

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно - інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Розробка та оптимізація вебінтерфейсу державного електронного
сервісу з урахуванням принципів зручності використання та доступності

Виконав: студент IV курсу, групи СН - 41

спеціальності 122 Комп'ютерні науки

(шифр і назва спеціальності)

(підпис)

Барда С.С.

(прізвище та ініціали)

Керівник

(підпис)

Небесний Р.М.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Шимчук Г.В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Боднарчук І.О.

(прізвище та ініціали)

Рецензент

(підпис)

(прізвище та ініціали)

Тернопіль
2026

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно - інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.
(підпис) (прізвище та ініціали)

« » 2026 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

Студенту Барді Святославу Степановичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка та оптимізація вебінтерфейсу державного електронного сервісу з урахуванням принципів зручності використання та доступності

Керівник роботи Небесний Руслан Михайлович, доктор філософії, доцент кафедри КН
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «14» травня 2026 року № 4/9 - 239

2. Термін подання студентом завершеної роботи 26 червня 2026р.

3. Вихідні дані до роботи Літературні та інтернет - джерела з питань проектування вебінтерфейсів, зручності використання (usability), вебдоступності (WCAG 2.1), електронного урядування та цифрових державних сервісів. Офіційна документація React, Node.js, PostgreSQL, а також матеріали щодо розробки клієнт - серверних вебзастосунків, автентифікації користувачів та організації REST API.

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ

- 1) Аналіз предметної області та існуючих державних електронних сервісів
- 2) Проектування вебінтерфейсу державного електронного сервісу електронних звернень громадян
- 3) Реалізація та тестування вебінтерфейсу державного електронного сервісу електронних звернень громадян
- 4) Безпека життєдіяльності, основи охорони праці

Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1. Титульна сторінка. 2. Актуальність дослідження. 3. Мета, Об'єкт, Предмет дослідження.

4. Завдання дослідження. 5. Аналіз існуючих державних електронних сервісів. 6. Порівняння існуючих аналогів систем електронних звернень. 7. Архітектура розробленої системи. 8. Структура бази даних системи. 9. Реєстрація, авторизація та розмежування прав доступу. 10. та взаємодія компонентів системи. 11. Створення та подання електронних звернень. 12.

Перегляд та відстеження статусів звернень. 13. Панель адміністратора та керування зверненнями. 14. Забезпечення доступності та зручності використання інтерфейсу. 15.

Результати тестування системи. 16. Висновки. 17. Завершальний.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці			

7. Дата видачі завдання 26 січня 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	26.01.2026	Виконано
2.	Підбір джерел по темі кваліфікаційної роботи	31.01.2026 - 03.02.2026	Виконано
3.	Опрацювання джерел по темі кваліфікаційної роботи	04.02.2026 - 06.02.2026	Виконано
4.	Виконання дослідження щодо проєктування, розробки та тестування вебінтерфейсу державного електронного сервісу електронних звернень громадян	07.02.2026 - 11.02.2026	Виконано
5.	Оформлення розділу «Аналіз предметної області та існуючих державних електронних сервісів»	17.05.2026 - 23.05.2026	Виконано
6.	Оформлення розділу «Вибір технологій та засобів реалізації вебінтерфейсу державного електронного сервісу електронних звернень громадян»	24.05.2026 - 28.05.2026	Виконано
7.	Оформлення розділу «Проєктування вебінтерфейсу державного електронного сервісу електронних звернень громадян»	29.05.2026 - 04.06.2026	Виконано
8.	Оформлення розділу «Реалізація та тестування вебінтерфейсу державного електронного сервісу електронних звернень громадян»	05.06.2026 - 06.06.2026	Виконано
9.	Виконання завдання до розділу «Безпека життєдіяльності» праці»	07.06.2026 - 08.06.2026	Виконано
10.	Виконання завдання до підрозділу «Основи охорони праці»	09.06.2026 - 10.06.2026	Виконано
11.	Нормоконтроль	11.06.2026 - 13.06.2026	Виконано
12.	Перевірка на плагіат	25.06.2026	Виконано
13.	Попередній захист кваліфікаційної роботи	26.06.2026	Виконано
14.	Захист кваліфікаційної роботи	27.06.2026	

Студент

(підпис)

Барда.С.С.

(прізвище та ініціали)

Керівник роботи

(підпис)

Небесний Р.М.

(прізвище та ініціали)

АНОТАЦІЯ

Розробка та оптимізація вебінтерфейсу державного електронного сервісу з урахуванням принципів зручності використання та доступності // Кваліфікаційна робота освітнього рівня «Бакалавр» // Барда Святослав Степанович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно - інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СН - 41 // Тернопіль, 2026 // С.58, рис. – 12, табл. – 1, кресл. – 3, додат. – 0, бібліогр. – 35.

Ключові слова: Electronic Appeals, Web Interface, Accessibility, Usability, React, Node.js, PostgreSQL, REST API, JWT, WCAG.

Кваліфікаційна робота присвячена розробленню вебінтерфейсу державного електронного сервісу електронних звернень громадян.

У першому розділі кваліфікаційної роботи проведено аналіз предметної області, досліджено особливості державних електронних сервісів та розглянуто сучасні підходи до побудови вебінтерфейсів для електронної взаємодії громадян із державними установами. Також проаналізовано вимоги до usability та accessibility вебресурсів.

У другому розділі кваліфікаційної роботи обґрунтовано вибір технологічного стеку для реалізації програмного рішення. Описано архітектуру системи, структуру бази даних та основні принципи побудови клієнтської і серверної частин вебсервісу.

У третьому розділі кваліфікаційної роботи описано практичну реалізацію вебінтерфейсу державного електронного сервісу електронних звернень громадян. Реалізовано клієнтську частину на основі React, серверну логіку із використанням Node.js та Express.js, а також базу даних PostgreSQL. Проведено тестування функціональності, доступності та продуктивності розробленого вебсервісу.

Об'єкт дослідження: процес електронної взаємодії громадян із державними установами під час подання та обробки електронних звернень.

Предмет дослідження: методи, технології та програмні засоби розроблення вебінтерфейсу державного електронного сервісу електронних звернень громадян.

ANNOTATION

Development and Optimization of a Web Interface for a Government E-Service Considering Usability and Accessibility Principles// Qualification work of the educational level "Bachelor" // Barda Sviatoslav // Ternopil Ivan Pulyu National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Computer Science, group SN - 41 // Ternopil, 2026 // P. 58, fig. – 12, tabl. – 1, drawings – 3, annexes – 0, references – 35.

Keywords: Electronic Appeals, Web Interface, Accessibility, Usability, React, Node.js, PostgreSQL, REST API, JWT, WCAG.

The qualification work is devoted to the development of a web interface for the state electronic service of citizens' electronic appeals.

The first chapter of the qualification work presents an analysis of the subject area, examines the features of state electronic services, and considers modern approaches to building web interfaces for electronic interaction between citizens and government institutions. Special attention is paid to usability and accessibility requirements for web resources.

The second chapter of the qualification work substantiates the choice of the technological stack for implementing the software solution. The system architecture, database structure, and the main principles of building the client - side and server - side components are described.

The third chapter of the qualification work describes the practical implementation of the web interface of the state electronic service for citizens' electronic appeals. The client - side part was developed using React, the server - side logic was implemented using Node.js and Express.js, and PostgreSQL was used as the database system. Functional, accessibility, and performance testing of the developed web service were carried out.

Object of research: the process of electronic interaction between citizens and government institutions during submission and processing of electronic appeals.

Subject of research: methods, technologies, and software tools for developing a web interface of the state electronic service for citizens' electronic appeals.

ПЕРЕЛІК СКОРОЧЕНЬ

API (англ. Application Programming Interface) - програмний інтерфейс застосунку.

JWT (англ. JSON Web Token) - вебтокен у форматі JSON для автентифікації та авторизації користувачів.

REST (англ. Representational State Transfer) - архітектурний стиль побудови вебсервісів.

JSON (англ. JavaScript Object Notation) - текстовий формат обміну даними.

HTTP (англ. HyperText Transfer Protocol) - протокол передачі даних у мережі Інтернет.

HTML (англ. HyperText Markup Language) - мова розмітки вебсторінок.

CSS (англ. Cascading Style Sheets) - каскадні таблиці стилів для оформлення вебсторінок.

SQL (англ. Structured Query Language) - структурована мова запитів до баз даних.

CRUD (англ. Create, Read, Update, Delete) - основні операції створення, читання, оновлення та видалення даних.

SPA (англ. Single Page Application) - односторінковий вебзастосунок.

DBMS (англ. Database Management System) - система керування базами даних.

WCAG (англ. Web Content Accessibility Guidelines) - рекомендації щодо забезпечення доступності вебконтенту.

ORM (англ. Object - Relational Mapping) - технологія взаємодії між об'єктами програмного коду та реляційною базою даних.

UI (англ. User Interface) - користувацький інтерфейс.

UX (англ. User Experience) - досвід взаємодії користувача із системою.

ЗМІСТ

ВСТУП.....	10
РОЗДІЛ 1. АНАЛІТИЧНА ЧАСТИНА.....	12
1.1 Поняття та роль державних електронних сервісів у сучасному цифровому середовищі	12
1.2 Особливості вебінтерфейсів державних інформаційних систем	13
1.3 Принципи usability у веброзробці.....	15
1.4 Accessibility як обов’язкова вимога сучасних вебсервісів.....	16
1.5 Аналіз існуючих аналогів та постановка задачі (з посиланнями)	18
1.6 Висновок до розділу 1	20
РОЗДІЛ 2. ПРОЄКТУВАННЯ ВЕБІНТЕРФЕЙСУ ДЕРЖАВНОГО ЕЛЕКТРОННОГО СЕРВІСУ	22
2.1 Обґрунтування вибору технологій розробки	22
2.2 Проєктування архітектури системи	23
2.3 Проєктування структури бази даних.....	25
2.4 Проєктування REST API	27
2.5 Проєктування користувацького інтерфейсу.....	29
2.6 Висновок до розділу 2	31
РОЗДІЛ 3. ПРАКТИЧНА ЧАСТИНА (ДЕМОНСТРАЦІЯ, АНАЛІЗ ТА УЗАГАЛЬНЕННЯ ОТРИМАНИХ РЕЗУЛЬТАТІВ).....	32
3.1 Проєктування архітектури вебсистеми електронних звернень громадян	32
3.2 Розробка серверної частини та бази даних.....	34
3.3 Реалізація клієнтської частини вебзастосунку.....	38
3.4 Реалізація принципів usability та accessibility	44
3.5 Тестування системи та аналіз результатів.....	45
3.6 Висновок до розділу 3	49
РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ ТА ОСНОВИ ОХОРОНИ ПРАЦІ.....	50
4.1 Безпека життєдіяльності.....	50

	9
4.2 Основи охорони праці	51
4.3 Висновок до розділу 4	52
ВИСНОВКИ.....	54
ПЕРЕЛІК ДЖЕРЕЛ.....	55

ВСТУП

Актуальність теми. У сучасних умовах цифрової трансформації державного управління особливої актуальності набуває впровадження електронних сервісів для взаємодії громадян із органами влади. Використання інформаційних технологій дозволяє спростити процес подання звернень, підвищити оперативність їх опрацювання та забезпечити прозорість комунікації між громадянами і державними установами. Одним із важливих напрямів розвитку електронного урядування є створення веборієнтованих систем, які забезпечують можливість дистанційного подання та відстеження статусу звернень. У зв'язку з цим розроблення вебінтерфейсу державного електронного сервісу звернень громадян є актуальним завданням, спрямованим на підвищення доступності та ефективності надання електронних послуг.

Мета і задачі дослідження. Метою роботи є проектування, реалізація та тестування вебінтерфейсу державного електронного сервісу електронних звернень громадян, який забезпечує зручну взаємодію користувачів із системою подання та перегляду звернень.

Для досягнення поставленої мети необхідно вирішити такі задачі: – провести аналіз особливостей функціонування електронних сервісів звернень громадян;

- дослідити сучасні підходи до розробки вебінтерфейсів державних інформаційних систем;
- проаналізувати вимоги до доступності та зручності використання вебресурсів;
- спроектувати архітектуру програмної системи;
- розробити структуру бази даних для зберігання інформації про користувачів та звернення;
- реалізувати клієнтську частину вебзастосунку з використанням React;

- реалізувати серверну частину із застосуванням Node.js та Express.js;
 - впровадити механізми автентифікації та авторизації користувачів;
 - реалізувати функціонал створення, перегляду та відстеження статусу звернень;
- провести тестування роботи розробленого вебсервісу.

Об'єкт дослідження. Об'єктом дослідження є процес електронної взаємодії громадян із державними установами під час подання та опрацювання звернень.

Предмет дослідження. Предметом дослідження є методи, засоби та програмні технології розробки вебінтерфейсів електронних сервісів звернень громадян, а також механізми зберігання, обробки та відображення інформації у вебзастосунках.

Практичне значення одержаних результатів. Практичне значення роботи полягає у створенні функціонального вебсервісу для подання та перегляду електронних звернень громадян. Розроблене програмне рішення забезпечує реєстрацію користувачів, автентифікацію, створення нових звернень, перегляд історії поданих заяв та контроль поточного статусу їх розгляду. Реалізований вебінтерфейс може бути використаний як основа для подальшого розвитку електронних сервісів державних установ, інтеграції з іншими інформаційними системами та впровадження додаткових механізмів електронної взаємодії громадян із органами влади.

РОЗДІЛ 1. АНАЛІТИЧНА ЧАСТИНА

1.1 Поняття та роль державних електронних сервісів у сучасному цифровому середовищі

За останні роки цифрові технології суттєво змінили підхід до надання державних послуг [7]. Якщо раніше більшість адміністративних процедур вимагала особистої присутності, паперових документів та значних витрат часу, то сьогодні значна частина таких процесів переноситься в онлайн - середовище.

Саме в цьому контексті важливу роль відіграють державні електронні сервіси [7]. Їх поява стала логічним етапом розвитку електронного урядування та цифровізації державного сектору.

Державний електронний сервіс - це інформаційна система або цифрове рішення, через яке громадяни можуть отримувати адміністративні, інформаційні чи комунікаційні послуги в електронному форматі [7].

Практичне значення державних електронних сервісів полягає у спрощенні взаємодії між громадянином і державними установами. Користувач отримує можливість виконувати необхідні дії дистанційно, без фізичного відвідування державних органів.

Цифровий формат надання послуг має низку переваг. Насамперед скорочується час отримання послуги. Крім цього, знижується навантаження на працівників установ, оскільки частина процесів автоматизується.

Впровадження державних електронних сервісів сприяє:

- зменшенню бюрократичних процедур;
- скороченню паперового документообігу;
- підвищенню прозорості процесів;
- покращенню якості обслуговування громадян [7].

В Україні цифровізація державного сектору активно розвивається протягом останніх років. З'явилися сервіси, які дозволяють громадянам

отримувати широкий спектр послуг дистанційно. Одним із найбільш відомих прикладів є Дія [3].

Окремий напрям державних електронних сервісів пов'язаний із системами електронних звернень громадян. Такі сервіси дають можливість подавати звернення, заяви, скарги або пропозиції через інтернет.

Подібний формат має суттєві переваги для користувачів. Людина може подати звернення у зручний час, відстежувати статус його розгляду та отримувати відповідь без необхідності особистого контакту з установою.

Попри активний розвиток цифрових рішень, проблема якості багатьох державних електронних сервісів залишається актуальною. Часто користувачі стикаються зі складною навігацією, перевантаженим інтерфейсом або недостатньо зрозумілою логікою роботи системи.

З цієї причини під час розробки вебінтерфейсу державного електронного сервісу важливо враховувати не лише функціональні можливості, а й якість взаємодії користувача із системою.

1.2 Особливості вебінтерфейсів державних інформаційних систем

Вебінтерфейс є основним засобом взаємодії користувача з інформаційною системою. Саме через нього виконується більшість дій: введення даних, навігація між сторінками, отримання інформації та використання функціоналу сервісу [1].

Для державних інформаційних систем якість вебінтерфейсу має особливе значення. На відміну від багатьох комерційних платформ, державними електронними сервісами користуються дуже різні категорії населення.

Серед користувачів можуть бути люди різного віку, рівня технічної підготовки та цифрової грамотності. Саме тому вебінтерфейс державного електронного сервісу повинен бути максимально зрозумілим.

Користувач не має витратити багато часу на те, щоб розібратися у структурі сторінки або знайти потрібну функцію.

Однією з характерних особливостей державних інформаційних систем є орієнтація на виконання конкретних сценаріїв взаємодії:

- подання заяв;
- реєстрація звернень;
- отримання довідок;
- відстеження статусу документів.

Кожен із цих сценаріїв має виконуватися просто та передбачувано.

Якісний вебінтерфейс державного електронного сервісу повинен забезпечувати логічну структуру сторінок. Інформація має бути організована послідовно, а навігація - залишатися зрозумілою навіть для нових користувачів [1].

Важливим фактором є мінімізація зайвих дій. Якщо для виконання простої операції користувачу потрібно проходити надто багато етапів, це негативно впливає на зручність використання.

Окрему увагу необхідно приділяти адаптивності інтерфейсу. Значна частина користувачів працює з державними електронними сервісами через смартфони або планшети.

Не менш важливою вимогою є стабільність роботи системи. Повільне завантаження сторінок, технічні помилки або нестабільна робота негативно впливають на досвід користувача.

Для державних сервісів критичним є також питання доступності. Вебінтерфейс повинен бути придатним для використання різними категоріями користувачів, включно з людьми, які використовують assistive technologies [2].

Сучасний підхід до розробки вебінтерфейсу державного електронного сервісу базується на поєднанні кількох характеристик:

- функціональності;
- простоти;
- безпеки;
- адаптивності;
- доступності.

Саме поєднання цих характеристик визначає якість державного електронного сервісу та рівень його ефективності.

1.3 Принципи usability у веброзробці

Якість вебінтерфейсу значною мірою визначається тим, наскільки легко користувач може взаємодіяти із системою. Саме це описує поняття usability, яке є одним із базових у сучасній веброзробці [1].

Usability характеризує зручність використання цифрового інтерфейсу. Фактично йдеться про те, наскільки швидко користувач здатний зрозуміти логіку роботи системи, виконати потрібну дію та досягти бажаного результату без зайвих труднощів [8].

Для вебінтерфейсів високий рівень usability безпосередньо впливає на ефективність роботи із системою. Якщо інтерфейс зрозумілий, користувач швидше адаптується до сервісу та рідше допускає помилки.

Особливого значення usability набуває під час розробки вебінтерфейсу державного електронного сервісу. Такі сервіси повинні бути зручними не лише для досвідчених користувачів, а й для людей із базовими цифровими навичками.

Під час оцінювання usability зазвичай враховують кілька ключових критеріїв [1]:

- простота освоєння;
- ефективність роботи;
- швидкість виконання завдань;
- кількість помилок;
- загальний комфорт використання.

Одним із важливих критеріїв є зрозуміла структура інтерфейсу. Користувач повинен швидко орієнтуватися в розташуванні основних елементів сторінки та розуміти призначення кожного блоку.

Не менш важливою є логічна навігація. Меню, кнопки та переходи між сторінками повинні бути організовані таким чином, щоб шлях до основного функціоналу залишався коротким і передбачуваним.

Для державного електронного сервісу електронних звернень громадян це має особливе значення. Наприклад, створення нового звернення не повинно вимагати складних або зайвих дій.

Важливу роль відіграє також робота з формами введення даних. Якщо користувач заповнює форму реєстрації, авторизації чи створення звернення, інтерфейс має допомагати йому на кожному етапі.

Це реалізується за допомогою:

- зрозумілих назв полів;
- валідації введених даних;
- повідомлень про помилки;
- підказок для користувача.

Ще одним фактором, який впливає на usability, є швидкодія системи. Навіть якісно спроектований вебінтерфейс державного електронного сервісу може створювати негативний досвід, якщо сторінки завантажуються повільно або працюють нестабільно.

У сучасній веброзробці usability розглядається як невід’ємна частина проектування інтерфейсу. Чим простішою та зрозумілішою є взаємодія із системою, тим вищою буде якість користувацького досвіду.

1.4 Accessibility як обов’язкова вимога сучасних вебсервісів

Розвиток сучасних вебтехнологій призвів до того, що доступність цифрових продуктів стала одним із ключових критеріїв їх якості [2]. Під accessibility розуміють здатність вебінтерфейсу бути доступним для максимально широкого кола користувачів незалежно від фізичних, сенсорних або технічних обмежень [9].

Ідея доступності полягає в тому, що вебінтерфейс державного електронного сервісу повинен залишатися придатним для використання в різних сценаріях взаємодії. Це стосується не лише людей з інвалідністю, але й користувачів із тимчасовими обмеженнями або специфічними умовами роботи.

Для державних електронних сервісів питання доступності має особливе значення. Оскільки такі сервіси створюються для всіх громадян, доступ до їх функціоналу повинен бути універсальним.

Основою сучасного підходу до accessibility є міжнародний стандарт WCAG 2.1 [2]. Ці рекомендації описують вимоги до побудови доступних вебінтерфейсів та широко використовуються під час розробки цифрових систем.

WCAG базується на чотирьох фундаментальних принципах [2]:

- perceivable;
- operable;
- understandable;
- robust.

Перший принцип стосується сприйняття інформації. Контент повинен бути поданий таким чином, щоб користувач міг його коректно сприймати. Наприклад, зображення мають містити альтернативний текст, а текстовий контент - залишатися читабельним.

Другий принцип пов'язаний із керованістю інтерфейсу. Елементи вебінтерфейсу повинні бути доступними не лише через мишу, а й через клавіатуру або допоміжні засоби взаємодії.

Принцип зрозумілості передбачає, що структура сторінок, логіка навігації та поведінка інтерфейсу повинні бути передбачуваними.

Останній принцип орієнтований на технічну сумісність із браузером та assistive technologies.

У межах веброзробки доступність забезпечується за рахунок набору конкретних технічних рішень [9]:

- достатньої контрастності кольорів;
- семантичної HTML - розмітки;

- клавіатурної навігації;
- коректної структури заголовків;
- використання ARIA - атрибутів.

Контрастність є важливою для читабельності тексту та сприйняття вмісту сторінки. Низький контраст між текстом і фоном ускладнює взаємодію із сервісом.

Семантична HTML - структура покращує сприйняття сторінки допоміжними технологіями. Коректне використання заголовків, форм та навігаційних блоків робить взаємодію із системою більш зрозумілою.

ARIA - атрибути доповнюють стандартну HTML - розмітку та дозволяють точніше описувати призначення елементів інтерфейсу.

У сучасній веброзробці accessibility вже не розглядається як додаткове покращення. Для вебінтерфейсу державного електронного сервісу це базова вимога, яка безпосередньо впливає на якість взаємодії користувачів із системою.

1.5 Аналіз існуючих аналогів та постановка задачі (з посиланнями)

Перед розробкою вебінтерфейсу державного електронного сервісу електронних звернень громадян було проведено аналіз існуючих аналогів. Метою аналізу стало визначення сильних і слабких сторін сучасних державних електронних сервісів, а також виявлення проблем у сфері зручності використання та доступності.

Для аналізу було обрано три категорії державних електронних сервісів:

- Дія;
- електронні приймальні органів місцевого самоврядування;
- сервіси електронних петицій [3], [7].

Першим аналогом є Дія - один із найбільш розвинених державних електронних сервісів в Україні [3].

Серед основних переваг цього державного електронного сервісу можна виділити сучасний дизайн, високу швидкість роботи та якісно реалізовану

навігацію. Інтерфейс побудований відповідно до сучасних вимог UX/UI, що позитивно впливає на зручність використання.

Разом із цим функціональність сервісу орієнтована на широкий спектр послуг, а не конкретно на роботу з електронними зверненнями громадян.

Другу групу аналогів становлять електронні приймальні органів місцевого самоврядування [7]. Такі державні електронні сервіси дозволяють громадянам подавати звернення, скарги та заяви через вебінтерфейс.

Головною перевагою таких рішень є наявність безпосереднього функціоналу для роботи з електронними зверненнями громадян.

Водночас аналіз показав, що багато електронних приймалень мають суттєві недоліки. Найчастіше спостерігається застарілий дизайн, складна структура сторінок та недостатньо зрозуміла навігація. Крім того, у багатьох випадках рівень доступності залишається недостатнім [2].

Третім аналогом були обрані сервіси електронних петицій [7]. Такі державні електронні сервіси дають можливість громадянам створювати петиції та підтримувати вже опубліковані ініціативи.

Сильними сторонами цих сервісів є відносно проста структура інтерфейсу та зрозуміла логіка взаємодії з користувачем.

Проте функціональність сервісів електронних петицій є вузькоспеціалізованою. Вони не орієнтовані на повноцінну роботу із персональними електронними зверненнями громадян та контролем статусу їх опрацювання.

Для узагальнення результатів проведеного аналізу було сформовано порівняльну таблицю існуючих аналогів (табл. 1.1).

Таблиця 1.1 – Порівняння існуючих аналогів

Критерій	Дія	Електронні приймальні	Електронні петиції
Сучасний інтерфейс	Високий	Середній	Середній
Зручність використання	Висока	Середня	Висока
Адаптивність	Висока	Низька	Середня
Рівень доступності	Середній	Низький	Середній
Електронні звернення громадян	Частково	Так	Ні

Результати аналізу показали, що існуючі державні електронні сервіси мають як сильні сторони, так і певні недоліки.

Найбільш поширеними проблемами залишаються:

- недостатня зручність використання;
- складна структура інтерфейсу;
- слабка адаптивність;
- обмежена підтримка доступності [1], [2].

Отримані результати підтверджують актуальність теми кваліфікаційної роботи.

Основним завданням роботи є розробка та оптимізація вебінтерфейсу державного електронного сервісу електронних звернень громадян з урахуванням принципів зручності використання та доступності.

1.6 Висновок до розділу 1

У першому розділі було розглянуто теоретичні основи, що стосуються розробки державних електронних сервісів та сучасних вебінтерфейсів.

Проведений аналіз показав, що цифровізація державного сектору суттєво змінює формат взаємодії між громадянами та органами влади. Електронні

сервіси спрощують доступ до адміністративних послуг, скорочують час обробки запитів та підвищують ефективність комунікації.

У межах розділу було визначено особливості вебінтерфейсів державних інформаційних систем. Встановлено, що такі системи повинні поєднувати функціональність, простоту використання, адаптивність, безпеку та стабільність роботи.

Окрему увагу приділено принципам usability, які безпосередньо впливають на якість користувацького досвіду. Зручність навігації, логічна структура інтерфейсу та простота виконання основних дій є важливими характеристиками сучасного вебсервісу.

Також було розглянуто питання accessibility як невід'ємної складової сучасної веброзробки. Доступність інтерфейсу є особливо важливою для державних сервісів, оскільки вони повинні бути однаково зручними для всіх категорій користувачів.

Аналіз існуючих аналогів дозволив виявити типові недоліки сучасних систем електронних звернень, серед яких складність інтерфейсу, недостатня адаптивність та обмежена підтримка доступності.

За результатами проведеного аналізу було сформовано основні вимоги до розробки вебзастосунку електронного сервісу звернень громадян, проектування та реалізація якого розглядатимуться у наступних розділах роботи.

РОЗДІЛ 2. ПРОЄКТУВАННЯ ВЕБІНТЕРФЕЙСУ ДЕРЖАВНОГО ЕЛЕКТРОННОГО СЕРВІСУ

2.1 Обґрунтування вибору технологій розробки

Під час розробки вебінтерфейсу державного електронного сервісу електронних звернень громадян було обрано технології, які забезпечують зручність розробки, стабільність роботи та можливість подальшого розширення функціоналу системи.

Для реалізації клієнтської частини використано бібліотеку React [5]. Дана технологія дозволяє будувати інтерфейс із незалежних компонентів, що спрощує структуру проєкту та полегшує повторне використання окремих елементів. Крім того, React забезпечує швидке оновлення даних на сторінці без її повного перезавантаження, що позитивно впливає на зручність роботи користувача.

Для створення та запуску клієнтського застосунку було використано інструмент Vite. Його використання дало можливість прискорити процес розробки завдяки швидкому запуску локального середовища та оптимізованій збірці проєкту.

Організацію навігації між сторінками реалізовано за допомогою бібліотеки React Router DOM. Це дозволило забезпечити переходи між головною сторінкою, сторінками реєстрації та авторизації, особистим кабінетом користувача, а також сторінками роботи зі зверненнями.

Для оформлення інтерфейсу було використано Tailwind CSS. Даний інструмент дозволяє швидко створювати адаптивні сторінки за допомогою готових CSS - класів без необхідності написання великої кількості власних стилів. Використання Tailwind CSS спростило реалізацію адаптивного дизайну та забезпечило єдиний стиль усіх сторінок системи.

Серверну частину вебзастосунку реалізовано на платформі Node.js [6] із використанням фреймворку Express.js. Такий вибір дозволив використовувати мову JavaScript як на клієнтській, так і на серверній стороні, що спростило

розробку та підтримку програмного коду. Express.js використовується для створення REST API та обробки HTTP - запитів від клієнтської частини.

Для зберігання інформації про користувачів та звернення використано систему керування базами даних PostgreSQL [7]. Вибір цієї СКБД обумовлений її надійністю, підтримкою реляційної моделі даних та можливістю ефективної роботи зі структурованою інформацією.

Автентифікацію користувачів реалізовано за допомогою JWT - токенів. Для безпечного зберігання паролів використано механізм хешування на основі бібліотеки bcryptjs. Такий підхід дозволяє підвищити рівень захисту облікових даних користувачів.

Під час вибору клієнтської технології розглядалися також Angular та Vue. Проте для даного проєкту React виявився більш доцільним завдяки великій кількості готових бібліотек і простоті інтеграції з іншими компонентами системи.

Отже, обраний технологічний стек забезпечує реалізацію всіх функціональних можливостей вебінтерфейсу державного електронного сервісу електронних звернень громадян та відповідає поставленим вимогам до розроблюваної системи.

2.2 Проєктування архітектури системи

Для реалізації вебінтерфейсу державного електронного сервісу електронних звернень громадян було обрано клієнт - серверну архітектуру. Такий підхід передбачає розподіл системи на окремі компоненти, кожен із яких виконує власні функції. Загальна структура взаємодії між компонентами розробленої системи наведена на рисунку 2.1.

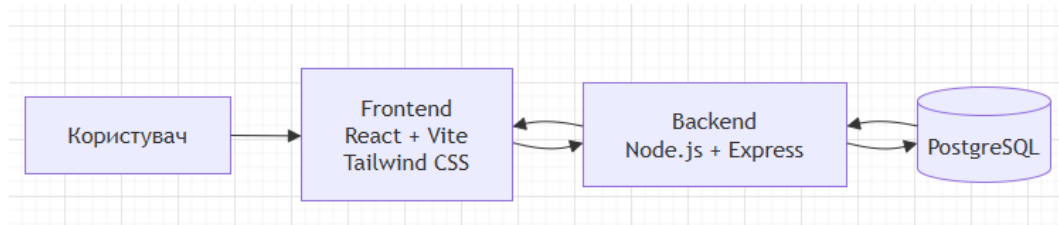


Рисунок 2.1 – Архітектура вебінтерфейсу державного електронного сервісу електронних звернень громадян

Архітектура системи складається з трьох основних рівнів:

- клієнтська частина;
- серверна частина;
- база даних.

Такий спосіб організації системи спрощує подальше розширення функціональних можливостей сервісу без необхідності суттєвої зміни вже реалізованих компонентів.

Клієнтська частина реалізована засобами React [5] і відповідає за взаємодію користувача з вебінтерфейсом. На цьому рівні відображаються сторінки системи, форми введення даних, елементи навігації та результати виконання запитів.

Основними функціями клієнтської частини є:

- відображення інформації користувачеві;
- обробка введених даних;
- валідація форм;
- надсилання HTTP - запитів до сервера.

Серверна частина реалізована на платформі Node.js [6] із використанням Express.js. Вона забезпечує обробку запитів від клієнта, перевірку прав доступу користувачів, роботу з базою даних та формування відповідей для інтерфейсу.

До функцій серверної частини належать:

- реєстрація користувачів;
- авторизація;
- обробка електронних звернень;

- перевірка даних;
- взаємодія з базою даних.

Для зберігання інформації використовується база даних PostgreSQL [7]. У ній зберігаються відомості про користувачів системи та створені ними електронні звернення.

Взаємодія між компонентами системи здійснюється таким чином:

1. Користувач виконує певну дію у вебінтерфейсі.
2. Клієнтська частина формує HTTP - запит до сервера.
3. Серверна частина обробляє отриманий запит.
4. За потреби виконується звернення до бази даних.
5. Результат обробки повертається клієнту у форматі JSON.
6. Отримані дані відображаються у вебінтерфейсі.

Використання клієнт - серверної архітектури дозволяє відокремити інтерфейс користувача від серверної логіки та сховища даних. Це спрощує подальшу підтримку проєкту, модернізацію окремих компонентів системи та розширення її функціональних можливостей.

2.3 Проєктування структури бази даних

Проєктування структури бази даних є одним із важливих етапів розробки вебінтерфейсу електронного сервісу звернень громадян. Саме база даних забезпечує зберігання інформації, необхідної для роботи системи, а також підтримує взаємозв'язки між її основними компонентами.

Для реалізації сховища даних використано систему керування базами даних PostgreSQL [7]. Вибір цієї СКБД обумовлений її надійністю, підтримкою реляційної моделі даних та можливістю забезпечення цілісності інформації під час роботи з великими обсягами записів.

У межах розробленої системи було спроектовано дві основні таблиці: `users` та `appeals`. Структуру основних сутностей та зв'язків між ними подано на

рисунку 2.2. Така структура є достатньою для реалізації функцій реєстрації користувачів, авторизації та роботи зі зверненнями.

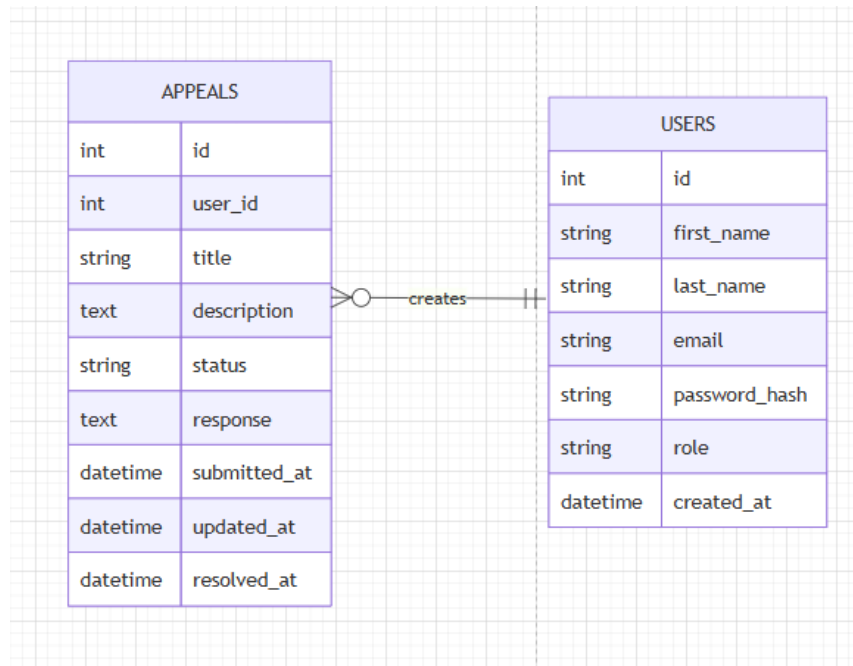


Рисунок 2.2 – ER - діаграма бази даних системи електронних звернень громадян

Таблиця users призначена для зберігання відомостей про зареєстрованих користувачів системи. Для кожного користувача зберігаються унікальний ідентифікатор, ім'я, прізвище, адреса електронної пошти, пароль у хешованому вигляді, роль користувача та службова інформація про створення й оновлення запису.

Для реалізації механізму розмежування прав доступу використовується поле role. У системі передбачено дві ролі: citizen та admin. Перша роль використовується для звичайних користувачів сервісу, друга - для адміністраторів, які здійснюють обробку звернень.

Таблиця appeals використовується для зберігання електронних звернень громадян. Для кожного звернення зберігаються заголовок, текст опису, поточний статус розгляду, відповідь адміністратора, а також дати створення, оновлення та завершення обробки звернення.

Для відображення етапів опрацювання звернень використовується набір статусів: `submitted`, `in_progress`, `resolved` та `rejected`. Завдяки цьому користувач має можливість контролювати поточний стан розгляду свого звернення.

Між таблицями `users` та `appeals` реалізовано зв'язок типу «один до багатьох». Один користувач може створювати декілька звернень, проте кожне звернення належить лише одному користувачеві. Зв'язок реалізовано за допомогою зовнішнього ключа `appeals.user_id`, який посилається на запис у таблиці `users`.

Для підвищення продуктивності бази даних створено індекси для полів, які найчастіше використовуються під час пошуку та фільтрації інформації. Зокрема, індексування застосовано до полів `email`, `status` та `user_id`. Це дозволяє скоротити час виконання запитів та покращити швидкодію системи.

Таким чином, спроектована структура бази даних забезпечує надійне зберігання інформації та підтримує всі функціональні можливості вебінтерфейсу електронних звернень громадян.

2.4 Проєктування REST API

Для забезпечення взаємодії між клієнтською та серверною частинами системи було реалізовано REST API. Такий підхід дозволяє відокремити інтерфейс користувача від серверної логіки та забезпечити зручний обмін даними між компонентами програмного продукту.

Серверну частину системи реалізовано на платформі Node.js [6], яка забезпечує обробку HTTP - запитів та формування відповідей для клієнтського застосунку.

Обмін даними між клієнтом і сервером здійснюється за допомогою протоколу HTTP, а передавання інформації виконується у форматі JSON. Це дозволяє забезпечити сумісність між усіма компонентами системи та спрощує інтеграцію окремих модулів.

Усі маршрути API умовно поділено на три основні групи:

- маршрути автентифікації;
- маршрути роботи з профілем користувача;
- маршрути роботи зі зверненнями громадян.

Маршрути автентифікації використовуються для створення нового облікового запису та входу користувача до системи. Під час реєстрації виконується перевірка коректності введених даних, після чого інформація зберігається у базі даних. Після успішної авторизації користувач отримує токен доступу для роботи із захищеними ресурсами системи.

Функціональність роботи з профілем користувача передбачає отримання інформації про поточного користувача та оновлення персональних даних. Доступ до таких маршрутів мають лише авторизовані користувачі.

Основна частина REST API призначена для роботи з електронними зверненнями громадян. За допомогою відповідних маршрутів користувач може створювати нові звернення, переглядати список власних звернень та отримувати детальну інформацію щодо кожного з них.

Під час створення звернення сервер перевіряє отримані дані та формує новий запис у базі даних. Для нового звернення автоматично встановлюється початковий статус `submitted`. Надалі статус може змінюватися залежно від результатів його опрацювання.

Для захисту API використовується механізм автентифікації на основі JWT - токенів. Перед виконанням захищених запитів сервер перевіряє дійсність токена та визначає користувача, який ініціював запит. Такий підхід дозволяє обмежити доступ до персональної інформації та запобігти виконанню несанкціонованих операцій.

Обрана структура API орієнтована на використання як вебклієнтом, так і потенційними мобільними застосунками, що може бути корисним у разі подальшого розвитку проекту.

Отже, розроблений REST API забезпечує взаємодію між усіма компонентами системи та реалізує функціональні можливості, необхідні для роботи вебінтерфейсу електронних звернень громадян.

2.5 Проектування користувацького інтерфейсу

Проектування користувацького інтерфейсу є важливим етапом розробки вебінтерфейсу електронного сервісу звернень громадян, оскільки саме інтерфейс забезпечує взаємодію користувача із системою. Під час проектування основна увага приділялася зручності використання, зрозумілій навігації, адаптивності та доступності інтерфейсу.

Загальну структуру користувацького інтерфейсу вебсервісу наведено на рисунку 2.3. Розроблений інтерфейс складається з публічної та авторизованої частин. Такий підхід дозволяє розмежувати функціональні можливості залежно від статусу користувача в системі.

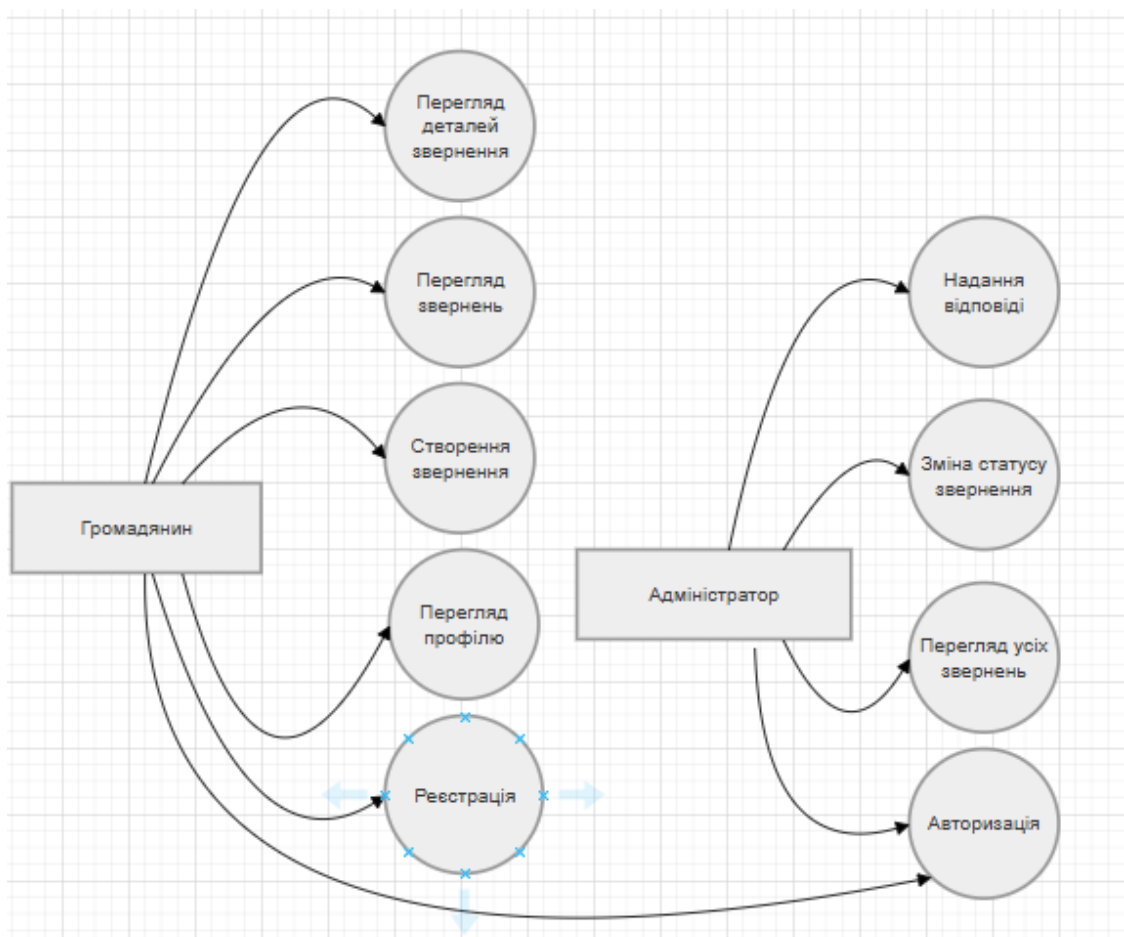


Рисунок 2.3 – Структура користувацького інтерфейсу вебсервісу електронних звернень громадян

Публічна частина вебінтерфейсу доступна без проходження авторизації. До її складу входять головна сторінка, сторінки реєстрації та входу до системи. Головна сторінка містить коротку інформацію про призначення сервісу та надає можливість перейти до основних дій користувача. Сторінки реєстрації та авторизації реалізовані у вигляді форм введення даних із перевіркою коректності заповнення полів.

Після успішного входу користувач отримує доступ до авторизованої частини системи. Основним елементом цієї частини є особистий кабінет, з якого здійснюється доступ до функцій роботи зі зверненнями.

Для створення нового звернення передбачено окрему сторінку з формою введення даних. Користувач заповнює заголовок та опис звернення, після чого інформація надсилається на сервер для подальшого опрацювання.

Для перегляду раніше створених звернень реалізовано сторінку «Мої звернення». На ній відображається перелік звернень користувача із зазначенням назви, дати створення та поточного статусу розгляду. Для кожного запису передбачено можливість переходу до сторінки детального перегляду.

Сторінка деталей звернення містить повну інформацію про звернення, включаючи текст повідомлення, дату подання, поточний статус та відповідь адміністратора у разі її наявності.

Під час проектування інтерфейсу враховувалися рекомендації щодо забезпечення доступності вебресурсів WCAG 2.1 [2]. Для цього було використано семантичну структуру HTML - документів, логічну ієрархію заголовків та підтримку навігації за допомогою клавіатури. Також реалізовано адаптивне відображення сторінок для різних розмірів екранів, що забезпечує коректну роботу системи як на персональних комп'ютерах, так і на мобільних пристроях.

Таким чином, спроектований користувацький інтерфейс забезпечує зручну взаємодію користувача із системою та дозволяє ефективно виконувати основні операції, пов'язані з поданням і переглядом електронних звернень громадян.

Під час проектування інтерфейсу враховувалося, що користувачами сервісу можуть бути люди з різним рівнем цифрових навичок. Саме тому особлива увага приділялася простоті навігації та мінімізації кількості дій, необхідних для створення звернення

2.6 Висновок до розділу 2

У другому розділі дипломної роботи було виконано проектування вебінтерфейсу державного електронного сервісу електронних звернень громадян. У ході роботи визначено основні технології, необхідні для реалізації клієнтської та серверної частин програмного продукту, а також обґрунтовано доцільність їх використання в межах поставлених завдань.

Для розробки користувацького інтерфейсу обрано бібліотеку React [5], що дозволяє створювати динамічні вебзастосунки з компонентною структурою. Серверну частину системи побудовано на платформі Node.js [6] із використанням Express.js, що забезпечує обробку HTTP - запитів та взаємодію з базою даних. Для зберігання інформації про користувачів і звернення використано систему керування базами даних PostgreSQL [7].

У межах проектування було сформовано клієнт - серверну архітектуру застосунку, визначено структуру бази даних, основні сутності та зв'язки між ними. Також розроблено REST API, яке забезпечує обмін даними між окремими компонентами системи та реалізацію функцій авторизації, створення і обробки електронних звернень.

Під час проектування користувацького інтерфейсу особливу увагу приділено зручності використання, логічній навігації та адаптивності вебзастосунку. Крім того, враховано рекомендації WCAG 2.1 [2], що сприяє підвищенню доступності сервісу для різних категорій користувачів.

Отримані результати стали основою для подальшої практичної реалізації вебінтерфейсу електронного сервісу звернень громадян та його функціональних можливостей.

РОЗДІЛ 3. ПРАКТИЧНА ЧАСТИНА (ДЕМОНСТРАЦІЯ, АНАЛІЗ ТА УЗАГАЛЬНЕННЯ ОТРИМАНИХ РЕЗУЛЬТАТІВ)

3.1 Проєктування архітектури вебсистеми електронних звернень громадян

У практичній частині роботи було розроблено вебзастосунок для електронного подання звернень громадян до державних установ. Основна ідея проєкту полягає у спрощенні взаємодії користувача з електронним сервісом за рахунок зручного інтерфейсу та цифрової форми подачі звернень.

Розроблена система дозволяє користувачу:

- зареєструвати обліковий запис;
- увійти у систему;
- створити нове звернення;
- переглядати власні звернення;
- відстежувати статус їх обробки.

Під час розробки було обрано класичну клієнт - серверну архітектуру. Frontend і backend працюють як окремі частини системи та взаємодіють між собою через REST API.

Такий підхід має кілька переваг:

- зрозуміла структура проєкту;
- простіша підтримка коду;
- зручне масштабування функціоналу.

Клієнтська частина відповідає за взаємодію з користувачем: відображення сторінок, обробку подій інтерфейсу та надсилання запитів до сервера. Серверна частина виконує обробку запитів, перевірку даних, авторизацію та роботу з базою даних.

Для розробки frontend - частини використано:

- React;
- Vite;

- React Router DOM;
- Axios;
- Tailwind CSS.

Під час вибору технології для реалізації клієнтської частини розглядалися різні підходи до побудови інтерфейсу. Перевагу React було надано через можливість організувати застосунок у вигляді незалежних компонентів, що спрощує внесення змін та подальше розширення функціоналу.

Vite використовувався як середовище збірки та локального запуску. Завдяки цьому проєкт швидко запускається у development - режимі.

Навігацію між сторінками реалізовано через React Router DOM. Це дозволило створити SPA - застосунок, у якому перехід між сторінками відбувається без повного перезавантаження браузера.

Для взаємодії з backend API використано Axios. Через нього виконуються запити для:

- авторизації;
- роботи з профілем;
- створення звернень;
- отримання списку звернень.

Для стилізації було використано Tailwind CSS. Це дозволило швидко створити сучасний адаптивний інтерфейс без написання великої кількості окремих CSS - файлів. Загальну архітектуру розробленої системи наведено на рисунку 3.1.

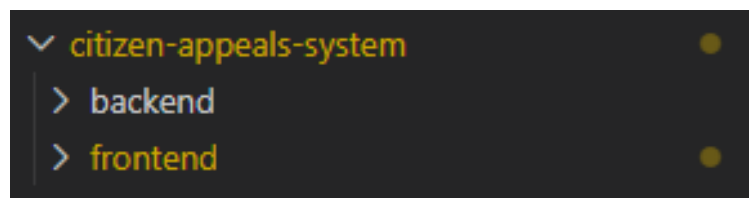


Рисунок 3.1 – Загальна структура проєкту

Серверну частину реалізовано з використанням:

- Node.js;

- Express.js;
- PostgreSQL;
- JWT;
- bcryptjs.

Node.js використовується для виконання серверного JavaScript - коду. На його основі побудовано backend - застосунок з використанням Express.js.

Express забезпечує реалізацію API - маршрутів та обробку HTTP - запитів.

Для зберігання даних використовується PostgreSQL. У базі даних зберігається інформація про користувачів та створені ними звернення.

JWT застосовується для автентифікації користувачів після входу в систему. Паролі перед збереженням у базі хешуються за допомогою bcryptjs.

Структурно проєкт поділено на дві основні частини:

- frontend;
- backend.

Frontend містить сторінки, компоненти, маршрути та API - сервіси. Backend включає маршрути, middleware, конфігурацію бази даних та бізнес - логіку.

Такий поділ забезпечує зрозумілу архітектуру системи та спрощує подальший розвиток вебсервісу.

3.2 Розробка серверної частини та бази даних

Серверна частина вебзастосунку реалізована на базі Node.js та Express.js. Основним завданням backend - рівня є обробка HTTP - запитів, реалізація бізнес - логіки, автентифікація користувачів та взаємодія з базою даних PostgreSQL.

Для зберігання даних було обрано PostgreSQL. Це реляційна система керування базами даних, яка добре підходить для побудови вебсервісів із чітко визначеними зв'язками між сутностями.

У розробленій системі реалізовано дві основні таблиці:

- users;
- appeals.

Таблиця users містить інформацію про зареєстрованих користувачів системи.

Основними полями таблиці є:

- id;
- first_name;
- last_name;
- email;
- password_hash;
- role;
- is_active;
- created_at;
- updated_at.

Для кожного користувача зберігаються персональні дані, облікові дані для входу та службова інформація, необхідна для роботи вебсервісу. Структуру таблиці users у базі даних наведено на рисунку 3.2.

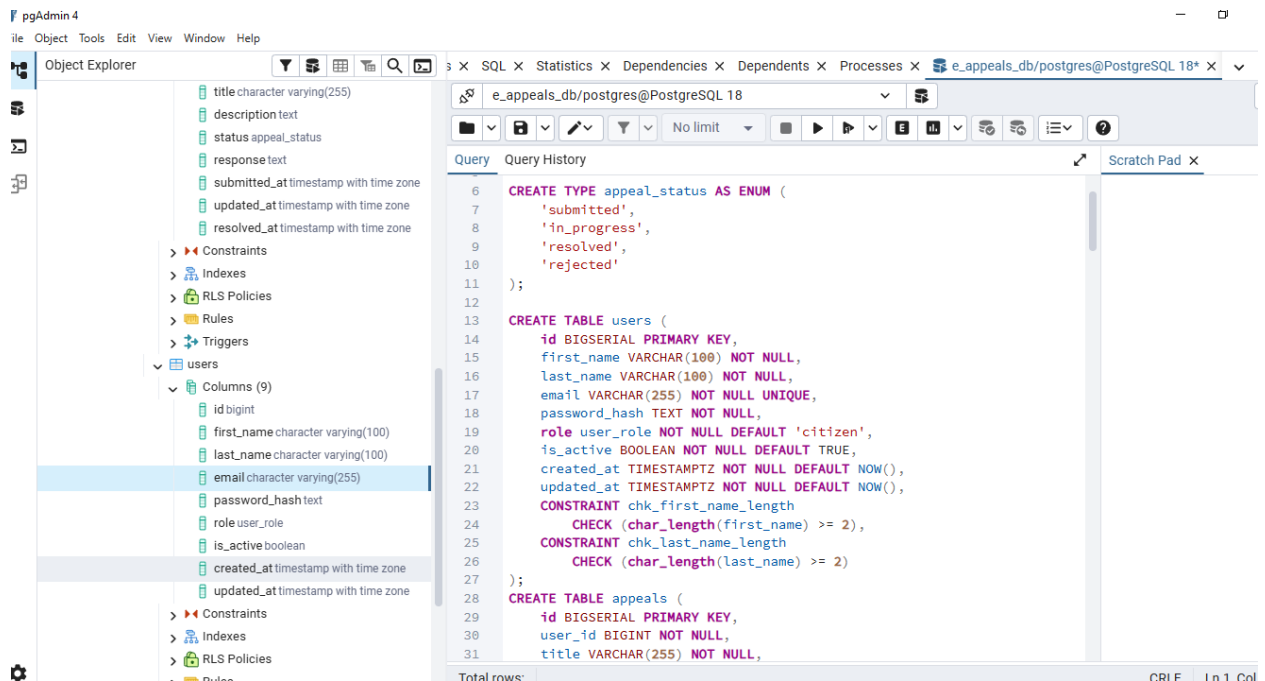


Рисунок 3.2 – Таблиця users

Таблиця `appeals` використовується для збереження інформації про електронні звернення громадян.

Вона містить такі поля:

- `id`;
- `user_id`;
- `title`;
- `description`;
- `status`;
- `response`;
- `submitted_at`;
- `updated_at`;
- `resolved_at`.

Для статусів звернень створено ENUM - тип `appeal_status` із такими значеннями:

- `submitted`;
- `in_progress`;
- `resolved`;
- `rejected`.

```

28 CREATE TABLE appeals (
29     id BIGSERIAL PRIMARY KEY,
30     user_id BIGINT NOT NULL,
31     title VARCHAR(255) NOT NULL,
32     description TEXT NOT NULL,
33     status appeal_status NOT NULL DEFAULT 'submitted',
34     response TEXT,
35
36     submitted_at TIMESTAMPTZ NOT NULL DEFAULT NOW(),
37
38     updated_at TIMESTAMPTZ NOT NULL DEFAULT NOW(),
39
40     resolved_at TIMESTAMPTZ,
41
42     CONSTRAINT fk_appeals_user
43     FOREIGN KEY (user_id)
44     REFERENCES users(id)
45     ON DELETE CASCADE,
46
47     CONSTRAINT chk_title_length
48     CHECK (char_length(title) >= 5),
49
50     CONSTRAINT chk_description_length
51     CHECK (char_length(description) >= 20)
52 );
53
Total rows: CRLF Ln 1, 0

```

Рисунок 3.3 – Таблиця `appeals`

Між таблицями `users` та `appeals` реалізовано зв'язок типу «один до багатьох». Один користувач може створювати необмежену кількість звернень, тоді як кожне звернення належить лише одному користувачу.

Для забезпечення цілісності даних використано:

- PRIMARY KEY;
- FOREIGN KEY;
- CHECK constraints;
- UNIQUE constraints.

Додатково для підвищення продуктивності створено індекси для часто використовуваних полів, зокрема `email`, `user_id`, `status` та `submitted_at`.

Для автоматичного оновлення службового поля `updated_at` реалізовано SQL - trigger, який спрацьовує під час зміни записів у таблицях.

Система підтримує JWT - автентифікацію. Після успішного входу користувача сервер генерує токен, який містить базову інформацію про користувача:

- id;
- email;
- role.

Токен використовується для доступу до захищених endpoint API.

У межах серверної частини реалізовано такі основні маршрути API.

Authentication:

- POST /api/auth/register
- POST /api/auth/login

Users:

- GET /api/users/profile
- PUT /api/users/profile

Appeals:

- POST /api/appeals
- GET /api/appeals/my
- GET /api/appeals/id

Паролі користувачів не зберігаються у відкритому вигляді. Перед записом у базу даних виконується хешування із використанням bcryptjs, що підвищує безпеку системи.

3.3 Реалізація клієнтської частини вебзастосунку

Після реалізації серверної частини та бази даних було виконано розробку клієнтської частини вебзастосунку. Основною задачею frontend - рівня стало створення зручного, інтуїтивного та адаптивного інтерфейсу для взаємодії користувача із системою електронних звернень.

Клієнтська частина реалізована за допомогою React із використанням компонентного підходу. Це дозволило розділити інтерфейс на окремі незалежні модулі, що значно спростило підтримку та повторне використання коду.

У межах застосунку реалізовано сторінки для двох категорій користувачів:

- неавторизовані користувачі;
- авторизовані користувачі.

До публічної частини належать:

- головна сторінка;
- сторінка входу;
- сторінка реєстрації;
- сторінка з інформацією про сервіс.

Головна сторінка є точкою входу в систему та містить базову інформацію про призначення сервісу. На ній розміщено основний опис системи, навігаційне меню та кнопки швидкого переходу до сторінок реєстрації та входу. Інтерфейс головної сторінки вебзастосунку наведено на рисунку 3.4

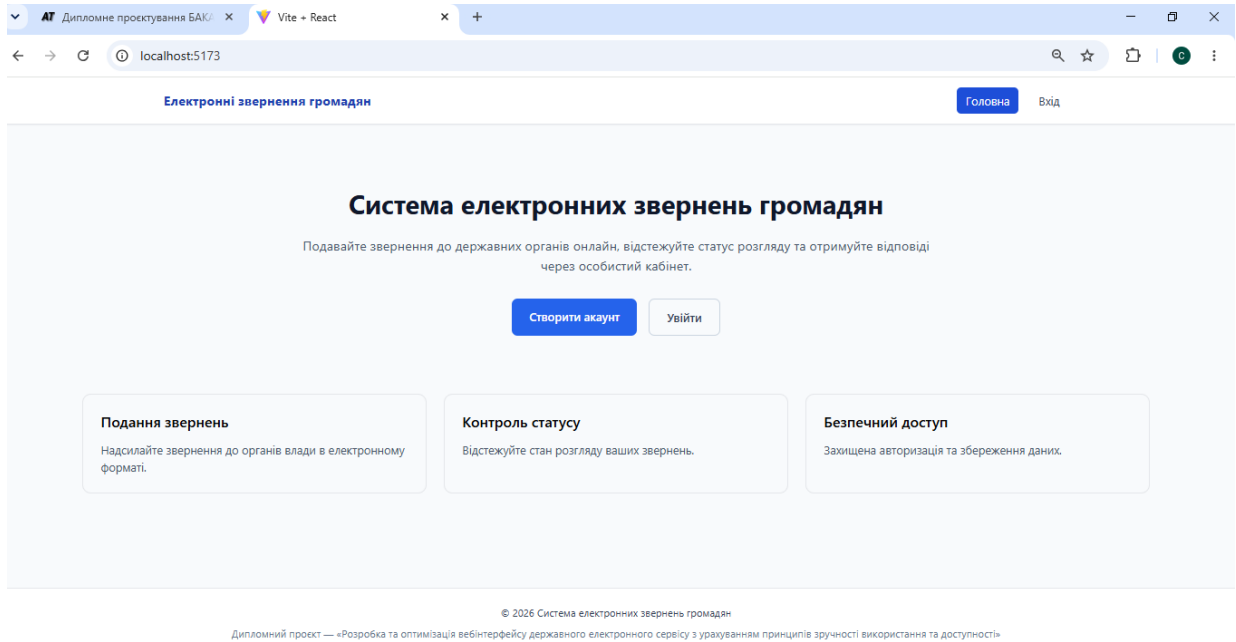


Рисунок 3.4 – Головна сторінка

Інтерфейс головної сторінки спроектовано таким чином, щоб користувач одразу розумів основне призначення системи та її ключові можливості.

Для входу користувача у систему створено окрему сторінку авторизації. Вона містить форму введення електронної пошти та пароля, необхідних для перевірки облікових даних. Інтерфейс сторінки авторизації наведено на рисунку 3.5.

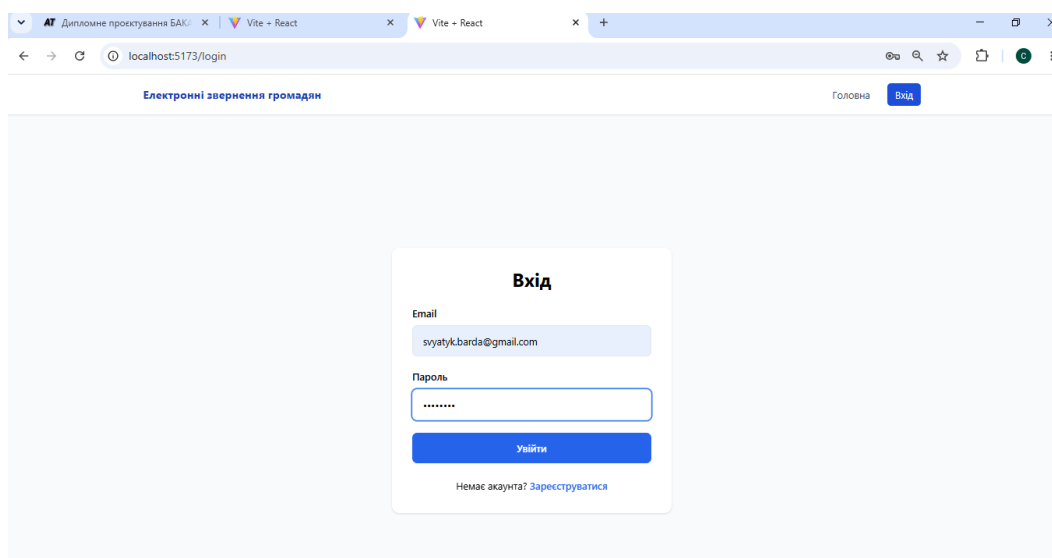


Рисунок 3.5 – Сторінка входу

Форма входу містить поля введення електронної пошти та пароля. Після надсилання форми виконується запит до backend API для перевірки введених облікових даних.

У разі успішної автентифікації користувач отримує JWT - токен, який зберігається у localStorage браузера. Надалі цей токен використовується для доступу до захищених маршрутів системи.

Для нових користувачів передбачено окрему сторінку реєстрації. Через цю форму користувач створює обліковий запис для подальшої роботи із сервісом. Інтерфейс сторінки реєстрації користувача наведено на рисунку 3.6.

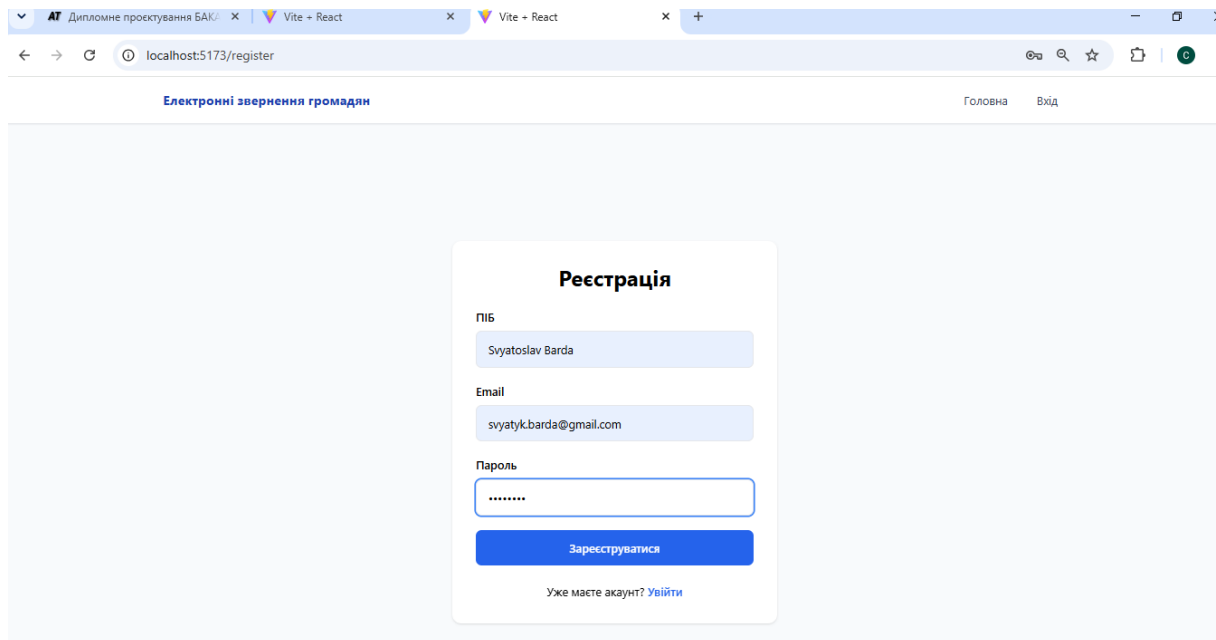


Рисунок 3.6 – Сторінка реєстрації

Форма реєстрації містить поля для введення:

- імені;
- прізвища;
- email;
- пароля.

Після проходження авторизації користувач отримує доступ до захищеної частини застосунку.

Після успішного входу користувач переходить до особистого кабінету. Саме він є основною сторінкою для подальшої роботи із системою. Інтерфейс особистого кабінету користувача наведено на рисунку 3.7.

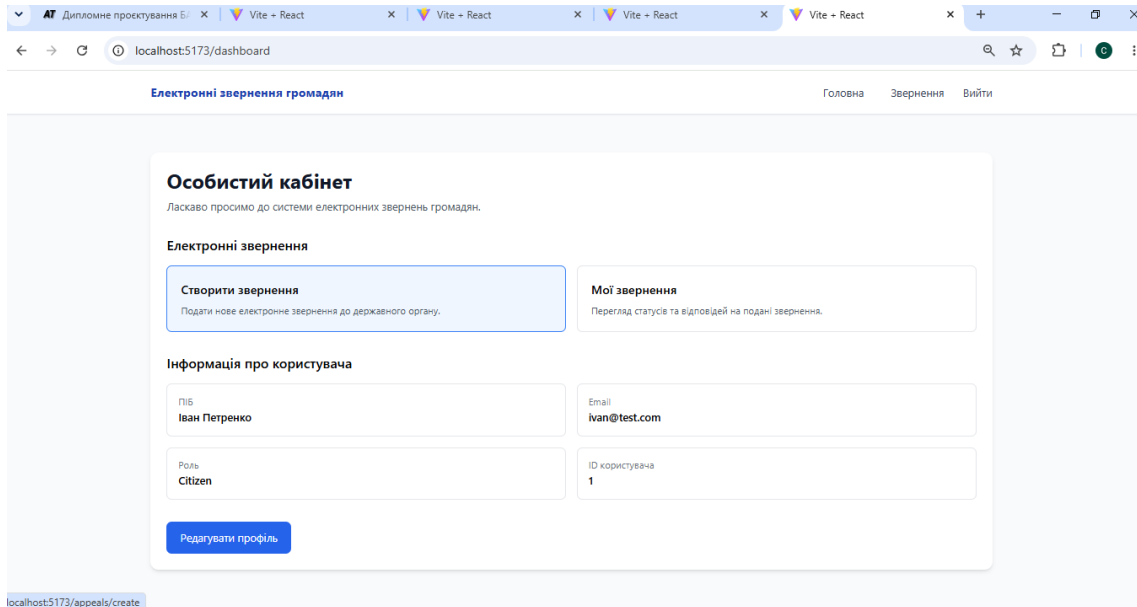


Рисунок 3.7 – Особистий кабінет

Особистий кабінет виступає основним центром навігації та забезпечує швидкий доступ до ключового функціоналу системи.

Через нього користувач може:

- переглядати профіль;
- створювати нові звернення;
- переглядати власні звернення;
- відстежувати статус звернень.

Створення нового звернення виконується через окрему сторінку з формою введення даних. Інтерфейс максимально спрощено, щоб користувач міг швидко подати звернення. Сторінку створення нового звернення наведено на рисунку 3.8.

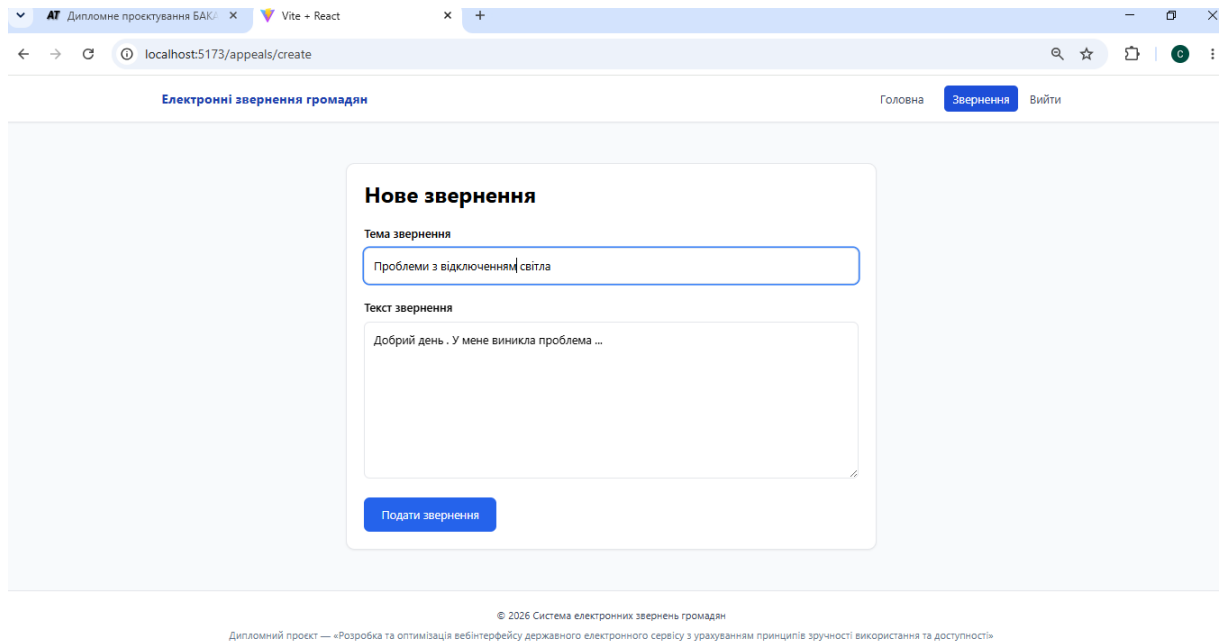


Рисунок 3.8 – Створення звернення

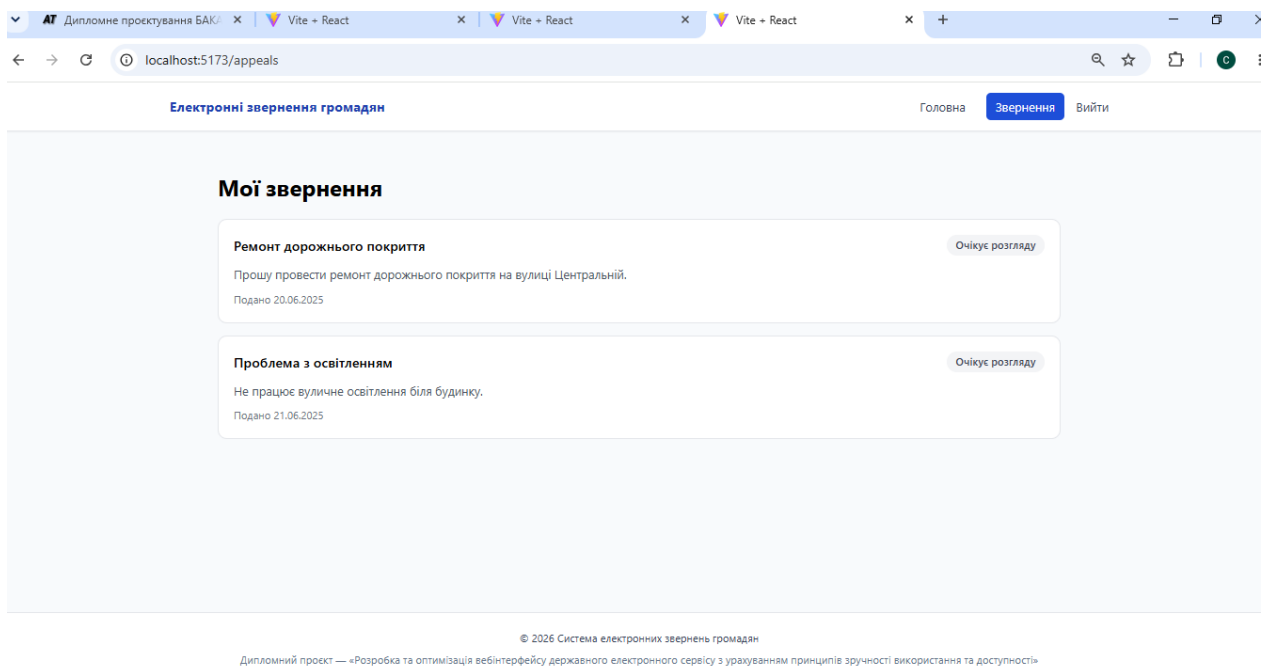


Рисунок 3.9 – Мої звернення

Форма подання звернення містить два основні поля:

- тема звернення;
- детальний опис.

Перед надсиланням даних виконується клієнтська валідація для перевірки коректності введених значень. Після успішного проходження перевірки звернення передається на сервер і записується у базу даних.

Після створення звернення йому автоматично присвоюється початковий статус submitted.

Для перегляду історії звернень реалізовано окрему сторінку.

На цій сторінці користувач бачить список усіх поданих звернень із короткою інформацією про кожне з них.

Для кожного звернення відображаються:

- назва;
- дата створення;
- поточний статус.

Користувач також може перейти до детального перегляду звернення. Інтерфейс сторінки перегляду звернень наведено на рисунку 3.9.

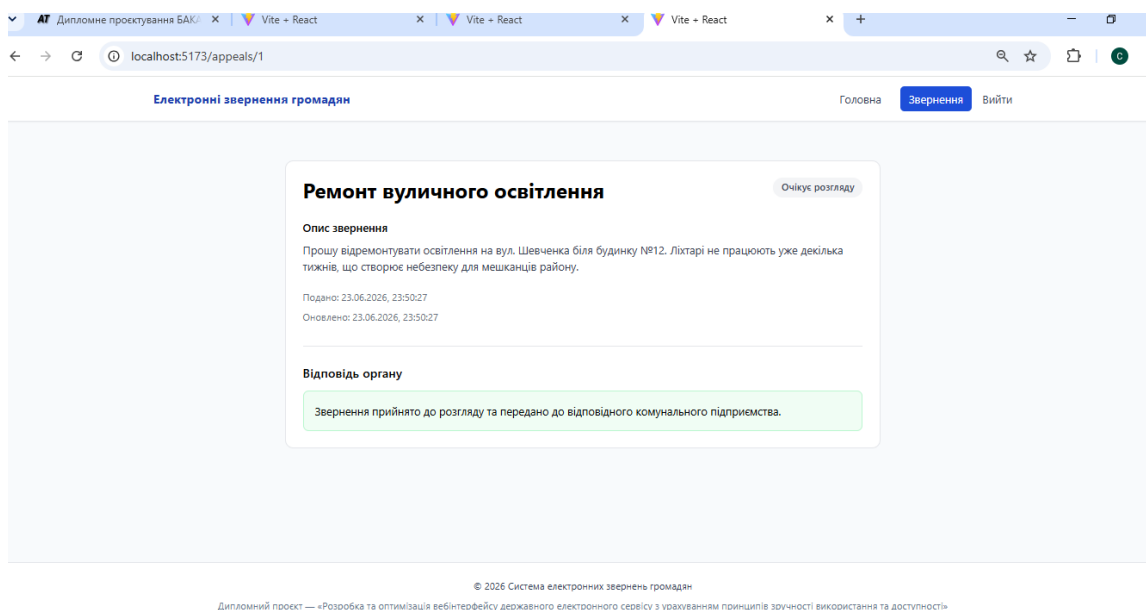


Рисунок 3.10 – Деталі звернення

Сторінка деталей містить повну інформацію про звернення, включаючи тему, опис, поточний статус та часові мітки створення й оновлення.

Таким чином, клієнтська частина повністю покриває основні сценарії роботи користувача із системою електронних звернень та забезпечує зручний доступ до всіх ключових функцій сервісу.

3.4 Реалізація принципів usability та accessibility

Під час розробки вебінтерфейсу окрему увагу було приділено не лише функціональності, а й зручності використання. Оскільки йдеться про державний електронний сервіс, інтерфейс повинен бути зрозумілим для широкого кола користувачів, включаючи людей із різним рівнем цифрових навичок.

під час розробки враховувалися базові принципи usability та окремі рекомендації WCAG 2.1 щодо доступності вебінтерфейсів.

Одним із перших рішень стало використання простої та логічної структури сторінок. Інтерфейс побудований таким чином, щоб користувач міг швидко знайти основні функції без зайвих дій.

Навігація між сторінками зроблена максимально простою. Основні дії - реєстрація, вхід, створення звернення та перегляд звернень - доступні без складних переходів.

Для покращення доступності використано семантичну HTML - розмітку. На сторінках застосовуються заголовки h1 та h2, що формують зрозумілу структуру документа.

Така структура покращує навігацію як для звичайних користувачів, так і для користувачів допоміжних технологій, зокрема програм екранного озвучення.

Усі основні елементи керування підтримують навігацію з клавіатури. Користувач може переходити між кнопками, посиланнями та формами за допомогою клавіші Tab.

Для інтерактивних елементів реалізовано видимі стани фокусу. Це дозволяє легко визначити, який саме елемент зараз активний.

Форми входу, реєстрації та створення звернень також проєктувалися з урахуванням usability. Для полів введення використано label, placeholder та базову HTML - валідацію.

У формах застосовано такі механізми перевірки:

- required;
- minLength;
- type=email;
- type=password.

Це допомагає зменшити кількість помилок під час введення даних.

Для адаптації під різні пристрої використано можливості Tailwind CSS. Інтерфейс коректно відображається як на мобільних пристроях, так і на планшетах та десктопах.

Окремо було перевірено масштабування інтерфейсу до 200%. Під час тестування критичних порушень структури сторінок виявлено не було.

Кольорова схема інтерфейсу підібрана з урахуванням достатньої контрастності між текстом і фоном. Це покращує читабельність контенту та знижує навантаження на користувача.

У частині компонентів також використано ARIA - атрибути, зокрема aria - label для навігаційних елементів.

Загалом реалізовані рішення дозволили створити інтерфейс, який відповідає базовим вимогам usability та демонструє хороший рівень доступності для вебсервісу такого типу.

3.5 Тестування системи та аналіз результатів

Після завершення розробки вебзастосунку було проведено тестування основного функціоналу та оцінку якості реалізованого інтерфейсу. Основною метою тестування була перевірка коректності роботи системи, зручності використання та рівня доступності вебінтерфейсу.

Тестування виконувалося у кілька етапів:

- функціональне тестування;
- тестування доступності;
- аналіз продуктивності.

Функціональне тестування проводилося вручну шляхом перевірки основних сценаріїв роботи користувача із системою.

Було перевірено:

- реєстрацію нового користувача;
- вхід у систему;
- перегляд профілю;
- створення нового звернення;
- перегляд списку звернень;
- перегляд деталей звернення.

У процесі ручного тестування критичних помилок у роботі основного функціоналу виявлено не було. Усі ключові сценарії взаємодії користувача із системою працюють коректно.

Для оцінки технічних характеристик вебзастосунку було використано інструмент Lighthouse, вбудований у Google Chrome DevTools. Результати тестування вебзастосунку за допомогою Lighthouse наведено на рисунку 3.11.

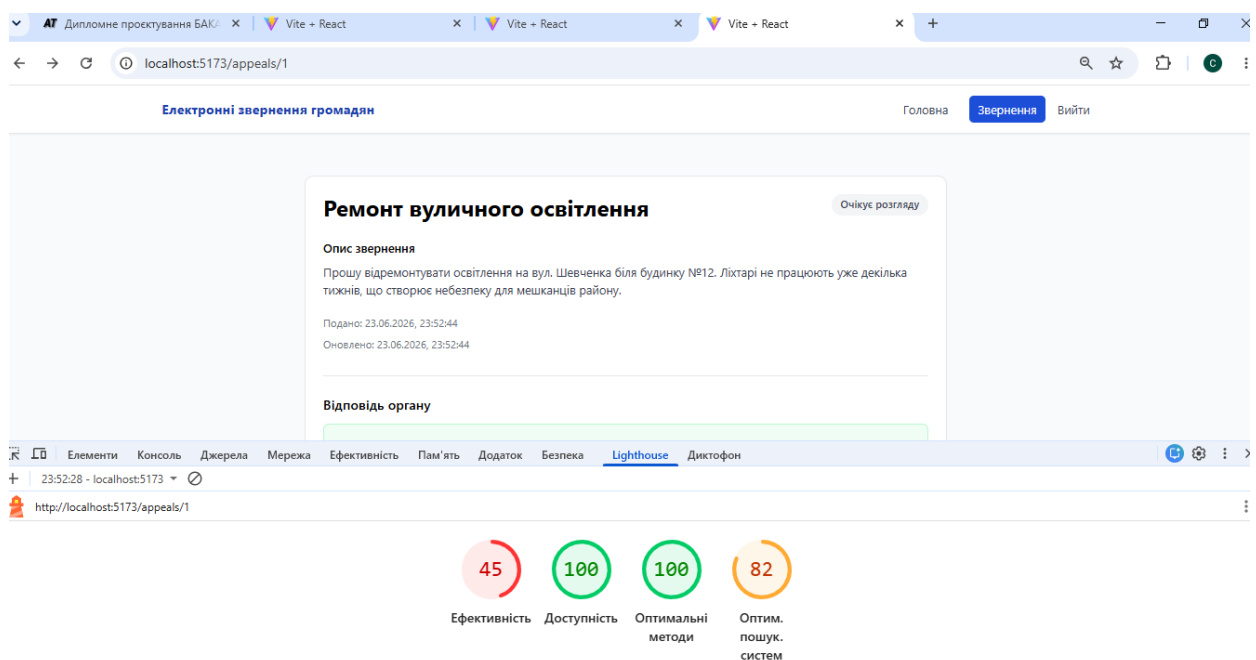


Рисунок 3.11 – Результати тестування Lighthouse

За результатами аудиту отримано такі показники:

- Performance - 45;
- Accessibility - 100;
- Best Practices - 100;
- SEO - 82.

Максимальний результат у категорії Accessibility підтверджує високий рівень доступності інтерфейсу та відповідність основним вимогам accessibility.

Оцінка Best Practices також склала 100 балів, що свідчить про дотримання сучасних практик веброзробки та безпечної взаємодії із застосунком.

Показник SEO дорівнює 82 балам. Основними факторами, які вплинули на цей результат, стали відсутність meta description та некоректна конфігурація robots.txt у локальному середовищі розробки.

Найнижчий показник зафіксовано в категорії Performance. Аналіз результатів показав, що на швидкодію вплинули такі фактори:

- значний обсяг JavaScript - коду;
- невикористовуваний JavaScript;
- підвищене навантаження на main thread;
- запуск застосунку у development - середовищі;
- активні браузерні розширення під час тестування.

Слід зазначити, що тестування проводилося на локальному сервері (localhost), а не у production - збірці. Через це результати Performance є нижчими, ніж очікувано для фінальної версії застосунку.

Для додаткової перевірки доступності інтерфейсу було використано WAVE (Web Accessibility Evaluation Tool). Результати аналізу доступності за допомогою WAVE наведено на рисунку 3.12.

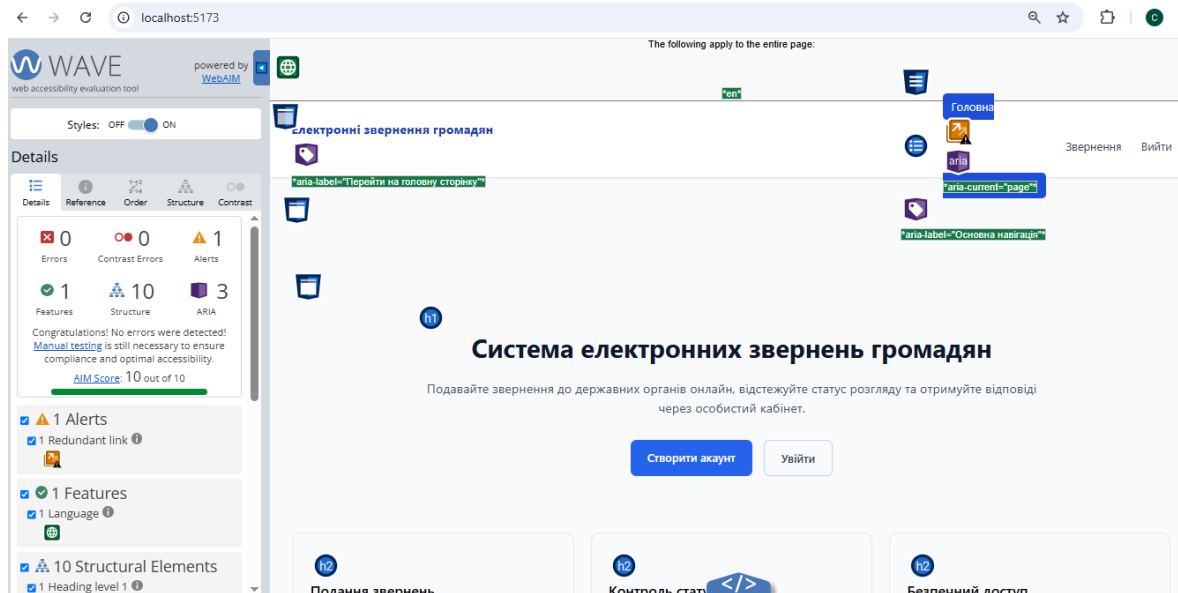


Рисунок 3.12 – Результати тестування WAVE

Результати перевірки WAVE показали:

- Errors - 0;
- Contrast Errors - 0;
- Alerts - 1;
- Structure - 10;
- ARIA - 3.

Відсутність помилок типу Errors та Contrast Errors свідчить про те, що у вебінтерфейсі не виявлено критичних проблем доступності або порушень контрастності.

Єдине попередження стосувалося redundant link, що не є критичною помилкою та не створює суттєвих перешкод для користувача.

Отримані результати тестування підтверджують, що розроблений вебзастосунок успішно виконує основний функціонал, забезпечує стабільну роботу та відповідає сучасним вимогам до usability і accessibility для державних електронних сервісів.

3.6 Висновок до розділу 3

У третьому розділі було розглянуто практичну реалізацію вебзастосунку електронного сервісу звернень громадян, починаючи від проєктування архітектури та завершуючи тестуванням готового програмного продукту.

У межах практичної частини реалізовано клієнтську та серверну частини системи, розроблено структуру бази даних PostgreSQL та впроваджено механізми автентифікації користувачів на основі JWT.

Створений вебзастосунок забезпечує основний цикл роботи користувача із системою: реєстрацію, авторизацію, подання електронних звернень та контроль статусу їх опрацювання.

Особливу увагу під час розробки було приділено реалізації принципів usability та accessibility. Інтерфейс системи адаптовано для роботи на різних пристроях, реалізовано клавіатурну навігацію, видимі focus states та базову підтримку ARIA - атрибутів.

Результати тестування за допомогою Lighthouse та WAVE підтвердили високий рівень доступності та коректність реалізованого інтерфейсу.

Таким чином, поставлені завдання практичної частини роботи виконано, а отримані результати підтверджують ефективність використаних технологій та обраних підходів до розробки сучасного державного електронного сервісу.

РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ ТА ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Безпека життєдіяльності

Під час розробки та використання вебінтерфейсу державного електронного сервісу електронних звернень громадян важливе значення має забезпечення безпеки життєдіяльності користувачів і працівників, які працюють із комп'ютерною технікою. Робота над програмним забезпеченням передбачає тривале перебування за персональним комп'ютером, що може негативно впливати на стан здоров'я людини.

До основних факторів, які можуть негативно впливати на працівника під час роботи з комп'ютерною технікою, належать тривале статичне навантаження, перевтома органів зору, нервово - емоційне напруження та недостатня фізична активність. Тривале перебування у сидячому положенні може призводити до порушень опорно - рухового апарату, болю в шії, спині та попереку.

Для зменшення негативного впливу необхідно правильно організувати робоче місце. Робочий стіл та крісло повинні відповідати антропометричним параметрам користувача. Монітор рекомендується розташовувати на відстані 50–70 см від очей, а його верхню частину – на рівні очей або трохи нижче. Клавіатура та миша повинні бути розміщені таким чином, щоб забезпечувати комфортне положення рук під час роботи.

Важливим фактором збереження здоров'я є дотримання режиму праці та відпочинку. Після кожних 45–60 хвилин роботи рекомендується робити короткі перерви тривалістю 5–10 хвилин. Під час перерв доцільно виконувати вправи для очей та нескладні фізичні вправи для зменшення статичного навантаження на організм.

Особливу увагу необхідно приділяти умовам освітлення робочого місця. Освітлення повинно бути достатнім та рівномірним, без різких контрастів і

відблисків на екрані монітора. Недостатнє або надмірне освітлення може спричиняти швидку втому очей та зниження працездатності.

Під час роботи із системою електронних звернень важливим аспектом є також інформаційна безпека користувачів. Оскільки система працює з персональними даними громадян, необхідно забезпечити захист інформації від несанкціонованого доступу. Для цього використовуються механізми автентифікації користувачів, контроль доступу до даних та захищений обмін інформацією між клієнтською і серверною частинами системи.

Таким чином, дотримання вимог безпеки життєдіяльності під час роботи з комп'ютерною технікою сприяє збереженню здоров'я працівників, підвищенню продуктивності праці та забезпечує безпечне використання інформаційних систем.

4.2 Основи охорони праці

Охорона праці є важливою складовою організації робочого процесу під час розробки програмного забезпечення. Основною метою охорони праці є створення безпечних і комфортних умов роботи, що дозволяють мінімізувати ризики виникнення професійних захворювань та нещасних випадків.

Робоче місце розробника програмного забезпечення повинно відповідати санітарно - гігієнічним вимогам. Площа приміщення, рівень освітлення, температура та вологість повітря повинні забезпечувати комфортні умови для тривалої роботи за комп'ютером. Рекомендована температура повітря в приміщенні становить від 18 до 24 °С, а відносна вологість – від 40 до 60 %.

Особлива увага приділяється електробезпеці. Усі технічні засоби повинні бути справними та відповідати встановленим вимогам безпеки. Комп'ютери, монітори та інше обладнання необхідно підключати до електромережі через справні мережеві фільтри або джерела безперебійного живлення. Забороняється використовувати пошкоджені кабелі або несправне обладнання.

Під час експлуатації комп'ютерної техніки необхідно дотримуватися правил пожежної безпеки. У приміщенні повинні бути наявні первинні засоби пожежогасіння, а працівники повинні знати порядок дій у разі виникнення аварійної ситуації. Не допускається перевантаження електромережі або використання несправних електроприладів.

Для зменшення професійної втоми необхідно раціонально організувати режим праці та відпочинку. Виконання короткочасних перерв протягом робочого дня сприяє зниженню навантаження на зір та опорно - руховий апарат.

Під час розробки вебінтерфейсу державного електронного сервісу електронних звернень громадян використовувалися сучасні програмні засоби та технології, що забезпечують стабільність роботи програмного забезпечення та сприяють підвищенню надійності обробки даних. Використання сучасних інструментів розробки дозволяє зменшити ризик виникнення програмних помилок та втрати інформації.

Отже, дотримання вимог охорони праці під час виконання робіт з розробки програмного забезпечення створює безпечні умови праці, сприяє збереженню здоров'я працівників та підвищує ефективність виконання професійних обов'язків.

4.3 Висновок до розділу 4

У четвертому розділі було розглянуто питання безпеки життєдіяльності та охорони праці під час розробки й використання веборієнтованих інформаційних систем. Проведено аналіз основних факторів, які можуть негативно впливати на здоров'я користувачів та розробників під час роботи з комп'ютерною технікою, зокрема навантаження на органи зору, тривале перебування у сидячому положенні та нервово - емоційне напруження.

Розглянуто основні вимоги до організації безпечного робочого місця, дотримання режиму праці та відпочинку, забезпечення належних умов освітлення, мікроклімату приміщення та електробезпеки. Також проаналізовано

заходи пожежної безпеки та рекомендації щодо безпечної експлуатації комп'ютерного обладнання.

У результаті встановлено, що дотримання вимог безпеки життєдіяльності та охорони праці сприяє збереженню здоров'я працівників, підвищенню продуктивності праці та створенню безпечних умов під час розробки й експлуатації інформаційних систем. Отже, поставлені завдання розділу виконано, а розглянуті заходи можуть бути використані для організації безпечного робочого середовища під час роботи з програмним забезпеченням та комп'ютерною технікою.

ВИСНОВКИ

У кваліфікаційній роботі розглянуто питання створення вебінтерфейсу державного електронного сервісу для подання та опрацювання звернень громадян. У ході виконання роботи було проаналізовано особливості функціонування електронних державних сервісів, вимоги до їх зручності використання, доступності та інформаційної безпеки.

На основі проведеного аналізу сформовано функціональні вимоги до програмного продукту та обґрунтовано вибір технологій для його реалізації. Для побудови клієнтської частини застосовано React, для реалізації серверної логіки - Node.js та Express.js, а для зберігання даних використано систему керування базами даних PostgreSQL. Такий набір технологій дозволив створити цілісне рішення з чітким розподілом функцій між окремими компонентами системи.

У процесі роботи виконано проєктування архітектури вебзастосунку, структури бази даних та механізмів взаємодії між клієнтською і серверною частинами. Реалізований вебсервіс підтримує реєстрацію користувачів, автентифікацію за допомогою JWT, створення електронних звернень, перегляд історії поданих заяв і контроль поточного статусу їх розгляду.

Особливу увагу приділено питанням usability та accessibility. Під час розробки інтерфейсу враховувалися рекомендації WCAG 2.1, що дало змогу забезпечити зрозумілу навігацію, адаптивність інтерфейсу та можливість комфортної роботи із сервісом на різних типах пристроїв.

Проведене тестування підтвердило коректність функціонування основних компонентів системи. Перевірка показала, що реалізовані сценарії взаємодії користувача із сервісом виконуються без порушення логіки роботи застосунку, а обмін даними між його складовими здійснюється стабільно та передбачувано.

Отримані результати підтверджують доцільність використання сучасних вебтехнологій для цифровізації процесу подання звернень громадян.

ПЕРЕЛІК ДЖЕРЕЛ

1. ISO 9241 - 11:2018. Ergonomics of human - system interaction - Usability: Definitions and concepts. 2018.
2. Web Content Accessibility Guidelines (WCAG) 2.1. W3C. URL: <https://www.w3.org/TR/WCAG21/> (дата звернення: 25.05.2026).
3. Дія : офіційний портал. URL: <https://diia.gov.ua/> (дата звернення: 25.05.2026).
4. React Documentation. URL: <https://react.dev/> (дата звернення: 25.05.2026).
5. Node.js Documentation. URL: <https://nodejs.org/en/docs> (дата звернення: 25.05.2026).
6. PostgreSQL Documentation. URL: <https://www.postgresql.org/docs/> (дата звернення: 25.05.2026).
7. Матеріали з електронного урядування та цифрової трансформації державних послуг України.
8. Nielsen Norman Group. Usability 101: Introduction to Usability. URL: <https://www.nngroup.com/articles/usability-101-introduction-to-usability/> (дата звернення: 25.05.2026).
9. Accessibility Resources. W3C Web Accessibility Initiative (WAI). URL: <https://www.w3.org/WAI/> (дата звернення: 25.05.2026).
10. Небесний Р. М. Рекомендаційна система формування команд виконавців з відповідними фаховими компетентностями : дис. ... д - ра філософії. Тернопіль, 2023. URL: <https://elartu.tntu.edu.ua/handle/lib/43005> .
11. Nebesnyi R., Palka O., Dmytrotsa L., Kozbur H. Smart people: the role of big data analytics in digital transformation // БАІТmp 2025. 2026. Vol. 2. P. 163–174. URL: <https://ceur-ws.org/Vol-4159/paper14.pdf> .
12. Vaskiv R., Veretennikova N., Nebesnyi R., Bilovus H., Zhovnir Y. Formation of an IT Project Team by Analogy with a Flock // 2024 IEEE 19th

International Conference on Computer Science and Information Technologies (CSIT). 2024. P. 1–4. DOI: 10.1109/CSIT65290.2024.10982684.

13. OWASP Top 10 Web Application Security Risks. OWASP Foundation. URL: <https://owasp.org/www-project-top-ten/> (дата звернення: 25.05.2026).

14. Cross Site Scripting Prevention Cheat Sheet. OWASP Foundation. URL: <https://cheatsheetseries.owasp.org/> (дата звернення: 25.05.2026).

15. Authentication Cheat Sheet. OWASP Foundation. URL: <https://cheatsheetseries.owasp.org/> (дата звернення: 25.05.2026).

16. HTML Documentation. Mozilla Developer Network. URL: <https://developer.mozilla.org/en-US/docs/Web/HTML> (дата звернення: 25.05.2026).

17. CSS Documentation. Mozilla Developer Network. URL: <https://developer.mozilla.org/en-US/docs/Web/CSS> (дата звернення: 25.05.2026).

18. JavaScript Documentation. Mozilla Developer Network. URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> (дата звернення: 25.05.2026).

19. React Router Documentation. URL: <https://reactrouter.com/> (дата звернення: 25.05.2026).

20. Tailwind CSS Documentation. URL: <https://tailwindcss.com/docs> (дата звернення: 25.05.2026).

21. Vite Documentation. URL: <https://vitejs.dev/guide/> (дата звернення: 25.06.2026).

22. Express.js Documentation. URL: <https://expressjs.com/> (дата звернення: 25.05.2026).

23. PostgreSQL Tutorial. PostgreSQL Global Development Group. URL: <https://www.postgresql.org/docs/current/tutorial.html> (дата звернення: 25.05.2026).

24. ECMAScript Language Specification. URL: <https://tc39.es/ecma262/> (дата звернення: 25.05.2026).

25. HTML Living Standard. WHATWG. URL: <https://html.spec.whatwg.org/> (дата звернення: 25.05.2026).

26. Accessible Rich Internet Applications (WAI - ARIA). W3C. URL: <https://www.w3.org/WAI/standards-guidelines/aria/> (дата звернення: 25.05.2026).
27. Web Fundamentals. Google Developers. URL: <https://web.dev/> (дата звернення: 25.05.2026).
28. Nebesnyi R., Kunanets N., Vaskiv R., Veretennikova N. Formation of an IT Project Team in the Context of PMBOK Requirements // 2021 IEEE 16th International Conference on Computer Sciences and Information Technologies. 2021. Vol. 2. P. 431–436. DOI: 10.1109/CSIT52704.2021.9615291.
29. ASP.NET and Web Accessibility Guidance. Microsoft Learn. URL: <https://learn.microsoft.com/> (дата звернення: 25.05.2026).
30. State of JS Survey 2024. JavaScript Ecosystem Report. URL: <https://stateofjs.com/> (дата звернення: 25.05.2026).
31. Nebesnyi R., Kunanets N., Veretennikova N. et al. Portfolio project management // ITPM. 2024. P. 141–152. URL: <https://ceur-ws.org/Vol-3709/paper12.pdf> (дата звернення: 17.05.2026).
32. User Experience and Accessibility Guidelines. Digital.gov. URL: <https://digital.gov/> (дата звернення: 25.05.2026).
33. United Nations E - Government Survey 2024. United Nations Publications, 2024.
34. Шкідливий вплив персонального комп'ютера на здоров'я людини. URL: <https://ir.lib.vntu.edu.ua/bitstream/handle/123456789/21370/5363.pdf> (дата звернення: 25.05.2026).
35. Желібо Є. П., Сагайдак І. С. Безпека життєдіяльності : навч. посіб. Київ : ЕКОМЕН, 2011. 200 с.