

WEB-ЗАСТОСУНОК «ІГРОВА ПЛАТФОРМА»

Вінницький національний технічний університет

Анотація

У цій роботі розглянуто розробку платформи, яка дозволяє завантажувати архіви ігор у вільному доступі для будь якого користувача сайту із використанням таких сучасних технологій як React та CSS для інтерфейсу та серверною частиною на ASP.NET Core із MySQL. Платформа спрямована на підвищення ефективності розповсюдження ігор для звичайних користувачів Інтернет.

Ключові слова: WEB-застосунок, бекенд, фронтенд, SPA, MySQL, React, C#, HTTP-запити, Render, Vercel, Aiven.

Abstract

This paper discusses the development of a platform that allows downloading game archives in free access for any site user using such modern technologies as React and CSS for the interface and the server part on ASP.NET Core with MySQL. The platform is aimed at increasing the efficiency of game distribution for ordinary Internet users.

Keywords: WEB-application, backend, frontend, SPA, MySQL, React, C#, HTTP requests, Render, Vercel, Aiven.

Вступ

Ера цифрових технологій зумовила глобальний перехід індустрії відеоігор до цифрової дистрибуції, яка сьогодні охоплює близько 90% ігрового контенту. Відповідно до аналізу глобального ринку, його обсяг склав понад 185 мільярдів доларів [1]. Незважаючи на домінування таких платформ-гігантів, як Steam чи Epic Games Store, користувачі все частіше стикаються з проблемою перевантаженості інтерфейсів, високими вимогами до обчислювальних ресурсів та необхідністю встановлення громіздкого десктопного софту. Обов'язкова реєстрація, нав'язування внутрішніх соціальних сервісів і систем підписок створюють додаткові бар'єри для звичайних гравців. Водночас індустрія незалежних розробників (інді-ігор) потребує простих та гнучких інструментів для демонстрації та дистрибуції власних проєктів без жорстких корпоративних обмежень. Відповідно, виникає гостра необхідність розробки легкого WEB-застосунку, який мінімізує технічний поріг входу та забезпечує швидкий доступ до ігрового каталогу безпосередньо через браузер.

Метою дослідження є розширення функціональних можливостей доступу до ігрового контенту шляхом розробки Web-застосунку, що забезпечує прямий доступ до каталогів та завантаження файлів.

Результати дослідження

У межах даного дослідження було проаналізовано існуючі платформи цифрової дистрибуції (Steam, Epic Games Store, itch.io) та спроектовано гнучку клієнт-серверну архітектуру системи. Для забезпечення миттєвого відгуку та плавної роботи інтерфейсу клієнтську частину реалізовано як односторінковий застосунок (SPA) на базі бібліотеки React [2]. Завдяки компонентному підходу це дозволило впровадити динамічний пошук (із застосуванням патерну «debouncing» для оптимізації трафіку) та фільтрацію ігор без повного перезавантаження сторінок браузером. Візуальне оформлення побудовано з використанням сіткової моделі CSS Grid, що гарантує адаптивність інтерфейсу та коректне відображення карток на різних пристроях.

Серверну частину побудовано на базі платформи ASP.NET Core Web API із дотриманням принципів RESTful-архітектури та використанням багатозарової моделі (рівні Repository та Service). Взаємодію з базою даних реалізовано за допомогою ORM Entity Framework Core [3]. Головною архітектурною особливістю розробленого рішення є використання гібридної моделі збереження даних. Для забезпечення максимальної швидкодії обробки запитів:

- Структурована інформація про ігри зберігається у реляційній базі даних MySQL.
- Графічні матеріали (банери, іконки, скріншоти) автоматично оптимізуються та доставляються через хмарний сервіс Cloudinary [4].

- Бінарні архіви ігор завантажуються та зберігаються на платформі Google Drive.

Такий підхід дозволяє не перевантажувати головний сервер важкими бінарними об'єктами та суттєво мінімізує витрати на хостинг. Усю розподілену екосистему розгорнуто за допомогою сучасних хмарних сервісів: Vercel для фронтенду, Render для бекенду та Aiven.io для управління базою даних.

Для формування активної спільноти незалежних розробників у систему впроваджено підсистему профілювання та публікації власного контенту. Захист облікових записів забезпечується криптографічним алгоритмом PBKDF2 із застосуванням хеш-функції SHA256 та генерацією випадкової 16-байтної солі (через клас Rfc2898DeriveBytes), що гарантує надійну верифікацію без необхідності зберігання паролів у відкритому вигляді [5]. Обробка багатокомпонентних запитів (multipart/form-data) налаштована з лімітом до 1,5 гігабайта для паралельної передачі великих ігрових архівів у хмарні сховища. Працездатність усіх модулів, алгоритмів пагінації та обробку виняткових ситуацій було успішно підтверджено шляхом комплексного ручного тестування API через Swagger UI.

Висновки

Отже, створений WEB-застосунок «Ігрова платформа» ефективно вирішує проблему перевантаженості десктопних ігрових клієнтів, мінімізує технічний поріг входу та гарантує максимально короткий шлях від пошуку гри до її безпосереднього скачування без обов'язкової авторизації. Завдяки впровадженню гібридної моделі збереження даних (MySQL, Cloudinary, Google Drive) досягнуто високої масштабованості системи та повністю ізольовано роботу з масивним бінарним контентом. Використання сучасного компонентного стеку технологій (React, ASP.NET Core) дозволило створити стабільний, швидкий та економічно ефективний продукт, повністю готовий до практичної експлуатації.

Мета дослідження - розширення функціональних можливостей доступу до ігрового контенту досягається шляхом розробки Web-застосунку, що забезпечує прямий доступ до каталогів та завантаження файлів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Statista. Digital Transformation of the Video Game Industry: Worldwide Statistics 2025 [Електронний ресурс] – Режим доступу: <https://www.statista.com/outlook/dmo/digital-media/video-games/worldwide>.
2. React Documentation. Building Modern Single Page Applications with Components [Електронний ресурс] - Режим доступу: <https://react.dev/learn>
3. Microsoft Learn. ASP.NET Core Web API Documentation and C# Development Guide [Електронний ресурс] – Режим доступу: <https://learn.microsoft.com/en-us/aspnet/core/web-api/>
4. Cloudinary Documentation. Image and Video Management API Reference [Електронний ресурс] – Режим доступу: <https://cloudinary.com/documentation>.
5. Rfc2898DeriveBytes Class [Електронний ресурс] – Режим доступу: <https://learn.microsoft.com/uk-ua/dotnet/api/system.security.cryptography.rfc2898derivebytes>.

Коломійчук Денис Вадимович – студент групи ЗКН-22б, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, Вінниця, e-mail: denyskolomiichuk@gmail.com

Сілагін Олексій Віталійович – кандидат технічних наук, доцент кафедри комп'ютерних наук, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, Вінниця, e-mail: avsilagin@vntu.edu.ua

Kolomiichuk Denys Vadymovych – student of group 3CS-22b, Faculty of Intellectual Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: denyskolomiichuk@gmail.com

Silagin Olexsiy Vitaliyovych – Candidate of Technical Sciences, Associate Professor of the Department of Computer Science, Faculty of Intellectual Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: avsilagin@vntu.edu.ua