

## АРХІТЕКТУРА КЛІЄНТ-СЕРВЕРНОЇ СИСТЕМИ ДЛЯ ОНЛАЙН-ЗАМОВЛЕННЯ ТА ДОСТАВКИ ПРОДУКТІВ ХАРЧУВАННЯ

Якубовський А.В., Філімончук Т.В.

Харківський національний університет радіоелектроніки, Харків, Україна

Сучасний розвиток цифрових технологій зумовив трансформацію сфери громадського харчування, де онлайн-сервіси доставки їжі стали критично важливим елементом міської інфраструктури [1].

Зростання популярності таких сервісів зумовлене зміною споживчих звичок, темпом життя та прагненням до комфорту [2].

Доцільність проектування кросплатформних вебрішень диктується потребою у забезпеченні широкого доступу до послуг без необхідності встановлення додаткового програмного забезпечення. Впровадження архітектури Single Page Application дозволяє перенести значну частину логіки обробки інтерфейсу на сторону клієнта, що мінімізує кількість повних перезавантажень сторінок та створює ефект безперервної взаємодії, подібної до нативних мобільних застосунків [3].

Водночас, успішна реалізація таких проєктів залежить від архітектурної гнучкості, що дозволяє інтегрувати платіжні системи, модулі відстеження замовлень та персоналізовані кабінети користувачів, але для того щоб зробити таку систему справді працездатною, потрібно забезпечити, щоб вона могла працювати з великою кількістю даних, швидко обробляти транзакції та при цьому працювати стабільно без відмов. Головна складність є у тому, що потрібно правильно поєднати клієнтську та серверну частини, для чого зазвичай використовують сучасні архітектурні рішення та відповідні фреймворки.

**Метою доповіді** є обґрунтування технічних рішень та розробка архітектури клієнт-серверної системи для онлайн-замовлення та доставки продуктів харчування, що базується на використанні Django та React.

Структура розробленого рішення базується на розподілі функціональних обов'язків між незалежними сервісами. Бекенд-складова системи включає спеціалізовані застосунки для ведення бази даних страв, керування профілями користувачів та життєвим циклом замовлень, для збереження та структурування даних обрано реляційну базу даних, яка забезпечує строгу відповідність типів та надійність збереження транзакційної інформації. Використання вбудованої в Django ORM-системи дозволяє оперувати об'єктами бази даних на високому рівні абстракції, що не тільки прискорює процес розробки, а й автоматично нівелює ризики виникнення SQL-ін'єкцій [4].

Використання такої системи гарантує цілісність складних реляційних зв'язків між таблицями інгредієнтів, категорій меню та фінальних замовлень. Фронтенд-частина побудована за ієрархічним принципом, де основні модулі відповідають за динамічне відображення меню, інтерактивний кошик та особистий кабінет клієнта. Такий підхід дозволяє забезпечити високу

модульність коду, спрощує процес тестування окремих частин платформи та створює умови для її подальшого масштабування без зміни фундаментальної архітектури.

У доповіді представлено результати реалізації системи, що використовує REST-орієнтовану архітектуру для взаємодії через API. Наведені дані показують, що на продуктивність платформи впливають ефективність маршрутизації HTTP-запитів, методи управління станом на фронтенді та швидкість обміну даними у форматі JSON.

Використання Single Page Application архітектури та компонентного підходу React забезпечує мінімізацію затримок при навігації та підвищує зручність інтерфейсу користувача.

Ключовою перевагою обраного клієнтського фреймворку є використання технології Virtual DOM, яка дозволяє виконувати точкове оновлення елементів сторінки без перерендерингу всього інтерфейсу, що суттєво знижує навантаження на центральний процесор пристрою користувача.

Особливу увагу приділено оптимізації взаємодії між клієнтом та сервером шляхом впровадження асинхронних запитів, що дозволяє уникнути повного перезавантаження сторінок при маніпуляціях з кошиком або оновленні статусів замовлень.

Для забезпечення безпеки та розмежування прав доступу реалізовано механізм авторизації на основі токенів, що гарантує цілісність даних у межах користувацьких сесій. Використання інструментарію Django Rest Framework дозволило створити гнучку систему серіалізації, яка мінімізує об'єм мережного трафіку та прискорює час відгуку інтерфейсу.

Важливим елементом архітектури є логічне відокремлення модулів управління каталогом товарів та обробки замовлень, що забезпечує високу відмовостійкість системи.

Результати тестування підтвердили, що обраний підхід дозволяє ефективно обробляти черги запитів у години пікового навантаження, забезпечуючи стабільну роботу сервісу та високу швидкість візуалізації даних.

### Список літератури

1. Hong C., Choi H., Choi E.-K., Joung H.-W. Factors affecting customer intention to use online food delivery services before and during the COVID-19 pandemic. *Journal of Hospitality and Tourism Management*. V. 48, 2021, P. 509–518. URL: <https://doi.org/10.1016/j.jhtm.2021.08.012>
2. Gavilan D., Balderas-Cejudo A., Fernandez-Lores S., Martinez-Navarro G. Innovation in online food delivery: Learnings from COVID-19. *International Journal of Gastronomy and Food Science*. V. 24, 2021, 100330. URL: <https://doi.org/10.1016/j.ijgfs.2021.100330>
3. Banks A., Porcello E. *Learning React: Modern Patterns for Developing Real-World Applications*. 2nd edition. O'Reilly Media, 2020. 310 p.
- Windsor W., Kaplan J. *Django for APIs: Build web APIs with Python and Django*. 4th edition. WelcomeToCode, 2022. 208 p.