

УДК 004.738.5

*Козлик С.О., здобувач,
Фант М.О., к.філол.н., доцент
Державний університет «Житомирська політехніка»*

ТЕХНОЛОГІЧНИЙ СТЕК ДЛЯ СИСТЕМИ КЕРУВАННЯ ПРОЦЕСАМИ ВИДАВНИЦТВА

У робочих процесах видавництва книг застосовують велику кількість різноманітного програмного забезпечення для вирішення варіативних задач, наприклад, для написання, редагування та форматування текстів книги використовують Microsoft Word або Scrivener, для верстки макету до друку – Affinity Publisher чи Adobe InDesign, а для управління проектами, відслідковування таймлайнів та надання завдань працівникам – Trello, Asana тощо. Така різноманітність інструментів ускладнює роботу над проектом, потребує багато часу на навчання користування ними. Тому було вирішено розробити єдину систему керування процесами видавництва, яка зекономить час та підвищить ефективність роботи у видавництві.

Для реалізації даного застосунку було обрано сучасний технологічний стек, який відповідає актуальним тенденціям розробки, орієнтований на роботу зі значними обсягами текстових матеріалів та забезпечує зручність розробки, підтримки і розширення системи.

У процесі розробки системи керування процесами видавництва було обрано клієнт-серверну архітектуру [1], в якій клієнтська частина взаємодіє з серверною, передаючи запити користувача та отримуючи сформовані на них відповіді від сервера.

Серверна частина застосунку побудована на ASP.NET Core Web API [3]. Він є кросплатформенним, має легку та модульну структуру, зручну маршрутизацію, легко інтегрується з сучасними інструментами для роботи з базами даних, що робить його гарним та надійним варіантом для створення REST API.

Для автентифікації та авторизації у системі використовується ASP.NET Core Identity у поєднанні з JWT, завдяки якому відбувається безпечна передача та перевірка даних користувачів без потреби збереження стану на сервері. Identity відповідає за керування та перевірку особистих даних, відновлення доступу та надає можливість розширювати стандартні моделі і встановлювати ролі для користувачів. Інформацію про користувача та його роль JWT передає у вигляді claims під час взаємодії клієнтської та серверної частин.

Для збереження інформації системи було обрано СУБД PostgreSQL, оскільки вона підтримує транзакції, розширення, індекси та паралельне

виконання запитів, а також забезпечує стабільність і надійне зберігання даних. Її пропонується використовувати для збереження наступної інформації: дані користувачів, видавництв; створені проекти, пов'язані з ними книги, матеріали; встановлені для проектів етапи, додані до них завдання.

Для взаємодії з базою даних використовується Entity Framework Core, яка має об'єктно-орієнтований підхід, гнучку конфігурацію, підтримує створення та виконання міграцій.

Текстові та графічні матеріали вирішено зберігати не в базі даних, а завантажувати у хмарне сховище Amazon S3, тому що це дозволить покращити продуктивність БД. Також хмарні сховища оптимізовані для зберігання файлів великих обсягів, забезпечують автоматичне резервне копіювання та відновлення файлів у разі їхньої втрати чи пошкодження, мають низьку вартість зберігання. У БД будуть зберігатися лише посилання на файли, дати створення, а їх завантаження та видалення буде здійснюватися через API хмарного сервісу.

Для написання інтерфейсу користувача було вирішено застосувати бібліотеку React [2]. Вона має компонентно-орієнтований підхід, тобто React дозволяє створювати та перевикористовувати незалежні компоненти при побудові UI. Також він дозволяє створити віртуальний DOM, тобто спрощену копію DOM; завдяки цьому реальний DOM буде оновлювати і відображати лише ті змінені частини, які виявлені у віртуальному DOM. На відміну від Angular, в React компоненти можна створювати без використання класів, застосовуючи лише функції; дані в них передаються зверху вниз, тобто від батьківського компонента до дочірніх, що зменшує можливість виникнення будь-яких неочікуваних змін стану.

Таким чином, було обрано технологічний стек для розробки системи керування процесами видавництва, який забезпечить гнучкість, масштабованість, безпеку та продуктивність кінцевого продукту. У сукупності ці технології повністю відповідають вимогам системи для реалізації її функціоналу.

Список використаних джерел:

1. Bass L., Clements P., Kazman R. Software Architecture in Practice. 4-те вид. Addison-Wesley Professional, 2021. 464 с.
2. Rippon C. Learn React with TypeScript. 2-ге вид. Packt Publishing, 2023. 474 с.
3. Tanure A. S. ASP.NET Core 9 Essentials. Packt Publishing, 2025. 402 с.