DOI: https://doi.org/10.26642/ten-2024-2(94)-136-143
УДК 004.7

С.М. Кравченко, ст. викладач
О.М. Свінцицька, к.е.н., доц.
О.В. Кузьменко, ст. викладач
І.А. Толстой, ст. викладач
Державний університет «Житомирська політехніка»

Інформаційна система для онлайн-платформи письменників і читачів

На даний час дедалі більше популярними стають онлайн-платформи для письменників та читачів, які є досить новими для ринку цифрової електронної комерції. Основна мета створення такої платформи – надати багатьом письменникам, які не розкрили себе, або можливо не впевнені, що їхні твори сподобаються іншим, можливість спробувати написати свою роботу та дозволити оцінити книгу іншим. Тому в статті представлено процес проєктування додатка та етапи моделювання процесу. Вебзастосунок призначений для публікації своїх робіт письменниками, зрозумілий інтерфейс для читачів, багатофункціональна адмінпанель для ефективної роботи бек-офісу, зручний дашборд для моніторингу роботи девелоперського та продакшн серверів, системи для відслідковування помилок, система управління базами даних (СУБД) для нереляційних та реляційних БД, які будуть використовуються в додатку та оркестраторів Docker контейнерів. В роботі проведено аналіз напрямів використання системи, визначено архітектуру онлайн-платформи, представлено алгоритм роботи додатка. Визначено архітектуру онлайн-платформи такої інформаційної системи, де фреймворк заснований на Enterprise патерні MVC Laravel. PostgreSQL як основна – реляційна БД, Redis як нереляційна БД для кешування, Jquery для реалізації frontend, Elastic Search як пошуковий двигун, RabbitMQ для реалізації системи черг в додатку, NodeJS в поєднанні з Laravel Echo для real time функціональності (чат, сповіщення), Stripe – платіжна система. Такий вебдодаток пройшов функціональне і нефункціональне тестування і готовий для використання.

Ключові слова: *онлайн-платформа; письменники та читачі; Laravel; NodeJS; інформаційна система.*

Актуальність теми. У сучасному світі досить популярною є тема онлайн-платформ, оскільки можна охопити досить велику аудиторію, надати простір людям з різними точками зору. Користувачі інтернету, які хочуть поділитись інформацією, звикли до того типу подачі, який нав'язаний їм блогами та соціальними мережами, обидва варіанти подачі інформації є досить зручними, але вони ставлять письменника в жорсткі рамки:

- часто соціальні мережі обмежують кількість символів;
- для того, щоб почати вести блог, його треба створити, що майже неможливо для некваліфікованого користувача;
- під час використання соціальних мереж та блогів, досить тяжко заманити читачів до написаних вами робіт;
- рідко трапляється зрозуміла система монетизації для письменника.

Цифрові технології, нові способи відпочинку та дозвілля не знищили культуру читання, навіть більше, вони подарували нові можливості для комунікації та взаємодії між людьми. І як виявилось, ця взаємодія може приносити не лише задоволення, але й кошти.

Наразі платформи для читання є досить новими на ринку електронної комерції, тому розробка, покращення та система інтеграції цих платформ є актуальними на даний час. Основною проблемою нинішніх, уже розроблених платформ для читання є те, що вони досить «сирі» для сучасного споживача. Хоч і соціальні мережі та блоги дають можливість розкрити талант письменника в собі, але почати монетизувати свою роботу досить складно.

Аналіз останніх досліджень та публікацій. Під час дослідження створення онлайн-платформи для письменників і читачів доцільно буде проаналізовати подібні платформи, які вже використовуються. В цілому є всього декілька відомих аналогів-додатків, де читачі зможуть читати та публікувати власні роботи. Наприклад, сайт «inkitt.com» [6]. Цей вебсайт не обмежує доступу до книг, в ньому є розподіл книг за категорією, що дає можливість легко знайти ту чи іншу книгу.

Другий популярний аналог «wattpad.com» [7]. Додаток дає можливість написати та прочитати книги, але ми не можемо побачити хоч частину каталогу, оскільки ми не зареєстровані, що є не дуже зручно. Wattpad є найбільш поширеною платформою для читання, але вона не надає можливості письменникам заробляти кошти на своїй праці, в такому вигляді, до якого звик сучасний споживач (місячні / річні пакети, зрозуміле донесення інформації про те, як почати заробляти кошти, гнучка система монетизації).

 © С.М. Кравченко, О.М. Свінцицька, О.В. Кузьменко, І.А. Толстой, 2024

Метою статті є доцільність використання моделювання процесів на стадії проєктування онлайн-платформи, визначення архітектури та інструментальних засобів під час реалізації вебдодатка.

Викладення основного матеріалу. В роботі розглядається можливість розробити інформаційну систему для онлайн-платформи, передбачити засоби управління інформаційним контентом, модуль для пошуку книг / авторів, модуль для управління додатком та модуль оплати.

Під час розробки веб-додатка було поставлено завдання, для реалізації якого основними етапами є:

1. Створення списку завдань у системі управління проєктами;
2. Розробка дизайну клієнтської частини сайту;
3. Реалізація ядра системи управління контентом;
4. Реалізація структури збереження даних.

Результатом реалізації поставленого завдання є реалізована онлайн-платформа для письменників та читачів, що містить в собі серверну структуру збереження даних, багатокористувацький клієнтський додаток для реалізації функціональних можливостей і засоби контролю та керування доступом.

Для цього використовується мова програмування PHP 8.1, фреймворк, оснований на Enterprise патерні MVC Laravel, Docker, Docker Compose для оркестровки контейнерів на девелоперському та локальному середовищах, PostgreSQL як основна – реляційна БД, Redis як нереляційна БД для кешування, Jquery для реалізації frontend, Elastic Search як пошуковий двигун, RabbitMQ для реалізації системи черг у додатку, NodeJS [4, 10] у поєднанні з Laravel Echo для real time функціональності (чат, сповіщення), Stripe – платіжна система.

Для реалізації веборієнтованої онлайн-платформи для письменників та читачів було обрано архітектурний патерн MVC, який вбудовано в обраний фреймворк Laravel [5, 8]. Цей архітектурний шаблон забезпечує всі технологічні можливості для побудови гнучкого додатка із дотриманням принципів SOLID. Доцільно використовувати мову програмування PHP 8 [1, 2], яка має декілька переваг: швидкодія, нові можливості, типізація, зменшення використання ресурсів порівняно з попередніми версіями, оновлення бібліотек та інструментів.

Як базу даних було обрано використовувати PostgreSQL [3, 9], оскільки однією з ключових переваг є його розширена система типів даних та можливість визначення користувацьких типів. Це дозволяє більш гнучко моделювати дані та реалізовувати складні структури даних. Крім того, PostgreSQL має потужну систему транзакцій і підтримує вищий рівень стандартів SQL, що робить його більш надійним та відповідним для багатьох типів додатків.

Аналіз вимог користувачів дав можливість визначити варіанти використання платформи для письменників та читачів.

Вимоги користувачів

Користувачі мають доступ до таких функцій:

1. Можливість читати та переглядати книги;
2. Можливість коментувати;
3. Можливість створювати особистий профіль;
4. Можливість взаємодіяти з авторами;
5. Можливість створювати списки читань, тобто створювати публічні бібліотеки;
6. Можливість пошуку та фільтрації контенту;
7. Можливість отримувати сповіщення;
8. Можливість написати та опублікувати книги.

Користувачі бек-офісу – адміністратори сайту, відділ підтримки, модератор – мають доступ до таких функцій:

1. Можливість керувати користувачами;
2. Можливість управляти контентом;
3. Можливість вести аналітику;
4. Можливість керувати ресурсами;
5. Можливість контролювати систему безпеки та авторизацію.

Функціональні вимоги:

1. Реєстрація на платформі;
2. Можливість збереження інформації;
3. Можливість написання книги.

На рисунку 1 представлено реалізацію алгоритмів роботи додатка за допомогою UML-діаграми діяльності для написання книг.

Зазначена діаграма (рис. 1) моделює взаємодію автора з системою під час творчого процесу написання книги на платформі.

Етапи діаграми: спочатку користувач бачить перед собою сторінку створення книги, потім йде перевірка на те, чи зареєстрований користувач, якщо ні – користувача відправляє на сторінку авторизації, якщо користувач зареєстрований – він має отримати доступ на написання книг від імені видавництва.

Якщо видавництво відмовило користувачу, то приходить сповіщення про те, що видавництво відмовило, тому доступу до написання книг немає, потрібно вибрати інше видавництво. Якщо видавництво підтвердило користувача, то перед ним з'являється форма створення книги, після введення даних йде валідація форми, якщо форма не валідна, то показуємо форму знову, для вводу нових даних, якщо валідна, то перед собою користувач бачить сторінку для написання вмісту книги, після того користувач може опублікувати її.

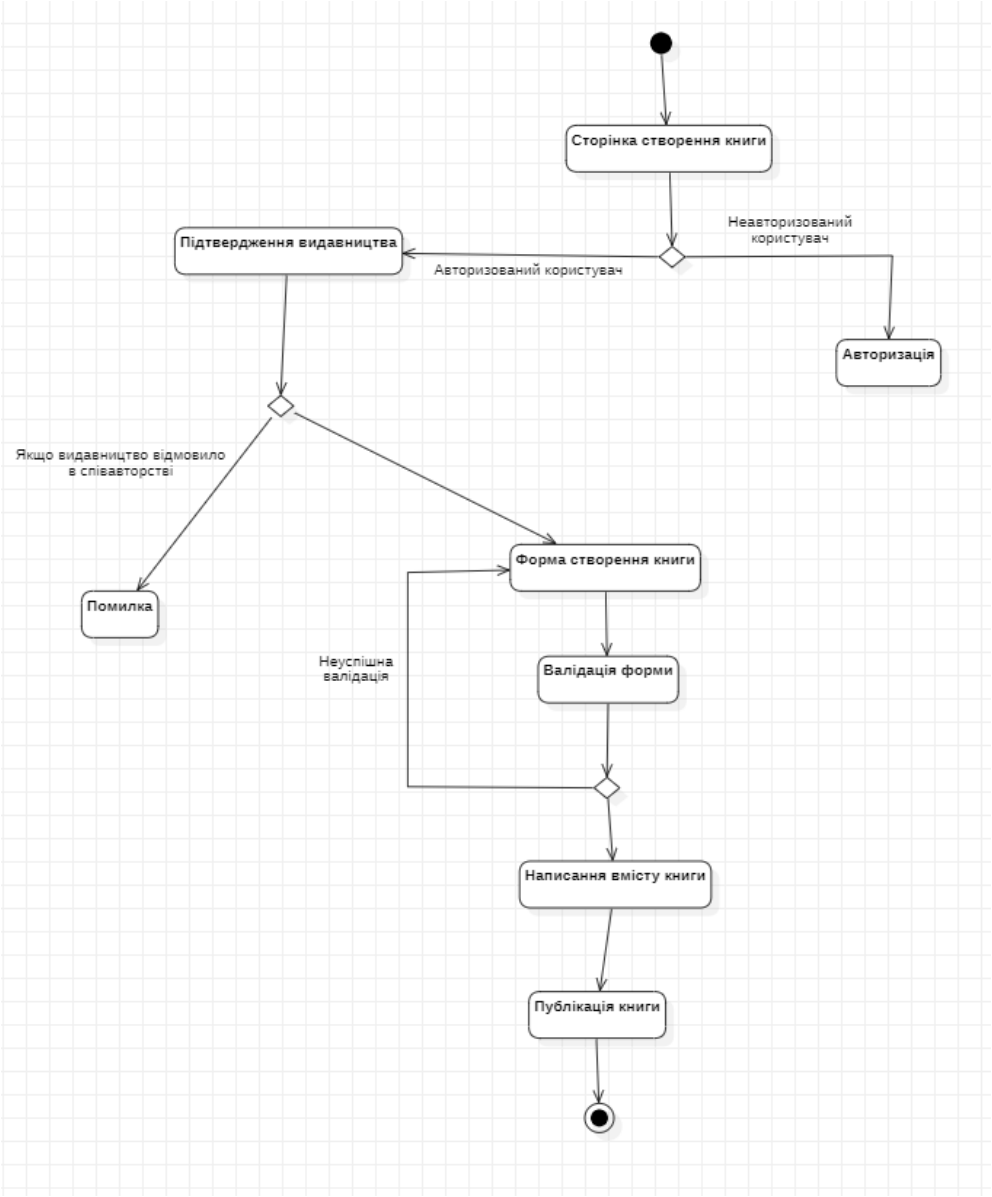

Рис. 1. Діаграма діяльності створення книги

Маючи варіанти використання платформи, тепер можна побудувати діаграму класів. Як згадувалося раніше, як архітектура буде використовуватися модель MVC, що нагадає змогу описувати для кожної сутності майже однакові класи, які по суті відрізняються лише властивостями та операціями.

Кожна сутність буде мати свій контролер, модель та набір представлень. Також для того, щоб тримати свій код в чистоті, буде додано сервісний шар. До того ж використовуються потужності Laravel, щоб оброблювати форму (валідація, попередня обробка даних), використовуючи FormRequests.

Обробка запиту користувача до сутності книги (рис. 2) починається зі звернення до контролера, який оброблює запит користувача для підготовки даних для передачі в сервіс. Сервіс сутностей книг своєю чергою підготовлює дані до запису в БД, використовуючи модель, після чого модель повертає результат запису в БД назад до контролера.

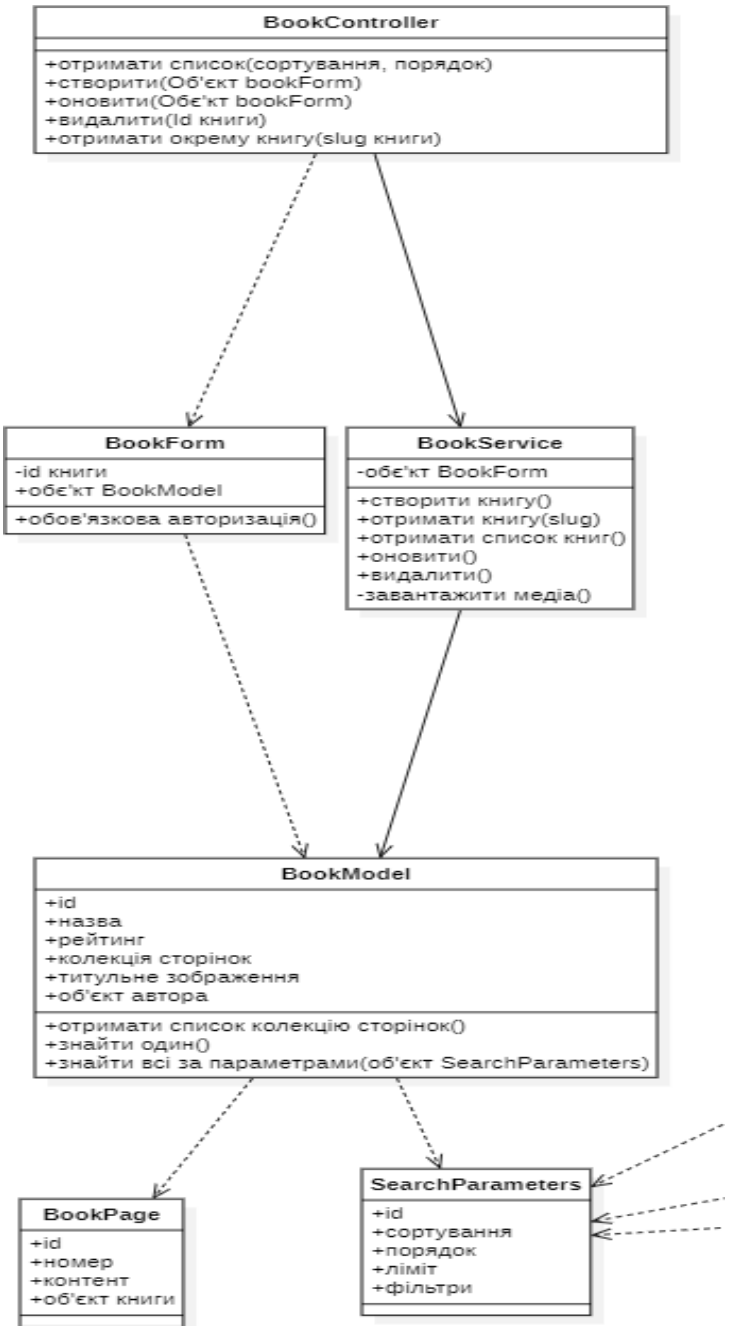

Рис. 2. Діаграма класів сутності книги

Також в сутності книги є модель BookPage, що відповідає за контент книги, в сутності сторінки таких зв’язків немає. Діаграма класів сутності користувач показано на рисунку 3.

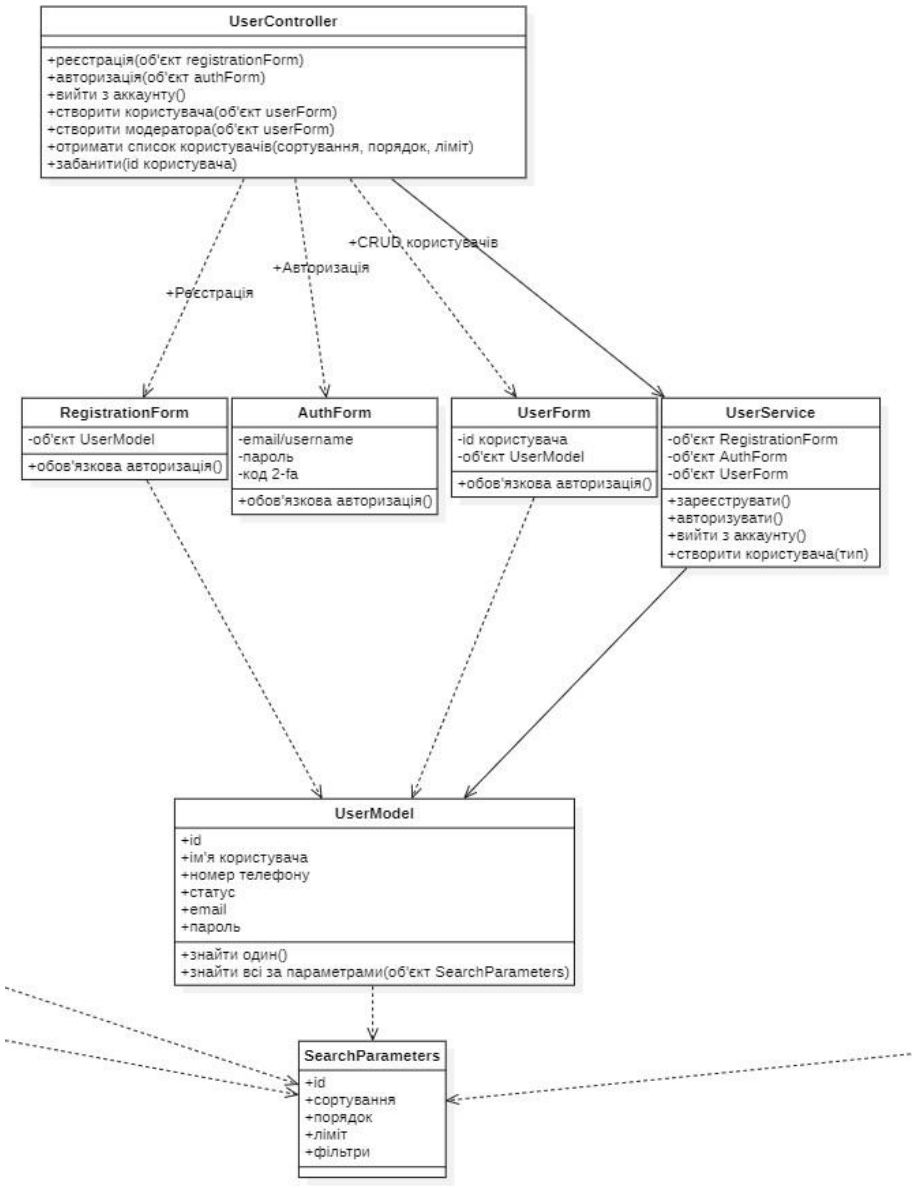

Рис. 3. Діаграма класів сутності користувач

Як СУБД було обрано PostgreSQL, через її надійність та стійкість. Також PostgreSQL за замовчуванням має підтримку InnoDB, що робить її привабливішою над її головним конкурентом MySQL. Також PostgreSQL має відкритий код, через що постійно оновлюється, дозволяє уникати витрат на ліцензію та мати повний контроль над кодом. База даних додатка складається з 24 таблиць (рис. 4).

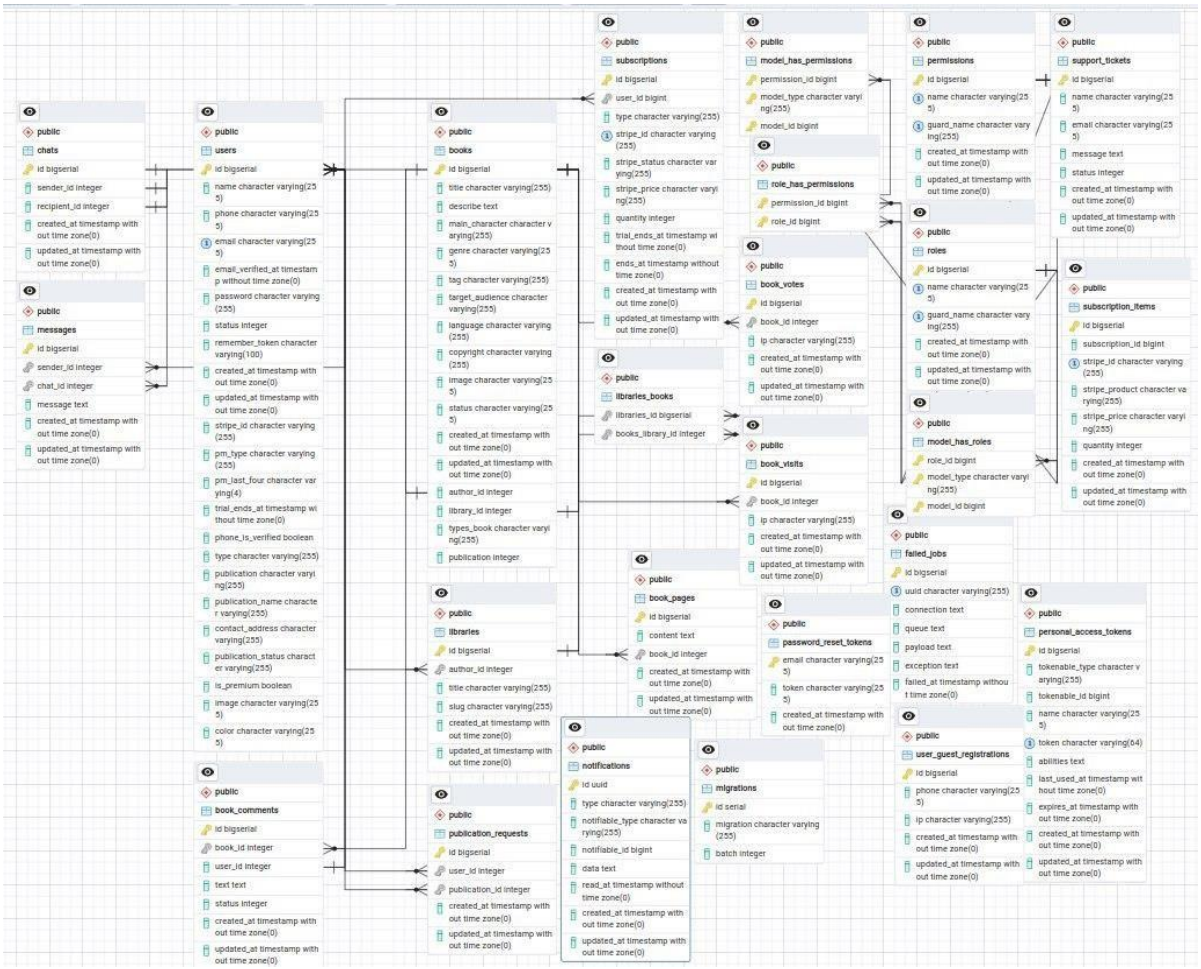

Рис. 4. Діаграма БД

Додаток складається з низки компонентів, для розгортання яких використовується Docker Compose. Використовуючи такий підхід, користувач має можливість розгорнути будь-яку кількість компонентів системи, використавши одну консольну команду.

Лістинг команди запуску

docker композиції: docker compose up –d

Для налаштування окремого компонента системи використовується Dockerfile, в якому прописані Linux-команди потрібні для налаштування компонента. Як приклад, компонент php якому для розгортання потрібен встановлений composer та низка php-модулів.

Лістинг коду Dockerfile:

```
FROM composer:2.2 AS composer-image
FROM php:8.2-fpm AS app
RUN apt-get update --fix-missing
RUN apt-get -y install postgresql
RUN apt-get -y install libpq-dev
RUN docker-php-ext-install pdo pdo_pgsql
RUN docker-php-ext-install sockets
RUN pecl install -o -f redis \
 && rm -rf /tmp/pear \
&& docker-php-ext-enable redis
COPY --from=composer-image /usr/bin/composer /usr/local/bin/composer
RUN apt-get install -y libmagickwand-dev --no-install-recommends \
&& pecl install imagick \
&& docker-php-ext-enable imagick \
&& apt-get install -y zlib1g-dev \
 && apt-get install -y libgmp-dev \
&& docker-php-ext-install gmp \
```

*&& docker-php-ext-enable gmp *
*&& apt-get install -y libtidy-dev libffi-dev libc-client-dev libkrb5-dev --no-install-recommends \ && docker-php-ext-configure imap --with-kerberos --with-imap-ssl *
*&& docker-php-ext-install -j$(nproc) imap *
*&& docker-php-ext-enable imap *
*&& apt-get install -y gcc make libssh2-1-dev libssh2-1 *
*&& pecl install ssh2 *
*&& docker-php-ext-enable ssh2 *
*&& apt-get install -y libmsgpack-dev *
&& docker-php-ext-install -j$(nproc) opcache

На рисунку 5 представлено головну сторінку додатка.

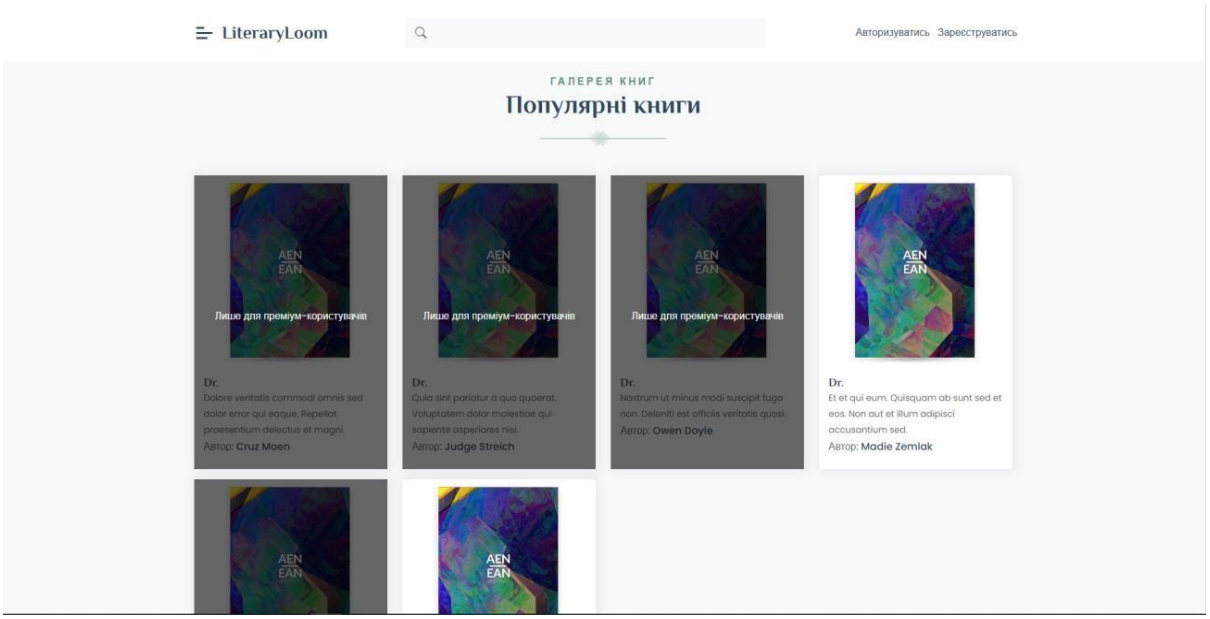

Рис. 5. Головна сторінка

Висновки та перспективи подальших досліджень. Під час досліджень було спроєктовано процес побудови додатка інформаційної системи, представлено моделювання етапів проєктування. Описано роботу платформи, поєднавши всі компоненти системи та описано їх. Інформаційна система такої онлайн-платформи може використовуватися для публікації своїх робіт письменниками та мати зрозумілий інтерфейс для читачів. Було використано сучасні засоби та технології, такі як: мова програмування PHP 8.1, фреймворк, оснований на Enterprise патерні MVC Laravel, Docker, Docker Compose для оркестровки контейнерів на девелоперському та локальному середовищах, PostgreSQL як основна – реляційна БД, Redis як нереляційна БД для кешування, Jquery для реалізації frontend, Elastic Search як пошуковий двигун, RabbitMQ для реалізації системи черг в додатку, NodeJS в поєднанні з Laravel Echo для real time функціональності (чат, сповіщення), Stripe – платіжна система. В подальшому цей проєкт може розвиватися з виявленням нового функціоналу, його реалізації та оновлення для всіх користувачів системи.

References:

1. Atencio, L. (2021), *Clean Code in PHP: Expert Tips and Techniques*, O‘Reilly, 240 p.
2. Bierer, D. and Evans, C. (2021), *PHP 8 Programming Tips, Tricks and Best Practices*, Packt, 528 p.
3. Shaik, B. and Kaur, M. (2016), *PostgreSQL Development Essentials*, Packt Publishing, Limited, 45 p.
4. Alves, C. (2021), *Node. js: NODEJS para Principiantes*, Independently Published, 197 p.
5. Staffer, M. (2019), *Laravel*, 2nd ed., O‘Reilly, 512 p.
6. *Inkitt*, [Online], available at: https://www.inkitt.com/
7. *Wattpad*, [Online], available at: https://www.wattpad.com/
8. *Laravel*, [Online], available at: https://laravel.com/docs/10.x
9. *PostgreSQL*, [Online], available at: https://www.postgresql.org/docs/16/index.html
10. «What Is Node.js? A Complete Guide for Developers», [Online], available at: https://www.simplilearn.com/tutorials/nodejs-tutorial/what-is-nodejs

Кравченко Світлана Миколаївна – старший викладач кафедри комп'ютерних наук Державного університету «Житомирська політехніка».

https://orcid.org/0000-0002-5895-9615.

Наукові інтереси:

- моделювання та аналіз програмного забезпечення;
- проєктування інтерфейсів;
- бази даних.

Свінцицька Олександра Миколаївна – кандидат економічних наук, доцент, доцент кафедри комп'ютерних наук Державного університету «Житомирська політехніка».

https://orcid.org/0000-0002-2613-2437.

Наукові інтереси:

- управління IT-проєктами;
- інформаційні системи і технології в креативних індустріях.

Кузьменко Олександр Вікторович – старший викладач кафедри комп'ютерних наук Державного університету «Житомирська політехніка».

https://orcid.org/0000-0002-4937-3284.

Наукові інтереси:

- інтерактивні навчальні курси;
- вебтехнології;
- розробка інформаційних систем.

Толстой Ігор Анатолійович – старший викладач кафедри ІПЗ Державного університету «Житомирська політехніка».

https://orcid.org/0000-0001-8879-8827.

Наукові інтереси:

- проєктний менеджмент;
- інформаційні системи;
- блокчейн-технології.

Kravchenko S.M., Svintsytska O.M., Kuzmenko O.V., Tolstoy I.A.

Information system for online platform for writers and readers

Nowadays, online platforms for writers and readers are becoming more and more popular, which are quite new to the digital e-commerce market. The main purpose of creating such a platform is to give many writers who have not revealed themselves, or perhaps are not sure that their works will be liked by others, the opportunity to try to write their work and to give the opportunity for others to evaluate the book. Therefore, this article presents the application design process and process modeling stages. The web application is designed for writers to publish their works, a clear interface for readers, a multifunctional admin panel for efficient back-office work, a convenient dashboard for monitoring the work of development and production servers, a system for tracking errors, DBMS for non-relational and relational databases that will be used in the application and Docker container orchestrators. In the paper, an analysis of the system's usage directions was carried out, the architecture of the online platform was determined, and the algorithm of the application was presented. The architecture of the online platform is defined, the framework is based on MVC Laravel Enterprise pattern. PostgreSQL as the main relational database, Redis as a non-relational database for caching, Jquery for frontend implementation, Elastic Search as a search engine, RabbitMQ for implementing the queue system in the application, NodeJS in combination with Laravel Echo for real-time functionality (chat, notifications), Stripe is a payment system. This web application has passed functional and non-functional testing and is ready for use.

Keywords: online platform; writers and readers; Laravel; NodeJS; information system.

Стаття надійшла до редакції 11.10.2024.