

УДК 004.4

Корлиханов О., Мудрик І.

Тернопільський національний технічний університет імені Івана Пулюя

РОЗРОБКА ВЕБ-САЙТУ ДЛЯ ОНЛАЙН ГРИ «МАФІЯ» З ВИКОРИСТАННЯМ ТЕХНОЛОГІЙ NODE.JS ТА ФРЕЙМВОРКУ – REACT

Korlykhanov O., Mudryk I.

Ternopil Ivan Puluj National Technical University

WEBSITE DEVELOPMENT FOR THE ONLINE GAME "MAFIA" USING NODE.JS TECHNOLOGIES AND THE FRAMEWORK – REACT

Наявність онлайн аналогів відомих настільних ігор не є рідкістю в сучасному світі. Проте в Українськомовному середовищі ринок не настільки розвинений. Розробка аналогів з використанням сучасних технологій позитивно вплине на позицію України у сфері розробки веб-застосунків, заклавши недостатні для сучасного українського користувача потреби у веб-продуктах.

Для якісної реалізації проєктів у даній ніші, варто застосовувати технології Node.js та фреймворк – React. Застосунки, розроблені з метою закрити потребу українських користувачів в даній сфері, з використанням більш доцільних та сучасних технологій, з підтримкою українською мови та покращеним візуальним інтерфейсом важливі для розвитку українського ринку.

Проаналізувавши основні типи архітектури веб-сайтів подібного типу, були обрані основні патерни та структури програмування, доцільні для даної сфери. Архітектура, заснована на деревоподібній структурі сайту, є доцільним вибором для специфікації розробки програмного забезпечення такого типу. [1] Основні принципи цієї структури полягають у розгалуженні сторінок веб-сайту аналогічно гілкам від кореня дерева. Потреба у великій кількості унікальних сторінок, підтверджує доцільність цієї структури [2].

Щодо технологій, що базуються на Node.js, варто особливо виділити WebSocket та фреймворк Nest.js. Nest.js є сучасним інструментом для побудови масштабованих серверних застосунків, що базується на архітектурних принципах Angular і використовує TypeScript. Завдяки цьому фреймворку доцільно реалізовувати серверну логіку, дотримуючись чіткої структури модулів, контролерів та сервісів.

Такий підхід забезпечує високу читабельність коду, його легке тестування та повторне використання, що є важливим при розробці складних багатофункціональних систем. Це дозволить ефективно керувати ігровими процесами, зручно обробляти запити користувачів та швидко вносити зміни до логіки застосунку[3].

WebSocket, в свою чергу, є невід'ємною частиною таких проєктів, оскільки забезпечує двосторонній обмін повідомленнями в реальному часі між клієнтом і сервером. Це критично важливо для багатокористувацьких ігор, де затримка в передачі інформації може значно вплинути на ігровий процес та досвід користувача. Реалізація WebSocket дозволяє досягти миттєвої реакції системи на дії гравців, що значно покращує інтерактивність та динаміку гри.

Також, в подібних проєктах необхідно застосовувати бази даних. Діалект SQL – PostgreSQL, є доцільним вибором, оскільки забезпечую швидкий обмін даними з сервером, що є критично важливим для оптимізації та комфортної взаємодії з

користувачем. Реляційна структура бази даних дозволяє ефективно організувати зв'язки між сутностями та забезпечити швидкий доступ до необхідної інформації. А розширені вбудовані типи даних забезпечують гнучкість розробки та організації взаємодії з іншими обраними технологіями[4].

Оскільки, проекти з розробки веб-ігор потребують часте оновлення багатьох компонентів, що може значно навантажувати сервер великою кількістю запитів. Тому для забезпечення оптимізації оновлення інформації на сторінках, використання фреймворку React необхідно.

Данна технологія дозволить створити динамічний, інтуїтивно зрозумілий інтерфейс з компонентною архітектурою, що ефективно вирішує вищеописану проблему. Також спрощує масштабування та усуває проблему повторного використання коду. React забезпечує ефективну роботу з DOM завдяки використанню віртуального DOM та підтримує швидке оновлення інтерфейсу при зміні стану. [5]

Отже, оптимальне технологічне рішення включає використання Node.js із Nest.js для побудови структурованої та масштабованої серверної логіки, WebSocket для забезпечення взаємодії в реальному часі, а також PostgreSQL як продуктивної реляційної бази даних. На клієнтській стороні доцільно використовувати React, що дозволяє створити динамічний інтерфейс, зручний у підтримці та масштабуванні.

Обрана деревоподібна архітектура сайту відповідає специфіці застосунків цього типу, де необхідна велика кількість унікальних сторінок та логічна організація контенту. Комплексне застосування зазначених технологій та архітектурних підходів дозволить створювати сучасні, стабільні та зручні веб-продукти, орієнтовані на українського користувача.

Список використаних джерел:

1. COI.UA. Website Structures: How to Choose the Best Option for Your Web Project URL: <https://coi.ua/blog/Cbc/Website-Structures-How-to-Choose-the-Best-Option-for-Your-Web-Project/>
2. Ivan Osichuk, Vitaly Brevus, Dmytro Bishchak, Yaroslav Mashtaliar, Ivan Mudryk: Leveraging graphics tablet and JPen library to detect essential tremor. CITI 2024: 111-126
3. Turing. What is NestJS & Why Use It? URL: <https://www.turing.com/blog/what-is-nest-js-why-use-it>
4. Data Life UA. Postgres is Eating the Database World. URL: <https://data-life-ua.com/db/postgres-is-eating-the-database-world/>
5. Ranktracker. Why React JS is the Most Favored Front-End Technology for Startups URL: <https://www.ranktracker.com/uk/blog/why-react-js-is-the-most-favored-front-end-technology-for-startups/>