

УДК 004.41

Карпець Д.–ст. гр. СП-41

Тернопільський національний технічний університет імені Івана Пулюя

ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ МУЛЬТИСЕРВІСНОЇ АРХІТЕКТУРИ ДЛЯ ІНТЕЛЕКТУАЛЬНОГО НАВЧАЛЬНОГО КОНТЕНТУ

Науковий керівник: д.ф.-м.н., проф. Петрик М. Р.

Karpets D.

Ternopil Ivan Puluj National Technical University

DESIGNING SOFTWARE BASED ON MICROSERVICES ARCHITECTURE FOR INTELLIGENT EDUCATIONAL CONTENT

Supervisor: PhD Petryk M. R.

Ключові слова: мультисервісна архітектура, інтелектуальний контент, освітні системи, мікросервіси, програмне забезпечення.

Keywords: microservices architecture, intelligent content, educational systems, microservices, software design.

Вступ. Сучасні освітні платформи активно впроваджують інтелектуальні технології для адаптації навчального контенту під потреби користувачів. Це зумовлює необхідність створення гнучких, масштабованих та ефективних програмних систем. Мультисервісна архітектура (microservices) є одним із ключових підходів до розробки таких систем, оскільки дозволяє розділити функціональність на незалежні сервіси, що спрощує розробку, тестування та масштабування[1].

Особливістю інтелектуального навчального контенту є необхідність обробки великих обсягів даних, персоналізації матеріалів та інтеграції з різними сервісами, такими як системи рекомендацій, аналітики та оцінювання знань. Це вимагає застосування сучасних архітектурних підходів.

Мета роботи – дослідження принципів проектування програмного забезпечення на основі мультисервісної архітектури для створення інтелектуального навчального контенту.

Основна частина. Мультисервісна архітектура передбачає розбиття системи на набір незалежних сервісів, кожен із яких виконує окрему функцію. Це дозволяє підвищити гнучкість системи та забезпечити її масштабованість.

Ключові аспекти проектування включають:

1. Декомпозиція системи. Поділ функціональності на окремі сервіси, такі як управління користувачами, генерація контенту, аналітика та рекомендаційні системи. Це дозволяє кожному сервісу розвиватися незалежно.

2. Використання API. Взаємодія між сервісами здійснюється через чітко визначені інтерфейси (REST, gRPC), що забезпечує незалежність компонентів та спрощує інтеграцію.

3. Масштабованість. Кожен сервіс може масштабуватися окремо залежно від навантаження, що є важливим для систем із великою кількістю користувачів.

4. Використання контейнеризації. Технології, такі як Docker та Kubernetes, дозволяють ефективно розгорнути та керувати мікросервісами[2].

5. Інтелектуалізація контенту. Застосування алгоритмів машинного навчання для персоналізації навчального матеріалу, аналізу поведінки користувачів та формування рекомендацій.

Особливу увагу приділено забезпеченню узгодженості даних між сервісами та обробці запитів у реальному часі. Використання черг повідомлень (наприклад, Kafka або RabbitMQ) дозволяє реалізувати асинхронну взаємодію між компонентами системи.

Крім того, важливим аспектом є забезпечення надійності системи, що досягається за рахунок відмовостійкості сервісів, балансування навантаження та моніторингу.

Результати. У результаті дослідження визначено основні підходи до проєктування програмного забезпечення на основі мультисервісної архітектури. Їх застосування дозволяє:

- підвищити масштабованість та гнучкість системи;
- забезпечити ефективну обробку навчального контенту;
- реалізувати персоналізацію навчання;
- спростити супровід та розвиток програмного забезпечення;

Висновки. У роботі досліджено особливості проєктування програмного забезпечення на основі мультисервісної архітектури для інтелектуального навчального контенту. Показано, що використання мікросервісів дозволяє створювати масштабовані, гнучкі та ефективні освітні системи, здатні адаптуватися до потреб користувачів.

Список використаних джерел:

- [1] S. Newman, Building Microservices, 2nd ed., O'Reilly Media, 2021.
- [2] B. Burns, B. Grant, D. Oppenheimer, E. Brewer, J. Wilkes, Kubernetes: Up and Running, 3rd ed., O'Reilly Media, 2022.
- [3] M. Richards, N. Ford, Fundamentals of Software Architecture, O'Reilly Media, 2020.