



Наукові перспективи
Видавнича група

№ 8 (49)

2025

НАУКА і ТЕХНІКА

серія: *право*, серія: *економіка*, серія: *педагогіка*,
серія: *техніка*, серія: *фізико-математичні науки*

СЬОГОДНІ



З Україною

в серці!



Видавнича група «Наукові перспективи»

Всеукраїнська Асамблея докторів наук із державного управління

Асоціація науковців України

«Наука і техніка сьогодні»

*(Серія «Педагогіка», Серія «Право», Серія «Економіка»,
Серія «Фізико-математичні науки», Серія «Техніка»)*

Випуск № 8(49) 2025

Київ – 2025

Publishing Group «Scientific Perspectives»

Ukrainian Assembly of Doctors of Sciences in Public Administration

Association of Scientists of Ukraine

"Science and technology today"

***("Pedagogy" series, "Law" series, "Economics" series,
"Physical and mathematical sciences" series, "Technics" series)***

Issue № 8(49) 2025

Kyiv – 2025

**«Наука і техніка сьогодні» (Серія «Педагогіка», Серія «Право», Серія «Економіка»,
Серія «Фізико-математичні науки», Серія «Техніка»): журнал.
2025. № 8(49) 2025. С. 1879**



*Згідно наказу Міністерства освіти і науки України від 07.04.2022 № 320 журналу
присвоєно категорію "Б" із економіки та педагогіки (спеціальності – 015 -
Педагогічні науки; 076 - Економічні науки)*

*Згідно наказу Міністерства освіти і науки України від 06.06.2022 № 530 журналу
присвоєно категорію "Б" із права (спеціальність – 081 Юридичні науки)*

*Згідно наказу Міністерства освіти і науки України від 10.10.2022 № 894 журналу присвоєно
категорію "Б" із техніки (спеціальність - 122 Комп'ютерні науки)*

*Журнал видається за підтримки Міждержавної гільдії інженерів консультантів, Інституту філософії та
соціології Національної Академії Наук Азербайджану (Баку, Азербайджан), громадської організації «Християнська
академія педагогічних наук України» та громадської організації «Всеукраїнська асоціація педагогів і психологів з
духовно-морального виховання»*

*Рекомендовано до видавництва Президією Всеукраїнської Асамблеї докторів наук з державного управління
(Рішення від 25.08.2025, № 8/8-25)*



Журнал включено до міжнародної наукометричної бази Index
Copernicus (IC), міжнародної пошукової системи Google Scholar та до
міжнародної наукометричної бази даних Research Bible



Головний редактор: Сопілко Ірина Миколаївна - доктор юридичних наук,
професор, Відмінник освіти України, Лауреат Премії Президента України для молодих
вчених, Лауреат Премії Верховної Ради України найталановитішим молодим ученим в
галузі фундаментальних і прикладних досліджень та науково-технічних розробок,
академік Академії наук вищої школи України, Заслужений юрист України (Київ, Україна)

Редакційна колегія:

- Бахов Іван Степанович — доктор педагогічних наук, професор, завідувач кафедри іноземної філології та перекладу Міжрегіональної академії управління персоналом (Київ, Україна)
- Будник Вікторія Анатоліївна - кандидат економічних наук, професор, професор кафедри бізнес-логістики та транспортних технологій Державного університету інфраструктури та технологій (Київ, Україна)
- Волк Павло Павлович — доцент кафедри водної інженерії та водних технологій Національного університету водного господарства та природокористування (Рівне, Україна)
- Гирка Ольга Ігорівна - кандидат технічних наук, доцент, доцент кафедри товарознавства, митної справи та управління якістю Львівського торговельно-економічного університету (Львів, Україна)
- Гнатюк Сергій Олександрович - кандидат технічних наук, доцент, заступник декана факультету аеронавігації, електроніки та телекомунікацій Національного авіаційного університету (Київ, Україна)
- Дацій Олександр Іванович - доктор економічних наук, професор, Заслужений працівник освіти України, завідувач кафедри фінансів, банківської та страхової справи Міжрегіональної академії управління персоналом (Київ, Україна)
- Дівізніук Михайло Михайлович - доктор фізико-математичних наук, професор, Завідувач відділу Відділу цивільного захисту та інноваційної діяльності Державної установи "Інститут геохімії навколишнього середовища Національної академії наук України" (Київ, Україна)
- Дяденчук Альона Федорівна - кандидат технічних наук, старший викладач кафедри вищої математики і фізики Таврійського державного агротехнологічного університету імені Дмитра Моторного (Мелітополь, Україна)
- Забулонов Юрій Леонідович - доктор технічних наук, професор, Член-кореспондент НАН України, директор Державної установи «Інститут геохімії навколишнього середовища Національної академії наук України» (Київ, Україна)
- Ільїн Валерій Юрійович - доктор економічних наук, професор (Київ, Україна)
- Ільїна Анастасія Олександрівна - кандидат економічних наук, доцент, доцент кафедри публічного управління і адміністрування Національного торговельно-економічного університету (Київ, Україна)
- Кардаш Оксана Любомирівна — кандидат економічних наук, доцент кафедри комп'ютерних технологій та економічної кібернетики Навчально-наукового інституту автоматизації, кібернетики та обчислювальної техніки Національного університету водного господарства та природокористування (м. Рівне, Україна)

- Гаврилюк М.А.** 1201
СТЕКІНГОВИЙ ПІДХІД ДЛЯ ПОДОЛАННЯ ВИКЛИКІВ МАЛИХ НЕЗБАЛАНСОВАНИХ МЕДИЧНИХ ДАНИХ
- Гірянський Б.П., Булах Б.В.** 1214
СТВОРЕННЯ НЕЙРОННИХ МЕРЕЖ ШЛЯХОМ НЕЙРО-ЕВОЛЮЦІЇ ТА АВТОМАТИЗОВАНОГО НАВЧАННЯ З ВИКОРИСТАННЯМ ЕВОЛЮЦІЙНИХ АЛГОРИТМІВ
- Гончаров М.О., Чепель Д.О., Малахов С.В.** 1228
ОЦІНКА ОБЧИСЛЮВАЛЬНОЇ СКЛАДНОСТІ ЕТАПУ ПОПЕРЕДНЬОЇ ОБРОБКИ ВХІДНИХ ДАНИХ ПРИ РЕАЛІЗАЦІЇ ПРОЦЕДУР СТЕГАНОВСТАВКИ ЗОБРАЖЕНЬ
- Горбачук В.М., Дунаєвський М.С., Ніколенко Д.І.** 1246
ПРИНЦИПИ ЦІНОУТВОРЕННЯ НА РИНКАХ ЕЛЕКТРОЕНЕРГІЇ
- Горячев Г.В., Громик А.П., Ткачук Р.О.** 1262
АДАПТИВНЕ УПРАВЛІННЯ ЕНЕРГОСПОЖИВАННЯМ РОЗУМНОГО БУДИНКУ НА ОСНОВІ НЕЙРОННИХ МЕРЕЖ
- Гуменний П.В., Федорчук А.Л., Лось М.В.** 1281
ЕВОЛЮЦІЯ ОПЕРАЦІЙНИХ СИСТЕМ: ВІД ПАКЕТНОЇ ОБРОБКИ ДАНИХ ДО ВІРТУАЛІЗОВАНИХ СЕРЕДОВИЩ
- Данилюк І.М.** 1297
ЯДРО АВТОМАТИЗОВАНОЇ СИСТЕМИ ТЕСТУВАННЯ КОДУ З ГЕНЕРАЦІЄЮ ТЕСТІВ ЗА ДОПОМОГОЮ ШТУЧНОГО ІНТЕЛЕКТУ
- Денисюк В.О., Сторожук І.І.** 1311
ДОСЛІДЖЕННЯ ТА РОЗРОБКА WEB-РЕСУРСУ ПЕРСОНАЛЬНОГО КРЕАТИВНОГО ПРОСТОРУ З УНІКАЛЬНИМ UI-ДИЗАЙНОМ
- Дербенцев В.Д., Лозовик Ю.М., Ляскович О.В., Рзаєв Д.О., Даценко Н.В.** 1332
СИСТЕМНА МЕТОДОЛОГІЯ ІНТЕГРАЦІЇ МУЛЬТИМОДАЛЬНИХ ДАНИХ ТА ІНТЕЛЕКТУАЛЬНИХ АГЕНТІВ ДЛЯ ПРОГНОЗУВАННЯ ЦІН ФІНАНСОВИХ АКТИВІВ

УДК 004.8

[https://doi.org/10.52058/2786-6025-2025-8\(49\)-1311-1331](https://doi.org/10.52058/2786-6025-2025-8(49)-1311-1331)

Денисюк Валерій Олександрович кандидат технічних наук, доцент, доцент кафедри комп'ютерних наук, Вінницький національний технічний університет, м. Вінниця, <https://orcid.org/0000-0003-1057-3518>

Сторожук Іоанна Ігорівна студентка кафедри комп'ютерних наук, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м. Вінниця

ДОСЛІДЖЕННЯ ТА РОЗРОБКА WEB-РЕСУРСУ ПЕРСОНАЛЬНОГО КРЕАТИВНОГО ПРОСТОРУ З УНІКАЛЬНИМ UI-ДИЗАЙНОМ

Анотація. Досліджено процес розробки можливого варіанту персонального WEB-ресурсу для творчих особистостей. Головною особливістю є унікальний UI-дизайн із інтерактивними елементами, анімацією та мультимедійним контентом. Проаналізовано сучасні платформи для митців, методи UI/UX-дизайну та технології реалізації веб-додатків.

Запропонований варіант веб-ресурсу дозволяє митцю поєднувати музику, відео, текстові матеріали в єдиному інтерактивному середовищі, що є альтернативою стандартним платформам. Вибір технологій може коригуватися відповідно до технічних вимог і можливостей.

Проаналізовано сучасні платформи для творчої самопрезентації (Behance, Webflow, Tumblr); сформульовано постановку задачі та функціональні вимоги; спроектовано структуру WEB-ресурсу та інтерфейс користувача; розроблено клієнтську частину з HTML, CSS і JS з унікальним адаптивним дизайном; створено серверну частину на Node.js і Express, реалізовано API та обробку медіа; інтегровано мультимедійний контент (аудіо, зображення, відео з YouTube) і базову авторизацію; проведено ручне тестування функцій: додавання, редагування, валідація, адаптивність, взаємодія.

Дослідження та розробка дозволили розширити функціональні можливостей персонального WEB-ресурсу для творчої самопрезентації через розробку адаптивного середовища з підтримкою мультимедійного контенту, авторизації та індивідуального UI-дизайну. Зазначене досягнуто шляхом реалізації повноцінної клієнт-серверної архітектури, ручного проектування інтерфейсу та підтримки мультимедіа.

Підтвердженням є успішне проходження ручного тестування основних функцій. У порівнянні з готовими конструкторами сайтів, створений ресурс забезпечує повну гнучкість у дизайні, логіці взаємодії та можливість масштабування під конкретні задачі користувача. Ресурс може бути використаний як персональний арт-блог, цифрове портфоліо або як основа для масштабованої онлайн-платформи. Він є доцільним для митців, дизайнерів, контент-креаторів, а також може бути адаптований під освітні, презентаційні або експериментальні проекти.

Ключові слова: WEB-ресурс, UI-дизайн, HTML, CSS, JavaScript, Node.js, мультимедіа.

Denisyuk Valeriy Oleksandrovykh Candidate of Technical Sciences, Associate Professor, Associate Professor of the Department of Computer Science, Vinnytsia National Technical University, Vinnytsia, <https://orcid.org/0000-0003-1057-3518>

Storozhuk Ioanna Ihorivna Student of the Department of Computer Science, Faculty of Intellectual Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia

RESEARCH AND DEVELOPMENT OF A WEB RESOURCE OF A PERSONAL CREATIVE SPACE WITH A UNIQUE UI DESIGN

Abstract. The process of developing a possible version of a personal WEB resource for creative individuals has been studied. The main feature is a unique UI design with interactive elements, animation and multimedia content. Modern platforms for artists, UI/UX design methods and web application implementation technologies have been analyzed. The proposed version of the web resource allows the artist to combine music, video, text materials in a single interactive environment, which is an alternative to standard platforms. The choice of technologies can be adjusted in accordance with technical requirements and capabilities. Modern platforms for creative self-presentation (Behance, Webflow, Tumblr) have been analyzed; the task statement and functional requirements have been formulated; the structure of the WEB resource and the user interface have been designed; the client part has been developed with HTML, CSS and JS with a unique adaptive design; the server part has been created on Node.js and Express, API and media processing have been implemented; multimedia content (audio, images, videos from YouTube) and basic authorization have been integrated; manual testing of functions was carried out: adding, editing, validation, adaptability, interaction. Research and development

allowed to expand the functionality of a personal WEB resource for creative self-presentation through the development of an adaptive environment with support for multimedia content, authorization and individual UI design. This was achieved by implementing a full-fledged client-server architecture, manual interface design and multimedia support.

The confirmation is the successful completion of manual testing of the main functions. Compared to ready-made site designers, the created resource provides complete flexibility in design, interaction logic and the ability to scale to specific user tasks. The resource can be used as a personal art blog, digital portfolio or as the basis for a scalable online platform. It is appropriate for artists, designers, content creators, and can also be adapted for educational, presentation or experimental projects.

Keywords: WEB resource, UI design, HTML, CSS, JavaScript, Node.js, multimedia.

Постановка проблеми. У цифрову епоху митці прагнуть мати власні WEB-ресурси для презентації творчості у персональному стилі — без обмежень форматів, типових для соціальних мереж.

Попри популярність платформ на кшталт YouTube, Instagram, Behance чи Webflow, жодна з них не надає повного контролю над дизайном, структурою та логікою подання контенту.

Огляд менш відомих рішень (Tumblr, Audiotool, Artsteps тощо) підтверджує: хоча кожне має свої переваги, жодне не забезпечує універсальності та гнучкості.

Тому створення адаптивного, мультимедійного WEB-ресурсу «з нуля» є актуальним і доцільним.

Такий ресурс стає не просто черговим акаунтом, а центром творчої активності автора — платформою, що об'єднує музику, відео, тексти, графіку й інтерактив. Він водночас виконує функції портфоліо, каналу комунікації та самовираження митця в інтернеті.

Аналіз останніх досліджень і публікацій. У цифрову добу творчі особистості — митці, блогери, музиканти, розробники — дедалі частіше прагнуть представити свої ідеї у власному мультимедійному веб-просторі. Популярні платформи (YouTube, Instagram, Behance, Notion, SoundCloud) дають змогу публікувати окремі типи контенту, але обмежують гнучкість у дизайні, структурі та поєднанні медіа.

Деякі альтернативи, як-от Tumblr чи Audiotool, мають цікаві функції: Tumblr дозволяє редагування HTML/CSS, але вимагає знань кодування; Audiotool — зручний музичний інструмент, але не підтримує інші формати. Artsteps дає змогу створювати 3D-виставки, однак має складний інтерфейс і вузьку спеціалізацію.

Behance також обмежений шаблонами, а YouTube та Instagram взагалі не дозволяють створення сторінок зі змішаним контентом [1-5]. Крім того, більшість платформ керуються алгоритмами, які диктують, що бачить аудиторія, зводячи творчість до гонитви за переглядами. Автор стає залежним від системи, а не її господарем.

Отже, сучасні інструменти або обмежують типи контенту, або не дають контролю над взаємодією.

Альтернативою стає персональний WEB-ресурс, що поєднує різні формати і підпорядковується авторському баченню - і в дизайні, і в структурі [5].

Мета статті - дослідити та розширити функціональні можливості персонального WEB-ресурсу для творчої самопрезентації через розробку адаптивного середовища з підтримкою мультимедійного контенту, авторизації та індивідуального UI-дизайну (UI, User Interface – користувацький інтерфейс).

Виклад основного матеріалу. WEB-ресурс має функціонувати не як шаблонна сторінка, а як унікальний простір самовираження – незалежний від соціальних мереж, проте взаємодіючі із ними.

Вибір інструментів та архітектури реалізації. Клієнтську частину проекту реалізовано за допомогою HTML5, CSS3 і JavaScript. HTML відповідає за структуру, CSS - за адаптивне оформлення, а JavaScript - за динамічну взаємодію, включно з перемиканням тем, авторизацією тощо. Використання чистих мов без фреймворків забезпечує повний контроль над інтерфейсом і хорошу сумісність [6, 7].

Розробку здійснювали у Visual Studio Code - зручному редакторі з підтримкою синтаксису, Git і запуску локального сервера [8]. Архітектура побудована за класичною клієнт-серверною моделлю (рис.1), що розділяє frontend та backend, дозволяє масштабувати код і окремо тестувати логіку: обробку медіа, чернеток, права доступу тощо.

Серверна частина реалізована на Node.js з використанням Express для побудови RESTful API [9].

Спільна мова програмування на клієнтському й серверному рівнях (JavaScript) спрощує підтримку на обох рівнях (frontend і backend) і зменшує складність проєкту (рис.2).

Для завантаження медіа використовується Multer, а дані зберігаються у форматі JSON (файл posts.json) [10, 11].

Це дає змогу уникнути складності з базами даних і забезпечує зрозумілу структуру.

Логіку обробки запитів винесено у файл postRoutes з методами GET, POST, PUT і DELETE, згідно з принципами розділення відповідальностей.

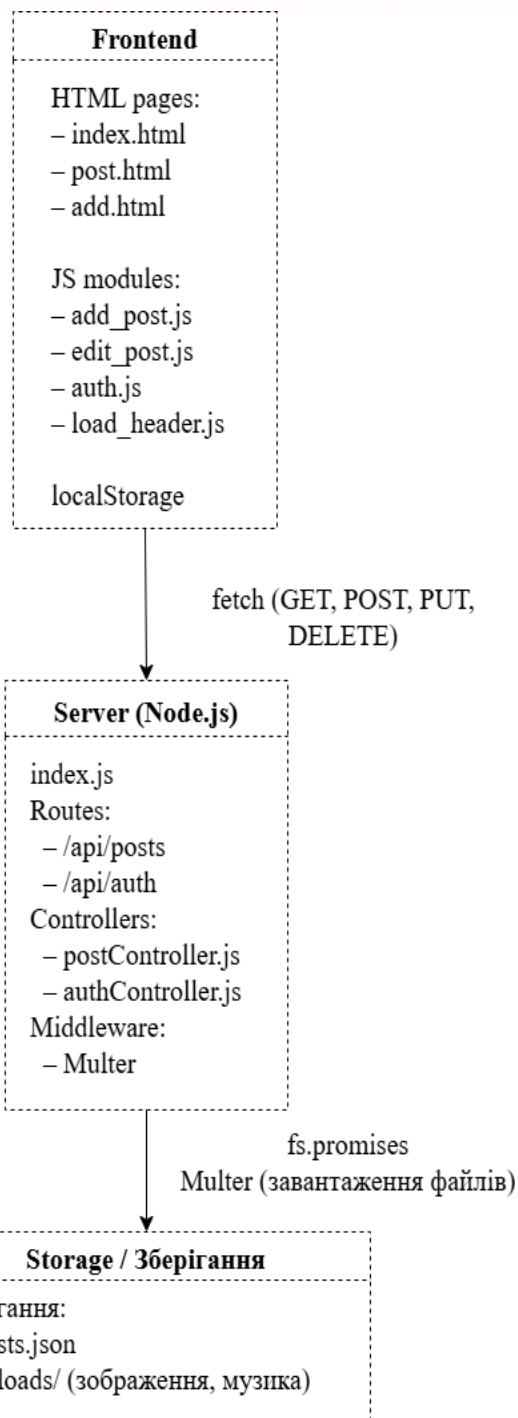


Рис. 1. Архітектура клієнт-серверної взаємодії

Локальне розгортання WEB-ресурсу стало оптимальним рішенням на етапі розробки: це дозволило уникнути залежності від хостингів, спростити налаштування серверного середовища та повністю зосередитись на архітектурі й функціональності. Хоча такий підхід має обмеження у масштабуванні та безпеці, він ідеально підходить для демонстраційних і персональних проєктів.

Клієнтська частина - це повна кастомізація без фреймворків. Кожна сторінка WEB-ресурсу має власний JavaScript-файл: «add_post.js», «edit_post.js», «auth.js», «load_header.js» тощо.

Frontend (HTML-сторінки)	Backend API (Express)
add.html → POST /api/posts	GET /api/posts
edit.html → PUT /api/posts/:id	GET /api/posts/:id
drafts.html → GET /api/posts + фільтрація по draft:true	POST /api/posts
post.html → GET /api/posts/:id	PUT /api/posts/:id
index.html → GET /api/posts (всі пости)	DELETE /api/posts/:id
login.html → auth.js → POST /api/auth	POST /api/auth

Рис. 2. Діаграма frontend/backend

Спільні елементи сайту (header, menu, footer) динамічно підвантажуються JavaScript із папки components/, що забезпечує модульність і зручне оновлення без дублювання. Це ручна альтернатива шаблонним рушіям на кшталт Handlebars чи EJS [12].

Самостійна розробка надає повний контроль над логікою й дизайном, на відміну від платформ типу Wix або WordPress, та сприяє кращому розумінню веб-розробки [13, 14]. Інтерфейс передбачає чотири типи постів - блог, музика, відео, арт - кожен із власною структурою. UI у цьому проєкті є не лише стилізацією, а частиною логіки й емоційного впливу, що відповідає сучасним підходам до UI/UX (UI, User Interface - користувацький інтерфейс; UX, User Experience - користувацький досвід) [4]. Адаптивність частково реалізована за допомогою медіа-запитів, сіток на Flex і Grid, та функції clamp(); однак потребує доопрацювання для мобільних пристроїв — меню, форм і модальних вікон [15]. Проєкт створено без CMS і UI-бібліотек, що дозволило повністю сформулювати авторське бачення структури й взаємодії. Інтерфейс у цьому випадку виступає не лише засобом навігації, а формою самовираження — він є частиною мистецького послання.

Структура WEB-ресурсу, навігація, ролі користувача. Створення авторського WEB-ресурсу передбачає не лише технічну гнучкість, а й логічну організацію сторінок і зручність користування. Користувач має легко орієнтуватися між сторінками, а автор — ефективно керувати публікаціями. Архітектура WEB-ресурсу побудована за принципом модульності: кожна HTML-сторінка відповідає конкретному типу дії або функціональній області (рис.3).

Основними є:

- index.html – головна сторінка зі стрічкою постів;
- post.html – сторінка перегляду окремого поста;
- add.html / edit.html – додавання та редагування постів відповідно;
- drafts.html – список чернеток автора;
- favourite.html – улюблені пости;
- profile.html – сторінка профілю;
- about.html – інформаційна сторінка про WEB-ресурс;
- login.html – форма входу та реєстрації.

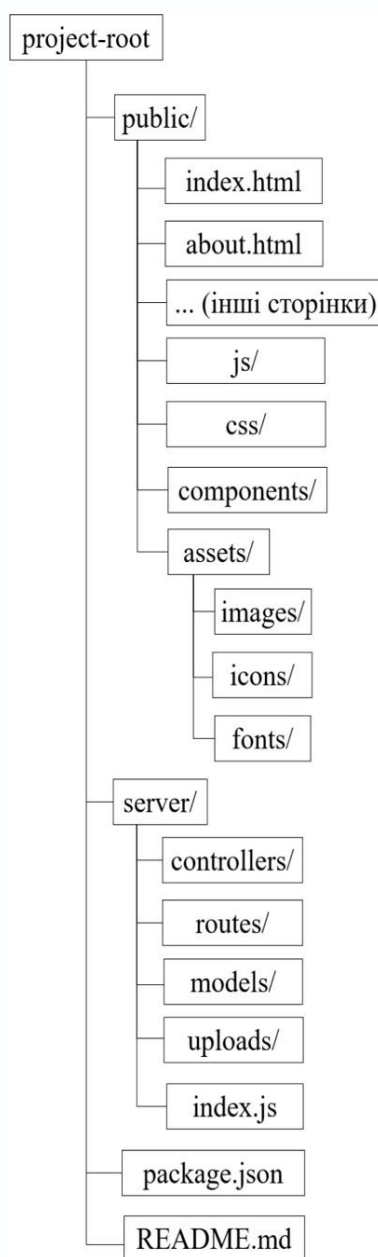


Рис. 3. Структура ресурсу

Такий поділ відповідає підходу компонентного UI, дозволяє уникнути перевантаження і забезпечує логічну масштабованість [13, 16].

Навігація адаптивна та ролезна. Основна навігація реалізована через компонент `button\_menu` – це панель, яка фіксується внизу екрана та відображає різні елементи залежно від ролі користувача. Існує два окремі варіанти:

- `button\_menu\_user.html` – для звичайного відвідувача;
- `button\_menu\_author.html` – для автора, з додатковими кнопками «Додати пост», «Редагувати», «Видалити».

Елементи навігації підвантажуються через JavaScript-модуль `load_header.js`, що централізує керування інтерфейсом. Шапка (`header.html`) винесена в окремий компонент і підключається динамічно з папки `components/` [14]. Замість класичних меню реалізовано фіксовану нижню панель, орієнтовану на мобільні пристрої — таке розташування зручне для взаємодії та відповідає сучасним UX-практикам [17].

Навігація доступна на всіх сторінках без дублювання коду, що поєднує підходи SPA та MPA: зберігається проста HTML-структура, але реалізується динамічна взаємодія [12].

У проєкті передбачено дві ролі: «Відвідувач» має доступ до перегляду контенту; «Автор» може створювати, редагувати, видаляти пости, додавати медіа та чернетки.

Роль визначається за ключем у `localStorage`: створюється об'єкт `currentUser` із `login`, `password` і міткою `isAuthor`. Уся перевірка прав доступу відбувається на клієнтській стороні (модулі `auth.js`, `button_menu.js`), що дозволяє уникнути складної серверної авторизації й підходить для локального застосування [15]. Логіка навігації між HTML-сторінками залежить від типу користувача (рис.4).

Guest	User	Author
index.html post.html about.html favourite.html activity.html login.html	index.html post.html about.html profile.html favourite.html activity.html	index.html post.html about.html profile.html add.html edit.html favourite.html activity.html

Рис. 4. Схема доступу до HTML-сторінок відповідно до ролей користувача

Залежно від ролі користувача змінюється не лише меню, а й структура сторінок: для автора доступні додаткові елементи - редагування медіа, зміна фону, кнопки «Скасувати» або «Вийти без змін». У звичайного користувача ці функції приховані. Схеми (рис. 5-7) ілюструють логіку переходів, доступ для ролей (Автор, Користувач, Гість) і поведінку модальних вікон (login.html).

Навігація реалізована як частина художнього задуму: замість стандартних меню - плаваюча панель, що не відволікає від змісту. Рольова система розмежовує не лише функціональність, а й враження - творець бачить інший інтерфейс, ніж глядач. Кожна сторінка адаптується до користувача, що відкриває шлях до глибшої персоналізації: реакції, коментарі, сповіщення, обране - ці можливості вже закладено в архітектурі.

Таким чином, ресурс розвивається органічно, як цифровий арт-простір, у якому технічні рішення підтримують авторське бачення. Клієнтська та серверна частини. Реалізація ресурсу з унікальним UI-дизайном передбачала чітке розмежування обов'язків між клієнтською та серверною частинами – для спрощення підтримки, масштабування та логіки взаємодії.

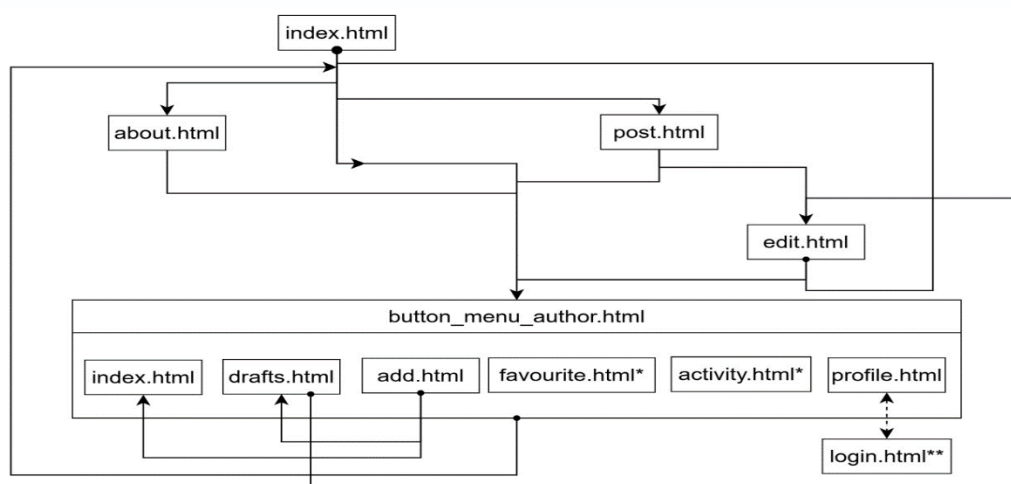


Рис. 5. Схема структури сторінок для ролі «Автор»

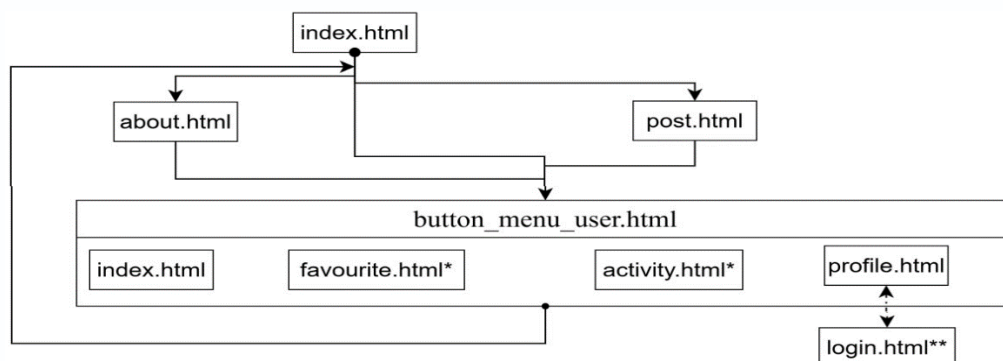


Рис. 6. Схема структури сторінок для ролі «Користувач»

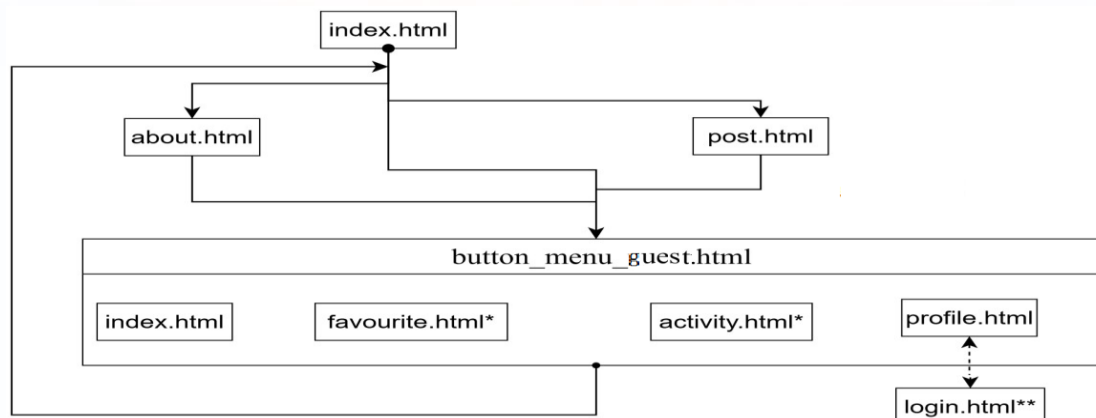


Рис. 7. Схема структури сторінок для ролі «Гість»

Клієнтська частина має гнучкий та динамічний інтерфейс. Реалізована на основі HTML5, CSS3 та JavaScript. Відповідає за відображення контенту, взаємодію користувача з інтерфейсом і первинну логіку перевірок. Основу структури складають окремі HTML-файли для кожного типу сторінки: index.html, post.html, add.html, edit.html, login.html, profile.html тощо. Дизайн інтерфейсу базується на принципі компонентності - всі повторювані елементи (шапка, меню, футер) винесено в окрему папку components, звідки вони динамічно підвантажуються на сторінки за допомогою JavaScript (модуль load_header.js). Це дозволяє централізовано змінювати структуру інтерфейсу, не вносячи правки до кожного HTML-файлу вручну [8]. Функціональність клієнтської частини підтримується набором JavaScript-файлів, які відповідають за ключові дії:

- add_post.js – додавання постів і мультимедіа;
- edit_post.js – редагування існуючих постів;
- auth.js – локальна авторизація та збереження ролі користувача;
- button_menu_author.js/button_menu user.js – керування інтерфейсом

залежно від ролі;

- post.js – завантаження сторінки поста за ID;
- load_header.js – підключення компонентів на сторінках.

Адаптивність реалізовано через медіа-запити, clamp(), Flexbox і Grid — інтерфейс працює від 320px, хоча потребує доопрацювання [9]. Після входу користувача у localStorage створюється ключ currentUser, що дозволяє керувати доступом до функцій без серверної авторизації — зручно для демонстраційного проєкту [18].

Серверна частина реалізована на Node.js з Express.js. REST API обробляє запити за маршрутами: /api/posts (створення, отримання), /api/posts/:id (редагування, видалення), /api/auth (авторизація). Дані постів зберігаються в posts.json у вигляді масиву з ключами: id, type, title, media, draft, author тощо.

Завантаження файлів (зображення, музика) обробляє middleware Multer, шляхи зберігаються у відповідних полях поста [19].

Клієнт надсилає запити через fetch() із FormData, типізовані як POST, PUT, DELETE. У відповідь отримуються JSON-дані або статус. ID редагованого поста зберігається у localStorage як editId, що дозволяє підвантажувати дані в форму редагування.

Такий мінімалістичний підхід із розділенням відповідальностей, асинхронною логікою, без повноцінної СУБД чи важких фреймворків забезпечує достатню функціональність для презентації контенту й демонстрації навичок веброзробки [20].

Інтерфейс додавання, редагування, чернеток. Сторінка add.html реалізує багатофункціональну форму, де автор може створити публікацію будь-якого типу: блог, арт, музика, відео. Тип поста визначається через sessionStorage, і форма адаптується під нього, відображаючи лише релевантні блоки - зображення, відео, треки або текст [21]. Особливістю реалізації є гнучка логіка валідації.

Наприклад, якщо користувач створює пост типу «арт», але не додає жодного зображення – блок «галерея» не відображається, і публікація не дозволяється. Аналогічно – для «музики» потрібен хоча б один трек, для «блогу» – не порожній текст. Усі ці перевірки виконуються до надсилання форми на сервер [22]. Форма містить кнопки для публікації, збереження чернетки, очищення блоків та повернення без змін.

Редагування постів - повторне відкриття та оновлення. Сторінка edit.html підвантажує дані поста за ID з URL. Блоки автоматично заповнюються, медіа-файли можна редагувати або видаляти, нові й старі обробляються окремо через FormData. Після завантаження поста у форму автоматично вставляються всі блоки контенту.

Кожен елемент (трек, арт, обкладинка) має кнопку видалення. Якщо видалити мультимедіа, воно не потрапить у повторне збереження. При повторній відправці форми старі та нові файли обробляються окремо: старі – через спеціальні поля artOld[], musicListOld[], нові – через файли artImages, music у FormData.

Рольова поведінка інтерфейсу. Усі функції (редагування, збереження, видалення) доступні лише автору, що визначається через isAuthor у localStorage [15]. Збереження чернеток - логіка на клієнті та сервері. Чернетки є повноцінними об'єктами, що зберігаються у файлі posts.json, як і звичайні пости. Їх відмінність – у значенні поля draft, яке встановлюється у true. При натисканні на кнопку «Зберегти як чернетку», форма проходить полегшену валідацію: достатньо хоча б одного заповненого елемента. Якщо ж не введено жодного тексту, зображення чи файлу – навіть чернетка не зберігається [23].

Сторінка `drafts.html` відображає лише пости з міткою `draft`, які завантажуються через JavaScript і показуються у вигляді плиток з обкладинкою та описом.

При кліку відкривається `edit.html` з передачею ID в URL [24]. У режимі редагування з'являється кнопка «Видалити», яка надсилає DELETE-запит до сервера й одразу оновлює інтерфейс.

На відміну від типових CMS, інтерфейс створено вручну, що дає повний контроль над логікою і зовнішнім виглядом.

Автор сам обирає, які мультимедійні блоки додати, створюючи пост як мікропроект, а не як звичайну статтю. Чернетки розглядаються не як незавершені матеріали, а як частина творчого процесу — простір для ідей, до яких можна повернутись у зручний момент. Це підкреслює гнучкість і емоційну природу взаємодії з ресурсом.

Інтерфейс гнучкий і не диктує фіксовану структуру: автор сам обирає потрібні блоки — лише текст, відео, галерею чи посилання. Це відповідає принципу «інтерфейс підлаштовується під контент», що є особливо важливим у творчих проєктах [25].

Контент редагується безпосередньо на сторінці через `contenteditable`, що створює ефект живого редактора. Можлива кастомізація: зміна фону, обкладинки, приховування блоків — інтерфейс стає частиною творчого задуму, унікального для кожного поста.

Таким чином, сторінки `add.html`, `edit.html` і `drafts.html` функціонують як динамічне середовище для створення, збереження та редагування контенту. Адаптивність до типу поста, рольова логіка й перевірки форм перетворюють інтерфейс на справжній інструмент креативу.

UI/UX-рішення та мультимедійна структура. Проєкт зосереджений не лише на функціональності, а й на візуальному стилі та зручності взаємодії. UI/UX-рішення впливають на сприйняття, зручність і атмосферу творчого простору [5]. UI - це мінімалізм, атмосфера, кастомізація. Інтерфейс розроблено у стилі мінімалізму: основна увага - контенту. Використано м'які тіні, плавні ефекти, закруглені блоки, адаптивну типографіку [27]. Підтримується можливість перемикання теми (світла/темна) через CSS-класи [28]. Усі сторінки мають єдиний стиль завдяки динамічному підключенню загальних компонентів (шапки, меню, футера) через `load_header.js`. UX орієнтований на зручність, передбачуваність і адаптивність, особливо для мобільних користувачів.

Кнопки логічно розміщено, використовуються медіа-запити, `clamp()`, `flex` та `grid`, що забезпечує комфортну роботу з ресурсом [24].

Підтримуються чотири типи постів: блог, арт, музика, відео. Структура кожного динамічна — виводяться лише заповнені блоки, що виключає порожній контент.

Мультимедіа завантажуються через FormData у папку uploads/, а шляхи до них зберігаються у posts.json, що дозволяє працювати без бази даних. Сторінка post.html має блоки: заголовок, основний і додатковий контент, галерею або плеєр залежно від типу поста, а також майбутній розділ коментарів. Реалізовано горизонтальні кнопки гортання між постами, ефекти fade-in і scroll-snap для плавного скролу. Передбачено також галерею пов'язаних постів, логіка якої ще в розробці. Сервер побудований на Node.js та Express, клієнтська частина – на JavaScript/CSS без сторонніх фреймворків. Структура проєкту модульна, підтримує адаптивність і мультимедіа. Ключові фрагменти коду, що реалізують клієнтську та серверну логіку WEB-ресурсу, такі: обробка постів (postController.js) (рис.8); основний маршрут (index.js) (рис.9); валідація форми створення поста (add_post.js) (рис.10); галерея постів (gallery.css) (рис.11); сторінка окремого поста (post.css) (рис.12); модальне вікно входу (form.css) (рис.13).

```
exports.createPost = (req, res) => { try {  
  const posts = readPosts(); const postId =  
  req.body.id;  
  const isDraft = req.body.draft === 'true';  
  ...  
  const musicFiles = req.files?.music || [];  
  ...  
  const newPost = { id,  
    title: title || prevPost.title || '',  
    ...  
    musicList : musicList.length > 0 ?  
    musicList : (prevPost.musicList || []),  
    ...  
  };  
  res.status(201).json({ message: 'Збережено', post: newPost });  
  } catch (err) {  
    res.status(500).json({message:'Внутрішня помилка сервера' });  
  }  
};
```

Рис. 8. Обробка постів (postController.js).

```
const express = require('express'); const  
app = express();  
const postRoutes = require('./routes/post');  
app.use('/api/posts', postRoutes);  
...  
app.listen(PORT, () => console.log(`Сервер запущено на  
порту ${PORT}`));
```

Рис. 9. Основний маршрут (index.js).

Рис. 10. Валідація форми створення поста (add_post.js).

Рис. 11. Галерея постів (gallery.css)

```
.post-page {
    padding: 20px;
}
.cover-rectangle {
    width: 100%;
    aspect-ratio: 4/3;
    overflow: hidden;
    border-radius: 20px;
}
.post-title {
    font-size: 28px;
    color: white;
    text-align: center;
}
```

Рис. 12. Сторінка окремого поста (post.css).

```
.modal-form {
  width: 90%;
  max-width: 400px;
  background-color: #1e1e1e;
  border-radius: 8px; padding: 20px;
  box-shadow: 0 5px 15px rgba(0,0,0,0.3);
}
.modal-form input { width: 100%;
  margin-bottom: 10px;
  padding: 10px; border-radius: 4px;
}
```

Рис. 13. Модальне вікно входу (form.css).

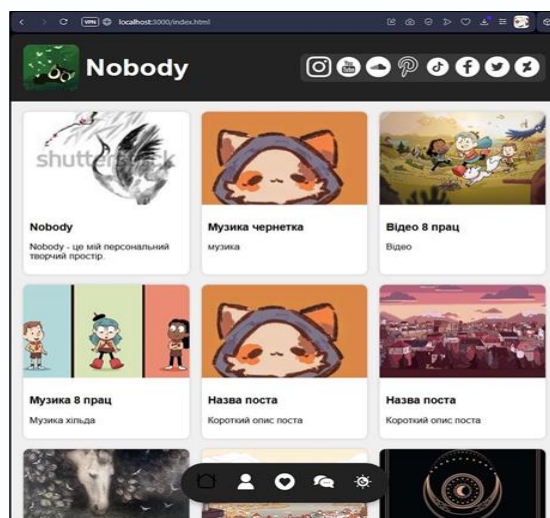


Рис. 14. Інтерфейс WEB-ресурсу в гостьовому режимі

Після авторизації інтерфейс змінюється. З'являються кнопки редагування, видалення постів (рис.15), а також розширене нижнє меню (button_menu_author.html).

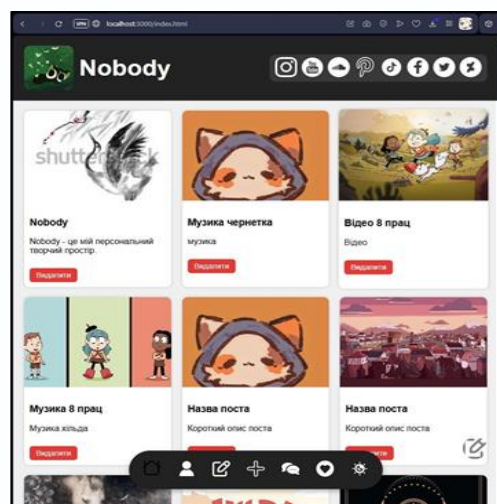


Рис. 15. Інтерфейс WEB-ресурсу після авторизації як автора

Рис. 16. Форма створення поста (тип «арт»)

Рис.17. Пост із вбудованим аудіоплеєром (тип «музика»)

На рисунку 18 показано приклад поста з відео, яке підвантажується з YouTube та адаптується до ширини екрану.

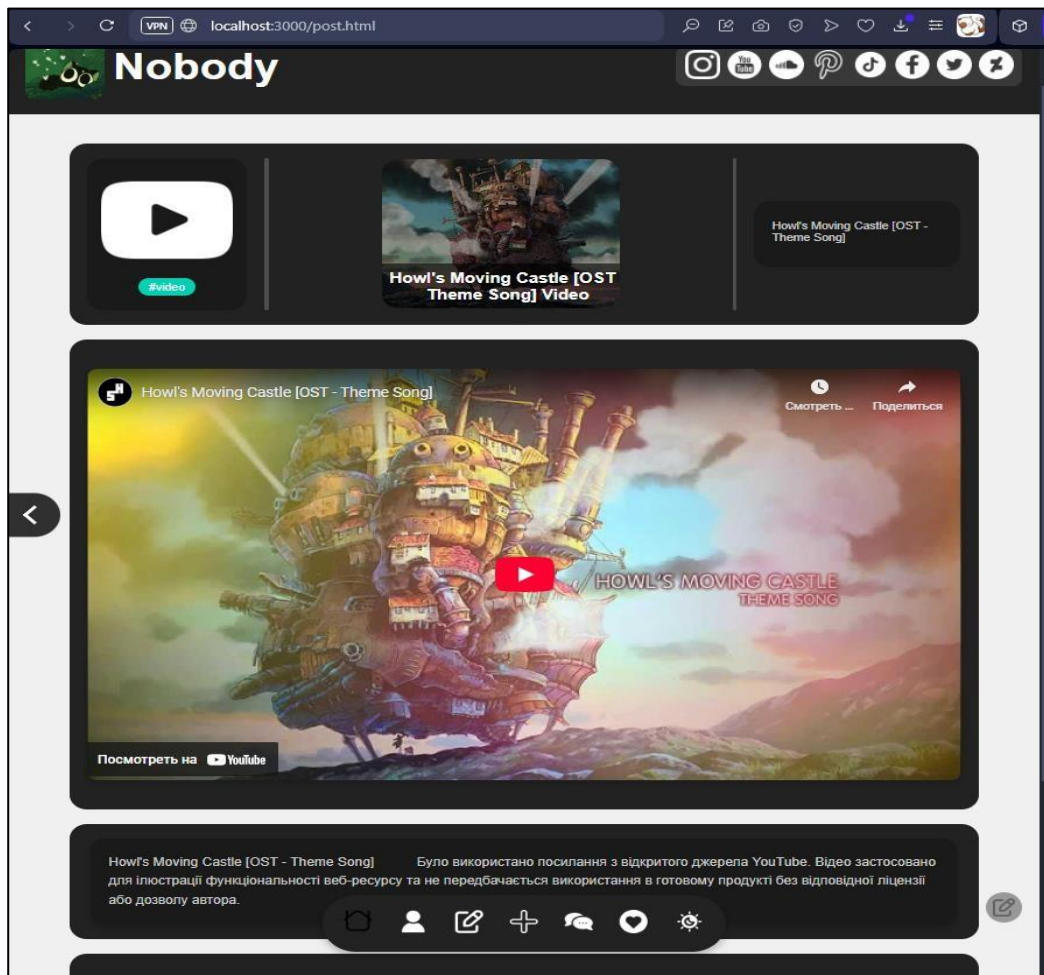


Рис.18. Пост із відео з YouTube (тип «відео»)

Тестування основної функціональності та оцінка інтерфейсу. Для перевірки функціональності WEB-ресурсу було проведено ручне тестування основних сценаріїв користувацької взаємодії.

Тестування виконувалося в браузері Opera через DevTools, серверна частина запускалась локально (Node.js), клієнтська – з папки public/.

Через відсутність онлайн- розгортання адаптивність перевірялась у режимі симуляції (320–768px).

Перевірка функціональності була проведена за ключовими сценаріями: створення поста; редагування поста; збереження чернетки; видалення поста; авторизація та ролі користувача; виведення постів та мультимедійного контенту; перевірка адаптивності інтерфейсу; валідація форм.

В таблиці 1 надано приклад тестування валідації форм.

Таблиця 1

Результати тестування валідації форм

№	Крок	Вхідні дані	Очікуваний результат	Фактичний результат	Статус
1	Залишити заголовок як defaultTitle	defaultTitle	Alert-повідомлення, пост не зберігається	Повідомлення з'являється, форма блокується	Успішно
2	Ввести валідний заголовок	«Новий пост»	Форма приймає дані	Успішно збережено	Успішно
3	Спробувати зберегти пост без опису	Порожнє поле «опис»	Виводиться попередження	Alert виводиться	Успішно
4	Створити відео-пост без відео	Тип: video, відео не задано	Відображається повідомлення про помилку	Збереження заблоковано	Успішно
5	Додати кілька блоків, не заповнюючи їх	Порожні мультимедійні блоки	Вони не повинні зберігатись	Порожні блоки не передаються	Успішно

Проведене тестування підтвердило загальну стабільність основних функцій WEB-ресурсу, відповідність реалізації заявленим функціональним вимогам, а також демонструє потенціал для подальшого вдосконалення, зокрема в аспектах адаптивності, зручності та масштабованості.

На основі проведеної розробки та тестування WEB-ресурсу з унікальним UI-дизайном і мультимедійною структурою, доцільно рекомендувати реалізацію таких можливостей.

1. Удосконалити адаптивність інтерфейсу, зокрема для пристроїв із шириною 320–768 пікселів, забезпечивши коректне масштабування та розташування елементів на всіх сторінках.

2. Завершити інтеграцію темної/світлої теми, реалізувавши повноцінне перемикання збереженого стану теми для всіх компонентів WEB-ресурсу, включно з модальними вікнами та динамічними блоками.

3. Розширити кастомізацію UI-дизайну, надавши кожному типу поста індивідуальне стилістичне оформлення (фони, іконки, шрифти) з урахуванням емоційної складової.

4. Перейти від JSON-файлів до повноцінної бази даних (наприклад, MongoDB або SQLite), що дозволить реалізувати масштабоване збереження, пошук, фільтрацію та оновлення даних.

5. Уніфікувати модальне вікно входу/реєстрації для використання на всіх сторінках WEB-ресурсу, забезпечивши доступ до нього в будь-який момент через інтерфейс.

6. Впровадити модулі, які на тепер реалізовані частково або заплановані, зокрема: повноцінну сторінку «Улюблене»; реалізацію блоку коментарів до постів; функціонал сповіщень про події; збереження профілю користувача з можливістю редагування; галерею пов'язаних постів як інструмент навігації та контекстного зв'язку.

Висновки. Дослідження та розробка розширили функціональні можливостей персонального WEB-ресурсу для творчої самопрезентації через розробку адаптивного середовища з підтримкою мультимедійного контенту, авторизації та індивідуального UI-дизайну. Для цього реалізовано повноцінну клієнт-серверну архітектуру, ручне проектування інтерфейсу та підтримка мультимедіа. Успішне проходження ручного тестування основних функцій підтвердило роботоспроможність ресурсу. У порівнянні з готовими конструкторами сайтів, створений ресурс забезпечує повну гнучкість у дизайні, логіці взаємодії та можливість масштабування під конкретні задачі користувача. Ресурс може бути використаний як персональний арт-блог, цифрове портфоліо або як основа для масштабованої онлайн-платформи. Використання ресурсу доцільно для митців, дизайнерів, контент-креаторів.

Також він може бути адаптований під освітні, презентаційні або експериментальні проекти.

У перспективі WEB-ресурс може розвиватися як масштабована платформа з підтримкою кількох мов, аналітики, гнучкого керування контентом і адаптації під різні аудиторії. Відкрита архітектура та гнучка структура ресурсу дають змогу створити сучасне середовище, що поєднує авторський стиль, інтерактивність і технічну ефективність.

Література:

1. Theme customization and editing. URL: https://www.tumblr.com/docs/en/custom_themes.
2. Audiotool. About the platform. URL: <https://www.audiotool.com/about>.
3. Artsteps Platform. Description. URL: <https://www.artsteps.com/>.
4. Все про професію UI/UX дизайнера. URL: <https://dan-it.com.ua/uk/blog/vse-pro-profesiuyu-ui-ux-dyzajnera/>.
5. Salach W. The Importance of Having a Personal Website for Creatives. URL: <https://www.weronikasalach.com/blog/the-importance-of-having-a-personal-website-for-creatives>
6. HTML та CSS – основи веброзробки. URL: https://w3schoolsua.github.io/html/html_css.html.
7. Duckett J. JavaScript and JQuery: Interactive Front-End Web Development. Indianapolis: Wiley, 2014. 640 p.
8. Visual Studio Code Documentation. URL: <https://code.visualstudio.com/docs>.

- 1330

11. JSON –Intro. Retrieved from https://w3schoolsua.github.io/js/js_json_intro.html [in Ukr]
12. Bahatostorinkovi ta odnostorinkovi dodatky, yikh perevahy ta nedoliky. Retrieved from <https://dou.ua/forums/topic/25444/> [in Ukrainian].
13. JavaScript from Beginner to Professional: Learn JavaScript quickly by building fun, interactive, and dynamic web apps, games, and pages. *Packt Publishing*, 2021. 546.
14. File System (fs/promises). Retrieved from <https://nodejs.org/api/fs.html>.
15. LocalStorage, sessionStorage. Retrieved from <https://uk.javascript.info/localstorage>.
16. Matt Frisbie. Professional JavaScript for Web Developers (Tech Today), 5th Edition. *Wiley*, 2023. 1104.
17. Thumb zone in mobile UI design. Retrieved from <https://uxdesign.cc/thumb-zone-in-mobile-ui-design>.
18. Duckett J. HTML and CSS: Design and Build Websites. *John Wiley & Sons*, 2014. 512.
19. API Documentation: The Developer's Guide. Retrieved from <https://www.youtube.com/watch?v=NDlgT6KgXjo>.
20. JavaScript.info. Retrieved from <https://uk.javascript.info/formdata>.
21. Client-side form validation. Retrieved from https://developer.mozilla.org/en-US/docs/Learn_web_development/Extensions/Forms/Form_validation.
22. Usability Heuristics for User Interface Design. Retrieved from <https://www.nngroup.com/articles/ten-usability-heuristics/>.
23. Marcotte E. Responsive Web Design. Retrieved from <https://www.youtube.com/watch?v=TwWGiO-lDDU>.
24. How to Build Your Online Presence as an Artist. Retrieved from <https://www.flapperpress.com/post/how-to-build-your-online-presence-as-an-artist>.
25. Publisher Quick Start Guide. Retrieved from <https://help.disqus.com/en/articles/1717056-what-is-disqus>.
26. Guidelines. Retrieved from <https://m2.material.io/design/guidelines-overview>.
27. Digital identity: an approach to its nature, concept, and functionalities. Retrieved from <https://academic.oup.com/ijlit/article/doi/10.1093/ijlit/eaee019/7760180>.
28. Osnovy Express.js. Retrieved from <https://itvdm.com/ua/video/nodejs-ua/expressjs> [Ukr]

Журнал

«Наука і техніка сьогодні»

*(Серія «Педагогіка», Серія «Право», Серія «Економіка»,
Серія «Фізико-математичні науки», Серія «Техніка»)*

Випуск № 8(49) 2025

Формат 60x90/8. Папір офсетний.
Гарнітура Times New Roman.
Ум. друк. арк. 8,2. Наклад 100 прим.
Зам. 1

Видавець:

Громадська наукова організація «Всеукраїнська асамблея докторів наук з державного управління» *Свідоцтво
серія ДК №4957 від 18.08.2015 р., Андріївський узвіз, буд.11, оф 68, м. Київ, 04070.*

Підготовлено рекламним агентством
«GoToPrint» Адреса, Україна, Київська обл., м. Київ, вул. Льва Толстого, 63
тел.+38073 996 46 57, +38050 771 27 31, e-mail: gotoprint@gmail.com