враженнями про прочитане; Сцена (Scene) - сцени є складовими частинами історій. Вони дозволяють структурувати історії на більш дрібні частини, кожна з яких може містити текст, медіа або інші елементи. Це забезпечує зручність та структурованість у створенні складних та багат шарових історій; Вибір (Choice) - вибори додають інтерактивності до історій, дозволяючи користувачам приймати рішення, які впливають на подальший розвиток подій. Це робить історії більш динамічними та залученими, оскільки користувачі можуть впливати на перебіг подій у історії; Медіа (Media) - використання медіа (зображення, аудіо, відео) підвищує залученість користувачів та додає більшу динамічність та допомагає автору повністю розкрити свої задумки при створенні історій. Це важливо для створення повноцінного та багатого контенту; Аудіо (Audio), Відео (Video), Фото (Photo) - класи для різних типів медіа, які потрібні для завантаження потрібного елемента або його видалення.

Після визначення класів ідентифікуємо такі атрибути та методи класів:

Клас Користувач (User), буде мати такі атрибути, як: id, що буде відповідати за унікальний ідентифікатор користувача; role, як роль користувача (адміністратор, користувач, модератор), за замовчуванням при створенні акаунта вона буде «Користувач»; name що буде відповідати за ім'я та фамілію користувача; login, email, password – облікові дані, в яких login та

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 33   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |

email будуть унікальні; avatar що буде відповідати за зображення профілю, що додасть ще більшої індивідуальності для кожного користувача застосунку; reset\_token, reset\_token\_exp, які відповідають за збереження токенів для відновлення пароля та їх часу використання, що підвищить безпеку при зміні пароля.

Також, клас користувач буде мати такі методи, як: enter(), що буде відповідати за авторизацію користувача, а destroy() за вихід з системи. identify() – перевірка особи користувача та реєстрація у системі.

Далі клас Історія (Story), буде мати такі атрибути, як: id, history\_title – ідентифікатор та назва історії; author – де буде отримуватися і зберігатися логін користувача, який є автором історії; folder\_path – де буде зберігатися шлях до файлів історії; created\_at – дата створення історії; tags – жанри історії, які будуть використовуватися для подальшого фільтрування; data – зміст(json) історії; history\_cover – обкладинка історії, яка буде використовуватися у картці історії, як її афіша; status – статус історії, які допоможуть у тому, щоб визначати її стан (збережена в архів, опублікована, відмовлена, очікує публікації, видалена).

Також, клас буде мати такі методи, як: enter() – відкриття історії редагування або створення; show() – відображення історії, її прочитання; decision() відображення та запис рішень у історії, які зробив користувач, uploadHistory() – збереження історії у базі даних та її файлів у додатку; exit() – вихід з історії.

Далі клас Сцена (Scene), буде мати такі атрибути, як: id, history\_title – ідентифікатор та назва сцени; dialogs – діалоги сцени; story\_block – блок, у якому будуть знаходитися діалогі і медіа файли; media\_list – список медіа, а саме фото, відео і аудіо.

Також, клас буде мати такі методи, як: enter() – запуск сцени; show() – показ сцени; decision() – вибір у сцені, які будуть створені та їх шляхи, куди вони будуть вести(на наступну сцену чи продовження поточної); exit() – завершення сцени та перехід на нову чи просто завершення історії.

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 |      |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 | 34   |

Далі клас Вибір (Choice), який буде мати такі атрибути: id, choice\_description – ідентифікатор та опис вибору, який може зробити користувач; scene\_selector\_buttons – кнопки для вибору розвилки, яка може перенести на нову сцену або на продовження поточної; NextScene – перехід на обрану сцену.

Також, клас буде мати такі методи, як: enter() – функція активації вибору у поточній сцені; show() – відображення можливих варіантів вибору; decision() – функція для запису вибору користувача; exit() – завершення вибору.

Далі клас Медіа (Media), буде мати такі атрибути, як: id, Type – унікальний ідентифікатор та назва типу медіа; FilePath – шлях до файлу окремого медіа файлу; Scene – ідентифікатор сцени, до якої належить окремий медіа файл.

Також, клас буде мати такі методи, як: show() – відображення медіа файлів; save() – збереження змін медіа файлів; files() – обробка медіа файлів при їх завантаженні або видаленні.

Далі клас Коментар (Comment), буде мати такі атрибути, як: id, comment\_text – ідентифікатор та текст коментаря; created\_at – дата створення коментаря; user\_id – ідентифікатор автора коментаря; history\_id – ідентифікатор історії, до якої прив'язаний коментар.

Також, клас буде мати такі методи, як: write() – створення коментаря; show() – показ коментаря; delete() – видалення коментаря; save() – збереження коментаря та змін до нього.

Далі клас Лайки/дизлайки до коментарів, буде мати такі атрибути, як: id, comment\_id, user\_id – ідентифікатори коментаря, юзера та самого лайку/дизлайку до коментаря; is\_like – чи поставлено лайк чи дизлайк; created\_at – дата реакції.

Також, клас буде мати такі методи, як: isLike() – перевірка статусу реакції; deleteReaction() – видалення лайку/дизлайку.

Далі клас Лайки до історії, буде мати такі атрибути, як: id, user\_id, story\_id – ідентифікатори до історії та користувача який поставив свою реакцію.

Також, клас буде мати такі методи, як: setReaction() – встановлення лайку; deleteReaction() – видалення лайку.

Далі клас Оцінювання історії, буде мати такі атрибути, як: id, user\_id, story\_id – ідентифікатори до історії та користувача який поставив свою оцінку; rating – оцінка історії; created\_at – дата виставлення оцінки.

Також, клас буде мати такий метод, як: setRating() – встановлення своєї оцінки.

Далі клас Чорні мітки до історії, буде мати такі атрибути, як: id, user\_id, story\_id – ідентифікатори до історії та користувача який поставив свою реакцію; created\_at – дата виставлення реакції.

Також, клас буде мати такі методи, як: setBlackMark() – додавання чорної мітки; deleteBlackMark() – видалення чорної мітки.

На рисунку 2.8., 2.9., наведено Діаграму класів, яка використовуються для його роботи.

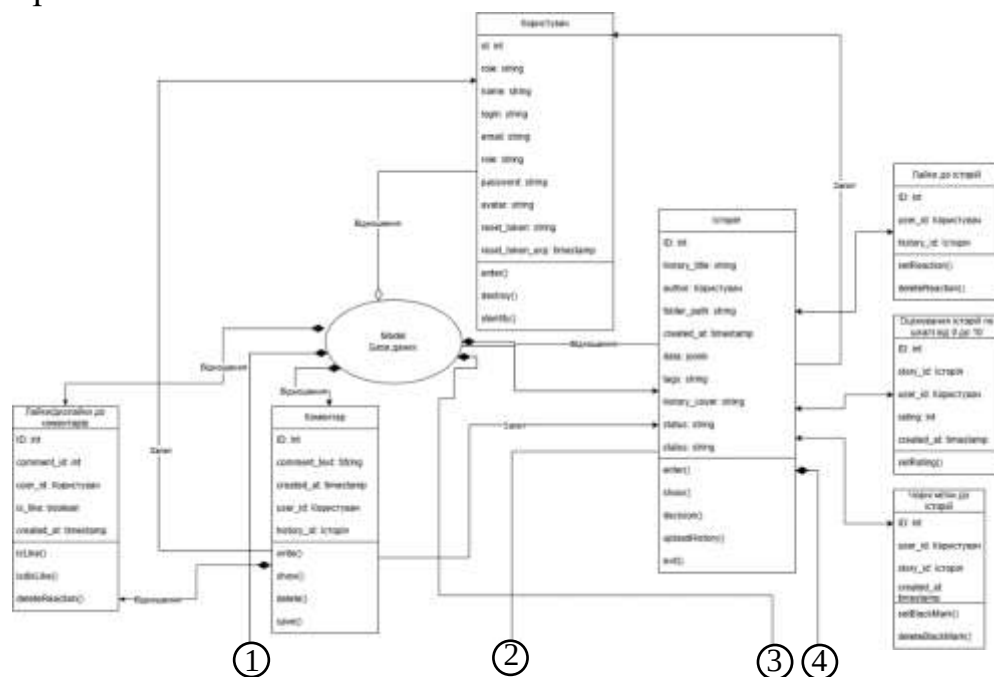


Рис. 2.8. Діаграма класів для застосунку



рішення, які впливають на розвиток історії, з'єднуючи поточну сцену з наступною.

Медіа включає різні типи файлів, зберігає шлях до файлу і пов'язані з певною сценою. Аудіо, відео та фото є спеціалізованими медіафайлами, що містять додаткову інформацію, таку як тривалість і формат.

Після створення діаграми, було проаналізовано поточну діаграму, і для покращення читабельності діаграми були додані паттерни.

Для цього були обрані такі паттерни:

1. Factory паттерн (Паттерн виклику інтерфейсу)
2. Singleton паттерн (Паттерн на запит до відтворення нової сцени)
3. Builder паттерн (Паттерн для створення полотна (сторінка сайту))
4. Prototype паттерн (Паттерн відтворення нової сцени)
5. Composite паттерн (Паттерн виклику коментарів)

Factory паттерн – був використаний для реалізації компоненту інтерфейсу, а саме сторінки історій. Також для завершення цього компоненту, був використаний Builder паттерн, який допоміг показати, що сторінка складається з медіа елементів, які є невід'ємною частиною історії.

Singleton паттерн був використаний для запиту на відтворення нової сцени, після виконання вибору, у якому напрямі буде рухатися історія.

Prototype паттерн був використаний для створення(відтворення) нової сцени після вибору напрямку історії.

Composite паттерн був використаний для відтворення коментарів, бо цей паттерн завжди використовується, коли потрібно обробляти групу об'єктів, як один об'єкт, що нам і потрібно у класі коментарів.

Після визначення, чому саме ці паттерни були використані, можна перейти до самої діаграми, яка представлена на рис. 2.10.

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 38   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |

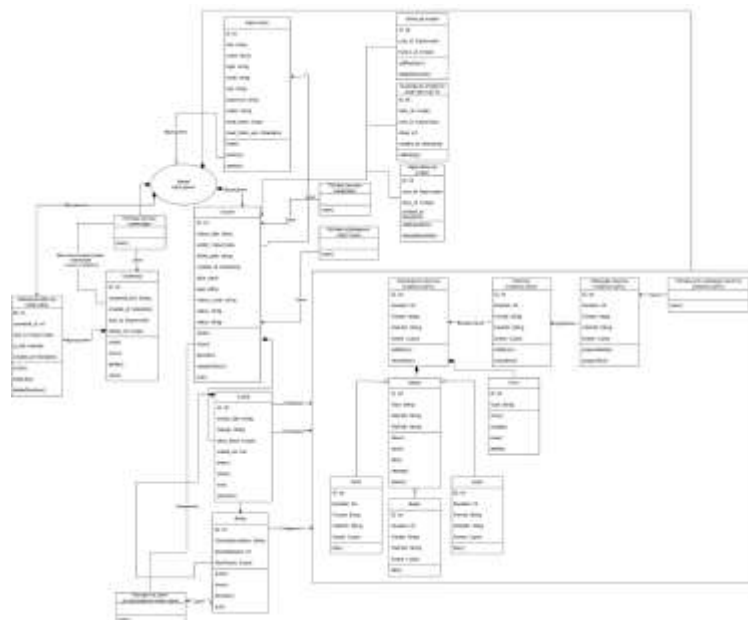


Рис. 2.10. Діаграма класів з використанням паттернів

## 2.2. Розробка бази даних системи

У додатку для інформації та шляхів до файлів застосовується PostgreSQL, що є системою управління базами даних, що працює за реляційною моделлю.

База даних буде використовуватися для різноманітних потреб, таких як, збереження даних користувачів, збереження даних історій та збереження даних про дії користувачів у застосунку.

Структура бази даних застосунку включає вісім таблиць:

1. Users – це таблиця, що використовується для збереження даних користувачів;
2. Story\_ratings – це таблиця, що використовується для збереження даних про оцінювання історії від 0 до 10 користувачем;
3. Story\_blackmarks – це таблиця, що використовується для збереження даних про виставлені чорні мітки користувачем.
4. Stories – ця таблиця використовується для збереження інформації про історії та їх дані, а саме json та шляхи до її файлів.
5. Historylikes – ця таблиця використовується для збереження лайків користувачів до конкретної історії.

6. For\_moderation – ця таблиця використовується для збереження історій, які були відправлені на модерацію для конкретного модератора.

7. Comments – ця таблиця використовується для збереження даних коментаря, а саме його автора та текст коментаря.

8. Coment\_likes – ця таблиця використовується для збереження лайків чи дизлайків користувачів до конкретного коментаря.

Опис кожної з таблиць бази даних можна побачити нижче, у таблицях 2.6 – 2.13.

Таблиця 2.6

Опис таблиці Users

| Поле                | Тип                      | Призначення                              |
|---------------------|--------------------------|------------------------------------------|
| id                  | intager                  | Ідентифікатор користувача                |
| name                | character varying        | Ім'я та прізвище користувача             |
| login               | character varying        | Логін користувача                        |
| email               | character varying        | Електронна пошта користувача             |
| role                | character varying        | Роль користувача                         |
| password            | text                     | Пароль користувача                       |
| avatar              | text                     | Аватар користувача                       |
| reset_token         | text                     | Токен для скидання пароля                |
| reset_token_expires | timestamp with time zone | Час дійсності токена для скидання пароля |

Таблиця 2.7

Опис таблиці Story\_ratings

| Поле       | Тип                         | Призначення               |
|------------|-----------------------------|---------------------------|
| id         | intager                     | Ідентифікатор оцінки      |
| user_id    | intager                     | Ідентифікатор користувача |
| story_id   | intager                     | Ідентифікатор історії     |
| rating     | intager                     | Рейтинг від 0 до 10       |
| created_at | timestamp without time zone | Дата оцінювання           |

Опис таблиці Story\_blackmarks

| Поле       | Тип                         | Призначення                |
|------------|-----------------------------|----------------------------|
| id         | intager                     | Ідентифікатор чорної мітки |
| user_id    | intager                     | Ідентифікатор користувача  |
| story_id   | intager                     | Ідентифікатор історії      |
| created_at | timestamp without time zone | Дата оцінювання            |

Таблиця 2.9

Опис таблиці Stories

| Поле          | Тип                      | Призначення            |
|---------------|--------------------------|------------------------|
| id            | intager                  | Ідентифікатор історії  |
| history_title | text                     | Назва історії          |
| author        | text                     | Логін авторів історії  |
| folder_path   | text                     | Шлях до файлів історії |
| created_at    | timestamp with time zone | Дата створення історії |
| data          | jsonb                    | Json дані історії      |
| tags          | text                     | Жанри історії          |
| history_cover | text                     | Обкладинка до історії  |
| status        | text                     | Статус історії         |
| description   | text                     | Опис історії           |

Таблиця 2.10

Опис таблиці Historylikes

| Поле       | Тип     | Призначення               |
|------------|---------|---------------------------|
| id         | intager | Ідентифікатор лайку       |
| user_id    | intager | Ідентифікатор користувача |
| history_id | intager | Ідентифікатор історії     |

Таблиця 2.11

Опис таблиці For\_moderation

| Поле             | Тип                         | Призначення                            |
|------------------|-----------------------------|----------------------------------------|
| story_id         | intager                     | Ідентифікатор історії                  |
| moderator_id     | intager                     | Ідентифікатор модератора               |
| created_at       | timestamp without time zone | Дата створення запиту на модерацію     |
| rejection_reason | text                        | Причина відмови від публікації історії |
| status           | text                        | Статус історії після модерації         |

Таблиця 2.12

Опис таблиці Comments

| Поле         | Тип                         | Призначення               |
|--------------|-----------------------------|---------------------------|
| id           | intager                     | Ідентифікатор коментаря   |
| history_id   | intager                     | Ідентифікатор історії     |
| user_id      | intager                     | Ідентифікатор користувача |
| comment_text | text                        | Текст коментаря           |
| created_at   | timestamp without time zone | Дата створення коментаря  |

Таблиця 2.13

Опис таблиці Coment\_likes

| Поле       | Тип                         | Призначення                          |
|------------|-----------------------------|--------------------------------------|
| id         | intager                     | Ідентифікатор реакції                |
| comment_id | intager                     | Ідентифікатор коментаря              |
| user_id    | intager                     | Ідентифікатор користувача            |
| is_like    | boolean                     | Тип реакції(true-like/false-dislike) |
| created_at | timestamp without time zone | Дата реації                          |

Кожна база даних буде використовуватися у певних ситуаціях, таких як запис та відображення коментарів, збереження і відображення історії, збереження та відображення реакцій, лайків та чорних міток до історії, збереження та відображення лайків та дизлайків до коментарів, фільтрація з використанням статусів при модерації та їх зміна для подальшої публікації.

Також під час виконання роботи були виконані такі зв'язки між базами даних, які можна побачити нижче на рис. 2.11.

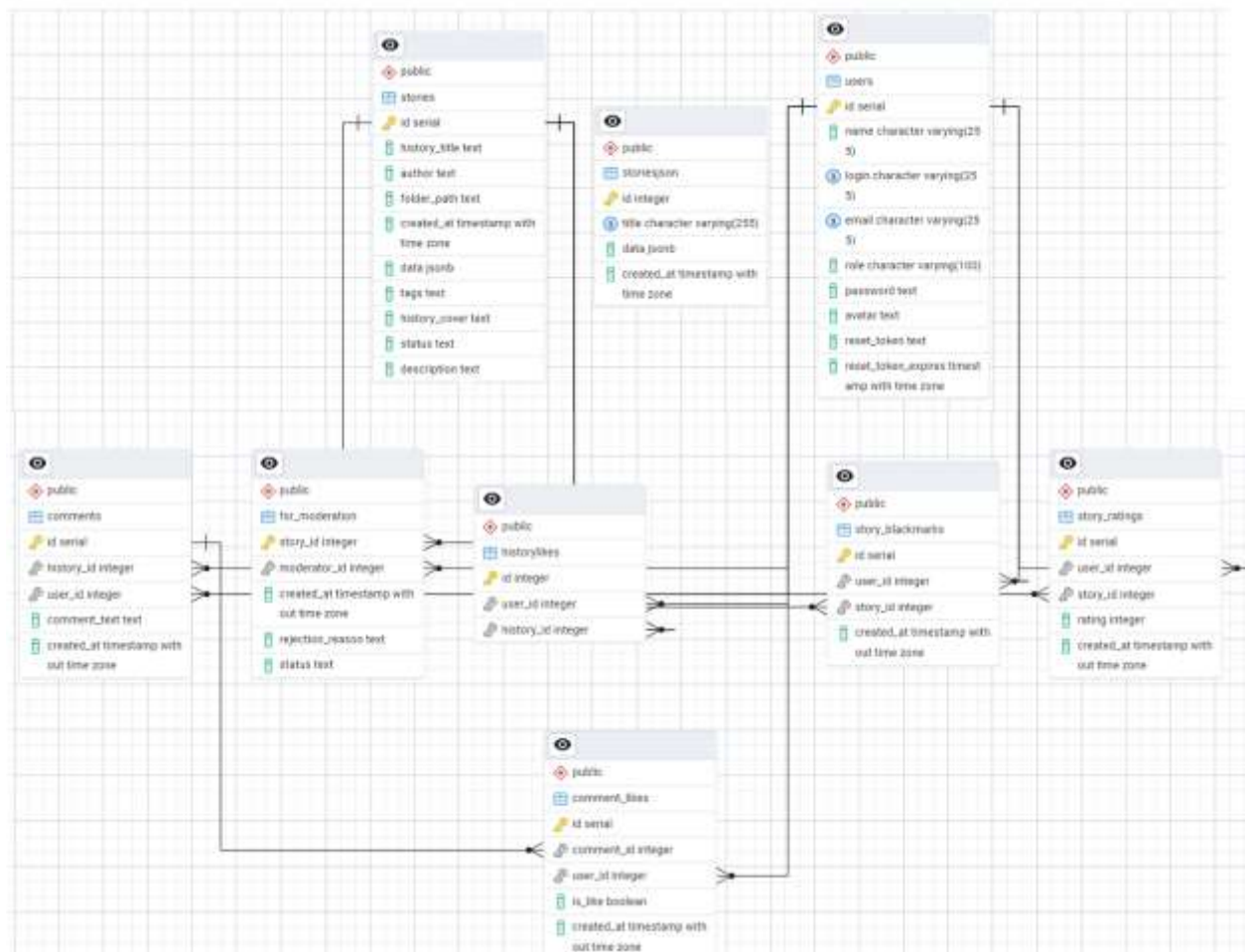


Рис. 2.11. Представлення зв'язків у таблицях

### 2.3. Проектування та реалізація алгоритмів роботи системи

Алгоритми роботи системи визначають основну логіку функціонування застосунку для створення мультимедійних історій. Вони забезпечують обробку дій користувача, збереження даних, навігацію між розділами історії, додавання медіафайлів, коментування, оцінювання коментарів та історії та модерацію контенту. Основна увага під час проектування приділялася

простоті, швидкодії та зручності інтеграції з базою даних PostgreSQL. Реалізація алгоритмів відбувалася з урахуванням потреб користувачів.

Тому була створена діаграма послідовності, де можна побачити всі алгоритми системи.

Користувач (User) має можливість входу в систему, реєстрації та управління профілем. Вони можуть створювати та редагувати історії, додавати, редагувати та видаляти власні коментарі, оцінювати історії шляхом виставлення оцінок від 0 до 10 та чорних міток та вподобайок, а також оцінювати коментарі інших шляхом виставлення лайку або дизлайку, переглядати особисті дані і так далі.

Особисті дані (Personal Data) представлені як функціональність зі зберігання інформації про користувача. Ця функція використовується класом Користувач для отримання та оновлення особистих даних, таких як логін, ім'я та прізвище, пароль, аватар та змінювати ці дані. Також користувач має доступ до особистого кабінету де зберігаються його опубліковані, збережені та оцінюючі публікації історії.

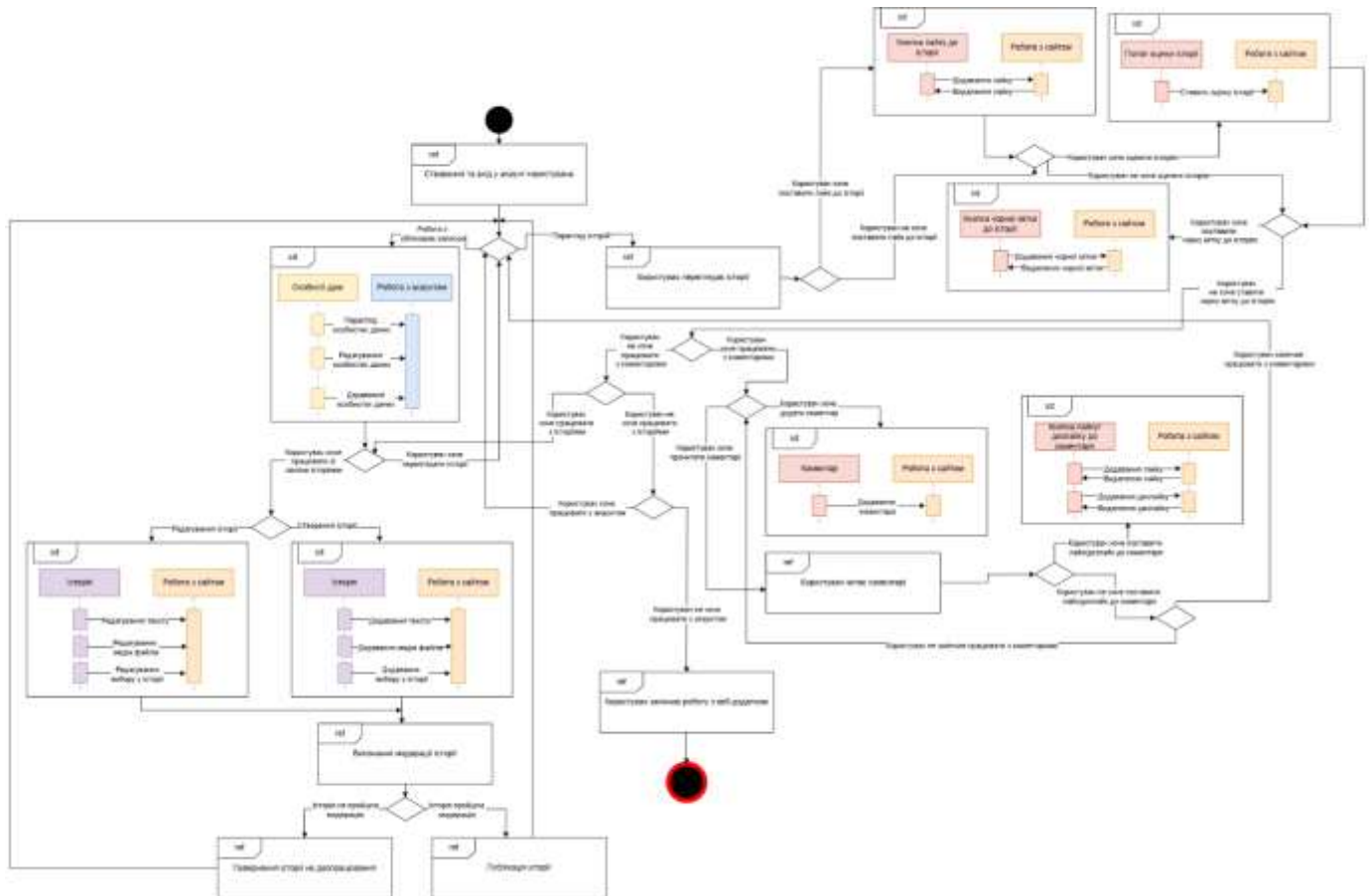
Історія (Story) реалізована як функціональність для створення, редагування, зберігання і публікації інтерактивних історій, які можуть містити текст, вибір вітки історії, фото, відео, аудіо та інші медіа. Вона взаємодіє з Користувачем та Модерацією історії. Користувач може створювати і редагувати історії, додавати медіа-файли, а модерація перевіряє та затверджує їх, або відправляє на доопрацювання описавши причину відмови.

Коментар (Comment) реалізує можливість додавання, редагування і видалення коментарів до історій. Взаємодіє з класами Користувач і Історія. Користувач може залишати коментарі до історій, переглядати коментарі інших користувачів, та реагувати на них шляхом виставлення лайку або дизлайку.

Робота з сайтом (Website Interaction) представлена як функціональність, що забезпечує користувацький інтерфейс для взаємодії користувача з застосунком. Користувач використовує цей клас для навігації по сайту, виконання дій, перегляду різних сторінок і т.д.

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 44   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |

На основі цієї інформації, була побудована діаграма взаємодії класів. Вона представлена на рис. 2.12.



Тут ми бачимо взаємодію класів на основі базового сценарію користувача, який складається з базових активностей, а саме: створення історії, редагування історії, створення та авторизація акаунту, перегляд історії та прочитанні і написання коментарів.

## 2.4. Реалізація функціоналу застосунку для створення мультимедійних історій

Для початку того щоб розбирати реалізацію функціоналу застосунку для створення мультимедійних історій, спочатку буде розглянуто структуру проекту, яку можна побачити на рис. 2.13.

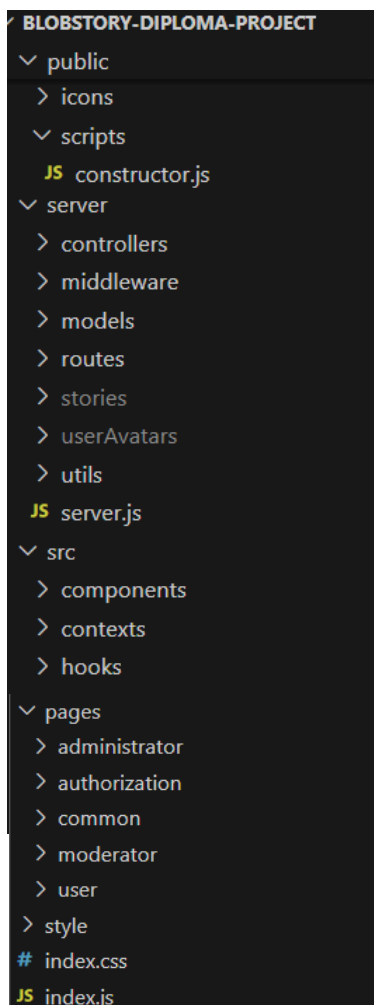


Рис. 2.13. Структура проекту

Ця структура дає нам можливість реалізувати наш проект, оскільки вона забезпечує чіткий поділ на етапи розробки та дозволяє ефективно організувати всі компоненти. Зокрема, розділення на фронтенд і бекенд дозволяє зосередитися на специфікаціях кожної частини системи, що сприяє оптимізації процесу розробки. Кожен з етапів — від створення користувацького інтерфейсу до інтеграції мультимедійних елементів — допомагає нам забезпечити високу якість та зручність у використанні кінцевого продукту, а також інтеграцію різноманітних медіа-ресурсів, що є ключовими для створення динамічних і захоплюючих історій.

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 46   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |

Після того, як ми ознайомилися з структурою проекту, ми можемо перейти до ознайомлення з кодовою реалізацією основних її частин, що потребують опису.

Першим для опису буде утиліта для підключення до бази даних, що є основою для нашого проекту.

```
const { Pool } = require('pg');  
const pool = new Pool({  
  user: 'admin',  
  host: 'localhost',  
  database: 'blobStoryDW',  
  password: '# protected',  
  port: 5432,  
});
```

```
module.exports = pool;
```

Тут ми можемо побачити, що була створена та використана база даних з назвою blobStoryDW, і також для зручності цей код був винесений в утиліту, щоб не вводити його кожного разу при його використанні.

Далі наступною частиною коду, що буде продемонстровано та описано, буде завантаження файлу історії на сервер. Для зручності, розділимо його на частини.

Ця частина коду, відповідає за отримання токена авторизації користувача, вона допомагає зафіксувати автора історії та у забезпеченні додаткової безпеки. Також ця частина коду потрібна для подальшої роботи функції, тому що перевіряється наявність токена і чи він не прострочений, якщо все відповідає дійсності, тоді подальші кроки функції виконуються.

```
function uploadFilesToServer(dataStr) {  
  const token = JSON.parse(localStorage.getItem('user_data'))?.userToken;  
  if (!token) {  
    modal_window("close");  
    toast("Error: User not authenticated.");  
    return;}  
}
```

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 47   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |

Ця частина коду відповідає за формування даних історії, які будуть збережені у базу, такі як назва, json, теги, статус та обкладинка

```
let title = 'untitled_project';
if (story_script?.parameters?.title) {
  title = story_script.parameters.title.en
  || story_script.parameters.title[story_script.parameters.languages?.[0]]
  || story_script.parameters.title;
title = String(title); }
const jsonData = {
  title, data: story_script,};
const formData = new FormData();
const tags = selectedTags.join(",");
formData.append('projectTitle', title);
formData.append('jsonData', JSON.stringify(story_script));
formData.append('tags', tags);
formData.append('status', currentStatus);
formData.append('description', storyDescription);
if (coverFile) { formData.append('cover', coverFile);}
```

Далі буде показана частина коду, яка формує та додає html та медіафайли, і у подальшому додаються лише ті медіа файли, які використовувалися у проекті.

```
const htmlBlob = new Blob([dataStr], { type: 'text/html' });
formData.append('files', htmlBlob, 'index.html');
for (let f = 0; f < project_files.length; f++) {
  if (
    project_files[f][2] !== "application/json" &&
    dataStr.includes(project_files[f][0])
  ) { const file = new Blob([load_files[f]], { type: project_files[f][2] });
    formData.append(
      'files', file,
```

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 48   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |

```

        project_files[f][2].includes('audio/mpeg') ? project_files[f][0] + '.mp3' :
project_files[f][0]
    ); }}

```

Остання частина коду, відповідає за відправлення запиту на сервер, а саме Дані надсилаються на сервер через POST-запит. Якщо запит виконується успішно, користувач отримує повідомлення про успішне збереження проєкту. Якщо виникає помилка, користувач бачить повідомлення про помилку при завантаженні.

Загалом, ця функція управляє процесом завантаження, перевіряє автентифікацію, формує необхідні дані та обробляє відповіді сервера, а саме у контроллер uploadController, де відбувається подальше створення папки.

Далі наступною частиною коду, що буде продемонстровано та описано, буде middleware, який відповідає за аутентифікацію користувача:

```

const jwt = require('jsonwebtoken'); const createError = require('../utils/appError');
const pool = require('../utils/db');
const authMiddleware = async (req, res, next) => {
  const authHeader = req.headers.authorization;
  if (!authHeader || !authHeader.startsWith('Bearer ')) {
    return next(new createError('Not authorized, no token', 401));
  }
  const token = authHeader.split(' ')[1];
  try { const decoded = jwt.verify(token, '# protected ');
    const { rows } = await pool.query('SELECT login FROM users WHERE id =
$1', [decoded._id]);
    if (rows.length === 0) { return next(new createError('User not found', 401)); }
    req.user = {
      _id: decoded._id, login: rows[0].login, }; next();
  } catch (err) {return next(new createError('Invalid token', 401));}};
module.exports = authMiddleware;

```

Цей код працює таким чином, він використовує JWT, отримує заголовок авторизації, перевіряє його на наявність та правильність формату, витягує

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 49   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |

токен і перевіряє його валідність. Після цього виконується запит до бази даних, щоб знайти користувача за ID, витягнутим із токена. Якщо користувач існує, його дані додаються до req.user, і запит продовжує обробку. У разі помилок, таких як відсутність токена або його некоректність, обробка зупиняється з відповідним повідомленням.

Тому, цей код забезпечує захист маршрутів, дозволяючи доступ лише автентифікованим користувачам.

І останнім, але не менш важливим, буде розглянуто код контролера, який відповідає за базові дії, які будуть виконуватися з історіями. Розділемо його на частини.

```
exports.getAllHistories = async (req, res) => {
  try { const result = await pool.query('SELECT * FROM stories ORDER BY
created_at DESC');
    res.json(result.rows);} catch (err) { console.error('Error getting histories:', err);
    res.status(500).json({ error: 'Server error' }); }};
exports.getHistoryById = async (req, res) => {
  const { id } = req.params;
  try {
    const result = await pool.query('SELECT * FROM stories WHERE id = $1', [id]);
    if (result.rows.length === 0) {return res.status(404).json({ error: 'Історія не
знайдена' }); } res.json(result.rows[0]);
  } catch (err) { console.error('Error getting history by id:', err);
    res.status(500).json({ error: 'Server error' }); }};
```

Ця частина витягує за допомогою SQL запитів інформацію та показує всі історії, та окрему історію за її ID. Далі буде показано код, який показує історії, які були написані конкретним користувачем.

```
exports.getHistoriesByUser = async (req, res) => {
  const { login } = req.params;
  try {
    const result = await pool.query(
```

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 50   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |

```

`SELECT s.*, fm.rejection_reason
FROM stories s
LEFT JOIN for_moderation fm ON s.id = fm.story_id
WHERE s.author = $1
ORDER BY s.created_at DESC`,
[login]
);
res.json(result.rows);
} catch (err) {
console.error('Error getting user histories:', err);
res.status(500).json({ error: 'Server error' });
}};

```

Нижче представлено код, який відповідає за отримання історій з запису у таблиці for\_moderation та відображення її у конкретного модератора.

```

exports.getHistoriesForModeration = async (req, res) => {
const { moderatorId } = req.params;
try {
const result = await pool.query(
`SELECT s.*
FROM for_moderation fm
JOIN stories s ON fm.story_id = s.id
WHERE fm.moderator_id = $1
ORDER BY fm.created_at DESC`, [moderatorId] );
res.json(result.rows); } catch (err) {
console.error('Error getting moderation histories:', err);
res.status(500).json({ error: 'Server error' });
}
};

```

І на останок, представлено код зміни статусу історії як у полі таблиці for\_moderation так і у таблиці stories

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 51   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |

```

exports.updateStoryStatus = async (req, res) => {
  const { id } = req.params;
  const { status, reason } = req.body;
  try { const storyResult = await pool.query(
    'UPDATE stories SET status = $1 WHERE id = $2 RETURNING *',
    [status, id]
  );
  if (storyResult.rowCount === 0) {
    return res.status(404).json({ error: 'Історія не знайдена' });}
  await pool.query(
    `UPDATE for_moderation
    SET status = $1,
    rejection_reason = CASE WHEN $1 = 'Rejected' THEN $2 ELSE
NULL END
    WHERE story_id = $3`,
    [status, reason || null, id]);
  res.json({ message: 'Статус оновлено', story: storyResult.rows[0] });
} catch (err) {
  console.error('Error updating story status:', err);
  res.status(500).json({ error: 'Server error' });
}};

```

Інші частини коду, такі як серверна частина, клієнтська частина, та компоненти застосунку. (дивитися Додаток А-В).

Ознайомившись з основними частинами коду, перейдемо до компонентів системи.

Розуміння точної роботи кожного компонента програмного забезпечення є ключовим для будь-якого програміста. Під «компонентами» розуміють модулі класів, які можуть бути незалежними системами або підсистемами з можливістю взаємодії з іншими елементами.

Тому нижче на рис. 2.14., представлена діаграма компонентів

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 52   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |

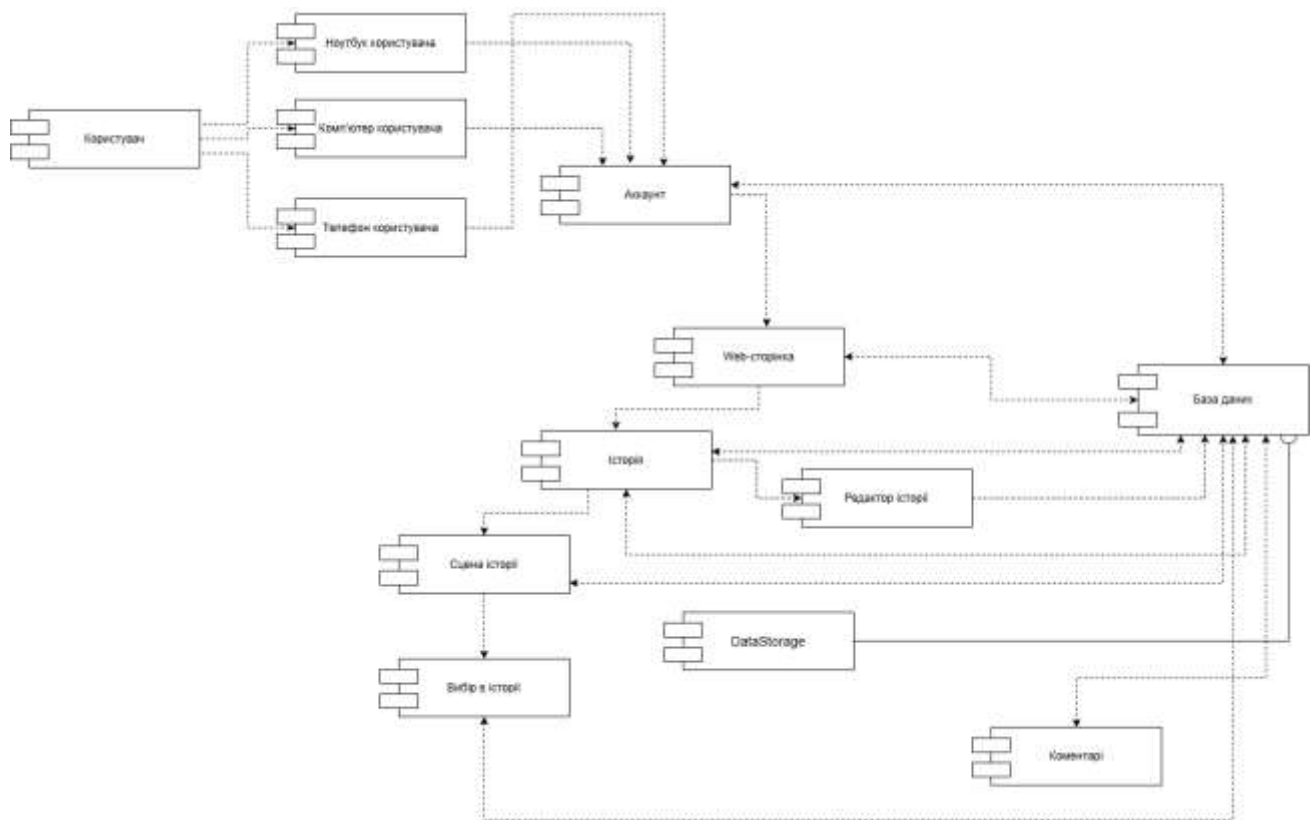


Рис. 2.14. Діаграма компонентів

Взаємодія компонентів системи на діаграмі має такий вигляд.

Користувачі використовують свої пристрої, щоб отримати доступ до своїх акаунтів.

Акаунти дають змогу користувачам взаємодіяти з web-сторінками, де вони можуть переглядати, редагувати та створювати контент.

Історії включають сцени та вибори, які користувачі можуть створювати та редагувати через редактор історій.

Коментарі дозволяють користувачам залишати відгуки і взаємодіяти з контентом і один з одним.

База даних зберігає всю необхідну інформацію, включаючи акаунти користувачів, історії та коментарі.

Таким чином, система забезпечує безперебійну роботу і взаємодію всіх її компонентів, надаючи користувачам інтуїтивно зрозумілий та ефективний спосіб взаємодії з контентом і один з одним.

## Висновки до другого розділу

Під час проектування застосунку для створення мультимедійних історій було проаналізовано варіанти використання програмного продукту згідно з вимогами користувачів. На основі цього була спроектована структура додатку, що включає взаємодію між інтерфейсом користувача, бекендом і системами зберігання мультимедійних файлів. Проектування передбачає створення бази даних для зберігання інформації про користувачів, історії, вподобайки до історій, лайки та дизлайки до коментарів, оцінки до історій, чорні мітки до історій, історії, що чекають на модерацію, що зберігаються на сервері. Описано алгоритм роботи застосунку, що включає етапи створення мультимедійних історій: від вибору шаблонів до додавання тексту, зображень і відео. Крім того, передбачено налаштування інтерактивних елементів, що дозволяють користувачам змінювати хід сюжету в залежності від вибору. Також розглянуті основні функції взаємодії користувача з додатком для редагування і публікації історій.

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 |      |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 | 54   |

## РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ ЗАСТОСУНКУ ДЛЯ СТВОРЕННЯ МУЛЬТИМЕДІЙНИХ ІСТОРІЙ

### 3.1. Порядок встановлення та налаштування параметрів застосунку

Встановлення та налаштування застосунку вимагає розгортання всіх його компонентів та налаштування як кожного елементу окремо, так і загальної системи в цілому. Тому для цього були визначені основні компоненти та після чого, створена діаграма розгортання, яка і показує архітектуру програмного комплексу та принципи його розгортання. Ця діаграма представлена на рис. 3.1

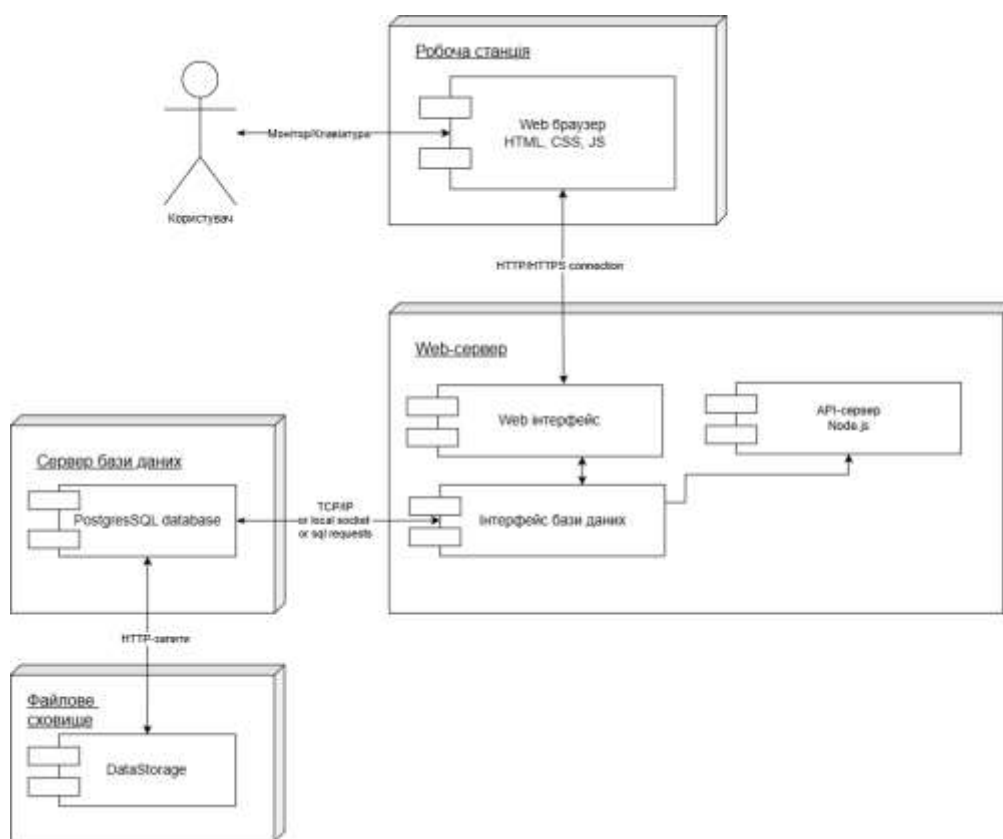


Рис. 3.1. Діаграма розгортання

Взаємодія на діаграмі розгортання описується так:

Користувач використовує пристрій для доступу до застосунку через веб-браузер. Браузер підтримує HTML, CSS та JavaScript.

Робоча станція користувача з'єднується з веб-сервером через HTTP або HTTPS. Це забезпечує безпечний та зашифрований зв'язок.

Веб-сервер містить кілька компонентів, а саме веб-інтерфейс, який обробляє запити від веб-браузера та повертає відповідні сторінки; інтерфейс

бази даних, який забезпечує зв'язок з базою даних для доступу до збережених даних; API-сервер на Node.js, який обробляє API-запити для взаємодії з додатками.

Сервер бази даних зберігає дані в PostgreSQL database. Веб-сервер взаємодіє з базою даних через SQL-запити, використовуючи TCP/IP або локальні сокети.

Файлове сховище розміщене окремо, а веб-сервер надсилає до нього HTTP-запити для збереження або отримання файлів. Усі зміни файлів зберігаються на стороні сервера.

### 3.2 Структура інтерфейсу застосунку для створення інтерактивних історій

Для початку, щоб користувач міг використовувати застосунок, йому потрібно авторизуватися або зареєструватися у системі.

Для реєстрації він вводить свої дані, такі як: ім'я та фамілія, логін, електронну пошту та пароль з підтвердженням його правильності.

Вигляд інтерфейсу реєстрації можна побачити на рис. 3.2:

Рис. 3.2. Реєстрація у застосунку

Після того, як користувач зареєструється, він одразу потрапляє на головний екран застосунку, а зверху з’являється повідомлення про успішну реєстрацію. Вигляд цього можна побачити на рис. 3.3.

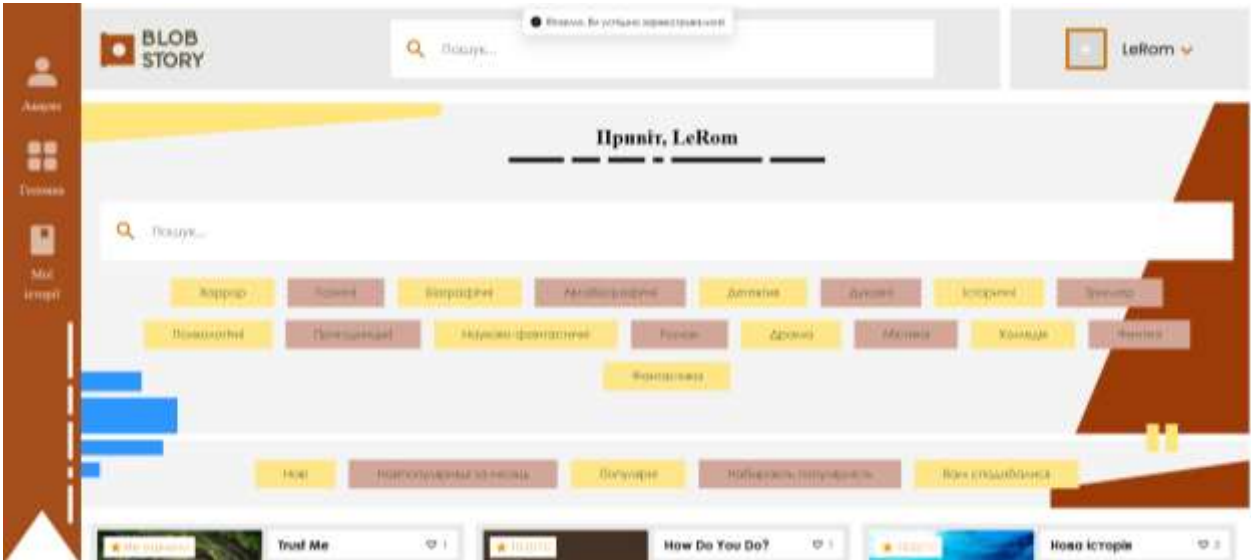


Рис. 3.3. Реєстрація пройшла успішно.

Якщо, ж у користувача вже створений обліковий запис, він вводить свою пошту та потрапляє на головний екран під своїм акаунтом. Вигляд форми авторизації представлений на рис. 3.4.

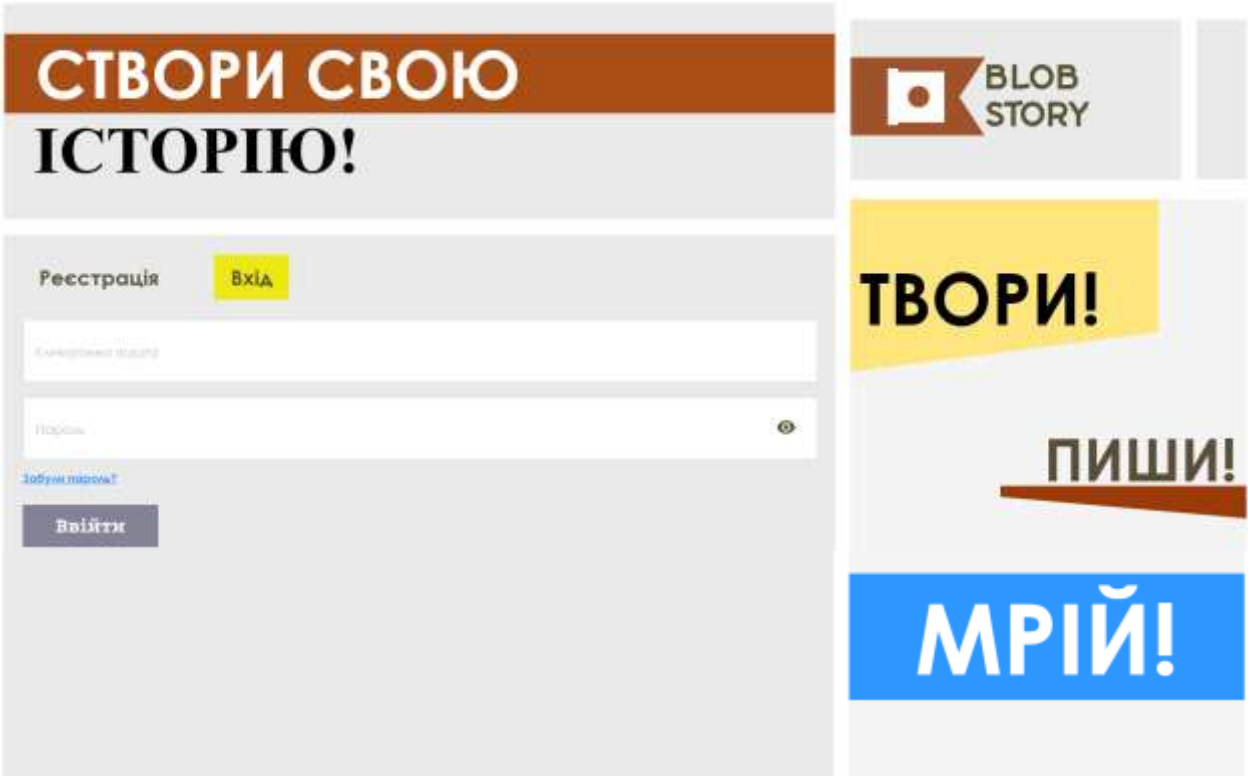


Рис. 3.4. Форма авторизації.

Після того, як користувач зайшов у застосунок, він бачить головний екран, з опублікованими історіям і може починати ознайомлення з контентом.

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 57   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |

Також він одразу може використовувати фільтр та пошук для більш легкого та комфортного пошуку історій. Вигляд головного екрану представлений на рис. 3.5.

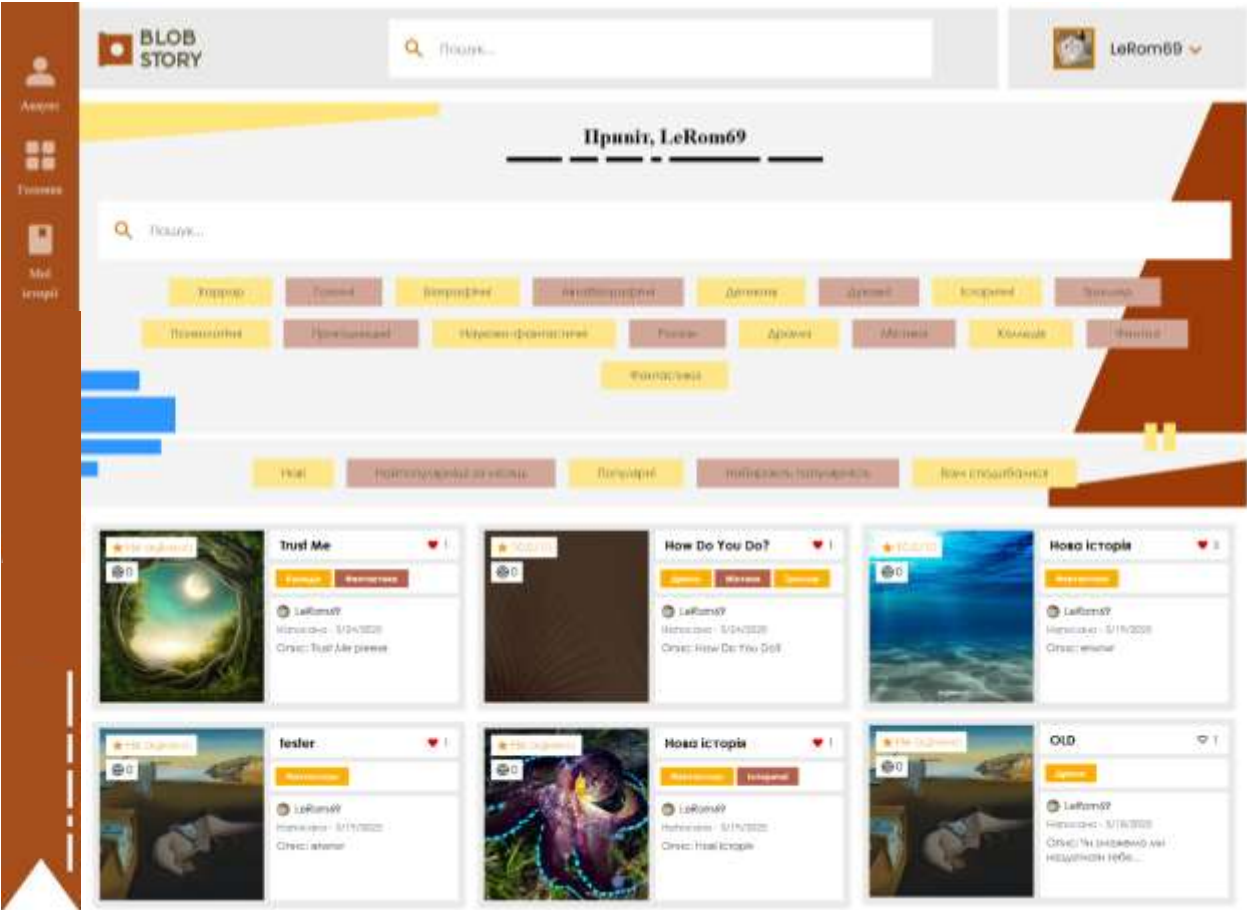


Рис. 3.5. Головний екран застосунку.

Далі, якщо користувач обере історію, яку хоче прочитати він може натиснути на її картку та перейти на її сторінку, де буде показана докладна про історію, з можливістю оцінити її, поставити чорну мітку, залишити коментар або оцінити коментарі інших користувачів або поділитися історією у соц мережах. Також натиснуваши на кнопку «Читати» він почне її прочитання, але якщо він перехоче читати історію, він може натиснути на кнопку «Головна» і далі продовжити пошук історій.

Вигляд сторінки історії представлений на рис. 3.6.

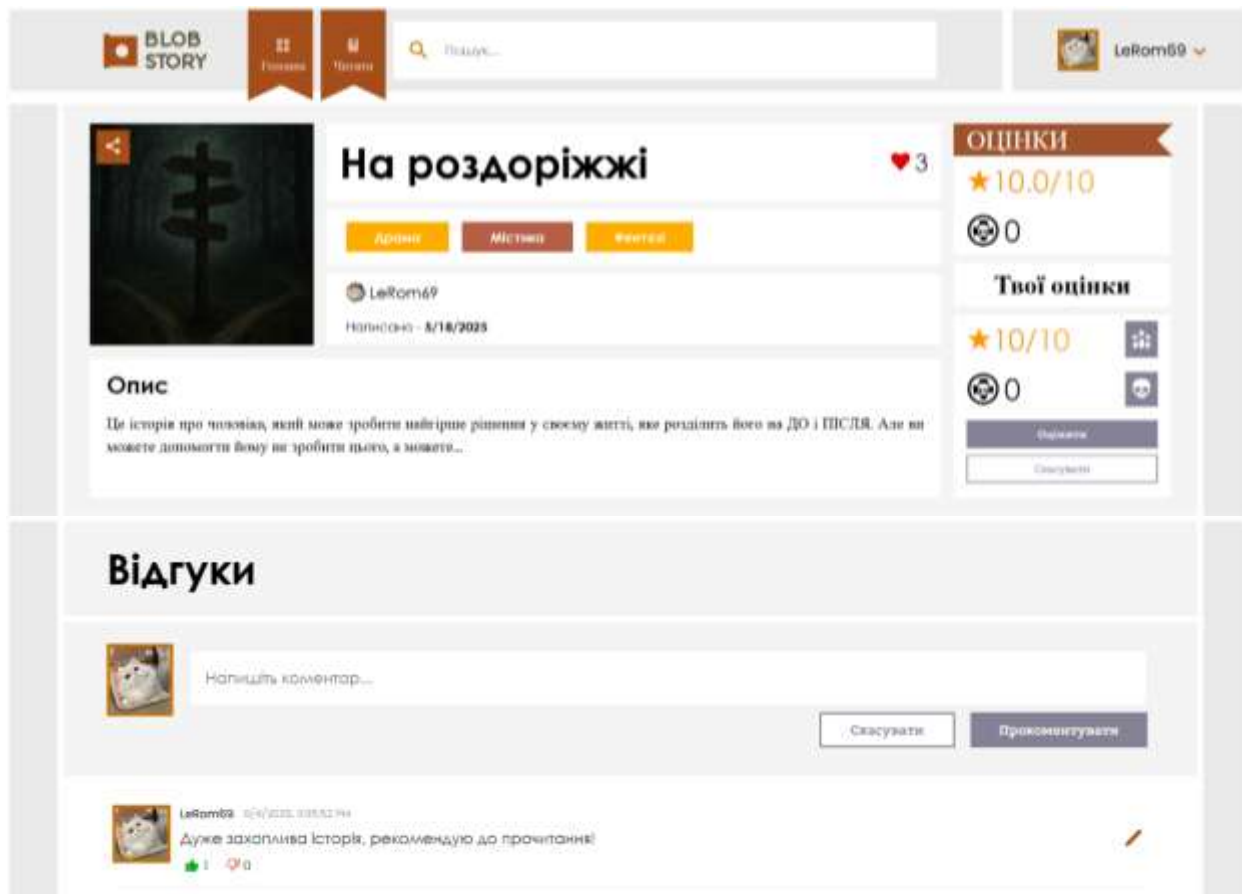


Рис. 3.6. Сторінка конкретної історії.

Далі, якщо користувач захоче створити нову історію, тоді він може використати панель навігації збоку, натиснувши на мої історії, де буде його робочий кабінет, де відображаються його історії та кнопка «Створити нову історію». Вигляд панелі робочого кабінету представлений на рис. 3.7.

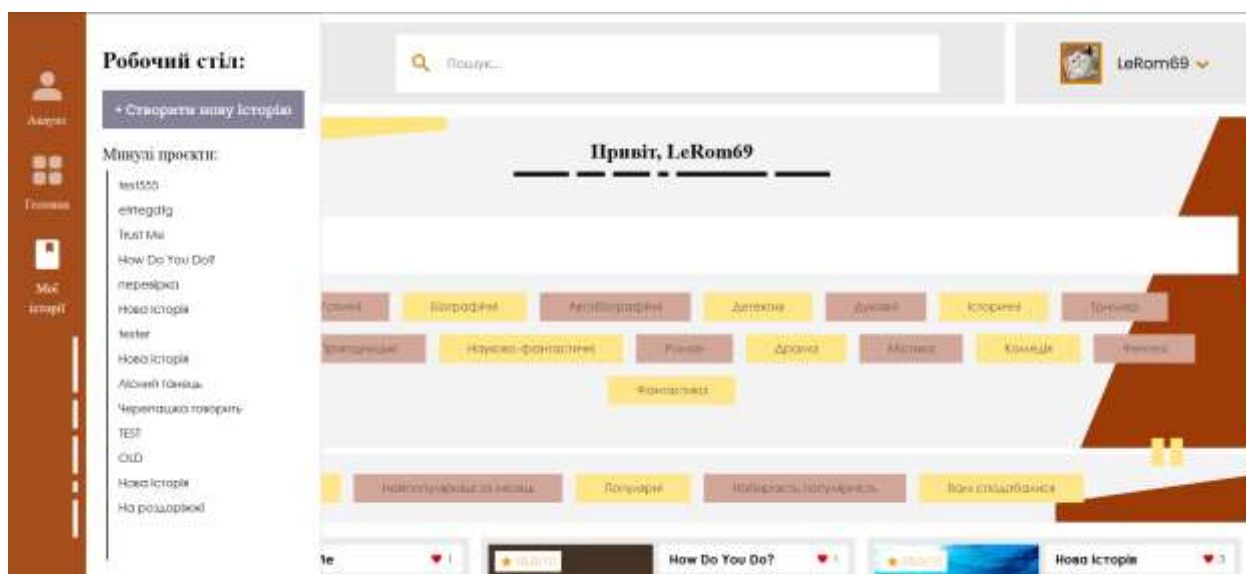


Рис. 3.7. Робочий кабінет користувача.

|      |      |                  |        |      |                                                 |            |
|------|------|------------------|--------|------|-------------------------------------------------|------------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк.<br>59 |
|      |      | Годлевський Ю.О. |        |      |                                                 |            |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |            |

Далі, якщо користувач захотів створити нову історію і натиснув на кнопку «Створити нову історію», він переходить на сторінку конструктора, та може починати творити.

Вигляд конструктору представлений на рис. 3.8.

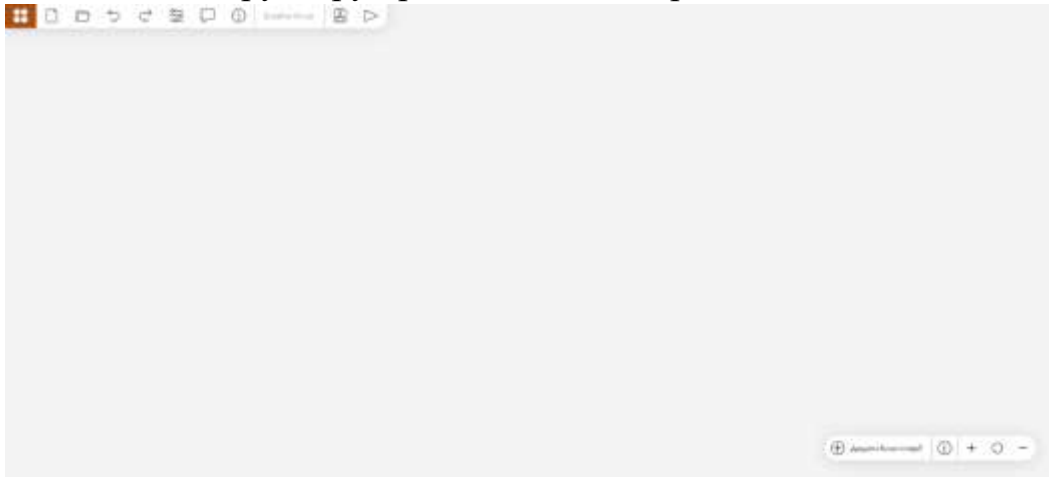


Рис. 3.8. Конструктор історій.

Тут він зможе створювати блоки сцен та додавати туди ріноманітні мультимедійні елементи.

Як відбувається сам процес створення історій, дивитися пункт 3.3. – Взаємодія з інтерфейсом та тестування застосунку.

Далі, перейдемо до особистого кабінету користувача, в який він може зайти або через рорир або за допомогою випадального списку, який активується при натисканні на блок з ім'ям та аватаркою користувача.

Вигляд випадального вікна показано на рис. 3.9., а особистого кабінету користувача на 3.10.



Рис. 3.9. Випадаюче вікно, з посиланнями на особистий кабінет користувача.

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 60   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |

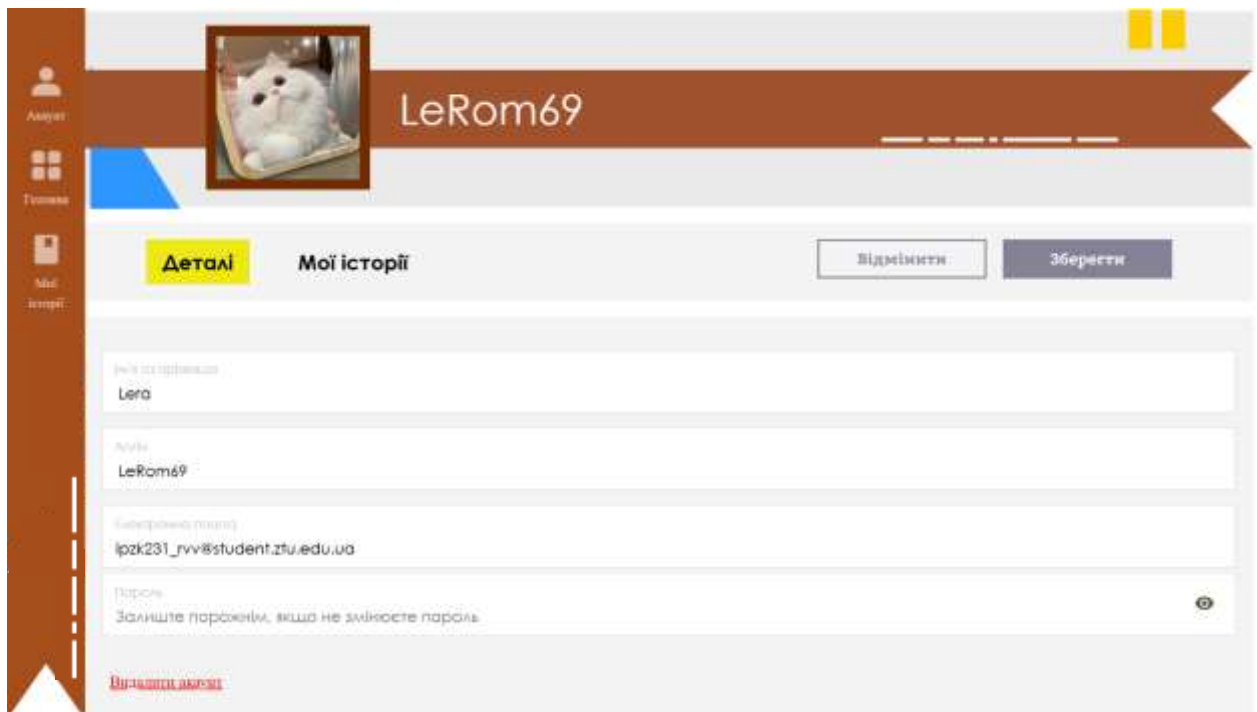


Рис. 3.10. Особистий кабінет користувача.

Як ми бачимо користувач, може змінювати свої облікові дані та переходити між вкладками, де знаходиться його особиста інформація та історії, які він написав. Далі, на рис. 3.11., 3.12. та 3.13, буде показаний зовнішній вигляд інтерфейсу вкладок.

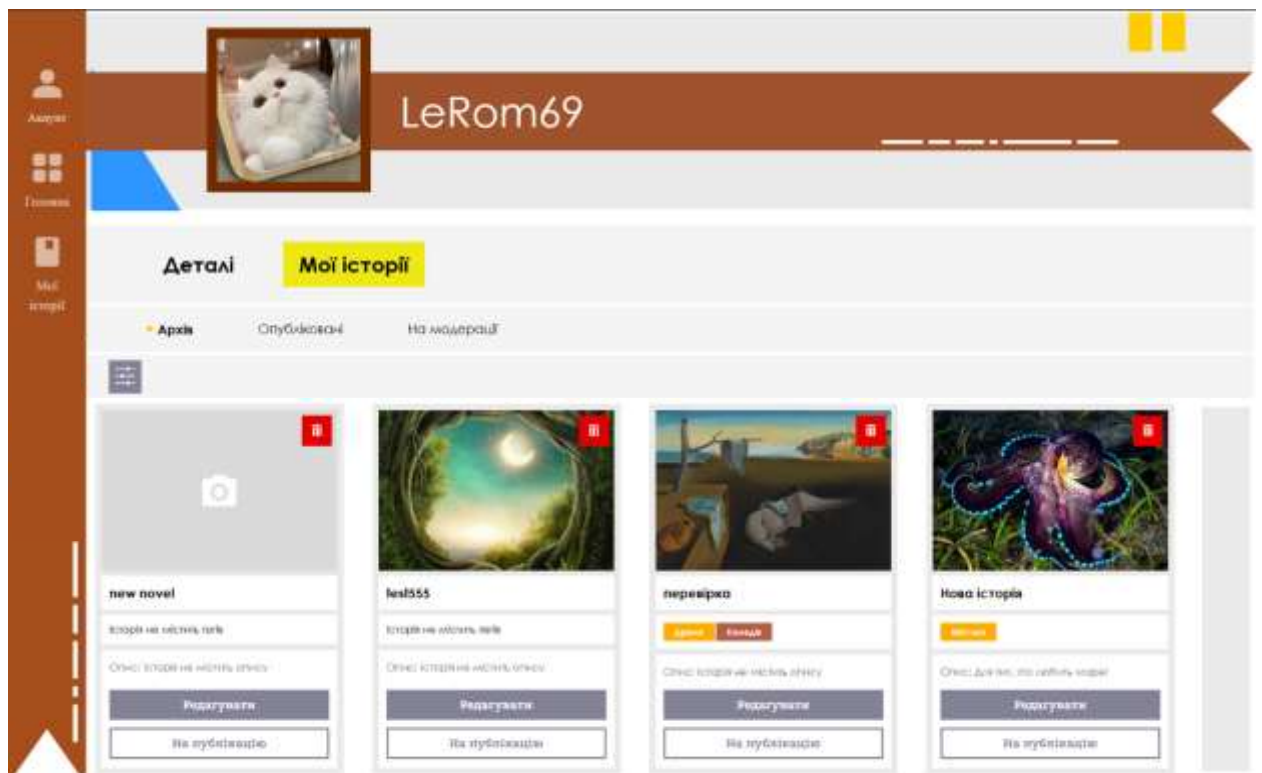


Рис. 3.11. «Архів» вкладка у «Мої історії».

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 61   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |

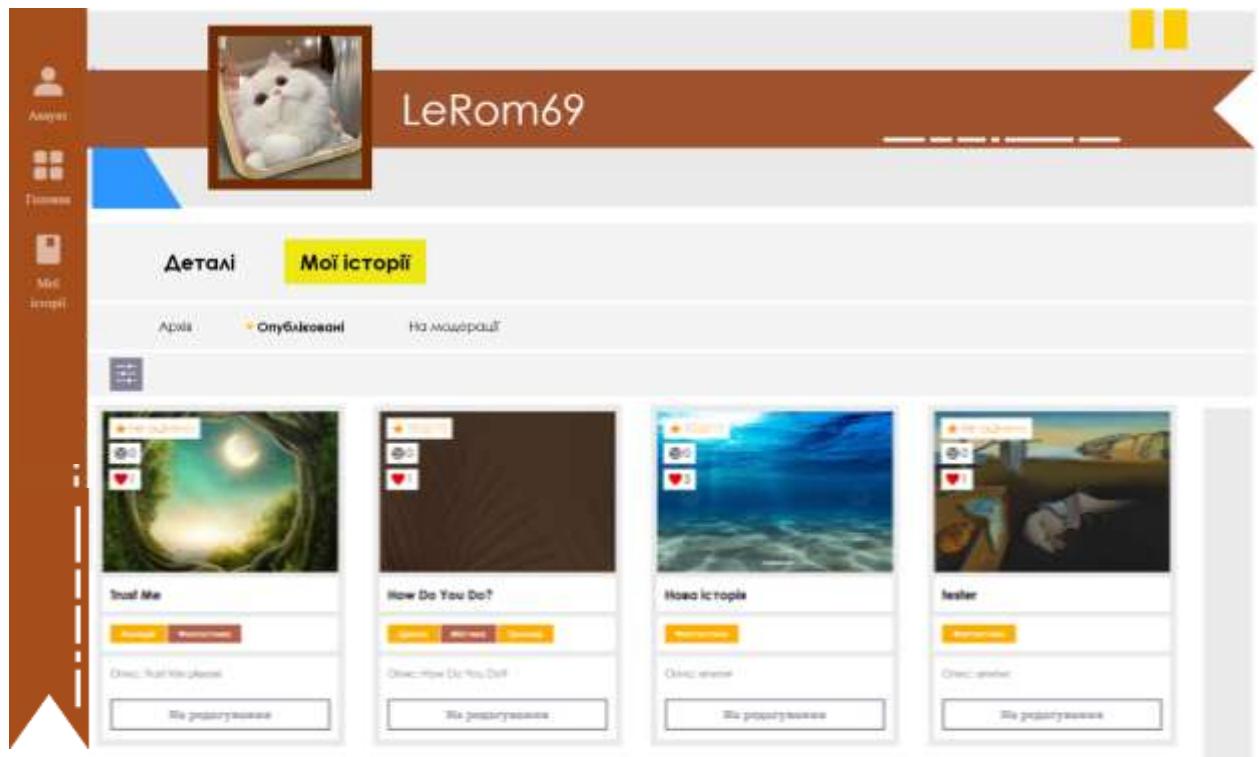


Рис. 3.12. «Опубліковані» вкладка у «Мої історії».

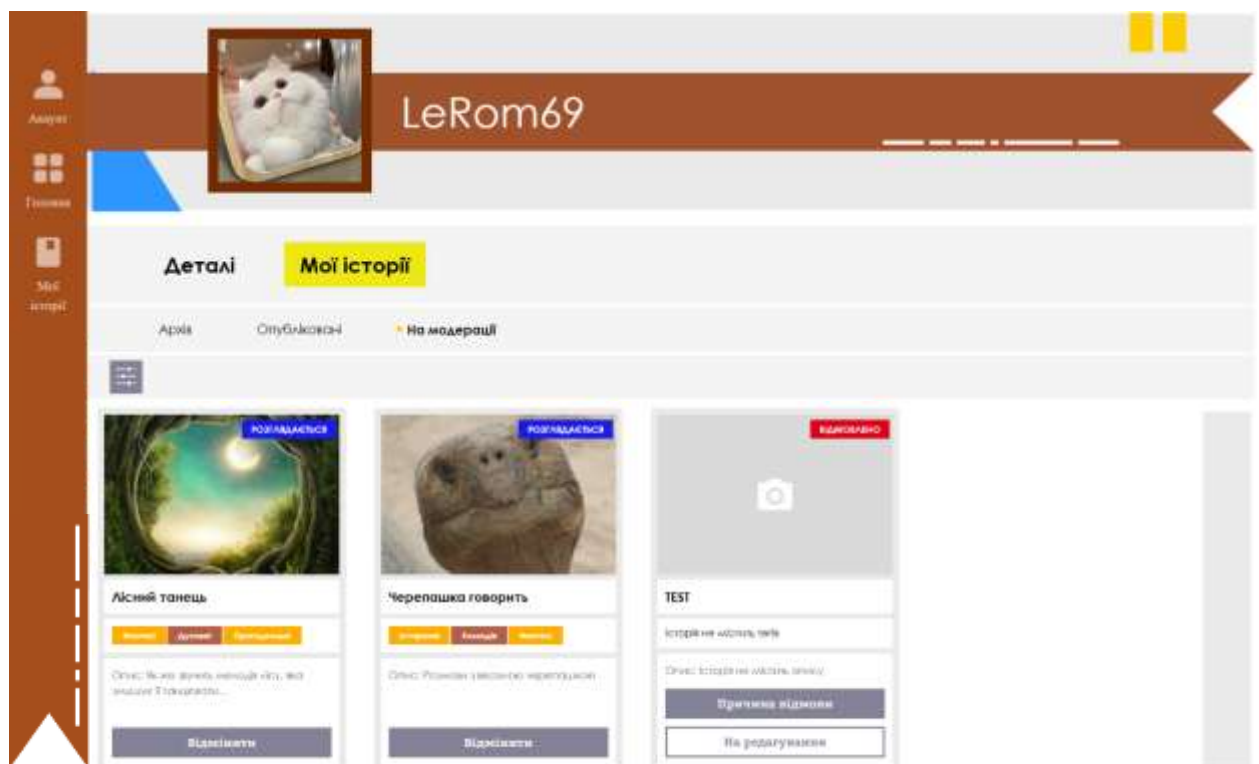


Рис. 3.13. «На модерації» вкладка у «Мої історії».

Як показано на рисунках, користувач має можливість зберігати свої історії у архів, без попередньої публікації, і при потребі відправляти їх на публікацію, видаляти, а також редагувати. Також користувач може предивлятися опубліковані історії, з статистикою, яка наразі наявна та моніторити, які історії розглядаються, а які відмовлені і може відмінити запит

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 62   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |

на публікацію та прочитати чому історія була скасована та почати її редагування.

Ознайомившись з інтерфейсом користувача, перейдемо до інтерфейсу адміністратора. Адміністратор також має можливості користуватися головним екраном, створювати нові історії, редагувати, відправляти на публікацію, видаляти та зберігати в архіві. Також він може змінювати свою особисту інформацію. Але, як адміністратор, він має адмін панель, де він може змінювати ролі користувачів та назначати модераторів на історії, які очікують модерацію.

Вигляд адмін панелі та її вкладок, представлений на рис. 3.14. та рис. 3.15.

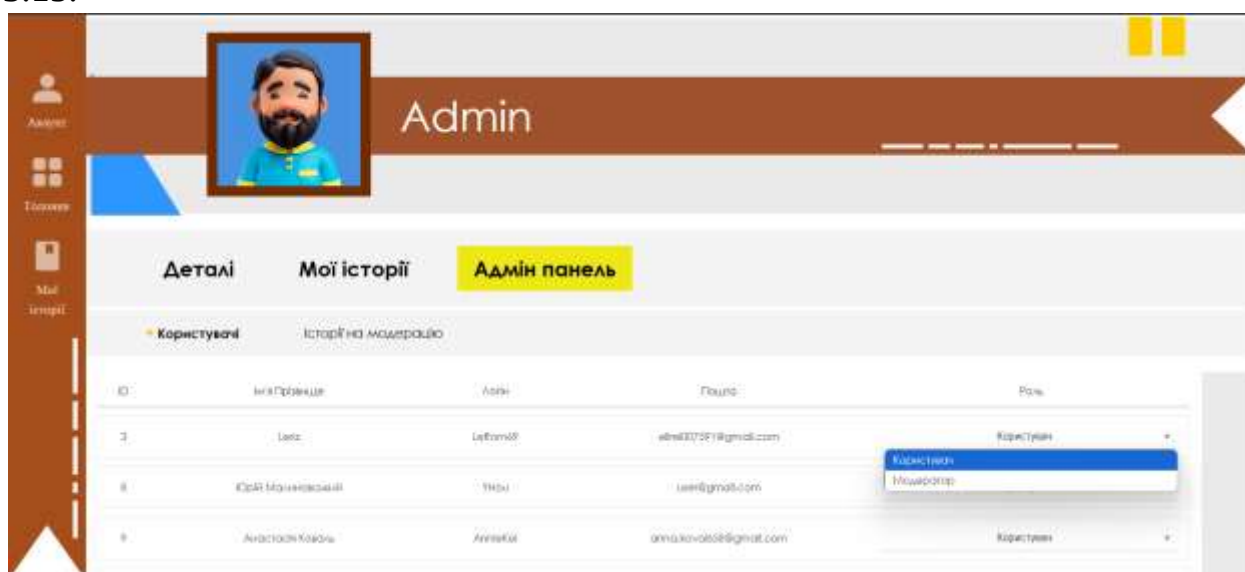


Рис. 3.14. «Користувачі» вкладка у «Адмін панель».

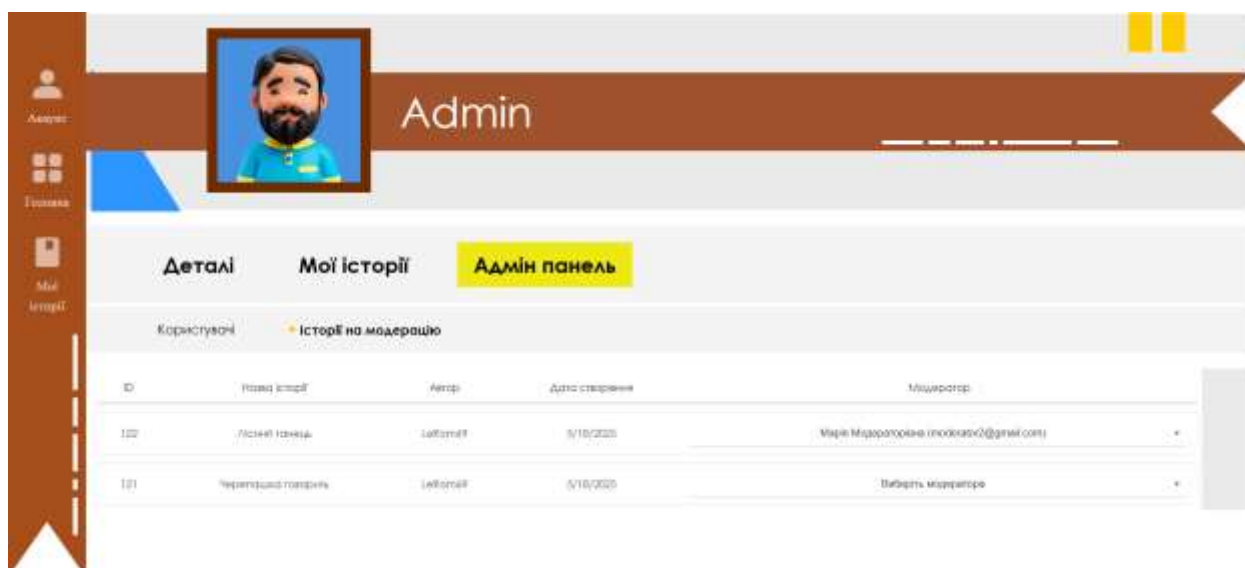


Рис. 3.15. «Історії на модерацію» вкладка у «Адмін панель».

Ознайомившись з інтерфейсом адміністратора, перейдемо до інтерфейсу модератора. Модератор, як і користувач та адміністратор, має можливості користуватися головним екраном, створювати нові історії, редагувати, відправляти на публікацію, видаляти та зберігати в архіві. Також він може змінювати свою особисту інформацію. Але, як модератор, він має мод-панель, де він може передивлятися історії, схвалювати їх або відхиляти, зазначаючи причину.

Вигляд мод-панелі та її вкладок, представлений на рис. 3.16. та рис. 3.17.

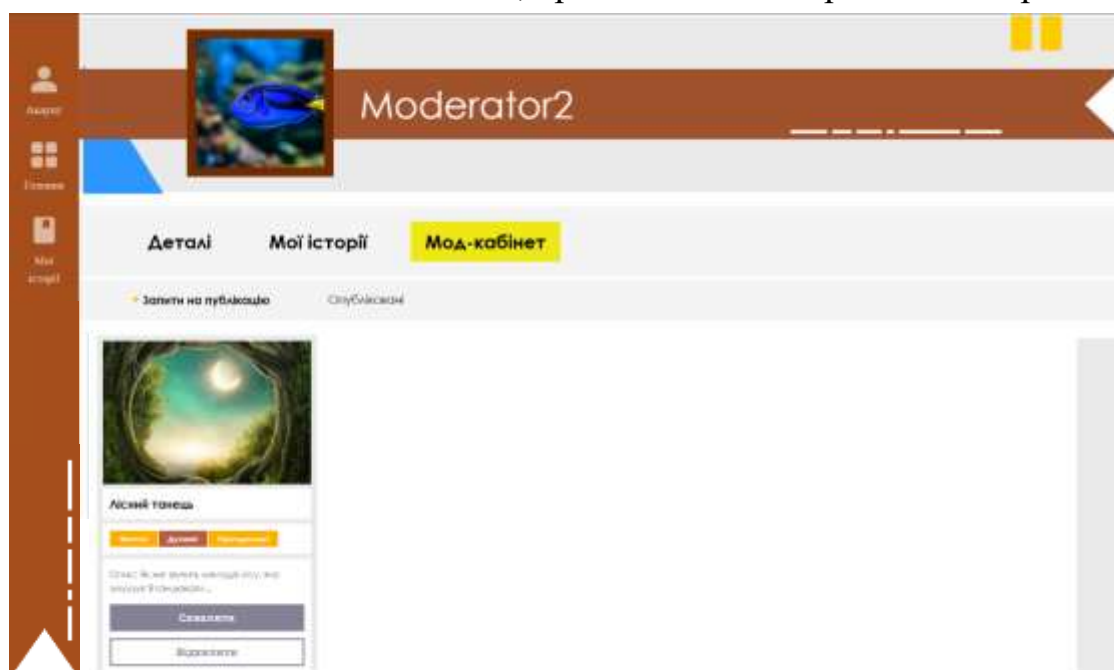


Рис. 3.16. «Запит на публікацію» вкладка у «Мод-панель».

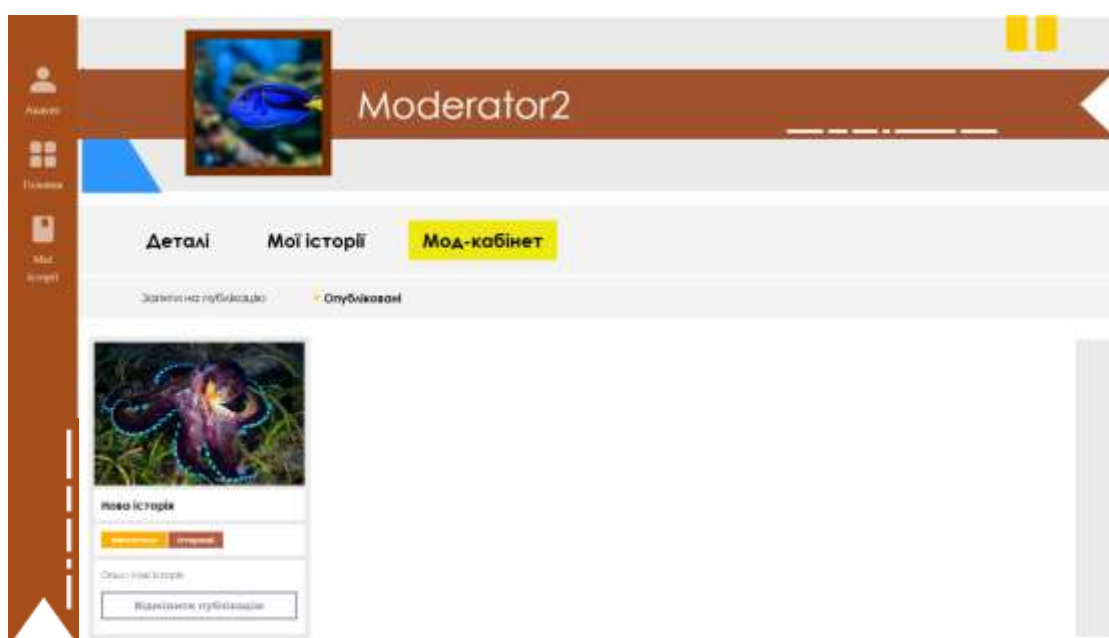


Рис. 3.17. «Опубліковані» вкладка у «Мод-панель».

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 64   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |

### 3.3. Взаємодія з інтерфейсом та тестування застосунку

Ознайомившись з інтерфейсом, ми можемо перейти до його тестування. Для початку розглянемо функції, які надані всім користувачам додатку, а саме фільтрація історій та їх пошук по назві. У застосунку представлено два блоки фільтрації, перший відповідає за фільтрацію по жанрам, а саме, коли користувач обирає жанр, на сторінці відображаються лише ті історії, у яких у жанрах є цей тег. Користувач також може обрати декілька жанрів, для більш широкої вибірки. Виконання цього функціоналу представлено на рис. 3.18.

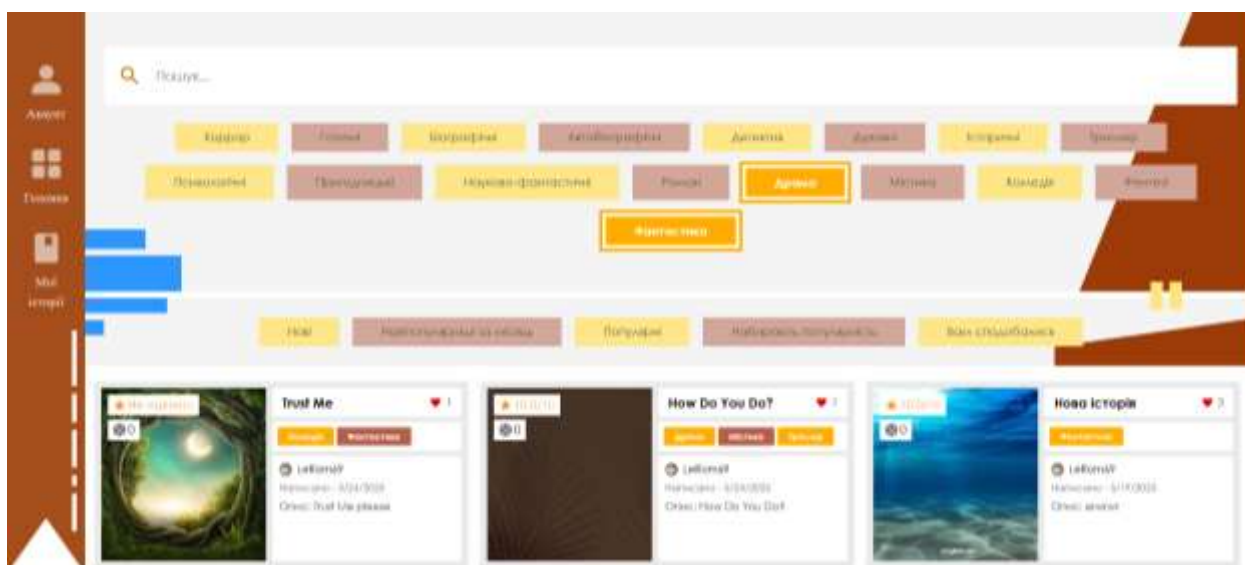


Рис. 3.18. Фільтрація історій по жанрам.

Другий блок фільтрації, відповідає за фільтрацію по відгукам користувачів та по даті створення історії. Нові відображають найновіші історії першими; найпопулярніші за місяць відображають першими історії, які набрали більше всього лайків за цей місяць; популярні, ті що набрали найбільшу кількість лайків; набирають популярність, ті що набрали деяку кількість лайків за день та які сподобалися користувачу – на які він сам поставив вподобайку. Також фільтр по жанрам та фільтр по по відгукам користувачів та по даті створення історії можна використовувати разом.

Як відбувається фільтрування представлено на рис.. 3.19., 3.20., 3.21., 3.22., 3.23.

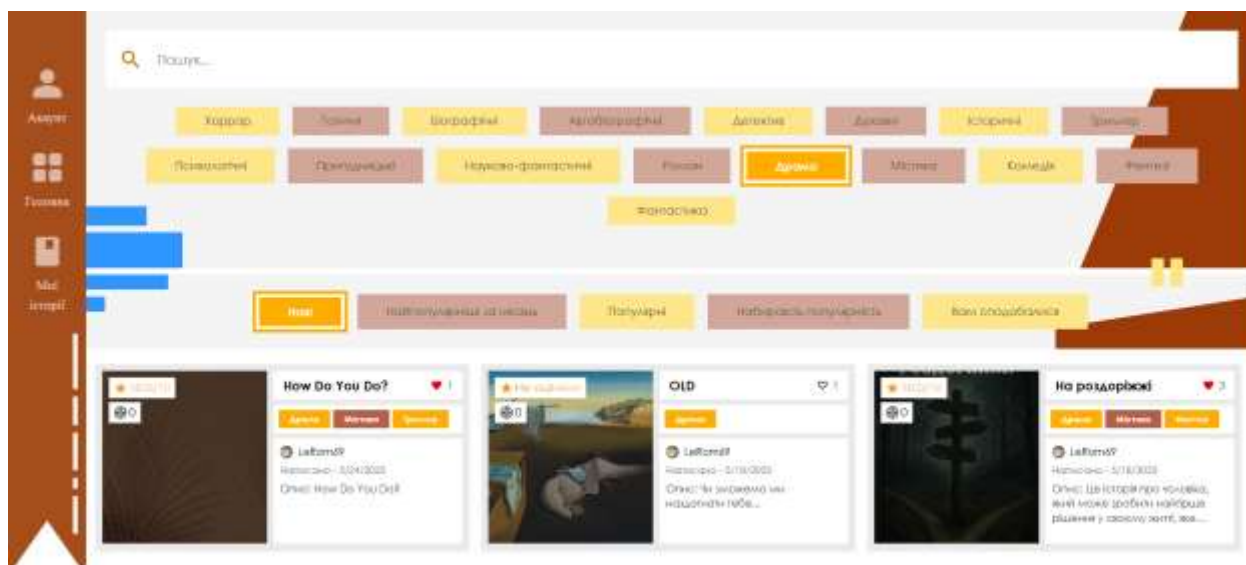


Рис. 3.19. Фільтрація історій по жанрам та дані створення.

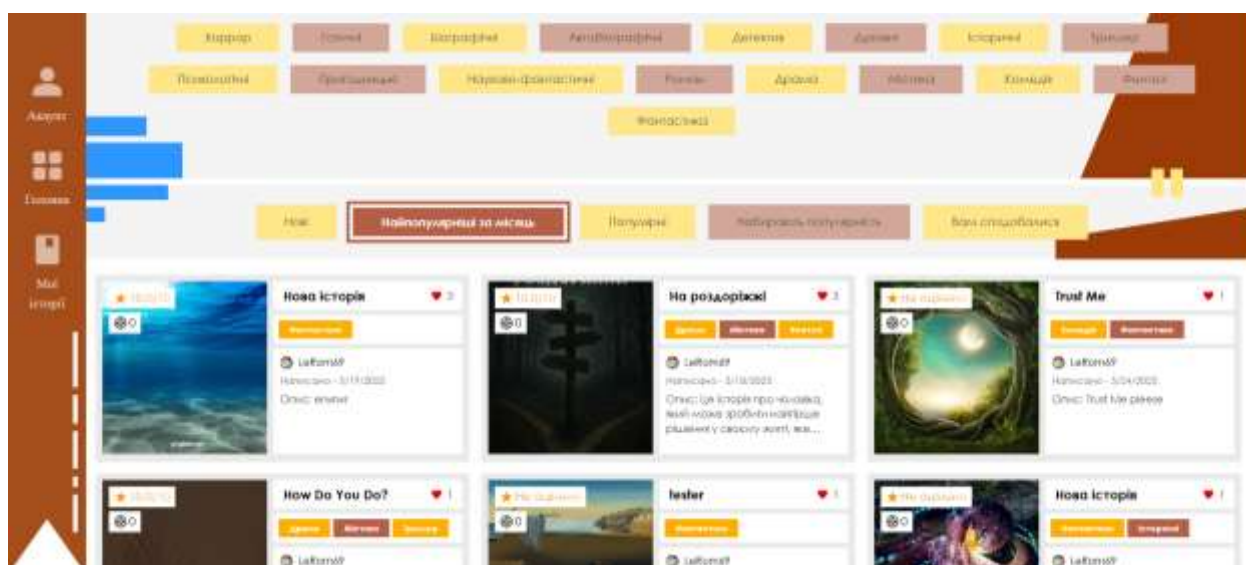


Рис. 3.20. Фільтрація історій по популярності за місяць.

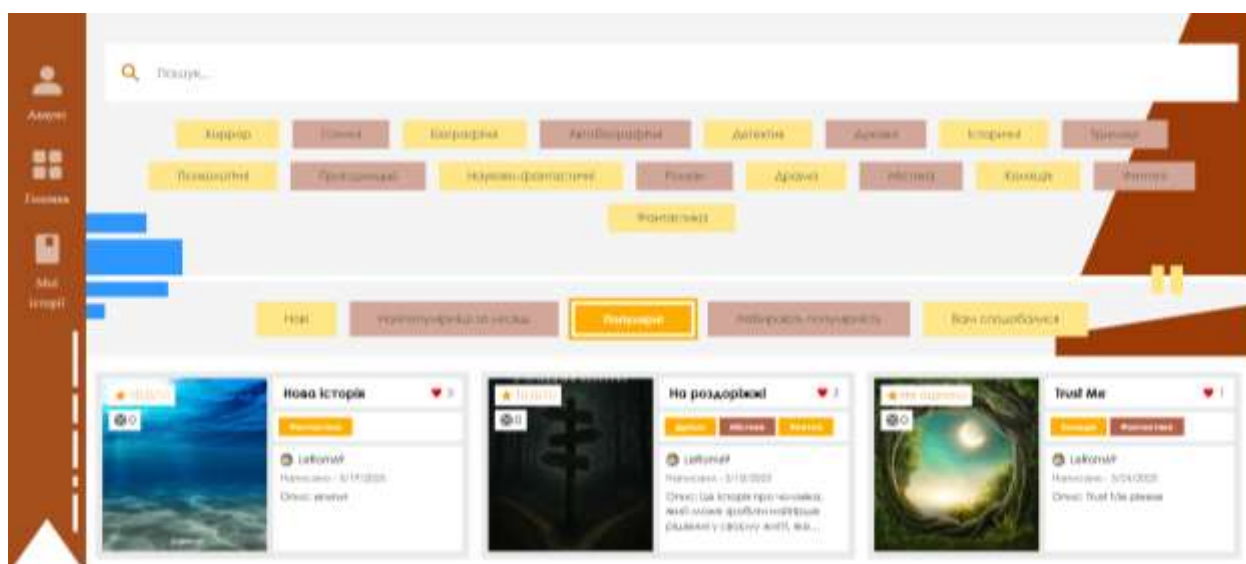


Рис. 3.21. Фільтрація історій по популярності.

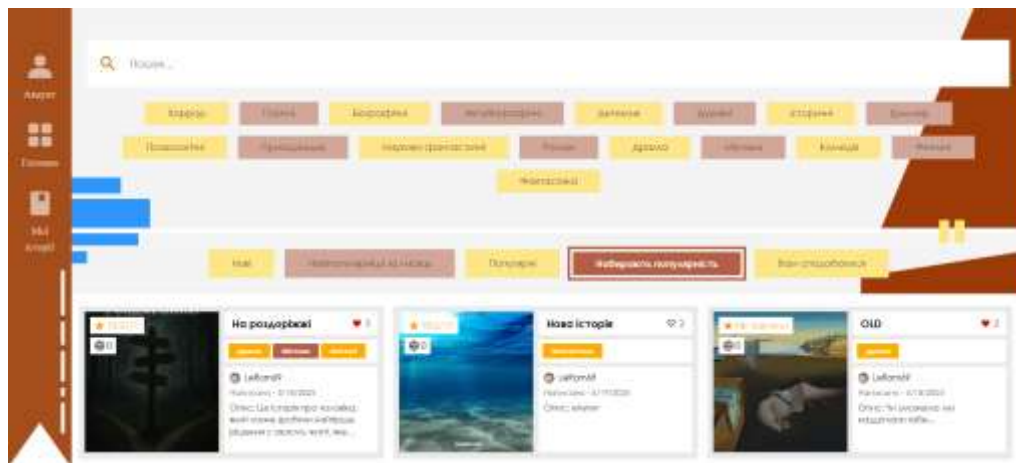


Рис. 3.22. Фільтрація історій які набирають популярність.

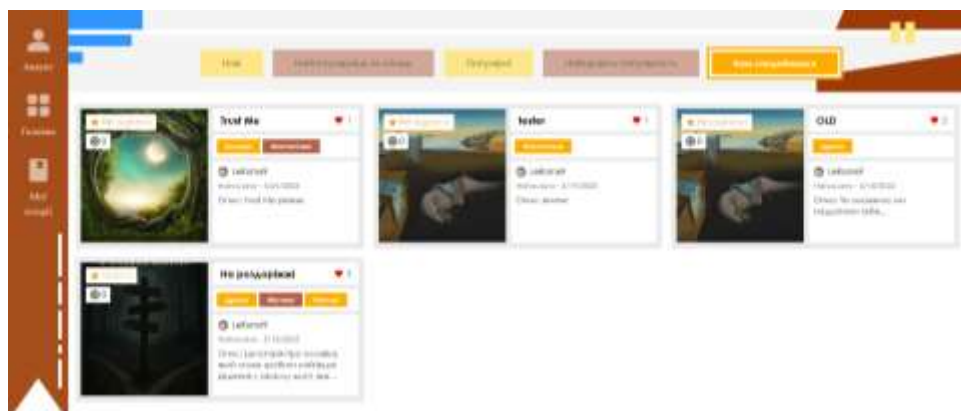


Рис. 3.23. Фільтрація історій які сподобалися користувачу.

Протестувавши фільтрацію, можемо перейти до пошуку на сайті, він виконується лише для опублікованих історій на сайті, не опубліковані історії, не будуть відображатися у пошуку. Також пошук за першими буквами, які введе користувач буде надавати можливі варіанти, які шукає користувач. Після того як користувач ввів назву історії він одразу потрапляє на її сторінку.

Як працює функціонал, представлено на рис. 3.24.



Рис. 3.24. Пошук історії по назві.

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 67   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |

Далі перейдемо до функціоналу прочитання історії. Коли користувач обрав історію і хоче її прочитати, він натискає на кнопку «Читати», після чого відкривається нова вкладка з самою історією.

Як може виглядати історія, представлено на рис. 3.25.



Рис. 3.25. Читання історії.

Як показано на рисунку, користувач має діалогове вікно та фон, але вміст може змінюватися від історії до історії, те що показано вище, є лише прикладом, як воно може виглядати.

Далі перейдемо оцінювання історій. Користувач соже виставити нову оцінку, або чорну мітку або все разом, але виставлення оцінки відбудеться лише тоді, коли користувач натисне на кнопку оцінити, тоді оцінка вадбудеться.

Як працює функціонал, представлено на рис. 3.26, 3.27.

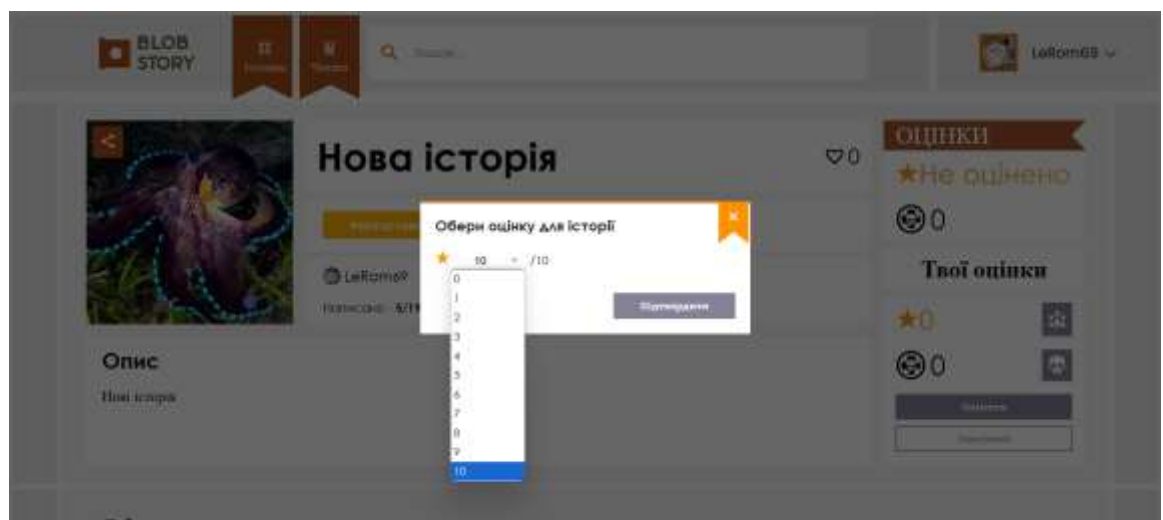


Рис. 3.26. Рорир оцінювання історії.

|      |      |                  |        |      |                                                 |            |
|------|------|------------------|--------|------|-------------------------------------------------|------------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк.<br>68 |
|      |      | Годлевський Ю.О. |        |      |                                                 |            |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |            |

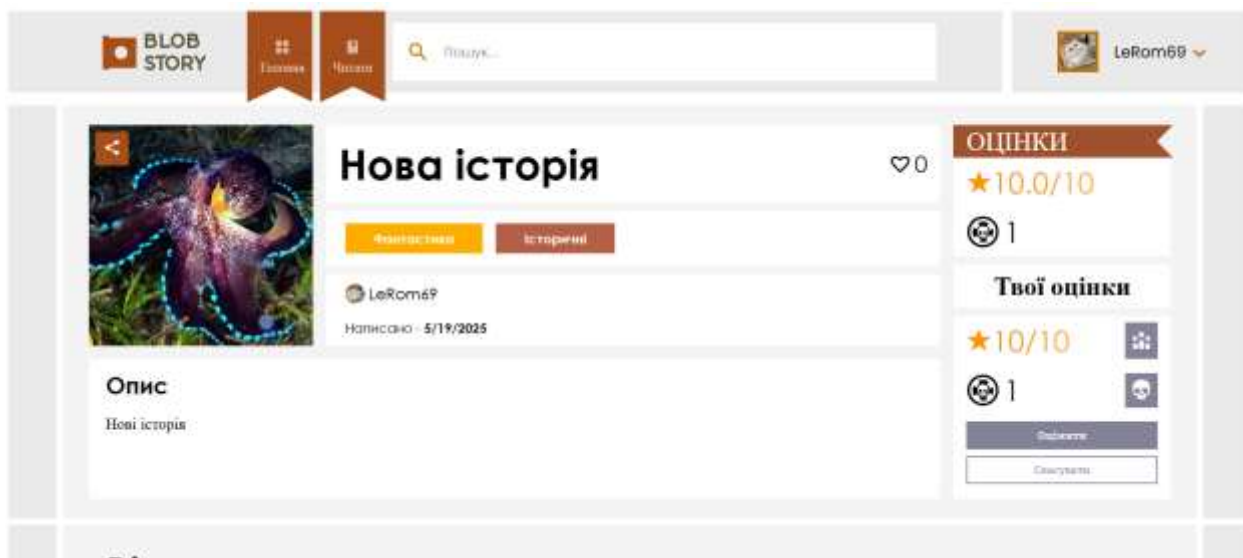


Рис. 3.27. Оцінку виставлено на відображено у табло «Оцінки».

Також користувач може виставити лайк історії, просто натиснувши на сердечко тоді воно загориться червоним, також користувач може його прибрати. Як працює функціонал, представлено на рис. 3.28.

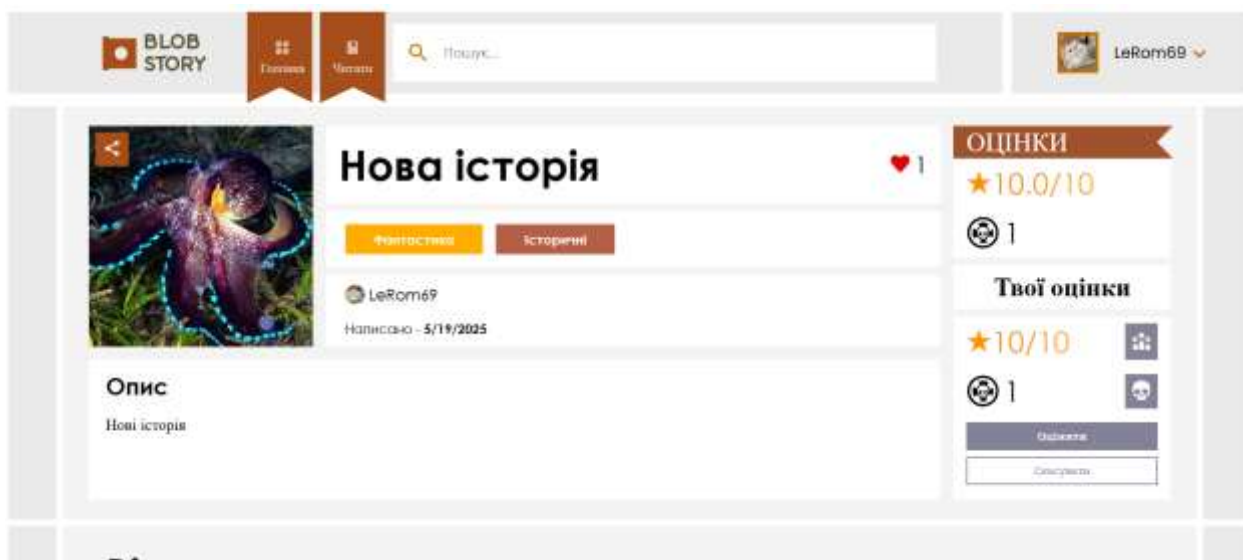


Рис. 3.28. Виставлення вподобайки.

Далі перейдемо до функціоналу відправки посилання історій. Для цього користувач натискає на кнопку зі значком «Поширити», та у попапі обирає спосіб відправки. Як працює цей функціонал, представлено на рис. 3.29. та рис.. 3.30.

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 69   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |

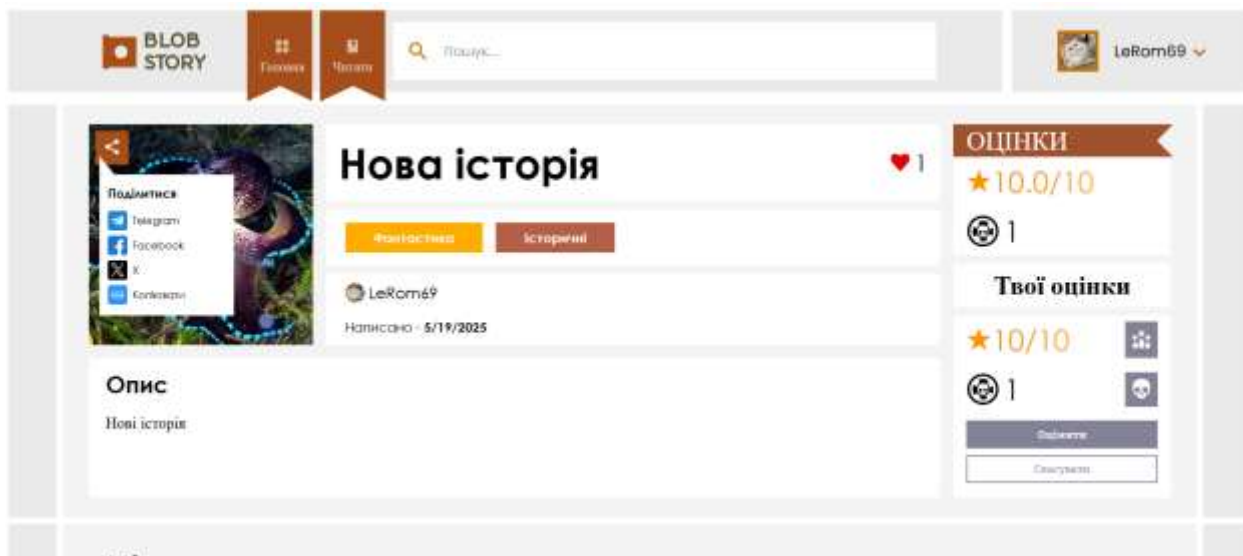


Рис. 3.29. Рорир для вибору способу поширення посилання.

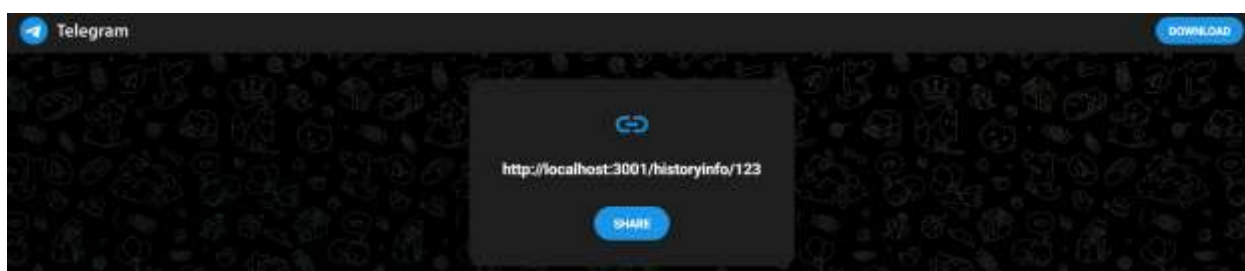


Рис. 3.30. Посилання готове для поширення через Telegram.

Далі користувач може залишити відгук та оцінити відгуки інших людей. Також редагувати або видаляти свій. Цей функціонал, представлено на рис. 3.31., рис.. 3.32., рис.. 3.33., рис.. 3.34. та рис.. 3.35.

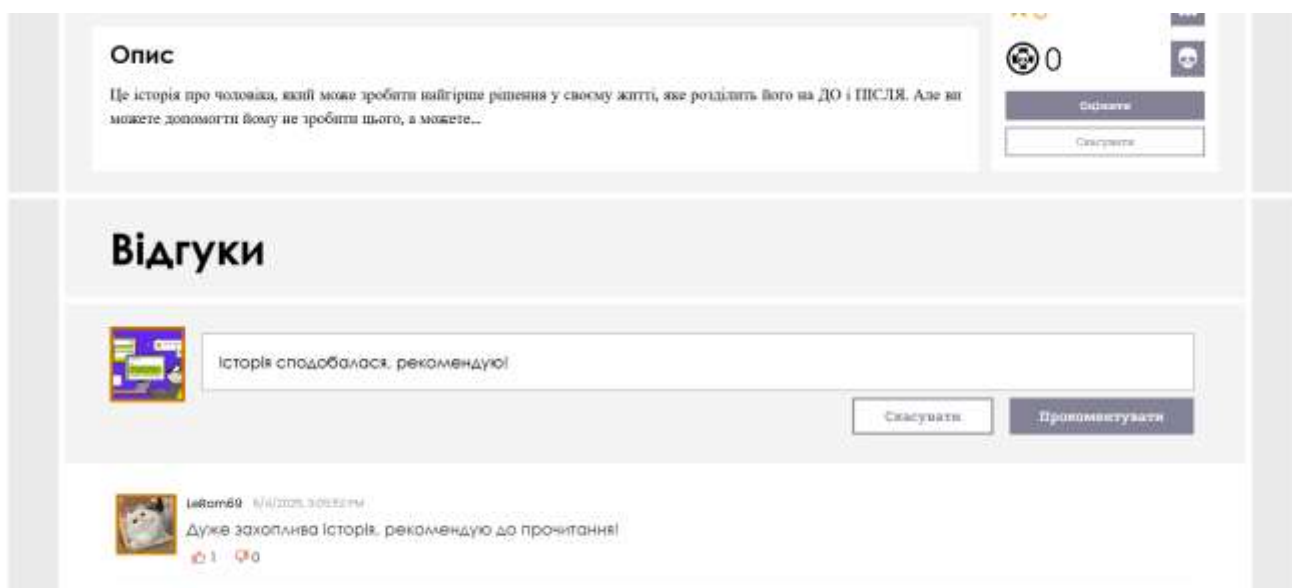


Рис. 3.31. Написання коментаря.

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 70   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |



Рис. 3.32. Коментар відправлено, з можливістю його редагування та видалення.

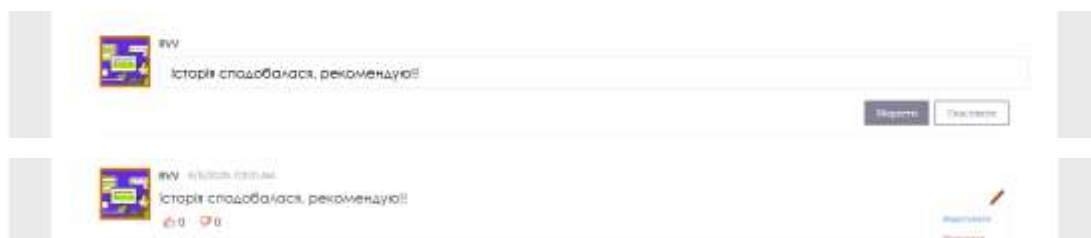


Рис. 3.33. Коментар відредаговано і зміни збережено.

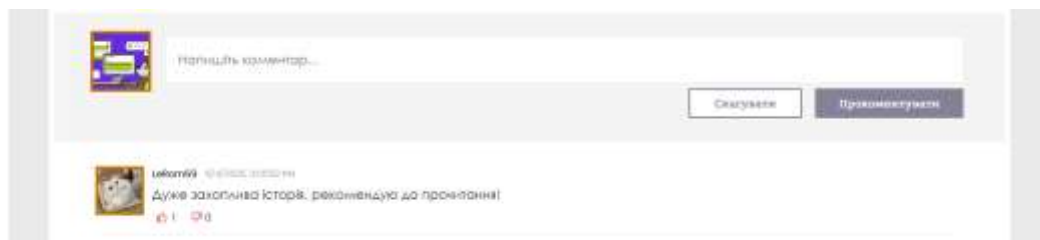


Рис. 3.34. Коментар видалено.



Рис. 3.35. Підтвердження роботи виставлення реагування на коментарі.

Протестувавши роботу з вже опублікованими історіями, перейдемо до тестування створення нових. Для цього ми переходимо у конструктор, натиснувши на кнопку у робочому кабінеті користувача.

Всі етапи створення історії, будуть проілюстровані на рис.. 3.36., рис.. 3.37., рис.. 3.38. та рис.. 3.39.

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 71   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |

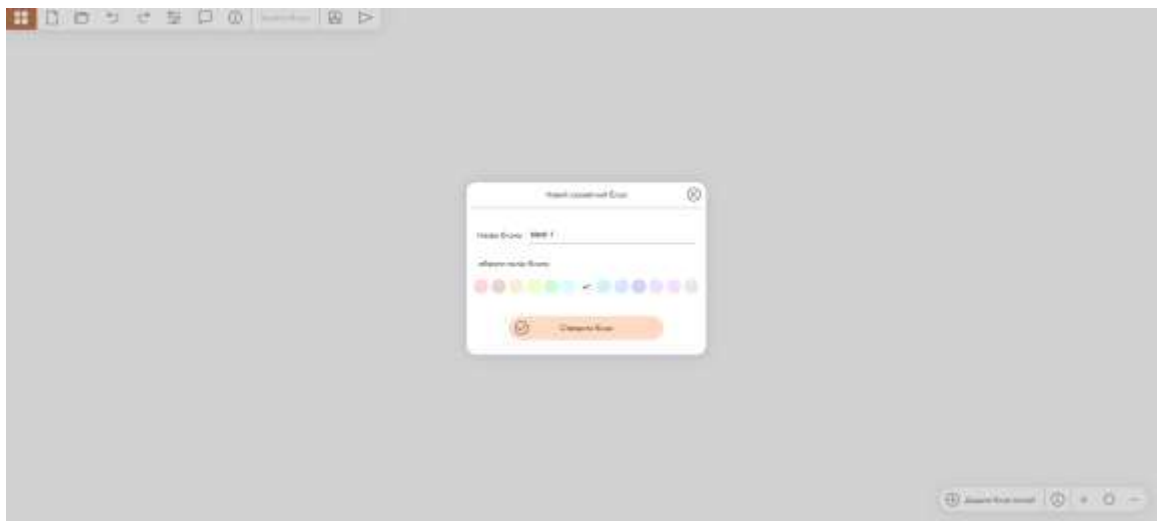


Рис. 3.36. Створення блоку сцени історії.



Рис. 3.37. Завантаження медіафайлів з комп'ютера користувача для налаштування заднього фону сцени.

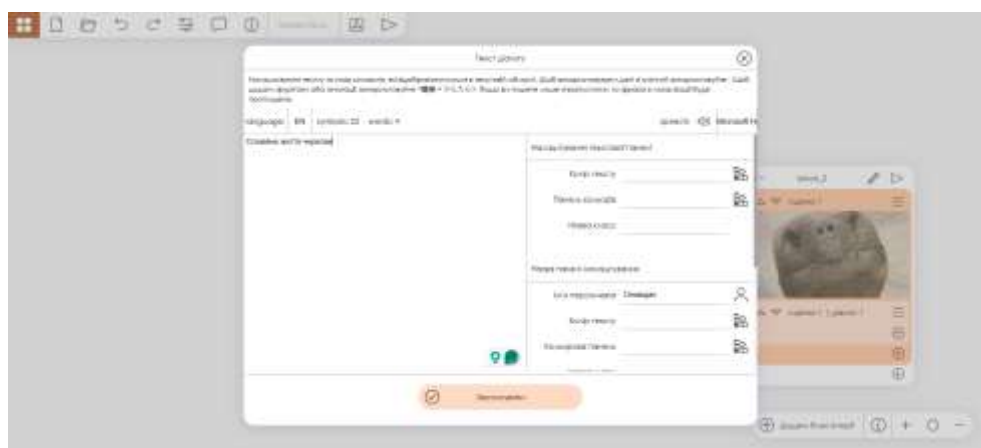


Рис. 3.38. Налаштування діалогу у сцені.



Тепер, налаштувавши проект, користувач може зберегти його, для цього він вводить його назву, обирає теги, опис та обкладинку, після чого зберігає проект або в архів, або відправляє одразу на публікацію.

Як працює цей функціонал, представлено на рис. 3.42., та рис. 3.43.

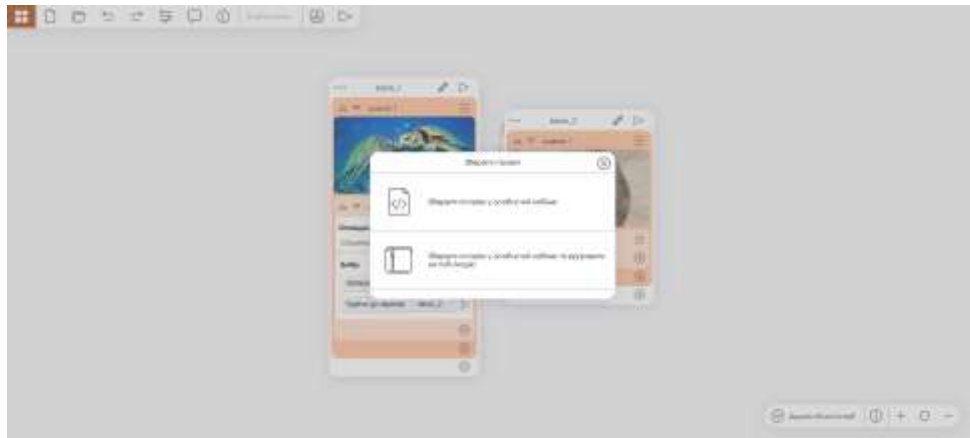


Рис. 3.42. Спосіб збереження історії.

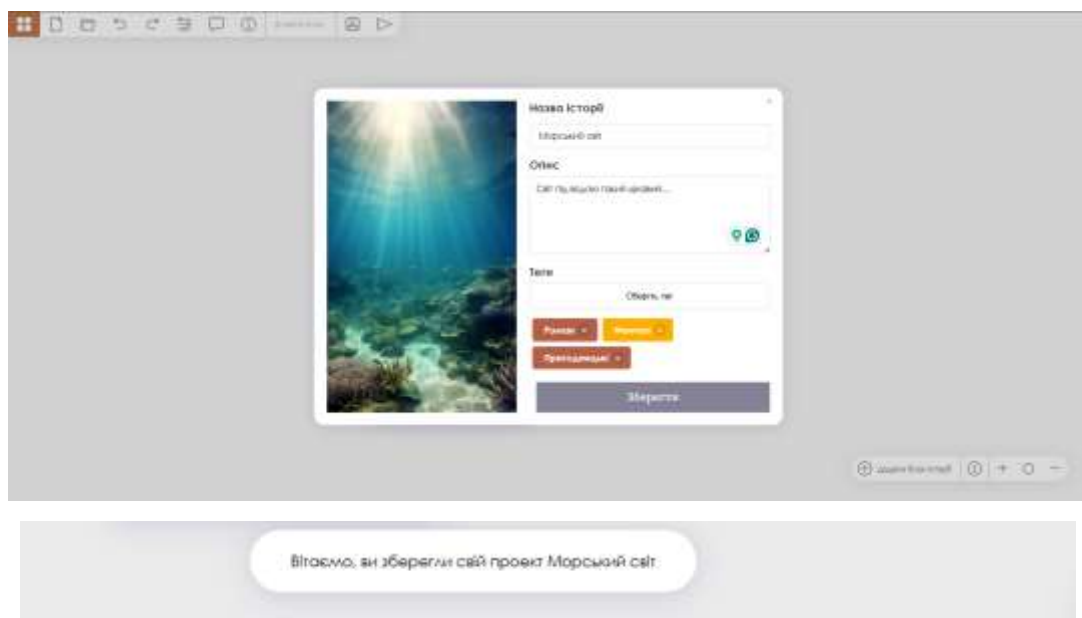


Рис. 3.43. Налаштування проекту перед публікацією та його збереження.

Після збереження проекту, він потрапляє у особистий кабінет та у робочий стіл користувача.

Як працює фільтрація та збереження у робочий кабінет вже було показано, тому далі буде протестовано відправку на модерацію, адміністрування та сама модерація. Спочатку буде протестовано зміну ролей та обрання модератора.

Роботу цього функціоналу, було проілюстровано на рис. 3.44. та 3.45.

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 74   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |

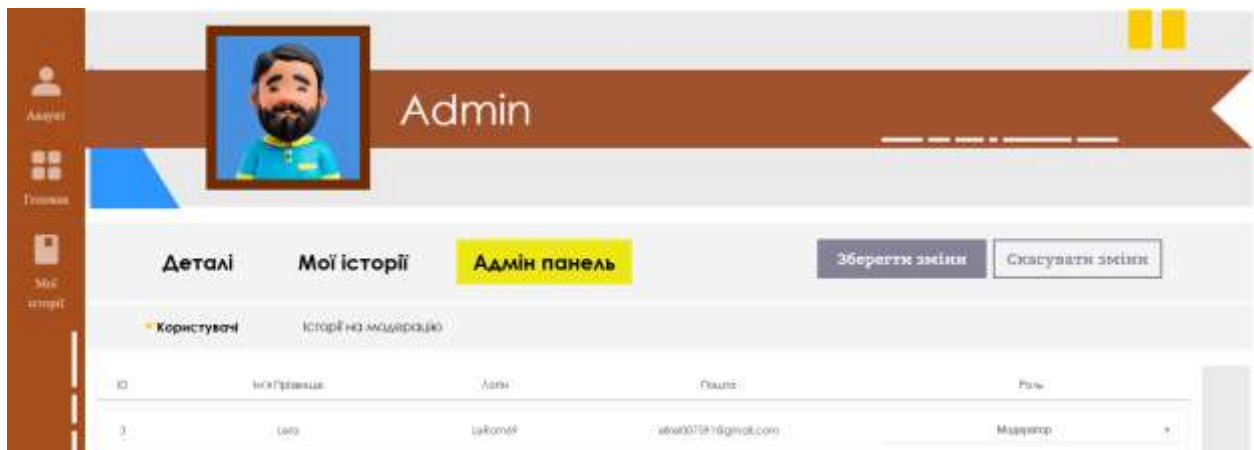


Рис. 3.44. Зміна ролі.

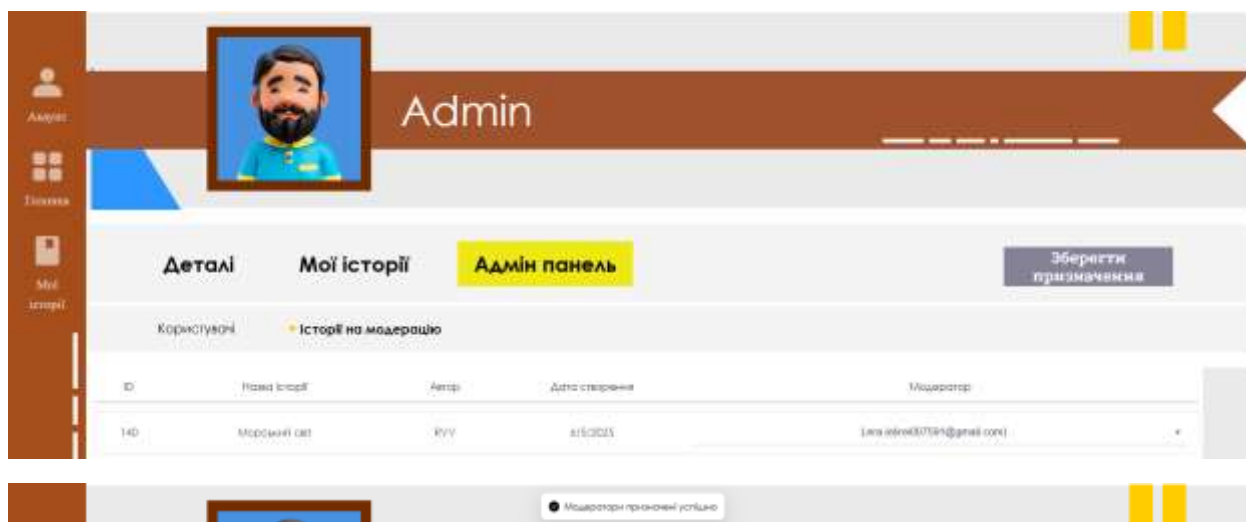


Рис. 3.45. Призначення модератора до історії.

Обравши модератора, ми можемо перейти до тестування функцій публікації історій.

Для цього модератор може сам прочитати історію, після чого затвердити або відхилити її, заздалегідь написавши причину.

Як працює цей функціонал, представлено на рис. 3.46. та 3.47.

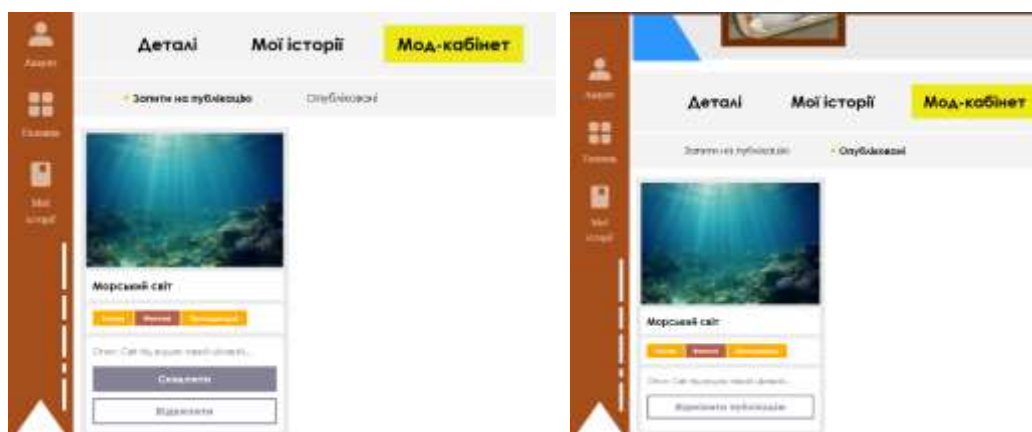


Рис. 3.46. Історію схвалено модератором.

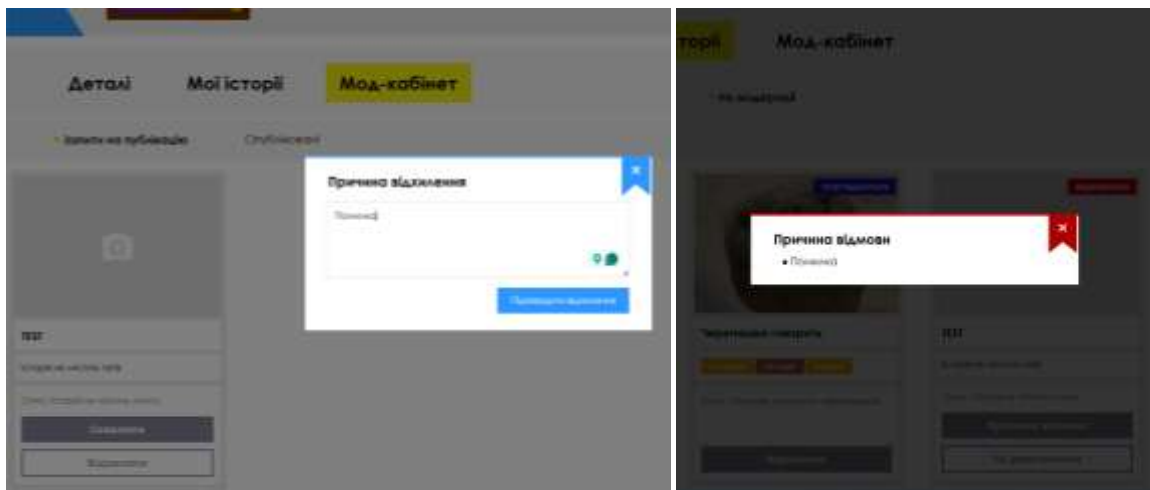


Рис. 3.47. Історію відхилено модератором.

Як показано, цей функціонал також працює.

Також, для модератора створено такий функціонал, якщо кількість чорних міток буде більше 100(для демонстрації буде вибірка більше 4), тоді у мод-панелі буде з'являтися червоний кружечок повідомлення, а на картці історії з'явиться червона плашка, які будуть свідчити про те, що історію треба зняти з публікації. Також модератор може зняти історію з публікації зазначивши причину такого вчинку.

Це проілюстровано на рис. 3.48.

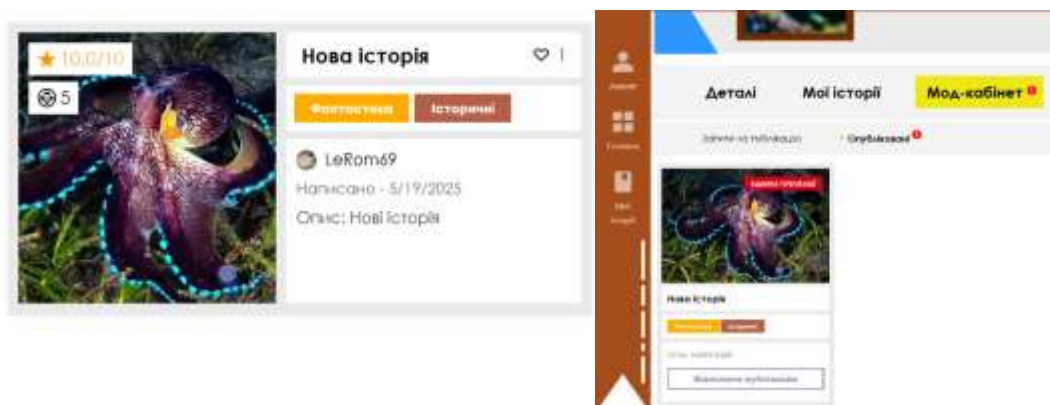


Рис. 3.48. Історія набрала забагато чорних міток, та буде відхилена від публікації.

Далі, буде протестовано функціонал видалення історії та акаунта. Він буде проілюстрований на рис. 3.49. та 3.50.

[illegible]

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 76   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |



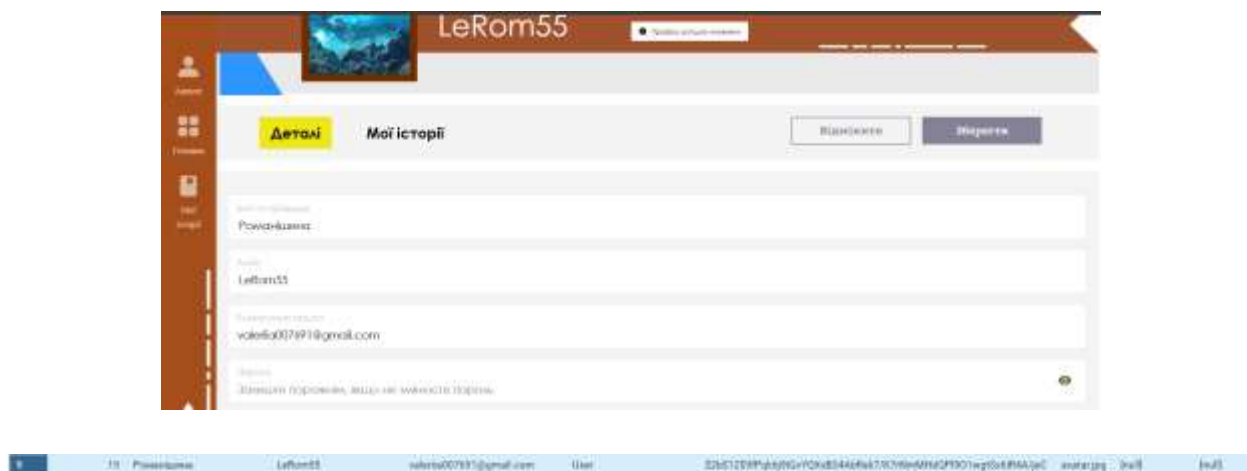


Рис. 3.51. Зміна даних акаунту

І останнім, що буде протестовано, буде скидання пароля. Цей функціонал був позроблений таким чином, що коли користувач вводить свою пошту, йому туди приходить лист з посиланням на сторінку, де і відбувається скидання пароля, для більшої безпеки для цього створюється тимчасовий унікальний токен, який після використання та скидання пароля видаляється, що не дасть повторно його використати.

Як працює цей функціонал, проілюстровано на рис. 3.52.

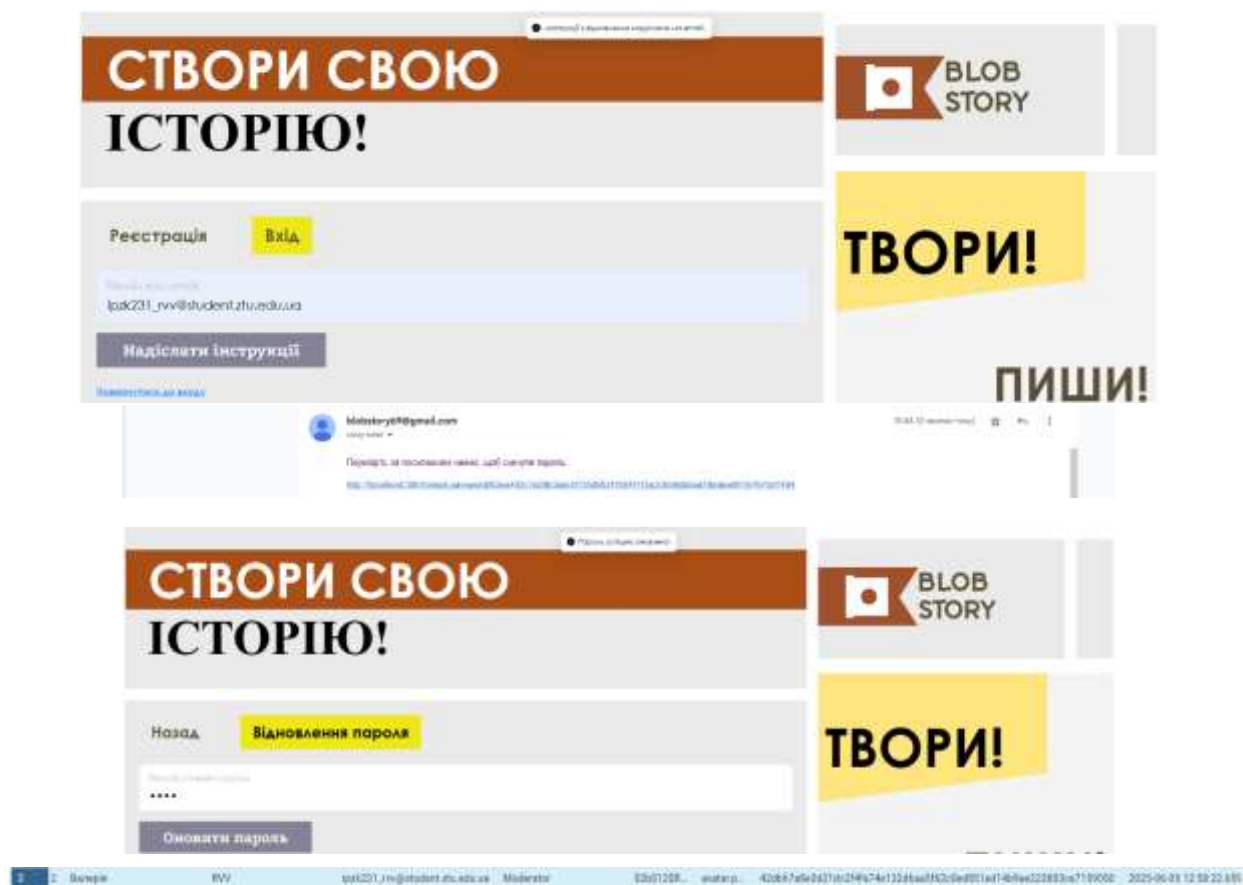


Рис. 3.52. Скидання пароля

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 78   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |

Отже, повністю протестувавши весь функціонал програми, можна сказати, що вона працює справно та відповідає всім заздалегідь висунутим вимогам щодо функціоналу.

### Висновки до третього розділу

Отже, у розділі було наведено схема розгортання застосунку з її детальним описом, що дозволило краще зрозуміти архітектуру системи та взаємозв'язки між її компонентами. Далі було надано опис та ілюстрації інтерфейсу розробленого програмного продукту. Були розглянуті всі функціональні можливості застосунку, залежно від ролі користувача та проілюстровано кожну з них, що показує що кожна роль отримала свої функції, а також, що у них є ідентичні можливості роботи з застосунком, які не прив'язані до ролі, такі як створення історій та їх прочитання. Було показано, як адміністратор виконує керування додатком та як виконується модерація історій, що дало більш прозоре розуміння роботи додатку. Також додаток був повністю протестований, що показало що він виконує всі функції, які були задані на початку проектування.

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 |      |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 | 79   |

## ВИСНОВКИ

Під час виконання кваліфікаційної роботи бакалавра було спроектовано та реалізовано застосунок для створення мультимедійних історій, проведено аналіз предметної області та порівняно різні застосунки, які використовуються для створення історій та статей.

На основі цього аналізу були визначені основні вимоги до застосунку. В процесі дослідження була розроблена архітектура, підходи та основні технології реалізації веб-застосунку. Проектування застосунку включало аналіз варіантів використання програмного продукту та проектування структури, включаючи моделювання взаємодії між складовими частинами.

Також було спроектовано базу даних, яка могла виконувати всі поставлені задачі, такі як: збереження даних про історії; збереження статистики оцінювання історій та кількість вподобайок до кожної; збереження коментарів користувачів та реакції інших юзерів на них; інформацію про користувачів та їх облікові записи. Описано алгоритм роботи з застосунком та варіанти його використання кожною роллю, яка була описана. Розглянуто основний функціонал, а саме: створення облікового запису, створення та редагування історій; створення коментарів та взаємодія з ними; взаємодія з історіями шляхом їх оцінювання; принцип виконання модерації історій та функції адміністратора. Також була надана схема розгортання платформи застосунку, описано взаємодію структурних елементів системи.

Також було описано і детально показано інтерфейс застосунку, що продемонструвало те, що він має зручний та інтуїтивно зрозумілий інтерфейс, який дозволяє користувачам легко створювати історії, редагувати їх. Також були показані мод-панель, де модератор може виконувати модерацію історій та адмін панель, де адміністратор може виконувати свої обов'язки, такі як зміна ролей користувачів та назначення модераторів на історії, що очікують перевірку.

Тому можна сказати, що мета кваліфікаційної роботи виконана, а і ідеї реалізовані.

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 80   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Швець О. Занурення в патерни проектування. 2022. 396 с.
2. Фленаган Д. JavaScript. Детальне керівництво. Дніпро : Символ-Плюс, 2021. 1080 с.
3. Wieruch R. The Road to React: Your journey to master plain yet pragmatic React.js. 2018. 286
4. Stoyan S. React: Up & Running Building Web Applications. 2021. 233 с.
5. Яциченко А. Інструкція Figma. *Scribd. Document Information*. URL: <http://scribd.com/document/702617111/%D0%86%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%86%D1%96%D1%8F-Figma>
6. Figma Learn. Learn design with Figma. URL: <https://www.figma.com/resource-library/design-basics/>.
7. GitBook Documentation. Gitbook. GitBook Documentation. URL: <https://gitbook.com/docs/>.
8. CSS Властивості. W3schoolsua. CSS Довідник. URL: <https://w3schoolsua.github.io/cssref/index.html#gsc.tab=0>.
9. Довідник по HTML тегам. Css.in.ua. Довідник по HTML тегам. URL: <https://css.in.ua/html/tags>.
10. Node.js / JavaScript. Visual Studio Code. Visual Studio Code documentation. URL: <https://code.visualstudio.com/docs/nodejs/working-with-javascript>.
11. React Reference Overview. React Official Docs. React Reference Overview. URL: <https://react.dev/reference/react>.
12. Node.js. Introduction to Node.js. URL: <https://nodejs.org/en/learn/getting-started/introduction-to-nodejs>.
13. Node.js. Node.js v24.1.0 documentation. URL: <https://nodejs.org/docs/latest/api/>.
14. Express.js Official Site. Guide. URL: <https://expressjs.com/>.
15. W3schoolsUA. Українською. Node.js Підручник. URL: <https://w3schoolsua.github.io/nodejs/index.html#gsc.tab=0>.

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 81   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |

16. *Git Extensions*. URL: <https://gitextensions.github.io/>.
17. *Github*. *Gitextensions* *wiki*. URL: <https://github.com/gitextensions/gitextensions/wiki>.
18. *DigitalOcean*. *How To Use Express*. URL: <https://docs.digitalocean.com/>.
19. *Mozilla Developer Network*. *JavaScript Guide*. URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>.
20. *MDN Web Docs*. *DOM Reference*. URL: [https://developer.mozilla.org/en-US/docs/Web/API/Document\\_Object\\_Model](https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model).
21. *W3Schools*. *PostgreSQL Tutorial*. URL: <https://www.w3schools.com/postgresql/index.php>.
22. *Harness*. *Introduction to Building a CRUD API with Node.js and Express*. URL: <https://www.harness.io/blog/introduction-to-building-a-crud-api-with-node-js-and-express>.
23. *Pgadmin*. *Documentation*. URL: <https://www.pgadmin.org/docs/>.
24. *Node-postgres*. URL: <https://node-postgres.com/>.
25. *Postman*. *Postman documentation overview*. URL: <https://learning.postman.com/docs/introduction/overview/>.
26. *Canada Health Infoway*. *Postman Instructions*. URL: <https://infocentral.infoway-inforoute.ca/en/postman-documentation>.
27. *Tutorialspoint*. *Design Pattern Quick Guide*. URL: [https://www.tutorialspoint.com/design\\_pattern/design\\_pattern\\_quick\\_guide.htm](https://www.tutorialspoint.com/design_pattern/design_pattern_quick_guide.htm).
28. *Lucidchart*. *Component Diagram Tutorial*. URL: <https://www.lucidchart.com/pages/uml-component-diagram>.
29. *Uml*. *Useful Links to UML resources*. URL: <https://www.uml.org/resource-hub.htm?form=MG0AV3>.
30. *Mindonmap*. *Повне розуміння діаграми компонентів UML за допомогою легкого методу*. URL: <https://www.mindonmap.com/uk/blog/uml-component-diagram/>.

|      |      |                  |        |      |                                                  |      |
|------|------|------------------|--------|------|--------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка». 24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                  | 82   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                  |      |

# ДОДАТКИ

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 |      |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 | 83   |

## Код компонента навігації

```

import { useState, useEffect } from "react";
import { Link } from "react-router-dom";
import { useAuth } from "../contexts/authContext";

const NavBookmark = () => {
  const [isOpen, setIsOpen] = useState(false);
  const [isSelect, setIsSelect] = useState(false);
  const [projects, setProjects] = useState([]);
  const { userData } = useAuth();

  const getAccountLink = () => {
    const role = userData?.role?.toLowerCase();
    if (role === "admin") return "/admin";
    if (role === "moderator") return "/moderator";
    return "/youraccount";
  };

  useEffect(() => {
    const fetchProjects = async () => {
      try {
        if (!userData?.login) return;

        const response = await
fetch(`http://localhost:3000/api/history/user/${userData.login}`);
        const text = await response.text();

        try {
          const data = JSON.parse(text);
          setProjects(data);
        } catch (jsonErr) {
          console.error("JSON parse error:", jsonErr, "\nОтримано:",
text);
        }
      } catch (err) {
        console.error("Помилка запиту:", err);
      }
    };

    fetchProjects();
  }, [userData?.login]);

  return (
    <div className={`bookmark ${isOpen ? "active" : ""}`}>
      <div className="navigation-wrapper">
        <Link to={getAccountLink()}>
          <div className="nav-element">
            
            Акаунт

```

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 84   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |

```

        </div>
</Link>

<Link to="/main">
    <div className="nav-element">
        
        Головна
    </div>
</Link>

<div
    onClick={() => {
        setIsOpen(!isOpen);
        setIsSelect(!isSelect);
    }}
    className={`nav-element ${isSelect ? "active" : ""}`}
>
    
    Мої історії
</div>
</div>

<div className="dots">
    <p>Почніть ваш день з нами</p>
</div>

<div
    className={`popup-overlay ${isOpen ? "active" : ""}`}
    onClick={() => setIsOpen(false)}
>
    <div className="popup-content" onClick={(e) =>
e.stopPropagation()}>
        <h1>Робочий стіл:</h1>
        <Link to="/creator">
            <button className="new-story-btn" >+ Створити нову
історію</button>
        </Link>

        <h2>Минулі проєкти:</h2>
        <div className="storylist-line"></div>
        <ul className={`storylist-container ${projects.length > 10 ?
"with-scroll" : ""}`}>
            {projects.length > 0 ? (
                projects.map((project, index) => (
                    <li className="storylist-item" key={index}
onClick={() => {
                                window.location.href =
`/creator/${project.folder_path}/`;
                            }}>
                        {project.history_title}
                    </li>
                )
            ) : null}
        </ul>
    </div>
</div>

```

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 85   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |

```

        ))
      ) : (
        <li className="storylist-item">Історій не
знайдено</li>
      )}
    </ul>
  </div>
</div>
</div>
);
};

export default NavBookmark;

```

Додаток Б

### Код серверної частини

```

const express = require('express');
const cors = require('cors');
const path = require('path');
const pool = require('./utils/db');

const uploadController = require('./controllers/uploadController');
const setStoryModeratorController =
require('./routes/setStoryModeratorController');

const authRouter = require('./routes/authRoute');
const userRoutes = require('./routes/userRoutes');
const historyRoutes = require('./routes/historyRoute');
const likesRouter = require('./routes/likesRoute');
const commentRoutes = require('./routes/commentsRoute');
const commentLikeRoutes = require('./routes/commentLikesRoute');
const rateRoutes = require('./routes/rateRoute');
const storyRoutes = require('./routes/storyRoutes');

const app = express();

// MIDDLEWARES
app.use(cors());
app.use(express.json());

// Статичні файли
app.use(express.static(path.join(__dirname, 'public')));
app.use('/stories', express.static(path.join(__dirname, 'stories')));
app.use('/userAvatars', express.static(path.join(__dirname, 'userAvatars')));

// API ROUTES
app.use('/api/auth', authRouter);
app.use('/api/users', userRoutes);
app.use('/api/history', historyRoutes);
app.use('/api/likes', likesRouter);
app.use('/api/comments', commentRoutes);

```

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 86   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |

```

app.use('/api/comments/likes', commentLikeRoutes);
app.use('/api/admin', setStoryModeratorController);
app.use("/api/rate", rateRoutes);
app.use('/api/loadStoryJson', storyRoutes);

// GLOBAL ERROR HANDLER
app.use((err, req, res, next) => {
  err.statusCode = err.statusCode || 500;
  err.status = err.status || 'error';

  res.status(err.statusCode).json({
    status: err.status,
    message: err.message,
  });
});

pool.query('SELECT NOW()', (err, res) => {
  if (err) {
    console.error('Database query error', err.stack);
  } else {
    console.log('Database connected at', res.rows[0].now);
  }
});

app.use('/api/creator', uploadController);

// SERVER
const PORT = process.env.PORT || 3000;
app.listen(PORT, () => {
  console.log(`Server running on port ${PORT}`);
});

```

Додаток В

### Код клієнтської частини додатку

```

import React from 'react';
import ReactDOM from 'react-dom/client';
import { BrowserRouter as Router, Routes, Route, Navigate } from 'react-router-dom';

import './style/register-authorize.css';
import './style/main-page.css';
import './style/history-info.css';
import './style/accounts-style.css';
import './style/constructor-styles.css';
import './index.css';

import Register from './pages/authorization/Register.jsx';
import Login from './pages/authorization/Login.jsx';
import ResetPassword from './pages/authorization/ResetPassword.jsx';
import MainPage from './pages/common/Main.jsx';

```

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 87   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |

```

import HistoryInfo from './pages/common/HistoryInfo.jsx';
import UserAccount from './pages/user/UserAccount.jsx';
import AdminAccount from './pages/administrator/AdminAccount.jsx';
import ModeratorAccount from './pages/moderator/ModeratorAccount.jsx';
import StoryConstructor from './pages/common/Constructor.jsx';

import { AuthProvider, useAuth } from './contexts/authContext';

const PrivateRoute = ({ element }) => {
  const { isAuthenticated } = useAuth();
  return isAuthenticated ? element : <Navigate to="/login" />;
};

const PublicRoute = ({ element }) => {
  const { isAuthenticated } = useAuth();
  return !isAuthenticated ? element : <Navigate to="/main" />;
};

const RoleBasedRoute = ({ allowedRoles, element }) => {
  const { isAuthenticated, userData } = useAuth();

  if (!isAuthenticated) {
    console.log("Not authenticated");
    return <Navigate to="/login" />;
  }

  console.log("Authenticated user:", userData);

  const userRole = userData?.role?.toLowerCase();
  const allowed = allowedRoles.map(role => role.toLowerCase());

  if (!allowed.includes(userRole)) {
    console.log(`Access denied. User role: ${userRole}, Allowed: ${allowed}`);
    return <Navigate to="/main" />;
  }

  return element;
};

const App = () => {
  const { isLoading } = useAuth();

  if (isLoading) {
    return <div>Завантаження...</div>;
  }

  return (
    <Router>
      <Routes>
        { /* Публічні маршрути */ }
        <Route path="/" element={ <PublicRoute element={ <Register /> } /> } />

```

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 88   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |

```

<Route path="/login" element={<PublicRoute element={<Login />} />} />
<Route path="/reset-password/:token" element={<ResetPassword />} />

{/* Приватні маршрути */}
<Route path="/main" element={<PrivateRoute element={<MainPage />} />} />
<Route path="/historyinfo/:id" element={<PrivateRoute
element={<HistoryInfo />} />} />

{/* Акаунти за ролями */}
<Route
  path="/youraccount"
  element={
    <RoleBasedRoute allowedRoles={['user']} element={<UserAccount />} />
  }
/>
<Route
  path="/moderator"
  element={
    <RoleBasedRoute allowedRoles={['moderator']}
element={<ModeratorAccount />} />
  }
/>
<Route
  path="/admin"
  element={
    <RoleBasedRoute allowedRoles={['admin']} element={<AdminAccount />}
/>
  }
/>

{/* Сторінка конструктора */}
<Route path="/creator" element={<PrivateRoute element={<StoryConstructor
key="new" />} />} />
<Route path="/creator/stories/:storyTitle" element={<PrivateRoute
element={<StoryConstructor />} />} />

<Route path="*" element={<Navigate to="/" />} />
</Routes>
</Router>
);
};

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <AuthProvider>
      <App />
    </AuthProvider>
  </React.StrictMode>
);

```

|      |      |                  |        |      |                                                 |      |
|------|------|------------------|--------|------|-------------------------------------------------|------|
|      |      | Романішина В.В.  |        |      | ДУ «Житомирська політехніка».24.121.19.000 - ПЗ | Арк. |
|      |      | Годлевський Ю.О. |        |      |                                                 | 89   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                                                 |      |