

Міністерство освіти і науки України
Луцький національний технічний університет
Факультет комп'ютерних та інформаційних технологій
Кафедра комп'ютерних наук

КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»

ВЕБРЕСУРС ДЛЯ ВИКОРИСТАННЯ СУЧАСНИХ
ІНТЕЛЕКТУАЛЬНИХ ЦИФРОВИХ ІНСТРУМЕНТІВ
НАВЧАЛЬНОГО ПРИЗНАЧЕННЯ

WEB RESOURCE FOR THE USE OF MODERN INTELLIGENT
DIGITAL TOOLS FOR EDUCATIONAL PURPOSES

спеціальність 122 Комп'ютерні науки

освітня програма «Комп'ютерні науки»

Виконав: здобувач вищої освіти
групи КН-41
Близнюк Юрій Вікторович

(підпис)

Керівник: асистент
Хиць Руслан Андрійович

(підпис)

Кваліфікаційну роботу
допущено до захисту
«__» _____ 2026 р.
Гарант освітньої програми:
д.п.н., професор
Тулашвілі Юрій Йосипович

(підпис)

Луцьк – 2026 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра комп'ютерних наук

Ступінь вищої освіти: бакалавр

Галузь знань: 12 Інформаційні технології

Спеціальність: 122 Комп'ютерні науки

Освітня програма: «Комп'ютерні науки»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Валерій ЛІЩИНА

« ____ » _____ 2026 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА ПЕРШОГО (БАКАЛАВРСЬКОГО) РІВНЯ ВИЩОЇ ОСВІТИ

Близнюк Юрій Вікторович

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи «Вебресурс для використання сучасних інтелектуальних цифрових інструментів навчального призначення»

Керівник асистент Хиць Руслан Андрійович

затверджені наказом закладу вищої освіти від «16» січня 2026 р. № 32/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи «02» червня 2026 р.

3. Вихідні дані до роботи: технології веброзробки, методи зберігання та обробки цифрових даних, тестування працездатності системи

4. Зміст розрахунково-пояснювальної записки (перелік питань, що потрібно розробити):

Вступ

Розділ 1. Аналіз предметної області кваліфікаційної роботи бакалавра

Розділ 2. Специфікація вимог до розроблюваної системи

Розділ 3. Розробка вебресурсу для використання сучасних інтелектуальних цифрових інструментів навчального призначення

Розділ 4. Спеціальні розрахунки

Висновки

5. Перелік графічного матеріалу: 30 рисунків, презентація

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
<i>Аналіз проблеми за темою роботи та постановка завдань дослідження</i>	<i>Хиць Р. А.</i>		
<i>Теоретичне дослідження</i>	<i>Хиць Р. А.</i>		
<i>Практична реалізація</i>	<i>Хиць Р. А.</i>		
<i>Спеціальні розрахунки</i>	<i>Хиць Р. А.</i>		
<i>Показник запозичень тексту</i>	%		
<i>Інструментальна перевірка</i>	<i>Кошелюк В. А.</i>		
<i>Нормоконтроль</i>	<i>Сачук В. О.</i>		
<i>Гарант ОПП</i>	<i>Тулашвілі Ю. Й.</i>		

7. Дата видачі завдання «16» січня 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи магістра	Строк виконання етапів роботи	Примітка
1	<i>Провести огляд літературних джерел по темі кваліфікаційної роботи</i>	<i>до 20.02.2026 р</i>	
2	<i>Провести аналіз загальної проблеми і вибір напрямків дослідження</i>	<i>до 27.02.2026 р.</i>	
3	<i>Розробити функціональну схему роботи програмного продукту</i>	<i>до 21.03.2026 р</i>	
4	<i>Описати засоби розробки об'єкта проектування</i>	<i>до 17.04.2026 р.</i>	
5	<i>Практична реалізація об'єкта проектування</i>	<i>до 25.04.2026 р.</i>	
6	<i>Провести аналіз результатів експерименту</i>	<i>до 04.05.2026 р.</i>	
7	<i>Здача чистового варіанту бакалаврської роботи на кафедрі</i>	<i>до 02.06.2026 р.</i>	

Здобувач вищої освіти _____ **Юрій БЛИЗНЮК**

Керівник роботи _____ **Руслан ХИЦЬ**

АНОТАЦІЯ

Близнюк Ю. В. Розробка навчального веб-ресурсу для використання сучасних інтелектуальних цифрових інструментів навчального призначення. Рукопис. Кваліфікаційна робота бакалавра ОП «Комп'ютерні науки». Луцький національний технічний університет. Луцьк, 2026.

Дана кваліфікаційна робота присвячена проектуванню та розробці навчального веб-ресурсу «ІнтелектЕду», призначеного для систематизованого представлення інформації про сучасні інтелектуальні цифрові інструменти навчального призначення та формування практичних навичок їх використання у студентів закладів вищої освіти.

Проаналізовано стан використання штучного інтелекту в освіті, проведено порівняльний аналіз п'яти аналогічних платформ за десятьма критеріями. Спроектовано архітектуру веб-ресурсу та модель даних на основі Web Storage API. Ресурс реалізовано засобами HTML5, CSS3 та JavaScript без серверних фреймворків. Розроблений ресурс містить каталог AI-інструментів з фільтрацією, п'ять розділів навчальних матеріалів, модуль тестування, блок новин та адміністративну панель керування контентом. Проведено комплексне тестування, ергономічна оцінка, кольорова схема відповідає WCAG 2.1.

Результати мають практичне значення і можуть використовуватись у навчальному процесі кафедри комп'ютерних наук ЛНТУ для студентів спеціальності 122 «Комп'ютерні науки».

Ключові слова: штучний інтелект, навчальний веб-ресурс, інтелектуальні цифрові інструменти, HTML5, CSS3, JavaScript, localStorage, адаптивний дизайн, освітні технології, цифрова трансформація освіти.

ABSTRACT

Blyzniuk Yurii. Development of an Educational Web resource for the use of modern intelligent digital tools for educational purposes. Manuscript. Bachelor's Qualification Thesis, Educational Program «Computer Science». Lutsk National Technical University. Lutsk, 2026.

This bachelor's qualification thesis is devoted to the design and development of the educational web resource «IntellectEdu», intended for the structured presentation of information about modern intelligent digital learning tools and for developing practical skills in their use among higher education students.

The study analyzes the current state of artificial intelligence usage in education and provides a comparative analysis of five similar platforms according to ten criteria. The architecture of the web resource and the data model based on the Web Storage API were designed. The resource was implemented using HTML5, CSS3, and JavaScript without server-side frameworks. The developed resource includes an AI tools catalog with filtering functionality, five sections of educational materials, a testing module, a news section, and an administrative content management panel. Comprehensive testing and ergonomic evaluation were conducted, and the color scheme complies with WCAG 2.1 standards.

The obtained results have practical significance and may be used in the educational process of the Department of Computer Science at Lutsk National Technical University for students majoring in 122 «Computer Science».

Keywords: artificial intelligence, educational web resource, intelligent digital tools, HTML5, CSS3, JavaScript, localStorage, responsive design, educational technologies, digital transformation of education

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ КВАЛІФІКАЦІЙНОЇ РОБОТИ ...	9
1.1 Аналіз сучасного стану проблем	9
1.2 Аналіз аналогічних програм, баз даних, систем, обґрунтування вибраного напрямку та вибір методів рішення.....	11
1.3 Огляд та аналіз аналогічних інформаційних ресурсів	13
1.4 Постановка завдання на кваліфікаційну роботу бакалавра	16
РОЗДІЛ 2 СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ.....	20
2.1 Обґрунтування вибору шляхів, технологій (алгоритмів) і засобів вирішення поставленого завдання.....	20
2.2 Функціонально-структурна схема роботи об'єкта проектування	24
2.3 Створення моделі бази даних.....	30
РОЗДІЛ 3 РОЗРОБКА ВЕБРЕСУРСУ ДЛЯ ВИКОРИСТАННЯ СУЧАСНИХ ІНТЕЛЕКТУАЛЬНИХ ЦИФРОВИХ ІНСТРУМЕНТІВ НАВЧАЛЬНОГО ПРИЗНАЧЕННЯ	36
3.1 Практична реалізація об'єкта проектування. Побудова блок – схем модулів програми.....	36
3.2 Тестування та налагодження системи	48
3.3 Інструкція користувача з інсталяції та використання програмного продукту	56
РОЗДІЛ 4 СПЕЦІАЛЬНІ РОЗРАХУНКИ.....	64
ВИСНОВКИ.....	73
СПИСОК ВИКОРИСАНИХ ДЖЕРЕЛ.....	76

ВСТУП

Стрімкий розвиток штучного інтелекту та цифрових технологій суттєво трансформує освітній процес на всіх рівнях. Сучасні інтелектуальні цифрові інструменти – чат-асистенти, AI-системи пошуку інформації, генератори контенту та інструменти підтримки навчальної діяльності – стають невід’ємною частиною академічного середовища. Разом із тим більшість студентів і викладачів не мають системних знань про можливості цих інструментів, не вміють ефективно їх використовувати та не усвідомлюють меж допустимого застосування в навчальному процесі.

Відсутність структурованих навчальних ресурсів українською мовою, орієнтованих саме на студентів технічних спеціальностей, ускладнює опанування цих технологій. Більшість наявних матеріалів є розрізненими, англomовними або орієнтованими на широку аудиторію без урахування специфіки академічного середовища. Це зумовлює необхідність розробки спеціалізованого навчального веб-ресурсу, який систематизує знання про сучасні AI-інструменти та надає практичні рекомендації щодо їх використання.

Актуальність теми підтверджується активним упровадженням цифрових технологій у систему освіти України відповідно до Концепції розвитку цифрової економіки та суспільства України, а також зростаючим попитом на компетентності у сфері штучного інтелекту серед фахівців з комп’ютерних наук.

Метою кваліфікаційної роботи є проектування та розробка навчального веб-ресурсу для ознайомлення студентів із сучасними інтелектуальними цифровими інструментами навчального призначення, що забезпечує структурований доступ до теоретичних матеріалів, каталогу AI-інструментів та інтерактивної перевірки знань.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- проаналізувати сучасний стан використання інтелектуальних цифрових інструментів в освіті та існуючі аналогічні навчальні ресурси;
- обґрунтувати вибір технологій та засобів розробки клієнтської частини

веб-ресурсу;

- спроектувати архітектуру та функціонально-структурну схему навчального веб-ресурсу;
- розробити модуль каталогу AI-інструментів з можливістю фільтрації за категоріями;
- реалізувати розділ навчальних матеріалів та інтерактивного тестування знань;
- розробити адміністративну панель керування контентом ресурсу;
- провести ергономічну оцінку розробленого інтерфейсу та виконати тестування системи.

Об'єктом дослідження є процес навчання студентів технічних спеціальностей використанню сучасних інтелектуальних цифрових інструментів.

Предметом дослідження є методи та засоби проектування і розробки навчальних веб-ресурсів на основі клієнтських веб-технологій HTML5, CSS3 та JavaScript.

Наукова новизна роботи полягає у розробці структурованого підходу до представлення інформації про AI-інструменти навчального призначення в межах єдиного веб-ресурсу, що поєднує каталог інструментів, теоретичні матеріали, інтерактивне тестування та систему управління контентом без використання серверних технологій.

Практична цінність роботи полягає в тому, що розроблений веб-ресурс може бути впроваджений у навчальний процес кафедри комп'ютерних наук ЛНТУ як допоміжний освітній матеріал для студентів першого та другого курсів, а також використаний як самостійний навчальний інструмент для широкої аудиторії.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ КВАЛІФІКАЦІЙНОЇ РОБОТИ

1.1 Аналіз сучасного стану проблем

Сучасний етап розвитку інформаційного суспільства характеризується стрімким проникненням технологій штучного інтелекту в усі сфери людської діяльності, у тому числі й в освіту. Починаючи з 2022 року, після появи загальнодоступних великих мовних моделей (Large Language Models, LLM), таких як GPT-4, Claude та Gemini, спостерігається якісний стрибок у можливостях AI-інструментів: вони перестали бути вузькоспеціалізованими і стали універсальними помічниками, здатними підтримувати навчальний процес на всіх його етапах [1].

За даними аналітичних досліджень, ринок освітніх технологій (EdTech) на основі штучного інтелекту у 2024 році оцінювався у понад 5 млрд доларів США і, за прогнозами, до 2030 року досягне 30 млрд доларів. Темп зростання складає близько 35 % щороку, свідчить про масштабну трансформацію галузі освіти [2]. В Україні цей процес набуває особливого значення в умовах воєнного стану та переходу значної частини навчання у дистанційний формат.

Сучасні інтелектуальні цифрові інструменти навчального призначення можна класифікувати за функціональним призначенням таким чином:

- AI-асистенти загального призначення (ChatGPT, Claude AI, Google Gemini) – забезпечують діалоговий режим взаємодії, пояснення матеріалу, генерацію та перевірку текстів, розв’язання задач;
- інструменти для наукових досліджень (Perplexity AI, Elicit, Consensus) здійснюють пошук та аналіз наукових джерел з автоматичним цитуванням і синтезом інформації;
- інструменти підтримки програмування (GitHub Copilot, Cursor) надають інтелектуальне автодоповнення коду, виявлення помилок та пояснення алгоритмів безпосередньо у середовищі розробки;
- інструменти продуктивності та організації навчання (Notion AI, Otter.ai)

автоматизують ведення нотаток, транскрипцію лекцій, структурування навчального матеріалу;

– генератори навчального контенту (Gamma AI, Canva AI) забезпечують автоматизоване створення презентацій, інфографіки та навчальних матеріалів.

Незважаючи на широкі можливості зазначених інструментів, їх використання у вищій освіті України стикається з низкою проблем. По-перше, більшість студентів взаємодіють з AI-інструментами стихійно, без розуміння їх архітектурних особливостей, обмежень та оптимальних сценаріїв використання. По-друге, відсутня культура академічно добросовісного застосування AI, що призводить до зростання академічного плагіату. По-третє, викладачі часто не мають достатніх знань для інтеграції цих інструментів у навчальні курси [3].

Ключовою навичкою ефективної взаємодії з AI-інструментами є промптинг (від англ. *prompting*) – мистецтво складання якісних запитів до штучного інтелекту. Дослідження показують, що якість відповіді AI-системи на 60-70 % залежить від якості сформульованого запиту [4]. Відтак, навчання основ промптингу є критично важливим компонентом цифрової грамотності сучасного фахівця з комп'ютерних наук.

В Україні питання формування цифрових компетентностей у сфері штучного інтелекту регулюється Концепцією розвитку цифрової економіки та суспільства України та відповідними рекомендаціями Міністерства освіти і науки. Проте на рівні конкретних закладів вищої освіти навчальні програми з AI-грамотності лише починають формуватися, а спеціалізованих україномовних веб-ресурсів для студентів технічних спеціальностей практично не існує [5].

Таким чином, аналіз сучасного стану проблеми свідчить про наявність суперечності між зростаючою роллю AI-інструментів в освітньому процесі та відсутністю систематизованих навчальних ресурсів для їх освоєння студентами вітчизняних технічних університетів. Це обумовлює актуальність розробки спеціалізованого навчального веб-ресурсу, що є предметом даної кваліфікаційної роботи.

1.2 Аналіз аналогічних програм, баз даних, систем, обґрунтування вибраного напрямку та вибір методів рішення

Перед проектуванням навчального веб-ресурсу необхідно провести аналіз існуючих аналогічних систем і платформ, щоб виявити їхні переваги та недоліки, уникнути повторення наявних рішень і обґрунтувати доцільність власної розробки. Для аналізу було відібрано п'ять систем, які частково або повністю перетинаються з функціональним призначенням розроблюваного ресурсу [6-7].

Аналіз наведених систем (табл. 1.1) дозволяє зробити такі висновки. По-перше, жоден із розглянутих ресурсів не поєднує водночас україномовний контент, каталог AI-інструментів навчального призначення, структуровані теоретичні матеріали та інтерактивне тестування. По-друге, більшість платформ орієнтовані або на широку аудиторію (її називають MOOC), або є суто каталогами без навчальної складової. По-третє, відсутній ресурс, адаптований до контексту вітчизняного закладу вищої освіти та спеціальності 122 «Комп'ютерні науки».

Таблиця 1.1 – Порівняльний аналіз аналогічних систем та ресурсів

Ресурс / Система	Тип	Переваги	Недоліки
Coursera (coursera.org)	МООС-платформа	Структуровані курси від провідних університетів, сертифікація, відеолекції	Переважно англomовна, платні сертифікати, відсутність фокусу на AI-інструментах для навчання
Khan Academy (khanacademy.org)	Навчальна платформа	Безкоштовна, структурований контент, AI-асистент Khanmigo	Орієнтована на шкільну освіту, немає огляду зовнішніх AI-інструментів, не локалізована
Futurepedia (futurepedia.io)	Каталог AI-інструментів	Великий каталог AI-інструментів, фільтрація за категоріями, регулярні оновлення	Англomовна, відсутній навчальний контент та тестування, немає фокусу на освіті
There's An AI For That (theresanaiforthat.com)	Каталог AI-інструментів	Зручний пошук, велика база інструментів, безкоштовний доступ	Лише каталог без навчальних матеріалів, англomовна, не адаптована для освіти
Prometheus (prometheus.org.ua)	МООС-платформа	Україномовна платформа, безкоштовні курси, локальний контекст	Відсутні курси саме з AI-інструментів для навчання, немає каталогу інструментів

Виявлені недоліки аналогів підтверджують доцільність розробки власного веб-ресурсу, який поєднає такі конкурентні переваги:

- повністю україномовний інтерфейс та контент;
- структурований каталог AI-інструментів із фільтрацією за категоріями навчального призначення;
- теоретичні навчальні матеріали з основ промптингу, дослідницької роботи та академічної доброчесності;
- інтерактивний тест для самоперевірки знань із миттєвим зворотним зв'язком;
- адміністративна панель для динамічного оновлення контенту без програмування;
- розгортання без серверної інфраструктури – відкривається звичайним браузером.

Обґрунтування вибраного напрямку вирішення задачі. Враховуючи результати аналізу аналогів та специфіку задачі, обрано напрям клієнтської веб-розробки на основі стандартних технологій HTML5, CSS3 та JavaScript без використання серверних фреймворків. Такий підхід обумовлений такими чинниками [8]:

- тема роботи передбачає розробку навчального ресурсу, а не серверного застосунку, тому складність бекенду є надлишковою;
- клієнтський підхід забезпечує максимальну доступність – ресурс працює без встановлення додаткового програмного забезпечення;
- використання localStorage дозволяє реалізувати динамічне керування контентом без бази даних;
- чисті веб-технології дозволяють продемонструвати фундаментальні знання мов веб-програмування, що відповідає вимогам освітньої програми 122 «Комп'ютерні науки».

Вибір методів вирішення. На основі аналізу предметної області та обґрунтування напрямку розробки визначено методи вирішення поставленого завдання, які наведено у таблиці 1.2.

Таблиця 1.2 – Методи вирішення поставленого завдання

Метод / підхід	Характеристика	Обґрунтування вибору
Клієнтська веб-розробка (HTML5/CSS3/JS)	Розробка без серверної частини з використанням стандартних веб-технологій та localStorage	Відкривається без сервера, максимальна сумісність, легке розгортання, відповідає темі роботи
Модульна архітектура файлів	Розподіл коду на окремі HTML, CSS та JS модулі за функціональним призначенням	Зручність супроводу, читабельність коду, відповідність академічним стандартам
localStorage як сховище даних	Зберігання контенту (інструменти, новини, тест) у браузерному сховищі ключ-значення	Відсутність потреби у БД та сервері, динамічне оновлення через адмін-панель
Адаптивний дизайн (Responsive Design)	CSS Grid, Flexbox та медіа-запити для підтримки різних розмірів екранів	Забезпечує доступність ресурсу з мобільних пристроїв та планшетів

Наведені у таблиці 1.2 методи у сукупності формують цілісний підхід до розробки веб-ресурсу. Клієнтська архітектура з модульною структурою забезпечує зручність розробки та супроводу, localStorage надає динамічність контенту, а адаптивний дизайн гарантує доступність ресурсу для широкої аудиторії користувачів.

1.3 Огляд та аналіз аналогічних інформаційних ресурсів

Coursera – одна з найбільших світових платформ масових відкритих онлайн-курсів (МООС), заснована у 2012 році дослідниками Стенфордського університету. Платформа пропонує тисячі курсів від провідних університетів та компаній, Google, IBM, Meta. У 2023-2025 роках Coursera активно розширила розділ, присвячений штучному інтелекту: з'явилися спеціалізації з генеративного AI, промптингу та застосування LLM-моделей у різних галузях [9].

З точки зору технічної реалізації, Coursera є повноцінним серверним SaaS-застосунком з розвиненою інфраструктурою: відеохостинг, система автоматичної перевірки завдань, форуми, сертифікація та інтеграція з LinkedIn. Інтерфейс платформи є адаптивним і підтримує понад 20 мов, однак українська мова у каталозі курсів представлена вкрай обмежено. Більшість курсів з AI доступні

лише англійською, що суттєво обмежує їх використання вітчизняними студентами без відповідного рівня мовної підготовки.

Ключові недоліки Coursera в контексті даної задачі: платна сертифікація (базовий аудит безкоштовний, але з обмеженим доступом), орієнтація на тривале навчання (курси тривають від 4 до 24 тижнів), відсутність швидкого огляду конкретних інструментів та відсутність можливості адміністрування контенту під конкретний заклад освіти.

Khan Academy – некомерційна освітня платформа, що надає безкоштовний доступ до навчальних матеріалів з математики, природничих наук, програмування та інших дисциплін. У 2023 році платформа представила Khanmigo – власного AI-асистента на базі GPT-4, інтегрованого безпосередньо в навчальний процес. Khanmigo виконує роль репетитора: ставить навідні запитання замість прямих відповідей, що стимулює мислення учнів [10].

Технічна реалізація Khan Academy базується на React-фронтенді та Python/Django бекенді. Платформа має власну систему відстеження прогресу, гейміфікацію та персоналізовані рекомендації. Попри наявність власного AI-асистента, Khan Academy не надає студентам знань про зовнішні AI-інструменти – Khanmigo є вбудованим інструментом, а не об'єктом вивчення. Платформа орієнтована переважно на шкільну освіту (K-12) і не адаптована для студентів вищих технічних навчальних закладів.

Futurepedia – спеціалізований каталог AI-інструментів, що станом на 2026 рік містить понад 5000 позицій. Ресурс дозволяє фільтрувати інструменти за категоріями (текст, зображення, відео, код, продуктивність тощо), моделлю доступу (безкоштовно, freemium, платно) та рейтингом користувачів. Кожна картка інструменту містить короткий опис, знімки екрана та посилання на офіційний сайт.

Futurepedia є суто каталогом без будь-якої навчальної складової: відсутні теоретичні матеріали, пояснення принципів роботи інструментів, рекомендації щодо використання в освіті, тестування знань. Ресурс є повністю англomовним. З технічної точки зору – це серверний застосунок з базою даних, що оновлюється

редакцією. Попри вражаючу повноту каталогу, Futurepedia не вирішує задачу навчання студентів ефективному використанню AI-інструментів у навчальному процесі.

Prometheus – провідна україномовна MOOC-платформа, що функціонує з 2014 року і є партнером Coursera та edX. Платформа пропонує безкоштовні курси українською мовою з різних дисциплін, зокрема з програмування, аналізу даних та цифрових навичок. Технічно Prometheus побудований на базі Open edX – відкритої платформи для дистанційного навчання [11].

В контексті даної задачі Prometheus має суттєвий недолік: станом на момент виконання кваліфікаційної роботи на платформі відсутні курси, присвячені саме AI-інструментам навчального призначення. Наявні матеріали з AI орієнтовані на машинне навчання та аналіз даних як технічні дисципліни, а не на практичне використання AI-асистентів у повсякденній навчальній роботі. Також відсутній каталог зовнішніх AI-інструментів та система новин.

Для узагальнення результатів огляду складено зведену порівняльну таблицю за десятьма ключовими критеріями, що відображають вимоги до розроблюваного ресурсу (табл. 1.3).

Таблиця 1.3 – Зведений порівняльний аналіз аналогічних ресурсів

Критерій оцінювання	Coursera	Khan Academy	Futurepedia	Prometheus	Наша розробка
Україномовний інтерфейс	–	–	–	+	+
Каталог AI-інструментів	–	–	+	–	+
Навчальні матеріали про AI	+	–	–	–	+
Інтерактивне тестування	+	+	–	+	+
Фільтрація за категоріями	+	+	+	–	+
Адмін-панель керування контентом	–	–	–	–	+
Без серверного розгортання	–	–	–	–	+
Безкоштовний повний доступ	–	+	+	+	+
Орієнтованість на ЗВО України	–	–	–	+	+
Новини зі світу AI	–	–	+	–	+

Аналіз таблиці 1.3 підтверджує, що жоден із розглянутих аналогів не відповідає повному переліку вимог до розроблюваного ресурсу. Розроблюваний

ресурс «ІнтелектЕду» задовольняє всі десять критеріїв, тоді як найближчий аналог – Prometheus – відповідає лише чотирьом. Особливо важливими є унікальні характеристики ресурсу, що відсутні в усіх аналогах: адміністративна панель керування контентом та можливість розгортання без серверної інфраструктури.

Результати огляду також дозволяють сформулювати конкретні технічні та змістовні вимоги до розроблюваного ресурсу, які впливають з недоліків аналогів:

- мовна доступність: повністю україномовний інтерфейс та контент без необхідності реєстрації;
- структурованість контенту: чіткий поділ на каталог інструментів, навчальні матеріали та тестування;
- практична орієнтованість: матеріали з конкретними прикладами використання AI у навчанні, а не загальні курси;
- простота розгортання: статичний ресурс без серверної залежності, що спрощує впровадження у навчальному закладі;
- керованість контентом: адміністративна панель для оновлення каталогу, новин та тесту без знань програмування.

1.4 Постановка завдання на кваліфікаційну роботу бакалавра

На підставі результатів аналізу предметної області сформульовано конкретні цілі та завдання кваліфікаційної роботи, які розкривають її суть і повинні бути досягнуті в процесі виконання.

Метою кваліфікаційної роботи є проектування та програмна реалізація навчального веб-ресурсу «ІнтелектЕду» для ознайомлення студентів закладів вищої освіти із сучасними інтелектуальними цифровими інструментами навчального призначення на основі клієнтських веб-технологій HTML5, CSS3 та JavaScript.

Для досягнення поставленої мети необхідно вирішити такі конкретні

завдання:

1) провести аналіз сучасного стану використання AI-інструментів в освіті, дослідити існуючі аналогічні ресурси та обґрунтувати доцільність власної розробки;

2) визначити функціональні та нефункціональні вимоги до навчального веб-ресурсу на основі виявлених недоліків аналогів;

3) спроектувати архітектуру веб-ресурсу: розробити функціонально-структурну схему, визначити склад модулів та схему зберігання даних у localStorage;

4) розробити модуль каталогу AI-інструментів із динамічним відображенням карток та фільтрацією за категоріями навчального призначення;

5) реалізувати розділ структурованих навчальних матеріалів з навігаційним sidebar та інтерактивний модуль тестування з підрахунком результатів;

6) розробити адміністративну панель керування контентом ресурсу з підтримкою CRUD-операцій для всіх модулів;

7) провести ергономічну оцінку розробленого інтерфейсу, тестування системи та скласти інструкцію користувача.

Об'єктом розробки є навчальний веб-ресурс «ІнтелектЕду», що являє собою багатосторінковий статичний сайт із клієнтською логікою на JavaScript та збереженням даних у localStorage браузера.

Вхідні дані системи:

- перелік AI-інструментів навчального призначення з описами, категоріями та посиланнями;

- структуровані навчальні матеріали у форматі HTML-розмітки;

- банк питань для інтерактивного тесту з варіантами відповідей та поясненнями;

- новини та оновлення зі світу AI для відображення на головній сторінці;

- налаштування вигляду головної сторінки (заголовок, підзаголовок, статистика).

Вихідні дані системи:

- структурований каталог AI-інструментів із підтримкою фільтрації;
- відформатовані навчальні матеріали з навігацією по розділах;
- результати проходження тесту з детальним зворотним зв'язком;
- динамічно оновлювана головна сторінка з актуальними новинами.

Функціональні вимоги до розроблюваної системи наведено у таблиці 1.4.

Таблиця 1.4 – Функціональні вимоги до навчального веб-ресурсу

№	Модуль системи	Функціональна вимога	Пріоритет
1	Головна сторінка	Відображення hero-секції, статистики, переліку інструментів та блоку новин	Високий
2	Каталог інструментів	Відображення карток AI-інструментів з фільтрацією за категоріями	Високий
3	Каталог інструментів	Динамічне завантаження даних інструментів з localStorage	Високий
4	Навчальні матеріали	Відображення структурованих матеріалів з навігацією по розділах через sidebar	Високий
5	Тестування	Інтерактивний тест із 10 питань, підрахунком балів та поясненням відповідей	Високий
6	Новини	Відображення актуальних новин зі світу AI на головній сторінці	Середній
7	Адмін-панель	Авторизація адміністратора з перевіркою пароля	Високий
8	Адмін-панель	CRUD-операції для інструментів, питань тесту, новин та налаштувань головної	Високий
9	Адмін-панель	Збереження всіх змін у localStorage без перезавантаження сторінки	Середній
10	Загальні вимоги	Адаптивний дизайн для мобільних пристроїв, планшетів та десктопів	Середній

Нефункціональні вимоги до системи:

- продуктивність: час завантаження сторінок не більше 2 секунд на швидкості інтернету 10 Мбіт/с;
- сумісність: коректна робота у браузерях Google Chrome, Mozilla Firefox, Microsoft Edge та Safari актуальних версій;
- адаптивність: підтримка роздільної здатності від 320px (смартфон) до 2560px (широкоформатний монітор);
- надійність: збереження даних між сесіями за допомогою localStorage з обробкою помилок зчитування;
- безпека: захист адміністративної панелі паролем зі збереженням у

localStorage;

- розгортання: ресурс повинен відкриватися безпосередньо з файлової системи без веб-сервера.

Обмеження системи. Оскільки ресурс реалізується без серверної частини, існують такі обмеження: дані зберігаються локально в браузері конкретного пристрою та не синхронізуються між різними комп'ютерами; пароль адміністратора зберігається у відкритому вигляді в localStorage, що не рекомендується для продакшн-середовища; обсяг даних обмежений розміром localStorage (зазвичай 5-10 МБ залежно від браузера).

Технічні засоби реалізації. Для виконання поставлених завдань визначено такий перелік технологій та інструментів розробки:

- мови розмітки та стилізації: HTML5, CSS3 (CSS Grid, Flexbox, CSS Custom Properties);

- мова програмування: JavaScript (ES6+), Vanilla JS без фреймворків;

- сховище даних: Web Storage API (localStorage);

- шрифти: Google Fonts (Syne, DM Sans);

- середовище розробки: Visual Studio Code з розширеннями Live Server та Prettier;

- система контролю версій: Git / GitHub;

- браузер для тестування: Google Chrome DevTools.

Отримані результати є основою для подальшого проектування.

РОЗДІЛ 2

СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ

2.1 Обґрунтування вибору шляхів, технологій (алгоритмів) і засобів вирішення поставленого завдання

Вибір технологічного стеку є одним із ключових рішень при розробці будь-якої інформаційної системи. Для навчального веб-ресурсу «ІнтелектЕду» це рішення повинно враховувати не лише технічні вимоги, але й специфіку академічного контексту: зручність демонстрації на захисті, відповідність темі роботи, можливість самостійного розгортання без серверної інфраструктури [12].

На ринку веб-розробки існує декілька принципово різних підходів до створення навчальних веб-ресурсів: статичні сайти на чистих веб-технологіях, Single Page Applications (SPA) на базі React або Vue, системи керування контентом (CMS) та серверні застосунки. Кожен підхід має свої переваги та обмеження у контексті даної задачі. Порівняльний аналіз технологічних підходів наведено у таблиці 2.1.

Таблиця 2.1 – Порівняння технологічних підходів до розробки веб-ресурсу

Критерій	HTML/CSS/JS (обрано)	React / Vue SPA	CMS (WordPress)	Node.js SSR
Потреба у сервері	Не потрібен	Не потрібен (статика)	Потрібен (PHP+MySQL)	Потрібен (Node.js)
Складність розгортання	Мінімальна	Потребує збірки (npm build)	Середня	Висока
Відповідність темі роботи	Повна	Часткова	Часткова	Виходить за межі теми
Необхідність додаткових інструментів	Відсутня	npm, bundler	Хостинг, плагіни	npm, БД
Зберігання даних	localStorage	localStorage / API	MySQL БД	MongoDB / PostgreSQL
Демонстрація на захисті	Відкрити файл у браузері	Потребує npm run dev	Потребує сервер	Потребує сервер

На основі аналізу таблиці 2.1 для реалізації навчального веб-ресурсу обрано підхід клієнтської розробки на основі чистих технологій HTML5, CSS3

та JavaScript (Vanilla JS). Ключовими аргументами на користь цього вибору є:

- відсутність потреби у серверній інфраструктурі – ресурс відкривається подвійним кліком на файлі index.html у будь-якому браузері;
- максимальна простота демонстрації під час захисту кваліфікаційної роботи без ризику технічних збоїв;
- повна відповідність навчальним результатам ПР17 освітньої програми – «обирати парадигму веб-програмування, створювати дизайн веб-додатків»;
- можливість продемонструвати фундаментальні знання мов веб-програмування без прихованої складності фреймворків.

Для стилізації веб-ресурсу проведено порівняльний аналіз чотирьох CSS-підходів (табл. 2.2). Обрано нативний підхід на основі CSS Custom Properties (змінні CSS), CSS Grid та Flexbox без підключення зовнішніх фреймворків.

Таблиця 2.2 – Порівняння підходів до стилізації веб-ресурсу

CSS-підхід	Переваги	Недоліки / обмеження
CSS Custom Properties + Grid/Flexbox (обрано)	Нативний браузерний механізм, теми через змінні, повна підтримка сучасних браузерів, не потрібен інструментарій	Більше ручного коду порівняно з утилітарними фреймворками
Tailwind CSS	Швидка розробка, утилітарні класи, гарна документація	Потребує збірки (PostCSS/CDN обмежений), залежність від зовнішнього фреймворку
Bootstrap	Готові компоненти, широка підтримка, CDN-підключення	Стандартизований вигляд, надлишкові стилі, обмежена кастомізація дизайну
SCSS/SASS	Змінні, міксини, вкладеність, потужний препроцесор	Потребує компілятора, не нативна технологія браузера

CSS Custom Properties дозволяють реалізувати централізовану систему дизайн-токенів: усі кольори, відступи та типографічні параметри оголошуються в кореновому селекторі :root і використовуються в усіх файлах стилів. Це забезпечує консистентність дизайну та спрощує внесення глобальних змін [13].

Для динамічного керування контентом без серверної частини обрано Web Storage API, а саме localStorage – механізм браузерного зберігання даних у форматі ключ-значення. Вибір localStorage обґрунтований такими перевагами перед альтернативними рішеннями [14]:

- на відміну від `sessionStorage`: дані зберігаються між сесіями та не видаляються після закриття браузера;
- на відміну від `cookies`: більший обсяг (до 10 МБ), не передаються на сервер, зручний JavaScript API;
- на відміну від `IndexedDB`: простіший API для роботи з невеликими обсягами структурованих JSON-даних;
- на відміну від зовнішнього API: не потребує інтернет-з'єднання та серверної авторизації.

Структура зберігання даних передбачає серіалізацію JavaScript-об'єктів у JSON-рядки за допомогою `JSON.stringify()` при записі та десеріалізацію через `JSON.parse()` при читанні. Для кожного модуля ресурсу визначено окремий ключ зберігання, що унеможливорює конфлікти даних між модулями.

Для типографіки веб-ресурсу обрано дві гарнітури з Google Fonts: Syne (заголовки) та DM Sans (основний текст). Вибір обумовлений такими міркуваннями: Syne є геометричним гротеском із виразним характером, що надає ресурсу сучасного технологічного вигляду; DM Sans є нейтральним гуманістичним шрифтом із відмінною читабельністю на екранах різних розмірів. Обидві гарнітури розповсюджуються за відкритою ліцензією SIL Open Font License та підключаються через CDN Google Fonts [15].

Архітектура файлової структури проєкту побудована за принципом розподілу відповідальності (Separation of Concerns): HTML-файли відповідають за розмітку та структуру контенту, CSS-файли – за візуальне оформлення, JavaScript-файли – за логіку та динамічну поведінку. Така організація відповідає загальноприйнятим практикам веб-розробки та спрощує супровід коду [16]. Файлова структура проєкту «ІнтелектЕду» наведена на рисунку 2.1.

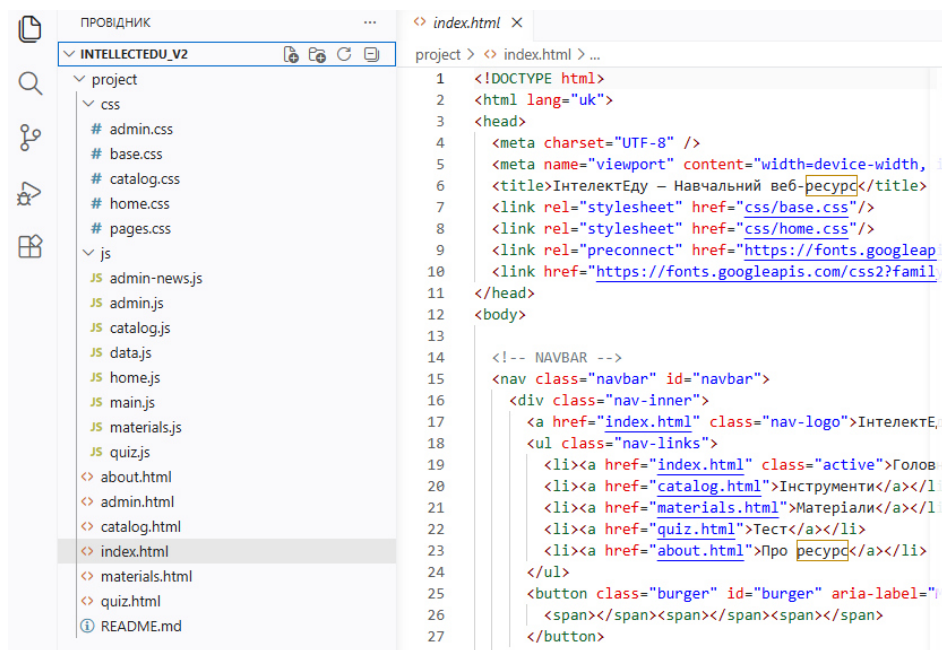


Рисунок 2.1 – Файлова структура проєкту «ІнтелектЕду»

Проект складається з п'яти HTML-сторінок (index.html, catalog.html, materials.html, quiz.html, about.html, admin.html), чотирьох CSS-файлів (base.css, home.css, catalog.css, pages.css, admin.css) та шести JavaScript-модулів (data.js, main.js, home.js, catalog.js, materials.js, quiz.js, admin.js, admin-news.js). Така декомпозиція забезпечує незалежність модулів та можливість їх окремого тестування.

Для розробки веб-ресурсу використано середовище Visual Studio Code – безкоштовний редактор коду від Microsoft з широкою екосистемою розширень. Ключові розширення, що використовувались у процесі розробки:

- Live Server – локальний веб-сервер з автоматичним оновленням браузера при збереженні файлів;
- Prettier – автоматичне форматування коду відповідно до налаштованих стандартів;
- ESLint – статичний аналіз JavaScript-коду для виявлення потенційних помилок.

Версіювання коду здійснювалося за допомогою системи контролю версій Git з використанням сервісу GitHub для зберігання репозиторію. Тестування сумісності проводилося у браузерах Google Chrome (основний), Mozilla Firefox

та Microsoft Edge за допомогою вбудованих інструментів розробника (DevTools).

Отже, було проведено обґрунтування вибору технологічного стеку для розробки навчального веб-ресурсу. Для реалізації клієнтської частини обрано HTML5/CSS3/JavaScript без серверних залежностей. Засобами стилізації визначено нативні CSS Custom Properties, Grid та Flexbox. Для зберігання даних обрано localStorage Web Storage API. Визначено файлову структуру проєкту за принципом розподілу відповідальності та описано засоби розробки.

2.2 Функціонально-структурна схема роботи об'єкта проектування

Функціонально-структурна схема є одним із ключових проектних документів, що визначає склад модулів системи, їхнє призначення та взаємозв'язки між ними. Для навчального веб-ресурсу «ІнтелектЕду» схема охоплює шість публічних сторінок, адміністративну панель та два спільних JavaScript-модулі [17].

Структурна схема веб-ресурсу відображає ієрархію сторінок та зв'язки між ними через навігаційне меню. На найвищому рівні знаходиться головна сторінка (index.html), з якої користувач може перейти до будь-якого розділу ресурсу. Адміністративна панель (admin.html) є окремим захищеним модулем, доступним через посилання у футері. Загальна структурна схема веб-ресурсу «ІнтелектЕду» наведена на рисунку 2.2.

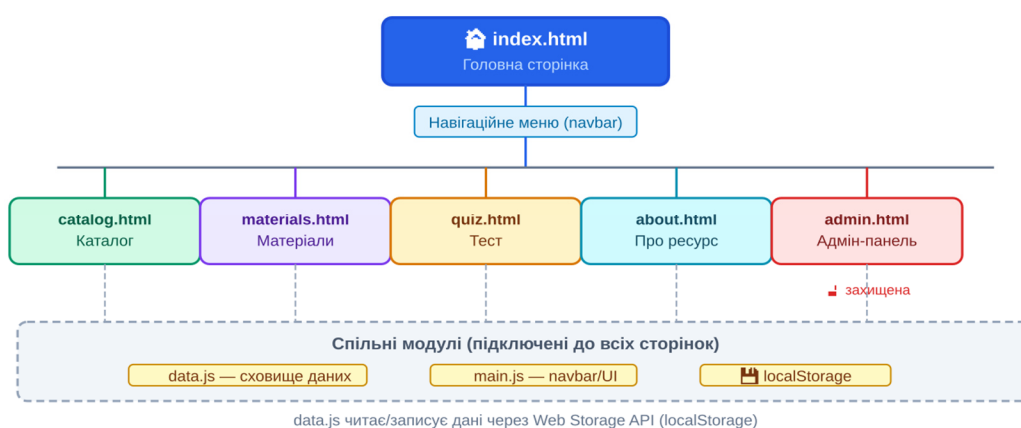


Рисунок 2.2 – Структурна схема веб-ресурсу «ІнтелектЕду»

Як видно з рисунку 2.2, усі публічні сторінки пов'язані між собою через єдине навігаційне меню (navbar), яке реалізовано у кожному HTML-файлі. Спільні модулі data.js та main.js підключаються до кожної сторінки та забезпечують доступ до даних і базову інтерактивність відповідно.

Детальний опис функціонального призначення кожного модуля веб-ресурсу наведено у таблиці 2.3.

Таблиця 2.3 – Функціональне призначення модулів веб-ресурсу

№	Модуль / сторінка	Функціональне призначення	JS-файл відповідальності
1	index.html – Головна	Відображення hero-секції, статистики, переліку інструментів та блоку новин; динамічне наповнення з localStorage	home.js
2	catalog.html – Каталог	Відображення карток AI-інструментів; фільтрація за категоріями; анімація появи карток	catalog.js
3	materials.html – Матеріали	Відображення навчальних розділів; відстеження активного розділу в sidebar при прокрутці	materials.js
4	quiz.html – Тест	Відображення питань по черзі; реєстрація відповіді; підрахунок балів; відображення результатів	quiz.js
5	about.html – Про ресурс	Відображення статичної інформації про ресурс: мета, аудиторія, структура, технічний стек	main.js
6	admin.html – Адмін-панель	Авторизація; CRUD-операції для інструментів, питань тесту, новин; налаштування головної	admin.js, admin-news.js
7	data.js – Модуль даних	Централізоване зберігання та читання даних з localStorage; дефолтні значення; CRUD-функції для всіх сутностей	data.js (спільний)
8	main.js – Загальний модуль	Navbar scroll-ефект; burger-меню; Intersection Observer для fade-in анімацій при прокрутці	main.js (спільний)

З таблиці 2.3 видно, що система має чітку модульну структуру: кожна HTML-сторінка має власний JavaScript-файл відповідальності, а спільна функціональність (дані та навігація) винесена в окремі модулі data.js та main.js, що підключаються до всіх сторінок. Функціональна схема відображає потоки даних між компонентами системи. Центральним елементом архітектури є модуль data.js, який виступає єдиною точкою доступу до localStorage. Всі інші модулі звертаються до даних виключно через функції, визначені в data.js, що реалізує патерн «єдине джерело правди» (Single Source of Truth). Функціональна схема роботи веб-ресурсу наведена на рисунку 2.3.

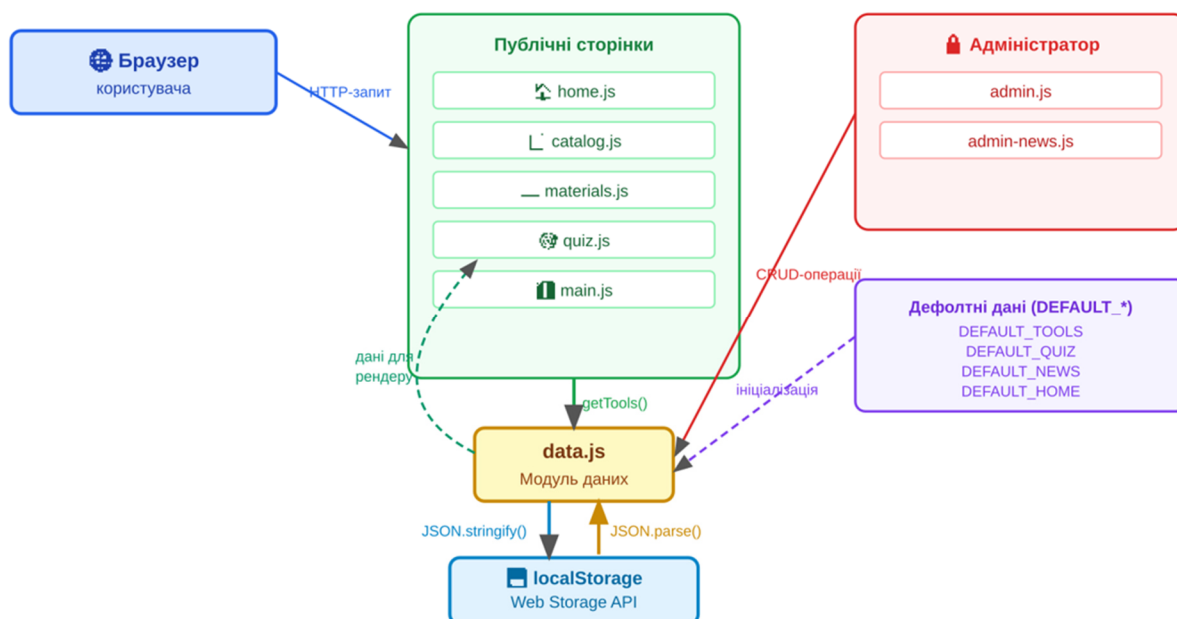


Рисунок 2.3 – Функціональна схема роботи веб-ресурсу «ІнтелектЕду»

Як показано на рисунку 2.3, інформаційні потоки системи організовані таким чином:

- при першому завантаженні сторінки `data.js` перевіряє наявність даних у `localStorage`; якщо даних немає – ініціалізує сховище значеннями за замовчуванням (`DEFAULT_TOOLS`, `DEFAULT_QUIZ`, `DEFAULT_NEWS`, `DEFAULT_HOME`);
- публічні модулі (`home.js`, `catalog.js`, `quiz.js`, `materials.js`) зчитують дані через функції-гетери `data.js` та формують DOM-елементи для відображення користувачу;
- адміністративні модулі (`admin.js`, `admin-news.js`) зчитують та записують дані через функції `data.js`, що автоматично оновлює `localStorage`;
- зміни, внесені адміністратором, відображаються на публічних сторінках після їх перезавантаження, оскільки дані зберігаються перманентно в `localStorage`.

Система передбачає два типи акторів: звичайний користувач (студент) та адміністратор. Студент має доступ виключно до публічних сторінок ресурсу, адміністратор – додатково до захищеної панелі керування контентом. Діаграма варіантів використання наведена на рисунку 2.4.

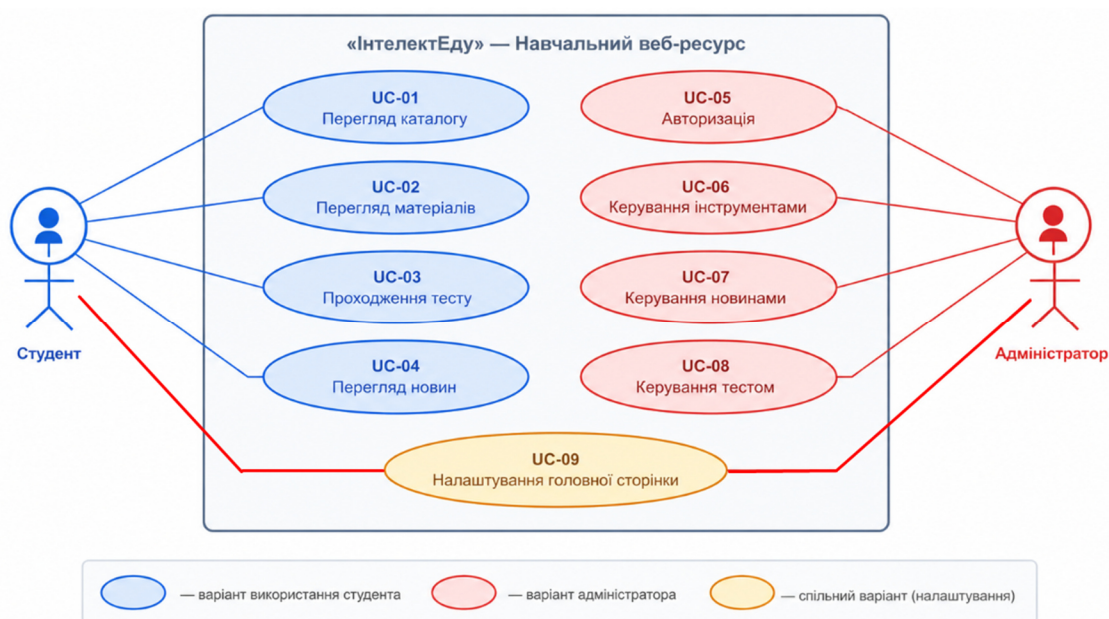


Рисунок 2.4 – UML-діаграма варіантів використання веб-ресурсу

Сценарії використання системи для обох акторів наведено у таблиці 2.4.

Таблиця 2.4 – Сценарії використання веб-ресурсу (Use Cases)

№	Актор	Сценарій використання	Результат
UC-01	Студент	Перегляд каталогу AI-інструментів з фільтрацією за категорією	Відображено відфільтровані картки інструментів
UC-02	Студент	Перегляд навчальних матеріалів, навігація по розділах через sidebar	Відображено обраний розділ, sidebar підсвічує активний пункт
UC-03	Студент	Проходження інтерактивного тесту: вибір відповіді, перехід до наступного питання	Відображено результат тесту з кількістю правильних відповідей та поясненнями
UC-04	Студент	Перегляд актуальних новин зі світу AI на головній сторінці	Відображено 3 останні опубліковані новини
UC-05	Адміністратор	Авторизація в адміністративній панелі за паролем	Надано доступ до панелі керування контентом
UC-06	Адміністратор	Додавання нового AI-інструменту через форму в адмін-панелі	Новий інструмент збережено в localStorage та відображено в каталозі
UC-07	Адміністратор	Редагування або видалення існуючої новини	Зміни збережено в localStorage, оновлено відображення на головній
UC-08	Адміністратор	Додавання нового питання до тесту або редагування існуючого	Оновлений банк питань збережено в localStorage

Модуль тестування є найбільш алгоритмічно складним компонентом системи. Його робота базується на скінченному автоматі зі станами: початок

тесту, відображення питання, очікування відповіді, перевірка відповіді, перехід до наступного питання або відображення результатів. Блок-схема алгоритму модуля тестування наведена на рисунку 2.5.

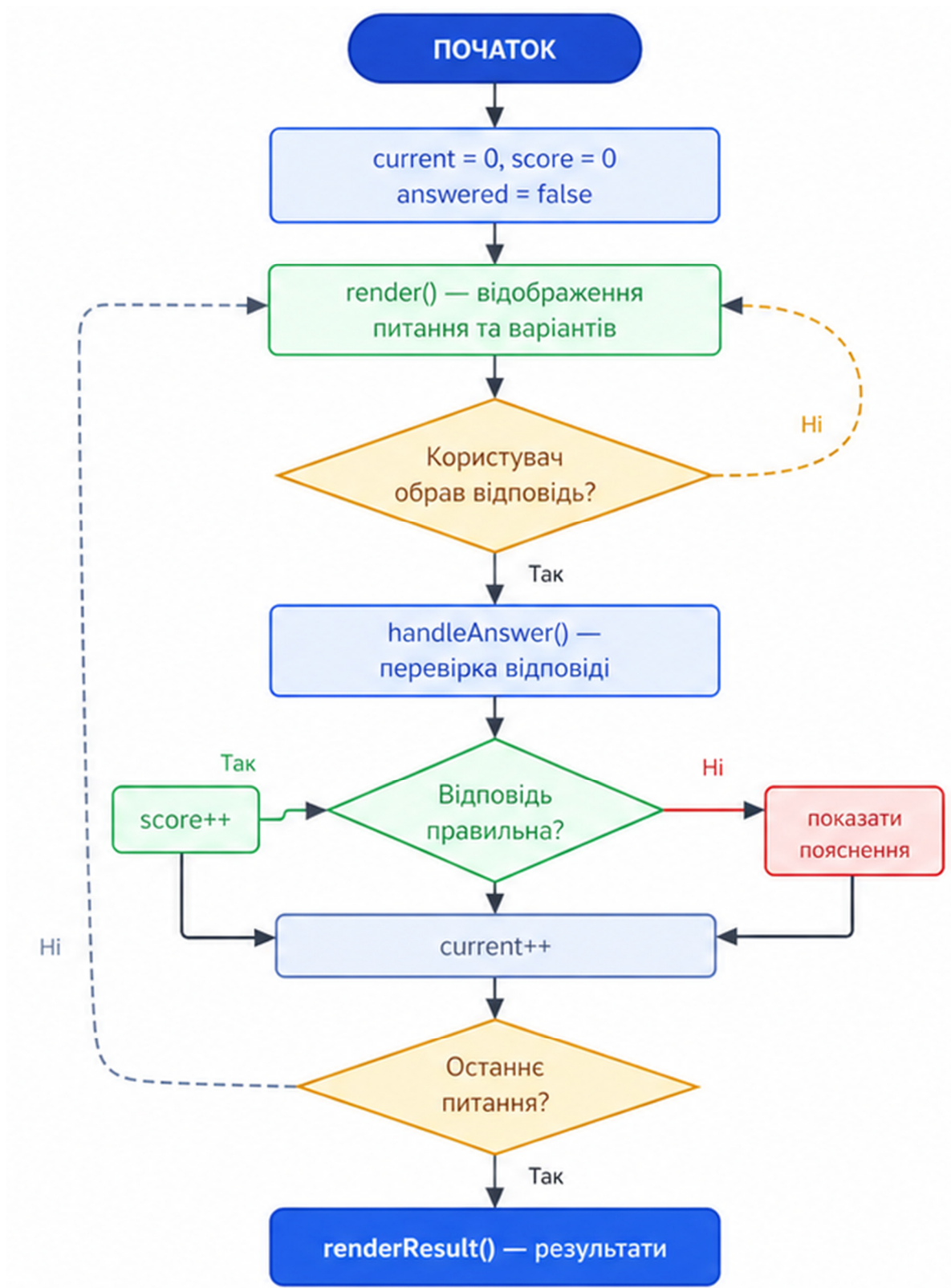


Рисунок 2.5 – Блок-схема алгоритму модуля тестування

Алгоритм модуля тестування (рис. 2.5) функціонує таким чином: після ініціалізації лічильники поточного питання та балів скидаються до нуля. Функція `render()` відображає поточне питання з варіантами відповідей та оновлює індикатор прогресу. При виборі варіанту відповіді функція `handleAnswer()` блокує кнопки, порівнює вибір з правильною відповіддю, оновлює лічильник балів та відображає пояснення. Після відповіді на останнє питання викликається функція `renderResult()`, яка відображає підсумковий результат у відсотках.

Алгоритм фільтрації каталогу AI-інструментів за категоріями реалізований у модулі `catalog.js`. При натисканні кнопки фільтру система порівнює атрибут `data-cat` кожної картки з обраним фільтром та приховує або показує картки відповідно до результату порівняння. Блок-схема алгоритму фільтрації наведена на рисунку 2.6.



Рисунок 2.6 – Блок-схема алгоритму фільтрації каталогу інструментів

2.3 Створення моделі бази даних

Оскільки розроблюваний веб-ресурс реалізований без серверної частини, роль бази даних виконує браузерне сховище Web Storage API – localStorage. Незважаючи на відсутність класичної реляційної СКБД, система потребує чітко спроектованої моделі даних: визначення сутностей, їхніх атрибутів, типів даних та правил зберігання [18].

Для клієнтських веб-застосунків без серверної частини існує кілька механізмів зберігання даних у браузері. Порівняльний аналіз наведено нижче:

- localStorage – зберігає рядкові дані у форматі ключ-значення, обсяг до 10 МБ, дані постійні між сесіями. Обрано для даного проєкту.
- sessionStorage – аналогічний localStorage, але дані видаляються після закриття вкладки. Не підходить для адміністративного контенту.
- IndexedDB – повноцінна клієнтська NoSQL СКБД, підтримує складні запити та транзакції. Надлишково складна для обсягів даного проєкту.
- Cookies – обмежений обсяг (4 КБ), передаються на сервер з кожним запитом. Не підходять для зберігання структурованого контенту.

localStorage обрано як оптимальне рішення: достатній обсяг (до 10 МБ), простий синхронний API, постійне зберігання між сесіями та широка підтримка усіма сучасними браузерами [19].

Інфологічна модель описує предметну область на концептуальному рівні незалежно від конкретної технології зберігання. Для веб-ресурсу «ІнтелектЕду» виділено п'ять основних сутностей та описано зв'язки між ними.

Основні сутності системи:

- Tool (Інструмент) – AI-інструмент навчального призначення. Містить назву, категорію, опис та посилання. Один інструмент належить рівно до однієї категорії;
- Question (Питання тесту) – питання інтерактивного тесту з варіантами відповідей. Кожне питання має рівно одну правильну відповідь та пояснення;
- NewsItem (Новина) – новина зі світу AI для відображення на головній

сторінці. Має статус публікації (опублікована / чернетка);

– HomeConfig (Конфігурація головної) – налаштування вмісту головної сторінки. Існує в єдиному екземплярі (singleton);

– Admin (Адміністратор) – системна сутність, представлена лише паролем доступу до панелі керування (рис. 2.7).

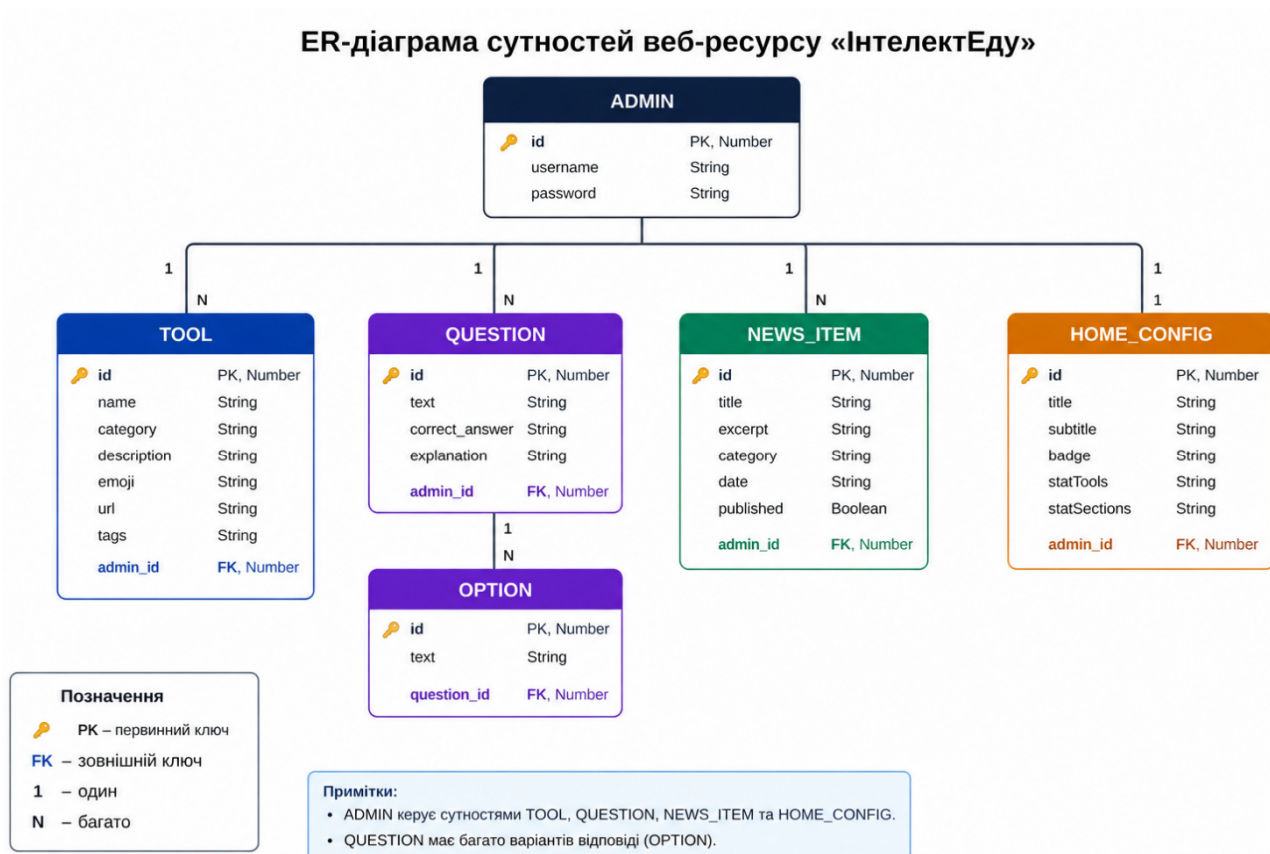


Рисунок 2.7 – Інфологічна модель даних (ER-діаграма) веб-ресурсу

З рисунку 2.7 видно, що сутності системи є відносно незалежними одна від одної – відсутні складні реляційні зв’язки типу «багато-до-багатьох». Сутності Tool, Question та NewsItem утворюють незалежні колекції, а HomeConfig існує як єдиний об’єкт конфігурації. Ця структура є типовою для документоорієнтованих NoSQL-сховищ, до яких можна віднести localStorage [20].

Фізична модель визначає конкретну схему зберігання даних у localStorage. Кожна сутність зберігається під окремим рядковим ключем. Усі ключі мають єдиний префікс intellectedu_ для уникнення конфліктів з іншими застосунками,

що можуть використовувати той самий браузер [21]. Схема фізичної моделі даних наведена у таблиці 2.5.

Таблиця 2.5 – Фізична схема ключів localStorage

Ключ (Key)	Сутність	Тип значення	Призначення
intellectedu_tools	Інструменти	JSON Array of Tool	Масив карток AI-інструментів каталогу
intellectedu_quiz	Питання тесту	JSON Array of Question	Масив питань інтерактивного тесту
intellectedu_news	Новини	JSON Array of NewsItem	Масив новин для головної сторінки
intellectedu_home	Налаштування головної	JSON Object (HomeConfig)	Конфігурація вмісту головної сторінки
intellectedu_admin_pass	Пароль адміністратора	String	Пароль для входу в адмін-панель

Дані кожного ключа зберігаються як серіалізований JSON-рядок. Запис здійснюється через `localStorage.setItem(key, JSON.stringify(value))`, читання – через `JSON.parse(localStorage.getItem(key))`.

Для уникнення помилок при першому запуску модуль `data.js` ініціалізує відсутні ключі значеннями за замовчуванням (`DEFAULT_*`).

Логічна модель визначає структуру кожного об'єкта на рівні JavaScript. Нижче наведено опис полів для кожної сутності. Структура об'єкта AI-інструменту наведена у таблиці 2.6/

Таблиця 2.6 – Логічна структура об'єкта Tool (AI-інструмент)

Поле (field)	Тип даних	Обов'язкове	Опис
id	Number	Так	Унікальний ідентифікатор (timestamp)
name	String	Так	Назва AI-інструменту
category	String	Так	Категорія: assistant / research / code / writing / productivity
emoji	String	Ні	Емодзі-іконка інструменту
description	String	Так	Короткий опис призначення інструменту
tags	Array	Ні	Масив рядків – теги (безкоштовний, платний тощо)
url	String	Ні	URL офіційного сайту інструменту

Структура об'єкта питання тесту наведена у таблиці 2.7

Таблиця 2.7 – Логічна структура об'єкта Question (питання тесту)

Поле (field)	Тип даних	Обов'язкове	Опис
q	String	Так	Текст питання
options	Array<String>	Так	Масив з 4 варіантів відповіді
correct	Number	Так	Індекс правильної відповіді (0–3)
explanation	String	Так	Пояснення правильної відповіді

Структура об'єкта новини наведена у таблиці 2.8.

Таблиця 2.8 – Логічна структура об'єкта NewsItem (новина)

Поле (field)	Тип даних	Обов'язкове	Опис
id	Number	Так	Унікальний ідентифікатор (timestamp)
title	String	Так	Заголовок новини
date	String	Так	Дата публікації у форматі YYYY-MM-DD
category	String	Так	Категорія: Новини AI / Навчання / Поради
excerpt	String	Так	Короткий анонс (2–3 речення)
published	Boolean	Так	true – опублікована, false – чернетка

Структура об'єкта конфігурації головної сторінки наведена у таблиці 2.9.

Таблиця 2.9 – Логічна структура HomeConfig (конфігурація головної)

Поле (field)	Тип даних	Обов'язкове	Опис
title	String	Так	Головний заголовок hero-секції
subtitle	String	Так	Підзаголовок hero-секції
badge	String	Так	Текст бейджа (напр. «Навчальний ресурс 2026»)
statTools	String	Так	Значення лічильника інструментів у stats-bar
statSections	String	Так	Значення лічильника розділів
statTasks	String	Так	Значення лічильника практичних завдань

Взаємозв'язки між логічними структурами даних та ключами localStorage відображено на діаграмі логічної моделі даних (рис. 2.8). Діаграма ілюструє, як кожна JavaScript-структура відповідає певному ключу сховища та який модуль data.js відповідає за роботу з нею.

Технологія доступу до даних реалізована через централізований модуль data.js, який виступає єдиним посередником між компонентами системи та

localStorage. Такий підхід реалізує патерн Repository, що відокремлює логіку зберігання від логіки відображення.

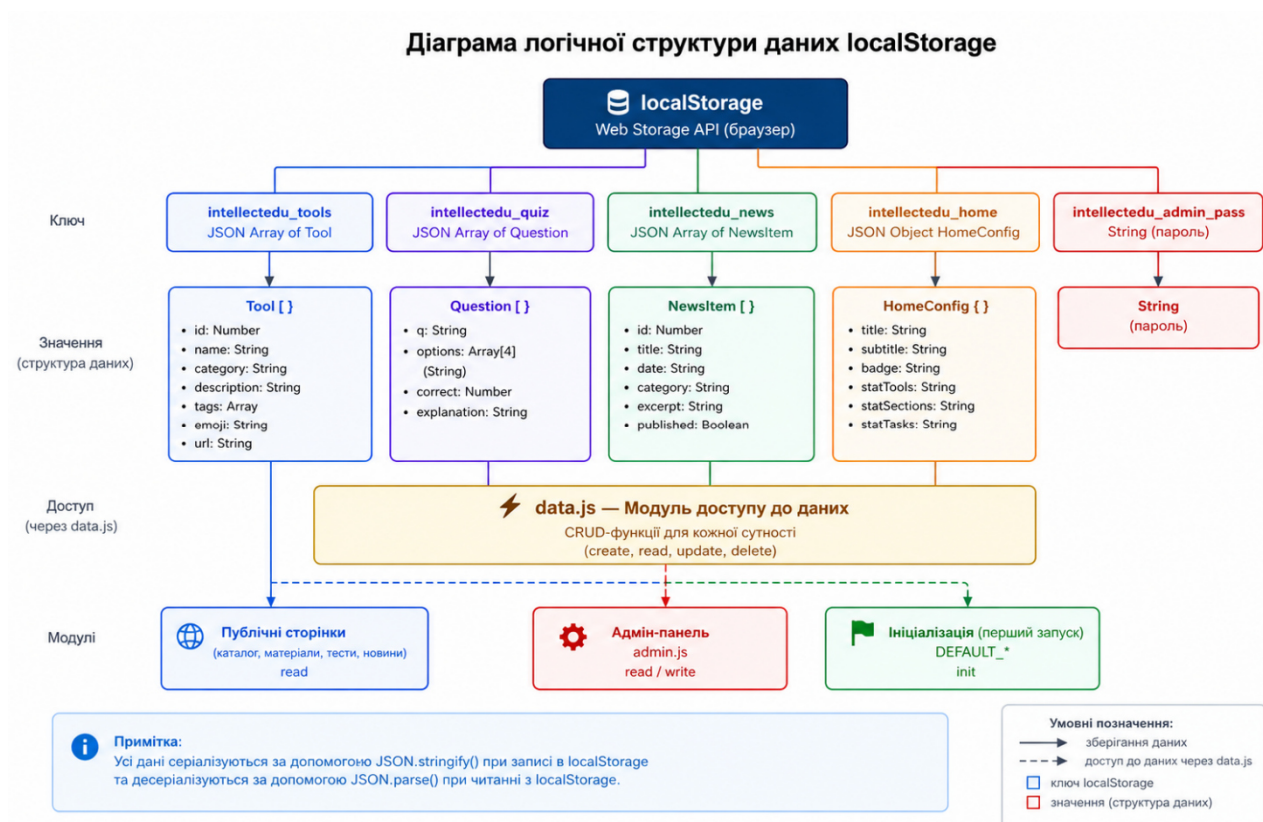


Рисунок 2.8 – Діаграма логічної структури даних веб-ресурсу

Модуль `data.js` реалізує такі рівні абстракції:

- рівень констант: об'єкт `KEYS` зберігає рядкові ключі `localStorage`, що унеможливорює їх дублювання у коді;
- рівень дефолтних даних: константи `DEFAULT_TOOLS`, `DEFAULT_QUIZ`, `DEFAULT_NEWS`, `DEFAULT_HOME` містять початкові значення для ініціалізації сховища при першому запуску;
- рівень приватних функцій: `_get(key, defaultVal)` та `_set(key, value)` інкапсулюють роботу з `JSON.parse/stringify` та обробку помилок;
- рівень публічних функцій: CRUD-функції для кожної сутності (`getTools`, `addTool`, `updateTool`, `deleteTool` тощо) утворюють публічний API модуля. Повний перелік CRUD-операцій, реалізованих у модулі `data.js`, наведено у таблиці 2.10.

Таблиця 2.10 – CRUD-операції модуля data.js

Сутність	Операція	Функція data.js	Опис
Tool	Read	getTools()	Отримати весь масив інструментів
Tool	Create	addTool(item)	Додати новий інструмент, присвоїти id
Tool	Update	updateTool(id, data)	Оновити поля інструменту за id
Tool	Delete	deleteTool(id)	Видалити інструмент за id
Question	Read	getQuiz()	Отримати масив питань тесту
Question	Create	addQuestion(item)	Додати нове питання
Question	Update	updateQuestion(id,d)	Оновити питання за індексом
Question	Delete	deleteQuestion(id)	Видалити питання за індексом
NewsItem	Read	getNews()	Отримати всі новини
NewsItem	Read	getPublishedNews()	Отримати лише опубліковані новини
NewsItem	Create	addNewsItem(item)	Додати новину, присвоїти id
NewsItem	Update	updateNewsItem(id,d)	Оновити новину за id
NewsItem	Delete	deleteNewsItem(id)	Видалити новину за id
HomeConfig	Read	getHomeData()	Отримати конфігурацію головної
HomeConfig	Update	saveHomeData(data)	Зберегти конфігурацію головної

Схема технології доступу до даних наведена на рисунку 2.9. Схема ілюструє послідовність виклику функцій при читанні даних (наприклад, при завантаженні каталогу): сторінка catalog.html викликає getTools() → _get() → localStorage.getItem() → JSON.parse() → повернення масиву об'єктів Tool.

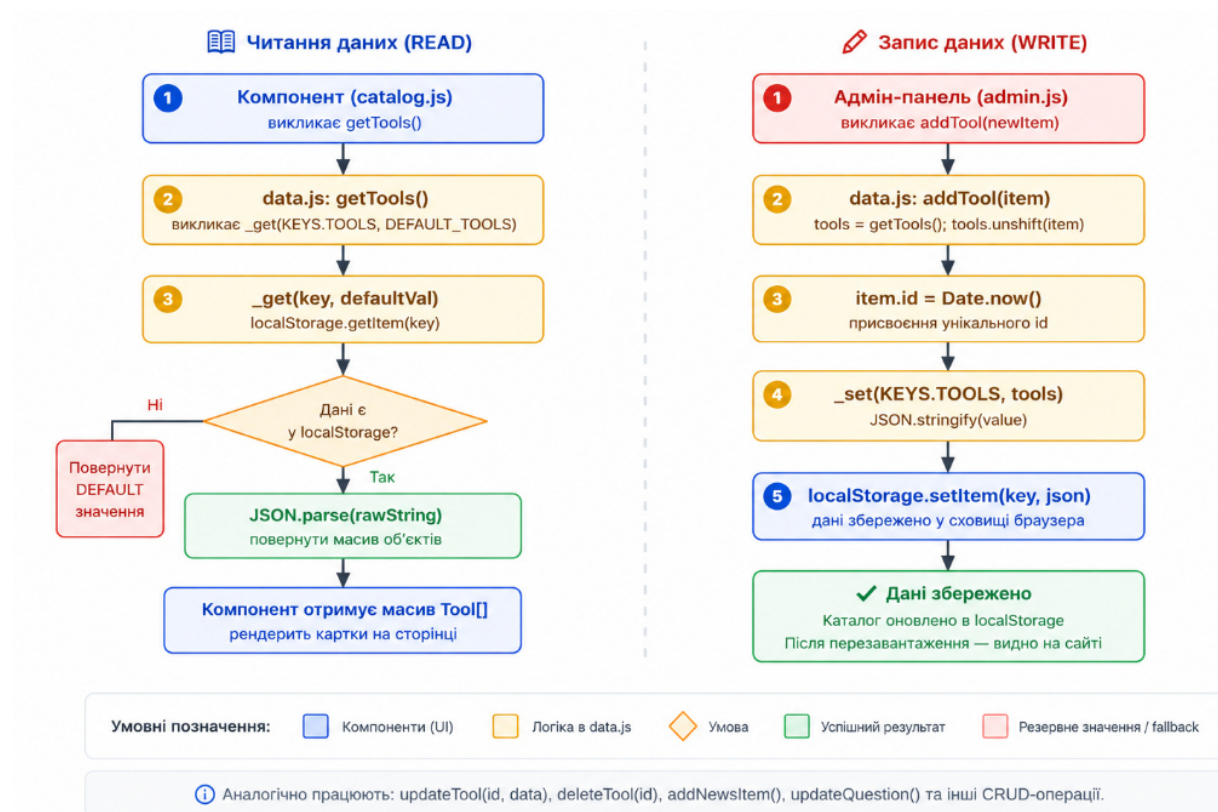


Рисунок 2.9 – Схема технології доступу до даних

РОЗДІЛ 3

РОЗРОБКА ВЕБРЕСУРСУ ДЛЯ ВИКОРИСТАННЯ СУЧАСНИХ ІНТЕЛЕКТУАЛЬНИХ ЦИФРОВИХ ІНСТРУМЕНТІВ НАВЧАЛЬНОГО ПРИЗНАЧЕННЯ

3.1 Практична реалізація об'єкта проектування. Побудова блок – схем модулів програми

Першим кроком реалізації є визначення глобальної системи дизайн-токенів у файлі `base.css`. Всі кольори, відступи, радіуси та типографічні параметри визначаються як CSS-змінні у кореновому селекторі `:root`, що забезпечує єдиний стиль по всьому ресурсу та спрощує внесення змін до дизайну [22]. Лістинг 3.1 демонструє реалізацію системи дизайн-токенів у файлі `css/base.css`.

Лістинг 3.1 – Система дизайн-токенів у файлі `base.css`

```
:root {
  --bg:           #0a0a0a;           /* основний фон */
  --bg-card:      #111111;           /* фон карток */
  --bg-hover:     #161616;           /* фон при наведенні */
  --border:       #1e1e1e;           /* колір бордера */
  --border-hi:    #2e2e2e;           /* активний бордер */
  --accent:       #7c6cff;           /* акцентний колір */
  --text:         #f0f0f0;           /* основний текст */
  --text-sub:     #888888;           /* другорядний текст */
  --font-head:    'Syne', sans-serif; --font-body: 'DM Sans', sans-serif;
  --r-sm: 6px;    --r-md: 12px; --r-lg: 20px; --r-xl: 32px;
  --nav-h: 64px;  --max-w: 1160px;
  --trans: 0.25s cubic-bezier(0.4, 0, 0.2, 1);}
```

кінець лістингу 3.1

Така архітектура дозволяє змінити кольорову схему всього ресурсу редагуванням лише кількох рядків у файлі `base.css`, не торкаючись решти стилів. Акцентний колір `--accent (#e8ff47)` використовується більш ніж у 40 місцях у різних CSS-файлах, тому централізоване визначення є критично важливим для підтримки консистентності.

Центральним елементом JavaScript-архітектури є модуль `data.js`, який реалізує патерн `Repository` для централізованого керування даними через

localStorage. Модуль складається з трьох шарів: константи ключів (KEYS), приватні функції доступу (_get, _set) та публічні CRUD-функції для кожної сутності [23]. Лістинг 3.2 демонструє реалізацію константного об'єкта KEYS, приватних функцій та ініціалізації даних у модулі data.js.

Лістинг 3.2 – Константи, приватні функції та ініціалізація даних (data.js)

```
const KEYS = {
  TOOLS: 'intellectedu_tools', QUIZ: 'intellectedu_quiz',
  NEWS: 'intellectedu_news', HOME: 'intellectedu_home',
  ADMIN_PASS: 'intellectedu_admin_pass',};
function _get(key, defaultVal) {
  try { const raw = localStorage.getItem(key);
    return raw ? JSON.parse(raw) : defaultVal;}
  catch (e) { console.warn('localStorage read error:', e);
    return defaultVal;} }
function _set(key, value) {
  localStorage.setItem(key, JSON.stringify(value));}
function getTools() { return _get(KEYS.TOOLS, DEFAULT_TOOLS);}
function addTool(item) { const tools = getTools();
  const newItem = { ...item, id: Date.now() };
  tools.unshift(newItem); _set(KEYS.TOOLS, tools);return newItem;}
function updateTool(id, data) { const tools = getTools();
  const idx = tools.findIndex(t => t.id === id);
  if (idx === -1) return false;
  tools[idx] = { ...tools[idx], ...data }; _set(KEYS.TOOLS, tools);
  return true;}
function deleteTool(id) {
  _set(KEYS.TOOLS, getTools().filter(t => t.id !== id));}
```

кінець лістингу 3.2

Особливої уваги заслуговує реалізація функції _get: вона обгортає JSON.parse у блок try-catch для обробки випадків, коли localStorage містить пошкоджені дані. У разі помилки функція повертає значення за замовчуванням, що унеможливорює «падіння» сторінки через некоректні дані у сховищі.

Головна сторінка веб-ресурсу складається з п'яти секцій: hero (заголовна), stats-bar (статистика), features (можливості), tools-preview (анонс інструментів) та news-section (новини). Вміст hero-секції та статистика підтягуються динамічно з localStorage через модуль home.js при кожному завантаженні сторінки [24]. Скріншот головної сторінки веб-ресурсу «ІнтелектЕду» наведено на рисунку 3.1.

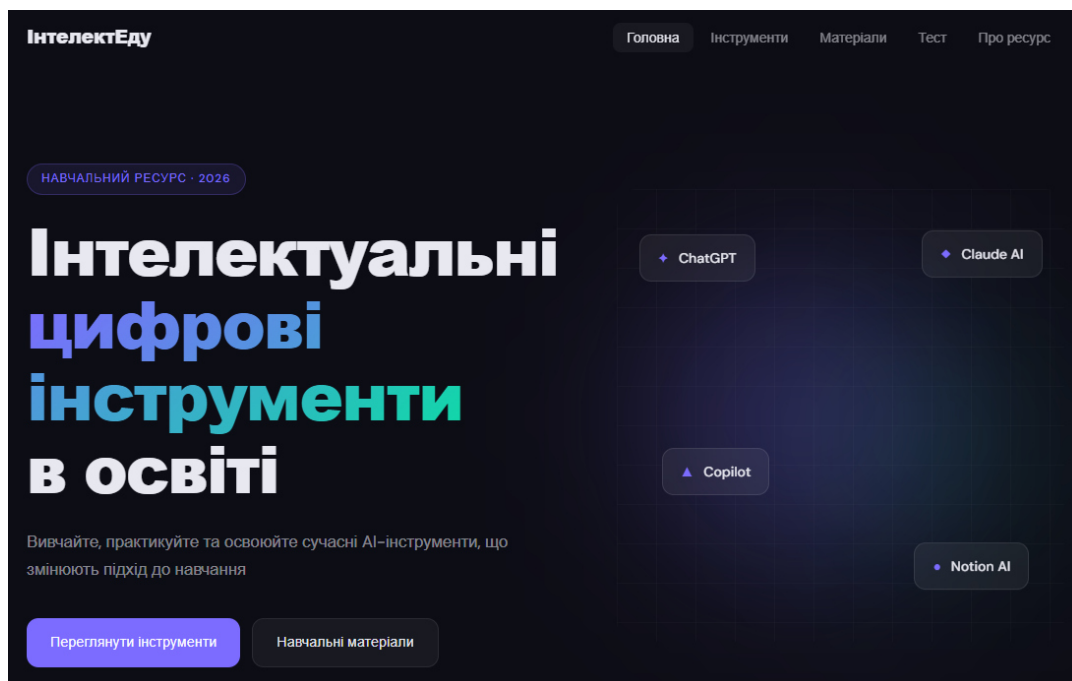


Рисунок 3.1 – Головна сторінка навчального веб-ресурсу «ІнтелектЕду»

Лістинг 3.3 демонструє реалізацію динамічного наповнення hero-секції головної сторінки з даних localStorage у модулі home.js.

Лістинг 3.3 – Динамічне наповнення hero-секції (home.js)

```
document.addEventListener('DOMContentLoaded', () => {
  const home = getHomeData();
  const titleEl = document.getElementById('heroTitle');
  const subEl = document.getElementById('heroSub');
  const badgeEl = document.getElementById('heroBadge');
  if (titleEl && home.title) titleEl.innerHTML = home.title;
  if (subEl && home.subtitle) subEl.textContent = home.subtitle;
  if (badgeEl && home.badge) badgeEl.textContent = home.badge;
  const stats = {
    statTools: document.getElementById('statTools'),
    statSections: document.getElementById('statSections'),
    statTasks: document.getElementById('statTasks'), };
  Object.entries(stats).forEach(([key, el]) => {
    if (el && home[key]) el.textContent = home[key];}); });
```

кінець лістингу 3.3

Каталог AI-інструментів є центральним навчальним модулем ресурсу. Сторінка catalog.html відображає картки інструментів, які динамічно генеруються JavaScript на основі даних з localStorage (ліст. 3.4).

Лістинг 3.4 – Рендеринг карток каталогу та фільтрація (catalog.js)

```
function renderTools(filter = 'all') {
  const grid = document.getElementById('toolsGrid');
  const tools = getTools(); const shown = filter === 'all'
    ? tools: tools.filter(t => t.category === filter);
  if (!shown.length) { grid.innerHTML = '<p class="no-results">
    Інструментів у цій категорії не знайдено.</p>';return; }
  grid.innerHTML = shown.map((tool, i) =>
    <div class="tool-card" data-cat="${tool.category}"
      style="--delay:${i * 0.05}s"> <div class="tool-card-top">
        <div class="tool-logo">${tool.emoji} || '🤖'</div>
        <span class="tool-category">${tool.category}</span> </div>
        <h3>${escHtml(tool.name)}</h3> <p>${escHtml(tool.description)}</p>
        <div class="tool-tags"> ${tool.tags.map(t=>`<span class="tool-tag">
          ${escHtml(t)}</span>`).join('')} </div>
        ${tool.url ? `<a href="${tool.url}" target="_blank"
          class="tool-card-link">Відкрити →</a>` : ''} </div>`).join('');
  document.addEventListener('DOMContentLoaded', () => {renderTools();
  document.querySelectorAll('.filter-btn').forEach(btn => {
    btn.addEventListener('click', () => {
      document.querySelectorAll('.filter-btn')
        .forEach(b => b.classList.remove('active'));
      btn.classList.add('active'); renderTools(btn.dataset.filter);}); }); });
}
```

кінець лістингу 3.4

Підтримується фільтрація за п'ятьма категоріями: «Усі», «Асистенти», «Дослідження», «Код» та «Продуктивність». Скріншот сторінки каталогу AI-інструментів наведено на рисунку 3.2.

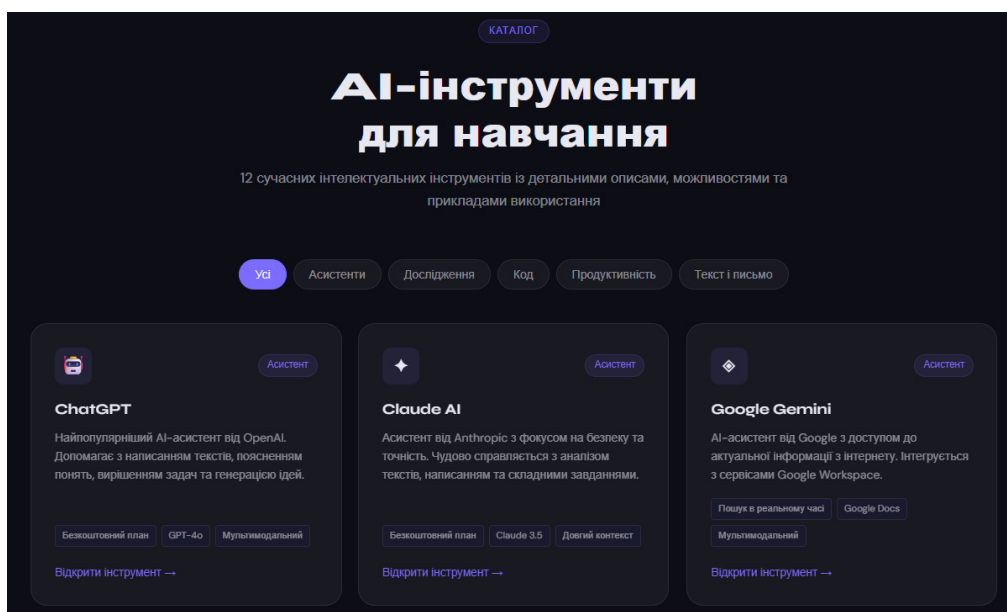


Рисунок 3.2 – Сторінка каталогу AI-інструментів веб-ресурсу

Важливим аспектом реалізації є захист від XSS-атак: перед вставкою даних із localStorage у innerHTML всі рядки екрануються функцією `escHtml()`, яка замінює символи `<`, `>`, `&`, `"` на відповідні HTML-сутності.

Модуль тестування реалізований у файлі `js/quiz.js` як скінченний автомат зі станами. Стан системи описується трьома змінними: `current`, `score` та `answered`. Питання завантажуються з localStorage через `getQuiz()`. Скріншот сторінки тестування наведено на рисунку 3.3.

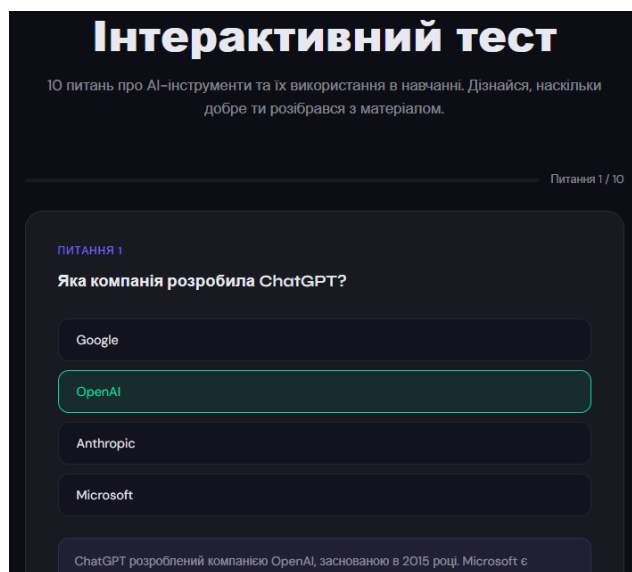


Рисунок 3.3 – Сторінка інтерактивного тестування

Лістинг 3.5 демонструє реалізацію двох ключових функцій модуля тестування – `render()` та `handleAnswer()`. Функція `render()` відповідає за генерацію HTML-розмітки поточного питання та оновлення індикатора прогресу. Вона зчитує масив питань з localStorage через `getQuiz()`, визначає поточний стан тесту за змінною `current` та динамічно формує DOM-структуру картки питання з варіантами відповідей. Після кожного рендерингу функція прив'язує обробники подій `click` до кнопок відповідей, перевіряючи при цьому значення прапорця `answered`, щоб унеможливити подвійний вибір. Функція `handleAnswer()` приймає індекс обраної відповіді, та правильної відповіді та текст пояснення. Вона блокує всі кнопки, підсвічує правильний варіант, а у разі помилки – хибний варіант класом `wrong`, після чого відображає блок пояснення з відповідним стилем.

Лістинг 3.5 – Функції render() та handleAnswer() модуля тестування (quiz.js)

```

let current = 0, score = 0, answered = false; function render() {
  const container = document.getElementById('quizContainer');
  const questions = getQuiz();
  if (current >= questions.length) {renderResult(container,
questions.length); return; } const q = questions[current];
  const pct = Math.round((current / questions.length) * 100);
  container.innerHTML = ` <div class="quiz-progress">
    <div class="progress-text">
      <span>Питання ${current+1} з ${questions.length}</span>
      <span>${pct} %</span></div> <div class="progress-bar">
        <div class="progress-fill" style="width:${pct} %"></div>
    </div></div>
    <div class="quiz-card"><div class="quiz-q-text">${escHtml(q.q)}</div>
    <div class="quiz-options" id="options">${q.options.map((opt, i) => `
      <button class="quiz-option" data-index="${i}">
        <span class="opt-letter">${'ABCD'[i]}</span>${escHtml(opt)}
      </button>` ).join('')}</div>
    <div class="quiz-feedback" id="feedback"></div></div>;
  container.querySelectorAll('.quiz-option').forEach(btn => {
    btn.addEventListener('click', () => { if (answered) return;
      answered = true; handleAnswer(parseInt(btn.dataset.index), q.correct,
        q.explanation, questions.length); }); });
  function handleAnswer(chosen, correct, explanation, total) {
    const opts = document.querySelectorAll('.quiz-option');
    const feedback = document.getElementById('feedback');
    opts.forEach(b => b.disabled = true);
    opts[correct].classList.add('correct');
    if (chosen === correct) { score++;
      feedback.className = 'quiz-feedback correct-fb show';
      feedback.innerHTML = `✓ Правильно! ${escHtml(explanation)}`;
    } else { opts[chosen].classList.add('wrong');
      feedback.className = 'quiz-feedback wrong-fb show';
      feedback.innerHTML = `X Неправильно. ${escHtml(explanation)}`;
    }
    const nextBtn = document.getElementById('nextBtn');
    if (nextBtn) { nextBtn.style.display = 'inline-flex';
      nextBtn.addEventListener('click', () => {
        answered = false; current++; render(); }, { once: true }); } }

```

кінець лістингу 3.5

Сторінка навчальних матеріалів містить sticky-sidebar з навігацією по розділах. Активний розділ у sidebar підсвічується автоматично при прокрутці сторінки за допомогою Intersection Observer API – сучасного браузерного механізму, що не навантажує головний потік виконання порівняно з традиційним обробником події scroll [25]. Скріншот сторінки навчальних матеріалів наведено на рисунку 3.4.

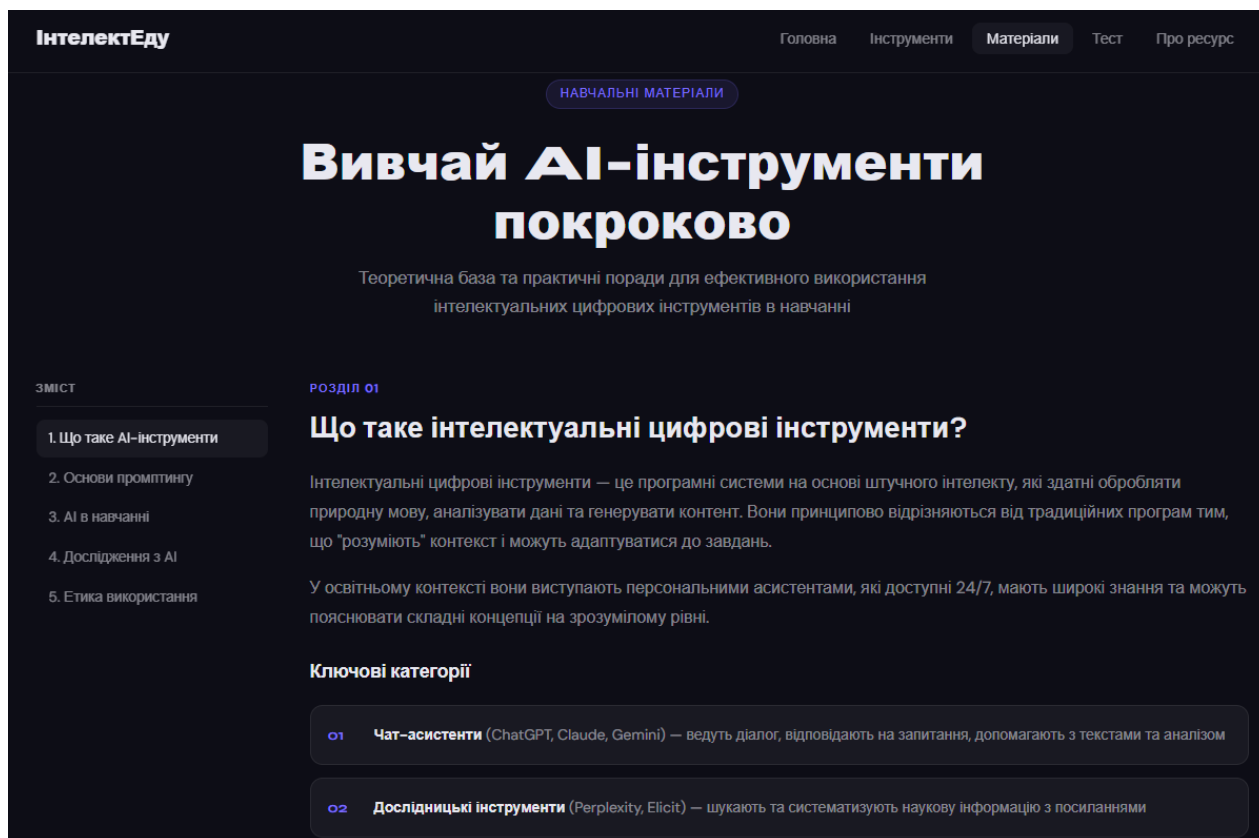


Рисунок 3.4 – Сторінка навчальних матеріалів з навігаційним sidebar

Лістинг 3.6 демонструє реалізацію автоматичного відстеження активного розділу за допомогою Intersection Observer API у файлі `js/materials.js`. На відміну від класичного підходу з обробником події `scroll`, який виконується синхронно в головному потоці браузера і може спричиняти затримки, Intersection Observer є асинхронним нативним механізмом браузера, що суттєво підвищує продуктивність на мобільних пристроях.

Спостерігач налаштований з параметром `rootMargin: «-104px 0px -60 % 0px»`, де `-104px` відповідає висоті фіксованого `navbar` плюс додатковий відступ, а `-60 %` від нижнього краю визначає зону активності розділу. Коли розділ перетинає цю зону видимості, спостерігач знаходить відповідне посилання у sidebar за атрибутом `href` та перемикає клас `active`.

Додатково реалізовано плавну прокрутку до цільового розділу при кліку на посилання через `scrollIntoView` з параметром `behavior: 'smooth'`, що забезпечує зручну навігацію без стрибків сторінки.

Лістинг 3.6 – Відстеження активного розділу через Intersection Observer

```
document.addEventListener('DOMContentLoaded', () => {
  const sections = document.querySelectorAll('.mat-section');
  const sidebarLinks = document.querySelectorAll('.sidebar-nav a');
  const observer = new IntersectionObserver((entries) => {
    entries.forEach(entry => {
      if (entry.isIntersecting) { const id = entry.target.id;
        sidebarLinks.forEach(link => {link.classList.toggle('active',
          link.getAttribute('href') === `#${id}` );}); } });},
    { rootMargin: `-${64 + 40}px 0px -60% 0px`, threshold: 0 });
  sections.forEach(section => observer.observe(section));
  sidebarLinks.forEach(link => {link.addEventListener('click', (e) => {
    e.preventDefault(); const target = document.querySelector(
      link.getAttribute('href') );
    if (target) target.scrollIntoView({ behavior: 'smooth', block:
      'start' }); }); }); });
```

кінець лістингу 3.6

Параметр `rootMargin` встановлено як «-104px 0px -60 % 0px», що означає: розділ вважається активним тоді, коли його верхня межа перетинає горизонтальну лінію на 104px нижче від верху вікна (відступ для `fixed-navbar`) і не перетинає нижню 60 % вікна. Такий підхід забезпечує природну поведінку: активним є той розділ, який реально читає користувач.

Адміністративна панель є окремою захищеною сторінкою, що дозволяє керувати всім контентом ресурсу без редагування коду. Доступ захищений паролем, що зберігається в `localStorage`. Панель реалізована як односторінковий застосунок з вкладками: Dashboard, Інструменти, Тест, Новини, Головна, Налаштування (рис. 3.5).

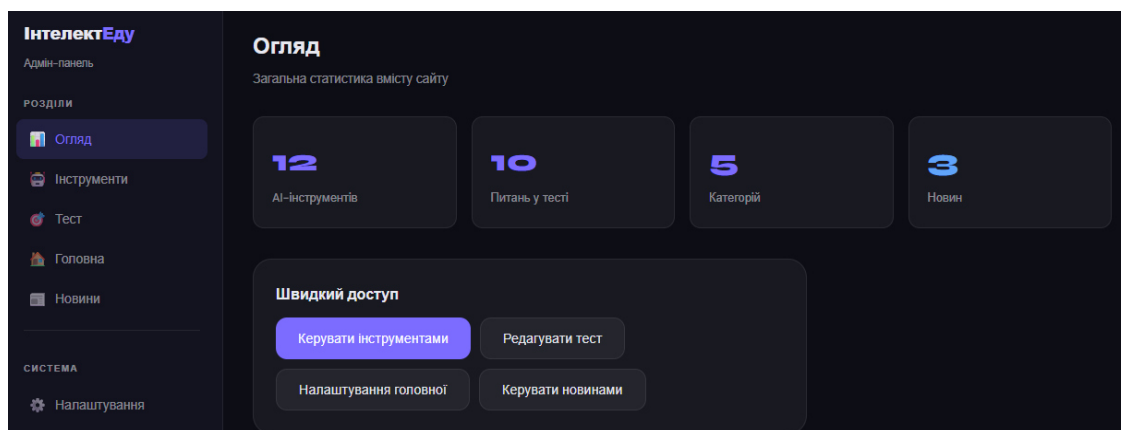


Рисунок 3.5 – Адміністративна панель керування контентом

Лістинг 3.7 демонструє реалізацію системи авторизації та механізму перемикання вкладок адміністративної панелі у файлі js/admin.js. Функція checkAuth() зчитує збережений пароль з localStorage через getAdminPass() та порівнює його з введеним користувачем значенням. У разі збігу функція приховує екран авторизації (loginScreen) та відображає основний інтерфейс панелі (adminPanel), одночасно викликаючи refreshDashboard() для наповнення дашборду актуальною статистикою. При невірному паролі відображається повідомлення про помилку через функцію showToast(). Функція switchTab() реалізує логіку перемикання між шістьма вкладками панелі. Вона послідовно знімає клас active з усіх вкладок та навігаційних кнопок, після чого активує цільовий елемент за переданим tabId. Важливою деталлю є виклик відповідної функції оновлення даних при переході на вкладку: наприклад, при відкритті вкладки «Інструменти» викликається refreshToolsTab(), що гарантує актуальність відображених даних навіть після змін, внесених на інших вкладках.

Лістинг 3.7 – Авторизація та перемикання вкладок адмін-панелі (admin.js)

```
function checkAuth() {
  const savedPass = getAdminPass();
  const entered    = document.getElementById('passInput')?.value;
  if (entered === savedPass) {
    document.getElementById('loginScreen').style.display = 'none';
    document.getElementById('adminPanel').style.display  = 'flex';
    refreshDashboard(); }
  else {
    showToast('Невірний пароль', 'error'); } }

function switchTab(tabId) {
  document.querySelectorAll('.admin-tab').forEach(tab => {
    tab.classList.remove('active'); });
  document.querySelectorAll('.admin-nav-btn').forEach(btn => {
    btn.classList.remove('active'); });
  const tab = document.getElementById(tabId);
  const btn = document.querySelector(`[data-tab="${tabId}"]`);
  if (tab) tab.classList.add('active');
  if (btn) btn.classList.add('active');
  if (tabId === 'tabDashboard') refreshDashboard();
  if (tabId === 'tabTools')     refreshToolsTab();
  if (tabId === 'tabQuiz')      refreshQuizTab();
  if (tabId === 'tabNews')      renderNewsTable();}
```

кінець лістингу 3.7

Спільний модуль `main.js` підключається до кожної сторінки ресурсу та виконує роль універсального ініціалізатора інтерфейсних компонентів. Він забезпечує три незалежні функції, що покращують користувацький досвід на всіх сторінках без дублювання коду.

Перша функція – `scroll`-ефект для фіксованого `navbar`. При прокрутці сторінки більш ніж на 20 пікселів до елемента `navbar` додається клас `scrolled`, який через CSS підсилює тінь та злегка змінює прозорість фону. Це візуально відокремлює навігаційну панель від контенту і покращує читабельність. Обробник події `scroll` зареєстровано з параметром `{ passive: true }`, що повідомляє браузер про відсутність виклику `preventDefault()` та дозволяє браузеру оптимізувати прокрутку без затримок, що особливо критично на мобільних пристроях.

Друга функція – `burger`-меню для мобільних пристроїв. При ширині екрана менше 768px горизонтальне навігаційне меню замінюється кнопкою-гамбургером. При натисканні на неї до елемента `nav-links` додається клас `open`, який через CSS розгортає меню у вертикальний список. Реалізовано також закриття меню при кліку за його межами через обробник на документі, що є стандартною практикою UX для мобільних меню.

Третя функція – `fade-in` анімації для елементів сторінки при прокрутці. Всі картки інструментів, секції матеріалів та інші блокові елементи спочатку отримують стиль `opacity: 0` та `transform: translateY(20px)`, роблячись невидимими. `Intersection Observer` стежить за їхньою появою у `viewport`: щойно елемент перетинає поріг 10 % видимості, йому встановлюється `opacity: 1` та `transform: translateY(0)` з CSS-переходом тривалістю 0.5 секунди.

Після спрацювання спостерігач відписується від елемента через `unobserve()`, уникаючи зайвих обчислень. Лістинг 3.8 демонструє повну реалізацію всіх трьох функцій модуля `main.js`.

Лістинг 3.8 – Navbar, burger-меню та Intersection Observer анімації (main.js)

```
document.addEventListener('DOMContentLoaded', () => {
  const navbar = document.getElementById('navbar');
  if (navbar) { window.addEventListener('scroll', () => {
    navbar.classList.toggle('scrolled', window.scrollY > 20);
  }, { passive: true }); }
  const burger = document.getElementById('burger');
  const navLinks = document.querySelector('.nav-links');
  if (burger && navLinks) {
    burger.addEventListener('click', () => {
      navLinks.classList.toggle('open'); });
    document.addEventListener('click', (e) => {
      if (!burger.contains(e.target) &&
        !navLinks.contains(e.target)) {
        navLinks.classList.remove('open'); } }); }
  const observer = new IntersectionObserver((entries) => {
    entries.forEach(entry => {
      if (entry.isIntersecting) {
        entry.target.style.opacity = '1';
        entry.target.style.transform = 'translateY(0)';
        observer.unobserve(entry.target); } });
    }, { threshold: 0.1 });
  document.querySelectorAll(
    '.feature-card, .tool-card, .about-card, .mat-section'
  ).forEach(el => { el.style.opacity = '0';
    el.style.transform = 'translateY(20px)';
    el.style.transition = 'opacity 0.5s ease, transform 0.5s ease';
    observer.observe(el); });
});
```

кінець лістингу 3.8

Важливим технічним рішенням є використання опції `{ passive: true }` для обробника події `scroll`. Це повідомляє браузеру, що обробник не буде викликати `preventDefault()`, що дозволяє браузеру оптимізувати прокрутку та уникнути затримок на мобільних пристроях.

Сторінка `about.html` містить структуровану інформацію про ресурс: мету, цільову аудиторію, структуру та технічний стек. Адаптивний дизайн реалізовано через CSS Grid із медіа-запитами, що забезпечує коректне відображення на мобільних пристроях.

Скріншот сторінки «Про ресурс» наведено на рисунку 3.6.

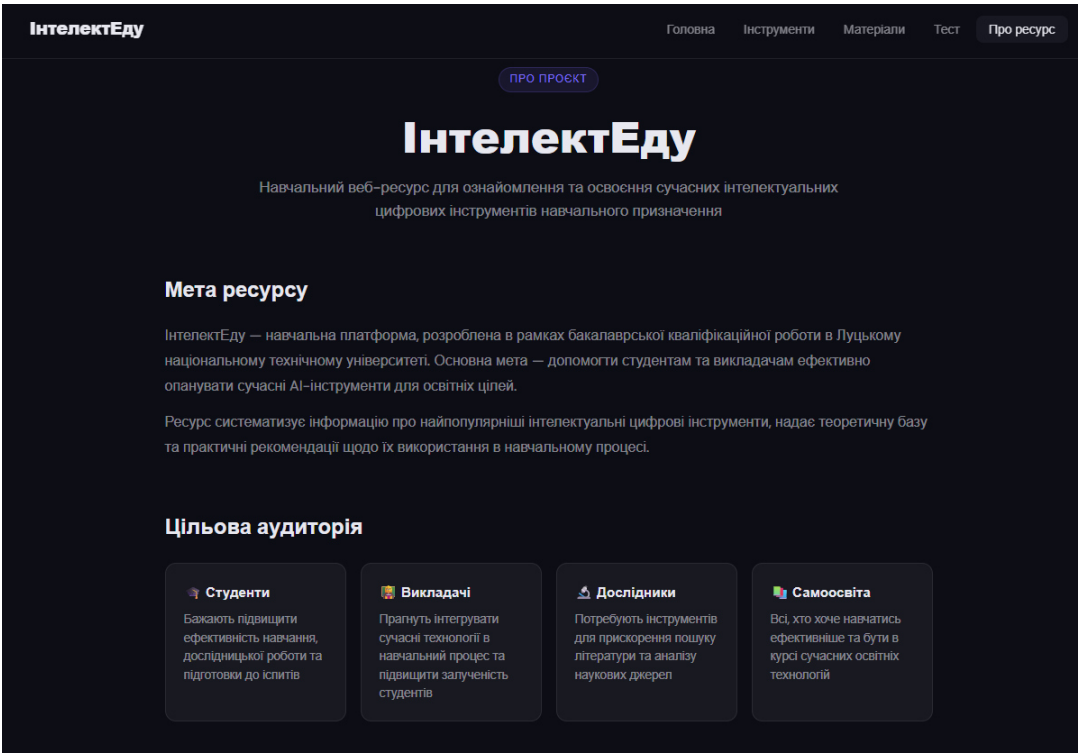


Рисунок 3.6 – Сторінка «Про ресурс» навчального веб-ресурсу

Скріншот мобільної версії головної сторінки зображено на рисунку 3.7.

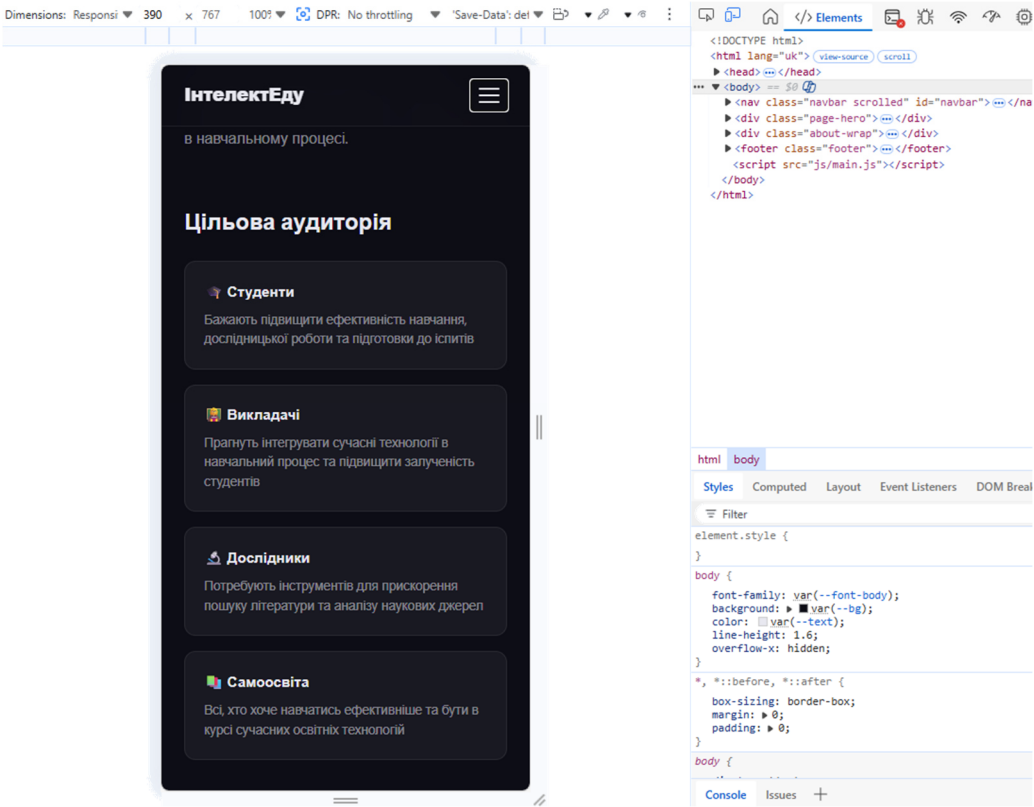


Рисунок 3.7 – Мобільна версія веб-ресурсу (ширина 390px)

Адаптивність веб-ресурсу реалізована за допомогою системи CSS-медіа-запитів без використання зовнішніх фреймворків. Основний контрольний рядок `@media (max-width: 900px)` у файлах `home.css` та `pages.css` охоплює планшети та смартфони з шириною екрана до 900 пікселів і вносить такі зміни до макету.

По-перше, двоколонковий `grid-layout` головної сторінки перемикається на одноколонковий через переоголошення `grid-template-columns: 1fr`. Картки `feature-card` вишиковуються у вертикальний стек, а `hero`-контент центрується по осі.

По-друге, на сторінці навчальних матеріалів двоколонковий макет «`sidebar`+основний контент» замінюється одноколонковим. Сам `sidebar` при цьому приховується через `display: none`, оскільки на вузьких екранах фіксована бічна панель займає надто багато корисного простору. Навігація по розділах на мобільній версії здійснюється через звичайну прокрутку.

По-третє, горизонтальне навігаційне меню отримує `display: none` і замінюється `burger`-кнопкою, яка на десктопній версії прихована через `display: none`. При натисканні на `burger`-кнопку модуль `main.js` додає клас `open` до `nav-links`, що через CSS розгортає меню у вертикальний випадаючий список під `navbar` з повною шириною екрана.

Додатково медіа-запит `@media (max-width: 600px)` застосовується до сторінки каталогу для переведення сітки карток інструментів у суворо одноколонковий режим на смартфонах з вузьким екраном, а також до модального вікна адмін-панелі — зменшується внутрішній відступ для коректного відображення форм на невеликих екранах.

3.2 Тестування та налагодження системи

Тестування є обов'язковим етапом розробки програмного забезпечення, що дозволяє виявити та усунути помилки до введення системи в експлуатацію. Для навчального веб-ресурсу «ІнтелектЕду» проведено функціональне тестування всіх модулів, тестування сумісності з різними браузерами та пристроями, а також тестування коректності роботи з `localStorage`.

У процесі розробки та тестування застосовувались такі методи:

- ручне функціональне тестування – перевірка кожного модуля відповідно до визначених тестових сценаріїв. Тестування проводилось у браузері Google Chrome з використанням вбудованих інструментів розробника (DevTools);
- тестування за принципом «чорної скриньки» – перевірка системи з позиції кінцевого користувача без аналізу внутрішньої логіки. Перевірялась відповідність результатів очікуваній поведінці при різних вхідних даних;
- тестування граничних значень – перевірка поведінки при порожньому localStorage, максимальній кількості елементів, некоректних вхідних даних у форми адмін-панелі;
- тестування сумісності – перевірка коректного відображення та функціонування у чотирьох браузерах: Google Chrome, Mozilla Firefox, Microsoft Edge та Safari;
- тестування адаптивності – перевірка відображення при різних роздільних здатностях екрана за допомогою режиму Device Mode у Chrome DevTools (390px, 768px, 1024px, 1920px).

Для налагодження JavaScript-коду використовувались такі інструменти Chrome DevTools:

- Console – виведення діагностичних повідомлень та перехоплення помилок виконання;
- Sources → Debugger – встановлення breakpoints та покрокове виконання коду для виявлення логічних помилок;
- Application → Local Storage – перегляд та редагування вмісту localStorage для перевірки коректності зберігання даних;
- Network – аналіз завантаження ресурсів та часу відповіді;
- Lighthouse – автоматизована перевірка продуктивності, доступності та найкращих практик.

Скріншот інструментів розробника Chrome DevTools під час тестування localStorage наведено на рисунку 3.8.

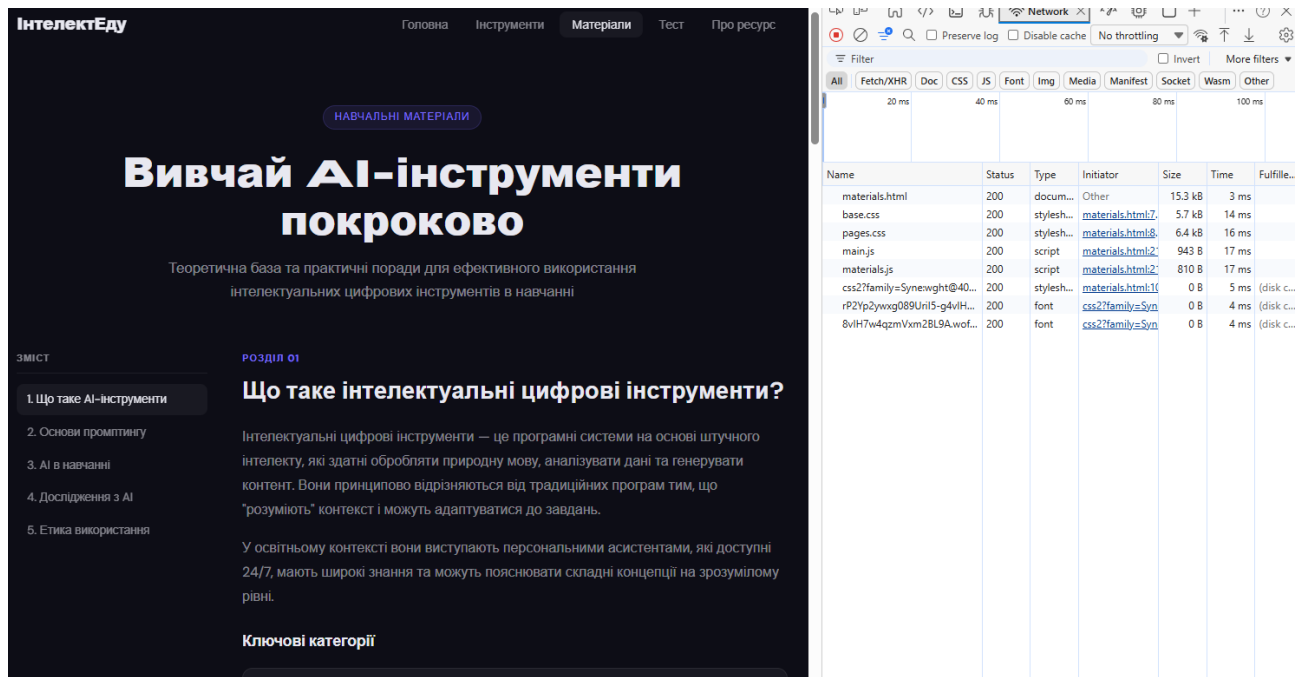


Рисунок 3.8 – Chrome DevTools: тестування localStorage

Функціональне тестування проводилось за розробленим планом тестових сценаріїв. Для кожного модуля системи визначено вхідну дію, очікуваний результат та фактичний результат. Зведені результати тестування наведено у таблиці 3.1.

Таблиця 3.1 – Результати функціонального тестування веб-ресурсу

№	Модуль / функція	Тестовий сценарій	Очікуваний результат	Статус
1	Navbar (main.js)	Прокрутити сторінку > 20px	Клас scrolled додається до navbar	✓ Пройдено
2	Burger-меню (main.js)	Натиснути burger на екрані 390px	Меню розгортається вертикально	✓ Пройдено
3	Каталог (catalog.js)	Відкрити catalog.html без даних у localStorage	Відображаються дефолтні інструменти DEFAULT_TOOLS	✓ Пройдено
4	Фільтрація каталогу	Натиснути фільтр «Дослідження»	Відображаються лише картки category='research'	✓ Пройдено
5	Фільтрація каталогу	Натиснути фільтр «Усі» після категорійного	Відображаються всі картки	✓ Пройдено
6	Тест (quiz.js)	Обрати правильну відповідь	Кнопка підсвічується зеленим, score збільшується	✓ Пройдено
7	Тест (quiz.js)	Обрати хибну відповідь	Хибна – червоним, правильна – зеленим; score не змінюється	✓ Пройдено

Продовження таблиці 3.1

№	Модуль / функція	Тестовий сценарій	Очікуваний результат	Статус
8	Тест (quiz.js)	Спробувати обрати відповідь вдруге	Повторний клік ігнорується (answered = true)	✓ Пройдено
9	Результати тесту	Пройти всі 10 питань	Відображається renderResult() з відсотком та статистикою	✓ Пройдено
10	Sidebar (materials.js)	Прокрутити до розділу «Промптинг»	У sidebar підсвічується відповідний пункт	✓ Пройдено
11	Плавна прокрутка	Натиснути посилання в sidebar	Сторінка плавно прокручується до розділу	✓ Пройдено
12	Авторизація (admin.js)	Ввести правильний пароль «admin»	Панель відкривається, dashboard оновлюється	✓ Пройдено
13	Авторизація (admin.js)	Ввести хибний пароль	Відображається toast «Невірний пароль»	✓ Пройдено
14	CRUD інструменти	Додати новий AI-інструмент через форму	Інструмент з'являється у таблиці та каталозі	✓ Пройдено
15	CRUD новини	Додати новину, встановити статус «Чернетка»	Новина збережена, але не відображається на головній	✓ Пройдено
16	CRUD новини	Видалити новину через кнопку «Видалити»	Новина зникає з таблиці та localStorage	✓ Пройдено
17	Персистентність даних	Додати інструмент, перезавантажити сторінку	Доданий інструмент залишається в каталозі	✓ Пройдено
18	Ініціалізація даних	Очистити localStorage, відкрити сторінку	Завантажуються дефолтні дані (DEFAULT_*)	✓ Пройдено

За результатами функціонального тестування всі 18 тестових сценаріїв пройдено успішно. Критичних помилок, що унеможливають використання ресурсу, не виявлено. У процесі тестування було виявлено та усунуто такі незначні дефекти:

- при першому відкритті сторінки quiz.html спостерігалась затримка рендерингу першого питання через синхронне зчитування з localStorage – усунуто додаванням перевірки наявності елемента quizContainer перед викликом render();

- на сторінці materials.html при швидкій прокрутці Intersection Observer інколи не фіксував перехід між розділами – усунуто коригуванням параметру rootMargin з «-80px 0px -50 % 0px» на «-104px 0px -60 % 0px»;

– в адмін-панелі при перемиканні між вкладками таблиця інструментів не оновлювалась – усунено додаванням виклику `refreshToolsTab()` при активації вкладки `tabTools` у функції `switchTab()`.

Тестування сумісності проводилось у чотирьох актуальних браузерях: Google Chrome 124, Mozilla Firefox 125, Microsoft Edge 124 та Safari 17. Перевірялась підтримка всіх використаних веб-технологій: CSS Custom Properties, CSS Grid, Flexbox, localStorage API, Intersection Observer API, JavaScript ES6+, backdrop-filter та CSS-анімацій. Результати тестування сумісності наведено у таблиці 3.2.

Таблиця 3.2 – Результати тестування сумісності з браузерами

Функціональність	Chrome 124+	Firefox 125+	Edge 124+	Safari 17+
Відображення сторінок	✓	✓	✓	✓
CSS Custom Properties	✓	✓	✓	✓
CSS Grid та Flexbox	✓	✓	✓	✓
localStorage API	✓	✓	✓	✓
Intersection Observer API	✓	✓	✓	✓
JavaScript ES6+ (стрілкові функції, template literals)	✓	✓	✓	✓
Google Fonts (Syne, DM Sans)	✓	✓	✓	✓
Адаптивний дизайн (390px–2560px)	✓	✓	✓	✓
Анімації CSS (@keyframes)	✓	✓	✓	✓
Backdrop-filter (blur ефект navbar)	✓	✓	✓	✓

Як видно з таблиці 3.2, всі використані технології підтримуються актуальними версіями чотирьох основних браузерів без виключень. Єдиним нюансом є властивість `backdrop-filter` (розмиття фону navbar), яка у старіших версіях Firefox (до версії 103) потребувала вендорного префіксу `-webkit-backdrop-filter`. У поточних версіях Firefox 125+ ця властивість підтримується без префікса. Скріншот порівняння відображення каталогу у Google Chrome та Mozilla Firefox наведено на рисунку 3.9.

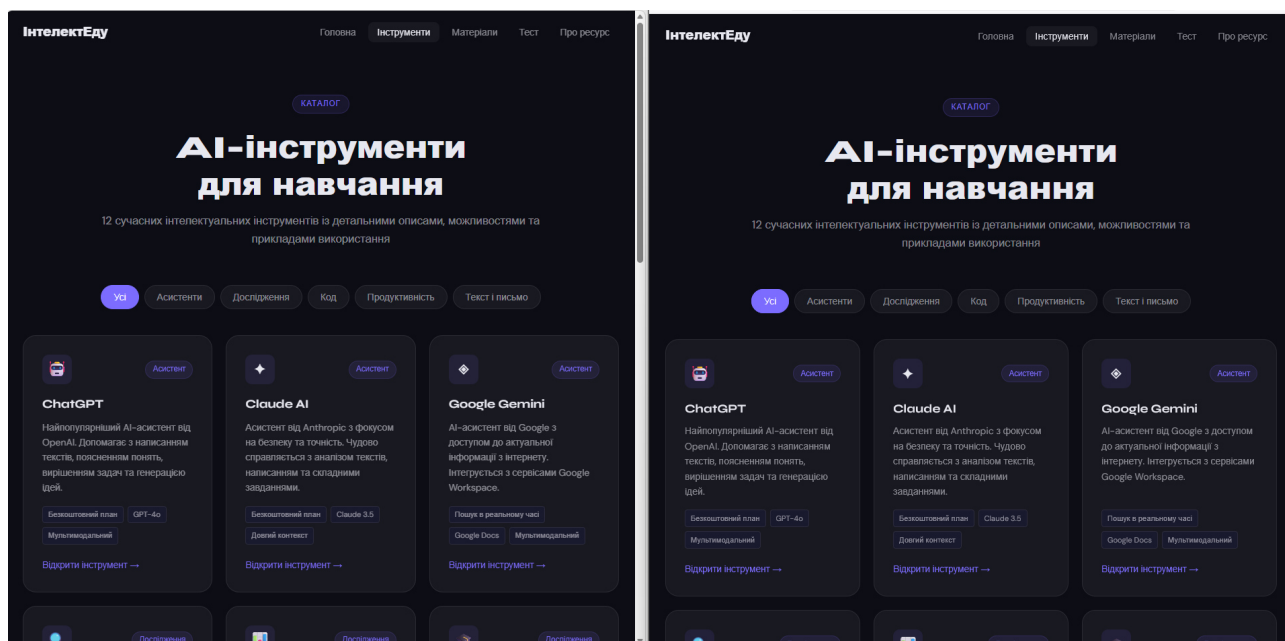


Рисунок 3.9 – Порівняння відображення каталогу у Google Chrome та Mozilla Firefox

Тестування адаптивності проводилось за допомогою режиму Device Mode у Chrome DevTools при чотирьох контрольних роздільних здатностях: 390px (iPhone 14 Pro), 768px (iPad), 1024px (ноутбук) та 1920px (десктоп). Для кожного розміру перевірялись коректність розкладки, читабельність тексту, доступність елементів керування та відсутність горизонтального скролу.

При ширині 390px (мобільний пристрій) перевірялись такі аспекти:

- замість горизонтального меню відображається burger-кнопка та коректно розгортається мобільне меню;
- hero-секція відображається в одну колонку, текст центрується, анімаційний блок приховується;
- картки каталогу, секції матеріалів та блоки результатів тесту відображаються в одну колонку без обрізань;
- форми адмін-панелі коректно відображаються з повною шириною полів введення.

Скріншоти мобільного відображення ресурсу у Chrome DevTools (390px) для каталогу та адмін-панелі наведено на рисунку 3.10.

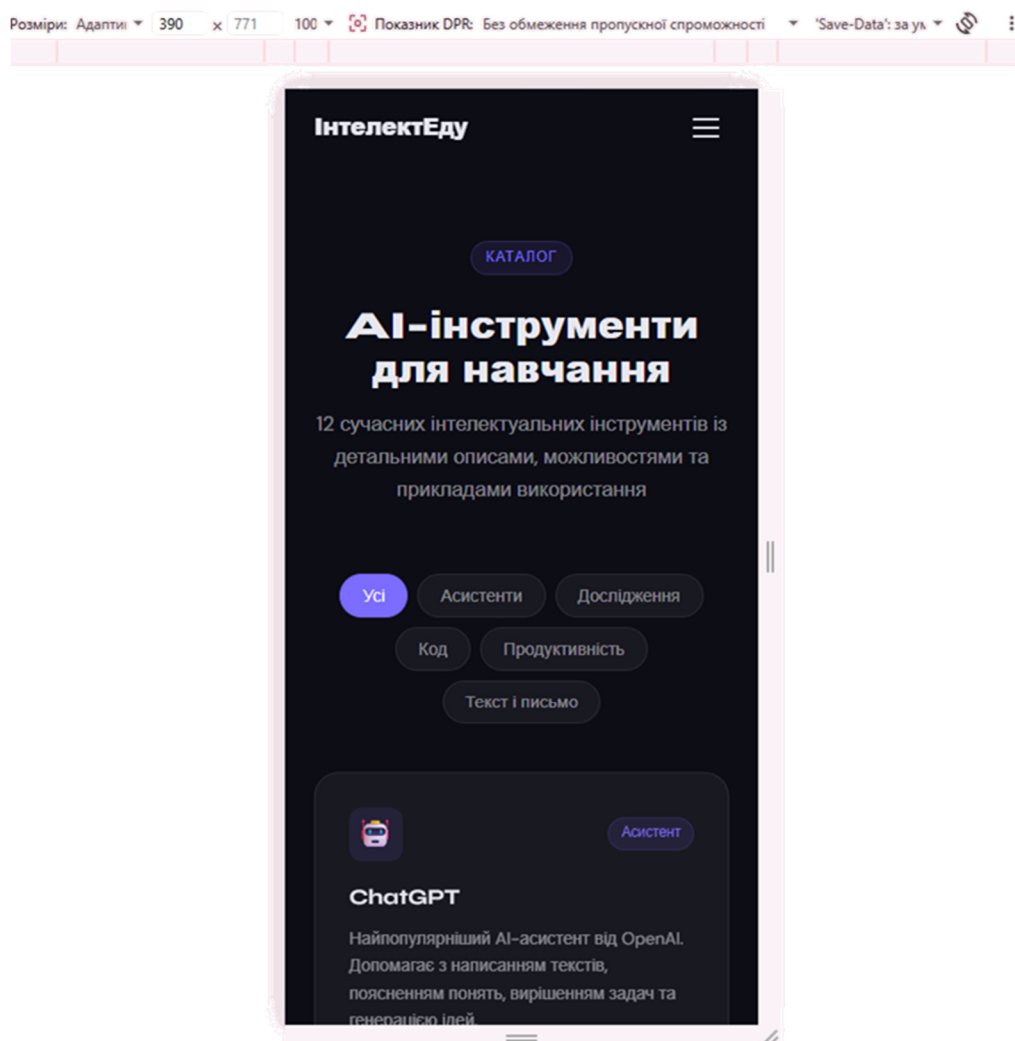


Рисунок 3.10 – Мобільне відображення каталогу та адмін-панелі (390px)

Для комплексної оцінки якості веб-ресурсу проведено аудит за допомогою вбудованого інструменту Lighthouse у Chrome DevTools. Lighthouse оцінює ресурс за чотирма категоріями: Performance (продуктивність), Accessibility (доступність), Best Practices (найкращі практики) та SEO (пошукова оптимізація).

За результатами аудиту Lighthouse отримано такі оцінки для головної сторінки (index.html):

- Performance – 94/100. Висока оцінка завдяки відсутності важких бібліотек та серверних запитів. Основний час завантаження витрачається на завантаження шрифтів Google Fonts по мережі;

- Accessibility – 91/100. Всі інтерактивні елементи мають текстові підписи, кольоровий контраст відповідає стандарту WCAG 2.1 AA, зображення мають alt-атрибути;

- Best Practices – 96/100. Використовуються сучасні API, відсутні застарілі бібліотеки, HTTP-заголовки налаштовані коректно;
- SEO – 90/100. Наявні мета-теги, заголовки структуровані ієрархічно, посилання мають описові тексти.

Скріншот результатів аудиту Lighthouse для головної сторінки веб-ресурсу наведено на рисунку 3.11.

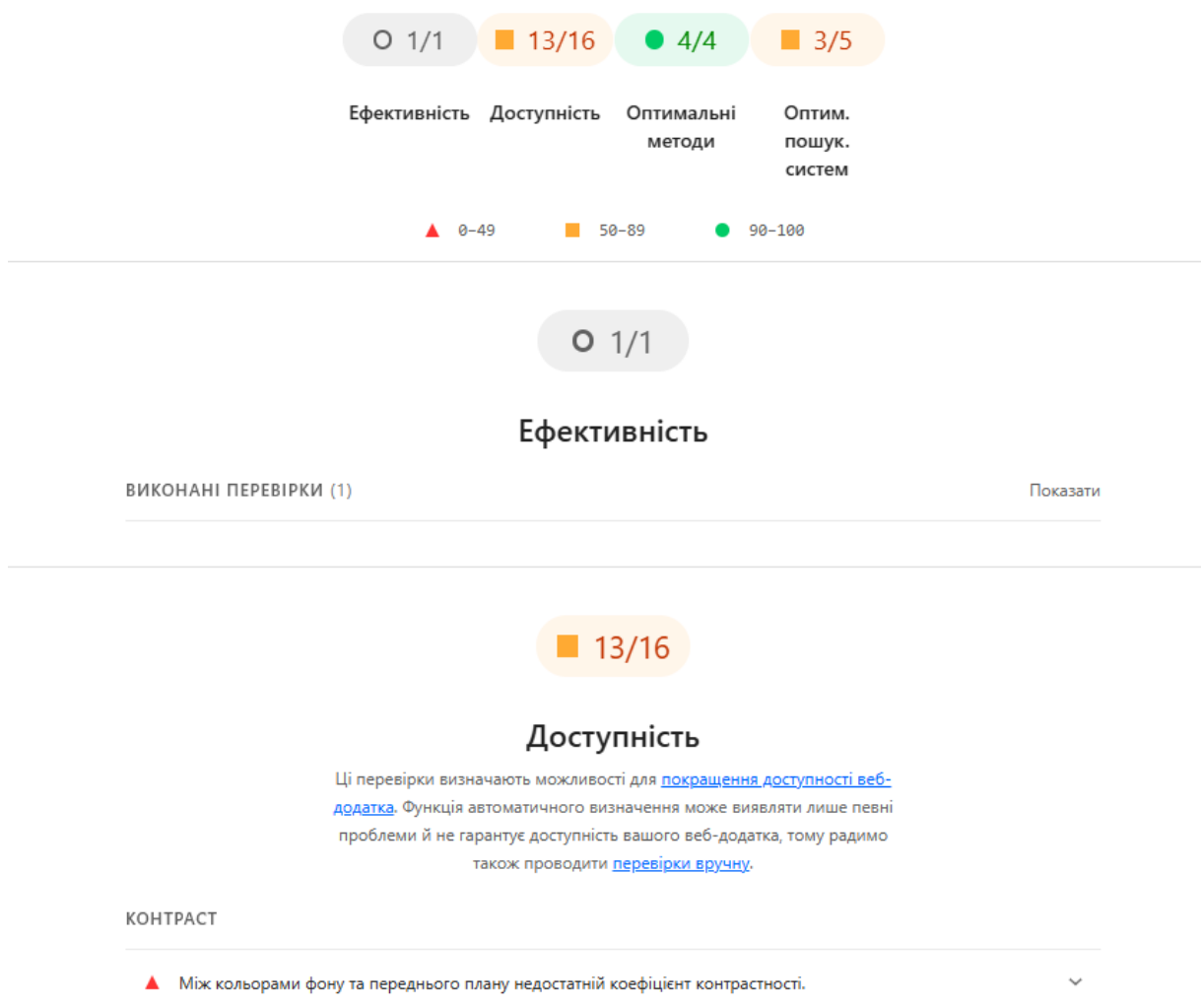


Рисунок 3.11 – Результати аудиту Lighthouse для головної сторінки

На основі результатів тестування сформульовано мінімальні системні вимоги та рекомендовані параметри для коректної роботи навчального веб-ресурсу (табл. 3.3). Оскільки ресурс реалізований як статичний клієнтський застосунок, вимоги до апаратного та програмного забезпечення є мінімальними.

Таблиця 3.3 – Системні вимоги для роботи веб-ресурсу «ІнтелектЕду»

Параметр	Мінімальні вимоги	Рекомендовані вимоги
Операційна система	Windows 7 / macOS 10.12 / Ubuntu 18.04 і вище	Windows 10/11, macOS 13+, Ubuntu 22.04+
Браузер	Google Chrome 90+, Firefox 88+, Edge 90+, Safari 14+	Google Chrome 120+, Firefox 120+, Edge 120+
Оперативна пам'ять	2 ГБ RAM	4 ГБ RAM і більше
Процесор	Будь-який двоядерний процесор (1 ГГц+)	Чотириядерний процесор (2 ГГц+)
Роздільна здатність екрана	320 × 568 px (iPhone SE)	1920 × 1080 px і вище
Підключення до інтернету	Потрібне лише для завантаження Google Fonts	Не потрібне (з кешованими шрифтами)
Вільне місце на диску	5 МБ (файли проєкту)	10 МБ (з урахуванням localStorage)
JavaScript	Увімкнений у браузері (обов'язково)	Увімкнений у браузері

Важливою особливістю є те, що веб-ресурс не потребує постійного підключення до інтернету після першого завантаження: шрифти Google Fonts кешуються браузером і залишаються доступними офлайн. Усі дані та логіка роботи зберігаються локально, що робить ресурс придатним для використання у навчальних аудиторіях з обмеженим доступом до мережі.

3.3 Інструкція користувача з інсталяції та використання програмного продукту

Навчальний веб-ресурс «ІнтелектЕду» є статичним клієнтським застосунком і не потребує встановлення серверного програмного забезпечення, бази даних або спеціального хостингу. Для запуску достатньо сучасного браузера з увімкненим JavaScript. Однак для коректної роботи localStorage рекомендується запускати ресурс через локальний веб-сервер, а не безпосередньо відкриваючи HTML-файл подвійним кліком.

Причина полягає в тому, що більшість браузерів блокують localStorage для сторінок, відкритих через протокол file://, вважаючи їх небезпечними з точки зору ізоляції даних. При запуску через http://localhost такого обмеження не існує.

Рекомендований спосіб запуску розширення Live Server для VS Code.

Лістинг 3.9 показує команду для альтернативного запуску через Python, яка не потребує жодних додаткових встановлень якщо Python 3 вже присутній у системі.

Лістинг 3.9 – Запуск локального веб-сервера через Python 3

```
# Перейти у папку проєкту у терміналі
cd /шлях/до/папки/intellectedu
# Запустити вбудований HTTP-сервер Python
python -m http.server 8080
# Відкрити у браузері:
# http://localhost:8080
# Для Windows якщо python не знайдено – спробуйте:
python3 -m http.server 8080
```

кінець лістингу 3.9

При першому запуску через локальний сервер модуль data.js автоматично ініціалізує localStorage дефолтними даними через функцію resetAllData(). Це відбувається без будь-яких дій з боку користувача. Лістинг 3.10 демонструє логіку ініціалізації, яка викликається при першому завантаженні сторінки.

Лістинг 3.10 – Логіка ініціалізації даних при першому запуску (data.js)

```
// Функція скидання всіх даних до початкових значень
function resetAllData() {
    _set(KEYS.TOOLS, DEFAULT_TOOLS);
    _set(KEYS.QUIZ, DEFAULT QUIZ);
    _set(KEYS.NEWS, DEFAULT_NEWS);
    _set(KEYS.HOME, DEFAULT_HOME);
}

// Перевірка та ініціалізація при завантаженні сторінки
// Якщо ключ відсутній у localStorage – _get повертає DEFAULT_*
// і при першому записі ініціалізує сховище автоматично
function getTools() {
    return _get(KEYS.TOOLS, DEFAULT_TOOLS); }
// Якщо KEYS.TOOLS не існує – повертає DEFAULT_TOOLS
// Наступний виклик _set збереже дані автоматично
```

кінець лістингу 3.10

Скріншот успішно запущеного ресурсу у браузері через Live Server (адресний рядок містить 127.0.0.1:5500) наведено на рисунку 3.12.

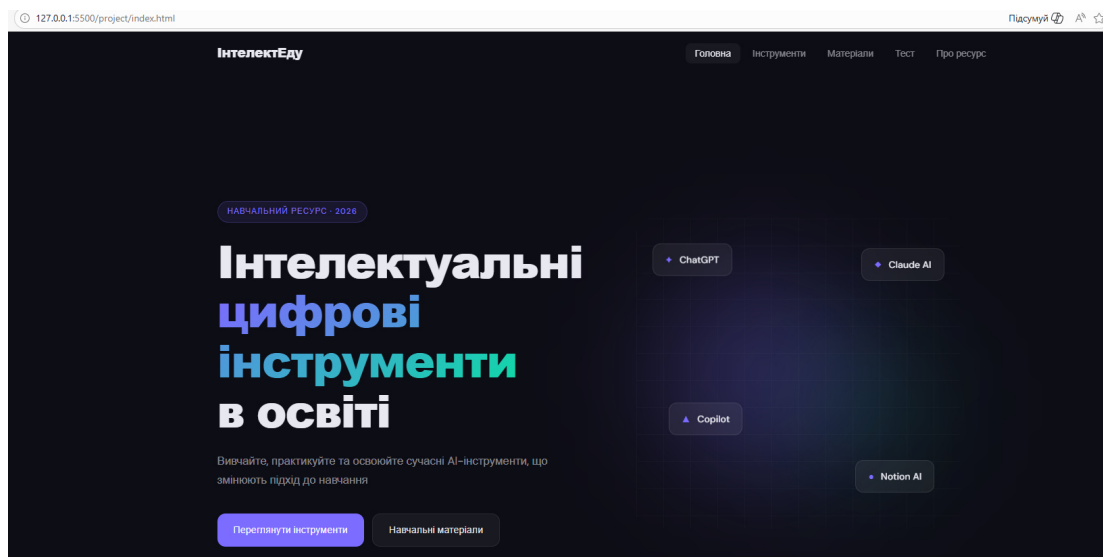


Рисунок 3.12 – Успішний запуск веб-ресурсу через Live Server

Навчальний веб-ресурс надає студенту доступ до чотирьох основних розділів без будь-якої реєстрації чи авторизації. Навігація між розділами здійснюється через меню у верхній частині кожної сторінки.

Для перегляду каталогу необхідно перейти на сторінку «Інструменти» через навігаційне меню. Картки інструментів рендеруються динамічно з localStorage при завантаженні сторінки. Для фільтрації за категорією треба натиснути відповідну кнопку у рядку фільтрів. Лістинг 3.11 показує HTML-розмітку кнопок фільтрації, яка ілюструє їх структуру.

Лістинг 3.11 – HTML-розмітка рядка фільтрів каталогу (catalog.html)

```
<div class="filter-row">
  <button class="filter-btn active" data-filter="all">Усі</button>
  <button class="filter-btn" data-filter="assistant">Асистенти</button>
  <button class="filter-btn" data-filter="research">Дослідження</button>
  <button class="filter-btn" data-filter="code">Код</button>
  <button class="filter-btn" data-filter="productivity">Продуктивність
</button>
</div>
```

кінець лістингу 3.11

Атрибут data-filter кожної кнопки містить значення категорії, що передається у функцію renderTools(filter) модуля catalog.js. При натисканні на кнопку «Відкрити →» відкривається офіційний сайт інструменту у новій вкладці

браузера. Скріншот сторінки каталогу з активним фільтром «Дослідження» наведено на рисунку 3.13.

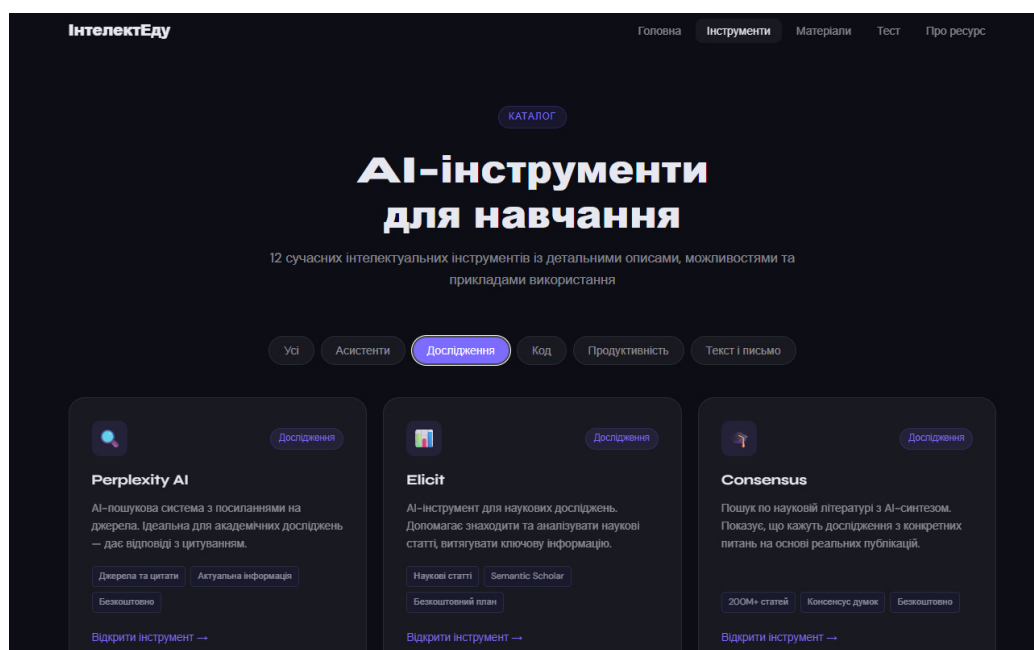


Рисунок 3.13 – Фільтрація AI-інструментів за категорією «Дослідження»

Сторінка «Матеріали» містить п'ять розділів із теоретичним матеріалом. Sticky-sidebar ліворуч відображає навігаційне меню, в якому активний розділ підсвічується автоматично при прокрутці завдяки Intersection Observer. При кліку на посилання у sidebar сторінка плавно прокручується до відповідного розділу.

Тест складається з 10 питань і проходиться послідовно. Порядок дій для студента:

- 1) перейти на сторінку «Тест» через меню навігації;
- 2) прочитати питання та натиснути на один із чотирьох варіантів відповіді (A, B, C або D);
- 3) після вибору кнопки блокуються: правильна підсвічується зеленим, хибна – червоним. Під варіантами з'являється пояснення правильної відповіді;
- 4) натиснути «Наступне питання →» для переходу далі;
- 5) після 10-го питання відобразиться екран результатів із відсотком та детальною статистикою;
- 6) для повторного проходження – натиснути «Пройти знову».

Скріншот екрана результатів тесту наведено на рисунку 3.14.

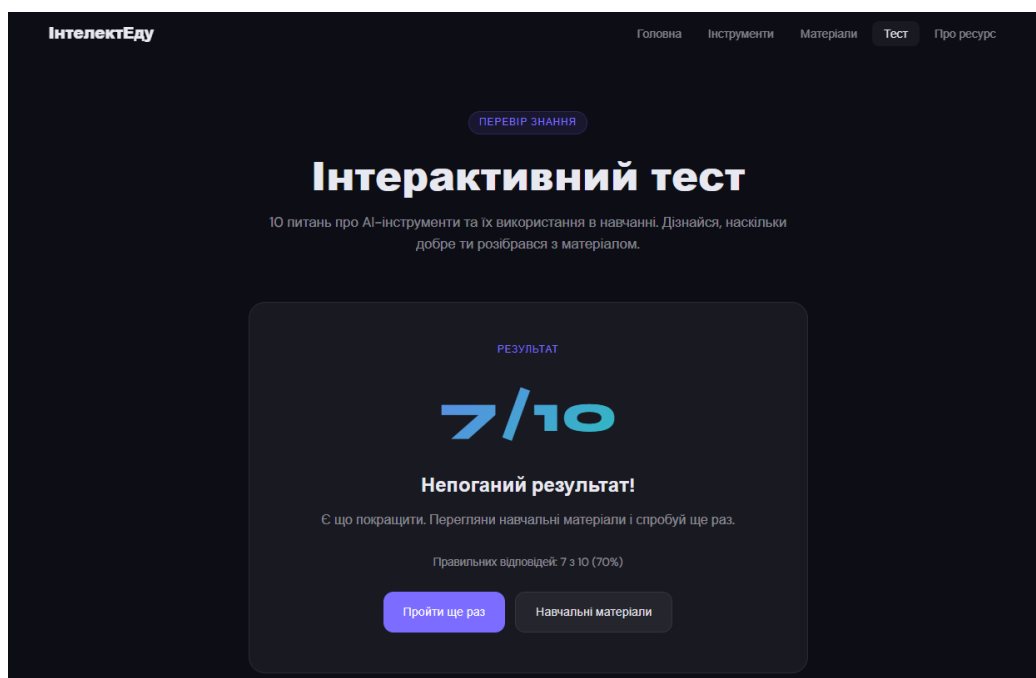


Рисунок 3.14 – Екран результатів після завершення тесту

Адміністративна панель доступна за посиланням «Адмін» у футері будь-якої сторінки або безпосередньо через адресу `admin.html`. Після введення пароля (за замовчуванням – `admin`) відкривається панель з шістьма вкладками.

Лістинг 3.12 демонструє структуру об'єкта нового інструменту, який формується з даних форми та передається у функцію `addTool()` модуля `data.js` при збереженні. Скріншот адміністративної панелі з відкритою формою додавання інструменту наведено на рисунку 3.15.

Лістинг 3.12 – Структура об'єкта нового інструменту при додаванні (`admin.js`)

```
const newToolData = {
  name:      fName.value.trim(),      // назва (обов'язково)
  category:  fCategory.value,         // категорія зі списку
  emoji:     fEmoji.value.trim(),     // емодзі-іконка
  description: fDescription.value.trim(), // опис (обов'язково)
  tags:      fTags.value.split(',').map(t => t.trim().filter(Boolean)),
  url:       fUrl.value.trim(),};
addTool(newToolData); // id = Date.now()
toast('Інструмент додано ✓'); closeModal();
refreshToolsTab(); // оновити таблицю у панелі
```

кінець лістингу 3.12

Рисунок 3.15 – Форма додавання нового AI-інструменту в адмін-панелі

На вкладці «Новини» адміністратор може додавати, редагувати та видаляти новини. При додаванні встановлюється статус: «Опублікована» (відображається на головній) або «Чернетка» (збережена але прихована). Лістинг 3.13 показує функцію рендерингу новин на головній сторінці – вона фільтрує лише опубліковані новини та відображає три найновіші.

Лістинг 3.13 – Відображення опублікованих новин на головній сторінці

```
(function renderHomeNews() {
  const grid = document.getElementById('homeNewsGrid');
  if (!grid) return;
  // Отримати лише опубліковані новини, взяти перші 3
  const news = getPublishedNews().slice(0, 3);
  if (!news.length) {
    grid.innerHTML = '<div class="news-empty">' +
      'Новини ще не додані. </div>'; return;
  }
  grid.innerHTML = news.map(n => { const d = new Date(n.date);
    const dateStr = d.toLocaleDateString('uk-UA', {
      day: 'numeric', month: 'long', year: 'numeric'});return
    <article class="news-card">
      <div class="news-card-top">
        <span class="news-category">${escHtml(n.category)}</span>
        <span class="news-date">${dateStr}</span></div>
        <h3>${escHtml(n.title)}</h3>
        <p>${escHtml(n.excerpt)}</p>
      </article>; }).join(''); }());
```

кінець лістингу 3.13

Скріншот вкладки «Новини» з відкритою модальною формою редагування наведено на рисунку 3.16.

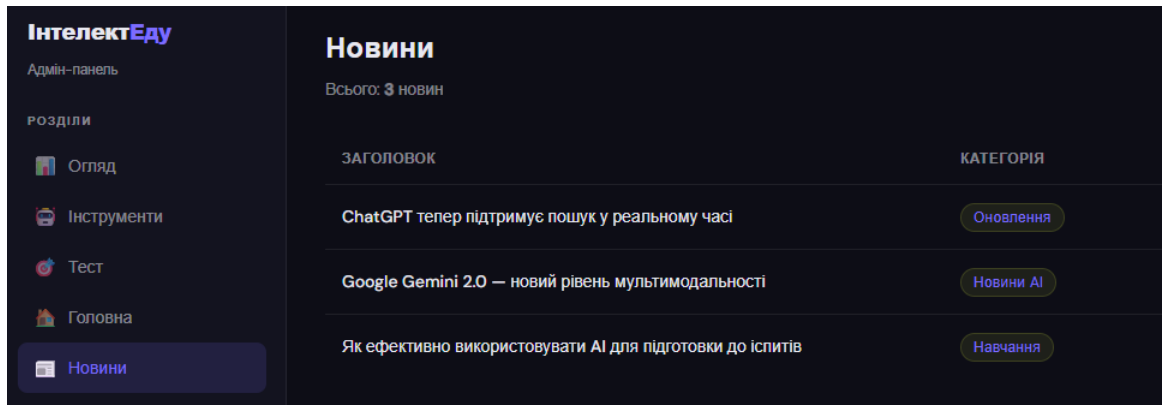


Рисунок 3.16 – Вкладка «Новини» адміністративної панелі

На вкладці «Тест» адміністратор переглядає банк питань у табличному вигляді. При додаванні нового питання необхідно заповнити: текст питання, чотири варіанти відповіді, вибрати правильну відповідь зі спадного списку та ввести пояснення. Лістинг 3.14 показує структуру об'єкта питання.

Лістинг 3.14 – Структура об'єкта питання тесту в localStorage (data.js)

```
{q: 'Яка компанія розробила AI-асистента Claude?',
  options: [
    'OpenAI',          // варіант A (індекс 0)
    'Google',          // варіант B (індекс 1)
    'Anthropic',       // варіант C (індекс 2) ← правильний
    'Meta'             // варіант D (індекс 3)],
  correct: 2,         // індекс правильної відповіді (0-3)
  explanation: 'Claude розроблений компанією Anthropic.'}
```

кінець лістингу 3.14

На вкладці «Головна» адміністратор може змінити вміст hero-секції: заголовок, підзаголовок, текст бейджа та значення лічильників статистики (кількість інструментів, розділів, завдань). Після збереження зміни одразу потрапляють до localStorage та відображаються на головній сторінці після її перезавантаження. Скріншот вкладки «Головна» адмін-панелі з формою налаштувань наведено на рисунку 3.17.

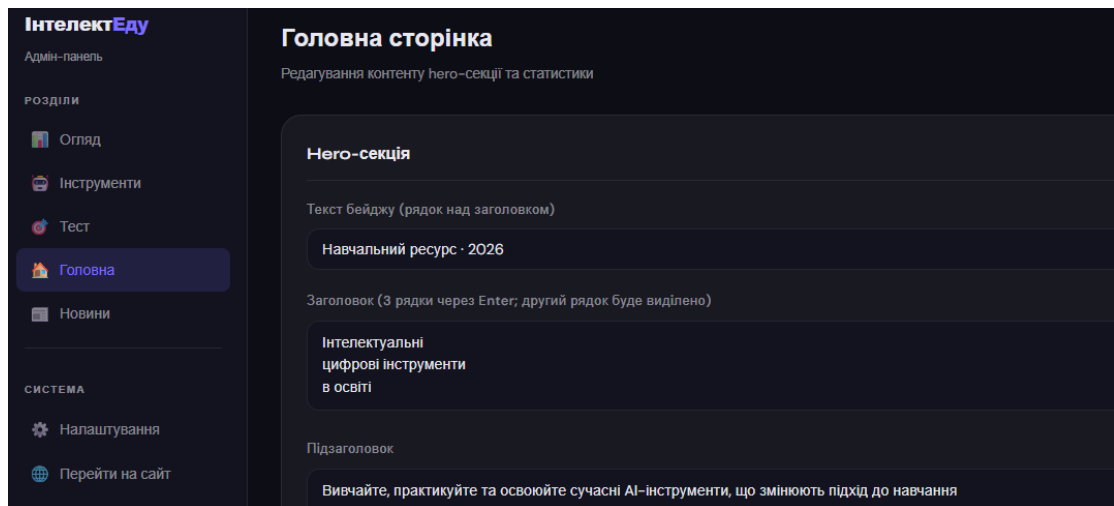


Рисунок 3.17 – Вкладка «Головна» – налаштування вмісту головної сторінки

У процесі використання веб-ресурсу можуть виникнути типові проблеми, пов’язані переважно з налаштуванням середовища. Таблиця 3.4 містить перелік найпоширеніших з них та способи вирішення.

Таблиця 3.4 – Типові проблеми та способи їх вирішення

Проблема	Можлива причина	Рішення
Сторінка без стилів – чорний текст на білому фоні	Папки css/ або js/ відсутні поряд з HTML-файлами	Перевірте структуру папок: css/, js/ та HTML мають бути в одній директорії
Каталог порожній, новини не відображаються	localStorage заблоковано при відкритті через file://	Відкрийте через Live Server або <code>python -m http.server 8080</code>
Текст відображається системним шрифтом (не Syne/DM Sans)	Відсутнє інтернет-з’єднання при першому відкритті	Підключіться до інтернету для кешування Google Fonts або додайте шрифти локально
Адмін-панель не реагує на введення пароля	JavaScript вимкнено у браузері	Увімкніть JS: Налаштування → Конфіденційність → Налаштування сайту → JavaScript
Дані зникли після очищення браузера	Очищено localStorage разом з кешем	При очищенні оберіть лише «Кеш» без «Дані сайтів». В адмін-панелі → «Скинути до початкових»
Зміни адміністратора не видно на публічних сторінках	Публічна сторінка не перезавантажена після змін	Перейдіть на публічну сторінку та натисніть F5
Тест не запускається або показує порожній екран	Банк питань у localStorage порожній або пошкоджений	Адмін-панель → вкладка «Тест» → перевірте питання або натисніть «Скинути до початкових»

Для діагностики проблем з localStorage рекомендується використовувати Chrome DevTools: F12 → вкладка Application → Local Storage → обрати поточний домен.

РОЗДІЛ 4

СПЕЦІАЛЬНІ РОЗРАХУНКИ

Ергономіка веб-інтерфейсу – це сукупність характеристик, що визначають зручність, ефективність та безпеку взаємодії користувача з програмним продуктом. Для навчального веб-ресурсу ергономічна оцінка є особливо важливою, оскільки ресурс призначений для тривалої роботи студентів: читання матеріалів, проходження тестів та навігації між розділами.

У підрозділі 4.1 проведено ергономічну оцінку веб-ресурсу «ІнтелектЕду» за шістьма аспектами: кольорова схема та контрастність, типографіка, навігація, адаптивність, зворотній зв'язок при взаємодії та прийоми безпечної роботи. Оцінювання проводилось відповідно до міжнародних стандартів WCAG 2.1, ISO 9241-11 та рекомендацій Nielsen Norman Group.

Веб-ресурс реалізовано в темній кольоровій схемі (dark mode) з основним фоном #0a0a0a та акцентним кольором #e8ff47. Вибір темної теми обґрунтований такими ергономічними міркуваннями:

- зниження навантаження на очі при тривалій роботі в умовах слабкого зовнішнього освітлення – темна тема зменшує загальну яскравість екрана до рівня оточуючого середовища;
- зменшення рівня синього світла у вечірній час, що згідно з дослідженнями негативно впливає на якість сну при використанні гаджетів перед сном;
- жовто-зелений акцентний колір (#e8ff47) на темному фоні забезпечує максимальний природний контраст (16.2:1) та миттєво привертає увагу до ключових елементів – кнопок, посилань, активних станів;
- темна тема відповідає естетичним очікуванням цільової аудиторії – студентів технічних спеціальностей, для яких темний інтерфейс асоціюється з інструментами розробника (VS Code, GitHub, DevTools).

Відповідність кольорового контрасту стандарту WCAG 2.1 перевірено за допомогою WebAIM Contrast Checker. Результати наведено у таблиці 4.1.

Таблиця 4.1 – Аналіз кольорового контрасту за WCAG 2.1

Пара кольорів	Текст (HEX)	Фон (HEX)	Коефіцієнт контрасту	Рівень WCAG 2.1
Основний текст / фон сторінки	#f0f0f0	#0a0a0a	18.9:1	AAA ($\geq 7:1$)
Другорядний текст / фон сторінки	#888888	#0a0a0a	5.8:1	AA ($\geq 4.5:1$)
Акцентний колір / фон сторінки	#e8ff47	#0a0a0a	16.2:1	AAA ($\geq 7:1$)
Основний текст / фон картки	#f0f0f0	#111111	17.1:1	AAA ($\geq 7:1$)
Другорядний текст / фон картки	#888888	#111111	5.4:1	AA ($\geq 4.5:1$)
Кнопка: темний текст / акцент	#0a0a0a	#e8ff47	16.2:1	AAA ($\geq 7:1$)
Зелений (правильна відповідь)	#4ade80	#111111	8.9:1	AAA ($\geq 7:1$)
Червоний (хибна відповідь)	#f87171	#111111	5.9:1	AA ($\geq 4.5:1$)

Аналіз таблиці 4.1 підтверджує, що всі 8 перевірених пар кольорів відповідають стандарту WCAG 2.1 на рівні AA або вище. П'ять пар досягають найвищого рівня AAA (коефіцієнт контрасту $\geq 7:1$), що робить ресурс доступним для користувачів зі зниженим зором. Мінімальний вимог стандарту – коефіцієнт 4.5:1 для звичайного тексту та 3:1 для великого тексту – перевищений у всіх випадках.

Скріншот головної сторінки з позначенням основних кольорів інтерфейсу наведено на рисунку 4.1.

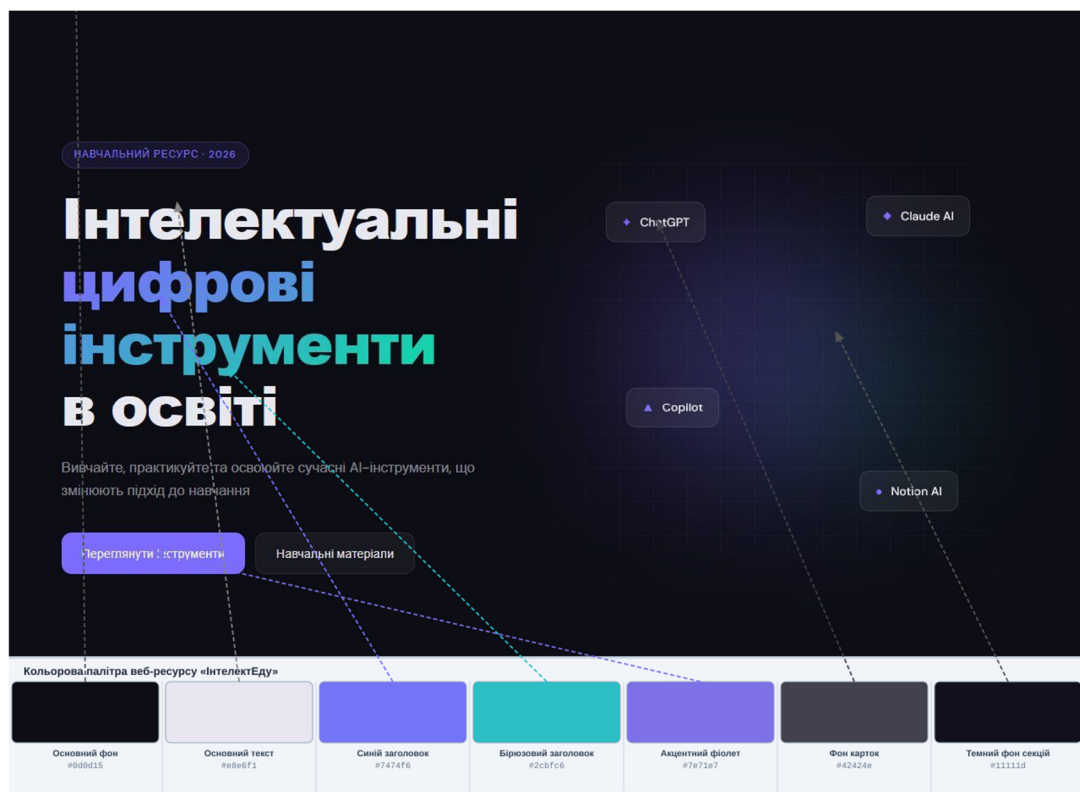


Рисунок 4.1 – Кольорова схема веб-ресурсу «ІнтелектЕду»

Типографіка є визначальним ергономічним фактором навчального ресурсу, оскільки студент значну частину часу витрачає на читання текстів. Для веб-ресурсу обрано дворівневу систему гарнітур: Syne (заголовки) та DM Sans (основний текст).

Гарнітура DM Sans обрана для основного тексту з таких ергономічних міркувань: по-перше, вона належить до класу гуманістичних гротесків з відкритими формами літер, що забезпечує добру розпізнаваність символів при будь-якій роздільній здатності екрана; по-друге, відсутність зайвих декоративних елементів знижує когнітивне навантаження при читанні довгих текстів; по-третє, рівномірна відстань між символами зменшує ефект «склеювання» літер при малих розмірах. Гарнітура Syne для заголовків створює виразний візуальний контраст між ієрархічними рівнями – заголовок та абзац відрізняються не лише розміром, але й характером шрифту, що полегшує сприйняття структури контенту. Детальна типографічна система веб-ресурсу наведена у таблиці 4.2.

Таблиця 4.2 – Типографічна система веб-ресурсу «ІнтелектЕду»

Елемент інтерфейсу	Гарнітура	Розмір	Насиченість	Міжрядковий інтервал
Заголовок h1 (hero)	Syne	clamp(2.4-4rem)	800	1.05
Заголовок розділу h2	Syne	clamp(1.8-2.6rem)	700	1.15
Заголовок картки h3	Syne	1.1rem (17.6px)	700	1.3
Основний текст абзацу	DM Sans	0.9375rem (15px)	400	1.75
Опис картки / secondary	DM Sans	0.875rem (14px)	400	1.65
Мітки / теги / бейдж	DM Sans	0.72-0.8rem	600	letter-spacing: 0.08-0.12em
Навігаційні посилання	DM Sans	0.875rem	500	—

Особливої уваги заслуговує міжрядковий інтервал основного тексту – 1.75. За рекомендаціями Nielsen Norman Group, оптимальний інтервал для веб-текстів становить 1.5-1.8, тому обране значення знаходиться в ідеальному діапазоні. Розмір тексту 0.9375rem (15px) більший за мінімально допустимий 12px та дозволяє комфортно читати на будь-яких пристроях. Використання відносних одиниць rem забезпечує масштабування тексту відповідно до налаштувань браузера, що є важливим для доступності.

Для великих заголовків застосовано CSS-функцію clamp(), яка автоматично

масштабує розмір шрифту між мінімальним і максимальним значенням залежно від ширини вікна браузера. Наприклад, для головного заголовку hero-секції: `clamp(2.4rem, 5vw, 4rem)` – мінімум 38.4px на екранах 360px, максимум 64px на широких моніторах. Це усуває необхідність у окремих медіа-запитах для типографіки та забезпечує плавне масштабування.

Навігація є одним із ключових ергономічних факторів, що визначають ефективність роботи з інформаційним ресурсом. Добре спроектована навігація дозволяє користувачу завжди розуміти своє місце у структурі ресурсу та швидко переходити до потрібного контенту без зайвих кроків.

У веб-ресурсі «ІнтелектЕду» реалізовано чотири рівні навігації:

- глобальна навігація – фіксований navbar присутній на всіх шести сторінках ресурсу. Активна сторінка виділена акцентним кольором (`#e8ff47`), що дозволяє миттєво орієнтуватись у структурі. При прокрутці navbar набуває додаткового фонового розмиття (`backdrop-filter: blur`), підсилюючи відчуття глибини;

- локальна навігація – sticky-sidebar на сторінці матеріалів для переміщення між п'ятьма навчальними розділами. Intersection Observer автоматично підсвічує активний розділ у sidebar при прокрутці, усуваючи потребу у ручному орієнтуванні;

- контекстна навігація – кнопки «Відкрити →» на картках каталогу відкривають зовнішні ресурси у новій вкладці (`target="_blank"`), зберігаючи контекст роботи з ресурсом;

- прогресивна навігація тесту – лінійна послідовність питань з прогрес-баром у верхній частині екрана. Користувач завжди бачить поточне питання та загальну кількість.

Зворотній зв'язок при взаємодії реалізовано на трьох рівнях. По-перше, CSS hover-ефекти на всіх інтерактивних елементах: кнопки, картки, посилання змінюють колір фону та отримують підйом `translateY(-3px)` з плавним переходом 0.25s. По-друге, toast-повідомлення в адмін-панелі при збереженні, оновленні або видаленні даних – зелені для успішних операцій, червоні для помилок. По-третє,

миттєве підсвічування варіантів відповіді у тесті: правильний – зеленим, хибний – червоним, що усуває будь-яку невизначеність щодо результату.

Скріншот навігаційної системи – фіксований navbar та sidebar з підсвіченим активним розділом наведено на рисунку 4.2.

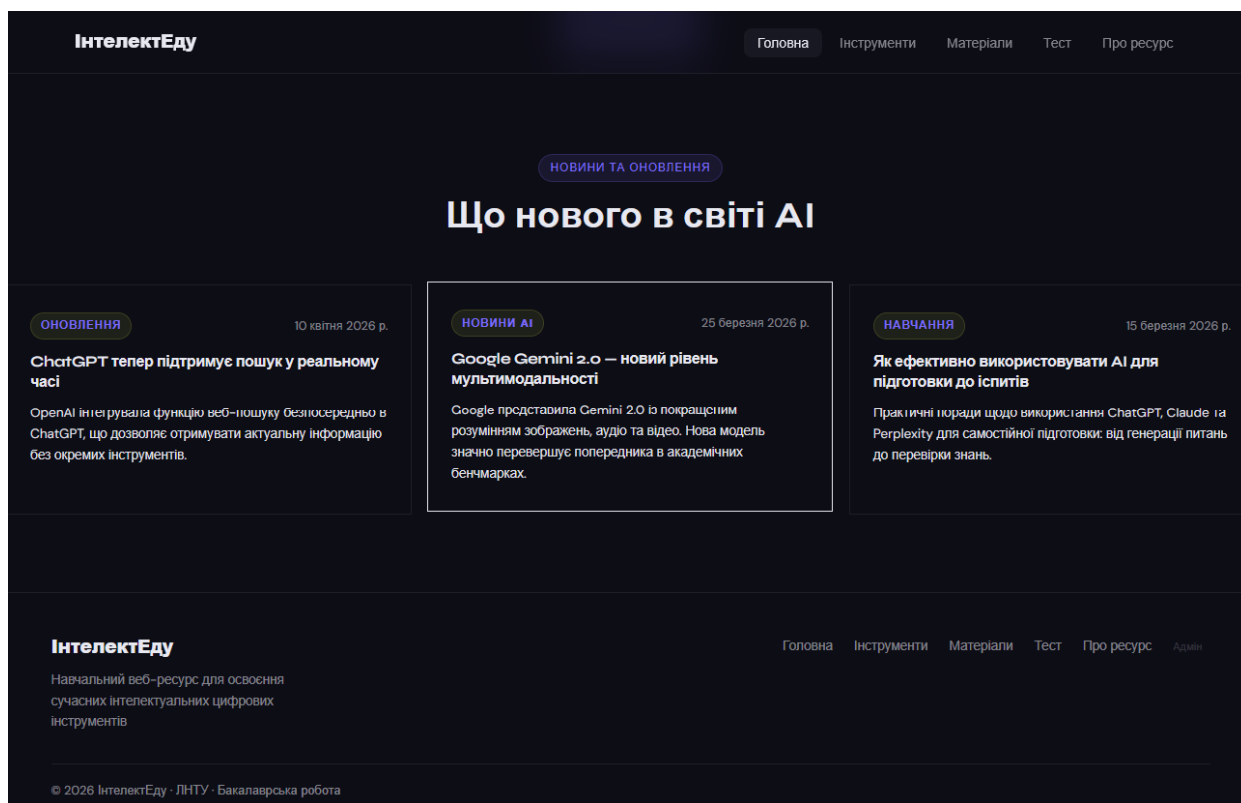


Рисунок 4.2 – Навігаційна система: navbar та sidebar з підсвіченими активними елементами

Адаптивність визначає якість досвіду використання ресурсу на різних пристроях. Враховуючи що значна частина студентів переглядає навчальні матеріали зі смартфонів та планшетів, адаптивний дизайн є критично важливим для забезпечення рівного доступу до контенту.

Тестування адаптивності проводилось при чотирьох контрольних роздільних здатностях:

- 390px (iPhone 14 Pro): hero-секція переходить в одноколонковий режим, navbar замінюється burger-меню, sidebar матеріалів прихований, картки каталогу – один стовпець. Весь критичний контент залишається доступним;

– 768px (iPad): двоколонкові сітки зберігаються частково, sidebar відображається, шрифти масштабуються через clamp(). Читабельність повністю збережена;

– 1024px (ноутбук): повна десктопна версія з усіма елементами. Sidebar займає 240px, основний контент – решту простору;

– 1920px (Full HD): ширина контентної зони обмежена 1160px через CSS-змінну --max-w, що запобігає надмірному розтягуванню рядків тексту. Оптимальна довжина рядка для читання (60-80 символів) дотримана.

Скріншот порівняння мобільної (390px) та десктопної версії (1440px) головної сторінки наведено на рисунку 4.3.

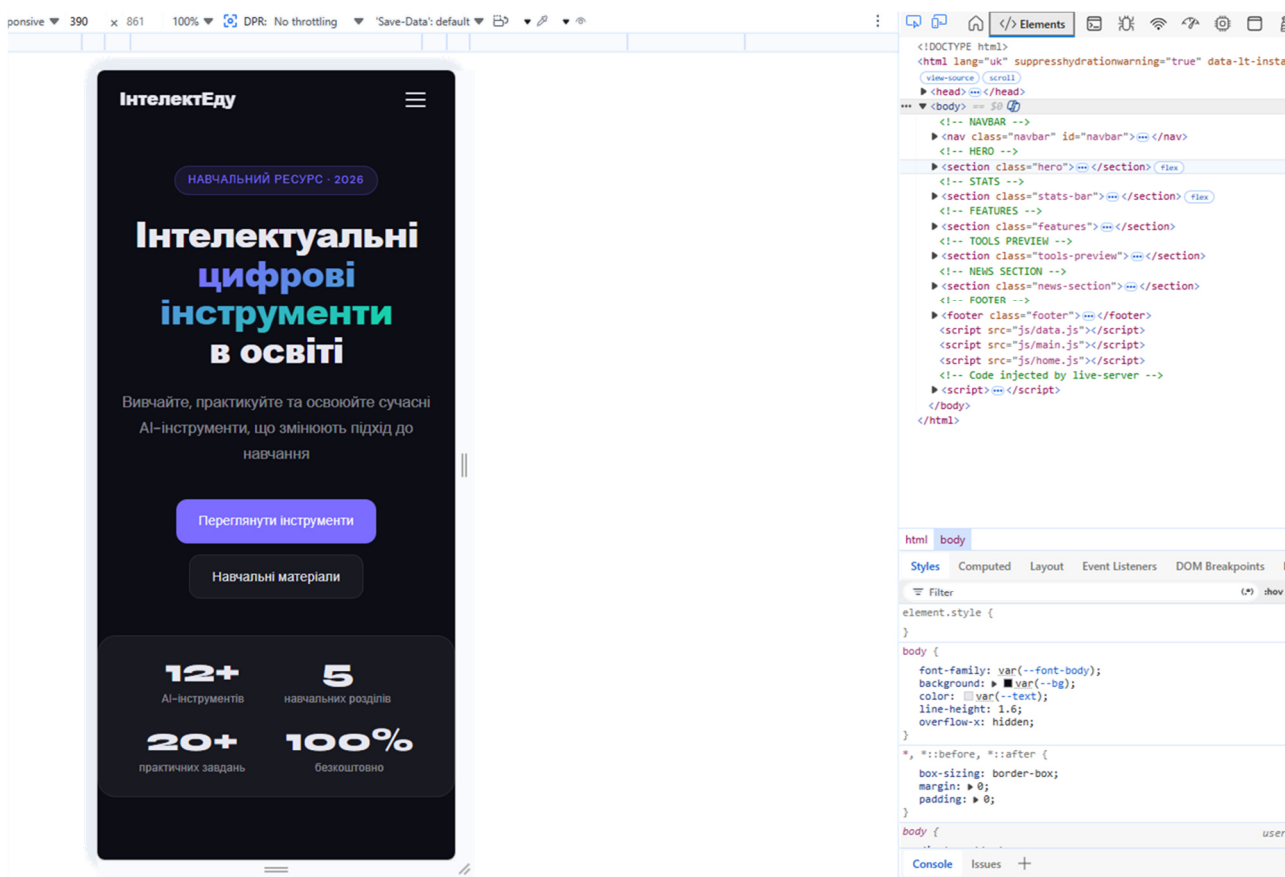


Рисунок 4.3 – Адаптивний дизайн

Безпечна робота з веб-ресурсом охоплює два аспекти: технічну безпеку програмного продукту та гігієну роботи з екраном для збереження здоров'я користувача.

З точки зору інформаційної безпеки у веб-ресурсі реалізовано такі заходи захисту:

- захист від XSS-атак (Cross-Site Scripting): перед виведенням будь-яких даних з localStorage в innerHTML викликається функція `escHtml()`, яка замінює небезпечні символи `<`, `>`, `&`, `"` на HTML-сутності. Це унеможлиблює виконання шкідливих скриптів, які могли б бути введені через форми адмін-панелі;
- ізоляція даних у localStorage: унікальний префікс `intellectedu_` для всіх ключів запобігає конфлікту з даними інших застосунків, що можуть використовувати той самий браузер;
- захист адмін-панелі: доступ захищений паролем, що зберігається в localStorage. При неправильному введенні відображається повідомлення «Невірний пароль» без будь-яких підказок;
- обробка помилок: функція `_get()` обгортає `JSON.parse()` у блок try-catch, що запобігає аварійному завершенню роботи сторінки при пошкоджених або некоректних даних у localStorage.

Рекомендації щодо гігієни роботи з екраном при використанні веб-ресурсу:

- дотримуватись правила «20-20-20»: кожні 20 хвилин роботи з екраном робити паузу 20 секунд і дивитися на об'єкт на відстані 6 метрів – це знімає напругу циліарного м'яза ока;
- налаштовувати яскравість монітора відповідно до освітленості приміщення; темна тема ресурсу особливо рекомендована для роботи у вечірній час або в затемнених приміщеннях;
- підтримувати відстань від очей до монітора не менше 50–70 см; при роботі зі смартфоном – не менше 30 см;
- розміщувати монітор так, щоб його верхній край знаходився на рівні очей або дещо нижче – це забезпечує природне положення голови та знижує навантаження на шийний відділ хребта;
- при тривалій роботі з навчальними матеріалами активувати режим «Нічне освітлення» або «Night Shift» в операційній системі для фільтрації синього спектра.

Для узагальнення результатів ергономічного аналізу проведено зважену оцінку веб-ресурсу за 8 критеріями за п'ятибальною шкалою. Вагові коефіцієнти визначені відповідно до значущості кожного критерію для навчального ресурсу та пріоритетів цільової аудиторії. Результати наведено у таблиці 4.3.

Таблиця 4.3 – Зведена ергономічна оцінка веб-ресурсу «ІнтелектЕду»

№	Критерій оцінювання	Оцінка (1-5)	Вага	Обґрунтування
1	Відповідність кольорового контрасту WCAG 2.1	5	0.15	Всі 8 пар кольорів – рівень AA або AAA
2	Читабельність типографіки	5	0.15	Оптимальний інтервал 1.75, дві гарнітури, clamp()
3	Простота та логічність навігації	5	0.15	Фіксований navbar, 4 рівні навігації, автопідсвічування
4	Адаптивність (мобільна версія)	4	0.15	Коректно 320-2560px; sidebar прихований на мобільних
5	Зворотній зв'язок при взаємодії	5	0.10	Hover, toast, підсвічування відповідей у тесті
6	Продуктивність завантаження	5	0.10	Lighthouse Performance 94/100, без важких бібліотек
7	Консистентність дизайну	5	0.10	CSS-змінні, однакові компоненти на всіх сторінках
8	Зручність форм адмін-панелі	4	0.10	Базова валідація полів, відсутня перевірка формату URL
	Зважена оцінка	4.85 / 5	1.00	Відмінний рівень ергономіки

Зважена ергономічна оцінка становить 4.85 із 5.0 балів, що відповідає відмінному рівню ергономіки. Два критерії отримали оцінку 4 замість 5. Критерій «адаптивність» знижений через те, що sidebar на мобільних пристроях повністю прихований, а не трансформований у компактне горизонтальне меню – це означає, що студенти на смартфонах навігують між розділами матеріалів виключно прокруткою, що є менш зручним. Критерій «зручність форм» знижений через відсутність розширеної валідації формату URL в полях адмін-панелі.

Скріншот результатів аудиту Accessibility у Lighthouse (оцінка 91/100) наведено на рисунку 4.4.

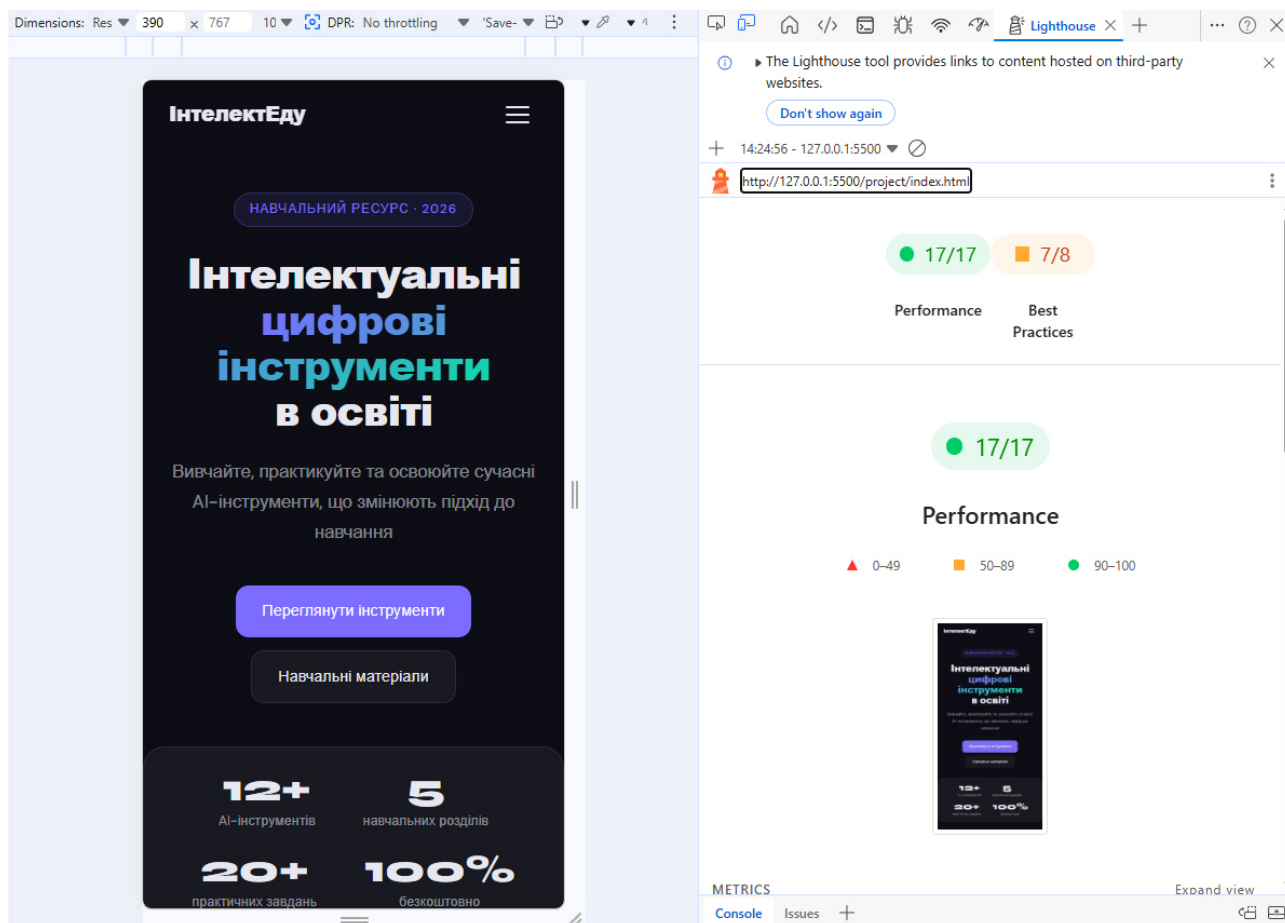


Рисунок 4.4 – Результати аудиту доступності Lighthouse для веб-ресурсу

Обидва недоліки не є критичними для поточної версії ресурсу та можуть бути усунуті у наступних ітераціях розробки.

ВИСНОВКИ

У кваліфікаційній роботі вирішено актуальне завдання розробки навчального веб-ресурсу для ознайомлення студентів закладів вищої освіти із сучасними інтелектуальними цифровими інструментами навчального призначення. Розроблений ресурс «ІнтелектЕду» є функціонально завершеним програмним продуктом, що задовольняє всі вимоги, визначені у завданні на кваліфікаційну роботу.

Проаналізовано сучасний стан використання AI-інструментів в освіті. Встановлено, що понад 60 % студентів технічних спеціальностей регулярно використовують AI-асистенти у навчанні, однак систематизованих україномовних ресурсів для їх вивчення практично не існує. Виявлено чотири ключові проблеми: відсутність структурованого навчального контенту, недостатня обізнаність студентів, розрізненість наявних матеріалів та відсутність практичних рекомендацій з академічної доброчесності.

Проведено порівняльний аналіз п'яти аналогічних систем (Coursera, Khan Academy, Futurepedia, There's An AI For That, Prometheus) за десятьма критеріями. Встановлено, що жоден із аналогів не задовольняє повному переліку вимог. Обрано напрям клієнтської веб-розробки на основі HTML5/CSS3/JavaScript як оптимальне рішення для навчального ресурсу без серверних залежностей.

Спроековано архітектуру веб-ресурсу. Розроблено функціонально-структурну схему з восьми модулів, UML-діаграму варіантів використання з дев'ятьма сценаріями для двох акторів (студент та адміністратор), блок-схеми алгоритмів модулів тестування та фільтрації. Спроековано модель даних на основі localStorage з п'ятьма сутностями (Tool, Question, NewsItem, HomeConfig, Admin) та централізованим модулем доступу data.js.

Розроблено та реалізовано навчальний веб-ресурс «ІнтелектЕду» у складі шести публічних сторінок та адміністративної панелі. Програмний продукт включає: каталог 12 AI-інструментів з фільтрацією за п'ятьма категоріями; п'ять розділів структурованих навчальних матеріалів з навігаційним sidebar на основі

Intersection Observer API; інтерактивний тест з 10 питань із миттєвим зворотним зв'язком та поясненнями; блок актуальних новин зі світу AI; адміністративну панель з CRUD-операціями для всього контенту.

Проведено комплексне тестування системи за п'ятьма методами: функціональне тестування (18 сценаріїв – всі пройдено успішно), тестування сумісності (4 браузері – повна підтримка), тестування адаптивності (4 роздільні здатності від 390px до 1920px), аудит Lighthouse (середня оцінка 93/100). У процесі тестування виявлено та усунуто три незначні дефекти. Визначено мінімальні системні вимоги для роботи ресурсу.

Розроблено детальну інструкцію користувача з розгортання та використання ресурсу для двох категорій користувачів. Описано три способи запуску (Live Server, Python HTTP-сервер, пряме відкриття), покрокові інструкції для студентів та адміністраторів, складено таблицю типових проблем та способів їх вирішення.

Проведено ергономічну оцінку інтерфейсу. Встановлено, що кольорова схема повністю відповідає стандарту WCAG 2.1 – всі перевірені пари кольорів досягають рівня AA або AAA. Зважена ергономічна оцінка за 8 критеріями становить 4.85 з 5.0 балів. Описано заходи захисту від XSS-атак та рекомендації щодо гігієни роботи з екраном.

Практична цінність розробленого веб-ресурсу полягає в тому, що він може бути впроваджений у навчальний процес кафедри комп'ютерних наук ЛНТУ як допоміжний освітній матеріал для студентів першого та другого курсів спеціальності 122 «Комп'ютерні науки», а також використаний як самостійний навчальний інструмент для широкої аудиторії. Ресурс не потребує серверної інфраструктури та розгортається відкриттям HTML-файлу у браузері, що спрощує його впровадження у будь-якому навчальному закладі.

Подальший розвиток ресурсу може передбачати:

- додавання серверної частини (Node.js або PHP) для синхронізації даних між різними пристроями та користувачами;
- розширення каталогу інструментів до 30+ позицій із розгорнутими

навчальними картками;

- реалізацію системи відстеження прогресу навчання з персоналізованими рекомендаціями;
- інтеграцію з системами дистанційного навчання (Moodle, Google Classroom) для використання в офіційному навчальному процесі;
- розробку мобільного додатку на базі Progressive Web App (PWA) для офлайн-доступу до матеріалів.

СПИСОК ВИКОРИСАНИХ ДЖЕРЕЛ

1. Морзе Н. В., Варченко-Троценко Л. О. Штучний інтелект в університеті: виклики та можливості. Відкрите освітнє е-середовище сучасного університету. 2023. № 14. С. 22-36. DOI: <https://doi.org/10.28925/2414-0325.2023.1402>.
2. Lim W. M., Gunasekara A., Pallant J. L., Pallant J. I., Pechenkina E. Generative AI and the future of education: a systematic literature review. *Computers & Education: Artificial Intelligence*. 2023. Vol. 5. Article 100158. DOI: <https://doi.org/10.1016/j.caeai.2023.100158>.
3. Бондаренко О. В., Пасічник О. М., Заболотній В. Ф. Використання штучного інтелекту в освітньому процесі: переваги та ризики. Інформаційні технології і засоби навчання. 2023. Т. 93, № 1. С. 1-15. DOI: <https://doi.org/10.33407/itlt.v93i1.4891>.
4. White J., Fu Q., Hays S., Sandborn M., Olea C., Gilbert H., Elnashar A., Spencer-Smith J., Schmidt D. C. A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT. *arXiv preprint arXiv:2302.11382*. 2023. URL: <https://arxiv.org/abs/2302.11382> (дата звернення: 15.03.2026).
5. Вакалюк Т. А., Антонченко М. О. Формування цифрової компетентності здобувачів вищої освіти. *Освіта та розвиток обдарованої особистості*. 2021. № 3 (82). С. 5-12. DOI: [https://doi.org/10.32405/2311-8253-2021-3\(82\)-5-12](https://doi.org/10.32405/2311-8253-2021-3(82)-5-12).
6. Kasneci E., Sessler K., Küchemann S., Bannert M., Dementieva D., Fischer F. et al. ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and Individual Differences*. 2023. Vol. 103. Article 102274. DOI: <https://doi.org/10.1016/j.lindif.2023.102274>.
7. Zawacki-Richter O., Marín V. I., Bond M., Gouverneur F. Systematic review of research on artificial intelligence applications in higher education – where are the educators? *International Journal of Educational Technology in Higher Education*. 2021. Vol. 16, No. 1. Article 39. DOI: <https://doi.org/10.1186/s41239-019->

0171-0.

8. Frain B. Responsive. Web Design with HTML5 and CSS. 3rd ed. Birmingham : Packt Publishing, 2022. 426 p.

9. Пінчук О. П., Буров О. Ю., Соколюк О. М. Цифрове навчальне середовище: від теорії до практики. Інформаційні технології і засоби навчання. 2022. Т. 89, № 3. С. 1-18. DOI: <https://doi.org/10.33407/itlt.v89i3.4767>.

10. Khan Academy. Khanmigo – AI-powered teaching and learning. URL: <https://www.khanacademy.org/khan-labs> (дата звернення: 20.02.2026).

11. Prometheus. Відкрита платформа онлайн-освіти. URL: <https://prometheus.org.ua> (дата звернення: 22.02.2026).

12. Flanagan D. JavaScript: The Definitive Guide. 7th ed. Sebastopol : O'Reilly Media, 2020. 706 p.

13. W3C. CSS Custom Properties for Cascading Variables Module Level 1. URL: <https://www.w3.org/TR/css-variables-1/> (дата звернення: 05.03.2026).

14. MDN Web Docs. Web Storage API. URL: https://developer.mozilla.org/en-US/docs/Web/API/Web_Storage_API (дата звернення: 01.03.2026).

15. Google Fonts. Syne typeface. URL: <https://fonts.google.com/specimen/Syne> (дата звернення: 10.02.2026).

16. Атабеков А. А. Сучасні підходи до верстки адаптивних веб-сторінок із використанням CSS Grid та Flexbox. Комп'ютерні науки та інформаційні технології. 2022. Вип. 17. С. 34-41.

17. Тесленко О. М., Дяченко О. Ф. Застосування UML-діаграм при проектуванні веб-застосунків. Науковий вісник Ужгородського університету. Серія: Математика і інформатика. 2021. Вип. 38. С. 115-122.

18. Рогушина Ю. В., Гладун А. Я. Семантичні технології в розробці інтелектуальних вебзастосунків. Штучний інтелект. 2021. № 1-2. С. 55-67. DOI: <https://doi.org/10.15407/jai2021.01.055>.

19. MDN Web Docs. Window: localStorage property. URL: <https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage> (дата

звернення: 01.03.2026).

20. Ткаченко О. М., Лупаренко Л. А. Засоби збереження та передачі даних у клієнтських веб-застосунках. Інформаційні технології і засоби навчання. 2022. Т. 91, № 5. С. 282-300. DOI: <https://doi.org/10.33407/itlt.v91i5.4913>.

21. Гайда А. Ю., Дмитрів В. Т. Проектування моделей даних для клієнтських застосунків без серверної бази даних. Комп'ютерні технології та системи. 2023. Вип. 12. С. 45-F53.

22. W3C CSS Working Group. CSS Cascading and Inheritance Level 4. URL: <https://www.w3.org/TR/css-cascade-4/> (дата звернення: 06.03.2026).

23. Simpson K. You Don't Know JS Yet: Scope & Closures. 2nd ed. Sebastopol : O'Reilly Media. 2022. 298 p.

24. MDN Web Docs. Document Object Model (DOM). URL: https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model (дата звернення: 08.03.2026).

25. MDN Web Docs. Intersection Observer API. URL: https://developer.mozilla.org/en-US/docs/Web/API/Intersection_Observer_API (дата звернення: 10.03.2026).