

тивних функцій, можливі проблеми з сумісністю між різними платформами, складність у підтримці та оновленні великих додатків.

Отже, зараз існує дуже багато технологій для розробки мобільних застосунків. Вибір технології залежить від конкретних потреб проєкту. Нативна розробка для кожної платформи може забезпечити найвищий рівень продуктивності і доступу до всіх можливостей пристрою, але вона також потребує більших зусиль у розробці та підтримці.

Список використаних джерел

1. Meta Platforms. React Native Architecture Overview: вебсайт. URL: <https://reactnative.dev/architecture/overview>
2. LePage Pete, Richard Sam. What are Progressive Web Apps? 2020. URL: <https://web.dev/articles/what-are-pwas>
3. Dembny Michal. What is Flutter and How Can It Benefit Your Business? 2024. URL: <https://www.thedroidsonroids.com/blog/what-is-flutter-app-development>
4. Petkovski Filip. Developing Mobile Applications with Cordova. 2014. URL: <https://www.toptal.com/app/developing-mobile-applications-with-apache-cordova>

УДК 004.738.5

*Лаптєва М. А., здобувачка 2 курсу спеціальності 122 Комп'ютерні науки,
Зелінська О. В., канд. техн. наук,
доцент, доцент кафедри
інформаційних технологій*

СУЧАСНІ ТЕНДЕНЦІЇ BACKEND МОБІЛЬНИХ ДОДАТКІВ

Донецький національний університет імені Василя Стуса, м. Вінниця

З розвитком мобільних технологій та зростанням попиту на мобільні додатки виникла потреба у створенні ефективних та надійних бекенд-рішень. Бекенд мобільних додатків забезпечує обробку даних, управління користувачами, інтеграцію з різними сервісами та гарантує безпеку. У цій статті розглядаються основні сучасні тенденції в цій галузі.

Однією з найпопулярніших тенденцій у розробці бекенду є серверлесс-архітектура. Ця модель дає змогу розробникам зосередитися на написанні коду, не турбуючись про управління серверами. Використання сервісів, як-от AWS Lambda, Google Cloud Functions та Azure Functions, значно спрощує процес масштабування та знижує витрати на інфраструктуру. Це також зменшує час на розгортання нових функцій, оскільки розробники можуть швидко реалізовувати та тестувати нові ідеї [1].

Мікросервісна архітектура стає дедалі популярнішою завдяки своїй гнучкості та масштабованості. Вона передбачає розділення додатка на незалежні модулі, кожен з яких виконує окрему функцію і може бути розгорнутий та масштабований окремо. Це дає змогу командам працювати автономніше та швидше реагува-

ти на зміни у вимогах. До того ж це сприяє покращенню надійності додатка, оскільки збій одного мікросервісу не призводить до відмови всього додатка.

GraphQL, розроблений Facebook, є сучасною альтернативою REST API. Він дає змогу клієнтам запитувати лише ті дані, які їм потрібні, що зменшує обсяг переданої інформації та покращує продуктивність. GraphQL також забезпечує більш гнучкий та ефективний підхід до обробки запитів, що особливо важливо для мобільних додатків з обмеженою пропускнуою здатністю мережі. Цей підхід дає змогу розробникам створювати більш чіткі та структуровані API, що полегшує їх підтримку та розвиток [2].

API Gateway стає невід'ємною частиною сучасної архітектури бекенду. Цей сервіс виконує роль єдиного входу для всіх клієнтських запитів до бекенду, забезпечуючи балансування навантаження, кешування, автентифікацію та моніторинг. Популярні рішення в цьому сегменті включають Amazon API Gateway, Apigee та Kong. Використання API Gateway дає змогу зменшити складність управління API та підвищити безпеку завдяки централізованому контролю доступу та моніторингу запитів.

Використання кешування даних, наприклад, Redis або Memcached, значно покращує продуктивність мобільних додатків, зменшуючи час відповіді сервера та знижуючи навантаження на базу даних. Це особливо важливо для додатків з високою інтенсивністю запитів, як-от соціальні мережі або платформи електронної комерції. Кешування дає змогу зберігати часто запитувані дані в пам'яті, що значно скорочує час доступу до них і покращує загальну швидкість додатка [3].

Безпека залишається критичним аспектом бекенду мобільних додатків. Сучасні підходи включають використання OAuth для автентифікації та авторизації, шифрування даних під час передачі та зберігання, а також регулярні перевірки безпеки коду. Впровадження DevSecOps допомагає інтегрувати безпеку на всіх етапах розробки. Це означає, що безпека стає частиною процесу розробки від початку, що знижує ризики вразливостей і забезпечує захист даних користувачів.

Важливість обробки даних у реальному часі зростає, особливо для додатків, які потребують миттєвого оновлення інформації, як-от месенджери або фінансові сервіси. Використання WebSocket та технологій, як-от Firebase Realtime Database або AWS AppSync, дає змогу забезпечити миттєву синхронізацію даних між клієнтом і сервером. Це допомагає користувачам отримувати оновлення в режимі реального часу, що підвищує зручність використання додатків і їх функціональність [4].

Інтеграція штучного інтелекту (AI) та машинного навчання (ML) стає все більш поширеною у мобільних додатках. Це дає змогу створювати інтелектуальні функції, як-от рекомендаційні системи, розпізнавання зображень або обробка природної мови. Платформи, як-от TensorFlow та PyTorch, спрощують розробку та впровадження таких функцій. Впровадження AI та ML дає змогу створювати більш персоналізовані та інтерактивні додатки, що підвищує залученість користувачів і задовольняє їх потреби більш ефективно.

Сучасні тенденції у розробці бекенду мобільних додатків зосереджені на підвищенні продуктивності, гнучкості та безпеки. Серверлесс-архітектура, мікросервіси, GraphQL, API Gateway, кешування даних, безпека, обробка в реальному

часі й інтеграція AI та ML є ключовими елементами, що формують майбутнє мобільних додатків. Врахування цих тенденцій допоможе створювати більш ефективні, надійні та масштабовані рішення для користувачів.

Список використаних джерел

1. Kreibich J. A. Redis: the Definitive Guide: Data Modeling, Caching, and Messaging. O'Reilly Media, Incorporated, 2015. 340 p.
2. Biswas N. Practical GraphQL. Berkeley, CA: Apress, 2023. DOI: 10.1007/978-1-4842-9621-9 (дата звернення: 13.05.2024).
3. Lewis J., Fowler M. Microservices. 2014. URL: <https://martinfowler.com/articles/microservices.html> (дата звернення: 15.05.2024).
4. What is Amazon API Gateway? URL: <https://docs.aws.amazon.com//developer/guide/welcome.html> (дата звернення: 11.05.2024).

УДК 004.738.5

Суліма В. К., здобувач 2 курсу спеціальності 122 Комп'ютерні науки, Зелінська О. В., канд. техн. наук, доцент, доцент кафедри інформаційних технологій

СУЧАСНІ ТЕНДЕНЦІЇ FRONTEND МОБІЛЬНИХ ДОДАТКІВ

Донецький національний університет імені Василя Стуса, м. Вінниця

З розвитком мобільних технологій та зростанням попиту на мобільні додатки значно змінилися підходи до розробки frontend-частини. Frontend мобільних додатків відповідає за інтерфейс користувача та взаємодію з додатком, забезпечуючи зручність, швидкодію та привабливість. У цій публікації розглядаються основні сучасні тенденції у цій галузі.

Однією з найважливіших тенденцій є використання фреймворків та бібліотек для кросплатформенної розробки, як-от React Native та Flutter. Ці інструменти дають змогу створювати додатки для iOS та Android з використанням єдиного коду, що значно знижує витрати на розробку та підтримку. React Native, розроблений Facebook, і Flutter, створений Google, забезпечують високу продуктивність та гарний вигляд додатків [1].

Прогресивні вебдодатки (PWA) стають дедалі популярнішими завдяки своїй здатності забезпечувати функціональність, подібну до нативних додатків, використовуючи вебтехнології. PWA дають змогу встановлювати додатки безпосередньо з веббраузера, що зменшує залежність від магазинів додатків та спрощує процес оновлення. Вони також можуть працювати офлайн, що підвищує зручність для користувачів у місцях з обмеженим інтернет-з'єднанням.

Адаптивний дизайн та гнучкі макети стають стандартом у розробці мобільних додатків. Використання CSS Grid та Flexbox дає змогу створювати інтерфейси, які автоматично підлаштовуються під різні розміри екранів та орієнтацію пристроїв. Це забезпечує оптимальний користувацький досвід, незалежно від того, який пристрій використовується [2].