

УДК 004.9

ІНТЕЛЕКТУАЛЬНІ ВЕБ-ДОДАТКИ З ВИКОРИСТАННЯМ JAVASCRIPT ТА API ШТУЧНОГО ІНТЕЛЕКТУ

Черкашина Г.І., Морозова А.І.

Харківський національний університет радіоелектроніки, Харків

Інтеграція технологій штучного інтелекту (ШІ) у веб-розробку стала ключовим напрямом цифрової трансформації. Сучасні компанії активно впроваджують ШІ-моделі для автоматизації завдань, аналітики поведінки користувачів, генерації контенту та персоналізації досвіду користувача [1]. Згідно з даними ЕРАМ, «розумні» веб-системи створюють нові конкурентні переваги, дозволяючи бізнесу знижувати витрати на підтримку й підвищувати точність рішень [2]. Для веб-дизайнера й розробника це означає розширення компетенцій: крім традиційного володіння HTML, CSS та JavaScript, необхідно знати принципи інтеграції API ШІ, роботу з даними, етичні норми та UX-підхід до створення інтелектуальних інтерфейсів.

Інтелектуальні веб-додатки — це програми, що здатні навчатися на даних, прогнозувати поведінку користувача й адаптувати свій інтерфейс або контент завдяки алгоритмам машинного навчання [1]. Вони поєднують у собі веб-технології (JavaScript, HTML, CSS) та зовнішні або локальні API ШІ, які обробляють запити й повертають результати у вигляді рекомендацій, зображень, тексту чи аналітики. За допомогою JavaScript можливо виконувати обробку даних безпосередньо у браузері — наприклад, використовуючи TensorFlow.js або виклики API до моделей OpenAI, Azure Cognitive Services чи Google Cloud AI [1], [2].

Роль JavaScript і API у реалізації інтелектуальних функцій така: JavaScript є центральною мовою для побудови інтелектуальних інтерфейсів. Завдяки асинхронним викликам (fetch, async/await) можна звертатися до ШІ-API, отримувати дані й виводити результати в реальному часі. На серверному боці Node.js забезпечує зв'язок між користувачем і ШІ-моделлю,

використовуючи REST або GraphQL-інтерфейси [2]. Такі можливості дають змогу створювати чат-ботів, рекомендаційні системи, адаптивні веб-портали. Наприклад, JavaScript-додаток може надсилати запит до API системи розпізнавання зображень та повертати опис об'єкта на екрані користувача [1].

Розробка інтелектуального веб-додатку проходить кілька послідовних етапів. Спершу визначається функціональна потреба: які процеси можна покращити за допомогою ШІ — наприклад, автоматизація відповіді на запити чи персоналізація рекомендацій [1]. Далі обирається стек технологій, зокрема JavaScript-бібліотеки (TensorFlow.js, Brain.js) та зовнішні API-провайдери (OpenAI, Microsoft Azure, Amazon Rekognition) [2]. Наступним кроком є підготовка даних: збір, очищення, нормалізація й тестування. Після інтеграції ШІ-модуля у веб-інтерфейс проводиться тестування продуктивності та UX-аналіз. Завершальним етапом є моніторинг і оптимізація, оскільки модель повинна навчатися або оновлюватися відповідно до поведінки користувачів [1].

Використання ШІ підвищує рівень персоналізації, точність аналітики та зручність користувацької взаємодії. Завдяки ШІ можна автоматично сегментувати аудиторію, передбачати наміри користувачів, оптимізувати навігацію чи пропонувати контент на основі історії пошуку [2]. Компанія Stfalcon зазначає, що впровадження інтелектуальних механізмів допомагає підвищити залученість клієнтів і зменшити час розробки за рахунок повторного використання готових API [1].

Найчастіші труднощі пов'язані з якістю даних і відсутністю достатньої експертизи в команді [1]. Недосконалі або упереджені дані можуть призвести до помилкових результатів. Крім того, інтеграція зовнішніх ШІ-API створює залежність від сторонніх провайдерів і потребує уваги до питань безпеки та конфіденційності користувацьких даних. Додатковою проблемою є прозорість алгоритмів: користувач має розуміти, як і для чого його дані аналізуються [4]. Для мінімізації ризиків компанії рекомендують проводити

аудит моделей, тестування етичної коректності та використовувати методи пояснюваного ШІ (Explainable AI) [2].

Інтелектуальні веб-додатки будуються за модульною або мікросервісною архітектурою. ШІ-модуль часто розташовується як окремий сервіс, який обробляє дані незалежно й повертає результат через API-виклик [2], [4]. На фронтенді JavaScript-код відповідає за інтерактивність, а бекенд — за взаємодію з ШІ-сервісами. Використання Node.js дозволяє масштабувати систему та підтримувати одночасно тисячі запитів. За результатами дослідження Stfalcon, такий підхід скорочує час розробки на 30–40 % [1].

Інтелектуальні веб-додатки застосовуються у багатьох галузях. У сфері електронної комерції ШІ аналізує поведінку користувачів і пропонує індивідуальні рекомендації товарів [1]. У сервісах підтримки клієнтів чат-боти на основі JavaScript та NLP-моделей відповідають на запити цілодобово [2]. В освітніх платформах ШІ автоматично оцінює рівень знань і пропонує матеріали різної складності. Окремий напрям — генератори коду на основі ШІ, які допомагають прискорювати процес програмування, роблячи його доступним навіть новачкам [3].

Інтеграція ШІ вимагає забезпечення етичних стандартів, прозорості алгоритмів і дотримання норм захисту персональних даних [1], [4]. Веб-додатки повинні чітко повідомляти користувача про використання його інформації, надавати можливість відмови від обробки та пояснювати логіку роботи системи. Згідно з рекомендаціями дослідження EPAM, розробники мають впроваджувати механізми аудиту моделей і тестування на відсутність дискримінації [2].

Подальший розвиток інтелектуальних веб-додатків відбуватиметься у напрямі автономності та інтеграції edge-обчислень. Моделі ШІ все частіше виконуватимуться безпосередньо у браузері завдяки JavaScript-бібліотекам, що зменшить затримки та підвищить конфіденційність [1], [4]. Окрім того, поширюється використання інструментів автоматизованого створення коду

на базі ШІ, що значно пришвидшує роботу фронтенд-розробників [3]. Зростатиме роль UX-дизайну, оскільки інтерфейси повинні пояснювати результати роботи ШІ-систем користувачам у зрозумілій формі.

Інтелектуальні веб-додатки, створені з використанням JavaScript і API ШІ, поєднують гнучкість веб-технологій із потужністю алгоритмів машинного навчання. Вони здатні радикально змінити спосіб взаємодії між людиною та цифровим середовищем, зробити веб-досвід адаптивним і персоналізованим. Для сучасного веб-розробника володіння технологіями ШІ стає не додатковою перевагою, а професійною необхідністю [1], [2], [3].

Література:

1. Stfalcon E. Як інтегрувати AI в додаток: детальний посібник. *Logistics & Transportation Software Development Agency / Stfalcon*. URL: <https://stfalcon.com/uk/blog/post/how-to-integrate-ai-into-an-app> (дата звернення: 12.11.2025).
2. Г. Прусова та ін., Математика. Київ, Україна: Освіта 2004.
2. EPAM Systems Ukraine. (2025, 24 березня). Створення розумних .NET-додатків: практичний посібник із інтеграції штучного інтелекту. URL: <https://careers.epam.ua/blog/building-smart-net-applications-a-practical-guide-to-ai-integration> (дата звернення: 12.11.2025).
3. Безкоштовні платформи генераторів коду на основі штучного інтелекту | CLAILA. URL: <https://www.claila.com/uk/blog/bezkostovni-platformi-generatoriv-kodu-na-osnovi-stucnogo-intelektu-sprosuut-programuvanna-dla-vsih> (дата звернення: 12.11.2025).
4. Beware the evolving 'intelligent' web service! An integration architecture tactic to guard AI-first components. URL: <https://arxiv.org/abs/2005.13186> (date of access: 12.11.2025).