

УДК 004.738.5

Бондар В.С., здобувач

Усата О.Ю. к.пед.н., доцент

Житомирський державний університет імені Івана Франка

ЗАСТОСУВАННЯ ФРЕЙМВОРКУ NEXT.JS ДЛЯ РОЗРОБКИ СИСТЕМИ РЕКОМЕНДАЦІЙ ВІДЕОКОНТЕНТУ

Цифрова трансформація впливає на усі напрями життєдіяльності суспільства, у тому числі й на збільшення обсягу відеоконтенту. Через стрімке інформаційне перевантаження користувачів, ефективна машинна фільтрація контенту стає ключовим фактором утримання аудиторії [1]. Саме тому системи рекомендацій стали важливим ядром будь-якої успішної медіаплатформи. Традиційні алгоритми часто спираються або на історію взаємодій користувача, або на характеристики контенту. Проте, використання лише одного з цих методів призводить до помилок, таких як проблема «холодного старту» для нових користувачів. Вирішенням цієї проблеми є впровадження гібридних моделей персоналізації, які органічно поєднують обидва підходи [2]. Проте інтеграція складних математичних моделей у вебсередовище вимагає використання передових технологій. Для забезпечення швидкої роботи платформи та безперебійного рендерингу рекомендацій вибір архітектури відіграє вирішальну роль.

Зважаючи на безперервну еволюцію цифрових медіа, створення конкурентоспроможного продукту вимагає адаптації передових технологічних рішень, якими вже користуються світові лідери індустрії. Відповідно, головним вектором цієї роботи є імплементація гібридної системи рекомендацій за допомогою фреймворку Next.js. Такий підхід дозволяє перейняти найкращі практики проектування архітектури та наблизити швидкодію вебплатформи до актуальних стандартів.

Архітектура продукту базується на інтеграції великих зовнішніх баз даних медіаконтенту та обробці сесій авторизованих користувачів. Гібридна модель рекомендацій формується за рахунок поєднання структурних метрик відео (жанри, хронометраж, роки випуску) із механізмами колаборативної фільтрації, що спираються на збережену історію взаємодій. Такий підхід дозволяє динамічно генерувати персоналізовані вибірки контенту, уникаючи при цьому необхідності ресурсомісткого аналізу описів чи застосування важких нейромереж [3].

Проте, математичний розрахунок релевантної вибірки – це лише бекенд-складова завдання. Критичним інженерним викликом є оптимізація процесу рендерингу та доставки цих даних на пристрій користувача. Класичні вебзастосунки типу Single Page Application (SPA), побудовані виключно на базовій бібліотеці React.js,

передбачають клієнтський рендеринг. У такій архітектурі браузер спочатку отримує практично порожній HTML-документ і змушений завантажувати об'ємний JavaScript-бандл. Лише після ініціалізації скриптів клієнт робить асинхронні запити до API для отримання масиву рекомендацій. Це створює так званий ефект «водоспаду» мережевих запитів, що призводить до погіршення досвіду взаємодії з платформою.

Для нівелювання описаних проблем у розробці застосовується фреймворк Next.js. Як показує практика використання React у поєднанні з можливостями Next.js дозволяє зменшити час завантаження сторінок, що є критичним для застосунків із великим обсягом медіаданих [4].

Завдяки підтримці рендерингу на стороні сервера, коли користувач відкриває вебплатформу, сервер самостійно звертається до гібридної моделі, отримує персоналізований список рекомендацій та генерує HTML-сторінку. Користувач миттєво бачить контент, що покращує показник Time-to-First-Byte та загальну продуктивність ресурсу.

Крім того, фреймворк Next.js має вбудовану систему маршрутизації. Це дозволяє розмістити логіку обробки даних користувачів та маршрутизацію запитів до алгоритмів персоналізації в межах одного фулстек-застосунку. Відпадає необхідність підтримувати окремий бекенд-сервер для операцій передачі даних, що робить архітектуру платформи більш лаконічною для подальшого масштабування [5].

Розробка конкурентоспроможної відеоплатформи є складним технічним завданням. Застосування гібридної моделі персоналізації підвищує релевантність рекомендацій, а використання Next.js забезпечує ефективну технічну реалізацію. Поєднання серверного рендерингу (SSR), гнучкої архітектури API та високої швидкодії робить Next.js оптимальним інструментом для створення інформаційних систем, що працюють з мультимедійними даними у реальному часі.

Список використаних джерел:

1. Литвин В. В. Технології розроблення інтелектуальних систем : навч. посіб. Львів : Видавництво Львівської політехніки, 2016. 308 с.
2. Burke R. Hybrid Recommender Systems: Survey and Experiments. User Modeling and User-Adapted Interaction. 2002. Vol.12, No4. P.331–370.
3. Adomavicius G., Tuzhilin A. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions IEEE Transactions on Knowledge and Data Engineering. 2005. Vol. 17, No 6. P. 734–749.
4. React – A JavaScript library for building user interfaces. Official React Documentation. URL: <https://react.dev/> (дата звернення: 20.03.2026).
5. API Routes. Next.js. URL: <https://nextjs.org/docs/pages/building-your-application/routing/api-routes> (дата звернення: 20.03.2026)