

## Ефективна розробка веб-застосунку для ресторанного бізнесу

Дмитро Дарнопих, здобувач вищої освіти<sup>1</sup>(ORCID: 0009-0007-7442-290X), Олена Рябчун, здобувачка вищої освіти<sup>1</sup> (ORCID: 0000-0002-1445-966X), Юлія Рябчун, PhD, доцент кафедри ІТ<sup>1</sup> (ORCID: 0000-0002-8320-4038)

<sup>1</sup> Київський національний університет будівництва і архітектури, 03037, м. Київ, проспект Повітряних Сил, 31, Україна

### АНОТАЦІЯ

У даній роботі розглядаються ключові аспекти розробки веб-застосунку для вирішення сучасних проблем ресторанного бізнесу. Проаналізовано як впровадження цифрового рішення сприяє покращенню комунікації з клієнтами та конкуренції на ринку. Розглянуто ефективні підходи до розробки інтеграції онлайн-замовлень, бронюванням столиків, а також інструментів для зворотного зв'язку з клієнтами.

*Ключові слова:* веб-застосунок, ресторанний бізнес, автоматизація.

### 1. ВСТУП

Сучасний ресторанний бізнес стикається з високою конкуренцією та швидко змінюваними потребами клієнтів, що робить використання цифрових рішень невід'ємною частиною успіху. Одним з таких рішень є створення веб-застосунку, який спростить процес замовлення, бронювання столиків, та комунікації з клієнтами. Веб-застосунок дозволить ресторану підвищити рівень обслуговування та забезпечити лояльність клієнтів через персоналізовані пропозиції.

### 2. АВТОМАТИЗАЦІЯ РЕСТОРАННОГО БІЗНЕСУ

Існує багато досліджень [1], які показують, що використання цифрових рішень позитивно впливає на прибутковість ресторанного бізнесу. Так, наприклад, «Deloitte» [2, 3] визначив, що багато клієнтів ресторанів віддають перевагу цифровому досвіду, особливо для доставки та вносу. Інвестування в ці рішення не тільки задовольняє вподобання клієнтів, але й створює нові джерела доходу, наприклад, кухні, які не мають своїх закладів та працюють виключно на винос (рис.). Системи цифрових замовлень допомагають ресторанам зменшити витрати на оплату праці, зберігаючи при цьому задоволеність клієнтів.

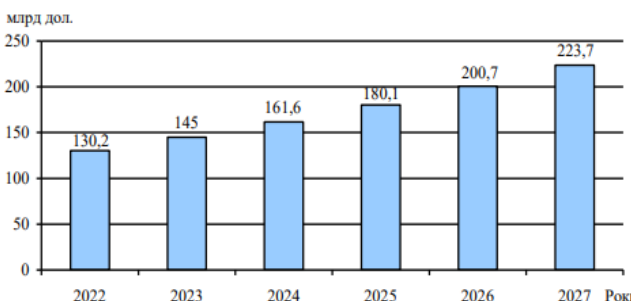


Рисунок. Приблизний обсяг ринку доставки їжі в усьому світі з 2022 до 2027 р.

### 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБ-ЗАСТОСУНКУ

Програмна реалізація веб-застосунку для ресторанного бізнесу включає декілька ключових етапів [4 – 6]: вибір

технологічного стека, розробка серверної та клієнтської частин, інтеграція сторонніх сервісів та забезпечення безпеки. Кожен з цих етапів має критичне значення для створення функціонального, надійного та масштабованого продукту. Розглянемо більш детально кожен з етапів.

#### 3.1. Вибір технологічного стека

Для серверної частини краще використати мову програмування Node.js на базі JavaScript. Вона забезпечує високу продуктивність і підтримує інтеграцію з базами даних, що є важливим для зберігання замовлень, меню, даних клієнтів тощо. Node.js має перевагу в кешуванні - можна зберігати дані або файли локально на комп'ютері користувача, а потім мати швидкий доступ до них, коли наступного разу їх запитуватимуть. Ця функція особливо важлива для веб-сайтів із високим трафіком, оскільки це зменшить навантаження на сервер.

Для баз даних найкраще підійде PostgreSQL, оскільки веб-застосунок потребує чіткої структури даних, складних зв'язків із клієнтами і замовленнями та транзакційної цілісності. PostgreSQL використовує реляційну модель, тобто, дані організовані в таблиці з чітко визначеними полями та типами даних. Також PostgreSQL гарантує цілісність даних завдяки підтримці транзакцій з властивостями ACID (атомарність, консистентність, ізольованість, надійність), що є важливим для фінансових операцій або будь-яких інших критично важливих процесів.

Для клієнтської частини доцільно використовувати React.js або Vue.js, які забезпечують створення динамічного інтерфейсу. React та Vue – це JavaScript-бібліотеки, які призначені для створення інтерфейсів користувача. Вони дозволяють компонувати складні інтерфейси з невеликих окремих частин коду – “компонентів”. Вони обидва використовують віртуальний DOM, що значно підвищує продуктивність. Це означає, що зміни на сторінці (оновлення даних меню або оформлення замовлення) обробляються швидко й ефективно. Також вони добре масштабуються, тому незалежно від розміру проекту (від малого ресторану до мережі), вони можуть забезпечити ефективну роботу інтерфейсу.

#### 3.2. Розробка компонентів серверної частини

Серверна частина відповідальна за обробку замовлень, управління базою даних та інтеграцію з платіжними системами. Основними завданнями серверної частини є: аутентифікація користувачів (адміністратори, персонал, клієнти); обробка замовлень (створення, оновлення,

скасування замовлень, розрахунок вартості); інтеграція з платіжними шлюзами (ApplePay або PayPal) для проведення онлайн-оплат; управління меню та запасами – можливість додавання, видалення або зміни страв, відстеження їх наявності.

### 3.3. Розробка компонентів клієнтської частини

Клієнтська частина – це інтерфейс, з яким взаємодіють користувачі. Основними функціями є: можливість перегляду меню, фільтрація за категоріями та пошук страв; форма для бронювання столиків або оформлення замовлень онлайн (із зазначенням часу, кількості людей або вибору доставки); особистий кабінет користувача, де зберігаються платіжні дані та персоналізовані пропозиції; адаптивність для коректної роботи на мобільних пристроях.

### 3.4. Інтеграція сторонніх сервісів

Успішний веб-застосунок для ресторанного бізнесу потребує інтеграції зі сторонніми сервісами, які допоможуть автоматизувати процеси, такі як платіжні шлюзи для забезпечення безпечних онлайн-транзакцій; системи для відстеження доставки (Google Maps API) для моніторингу процесу доставки замовлення; CRM-системи для збору та аналізу даних про клієнтів, що допоможе у створенні персоналізованих пропозицій.

Додавання функції відстеження замовлення – це інноваційне рішення та інструмент для поліпшення загального досвіду клієнта. Надання клієнтам можливості точно моніторити статус свого замовлення, його місце розташування і передбачуваний час доставки, що створить прозорість і впевненість у процесі обслуговування.

### 3.5. Хостинг

Для стабільної роботи веб-застосунку важливо правильно обрати середовище для його розміщення. Існує кілька варіантів хостингу, які підходять для різних масштабів проєктів, а також забезпечують необхідний рівень безпеки та продуктивності. Використання хмарних рішень, таких як Amazon Web Services (AWS), Microsoft Azure або Google Cloud Platform (GCP), забезпечують гнучкість, масштабованість та надійність. Ці платформи дозволяють автоматично масштабувати ресурси залежно від навантаження на сервери, що особливо важливо для пікових періодів у ресторанному бізнесі, наприклад, під час обідніх та вечірніх годин. Важливо також обрати оптимальне рішення для розміщення баз даних. Хмарні платформи надають послуги керування баз даних, що дозволяє не турбуватися про налаштування, резервне копіювання та масштабування бази. Наприклад, Amazon RDS, Google Cloud SQL або MongoDB Atlas є надійними рішеннями для зберігання інформації про замовлення, клієнтів та меню. Для швидкого завантаження контенту (зображення страв, меню) можна використовувати CDN, такі як Cloudflare або Fastly, що зменшує час завантаження сайту для користувачів у різних регіонах.

### 3.6. Безпека та резервне копіювання

Оскільки веб-застосунок обробляє персональні та платіжні дані клієнтів, важливо приділити увагу шифруванням даних за допомогою SSL-сертифікатів; впровадити безпечну аутентифікацію за допомогою використання JWT-токенів та двофакторної перевірки; та захистити дані від атак типу SQL-ін'єкцій та XSS (перевірка введених даних). При виборі хостингу важливо подбати про

регулярне резервне копіювання даних бази та веб-застосунку. Це гарантує швидке відновлення даних у разі збою. Крім того, потрібно забезпечити належну конфігурацію міжмережвих екранів (firewalls), моніторинг загроз і запровадити регулярні оновлення безпекових політик.

### 3.7. Тестування та підтримка

Фінальний етап – це тестування веб-застосунку на наявність помилок і забезпечення його коректної роботи під навантаженням. Після цього важливо підтримувати веб-застосунок, оновлюючи його відповідно до потреб користувачів та нових технологічних вимог.

Таким чином, ефективна програмна реалізація веб-застосунку для ресторанного бізнесу базується на правильному виборі технологічного стека, належному опрацюванні серверної та клієнтської частин, інтеграції зовнішніх сервісів та забезпеченні високого рівня безпеки.

## 4. ВИСНОВКИ

Підводячи підсумок цього дослідження, можна зробити висновок, що ефективна розробка веб-застосунку для ресторанного бізнесу включає комплексний підхід до вибору технологій, розробки та розміщення системи. Кожен етап – від вибору технологічного стека до тестування та підтримки – має критичне значення для створення надійного та функціонального продукту. Правильна реалізація всіх цих аспектів забезпечить успішну роботу веб-застосунку для ресторанного бізнесу, сприятиме покращенню обслуговування клієнтів і підвищенню ефективності бізнес-процесів.

## Список літератури

- [1] Олійник, О. В., Шестакова, А. В., & Ярмолук, Д. І. (2023). Напрями цифровізації ресторанного бізнесу. *Економіка, управління та адміністрування*, (1(103)), 15–21. DOI: [https://doi.org/10.26642/ema-2023-1\(103\)-15-21](https://doi.org/10.26642/ema-2023-1(103)-15-21)
- [2] Market size of the global online food delivery sector 2022–2027. URL: <https://cutt.ly/r3aQjCi>
- [3] Deloitte “The future of restaurants: The new normal and beyond”. URL: <https://www2.deloitte.com/us/en/pages/consumer-business/articles/future-of-restaurants-study.html>
- [4] Justify Community “Node.js - секрет популярності”. URL: <https://medium.com/@jstify.community/node-js-%D1%81%D0%B5%D0%BA%D1%80%D0%B5%D1%82-%D0%BF%D0%BE%D0%BF%D1%83%D0%BB%D1%8F%D1%80%D0%BD%D0%BE%D1%81%D1%82%D1%96-a3c00759f093>
- [5] Foxminded “Що таке PostgreSQL і для чого використовується?”. URL: <https://foxminded.ua/postgresql-shcho-tse/>
- [6] React “Посібник: знайомство з React”. URL: <https://uk.legacy.reactjs.org/tutorial/tutorial.html>