

**Автор:**

Шевцов Богдан Олександрович,  
студент 41ПЗ групи

**Науковий керівник:**

Малежик Петро Михайлович  
кандидат фізико-математичних наук,  
доктор педагогічних наук, професор

## **РОЗРОБЛЕННЯ ВЕБЗАСТОСУНКУ ДЛЯ АВТОМАТИЧНОГО СТВОРЕННЯ НАВЧАЛЬНИХ ТЕСТІВ НА ОСНОВІ ШТУЧНОГО ІНТЕЛЕКТУ**

**Анотація.** Метою дослідження є створення вебзастосунку для автоматичного створення навчальних тестів на основі штучного інтелекту, що дозволить заощадити час викладачів на підготовку матеріалів та підвищить мотивацію і самодисципліну студентів через зручний інструмент самоперевірки. Завданням дослідження є розробка програмного рішення, що поєднує серверну частину на Node.js та клієнтський веб-інтерфейс на React із залученням Google Generative AI API. Об'єктом дослідження є процес автоматизованого формування навчальних та перевірочних матеріалів за допомогою систем штучного інтелекту. Предметом дослідження є функціональні можливості, програмні засоби та архітектура клієнт-серверного вебзастосунку. Використані методи включають аналіз можливостей великих мовних моделей, проєктування архітектури за допомогою мови моделювання UML, розробку й тестування програмного забезпечення. Результатом дослідження є готовий прототип вебзастосунку «AI Test Generator», що забезпечує миттєву генерацію структурованих тестів за заданою темою чи PDF-документом, їх інтерактивне проходження та автоматичну перевірку.

**Ключові слова:** вебзастосунок, штучний інтелект, генерація тестів, великі мовні моделі (LLM), дистанційне навчання, React, Node.js.

**Вступ.** В умовах постійного росту популярності дистанційного навчання регулярний контроль та самоконтроль знань відіграють надзвичайно важливу роль. Водночас створення якісних тестових завдань вручну є складною і рутинною процедурою, що вимагає від викладачів великих витрат часу на формулювання питань та підбір правдоподібних дистракторів. Сучасні великі мовні моделі стають невід'ємною частиною освітнього процесу, пропонуючи інноваційні підходи до викладання та навчання [1]. Вони дозволяють автоматизувати адміністративні процеси, зокрема створення тестів та підготовку навчальних матеріалів, що значно спрощує робочі процеси та зменшує навантаження на персонал [1].

**Постановка задачі.** Існуючі платформи для тестування (наприклад, Moodle або Google Forms) забезпечують якісний інструментарій для оцінювання та проведення опитувань, але вимагають ручного введення кожного запитання та варіантів відповідей [2, 3]. Тому постає задача створити сервіс, який делегує завдання складання тестів штучному інтелекту. У межах роботи необхідно: проаналізувати можливості сучасних API великих мовних моделей, спроєктувати архітектуру застосунку, розробити серверну частину з REST-інтерфейсом на базі Node.js для взаємодії з ШІ, а також реалізувати інтерактивний веб-інтерфейс на React для відображення і проходження тестів [5,6].

**Мета роботи.** Метою роботи є розробка клієнт-серверного вебзастосунку для автоматичного створення навчальних тестів на основі ШІ, який забезпечує користувачам миттєву генерацію запитань за введеною темою або завантаженим конспектом, а також надає зручний інтерфейс для перевірки результатів.

**Основна частина.** Запропонована система побудована за класичною клієнт-серверною архітектурою, що гарантує безпечне приховування API-ключів від зовнішніх користувачів у змінних оточення. На серверному рівні реалізовано веб-сервер на базі платформи Node.js та фреймворку Express, який обробляє запити, взаємодіє з Google

Generative AI API (модель Gemini-2.5-flash) через спеціалізовані промпти та вилучає згенеровані дані у форматі JSON [4,6]. Для роботи з файлами користувачів сервер використовує бібліотеки `multer` та `pdf-parse`, що дозволяють зчитувати текст із завантажених PDF-документів безпосередньо в оперативній пам'яті. Клієнтський рівень розроблено як Single Page Application (SPA) за допомогою бібліотеки `React`, що забезпечує динамічне оновлення інтерфейсу без перезавантаження вебсторінки [5,7,8]. Клієнтський застосунок керує станами (тема, завантаження, відповіді користувача) та забезпечує миттєвий зворотний зв'язок (коректна чи хибна відповідь) одразу після завершення тестування. Для забезпечення безперебійної роботи впроваджено багаторівневий механізм обробки помилок, який враховує ймовірнісний характер генерації контенту великими мовними моделями та нівелює мережеві збої.

**Висновки.** У результаті виконання роботи створено функціональний вебзастосунок «AI Test Generator», який підтверджує ефективність використання генеративного штучного інтелекту для розробки освітніх матеріалів. Система успішно вирішує проблему витрат часу викладачів на формування питань і дистракторів, автоматизуючи цей процес. Продукт може бути використаний як для підготовки викладачами структурованих тестів, так і студентами для самостійної підготовки. В подальшому систему можна масштабувати шляхом інтеграції бази даних для збереження статистики успішності та додавання підтримки інших форматів навчальних матеріалів.

#### Список використаних джерел

1. Малежик М.П., Малежик П.М., Малежик П.М. Особливості використання великих мовних моделей в освітньому просторі. Розділ в колективній монографії Цифрова трансформація освіти: теоретико-методичні засади: монографія / за заг. ред. В. П. Сергієнка; за наук. ред. Н. П. Франчук – Київ :Вид-во УДУ імені Михайла Драгоманова, 2024. – 382 с.
2. Офіційна документація Moodle: Moodle Docs. URL: <https://docs.moodle.org/> (дата звернення: 21.04.2026).
3. Офіційна довідка Google Forms: Довідковий центр Документів Google. URL: <https://support.google.com/docs/> (дата звернення: 21.04.2026).
4. Офіційна документація Google Gemini API: Google AI for Developers: Gemini API. URL: <https://ai.google.dev/docs> (дата звернення: 21.04.2026).
5. Довідка з React: What Is React? [Easily Explained] | Simplilearn. URL: <https://www.simplilearn.com/tutorials/reactjs-tutorial/what-is-reactjs> (дата звернення: 21.04.2026).
6. Документація Node.js та Express: Express – Node.js web application framework. URL: <https://expressjs.com/> (дата звернення: 21.04.2026).
7. Офіційна документація пакету `multer`: `multer` - npm. URL: <https://www.npmjs.com/package/multer> (дата звернення: 24.04.2026).
8. Офіційна документація пакету `pdf-parse`: `pdf-parse` - npm. URL: <https://www.npmjs.com/package/pdf-parse> (дата звернення: 24.04.2026).