

**РОЗРОБЛЕННЯ ВЕБ-ЗАСТОСУНКУ
ДЛЯ РОЗМІЩЕННЯ КОМЕРЦІЙНИХ ОГолошень**
Національний аерокосмічний університет ім. М. Є. Жуковського
«Харківський авіаційний інститут»

В умовах швидкого розвитку цифрових технологій і глобальної мережі Інтернет, зростає потреба у зручних та ефективних інструментах для ведення бізнесу та комунікації між продавцями та покупцями. Важливість такого інструменту зросла ще більше у зв'язку з пандемією COVID-19, а згодом з повномасштабною війною, коли багато бізнесів були змушені адаптуватися до нових умов і перейти в онлайн-простір. Серед найвідоміших веб-застосунків для розміщення комерційних оголошень можна відмітити: OLX, PROM, NDO. OLX – пропонує широкий спектр категорій для розміщення оголошень, має простий інтерфейс та активну спільноту користувачів, але часто стикається з проблемами безпеки, такими як шахрайські оголошення та спам і має погано оптимізований функціонал. PROM є значущим гравцем на ринку торгових платформ в Україні. Платформа об'єднує як бізнеси, так і приватних продавців. Але її недоліком є те, що для нових користувачів інтерфейс може здаватися складним та незручним загалом у розміщенні оголошень, а також через велику кількість функцій та налаштувань. На відміну від OLX, PROM має меншу кількість приватних продавців, що може обмежити вибір товарів для покупців. NDO є платформою для розміщення оголошень в Україні, з акцентом на такі категорії як нерухомість, авто та послуги. Сайт відрізняється високим рівнем спеціалізації в цих категоріях, він пропонує розширені функціональні можливості для користувачів. Проте до його недоліків можна віднести дизайн, який виглядає простим і застарілим, та погані рекламні можливості з відсутньою інтеграцією платіжних систем. Таким чином, задача розробки веб-застосунку для розміщення комерційних оголошень є актуальною. Необхідно створити платформу, яка враховує недоліки існуючих рішень і надає користувачам більш зручний, безпечний та ефективний інструмент для ведення бізнесу. Для досягнення поставленої мети були вирішені наступні завдання: проаналізовані існуючі веб-застосунки для розміщення комерційних оголошень, проаналізовано проблеми та вимоги до веб-застосунку, створено адаптивний інтерфейс веб-застосунку, розроблено базу даних (БД) веб-застосунку, реалізовано клієнтську та серверну частини веб-застосунку, проведено тестування основних функцій та режимів роботи веб-застосунку.

Розроблений веб-застосунок побудовано за архітектурою Single Page Application [1], яка включає наступні компоненти системи: 1) клієнтська частина (front-end), відповідає за взаємодію з користувачем, вона надсилає HTTP-запити до серверної частини та отримує HTTP-відповіді; 2) серверна частина обробляє HTTP-запити, отримані від клієнта, виконує бізнес-логіку та взаємодіє з БД; 3) взаємодія з БД здійснюється через ORM Entity Framework, що дозволяє розробникам працювати з даними на рівні об'єктів, а не SQL-запитів. Режими роботи веб-застосунку такі: користувач відвідує сайт, отримує доступ до списку оголошень та детальної інформації щодо них, фільтрує оголошення за категоріями та іншими параметрами. Зареєстровані та авторизовані користувачі можуть виконувати все те, що і гості веб-застосунку, та мають доступ до персонального кабінету, де можна керувати своїми оголошеннями, додавати їх до обраних, створювати свої оголошення, редагувати існуючі, а також залишати відгуки до оголошень і спілкуватися з продавцями у спеціальному чаті, та багато іншого. При створенні або редагуванні оголошення користувачем, воно перевіряється та схвалюється адміністратором.

Вибір TypeScript як основної мови програмування для клієнтської частини обумовлений її перевагами над чистим JavaScript: TypeScript додає статичну типізацію, що допомагає виявляти помилки на етапі компіляції, зменшуючи ризик помилок при виконанні. React обраний для створення динамічних і масштабованих інтерфейсів користувача. Завдяки компонентній архітектурі, React дозволяє створювати повторно використовувані компоненти, що спрощує підтримку та розширення проекту. За допомогою React Context реалізовані «контексти» [2], які дозволяють передавати дані глибоко у дерево компонентів без необхідності передачі: 1) `userContext` – використовується для зберігання даних про авторизованого користувача та надання доступу до певних сторінок веб-застосунку, він забезпечує централізоване управління станом користувача і спрощує процес передачі даних між компонентами; 2) `adminContext` – використовується для управління статусом авторизації адміністратора та надає доступ до панелі адміністратора, що дозволяє легко контролювати доступ до адміністративних функцій застосунку; 3) `categoryContext` – контекст для зберігання категорій оголошень, що допомагає уникнути надмірних запитів до сервера при переході між сторінками або фільтрації оголошень, підвищити продуктивність та зменшити навантаження на сервер, забезпечуючи швидший доступ до даних. Для авторизації користувачів використовується JWT (JSON Web Token) та `localStorage`. Веб-токен JSON – компактний та безпечний спосіб представлення «заявок» (claims) для передачі між двома сторонами. Заявки у JWT являють собою JSON-об'єкт, підписаний за допомогою JSON Web Signature (JWS). Він потрібен, щоб ідентифікувати/здійснити автентифікацію користувача. Для реалізації серверної частини (back-end) було використано мову програмування C# та фреймворк ASP.NET Core [1], який був обраний за його кросплатформенність, високу продуктивність та можливість побудови сучасних веб-застосунків. Структура й функціонування основних рівнів серверної частини веб-застосунку така: 1) `Presentation Layer` відповідає за взаємодію з користувачем і реалізує контролери для обробки HTTP-запитів; 2) `Business Logic Layer` реалізує бізнес-логіку веб-застосунку, цей рівень включає сервіси, що обробляють дані та забезпечують основну функціональність системи; 3) `Data Access Layer` відповідає за доступ до БД та управління даними. Ще однією важливою складовою є бібліотека SignalR [2], яка забезпечує у реальному часі взаємодію між клієнтом та сервером. У кожного користувача своє підключення, він підключається автоматично на front-end частині, по ньому передається інформація оброблена back-end частиною. Для реалізації БД було використано Microsoft SQL Server разом з фреймворком Entity Framework Core, який є об'єктно-реляційним відображенням (Object-Relational Mapping, ORM) для .NET та дозволяє працювати з БД у вигляді об'єктів. Він підтримує підхід Code First, що дозволяє визначати структуру БД безпосередньо у коді C#, що спростило процес розробки і обслуговування БД, дозволяючи зосередитися на бізнес-логіці.

Запропонований веб-застосунок має ряд переваг, які вигідно відрізняють його від існуючих аналогів. По-перше, адаптивний інтерфейс забезпечує зручність використання на різних пристроях, включаючи смартфони і планшети. По-друге, інтеграція з сучасними технологіями, такими як ASP.NET Core і SignalR, забезпечує високу продуктивність та безпеку. По-третє, використання підходу Code First для роботи з базою даних спрощує процес розробки та підтримки системи. Ці переваги роблять новий веб-застосунок ефективним інструментом для розміщення та управління комерційними оголошеннями, сприяючи розвитку електронної комерції та малого бізнесу.

Список використаних джерел

1. Troelsen, A. Pro C# 10 with .NET 6 / A. Troelsen, P. Japikse. – Berkeley CA : Apress, 2022. – 1705 p.
2. Ingebrigtsen, E. SignalR: Real-time Application Development / E. Ingebrigtsen. – Packt Publishing, 2013. – 124 p.