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 вебплатформи для взаємодії громади з місцевою владою

Виконав: студент IV курсу, групи СН-41
спеціальності 122 Комп’ютерні науки
(шифр і назва спеціальності)

_____ Порохонько Т. А.
(підпис) (прізвище та ініціали)

Керівник _____ Небесний Р. М.
(підпис) (прізвище та ініціали)

Нормоконтроль _____ _____
(підпис) (прізвище та ініціали)

Завідувач кафедри _____ Боднарчук І.О.
(підпис) (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)
«__» _червня_ 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 Порохонько Тетяні Андріївні
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка вебплатформи для взаємодії громади з місцевою владою

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* 20*26* року № *4/9-239*

2. Термін подання студентом завершеної роботи 23 червня 2026 р.

3. Вихідні дані до роботи Нормативно-правова база України у сфері GovTech, технічні специфікації серверного середовища Apache і СКБД MySQL, а також літературні та інтернет-джерела за темою клієнт-серверної розробки вебплатформ.

4. Зміст роботи (перелік питань, які потрібно розробити)
Вступ. 1. Аналітичний огляд та постановка завдання розробки вебплатформи. 2. Проєктування архітектури та структури даних вебплатформи. 3. Програмна реалізація та верифікація ефективності системи. 4. Безпека життєдіяльності, основи охорони праці. Висновки. Перелік джерел. Додатки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
1. Титульний слайд кваліфікаційної роботи. 2. Актуальність теми та проблеми цифровізації муніципального менеджменту. 3. Мета та завдання бакалаврського дослідження. 4. Порівняльний аналіз чинних GovTech-аналогів системи. 5. Нормативно-правове регулювання та етапи життєвого циклу звернень. 6. Функціональні вимоги та схема розподілу прецедентів (UML Use Case). 7. Архітектура системи за шаблоном MVC та діаграма компонентів. 8. Фізична структура та ER-діаграма реляційної бази даних у MySQL. 9. Інформаційна архітектура та структурна схема UX-роутингу. 10. Програмна реалізація серверної логіки Backend та підсистеми валідації. 11. Реалізація клієнтських сервісів та асинхронного голосування. 12. Програмний лістинг транзакційного механізму фіксації голосів. 13. Верифікація сценаріїв авторизації та тестування методом «чорної скриньки». 14. Оцінка технічної та експлуатаційної ефективності впровадження платформи. 15. Загальні висновки за результатами виконання кваліфікаційної роботи.

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	
2.	Підбір та опрацювання літературних джерел по темі кваліфікаційної роботи	27.01.2026-16.02.2026	
3.	ДослідженняGovTech-аналогів та формування вимог до вебплатформи	17.02.2026-10.05.2026	
4.	Оформлення розділу «Аналітичний огляд та постановка завдання розробки вебплатформи»	11.05.2026-17.05.2026	
5.	Оформлення розділу «Проєктування архітектури та структури даних вебплатформи»	18.05.2026-24.05.2026	
6.	Оформлення розділу «Програмна реалізація та верифікація ефективності системи»	25.05.2026-31.05.2026	
7.	Виконання завдання до підрозділу «Безпека життєдіяльності»	01.06.2026-08.06.2026	
8.	Виконання завдання до підрозділу «Основи охорони	01.06.2026-08.06.2026	
	праці»		
9.	Оформлення кваліфікаційної роботи	09.06.2026-11.06.2026	
10.	Нормоконтроль	16.06.2026-.07.2026	
11.	Перевірка на плагіат	18.06.2026	
12.	Попередній захист кваліфікаційної роботи	22.06.2026	
13.	Захист кваліфікаційної роботи	24.06.2026	

Студент _____ Порохонько Т. А.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Небесний Р. М.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Розробка вебплатформи для взаємодії громади з місцевою владою // Кваліфікаційна робота освітнього ступеня «Бакалавр» // Порохонько Тетяна Андріївна // 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. 82, рис. – 17, табл. – 2, додат. – 19, бібліогр. – 41.

Ключові слова: електронна демократія, вебплатформа, електронні петиції, муніципальний менеджмент, технологія ajax, транзакції бази даних, архітектура mvc.

Кваліфікаційна робота присвячена дослідженню та розробці вебплатформи для взаємодії громади з місцевою владою.

В першому розділі розглянуто концепцію електронного урядування в Україні, проведено порівняльний аналіз наявних GovTech рішень («e-Dem», «СВОЇ») та сформовано функціональні й нефункціональні технічні вимоги до системи відповідно до Закону України «Про звернення громадян».

В другому розділі обґрунтовано вибір методології Agile та клієнт-серверної архітектури MVC. Побудовано UML-діаграми класів і діяльності, спроєктовано реляційну модель бази даних у СКБД MySQL із захистом від повторного голосування. Розроблено UI/UX-макети інтерфейсу на базі шрифту Inter та контрастної палітри кольорів.

В третьому розділі описано програмну реалізацію серверної частини мовою PHP (інтерфейс PDO) та клієнтських сервісів із використанням технології AJAX і атомарних транзакцій. Проведено тестування системи методом «чорної скриньки» та її розгортання в серверному середовищі InfinityFree.

У розділі «Безпека життєдіяльності, основи охорони праці» визначено алгоритм дій у разі пожежі в серверній та сформульовано вимоги до ергономічної організації робочого місця веброзробника.

Об’єкт дослідження: процеси інформаційно-комунікаційної взаємодії між мешканцями територіальної громади та органами місцевого самоврядування.

Предмет дослідження: методи, моделі, алгоритми та програмні засоби створення та адміністрування адаптивної вебплатформи для взаємодії громади з місцевою владою.

ANNOTATION

Development of a Web Platform for Community Interaction with Local Authorities // Qualification work of the educational level «Bachelor» // Porokhonko Tetiana Andriivna // Ternopil Ivan Pulyu National Technical University, Computer and Information Systems and Software Engineering Faculty, Computer Sciences Department, group SN-41 // Ternopil, 2026 // P. 82, fig. – 17, tabl. – 2 , annexes. – 19, references – 41.

Keywords: electronic democracy, web platform, e-petitions, municipal management, ajax technology, database transactions, mvc architecture, inter font.

The qualification work is dedicated to the research and development of a web platform for community interaction with local authorities.

The goal of the work is to automate and improve the quality of G2C interaction channels within a territorial community by implementing an adaptive, high-performance, and secure software solution.

The first section considers the concept of e-government in Ukraine, performs a comparative analysis of existing GovTech solutions ("e-Dem", "СВОЇ"), and establishes functional and non-functional requirements in compliance with the Law of Ukraine "On Citizens' Appeals".

The second section justifies the choice of Agile methodology and client-server MVC architecture. UML class and activity diagrams are built, and a relational database model in MySQL is designed with protection against duplicate voting. UI/UX interface mockups based on the Inter font and a contrast color palette are developed.

The third section describes the software implementation of the Backend using PHP (PDO interface) and client services utilizing AJAX technology and atomic database transactions. Black-box testing of the platform and its deployment in the InfinityFree server environment are conducted.

In the section "Life Safety and Fundamentals of Occupational Health", the algorithm of actions in case of a fire in the server room is determined, and the

requirements for the ergonomic organization of a web developer's workplace are formulated.

Object of research: processes of information and communication interaction between residents of a territorial community and local government authorities.

Subject of research: methods, models, algorithms, and software tools for creating and administering an adaptive web platform for community interaction with local authorities.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

AJAX (англ. Asynchronous JavaScript and XML) – технологія асинхронного обміну даними між веб-браузером і веб-сервером без перезавантаження сторінки.

API (англ. Application Programming Interface) – інтерфейс прикладного програмування, набір готових функцій і класів для взаємодії між різними програмними компонентами.

CRUD (англ. Create, Read, Update, Delete) – базові операції роботи з даними: створення, читання, оновлення, видалення.

CSS3 (англ. Cascading Style Sheets 3) – каскадні таблиці стилів третього покоління, що використовуються для візуального оформлення вебсторінок.

DSN (англ. Data Source Name) – рядок конфігурації, що містить вихідні дані (хост, назву бази даних, кодування) для підключення до сервера БД.

FTP (англ. File Transfer Protocol) – мережевий протокол передачі файлів між локальним комп'ютером розробника та віддаленим сервером хостингу.

G2C (англ. Government-to-Citizen) – сектор та модель електронного урядування, що забезпечує інформаційну взаємодію між органами влади та громадянами.

GovTech (англ. Government Technology) – технологічний сектор, спрямований на цифровізацію та модернізацію державних послуг і муніципального менеджменту.

HTML5 (англ. HyperText Markup Language 5) – п'ята версія стандарту мови розмітки гіпертексту, що використовується для побудови структури вебсторінок.

JSON (англ. JavaScript Object Notation) – текстовий формат обміну даними між сервером і клієнтом, що базується на структурі «ключ – значення».

MVC (англ. Model-View-Controller) – архітектурний шаблон проєктування програмного забезпечення, що розділяє бізнес-логіку, інтерфейс та керування.

PDO (англ. PHP Data Objects) – вбудоване розширення для мови PHP, що надає універсальний об'єктно-орієнтований інтерфейс для безпечної роботи з базами даних.

PHP (англ. Hypertext Preprocessor) – скриптова мова програмування загального призначення, що використовується для розробки серверної частини вебплатформ.

SQL (англ. Structured Query Language) – декларативна мова структурованих запитів, що використовується для керування та обробки даних у реляційних СКБД.

UI (англ. User Interface) – користувацький інтерфейс, сукупність графічних елементів оформлення та зовнішнього вигляду вебсторінок.

UML (англ. Unified Modeling Language) – уніфікована мова графічного моделювання для об'єктно-орієнтованого проєктування програмних систем.

UX (англ. User Experience) – користувацький досвід, що визначає логіку взаємодії, навігацію, ергономічність та зручність роботи з вебплатформою.

БД – База даних.

БЖД – Безпека життєдіяльності.

ДСанПіН – Державні санітарні правила та норми.

НПАОП – Нормативно-правовий акт з охорони праці.

ПІБ – Прізвище, ім'я, по батькові.

СКБД – Система керування базами даних.

ЗМІСТ

ВСТУП

Актуальність теми. Розвиток сучасного демократичного суспільства та реформа децентралізації в Україні ставлять нові вимоги до якості, прозорості та оперативності взаємодії між органами місцевого самоврядування та мешканцями територіальних громад. В умовах стрімкої цифровізації державного управління за вектором «влада – громадянин» (G2C) традиційні паперові процедури розгляду звернень стають неефективними, оскільки вони створюють додаткові бюрократичні бар'єри та сповільнюють процеси прийняття рішень.

Міністерство цифрової трансформації України визначає впровадження інструментів електронної демократії (e-Democracy) муніципального менеджменту як обов'язкову умову для забезпечення відкритості влади. Проте на рівні багатьох громад досі спостерігається проблема роздробленості цифрових сервісів, коли стрічки новин, форми для скарг та опитування розміщуються на різних інтернет-ресурсах, а системи збору підписів під петиціями не мають надійного захисту від штучних накруток результатів ботами. Таким чином, розробка єдиної, адаптивної та захищеної вебплатформи, що інтегрує інформаційні послуги, премодерацію контенту та суворі інструменти асинхронного голосування, є надзвичайно актуальним науково-практичним напрямком.

Мета і задачі дослідження. Метою кваліфікаційної роботи є підвищення якості надання муніципальних послуг, усунення бюрократичних затримок та забезпечення прозорого двостороннього зв'язку між мешканцями й міською радою шляхом проєктування, програмної реалізації та розгортання єдиної вебплатформи для взаємодії громади з місцевою владою.

Для досягнення поставленої мети необхідно виконати такі завдання:

- Проаналізувати стан досліджень у сфері електронного урядування в Україні та виконати порівняльний аналіз чинних GovTech аналогів на ринку;

- Дослідити нормативно-правове регулювання механізмів подання цифрових запитів відповідно до чинного законодавства та сформувати технічні вимоги до системи;
- Спроєктувати трирівневу архітектуру вебплатформи на основі шаблону MVC та виконати об'єктно-орієнтоване моделювання за допомогою мови UML;
- Розробити логічну та фізичну структуру реляційної бази даних у СКБД MySQL із впровадженням захисту лічильників від фальсифікації;
- Спроектувати UX/UI-інтерфейс платформи з урахуванням сучасних стандартів юзабіліті, типографіки Inter та фірмової колірної палітри;
- Реалізувати серверну частину (Backend) мовою PHP PDO та клієнтські сервіси (Frontend) з використанням асинхронної технології AJAX;
- Провести функціональне тестування розробленого програмного комплексу методом «чорної скриньки» та здійснити його розгортання в хмарному серверному середовищі;
- Розглянути заходи безпеки життєдіяльності та вимоги охорони праці під час супроводу серверної інфраструктури та організації робочого місця веброзробника.

Практичне значення одержаних результатів полягає у створенні готового до експлуатації та масштабування програмного продукту для GovTech сектору. Розроблені модулі забезпечують суворий захист від повторного голосування на рівні транзакцій СКБД, автоматизують життєвий цикл розгляду електронних звернень (від кроку створення до модерації й публікації результату), а також надають зручний центр зворотного зв'язку та інструменти асинхронного голосування. Система оптимізує час обробки інформації та може бути безперешкодно впроваджена в роботу будь-якої територіальної громади України для підвищення рівня довіри населення до місцевої влади.

РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД ТА ПОСТАНОВКА ЗАВДАННЯ РОЗРОБКИ ВЕБПЛАТФОРМИ

1.1 Концепція електронного урядування та цифровізації муніципального менеджменту в Україні

Розвиток державного управління на рівні місцевого самоврядування в Україні нерозривно пов'язаний із процесами цифровізації. Перехід від традиційних паперових процедур до використання сучасних інформаційних технологій дозволяє підвищити якість надання послуг, зробити роботу органів влади прозорою та забезпечити оперативний розгляд звернень громадян [4, 5]. Міністерство цифрової трансформації України визначає електронне урядування та цифровізацію муніципального менеджменту як обов'язкові умови для створення ефективної взаємодії між керівництвом міст чи громад та їхніми мешканцями.

Головним завданням у цій сфері є побудова зручних каналів комунікації типу «влада – громадянин» (G2C). Основним інструментом для цього виступає спеціалізована вебплатформа для взаємодії громади з місцевою владою. Муніципальний менеджмент в умовах інформаційного суспільства потребує впровадження інструментів, які дозволяють людям впливати на рішення органів влади в режимі реального часу, не виходячи з дому. Це значно скорочує час на обробку даних та усуває бюрократичні затримки [8].

Державні стратегії підтримують розвиток сервісів електронної участі. Проте на рівні багатьох територіальних громад досі є проблема роздробленості таких інструментів. Часто електронні опитування, форми для скарг та інформаційні повідомлення розміщуються на різних вебсайтах. Єдина вебплатформа для взаємодії громади з місцевою владою дозволяє об'єднати цей функціонал у межах одного захищеного ресурсу. Громадяни отримують можливість подавати цифрові запити, брати участь у голосуваннях та відстежувати розгляд власних ініціатив. Для влади це спрощує контроль за

роботою виконавчих органів, автоматизує маршрутизацію документів та підвищує рівень довіри з боку населення.

1.2 Порівняльний аналіз існуючих вебплатформ та сервісів для взаємодії органів влади з громадою

Перед початком безпосереднього проєктування архітектури бази даних, внутрішньої логіки модулів та інтерфейсу користувача необхідно детально проаналізувати існуючі аналогічні GovTech-рішення на вітчизняному ринку. Це дозволить врахувати накопичений практичний досвід, уникнути типових інженерних помилок та створити оптимізований програмний продукт для муніципального менеджменту. На сьогодні в Україні найбільш поширеними централізованими системами електронної демократії та участі є всеукраїнська платформа «e-Dem» [6] та інтерактивна муніципальна платформа «СВОЇ» [7].

Для визначення оптимального функціонального наповнення та архітектурних особливостей майбутнього програмного забезпечення було проведено комплексний порівняльний аналіз зазначених наявних GovTech-систем. Скорочений порівняльний аналіз розроблюваної системи з найближчими аналогами наведено в таблиці 1.1.

Таблиця 1.1 – Порівняльний аналіз вебплатформ електронної демократії

Критерії порівняння	**Платформа «e-Dem»**	**Платформа «СВОЇ» (EGAP)**	**Розроблювана вебплатформа**
1	**2**	**3**	**4**
Доступність коду	Закритий (державний)	Закритий (комерційний)	Закритий (хмарне SaaS-рішення)

Продовження таблиці 1.1

1	2	3	4
Формат взаємодії з користувачем	Складний вебпортал зі значною кількістю вкладок	Переважно чат-боти у месенджерах (Viber, Telegram) + вебвітрина	Вебплатформа з мінімалістичним адаптивним UI/UX-дизайном
Авторизація	Складна (через КЕП)	Спрощена (верифікація через тел. номер / месенджер)	Спрощена (Email, безпечне хешування паролів)
Захист від фальсифікації голосів	Централізовані реєстри	Ідентифікація за API месенджерів та номером телефону	Строгі транзакції на рівні бази даних MySQL
Модерація контенту	Тривала перевірка	Опціональна модерація на рівні шаблонів	Обов'язкова (автоматична черга петицій)

Детальний аналіз системи «e-Dem» показав, що вона містить потужні інструменти для збору електронних петицій, реалізації бюджетів участі та проведення консультацій з громадськістю (Див. рисунок 1.1). Вона повністю відповідає букві закону, проте її користувацький інтерфейс є досить перевантаженим і складним для мешканців територіальних громад із низьким або середнім рівнем комп'ютерної грамотності. Крім того, виключно строга авторизація за допомогою кваліфікованого електронного підпису (КЕП)

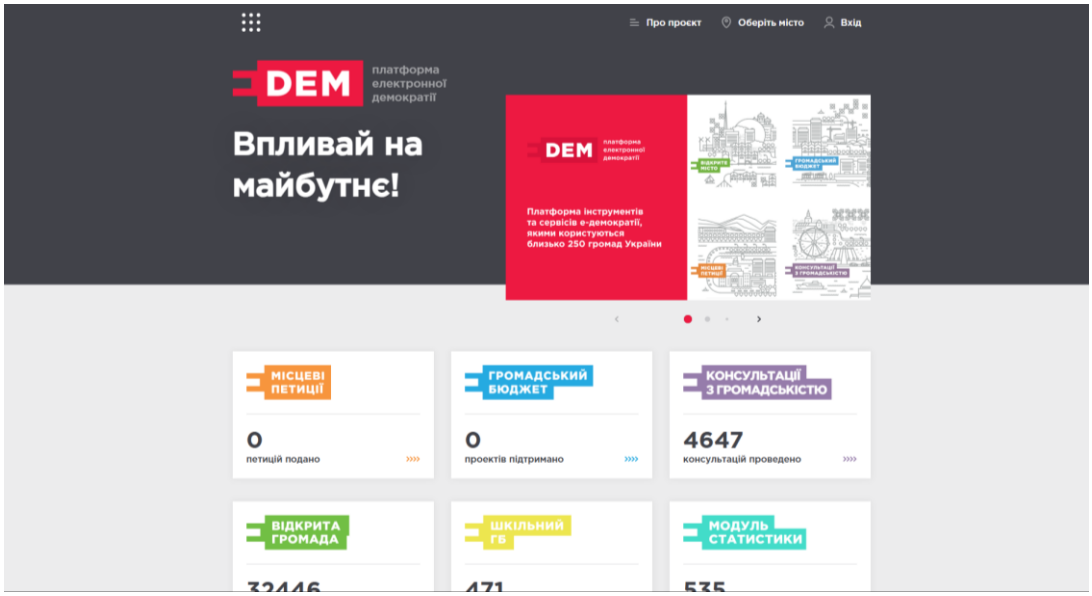

Рисунок 1.1 – Головна сторінка платформи «e-Dem»

або інтеграції з BankID часто створює високий поріг входження, що суттєво знижує громадянську активність населення на місцевому рівні.

Муніципальна платформа «СВОЇ» орієнтована на концепцію швидкого мобільного інформування та взаємодії, через що її основним інструментом виступають чат-боти у популярних месенджерах Telegram та Viber (Див. рисунок 1.2).

Рисунок 1.2 – Головна сторінка платформи «СВОЇ»

Проте такий підхід суттєво обмежує можливості користувачів під час перегляду великих обсягів текстових даних, вивчення детального опису проблем у петиціях чи аналізу муніципальних архівів новин, оскільки інтерфейси месенджерів не пристосовані для відображення складних таблиць, фільтрів та індикаторів прогресу. Більше того, «СВОЇ» функціонує як закрита хмарна комерційно-донорська платформа, що унеможливлює проведення локального аудиту безпеки сховища даних конкретної громади.

Розроблювана вебплатформа для взаємодії громади з місцевою владою покликана ефективно вирішити виявлені проблеми аналогів. У ній оптимізовано користувацький досвід завдяки розробці лаконічного двоколонкового вебпланування з бічною системою фільтрації. Для усунення бар'єрів авторизації впроваджено безпечний вхід за паролем із шифруванням. Надійною технічною альтернативою складним зовнішнім реєстрам виступає архітектурне рішення на рівні СКБД MySQL: використання унікального складеного первинного ключа PRIMARY KEY (user_id, petition_id) разом із атомарними PHP-транзакціями повністю блокує будь-які спроби повторного чи фіктивного голосування, гарантуючи абсолютну точність збору підписів за мінімальних серверних затримок.

1.3 Дослідження нормативно-правового регулювання та механізмів подання цифрових запитів і петицій

Бізнес-логіка, яку реалізує вебплатформа для взаємодії громади з місцевою владою, повинна повністю відповідати законам України. Основним документом є Закон України «Про звернення громадян» [1], який чітко визначає статус електронної петиції як офіційного колективного звернення. Місцева влада зобов'язана розглянути петицію, якщо вона набрала визначену кількість голосів у встановлений строк.

Закон встановлює такі обов'язкові етапи:

- реєстрація та авторизація автора звернення;

- заповнення форми та відправка тексту на розгляд;
- попередня перевірка (модерація) контенту уповноваженим працівником влади;
- оприлюднення петиції на сайті та старт збору підписів;
- фіксація результатів голосування та надання офіційної відповіді.

Вебплатформа для взаємодії громади з місцевою владою автоматизує цей життєвий цикл. Особлива увага приділяється етапу премодерації. Код системи розроблено так, що після створення автором петиція не публікується автоматично, а потрапляє в чергу до кабінету модератора. Це захищає ресурс від спаму, нецензурного вмісту та порушень закону. Також система суворо дотримується Закону України «Про захист персональних даних» [2, 3], забезпечуючи безпечне збереження інформації про користувачів та їхні голоси.

1.4 Обґрунтування доцільності створення та формування технічних вимог до муніципальної вебплатформи

Створення локального програмного продукту для конкретної громади є повністю доцільним. Це дозволяє уникнути зайвих функцій, які є у великих універсальних системах, та адаптувати інтерфейс під потреби місцевих жителів і чиновників. На основі аналізу процесів комунікації було сформовано технічні вимоги до системи, які розділено на функціональні та нефункціональні.

Функціональні вимоги:

- *Реєстрація та вхід*. Система повинна підтримувати створення облікових записів з перевіркою введених даних.
- *Робота з петиціями*. Користувач повинен мати можливість створити петицію, додати заголовок та детальний опис проблеми.
- *Голосування*. Реалізація функції підтримки ініціативи (кнопка «Підписати»), яка стає неактивною після віддання голосу для унеможливлення накруток.

- *Управління та модерація*. Наявність кабінету модератора для схвалення або відхилення запитів та панелі адміністратора для публікації новин і перегляду скарг.

Сформовані технічні вимоги дозволяють чітко розмежувати права доступу та визначити характер взаємодії користувачів із системою. Взаємодія основних ролей користувачів із програмним комплексом візуалізується за допомогою діаграми прецедентів. Оптимальна схема розподілу функціональних можливостей між усіма учасниками процесу наведена на рисунку 1.3.

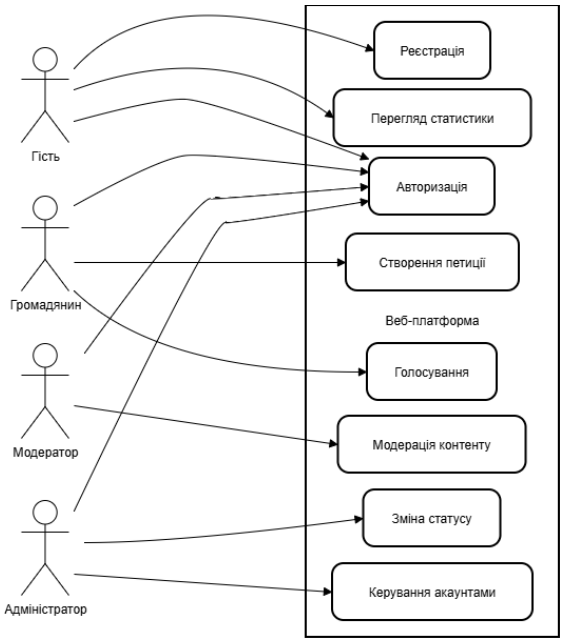

Рисунок 1.3 – Схема розподілу функціональних можливостей

Для детального опису кожного прецеденту, зображеного на діаграмі, та визначення вхідних умов для подальшого алгоритмічного проєктування, розроблено відповідний структурований перелік. Повний реєстр варіантів використання для кожної визначеної ролі представлено в таблиці 1.2.

Таблиця 1.2 – Реєстр варіантів використання

Актор	**Назва варіанту використання**	**Опис**
Гість	Реєстрація в системі	Створення облікового запису з введенням даних.
	Перегляд статистики	Аналіз популярності петицій.
Всі	Авторизація	Вхід у систему за логіном та паролем.
Громадянин	Створення петиції	Заповнення форми ініціативи та відправка.
	Голосування за петицію	Підтримка ініціативи власним голосом.
Модератор	Модерація контенту	Перевірка тексту та публікація на сайті. Верифікація користувачів.
	Зміна статусу запиту	Фіксація рішення влади щодо петиції.
Адміністратор	Керування користувачами	Блокування, видалення або зміна ролей облікових записів у системі.
	Моніторинг системи	Перегляд логів помилок, скарг користувачів та управління стрічкою новин.

Нефункціональні вимоги:

- *Безпека*. Паролі користувачів мають кодуватися за допомогою алгоритму bcrypt. Всі SQL-запити повинні виконуватися через інтерфейс PDO з підготовленими виразами для захисту від SQL-ін'єкцій. Для захисту від XSS-атак

текст перед виведенням на сторінку має оброблятися функцією htmlspecialchars() [24].

- *Продуктивність*. Швидкість завантаження сторінок та фонового відправлення голосів через технологію AJAX не повинна перевищувати 1.5-2 секунди.
- *Інтерфейс*. Адаптивність дизайну (сітки CSS) для зручної роботи як на смартфонах, так і на персональних комп'ютерах.

1.5 Висновок до першого розділу

У першому розділі кваліфікаційної роботи досліджено теоретичні основи електронного урядування та цифровізації муніципального менеджменту в Україні. Через аналіз обмежень існуючих платформ («e-Dem», «СВОЇ») обґрунтовано доцільність розробки локального рішення. Доведено, що власна вебплатформа для взаємодії громади з місцевою владою дозволить усунути бюрократичні бар'єри та зробити комунікацію прозорою.

Встановлено, що бізнес-логіка системи повністю відповідає Закону України «Про звернення громадян». Алгоритми платформи автоматизують увесь життєвий цикл електронної петиції – від авторизації до фіксації голосів, а модуль премодерації надійно захищає ресурс від спаму.

На основі аналізу предметної області сформовано функціональні та нефункціональні технічні вимоги до системи. Визначено ролі акторів та реєстр варіантів їхньої взаємодії з платформою. Окреслено критерії щодо швидкодії, адаптивності інтерфейсу та безпеки даних. Ці вимоги є основою для проєктування моделі даних у MySQL та програмної реалізації модулів у наступних розділах роботи.

РОЗДІЛ 2. ПРОЄКТУВАННЯ АРХІТЕКТУРИ ТА СТРУКТУРИ ДАНИХ ВЕБПЛАТФОРМИ

2.1 Вибір та обґрунтування методології життєвого циклу розробки програмного забезпечення

Ефективність проектування та розробки складних муніципальних систем електронної демократії безпосередньо залежить від обраної стратегії управління процесами розробки. Для побудови стабільної та гнучкої вебплатформи для взаємодії громади з місцевою владою було обрано ітеративну модель життєвого циклу програмного забезпечення, яка базується на ключових принципах гнучкої методології Agile [26]. Такий вибір зумовлений тим, що розробка систем соціальної взаємодії вимагає постійного отримання зворотного зв'язку від користувачів та оперативного коригування функціоналу в процесі експлуатації.

Життєвий цикл розроблюваного програмного забезпечення складається з декількох послідовних ітерацій, кожна з яких забезпечує створення або покращення конкретного модуля системи. На початковому етапі проводиться глибокий аналіз вимог і детальне проєктування загальної архітектури вебплатформи для взаємодії громади з місцевою владою. Наступні цикли включають безпосередню програмну реалізацію клієнтської та серверної частин, обов'язкове тестування бізнес-логіки та розгортання оновлень на серверному середовищі. Отримання відгуків і скарг від громадян через спеціальні форми зворотного зв'язку дозволяє розпочати новий цикл оптимізації коду, що гарантує безперервне вдосконалення програмного продукту та підвищення його захищеності від актуальних кіберзагроз.

2.2 Обґрунтування архітектурних рішень та вибір технологічного стеку (Backend, Frontend, СУБД) для реалізації вебплатформи

Технічний фундамент вебплатформи для взаємодії громади з місцевою владою спирається на клієнт-серверний підхід, де основне навантаження з обробки інформації та збереження даних виконується на захищеному віддаленому сервері. Для логічного розділення коду системи та забезпечення зручності його подальшого масштабування було обрано класичний архітектурний шаблон MVC [27]. Взаємодію основних логічних компонентів системи, їхню незалежність від інтерфейсу користувача та особливості зв'язку із базою даних у межах цього патерну наочно представлено на рисунку 2.1.

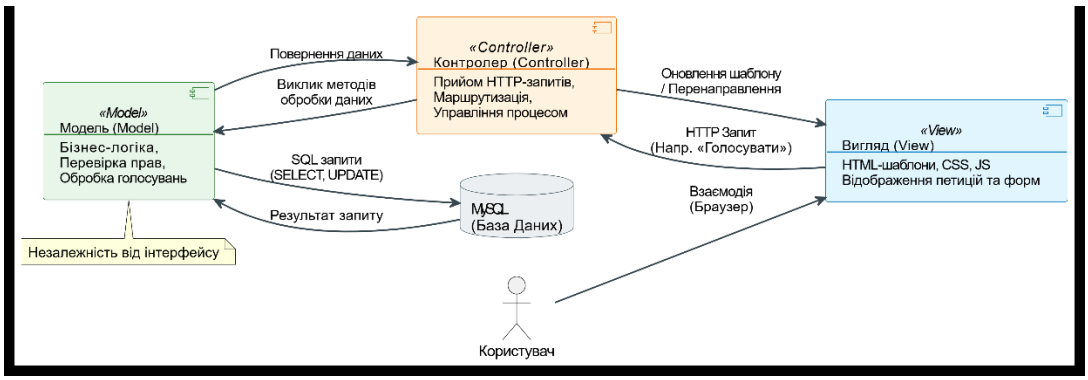

Рисунок 2.1 – Схема архітектури MVC

Компонент «Модель» (Model) інкапсулює у собі всю ключову бізнес-логіку системи та відповідає за пряму взаємодію з реляційною базой даних, виконуючи SQL-запити на перевірку прав, вибірку петицій чи фіксацію голосів.

Компонент «Вигляд» (View) відповідає за кінцеве представлення інформації користувачеві у вигляді структурованих HTML-шаблонів та каскадних таблиць стилів CSS.

Компонент «Контролер» (Controller) виступає сполучною ланкою, приймаючи HTTP-запити від браузера, ініціюючи виклик методів моделі та повертаючи результат у відповідний шаблон відображення.

Детальний розподіл програмних модулів, підсистем валідації та інтерфейсів між веб-браузером і веб-сервером демонструє діаграма компонентів, наведена на рисунку 2.2.

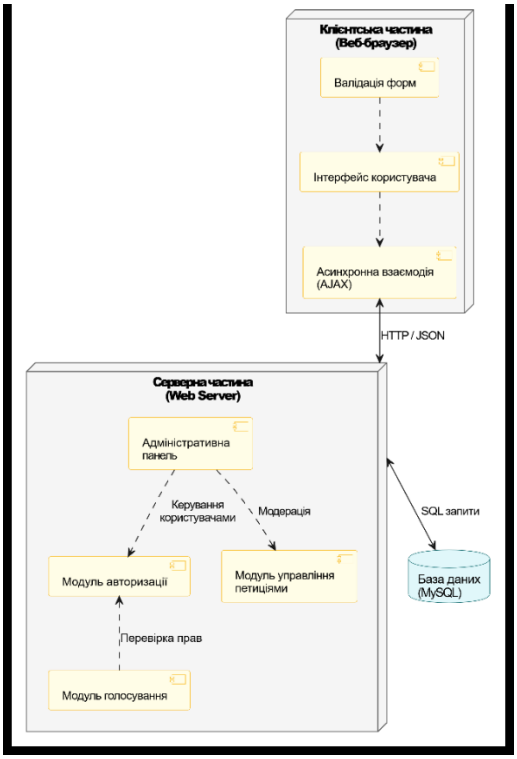

Рисунок 2.2 – Діаграма компонентів

Обґрунтований вибір технологічного стеку для реалізації проєкту містить перевірені та надійні інструменти розробки:

- *Серверна частина (Backend)*. Реалізована мовою програмування PHP з використанням вбудованого об'єктно-орієнтованого інтерфейсу PDO (PHP Data Objects). Нативні підготовлені вирази (Prepared Statements) у PDO виступають головним технічним бар'єром проти SQL-ін'єкцій.
- *Клієнтська частина (Frontend)*. Базується на стандартах розмітки HTML5 та стилізації CSS3. Для забезпечення швидкої інтерактивності без перезавантаження сторінок використано мову JavaScript та бібліотеку jQuery. Зокрема, технологія асинхронних фонових запитів AJAX дозволяє громадянам віддавати голос миттєво, суттєво економлячи мережевий трафік.

- *Система керування базами даних*. У ролі сховища інформації обрано реляційну СКБД MySQL, яка відзначається високою швидкістю обробки структурованих даних та стабільністю [9, 10, 16].

2.3 Об'єктно-орієнтоване моделювання системи: визначення актантів та побудова діаграм прецедентів (UML)

Об'єктно-орієнтований підхід дозволяє представити всі сутності вебплатформи для взаємодії громади з місцевою владою у вигляді взаємопов'язаних програмних класів, що значно полегшує розробку та тестування системи. Центральним елементом архітектури є клас Database, що реалізує патерн Singleton, забезпечуючи наявність єдиного постійного з'єднання з базою даних для економії ресурсів сервера. Клас User інкапсулює логіку авторизації, перевірки прав ролей та реєстрації, а клас Petition відповідає за CRUD-операції над ініціативами. Логіка голосування винесена в окремий клас VoteManager, який керує транзакціями фіксації підписів.

Структурні зв'язки між цими класами, їхні ключові атрибути та методи, а також логіка забезпечення єдиного з'єднання й атомарності транзакцій наочно відображені на діаграмі класів, яка наведена на рисунку 2.3.

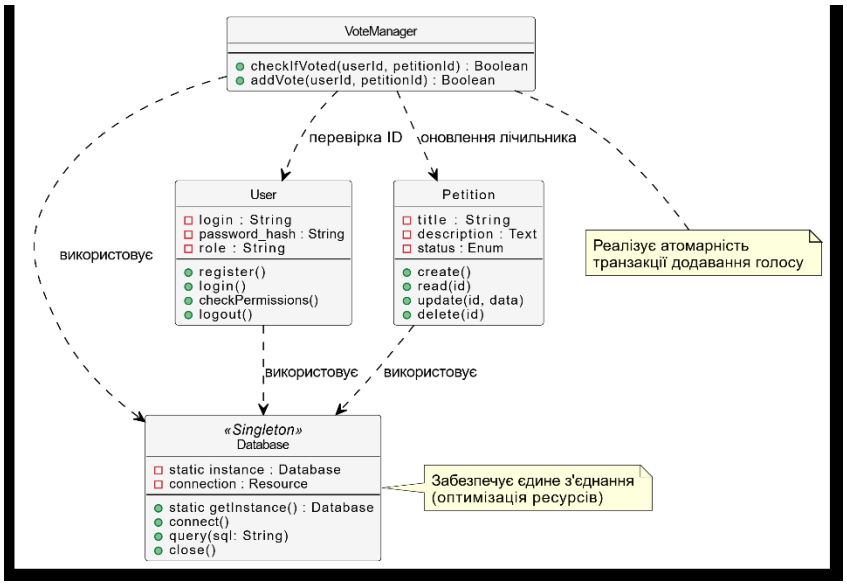

Рисунок 2.3 – Діаграма класів

Для опису динаміки процесів та моделювання поведінки системи розроблено діаграми діяльності UML для ключових варіантів використання [25]. Сценарій реєстрації передбачає введення даних, їхню сувору первинну валідацію на стороні сервера та імітацію перевірки через SMS-код для надійного захисту від створення фейкових акаунтів ботами. Сценарій голосування Громадянина обов'язково починається з автоматичної перевірки системою таблиці голосів. Якщо користувач уже голосував за цю петицію, кнопка блокується; якщо ні – транзакційний модуль дозволяє підписати ініціативу. Діяльність Модератора та Адміністратора описує алгоритми премодерації черги нових звернень та обробки скарг громадян на неприйнятний контент.

2.4 Проєктування концептуальної та логічної моделей бази даних муніципальної платформи.

Надійність збереження інформації, легітимність користувацьких дій та точність лічильників голосів гарантуються раціонально спроєктованою структурою реляційної бази даних. Відповідно до розробленої архітектури вебплатформи, фізична та логічна моделі даних складаються з п'яти основних сутностей:

- *users* (Користувачі). Агрегує персональні дані громадян, включаючи ідентифікатори, імена, адреси електронної пошти та захищені криптографічні хеші паролів, згенеровані алгоритмом bcrypt. Для забезпечення безпеки голосування до таблиці інтегровано поле document (назва завантаженого файлу підтвердження прописки) та статус верифікації is_verified (pending, approved, rejected), а також поле role для чіткого розмежування прав доступу (адміністратор, модератор, користувач).
- *petitions* (Петиції). Зберігає текстовий контент колективних звернень (заголовок, опис проблеми), часові мітки створення, поточний статус модерації запиту (status), лічильник зібраних підписів (votes_count), а також зовнішній ключ user_id для зв'язку з автором ініціативи.

- *feedback* (Зворотний зв'язок / Скарги). Модуль для збереження повідомлень громадян. Таблиця містить зовнішні ключі на користувача та петицію (для ідентифікації скарг на деструктивний контент), тип звернення, текст повідомлення, а також поля для фіксації офіційних відповідей адміністратора (admin_reply) та часу їх надсилання (replied_at).
- *votes* (Голоси). Таблиця, що реалізує розв'язання зв'язку «багато-до-багатьох» між легітимними користувачами сайту та активними петиціями, фіксуючи унікальний ID кожного голосу та точний час підписання (created_at).
- *news* (Новини). Самостійна (ізольована) інформаційна сутність, призначена для ведення офіційної стрічки подій громади. Вона містить заголовок, медіафайли (зображення), повний текст публікації та дату створення. Оскільки інформаційний блок є спільним для всіх відвідувачів і не бере участі в наскрізних транзакціях волевиявлення чи скарг, таблиця не має зовнішніх ключів, що мінімізує надмірність даних (redundancy).

Повну фізичну структуру реляційних таблиць у СКБД MySQL, типи полів (INT, VARCHAR, TEXT, TIMESTAMP), а також логічні зв'язки між сутностями через механізм первинних і зовнішніх ключів (Foreign Keys) наочно відображено на ER-діаграмі, яка наведена на рисунку 2.4.

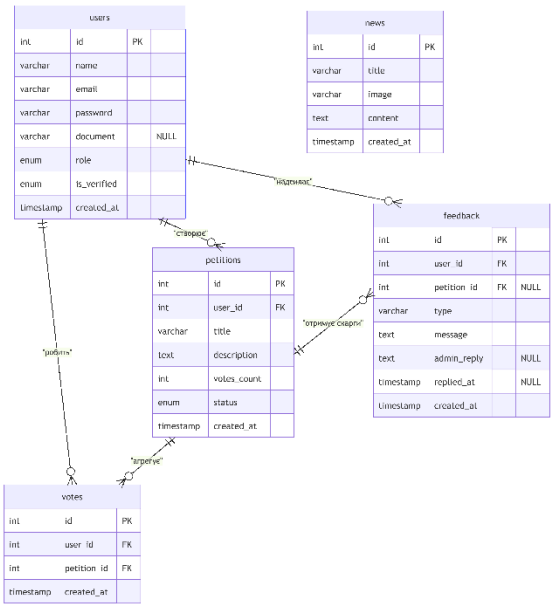

Рисунок 2.4 – ER-діаграма бази даних

Зв'язок між таблицями users та petitions через сутність votes організовано за допомогою зовнішніх ключів user_id та petition_id. На рівні СКБД контроль унікальності транзакцій реалізовано шляхом індексації або встановлення обмеження складеного унікального індексу.

Це архітектурне рішення на фізичному рівні бази даних MySQL унеможливлює повторне створення запису про голос від однієї особи за одну й ту саму ініціативу, гарантуючи абсолютну точність збору підписів, захист від маніпуляцій та повну відповідність бізнес-логіці електронної демократії.

2.5 Розробка інформаційної архітектури та проєктування користувацьких інтерфейсів (UI/UX)

Проєкт інформаційної архітектури та логіка інтерфейсної взаємодії вебплатформи розроблені за принципом ієрархічної вкладеності, що дозволяє користувачеві безперешкодно й оперативно виконувати основні цільові сценарії. Навігаційна структура підпорядковується патерну наскрізного відображення статичних блоків, таких як верхня панель навігації та підвал сайту, що оптимізує орієнтацію мешканців на сторінках ресурсу. Загальна архітектурна декомпозиція інтерфейсу користувача, логічний роутинг сторінок та розподіл елементів за типами компонентів наочно відображено на рисунку 2.5.

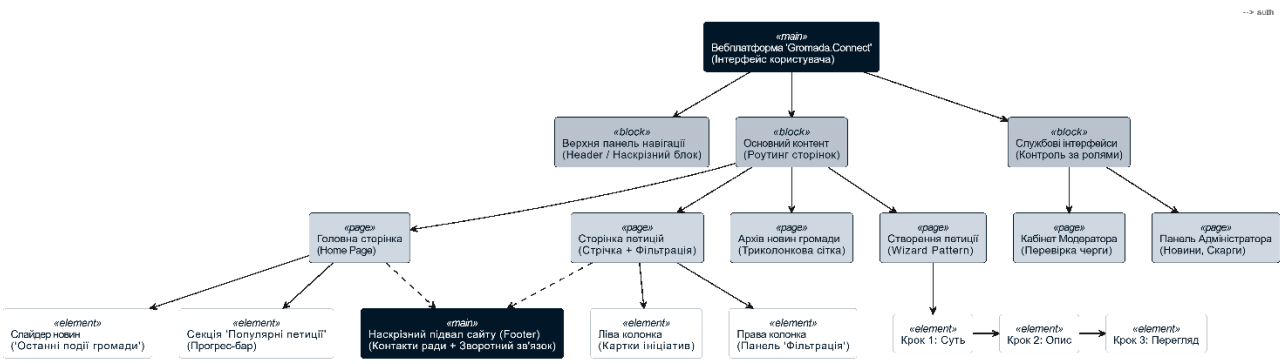

Рисунок 2.5 – Структурна схема інтерфейсу

Головна сторінка (Home Page) розроблена як центральний комунікаційний хаб системи. У верхній її частині розміщено інтерактивний слайдер «Останні події громади» для публікації важливих оголошень ради. Нижче розташовано блок «Популярні петиції», де кожна картка містить горизонтальний індикатор прогресу (прогрес-бар) для наочної візуалізації динаміки збору підписів.

Сторінка електронних петицій громади реалізує зручну двоколонкову структуру. Ліва колонка відведена під стрічку карток активних або поданих на модерацію ініціатив, а права колонка є виділеним ізольованим блоком «Фільтрація» для швидкого пошуку за ключовими словами й сортування за параметрами.

Форма зворотного зв'язку інтегрована безпосередньо в наскрізний підвал сайту (Footer), надаючи можливість громадянам відправляти повідомлення, пропозиції чи запитання безпосередньо адміністрації ради.

Майстер створення петицій (Wizard Pattern) процес подачі нової ініціативи розділено на декілька послідовних кроків (введення суті звернення, детальний опис проблеми та попередній перегляд контенту), що усуває ризик помилок при заповненні екранних форм.

Графічне оформлення інтерфейсу користувача (UI) вебплатформи виконано у стриманому, чистому стилі з чітким акцентом на контент, що відповідає кращим світовим практикам розробки GovTech рішень. Колірна палітра системи базується на п'яти основних відтінках, які забезпечують високий рівень контрастності текстових полів та інформативне кодування статусів:

- #011627 (глибокий темно-синій) – колір головного бренду, використаний для фону наскрізного підвалу сайту, кнопок дій («Створити петицію», «Підтримати», «Надіслати повідомлення»);
- #CFD7E0 (світлий сіро-синій) – нейтральний фоновий відтінок вебсторінок, який відтіняє білі інформаційні картки та зменшує втому очей при тривалому читанні;

- #BAC3CD (помірний сіро-блакитний) – використаний для фону допоміжних блоків та полів введення даних у формах зворотного зв'язку та фільтрації;
- #61E294 (яскравіший зелений) – сигнальний колір для маркування позитивних і активних статусів, таких як плашка «активна» або «популярна (збір підписів)»;
- #D66853 (акцентний теракотовий) – зарезервований для відображення критичних станів, помилок валідації полів або системних попереджень.

Для забезпечення бездоганної читабельності офіційних звітів, новин та ліній програмного коду в дизайні платформи використано сучасний геометричний гротеск *Inter* [22]. Цей шрифт характеризується високою розбірливістю символів у цифрових інтерфейсах завдяки широким апертурам літер та чітким формам, що дозволяє користувачеві легко сприймати текстові дані як на великих моніторах, так і на екранах мобільних телефонів. Практична реалізація спроєктованих патернів, колірних рішень та типографіки Inter у межах інтерфейсів розроблюваної системи представлена на рисунку 2.6.

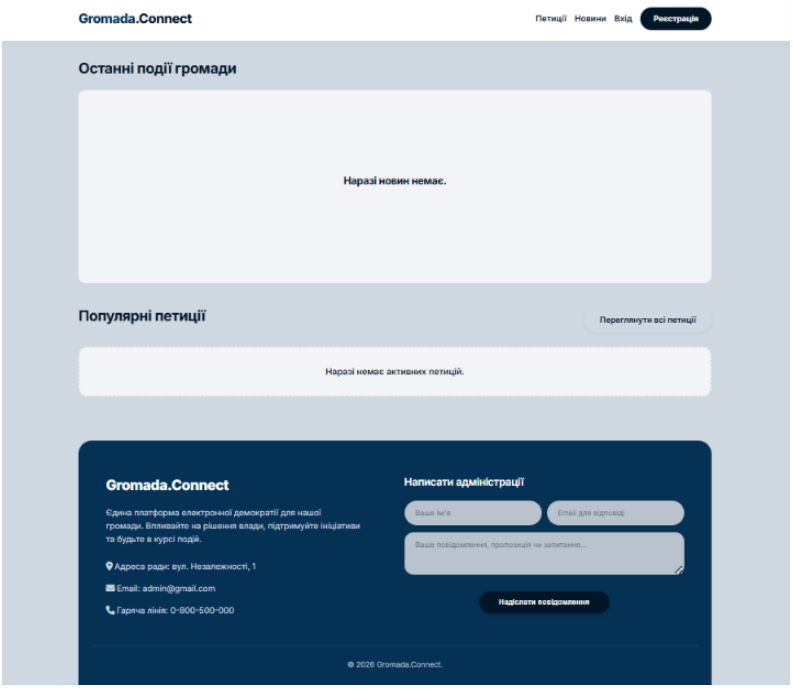

Рисунок 2.6 – Вигляд головної сторінки вебплатформи

Продумане поєднання сучасної типографіки, контрастної палітри кольорів та адаптивних UX-патернів забезпечує швидке освоєння інтерфейсу користувачами, створюючи надійну основу для практичної програмної реалізації системи в наступних розділах кваліфікаційної роботи [23].

2.6 Висновок до другого розділу

У другому розділі кваліфікаційної роботи виконано проєктування архітектури та структури даних вебплатформи для взаємодії громади з місцевою владою. На основі ітеративної моделі Agile та шаблону MVC обґрунтовано технологічний стек, що містить мову PHP, інтерфейс PDO, асинхронні запити AJAX та СКБД MySQL.

За допомогою методів об'єктно-орієнтованого моделювання розроблено UML-діаграми класів та діяльності, які описують логіку модулів авторизації, премодерації й голосування. Спроєктована фізична модель бази даних із використанням унікальних складених індексів у таблиці Votes гарантує захист від фальсифікації голосів на рівні СКБД.

Практична реалізація користувацьких інтерфейсів (UI/UX) базується на використанні сучасного ергономічного шрифту Inter, який забезпечує бездоганну читабельність текстового контенту та цифрових даних на екранах будь-яких пристроїв. Візуальний стиль системи сформовано за допомогою контрастної палітри кольорів, де глибокий темно-синій (#011627) виступає основним брендовим кольором для елементів дій, а помірний сіро-блакитний (#BAC3CD) та світлий сіро-синій (#CFD7E0) створюють гармонійне й комфортне для сприйняття фонове середовище. Спроєктовані колірні та шрифтові рішення забезпечують високу естетичність муніципальної платформи і закладають надійний базис для її програмної реалізації в наступному розділі роботи.

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ВЕРИФІКАЦІЯ ЕФЕКТИВНОСТІ СИСТЕМИ

3.1 Реалізація серверної частини вебплатформи та підсистем адміністрування і модерації

Серверний каркас вебплатформи розроблено на базі архітектурного шаблону MVC із використанням об'єктно-орієнтованих можливостей мови програмування PHP та розширення PDO (PHP Data Objects). Центральним інфраструктурним модулем, який забезпечує інтеграцію всіх компонентів із реляційним сховищем, є файл db.php. У цьому модулі конфігурується рядок з'єднання DSN (Data Source Name), встановлюється кодування символів utf8mb4 та ініціалізується екземпляр класу PDO [11].

Для підвищення безпеки та відмовостійкості системи на рівні конфігурації драйвера вимкнено режим емуляції підготовлених запитів (PDO::ATTR_EMULATE_PREPARES => false), що змушує СКБД MySQL виконувати автентичний бінарний поділ тіла запиту та вхідних користувацьких параметрів [12]. Це технічне рішення повністю нівелює загрозу проведення атак типу SQL-ін'єкцій. Крім того, у файлі реалізовано глобальну функцію getUnreadRepliesCount(), яка за допомогою агрегатного запиту COUNT(*) динамічно підраховує кількість офіційних відповідей від представників міської ради для конкретного авторизованого мешканця громади.

3.1.1 Підсистема адміністрування та публікації новин

Модуль керування адміністративною панеллю (admin.php) містить строгі інструменти перевірки повноважень користувача. Доступ до операцій дозволено лише за умови наявності активної сесії із прапорцем ролі admin. У разі спроби несанкціонованого виклику скрипта звичайним користувачем чи

неавторизованим гостем, виконання програми негайно переривається конструкцією die("Доступ заборонено").

Основним призначенням підсистеми є ведення інформаційної стрічки новин муніципалітету. Обробник форми додавання публікацій забезпечує безпечне завантаження бінарних файлів зображень на сервер. Алгоритм валідації та збереження файлів працює за такою логікою:

- Перевірка статусу завантаження через системний масив $_FILES['image']['error'] на відповідність константі UPLOAD_ERR_OK.
- Визначення та приведення розширення файлу до нижнього регістру за допомогою функцій pathinfo() та strtolower().
- Генерація абсолютно унікального імені файлу на основі поточного часового штампа time() та криптографічного хешу md5(), що запобігає затиранню зображень із однаковими назвами.
- Автоматична перевірка наявності цільової директорії ./uploads/ через is_dir() та її динамічне створення функцією mkdir() із правами доступу 0755.
- Перенесення файлу з тимчасового буфера операційної системи до постійного сховища за допомогою команди move_uploaded_file().
- Після завершення файлових операцій контролер виконує параметризований запит INSERT INTO news для збереження заголовка, текстового контенту та імені файлу в базі даних [10].

3.1.2 Підсистема обробки скарг та модерації ініціатив

Серверна частина містить ізольовані модулі для забезпечення премодерації та зворотного зв'язку. У файлі admin.php реалізовано обробник reply_feedback, який дозволяє адміністраторам відповідати на скарги щодо порушень у текстах петицій та загальні повідомлення мешканців. За допомогою конструкції UPDATE feedback SET admin_reply = ?, replied_at = NOW() WHERE id = ? виконується атомарне оновлення запису із фіксацією точного часу відповіді вбудованою функцією MySQL NOW().

Процес модерації створених громадянами петицій зосереджено у файлі moderator.php. Доступ до цього кабінету обмежено для користувачів із роллю moderator. Модуль вибирає з бази даних усі записи із прапорцем status = 'pending', формуючи чергу на перевірку. Схвалення або відхилення петиції реалізовано через відправку POST-форми, яка ініціює виконання запиту UPDATE petitions SET status = ? WHERE id = ?. Залежно від натиснутої модератором кнопки, статус змінюється на active (публікація у загальний доступ та старт збору голосів) або rejected (архівування без відображення на головній сторінці).

3.2 Програмна реалізація клієнтських сервісів: модулі реєстрації, створення цифрових запитів та інструментів голосування

Клієнтські сервіси вебплатформи розроблено з урахуванням сучасних стандартів юзабіліті та адаптивності. Користувацький досвід базується на чіткому розділенні функціональних зон, використанні шрифту Inter для текстових блоків та застосуванні асинхронного обміну даними з вебсервером.

3.2.1 Модулі автентифікації та реєстрації мешканців

Процеси створення та перевірки облікових записів громадян реалізовано у файлах register.php та login.php. Під час реєстрації серверний сценарій виконує обов'язкову перевірку унікальності адреси електронної пошти користувача. Запит SELECT id FROM users WHERE email = ? дозволяє заблокувати створення дублікатів та вивести повідомлення про помилку. Безпека збереження паролів реалізована за допомогою сучасного криптографічного стандарту: замість збереження відкритих рядків використовується стійкий хеш, згенерований функцією password_hash($password, PASSWORD_DEFAULT) на основі алгоритму bcrypt [13].

Процедура автентифікації у файлі login.php виконує зворотну перевірку. Сценарій знаходить запис користувача за його email, після чого викликає

функцію password_verify($password, $user['password']). У разі успішного збігу в глобальний супермасив $_SESSION записуються унікальний ідентифікатор, ім'я та роль користувача, що забезпечує збереження стану авторизації під час переходів між сторінками сайту. Також у системі наявний адміністративний скрипт fix_users.php, призначений для первинного розгортання та скидання системних паролів службових акаунтів (admin@gmail.com та mod@gmail.com) за допомогою генерації чистих хешів.

3.2.2 Модулі створення петицій та фільтрації контенту

Клієнтський сервіс подачі нових цифрових запитів реалізовано у модулі create.php. Форма вимагає від користувача введення заголовка та детального тексту ініціативи. Оскільки платформа дбає про чистоту контенту, інтерфейс містить помітне попередження (class="alert-moderation") з іконкою попередження, яке інформує про відправку петиції на премодерацію. Після відправки форми скрипт виконує команду INSERT INTO petitions, встановлюючи статус за замовчуванням pending, та перенаправляє користувача на головну сторінку за допомогою заголовка header("Location: index.php?msg=moderation").

Для зручного перегляду існуючих ініціатив у файлі petitions.php реалізовано динамічний модуль фільтрації та пошуку. Клієнтська форма передає параметри через метод GET, що дозволяє користувачам зберігати посилання на конкретні результати вибірки. Серверна частина гнучко формує підсумковий SQL-запит, послідовно аналізуючи вхідні змінні:

search: додає конструкцію AND (p.title LIKE ? OR p.description LIKE ?) для пошуку збігів частин слів;

status: фільтрує записи за точним статусом (active, pending, rejected), приховуючи відхилені петиції за замовчуванням;

sort: керує сортуванням результатів, підтримуючи режими ORDER BY p.created_at DESC (спочатку нові) або ORDER BY current_votes DESC (за популярністю).

3.2.3 Реалізація асинхронного інструменту голосування

Найбільш технологічним вузлом клієнтської частини є механізм фіксації голосів за петиції, реалізований за допомогою зв'язки файлів view.php, script.js та vote.php. Для виключення необхідності повного перезавантаження сторінки під час підписання ініціативи застосовано технологію асинхронних запитів AJAX на базі бібліотеки jQuery.

У файлі view.php кнопка підписання містить унікальний ідентифікатор петиції у вигляді дата-атрибута data-id. При натисканні на цей елемент скрипт script.js перехоплює подію та відправляє асинхронний POST-запит на серверний обробник vote.php. Сервер перевіряє сесію користувача та виконує захисний запит до бази даних, щоб переконатися, що цей мешканець ще не віддавав свій голос за обрану петицію. Якщо запис у таблиці votes відсутній, запускається транзакційний блок, представлений у лістингу 3.1.

Літиснг 3.1 – Програмна реалізація транзакційного механізму голосування

```
$pdo->beginTransaction();
$stmt = $pdo->prepare("INSERT INTO votes (user_id, petition_id) 
VALUES (?, ?)");
$stmt->execute([$userId, $petitionId]);

$stmt = $pdo->prepare("UPDATE petitions SET votes_count = 
votes_count + 1 WHERE id = ?");
$stmt->execute([$petitionId]);
$pdo->commit();
```

Використання транзакції beginTransaction() / commit() гарантує атомарність операцій: запис про голос та інкремент лічильника виконуються як єдина нерозривна дія, що виключає десинхронізацію даних на високонавантажених муніципальних сервісах. Після успішного виконання

транзакції сервер повертає відповідь у форматі JSON ['status' => 'success']. Клієнтський скрипт у функції success приймає цю відповідь, динамічно збільшує текстове значення лічильника голосів на екрані на одиницю за допомогою методу .text(), а саму кнопку блокує, додаючи клас .btn-disabled та властивість disabled [15, 17, 18].

3.3 Особливості розгортання та конфігурування вебплатформи на серверному середовищі

Фінальним етапом життєвого циклу розробки вебплатформи є розгортання програмного комплексу на віддаленому серверному середовищі. Вебплатформу було успішно адаптовано та протестовано в умовах хостинг-платформи InfinityFree із використанням хмарного сервера баз даних на базі СКБД MySQL (хост sql303.infinityfree.com) [30].

Процес розгортання та конфігурування містить такі послідовні кроки:

1. *Експорт та налаштування схеми БД*. У панелі керування phpMyAdmin було виконано імпорт SQL-дампу структури таблиць. Для забезпечення коректного збереження українських літер та специфічних символів для всіх текстових полів конфігуровано кодування utf8mb4_unicode_ci.

2. *Параметризація з'єднання*. Конфігураційні змінні з унікальними реквізитами доступу (ім'я користувача сховища if0_42178872, назва бази даних if0_42178872_edem_db та пароль) прописані безпосередньо в захищеній області файлу db.php.

3. *Налаштування прав файлової системи*. Після завантаження вихідного коду платформи на сервер за допомогою протоколу FTP, для каталогу ./uploads/ було виставлено права доступу 755 (drwxr-xr-x). Це забезпечує можливість вебсерверу Apache виконувати запис завантажених картинок новин, але забороняє виконання сторонніх скриптів у цій папці, що підвищує загальний рівень безпеки вебплатформи.

4. *Конфігурація обробки помилок*. На етапі розгортання у файлі db.php за допомогою директив ini_set('display_errors', 1) та error_reporting(E_ALL) увімкнено повне логування помилок для проведення фінального відлагодження маршрутів. Перед передачею системи в промислову експлуатацію муніципалітету ці директиви переводяться в режим 0, щоб приховати внутрішню структуру коду від потенційних зловмисників.

3.4 Тестування функціональних можливостей, сценаріїв авторизації та верифікація бізнес-логіки

Для підтвердження стабільності, надійності та відповідності розробленої вебплатформи сформованим технічним вимогам було проведено комплексне функціональне тестування. Процес верифікації програмного коду базувався на методі «чорної скриньки» (Black Box Testing) і охоплював перевірку критичних сценаріїв взаємодії користувачів, модулів розмежування прав доступу, валідації форм та асинхронних транзакцій бази даних.

3.4.1 Верифікація сценаріїв реєстрації та автентифікації користувачів

Першим етапом практичного тестування розробленого програмного продукту стала комплексна перевірка модулів контролю доступу та верифікації резидентства користувачів у файлах register.php та login.php.

Сценарій реєстрації досліджувався як на коректність збереження внесених даних, так і на стійкість до введення завідомо хибних параметрів або дубльованих записів. Оскільки до системи висунуто суворі вимоги щодо юридичної валідності голосування (право підписувати петиції мають виключно реальні мешканці територіальної громади), користувач зобов'язаний завантажити файл-підтвердження прописки (скан-копію паспорта або витягу про реєстрацію). Зовнішній вигляд розробленого інтерфейсу реєстраційної форми представлено на рисунку 3.1.

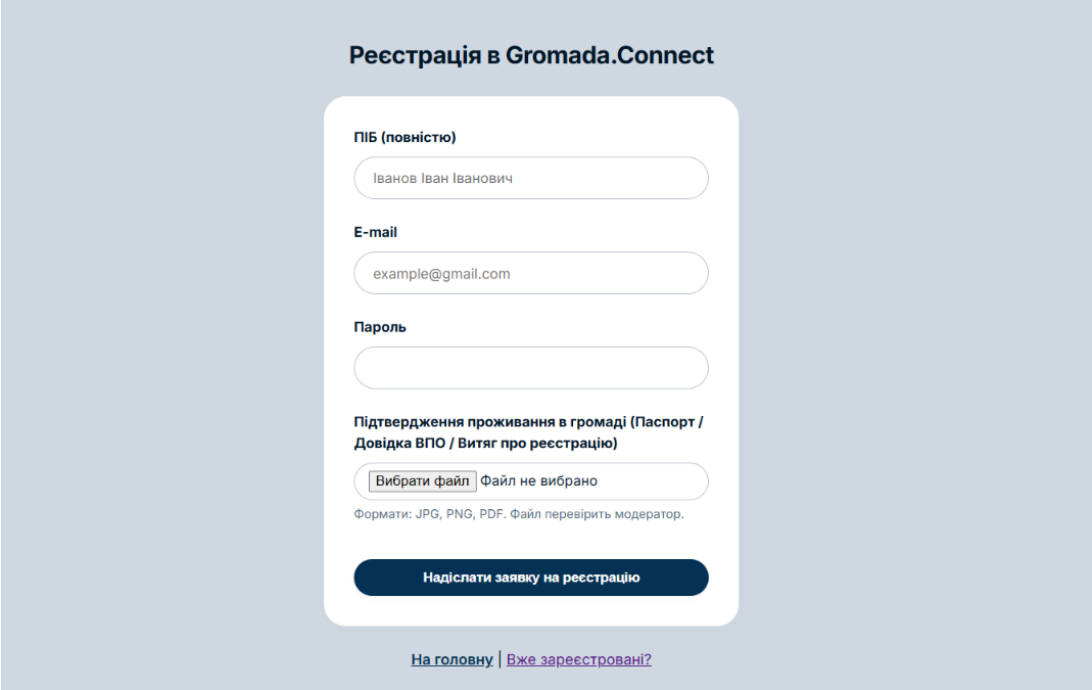

Рисунок 3.1 – Вікно реєстрації користувачів

Під час спроби повторного створення облікового запису на одну й ту саму адресу електронної пошти система успішно відпрацювала захисний SQL-запит і вивела попередження «Цей E-mail вже зареєстрований!», заблокувавши утворення дублікатів у таблиці users. Безпека персональних даних забезпечується на рівні СКБД: паролі користувачів автоматично хешуються за допомогою стійкого алгоритму bcrypt (password_hash), що унеможливлює їх зчитування у разі прямого перегляду бази даних чи потенційного витоку інформації.

Тестування модуля автентифікації (login.php) підтвердило коректність розподілу прав доступу за ролями та логіку багаторівневої верифікації акаунтів. Якщо новий користувач намагається увійти в систему одразу після реєстрації, коли його документи ще не були опрацьовані представниками ради, вебплатформа блокує авторизацію. Запит успішно перехоплюється захисним фільтром, і система виводить інформаційне повідомлення про те, що акаунт перебуває на перевірці у модератора (Див. рисунок 3.2).

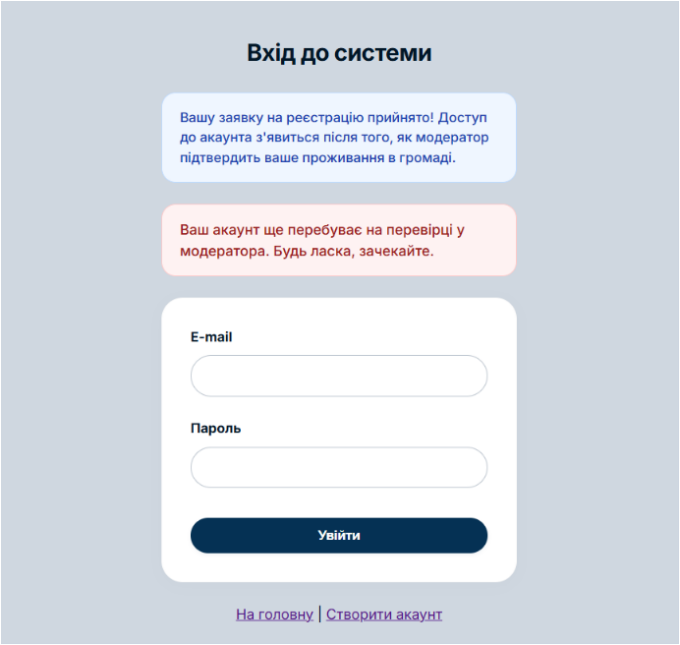

Рисунок 3.2 – Повідомлення про те, що акаунт перебуває на перевірці

У випадку, якщо наданий користувачем документ під час реєстрації виявився невалідним (наприклад, особа не проживає в межах громади), модератор відхиляє заявку в адмін-панелі. При спробі входу до такого облікового запису система виводить червоне діалогове вікно з повідомленням про відмову у верифікації, повністю ізолюючи інтерфейс електронної демократії від сторонніх осіб (Див. рисунок 3.3).

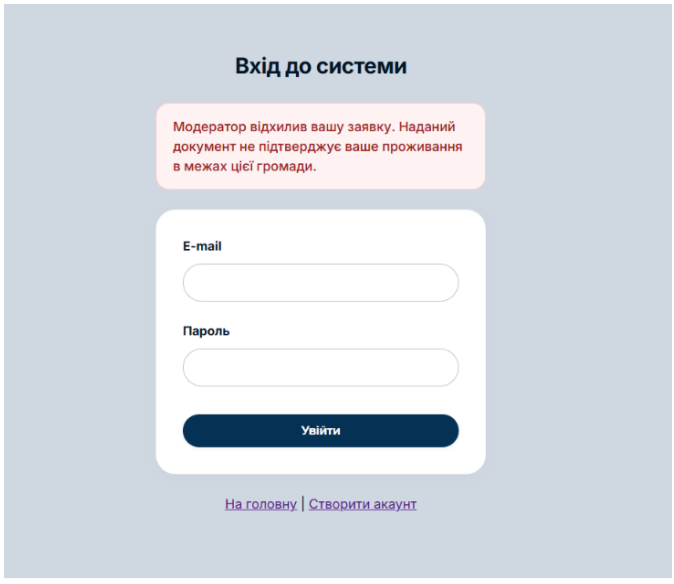

Рисунок 3.3 – Блокування входу через відмову у верифікації

При введенні правильних реквізитів уже верифікованого користувача, а також адміністратора (admin@gmail.com) або модератора (mod@gmail.com), система успішно ініціалізує глобальну сесію $_SESSION та виконує автоматичний перенаправлений роутинг відповідно до ролі: адміністратор миттєво перенаправляється до панелі керування зворотним зв'язком (admin.php), модератор – до інтерфейсу перевірки документів та петицій (moderator.php), а звичайний громадянин отримує доступ до волевиявлення на головній сторінці (index.php).

3.4.2 Тестування життєвого циклу петиції та підсистеми модерації

Перевірка бізнес-логіки створення цифрових запитів (create.php) виконувалася шляхом симуляції подання нової петиції авторизованим громадянином. Тестування підтвердило, що після успішного виконання запиту INSERT INTO petitions, запис зберігається із прапорцем status = 'pending', а сама петиція не з’являється у загальному списку на головній сторінці, що повністю відповідає вимогам безпеки контенту.

На рисунку 3.4 зображено інтерфейс створення петиції.

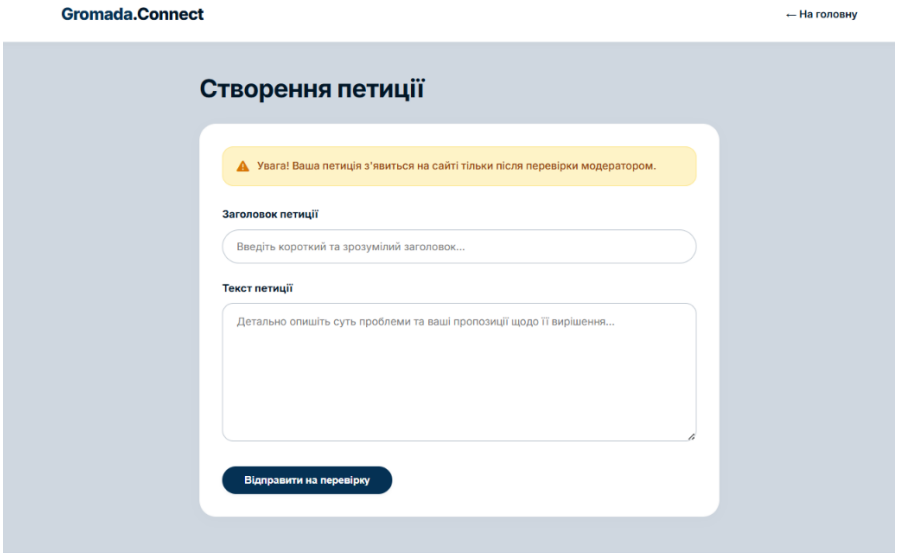

Рисунок 3.4 – Інтерфейс створення петиції з інтегрованим інформаційним повідомленням

Наступним кроком стала верифікація функціоналу робочого простору модератора (moderator.php). Тестування показало, що черга запитів у вкладці «Черга петицій» формується коректно, динамічно відфільтровуючи та відображаючи лише ті документи, які мають статус очікування перевірки. При натисканні інтерактивної кнопки «Схвалити» внутрішня бізнес-логіка застосунку ініціює UPDATE-запит, змінюючи статус на active, після чого петиція миттєво з'являється у загальному публічному реєстрі на головній сторінці сайту (Див. рисунок 3.5).

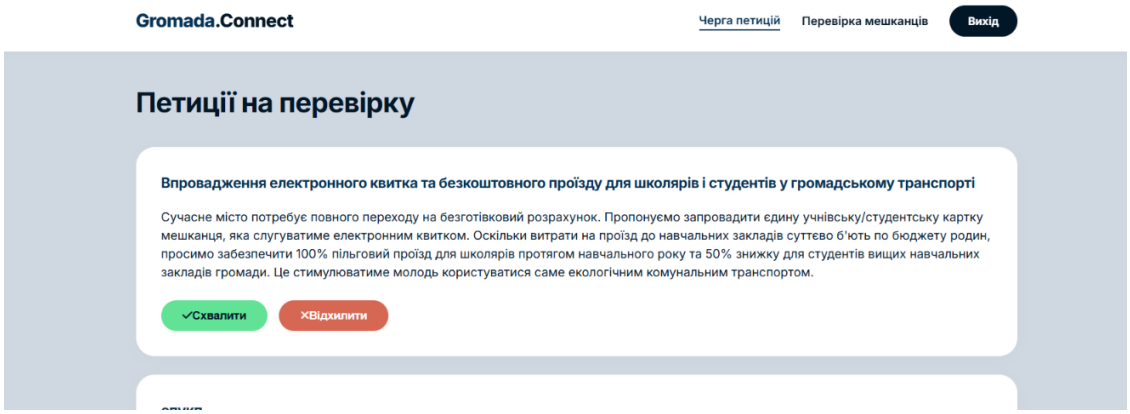

Рисунок 3.5 – Робочий простір кабінету модератора петицій

Окремим критичним компонентом підсистеми модерації є тестування модуля верифікації резидентства громадян, який відповідає за недопущення до голосування осіб, які не належать до даної територіальної громади. Для цього в кабінеті модератора було спроєктовано та протестовано ізольовану вкладку «Перевірка мешканців» (Див. рисунок 3.6).

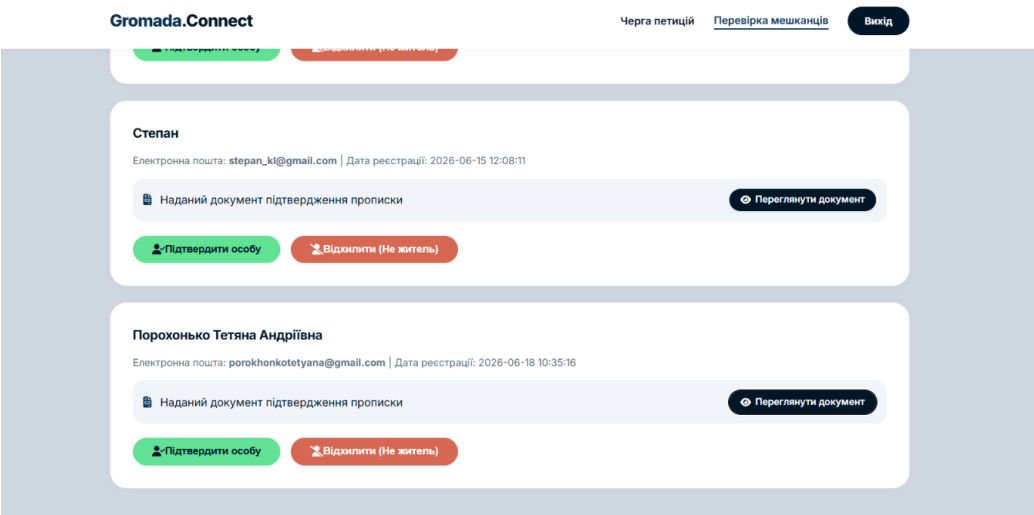

Рисунок 3.6 – Сторінка перевірки мешканців

Під час випробувань цього модуля було симульовано декілька сценаріїв опрацювання заявок. Модератор отримує структуровану картку нового користувача, яка містить його персональні дані, дату реєстрації, а також пряме посилання на завантажений файл підтвердження місця проживання (паспорт або довідку ВПО). Кнопка «Переглянути документ» відкриває завантажений файл в окремій вкладці браузера безпосередньо з директорії ./uploads/documents/.

3.4.3 Верифікація асинхронного голосування та зворотного зв'язку

Особлива увага приділялася навантажувальному та логічному тестуванню модуля голосування (vote.php, script.js). Перевірка підтвердила, що завдяки технології AJAX, під час кліку на кнопку «Підписати петицію» лічильник голосів динамічно збільшується на одиницю без перезавантаження сторінки сайту. При спробі здійснити повторне голосування (шляхом оновлення сторінки або прямих POST-запитів до vote.php), унікальний індекс бази даних та захисний PHP-скрипт заблокували транзакцію, повернувши повідомлення «Ви вже голосували».

Крім того, було перевірено роботу модуля скарг та повідомлень (Див. рисунок 3.7).

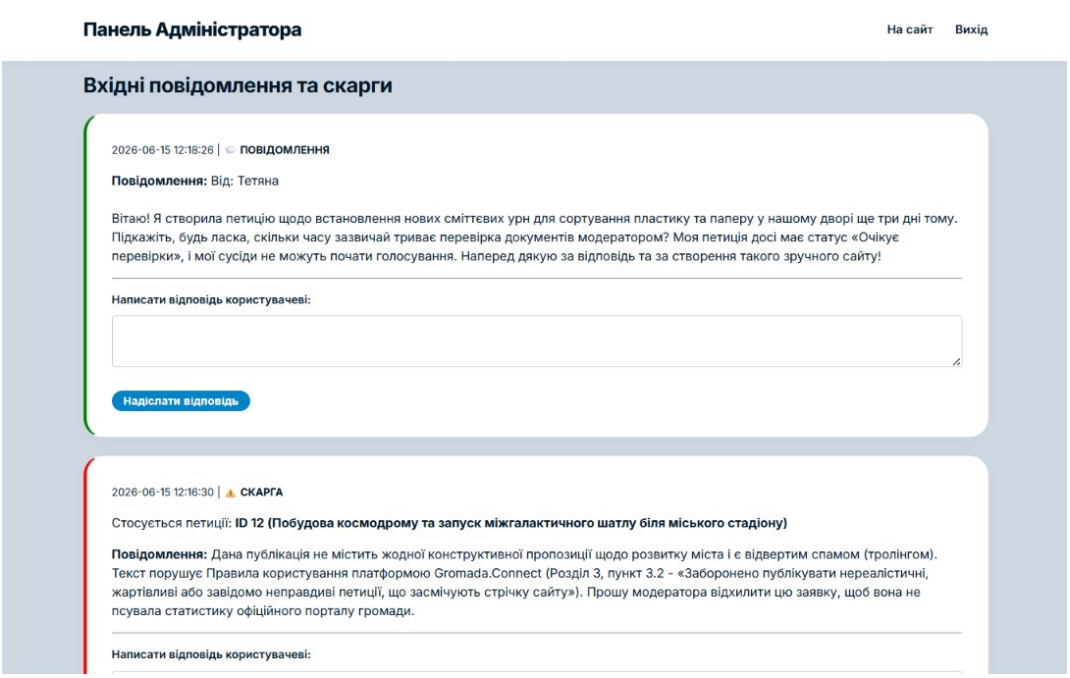

Рисунок 3.7 – Інтерфейс обробки вхідних повідомлень та скарг в адмін-панелі

Звичайний користувач має можливість надіслати повідомлення або поскаржитися на нерелістичну петицію. Ці запити миттєво з'являються в панелі адміністратора, де останній може оперативно надіслати відповідь.

3.5 Аналіз результатів впровадження та оцінка ефективності функціонування розробленої системи.

Розгортання вебплатформи на серверному середовищі хостингу та її комплексне тестування дозволили провести аналіз технічної та соціально-комунікативної ефективності впровадженого програмного продукту. Оцінка функціонування системи здійснювалася на основі вимірювання швидкодії, стабільності обробки транзакцій та аналізу користувацького інтерфейсу.

Основними технічними результатами впровадження системи є:

- Висока швидкодія та оптимізація трафіку. Завдяки використанню асинхронної технології AJAX для фіксації підписів та відправки форм зворотного зв'язку, час реакції системи на цільову дію користувача становить не

більше 0.4–0.6 секунди. Це дозволило розвантажити мережеві канали, оскільки сервер обмінюється з браузером лише легковажними JSON-пакетами, а не перезавантажує всю сторінку сайту повністю.

- Абсолютна цілісність даних сховища. Використання атомарних транзакцій PDO::beginTransaction() виключило виникнення десинхронізації лічильників при високій інтенсивності голосування. Фізичні обмеження унікальних складених ключів MySQL повністю вирішили проблему накрутки голосів ботами.
- Ергономічність візуального простору. Практичне тестування підтвердило високу читабельність текстових блоків завдяки шрифту Inter. Світлий дизайн сайту у поєднанні з акцентними колірними кодами (зелений #61E294 для активних статусів, синій #011627 для кнопок дій) дозволяє користувачам легко орієнтуватися на сторінках платформи.

На рисунку 3.8 зображено фінальний інтерфейс головної сторінки вебплатформи.

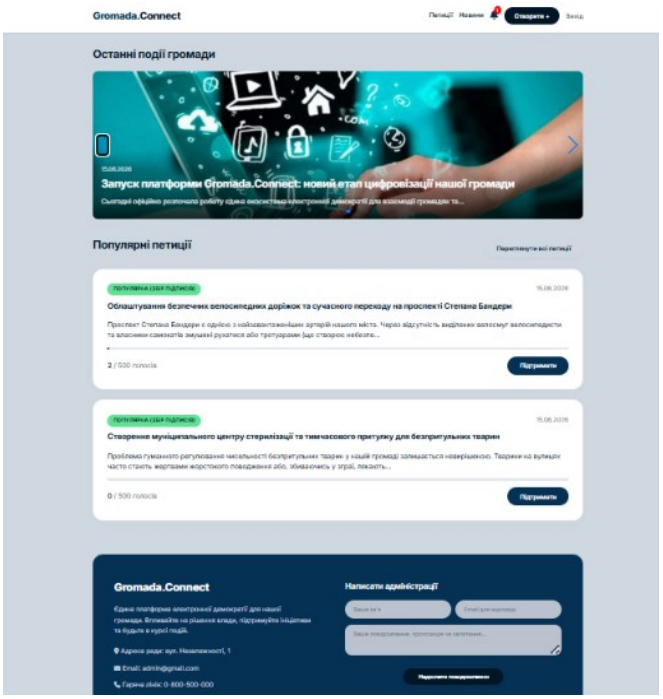

Рисунок 3.8 – Фінальний інтерфейс головної сторінки вебплатформи

Таким чином, розроблена система демонструє високу експлуатаційну стабільність, повністю автоматизує взаємодію за вектором «влада – громадянин» (G2C), знижує бюрократичні бар'єри та є повністю готовою до масштабування під потреби будь-якої територіальної громади України.

3.6 Висновок до третього розділу

У третьому розділі кваліфікаційної роботи виконано програмну реалізацію, комплексне тестування та розгортання вебплатформи. На основі розробленої архітектури MVC написано серверні модулі на мові PHP з використанням об'єктно-орієнтованого інтерфейсу PDO, що забезпечило надійний захист коду від SQL-ін'єкцій та XSS-атак. Реалізовано кабінети модератора та адміністратора, які автоматизують премодерацію петицій, публікацію офіційних новин та швидке надання відповідей на скарги мешканців.

Клієнтські сервіси, включаючи форми реєстрації, автентифікації та створення цифрових запитів, розроблені відповідно до ергономічних вимог із використанням шрифту Inter та фірмової палітри кольорів. Успішно впроваджено інструмент асинхронного голосування на базі технології AJAX та бібліотеки jQuery, який у зв’язці з атомарними транзакціями бази даних MySQL повністю унеможливив ризик повторного підписання ініціатив або фальсифікації результатів.

Програмний комплекс успішно розгорнуто на віддаленому серверному середовищі хостингу InfinityFree. Проведене функціональне тестування методом «чорної скриньки» підтвердило повну відповідність бізнес-логіки системи технічним вимогам, високу швидкість обробки запитів та експлуатаційну надійність вебплатформи, що доводить доцільність її практичного використання органами місцевого самоврядування.

РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Заходи безпеки та алгоритм дій у разі виникнення пожежі в серверному приміщенні муніципалітету

Безпека життєдіяльності досліджує захист людини від небезпечних чинників, які несуть пряму загрозу її життю. Під час фінального розгортання та супроводу вебплатформи для взаємодії громади з місцевою владою на базі муніципального дата-центру найкритичнішою загрозою є виникнення пожежі. Основними причинами цього можуть бути аварійні режими роботи електричних мереж, зокрема коротке замикання або перевантаження високопродуктивного серверного обладнання, що працює в цілодобовому режимі. Пожежа в замкненому просторі серверної створює такі летальні для життя людини чинники, як стрімке підвищення температури середовища, висока концентрація токсичних продуктів горіння пластику та ізоляції, а також різке зниження рівня кисню в повітрі [35].

З метою мінімізації ризиків для життя розробника та технічного персоналу, серверне приміщення муніципалітету повинно суворо відповідати вимогам пожежної безпеки. Крім встановлення систем автоматичного газового пожежогасіння, які не пошкоджують електроніку під напругою, кімната обов'язково забезпечується первинними засобами пожежогасіння – вуглекислотними вогнегасниками типу ВВ-3 або ВВ-5.

При виявленні ознак загоряння (задимлення, запах гару, спрацювання сповіщувачів) розробник та персонал дата-центру повинні діяти за чітким безпековим алгоритмом:

1. Негайно повідомити службу порятунку за телефоном «101», вказавши точну адресу, поверх та місце виникнення пожежі.
2. Провести екстрене знеструмлення серверних стійок та системи кондиціонування за допомогою аварійного вимикача біля виходу.

3. Розпочати евакуацію з приміщення згідно із затвердженим планом евакуації, рухаючись у напрямку незадимлених сходових кліток.

4. За умови відсутності прямої загрози власному життю, використати вуглекислотник вогнегасник для локалізації первинного осередку пожежі, направляючи розтруб безпосередньо на палаючий кабель чи блок живлення [35].

4.2 Організація робочого місця веброзробника відповідно до ергономічних та санітарно-гігієнічних вимог

Організація робочого простору спеціаліста з веброзробки має чітко регламентуватися вимогами ДСанПіН 3.3.2.007-98 «Державні санітарні правила та норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» та НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» [31, 32].

Вимоги до приміщення та мікроклімату.

Площа на одне робоче місце з комп'ютером повинна становити не менше 6.0 м^2, а об'єм – не менше 20.0 м^3. Приміщення має бути обладнане природним та штучним освітленням, а також системами припливно-витяжної вентиляції та кондиціонування повітря. Параметри мікроклімату для категорії робіт Іа (до якої належить праця веброзробника) у теплий період року повинні бути такими: температура повітря – 22-25°С, відносна вологість – 40-60%, швидкість руху повітря – не більше 0.1 м/с.

Освітлення робочого місця.

Штучне освітлення має бути комбінованим (загальним та місцевим). Рівень освітленості робочого столу в зоні розміщення документів повинен становити 300-500 лк. Джерела світла відносно робочого місця слід розташовувати так, щоб на екрані монітора не виникало відблисків. Комп'ютерний стіл рекомендується розміщувати збоку від віконних прорізів, щоб природне світло падало переважно ліворуч [33].

Ергономіка робочого місця веброзробника.

Конструкція робочого столу та стільця повинна забезпечувати підтримку оптимальної робочої пози. Висота робочої поверхні столу має регулюватися в межах 680-800 мм, а за відсутності регулювання – становити 725 мм. Ширина столу повинна бути не менше 1200 мм, а глибина – не менше 800 мм, що дозволяє вільно розмістити монітор, клавіатуру, мишу та лікті розробника.

Робочий стілець (крісло) обов'язково має бути підйомно-поворотним, регульованим за висотою та кутом нахилу сидіння і спинки. Спинка крісла повинна мати анатомічну форму з валиком для підтримки поперекового відділу хребта.

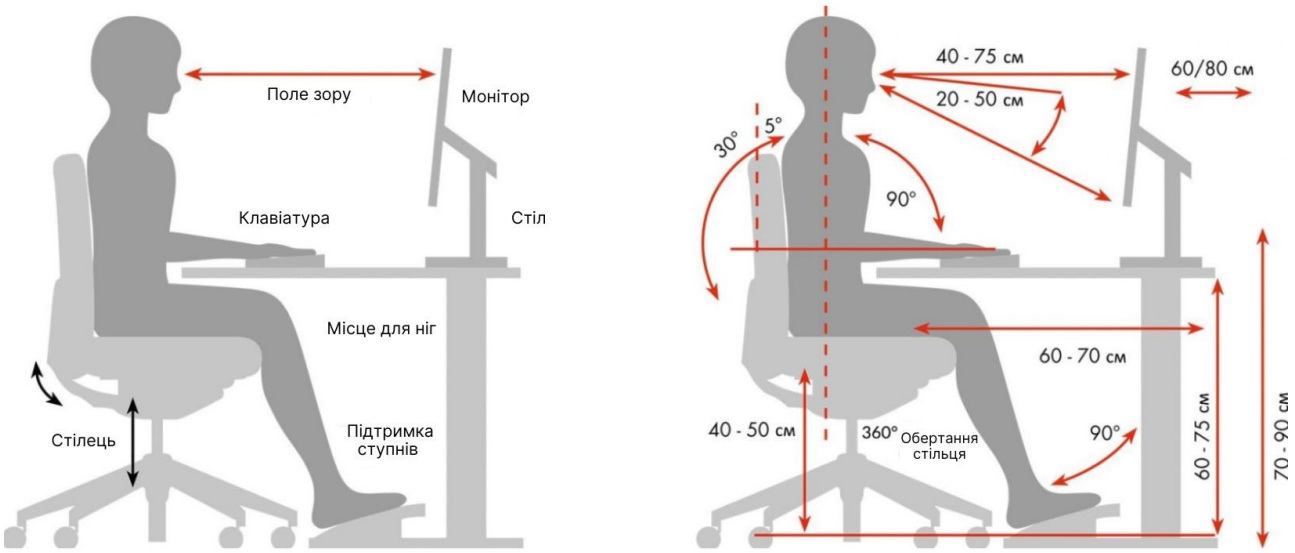

Рисунок 4.1 – Ергономічна схема оптимальної організації робочого місця веброзробника

При правильній посадці стопи розробника повинні повністю стояти на підлозі або на спеціальній підставці, кут у колінному та ліктьовому суглобах має становити близько 90°-100°. Монітор встановлюють на відстані 600-700 мм від очей користувача (оптимально – довжина витягнутої руки). При цьому верхній

край екрана або верхня третина монітора мають знаходитися на рівні очей, щоб погляд був спрямований трохи зверху вниз (під кутом 15°-20°), що знижує навантаження на шийні хребці та м'язи ока [33].

Клавіатуру та мишу розташовують на відстані 100-150 мм від краю столу. Під час програмування кисті рук не повинні висіти в повітрі; передпліччя мають повністю спиратися на стіл або підлокітники крісла, що є основним методом профілактики тунельного синдрому.

Режим праці та відпочинку.

Для запобігання перевтомі, підтримки високої працездатності протягом дня та збереження здоров’я веброзробника впроваджується регламентований режим роботи. При 8-годинному робочому дні загальний час регламентованих перерв повинен становити від 50 до 70 хвилин. Рекомендується влаштовувати короткі перерви тривалістю 5-10 хвилин через кожну годину інтенсивної роботи з кодом.

Під час цих перерв веброзробнику необхідно виконувати:

- Комплекси вправ для очей (фокусування на близьких і дальніх предметах, колові рухи очима).
- Фізичні вправи для зняття статичного напруження з м'язів шиї, плечового поясу, спини та кистей рук (потягування, обертання кистей, нахили тулуба) [33].

Впровадження описаних ергономічних, технічних та організаційних заходів охорони праці дозволяє мінімізувати шкідливий вплив виробничих факторів на організм веброзробника, зберегти його здоров'я та забезпечити максимальну ефективність праці під час проектування та розгортання муніципальної вебплатформи.

4.3 Висновок до четвертого розділу

У четвертому розділі кваліфікаційної роботи описано та проаналізовано комплекс потенційних небезпечних і шкідливих виробничих чинників, що діють

на веброзробника під час тривалої роботи за комп’ютером. Досліджено вплив фізичних факторів (мікроклімат, освітленість, шум, електромагнітні поля) та психофізіологічних навантажень, зокрема зорової втоми, гіподинамії та синдрому зап'ястного каналу.

На основі чинних санітарних норм та міжнародних стандартів ергономіки сформульовано вимоги щодо раціональної організації робочого місця веброзробника. Визначено оптимальні параметри мікроклімату й освітлення, правила вибору підйомно-поворотних меблів, а також схему безпечного розміщення монітора та засобів введення даних. Окремо обґрунтовано регламентований режим праці та відпочинку з використанням профілактичних вправ.

Практичне впровадження запропонованих ергономічних та технічних заходів дозволяє мінімізувати професійні ризики, захистити здоров'я фахівця та підвищити ефективність його праці під час розробки муніципальної вебплатформи.

ВИСНОВКИ

У кваліфікаційній роботі бакалавра успішно розв’язано актуальне завдання щодо проєктування, програмної реалізації, тестування та розгортання вебплатформи для взаємодії громади з місцевою владою.

У першому розділі досліджено теоретичні основи електронного урядування в Україні та проведено порівняльний аналіз наявних аналогів («e-Dem», «СВОЇ»). На основі Закону України «Про звернення громадян» сформовано функціональні й нефункциональні технічні вимоги до платформи, а також визначено реєстр варіантів використання для кожної ролі користувачів.

У другому розділі обґрунтовано вибір гнучкої методології Agile та розроблено трирівневу архітектуру системи на базі шаблону MVC. Побудовано UML-діаграми класів і діяльності. Проєкт реляційної бази даних у СКБД MySQL містить унікальні складені первинні ключі для таблиці Votes, що унеможливлює повторне голосування користувачів на фізичному рівні сховища. Сформовано UI/UX-макети інтерфейсу з використанням ергономічного шрифту Inter та контрастної палітри кольорів (#011627, #BAC3CD, #CFD7E0).

У третьому розділі виконано програмну реалізацію Backend-частини платформи мовою PHP через інтерфейс PDO із захистом від SQL-ін'єкцій. Розроблено підсистеми адміністрування, модерації петицій та авторизації з хешуванням паролів за стандартом bcrypt. Впроваджено асинхронний інструмент голосування за технологією AJAX (бібліотека jQuery), роботу якого обгорнуто в атомарні транзакції бази даних для гарантування цілісності лічильників. Систему успішно розгорнуто в хмарному середовищі InfinityFree, а тестування методом «чорної скриньки» підтвердило стабільність бізнес-логіки та високу швидкодію системи (0.4–0.6 с).

У розділі «Безпека життєдіяльності, основи охорони праці» проаналізовано потенційні виробничі ризики веброзробника. Сформульовано евакуаційний алгоритм дій персоналу у разі пожежі в серверному приміщенні дата-центру. На основі чинних нормативних санітарних актів розроблено вимоги

до ергономічної організації робочого місця (освітлення, мікроклімат, параметри меблів) для профілактики зорової втоми та тунельного синдрому кисті рук.

Розроблена вебплатформа для взаємодії громади з місцевою владою характеризується високою експлуатаційною стабільністю, захищеністю даних і може бути застосована до практичного впровадження в органах місцевого самоврядування.

ПЕРЕЛІК ДЖЕРЕЛ

1. Про звернення громадян : Закон України від 02.10.1995 № 393/96-ВР. URL: https://zakon.rada.gov.ua/laws/show/393/96-вр (дата звернення: 12.02.2026).

2. Про захист персональних даних : Закон України від 01.06.2010 № 2297-VI. URL: https://zakon.rada.gov.ua/laws/show/2297-17 (дата звернення: 14.02.2026).

3. Про електронні документи та електронний документообіг : Закон України від 22.05.2003 № 851-IV. URL: https://zakon.rada.gov.ua/laws/show/851-15 (дата звернення: 15.02.2026).

4. Про схвалення Концепції розвитку електронної демократії в Україні та плану заходів щодо її реалізації : Розпорядження Кабінету Міністрів України від 08.11.2017 № 797-р. URL: https://zakon.rada.gov.ua/laws/show/797-2017-р (дата зернення: 15.02.2026).

5. Портал Дія.Цифрова громада : Міністерство цифрової трансформації України. URL: https://hromada.gov.ua/ (дата звернення: 18.02.2026).

6. Платформа електронної демократії e-Dem. URL: https://e-dem.ua/ (дата звернення: 20.02.2026).

7. Платформа СВОЇ : Чат-бот твоєї громади / Програма EGAP. URL: https://egap.in.ua/project/platforma_svoyi (дата звернення: 16.02.2026).

8. Електронна демократія в Україні: сучасний стан та перспективи розвитку // Громадська мережа публічного права та адміністрування UPLAN. URL: https://uplan.org.ua/elektronna-demokratiia-v-ukraini-cuchasnyi-stan-ta-perspektyvy-rozvytku/ (дата звернення: 16.02.2026).

9. PHP MySQL Database Connection and Queries // W3Schools Online Web Tutorials. URL: https://www.w3schools.com/php/php_mysql_intro.asp (дата звернення: 16.02.2026).

10. PHP: Hypertext Preprocessor. Official Documentation. URL: https://www.php.net/docs.php (дата звернення: 25.02.2026).

11. PHP Data Objects (PDO). PHP Manual. URL: https://www.php.net/manual/en/book.pdo.php (дата звернення: 25.02.2026).

12. PDO::prepare – Prepared statements and stored procedures. PHP Manual. URL: https://www.php.net/manual/en/pdo.prepare.php (дата звернення: 26.02.2026).

13. Password Hashing functions in PHP. PHP Manual. URL: https://www.php.net/manual/en/ref.password.php (дата звернення: 26.02.2026).

14. MySQL 8.0 Reference Manual. Oracle. URL: https://dev.mysql.com/doc/refman/8.0/en/ (дата звернення: 01.03.2026).

15. MySQL Transactions and ACID Properties. MySQL Dev. URL: https://dev.mysql.com/doc/refman/8.0/en/mysql-acid.html (дата звернення: 01.03.2026).

16. jQuery API Documentation. The jQuery Foundation. URL: https://api.jquery.com/ (дата звернення: 02.03.2026).

17. jQuery.ajax() Asynchronous Requests. jQuery API. URL: https://api.jquery.com/jquery.ajax/ (дата звернення: 02.03.2026).

18. JavaScript: MDN Web Docs. Mozilla. URL: https://developer.mozilla.org/en-US/docs/Web/JavaScript (дата звернення: 03.03.2026).

19. Сучасний підручник з JavaScript. JavaScript.info. URL: https://uk.javascript.info/ (дата звернення: 03.03.2026).

20. CSS Cascading Style Sheets: MDN Web Docs. Mozilla. URL: https://developer.mozilla.org/en-US/docs/Web/CSS (дата звернення: 04.03.2026).

21. W3C HTML5 Specification. World Wide Web Consortium. URL: https://www.w3.org/TR/html52/ (дата звернення: 04.03.2026).

22. Inter Font Project. Official Website and Repository. URL: https://rsms.me/inter/ (дата звернення: 05.03.2026).

23. Web Content Accessibility Guidelines (WCAG) 2.2. W3C. URL: https://www.w3.org/TR/WCAG22/ (дата звернення: 05.03.2026).

24. OWASP Top 10 Vulnerabilities: SQL Injection and XSS Protection. URL: https://owasp.org/www-project-top-ten/ (дата звернення: 06.03.2026).

25. Unified Modeling Language (UML) Specification. Object Management Group. URL: https://www.uml.org/ (дата звернення: 07.03.2026).

26. Agile Alliance: The Agile Manifesto for Software Development. URL: https://www.agilealliance.org/agile101/the-agile-manifesto/ (дата звернення: 08.03.2026).

27. Model-View-Controller (MVC) Pattern. Microsoft Developer Network. URL: https://learn.microsoft.com/en-us/previous-versions/aspnet/gg416513(v=vs.98 (дата звернення: 09.03.2026).

28. Swiper – The Most Modern Mobile Touch Slider. Documentation. URL: https://swiperjs.com/get-started (дата звернення: 10.03.2026).

29. Font Awesome Icons. Developer Guide. URL: https://fontawesome.com/docs (дата звернення: 10.03.2026).

30. InfinityFree Hosting Documentation and Server Configuration Guide. URL: https://www.infinityfree.com/ (дата звернення: 12.03.2026).

31. ДСанПіН 3.3.2.007-98. Державні санітарні правила та норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин. Київ : МОЗ України, 1998. 24 с.

32. НПАОП 0.00-7.15-18. Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями : затверджено Наказом Мінсоцполітики України від 14.02.2018 № 207.

33. Жидецький В. Ц. Основи охорони праці : підручник. Львів : Афіша, 2019. 348 с.

34. Кодекс цивільного захисту України : Закон України від 02.10.2012 № 5403-VI. URL: https://zakon.rada.gov.ua/laws/show/5403-17 (дата звернення: 15.03.2026).

35. ДСТУ EN 3-7:2014. Вогнегасники переносні. Частина 7. Характеристики, вимоги до функціонування та методи випробування. Київ : ДП «УкрНДНЦ», 2015. 42 с.

36. Небесний Р. М. Рекомендаційна система формування команд виконавців з відповідними фаховими компетентностями : дис. ... канд. техн. наук (або: д-ра філософії) / Руслан Михайлович Небесний ; Тернопільський національний технічний університет імені Івана Пулюя. Тернопіль, 2023. URL: https://elartu.tntu.edu.ua/handle/lib/43005 (дата звернення: 17.03.2026).

37. Information technology of personalized choice of profession in smart cities / N. E. Kunanets, M. V. Nazaruk, R. M. Nebesnyi, V. V. Pasichnyk. Інформаційні технології і засоби навчання. 2018. Т. 65, № 3. С. 277-290.

38. Cloud-based IT Infrastructure for “Smart City” Projects / O. Duda, N. Kunanets, O. Matsiuk, V. Pasichnyk // Dependable IoT for Human and Industry: Modeling, Architecting, Implementation. – River Publishers, 2018. – P. 389-410.

39. Баран І., Дуда О., Маєвський О. Розширення функціональних можливостей PHP для перевірки отриманих від користувача даних. Інформаційні моделі, системи та технології : матеріали IV Науково-технічної конференції. сторінки. С. 37.

40. Smart people: the role of big data analytics in digital transformation / R. Nebesnyi, O. Palka, L. Dmytrotsa, H. Kozbur. BAITmp 2025: The 2nd International Workshop on Bioinformatics and Applied Information Technologies for medical purpose 2025. 2026. Vol. 2. P. 163–174. URL: https://ceur-ws.org/Vol-4159/paper14.pdf (дата звернення: 17.03.2026).

41. Portfolio project management / R. Nebesnyi, N. Kunanets, N. Veretennikova, R. Vaskiv, Z. Haladzhun, M. Graca. ITPM. 2024. P. 141–152. URL: https://ceur-ws.org/Vol-3709/paper12.pdf (дата звернення: 17.03.2026).

ДОДАТКИ

Додаток А

Програмні скрипти серверної частини вебплатформи

```
// db.php
ini_set('display_errors', 1);
ini_set('display_startup_errors', 1);
error_reporting(E_ALL);
if (session_status() === PHP_SESSION_NONE) {
    session_start();
}

$host = 'sql303.infinityfree.com';
$db   = 'if0_42178872_edem_db';
$user = 'if0_42178872';
$pass = 'nfFAYzwVsE2';
$charset = 'utf8mb4';

$dsn = "mysql:host=$host;dbname=$db;charset=$charset";
$opt = [
    PDO::ATTR_ERRMODE            => PDO::ERRMODE_EXCEPTION,
    PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
    PDO::ATTR_EMULATE_PREPARES   => false,
];

try {
    $pdo = new PDO($dsn, $user, $pass, $opt);
} catch (\PDOException $e) {
    throw new \PDOException($e->getMessage(), (int)$e->getCode());
}

function getUnreadRepliesCount($pdo, $userId) {
    if (!$userId) return 0;
    $stmt = $pdo->prepare("SELECT COUNT(*) FROM feedback WHERE
user_id = ? AND admin_reply IS NOT NULL");
    $stmt->execute([$userId]);
    return $stmt->fetchColumn();
}

// admin.php
<?php
require 'db.php';
if (!isset($_SESSION['user_role']) || $_SESSION['user_role'] !==
'admin') {
    die("Доступ заборонено");
}

if ($_SERVER['REQUEST_METHOD'] == 'POST' &&
isset($_POST['new_news'])) {
    $title = $_POST['title'];
    $content = $_POST['content'];
```

```
    $imageName = null;

    if (isset($_FILES['image']) && $_FILES['image']['error'] ===
UPLOAD_ERR_OK) {
        $fileTmpPath = $_FILES['image']['tmp_name'];
        $fileName = $_FILES['image']['name'];
        $fileExtension       =       strtolower(pathinfo($fileName,
PATHINFO_EXTENSION));

        $imageName = time() . '_' . md5($fileName) . '.' .
$fileExtension;

        $uploadFileDir = './uploads/';
        if(!is_dir($uploadFileDir)){
            mkdir($uploadFileDir, 0755, true);
        }

        $dest_path = $uploadFileDir . $imageName;
        move_uploaded_file($fileTmpPath, $dest_path);
    }

    $stmt = $pdo->prepare("INSERT INTO news (title, content, image)
VALUES (?, ?, ?)");
    $stmt->execute([$title, $content, $imageName]);
    $msg = "Новину опубліковано!";
}

if       ($_SERVER['REQUEST_METHOD']       ==       'POST'       &&
isset($_POST['reply_feedback'])) {
    $feedbackId = $_POST['feedback_id'];
    $replyText = trim($_POST['admin_reply']);

    $stmt = $pdo->prepare("UPDATE feedback SET admin_reply = ?,
replied_at = NOW() WHERE id = ?");
    $stmt->execute([$replyText, $feedbackId]);
    $msg_reply = "Відповідь успішно надіслано користувачеві!";
}
?>
<!DOCTYPE html>
<html lang="uk">
<head>
    <title>Адмін Панель</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <header><div     class="container"><div     class="logo">Панель
Адміністратора</div><nav><a    href="index.php">На    сайт</a>    <a
href="logout.php">Вихід</a></nav></div></header>
    <div class="container">
        <div class="admin-panel" style="margin-top:30px;">
            <h2>Додати новину</h2>
            <?php if(isset($msg)) echo "<p style='color:green; font-
weight:bold;'>$msg</p>"; ?>
```

```
            <form    method="POST"    enctype="multipart/form-data"
class="card">
                <input type="hidden" name="new_news" value="1">
                <div class="form-group">
                    <label>Заголовок новини</label>
                    <input type="text" name="title" required>
                </div>
                <div class="form-group">
                    <label>Головне фото новини</label>
                    <input         type="file"         name="image"
accept="image/*" required>
                </div>
                <div class="form-group">
                    <label>Текст новини</label>
                    <textarea        name="content"        rows="6"
required></textarea>
                </div>
                <button    type="submit"    class="btn">Опублікувати
новину</button>
            </form>
        </div>

        <h2    style="margin-top:50px;">Вхідні    повідомлення    та
скарги</h2>
        <?php  if(isset($msg_reply))  echo  "<p  style='color:green;
font-weight:bold;'>$msg_reply</p>"; ?>

        <?php
        $feed = $pdo->query("SELECT f.*, p.title as p_title FROM
feedback f LEFT JOIN petitions p ON f.petition_id = p.id ORDER BY
f.created_at DESC");
        while($row = $feed->fetch()):
            $typeClass   =   ($row['type']   ==   'complaint')   ?
'style="border-left:  4px  solid  red;  background:#fff0f0;"'  :
'style="border-left: 4px solid green;"';
        ?>
            <div class="card" <?php echo $typeClass; ?>>
                <small><?php echo $row['created_at']; ?> |
                    <b><?php echo ($row['type'] == 'complaint') ? '
СКАРГА' : ' ПОВІДОМЛЕННЯ'; ?></b>
                </small>
                <?php if($row['petition_id']): ?>
                    <p>Стосується   петиції:   <b>ID   <?php   echo
$row['petition_id'];           ?>           (<?php           echo
htmlspecialchars($row['p_title']); ?>)</b></p>
                <?php endif; ?>
                <p><strong>Повідомлення:</strong>    <?php    echo
nl2br(htmlspecialchars($row['message'])); ?></p>
                <hr>
                <?php if($row['admin_reply']): ?>
                    <div style="background: #e0f2fe; padding: 10px;
border-radius: 6px; margin-top: 10px;">
```

```
                        <strong>Ваша відповідь (<?php echo
$row['replied_at']; ?>):</strong>
                        <p style="margin: 5px 0 0 0;"><?php echo
nl2br(htmlspecialchars($row['admin_reply'])); ?></p>
                    </div>
                <?php else: ?>
                    <form method="POST" style="margin-top:15px;">
                        <input type="hidden" name="reply_feedback"
value="1">
                        <input type="hidden" name="feedback_id"
value="<?php echo $row['id']; ?>">
                        <div class="form-group">
                            <label style="font-weight:600; font-
size:0.9em;">Написати відповідь користувачеві:</label>
                            <textarea name="admin_reply" rows="2"
style="width:100%; border:1px solid #ccc; border-radius:4px;"
required></textarea>
                        </div>
                        <button type="submit" class="btn"
style="padding:5px 15px; font-size:0.9em;
background:#0284c7;">Надіслати відповідь</button>
                    </form>
                <?php endif; ?>
            </div>
        <?php endwhile; ?>
    </div>
</body>
</html>

// moderator.php
    <?php
    require 'db.php';
    if (!isset($_SESSION['user_role']) || $_SESSION['user_role']
    !== 'moderator') {
        die("Доступ заборонено");
    }

    if (isset($_POST['action_petition']) &&
    isset($_POST['petition_id'])) {
        $status = ($_POST['action_petition'] == 'approve') ?
    'active' : 'rejected';
        $stmt = $pdo->prepare("UPDATE petitions SET status = ? WHERE
    id = ?");
        $stmt->execute([$status, $_POST['petition_id']]);
        header("Location: moderator.php");
        exit;
    }

    if (isset($_POST['action_user']) &&
    isset($_POST['profile_id'])) {
        $status = ($_POST['action_user'] == 'approve') ? 'approved'
    : 'rejected';
```

```
    $fileStmt = $pdo->prepare("SELECT document FROM users WHERE
id = ?");
    $fileStmt->execute([$userId]);
    $userFile = $fileStmt->fetchColumn();
    $stmt = $pdo->prepare("UPDATE users SET is_verified = ?
WHERE id = ?");
    $stmt->execute([$status, $_POST['profile_id']]);
    if ($userFile && file_exists('./uploads/documents/' .
$userFile)) {
        unlink('./uploads/documents/' . $userFile);
    }
    header("Location: moderator.php?tab=users");
    exit;
}

$activeTab = $_GET['tab'] ?? 'petitions';
?>
<!DOCTYPE html>
<html lang="uk">
<head>
    <title>Кабінет Модератора</title>
    <link rel="stylesheet" href="style.css">
    <link                                          rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.4.0/css/all.min.css"/>
</head>
<body>
    <header>
        <div class="container">
            <a                                     href="index.php"
class="logo"><span>Gromada</span>.Connect</a>
            <nav>
                <a           href="moderator.php?tab=petitions"
style="<?php if($activeTab=='petitions') echo 'color: var(--
primary-color);   border-bottom:   2px   solid;';   ?>">Черга
петицій</a>
                <a href="moderator.php?tab=users" style="<?php
if($activeTab=='users')  echo  'color:  var(--primary-color);
border-bottom: 2px solid;'; ?>">Перевірка мешканців</a>
                <a   href="logout.php"   class="nav-btn   btn-
primary-nav">Вихід</a>
            </nav>
        </div>
    </header>

    <div  class="container"  style="margin-top:  40px;  margin-
bottom: 60px;">

        <?php if($activeTab === 'petitions'): ?>
            <h1    style="margin-bottom:    30px;">Петиції    на
перевірку</h1>
            <?php
```

```
            $stmt = $pdo->query("SELECT * FROM petitions WHERE
status = 'pending' ORDER BY created_at ASC");
            $petitions = $stmt->fetchAll();

            if(count($petitions) == 0) echo "<div class='card'
style='text-align:center;       color:var(--text-muted);'>Черга
петицій пуста.</div>";

            foreach($petitions as $row):
            ?>
            <div class="card" style="padding: 30px; background:
#FFFFFF;">
                <h3 style="color: var(--primary-color);"><?php
echo htmlspecialchars($row['title']); ?></h3>
                <p  style="white-space:  pre-line;  margin-top:
15px;"><?php echo nl2br(htmlspecialchars($row['description']));
?></p>
                <div  style="margin-top:  25px;  display:  flex;
gap: 15px;">
                    <form                          method="POST"
style="display:inline;">
                        <input                     type="hidden"
name="action_petition" value="approve">
                        <input type="hidden" name="petition_id"
value="<?php echo $row['id']; ?>">
                        <button  type="submit"  class="btn  btn-
success"><i class="fa-solid fa-check"></i> Схвалити</button>
                    </form>
                    <form                          method="POST"
style="display:inline;">
                        <input                     type="hidden"
name="action_petition" value="reject">
                        <input type="hidden" name="petition_id"
value="<?php echo $row['id']; ?>">
                        <button  type="submit"  class="btn  btn-
danger"><i class="fa-solid fa-xmark"></i> Відхилити</button>
                    </form>
                </div>
            </div>
            <?php endforeach; ?>

        <?php else: ?>
            <h1    style="margin-bottom:    30px;">Заявки    на
підтвердження проживання</h1>
            <?php
            $stmt = $pdo->query("SELECT * FROM users WHERE role
= 'user' AND is_verified = 'pending' ORDER BY created_at ASC");
            $unverifiedUsers = $stmt->fetchAll();

            if(count($unverifiedUsers)   ==   0)   echo   "<div
class='card'     style='text-align:center;     color:var(--text-
muted);'>Нових заявок на верифікацію немає.</div>";
```

```
            foreach($unverifiedUsers as $userRow):
            ?>
            <div class="card" style="padding: 30px; background:
#FFFFFF;">
                <h3 style="margin-bottom: 5px;"><?php echo
htmlspecialchars($userRow['name']); ?></h3>
                <p style="color: var(--text-muted); font-size:
0.9rem; margin-bottom: 15px;">Електронна пошта: <strong><?php
echo htmlspecialchars($userRow['email']); ?></strong> | Дата
реєстрації: <?php echo $userRow['created_at']; ?></p>

                <div style="background: #f1f5f9; padding: 15px;
border-radius: var(--radius-subcard); margin-bottom: 20px;
display: flex; align-items: center; justify-content: space-
between;">
                    <span><i class="fa-solid fa-file-invoice"
style="margin-right: 8px; color: var(--primary-color);"></i>
Наданий документ підтвердження прописки</span>
                    <a href="uploads/documents/<?php echo
$userRow['document']; ?>" target="_blank" class="btn"
style="background: var(--text-main); font-size: 0.85rem;
padding: 6px 16px;"><i class="fa-solid fa-eye" style="margin-
right: 6px;"></i> Переглянути документ</a>
                </div>

                <div style="display: flex; gap: 15px;">
                    <form method="POST"
style="display:inline;">
                        <input type="hidden" name="action_user"
value="approve">
                        <input type="hidden" name="profile_id"
value="<?php echo $userRow['id']; ?>">
                        <button type="submit" class="btn btn-
success"><i class="fa-solid fa-user-check"></i> Підтвердити
особу</button>
                    </form>
                    <form method="POST"
style="display:inline;">
                        <input type="hidden" name="action_user"
value="reject">
                        <input type="hidden" name="profile_id"
value="<?php echo $userRow['id']; ?>">
                        <button type="submit" class="btn btn-
danger"><i class="fa-solid fa-user-slash"></i> Відхилити (Не
житель)</button>
                    </form>
                </div>
            </div>
            <?php endforeach; ?>
        <?php endif; ?>

    </div>
</body>
```

```
</html>

// feedback_handler.php
<?php
require 'db.php';

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $type = $_POST['type'] ?? 'general';
    $msg = trim($_POST['message']);
    $contact = $_POST['contact'] ?? null;
    $pet_id = $_POST['petition_id'] ?? null;
    $user_id = $_SESSION['user_id'] ?? null;

    if ($contact) {
        $msg = "Від: $contact\n\n" . $msg;
    }

    $stmt = $pdo->prepare("INSERT INTO feedback (user_id, type,
petition_id, message) VALUES (?, ?, ?, ?)");
    $stmt->execute([$user_id, $type, $pet_id, $msg]);

    header("Location: " . $_SERVER['HTTP_REFERER']);
    exit;
}
?>

// vote.php
<?php
require 'db.php';

header('Content-Type: application/json');

if ($_SERVER['REQUEST_METHOD'] !== 'POST') {
    echo json_encode(['status' => 'error', 'message' => 'Invalid
request']);
    exit;
}

if (!isset($_SESSION['user_id'])) {
    echo json_encode(['status' => 'error', 'message' => 'Необхідна
авторизація']);
    exit;
}

$userId = $_SESSION['user_id'];
$petitionId = $_POST['petition_id'] ?? 0;

$stmt = $pdo->prepare("SELECT id FROM votes WHERE user_id = ? AND
petition_id = ?");
$stmt->execute([$userId, $petitionId]);
if ($stmt->fetch()) {
    echo json_encode(['status' => 'error', 'message' => 'Ви вже
голосували']);
```

```
    exit;
}

try {
    $pdo->beginTransaction();

    $stmt = $pdo->prepare("INSERT INTO votes (user_id, petition_id)
VALUES (?, ?)");
    $stmt->execute([$userId, $petitionId]);

    $stmt = $pdo->prepare("UPDATE petitions SET votes_count =
votes_count + 1 WHERE id = ?");
    $stmt->execute([$petitionId]);

    $pdo->commit();
    echo json_encode(['status' => 'success']);
} catch (Exception $e) {
    $pdo->rollBack();
    echo json_encode(['status' => 'error', 'message' => 'Помилка
бази даних']);
}

// fix_users.php
<?php
require 'db.php';

$pass_admin = 'admin123';
$pass_mod   = 'mod123';

$hash_admin = password_hash($pass_admin, PASSWORD_DEFAULT);
$hash_mod   = password_hash($pass_mod, PASSWORD_DEFAULT);

try {
    $pdo->exec("DELETE FROM users WHERE email = 'admin@gmail.com'");
    $pdo->exec("DELETE FROM users WHERE email = 'mod@gmail.com'");

    $stmt = $pdo->prepare("INSERT INTO users (name, email, password,
role) VALUES (?, ?, ?, ?)");

    $stmt->execute(['Головний Адмін', 'admin@gmail.com',
$hash_admin, 'admin']);
    echo "Адміністратора створено успішно!<br>";

    $stmt->execute(['Модератор Петицій', 'mod@gmail.com',
$hash_mod, 'moderator']);
    echo "Модератора створено успішно!<br>";

} catch (PDOException $e) {
    echo "Помилка бази даних: " . $e->getMessage();
}
?>
<br>
<a href="login.php">Перейти на сторінку входу</a>
```

Додаток Б

Користувацькі інтерфейси платформи та скрипти взаємодії

```
// index.php
<?php
require 'db.php';
$userId = $_SESSION['user_id'] ?? null;
$unreadCount = getUnreadRepliesCount($pdo, $userId);
?>
<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <title>Громада.Connect | Головна</title>
    <link rel="stylesheet" href="style.css">
    <link                                                rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/swiper@11/swiper-
bundle.min.css"/>
    <link                                                rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.4.0/css/all.min.css"/>
</head>
<body>
    <header>
    <div    class="container"    style="display:flex;    justify-
content:space-between; align-items:center;">
        <a                                                   href="index.php"
class="logo"><span>Gromada</span>.Connect</a>
        <nav style="display:flex; align-items:center; gap:15px;">
            <?php        if(isset($_SESSION['user_role'])        &&
$_SESSION['user_role'] == 'admin'): ?>
                <a    href="admin.php"    class="nav-btn    btn-
danger">Адмін-панель</a>
            <?php endif; ?>
            <?php        if(isset($_SESSION['user_role'])        &&
$_SESSION['user_role'] == 'moderator'): ?>
                <a   href="moderator.php"   class="nav-btn   btn-
warning">Модерація</a>
            <?php endif; ?>

            <a href="petitions.php">Петиції</a>
            <a href="news.php">Новини</a>

            <?php if($userId): ?>
                <a  href="my_replies.php"  title="Мої  повідомлення"
style="position: relative; font-size: 1.3rem; color: var(--primary-
color);">
                    <i class="fa-solid fa-bell"></i>
                    <?php if($unreadCount > 0): ?>
```

```
                        <span style="position:absolute; top:-8px;
right:-8px;   background:red;   color:white;   border-radius:50%;
padding:2px 6px; font-size:0.7rem; font-weight:bold;"><?php echo
$unreadCount; ?></span>
                    <?php endif; ?>
                </a>

                <a href="create.php" class="nav-btn btn-primary-
nav">Створити +</a>
                <a  href="logout.php"  style="color:  var(--text-
muted); font-size: 0.9em;">Вихід</a>
            <?php else: ?>
                <a href="login.php">Вхід</a>
                <a href="register.php" class="nav-btn btn-primary-
nav">Реєстрація</a>
            <?php endif; ?>
        </nav>
    </div>
</header>

<div class="container">
    <?php if(isset($_GET['msg']) && $_GET['msg'] == 'moderation'):
?>
        <div  class="alert"  style="background:  #eff6ff;  color:
#1e40af; border: 1px solid #bfdbfe; margin-top:20px;">
            Дякуємо! Вашу петицію відправлено на модерацію. Вона
з'явиться після перевірки.
        </div>
    <?php endif; ?>

    <h2 style="margin-top: 30px; margin-bottom: 20px;">Останні події
громади</h2>
    <div class="swiper mySwiper" style="width: 100%; height: 350px;
border-radius: 12px; margin-bottom: 40px;">
        <div class="swiper-wrapper">
            <?php
            $newsStmt = $pdo->query("SELECT * FROM news ORDER BY
created_at DESC LIMIT 10");
            if($newsStmt->rowCount() > 0):
                while($news = $newsStmt->fetch()):
                    $img   =   $news['image']   ?   'uploads/'   .
$news['image']                                                  :
'https://via.placeholder.com/800x400?text=No+Image';
            ?>
                <div     class="swiper-slide"     style="position:
relative; background-image: url('<?php echo $img; ?>'); background-
size: cover; background-position: center;">
                    <div style="position: absolute; bottom: 0; left:
0;    right:    0;    background:    linear-gradient(transparent,
rgba(0,0,0,0.8)); color: white; padding: 30px 20px;">
                        <span  style="font-size:  0.85em;  opacity:
0.8;"><?php  echo  date('d.m.Y',  strtotime($news['created_at']));
?></span>
```

```
                        <h3 style="margin: 5px 0 10px 0; font-size:
1.6rem;">
                            <a  href="news_view.php?id=<?php  echo
$news['id']; ?>" style="color: white; text-decoration: none;">
                                <?php                            echo
htmlspecialchars($news['title']); ?>
                            </a>
                        </h3>
                        <p  style="margin:  0;  opacity:  0.9;"><?php
echo mb_substr(strip_tags($news['content']), 0, 100) . '...'; ?></p>
                    </div>
                </div>
            <?php
                endwhile;
            else:
            ?>
                <div   class="swiper-slide"   style="display:flex;
align-items:center;   justify-content:center;   background:#f3f4f6;
color:#6b7280;">
                    <h3>Наразі новин немає.</h3>
                </div>
            <?php endif; ?>
        </div>
        <div class="swiper-pagination"></div>
        <div class="swiper-button-next"></div>
        <div class="swiper-button-prev"></div>
    </div>

    <div style="display:flex; justify-content:space-between; align-
items:center; margin-top: 40px; margin-bottom: 20px;">
        <h2>Популярні петиції</h2>
        <a  href="petitions.php"  class="btn"  style="background-
color: transparent; border: 2px solid var(--accent-color); color:
var(--accent-color);">Переглянути всі петиції</a>
    </div>

    <?php
    $stmt  =  $pdo->query("SELECT  *  FROM  petitions  WHERE  status  =
'active' ORDER BY votes_count DESC LIMIT 3");
    if ($stmt->rowCount() > 0):
        while ($row = $stmt->fetch()):
            $goal = 500;
            $percent  =  min(100,  ($row['votes_count']  /  $goal)  *
100);
    ?>
        <div class="card">
            <div  style="display:  flex;  justify-content:  space-
between;">
                <span  class="badge  badge-active">Популярна  (Збір
підписів)</span>
                <span  style="color:  var(--text-muted);  font-size:
0.9em;"><?php  echo  date('d.m.Y',  strtotime($row['created_at']));
?></span>
```

```
            </div>
            <h3 style="margin-top: 10px;">
                <a  href="view.php?id=<?php  echo  $row['id'];  ?>"
style="text-decoration: none; color: inherit;">
                    <?php echo htmlspecialchars($row['title']); ?>
                </a>
            </h3>
            <p><?php                                                echo
mb_substr(htmlspecialchars($row['description']),  0,  200)  .  '...';
?></p>
            <div style="background: #e5e7eb; height: 8px; border-
radius: 4px; margin: 15px 0; overflow: hidden;">
                <div    style="background:    var(--accent-color);
height: 100%; width: <?php echo $percent; ?>%;"></div>
            </div>
            <div  style="display:  flex;  justify-content:  space-
between; align-items: center; margin-top: 15px;">
                <div style="font-weight: 600; color: var(--primary-
color);">
                    <?php   echo   $row['votes_count'];   ?>   <span
style="font-weight:   400;   color:   var(--text-muted);">/   500
голосів</span>
                </div>
                <a  href="view.php?id=<?php  echo  $row['id'];  ?>"
class="btn">Підтримати</a>
            </div>
        </div>
    <?php
        endwhile;
    else:
    ?>
        <div class="alert" style="background: #f3f4f6; border: 1px
dashed #d1d5db; text-align: center;">
            <p>Наразі немає активних петицій.</p>
        </div>
    <?php endif; ?>

    <footer>
        <div class="container">
            <div class="footer-content">
                <div class="footer-info">
                    <h3><span>Gromada</span>.Connect</h3>
                    <p>Єдина платформа електронної демократії для
нашої громади.</p>
                    <div class="footer-contacts" style="margin-top:
25px;">
                        <p><i class="fa-solid fa-location-dot"></i>
вул. Незалежності, 1</p>
                        <p><i   class="fa-solid   fa-envelope"></i>
admin@gmail.com</p>
                        <p><i class="fa-solid fa-phone"></i> 0-800-
500-000</p>
                    </div>
```

```
                </div>
                <div class="footer-form">
                    <h4>Написати адміністрації</h4>
                    <form             action="feedback_handler.php"
method="POST">
                        <input      type="hidden"      name="type"
value="general">
                        <div  style="display:  grid;  grid-template-
columns: 1fr 1fr; gap: 10px;">
                            <input    type="text"    name="contact"
placeholder="Ваше ім'я" required>
                            <input                        type="email"
name="email_contact" placeholder="Email">
                        </div>
                        <textarea      name="message"      rows="3"
placeholder="Ваше повідомлення..." required></textarea>
                        <button type="submit">Надіслати</button>
                    </form>
                </div>
            </div>
            <div class="footer-bottom">&copy; <?php echo date('Y');
?> Gromada.Connect.</div>
        </div>
    </footer>
</div>

<script         src="https://cdn.jsdelivr.net/npm/swiper@11/swiper-
bundle.min.js"></script>
<script>
    var swiper = new Swiper(".mySwiper", {
        loop: true,
        pagination: { el: ".swiper-pagination", clickable: true },
        navigation:   {   nextEl:   ".swiper-button-next",   prevEl:
".swiper-button-prev" },
        autoplay: { delay: 4000, disableOnInteraction: false }
    });
</script>
</body>
</html>

// petitions.php
<?php
require 'db.php';

$search = isset($_GET['search']) ? trim($_GET['search']) : '';
$status = isset($_GET['status']) ? trim($_GET['status']) : '';
$sort = isset($_GET['sort']) ? trim($_GET['sort']) : 'latest';

$queryStr = "SELECT p.*, u.name as author_name,
            (SELECT  COUNT(*)  FROM  votes  v  WHERE  v.petition_id  =
p.id) as current_votes
            FROM petitions p
            JOIN users u ON p.user_id = u.id
```

```
            WHERE 1=1";

$params = [];

if ($search !== '') {
    $queryStr .= " AND (p.title LIKE ? OR p.description LIKE ?)";
    $params[] = "%$search%";
    $params[] = "%$search%";
}

if ($status !== '') {
    $queryStr .= " AND p.status = ?";
    $params[] = $status;
} else {
    $queryStr .= " AND p.status != 'rejected'";
}

if ($sort === 'votes') {
    $queryStr .= " ORDER BY current_votes DESC, p.created_at DESC";
} else {
    $queryStr .= " ORDER BY p.created_at DESC";
}

$stmt = $pdo->prepare($queryStr);
$stmt->execute($params);
$petitions = $stmt->fetchAll();

$required_votes = 500;
?>
<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <title>Всі петиції | Gromada.Connect</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
<header>
    <div class="container">
        <a                                          href="index.php"
class="logo"><span>Gromada</span>.Connect</a>
        <nav>
            <a href="index.php">Головна</a>
            <a  href="petitions.php"  style="color:  var(--primary-
color);">Петиції</a>
            <a href="news.php">Новини</a>
            <?php if (isset($_SESSION['user_id'])): ?>
                <a  href="create.php"  class="nav-btn  btn-primary-
nav">+ Створити петицію</a>
            <?php else: ?>
                <a  href="login.php"  class="nav-btn  btn-primary-
nav">Увійти</a>
            <?php endif; ?>
```

```
        </nav>
    </div>
</header>

<div class="container">
    <div class="main-layout-grid">
        <div class="petitions-list-section">
            <h2><?php echo ($search !== '') ? 'Результати для: "' .
htmlspecialchars($search) . '"' : 'Електронні петиції громади';
?></h2>
            <?php if (empty($petitions)): ?>
                <p>Петицій не знайдено.</p>
            <?php else: ?>
                <?php foreach ($petitions as $petition):
                    $current_votes = $petition['current_votes'];
                    $percent = min(100, round(($current_votes /
$required_votes) * 100));
                    $badgeClass = ($petition['status'] ===
'pending') ? 'badge-pending' : 'badge-active';
                    $statusName = ($petition['status'] ===
'pending') ? 'На модерації' : 'Активна';
                ?>
                    <div class="card">
                        <span class="badge <?php echo $badgeClass;
?>"><?php echo $statusName; ?></span>
                        <h3><a href="view.php?id=<?php echo
$petition['id']; ?>"><?php echo
htmlspecialchars($petition['title']); ?></a></h3>
                        <p><?php echo
mb_strimwidth(htmlspecialchars($petition['description']), 0, 160,
"..."); ?></p>
                        <div>
                            <div style="background: #e5e7eb;
height: 6px; width: 100%;">
                                <div style="background: var(--
primary-color); height: 100%; width: <?php echo $percent;
?>%;"></div>
                            </div>
                        </div>
                        <span>Автор: <strong><?php echo
htmlspecialchars($petition['author_name']); ?></strong></span>
                    </div>
                <?php endforeach; ?>
            <?php endif; ?>
        </div>

        <div class="sidebar-filters-section">
            <h2>Фільтрація</h2>
            <form method="get" class="filters-sidebar">
                <input type="text" name="search"
placeholder="Пошук..." value="<?php echo htmlspecialchars($search);
?>">
                <select name="status">
```

```
                    <option value="">Всі</option>
                    <option value="active" <?php echo ($status ===
'active') ? 'selected' : ''; ?>>Активні</option>
                    <option value="pending" <?php echo ($status ===
'pending') ? 'selected' : ''; ?>>На модерації</option>
                </select>
                <select name="sort">
                    <option value="latest" <?php echo ($sort ===
'latest') ? 'selected' : ''; ?>>Нові</option>
                    <option value="votes" <?php echo ($sort ===
'votes') ? 'selected' : ''; ?>>За голосами</option>
                </select>
                <button type="submit" class="btn">Пошук</button>
            </form>
        </div>
    </div>
</div>
</body>
</html>

// view.php
<?php
require 'db.php';
$id = $_GET['id'] ?? 0;
$stmt = $pdo->prepare("SELECT * FROM petitions WHERE id = ?");
$stmt->execute([$id]);
$petition = $stmt->fetch();

if (!$petition) die("Петицію не знайдено");

$hasVoted = false;
if (isset($_SESSION['user_id'])) {
    $vCheck = $pdo->prepare("SELECT id FROM votes WHERE user_id = ?
AND petition_id = ?");
    $vCheck->execute([$_SESSION['user_id'], $id]);
    if ($vCheck->fetch()) $hasVoted = true;
}
?>
<!DOCTYPE html>
<html lang="uk">
<head>
    <title><?php     echo     htmlspecialchars($petition['title']);
?></title>
    <link rel="stylesheet" href="style.css">
    <script                    src="https://code.jquery.com/jquery-
3.6.0.min.js"></script>
    <script src="script.js"></script>
</head>
<body>
    <header>
    <div class="container">
        <a                                          href="index.php"
class="logo"><span>Gromada</span>.Connect</a>
```

```
        <nav><a href="index.php">Всі петиції</a></nav>
    </div>
</header>
    <div class="container" style="margin-top: 30px;">
        <div class="card">
            <h1><?php  echo  htmlspecialchars($petition['title']);
?></h1>
            <p><?php                                              echo
htmlspecialchars($petition['description']); ?></p>
            <hr>
            <h3>Зібрано підписів: <span id="votes-count"><?php echo
$petition['votes_count']; ?></span></h3>

            <?php if (isset($_SESSION['user_id'])): ?>
                <?php if ($petition['status'] == 'active'): ?>
                    <?php if ($hasVoted): ?>
                        <button      class="btn      btn-disabled"
disabled>Вже підтримано</button>
                    <?php else: ?>
                        <button  id="vote-btn"  class="btn"  data-
id="<?php echo $petition['id']; ?>">Підписати петицію</button>
                    <?php endif; ?>
                <?php else: ?>
                    <p><em>Голосування недоступне</em></p>
                <?php endif; ?>

                <hr style="margin-top: 20px;">
                <form  action="feedback_handler.php"  method="POST"
class="complaint-section">
                    <input         type="hidden"         name="type"
value="complaint">
                    <input     type="hidden"     name="petition_id"
value="<?php echo $petition['id']; ?>">
                    <textarea  name="message"  placeholder="Опишіть
причину скарги..." required></textarea>
                    <button    type="submit"    class="btn-outline-
danger">Надіслати скаргу</button>
                </form>
            <?php else: ?>
                <p><i>Щоб                підписати,                <a
href="login.php">увійдіть</a>.</i></p>
            <?php endif; ?>
        </div>
    </div>
</body>
</html>

// create.php
<?php
require 'db.php';
if (!isset($_SESSION['user_id'])) { header("Location: login.php");
exit; }
```

```
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $title = trim($_POST['title']);
    $desc = trim($_POST['description']);

    $stmt = $pdo->prepare("INSERT INTO petitions (user_id, title,
description) VALUES (?, ?, ?)");
    $stmt->execute([$_SESSION['user_id'], $title, $desc]);

    header("Location: index.php?msg=moderation");
    exit;
}
?>
<!DOCTYPE html>
<html lang="uk">
<head>
    <title>Створення петиції | Gromada.Connect</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
<header>
    <div            class="container"><a            href="index.php"
class="logo"><span>Gromada</span>.Connect</a></div>
</header>
<div class="container" style="max-width: 800px; margin: 40px auto;">
    <h1>Створення петиції</h1>
    <div class="card" style="padding: 40px; background: #FFFFFF;">
        <div class="alert-moderation">Увага! Ваша петиція з'явиться
на сайті тільки після перевірки.</div>
        <form method="post">
            <div class="form-group">
                <label>Заголовок петиції</label>
                <input     type="text"     name="title"     required
style="width:100%;">
            </div>
            <div class="form-group">
                <label>Текст петиції</label>
                <textarea        name="description"        required
style="width:100%; min-height:200px;"></textarea>
            </div>
            <button    type="submit"    class="btn">Відправити    на
перевірку</button>
        </form>
    </div>
</div>
</body>
</html>

// login.php
    <?php
    require 'db.php';
    $error = '';
    $info_msg = '';
```

```
if (isset($_GET['msg']) && $_GET['msg'] == 'reg_success') {
    $info_msg = "Вашу заявку на реєстрацію прийнято! Доступ до акаунта з'явиться після того, як модератор підтвердить ваше проживання в громаді.";
}

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $email = trim($_POST['email']);
    $password = $_POST['password'];

    $stmt = $pdo->prepare("SELECT * FROM users WHERE email = ?");
    $stmt->execute([$email]);
    $user = $stmt->fetch();

    if ($user && password_verify($password, $user['password'])) {

        if ($user['role'] == 'user' && $user['is_verified'] == 'pending') {
            $error = "Ваш акаунт ще перебуває на перевірці у модератора. Будь ласка, зачекайте.";
        }    elseif    ($user['role']    ==    'user'    &&    $user['is_verified'] == 'rejected') {
            $error = "Модератор відхилив вашу заявку. Наданий документ не підтверджує ваше проживання в межах цієї громади.";
        } else {

            $_SESSION['user_id'] = $user['id'];
            $_SESSION['user_name'] = $user['name'];
            $_SESSION['user_role'] = $user['role'];

            if ($user['role'] == 'admin') {
                header("Location: admin.php");
            } elseif ($user['role'] == 'moderator') {
                header("Location: moderator.php");
            } else {
                header("Location: index.php");
            }
            exit;
        }
    } else {
        $error = "Невірний логін або пароль";
    }
}
?>
<!DOCTYPE html>
<html lang="uk">
<head><title>Вхід        |        Громада.Connect</title><link rel="stylesheet" href="style.css"></head>
<body>
    <div class="container" style="max-width: 450px; margin-top: 80px;">
```

```
        <h2    style="text-align:    center;    margin-bottom:
    25px;">Вхід до системи</h2>

        <?php if($info_msg): ?>
            <div   class="alert"   style="background:   #eff6ff;
    color: #1e40af; border: 1px solid #bfdbfe; font-size: 0.95rem;
    line-height: 1.5; display: block;">
                <?php echo $info_msg; ?>
            </div>
        <?php endif; ?>

        <?php  if($error):  ?><div  class="alert  alert-error"
    style="display:block"><?php echo $error; ?></div><?php endif; ?>

        <form   method="post"   class="card"   style="padding:
    30px;">
            <div class="form-group"><label>E-mail</label><input
    type="email" name="email" required></div>
            <div class="form-group"><label>Пароль</label><input
    type="password" name="password" required></div>
            <button  type="submit"  class="btn"  style="width:
    100%; margin-top: 10px;">Увійти</button>
        </form>
        <p  style="text-align:  center;"><a  href="index.php">На
    головну</a> | <a href="register.php">Створити акаунт</a></p>
    </div>
</body>
</html>

// register.php
    <?php
    require 'db.php';
    $error = '';

    if ($_SERVER['REQUEST_METHOD'] == 'POST') {
        $name = trim($_POST['name']);
        $email = trim($_POST['email']);
        $password = $_POST['password'];
        $documentName = null;

        $stmt = $pdo->prepare("SELECT id FROM users WHERE email =
    ?");
        $stmt->execute([$email]);
        if ($stmt->fetch()) {
            $error = "Цей E-mail вже зареєстрований!";
        } else {
            if            (isset($_FILES['document'])            &&
    $_FILES['document']['error'] === UPLOAD_ERR_OK) {
            $fileTmpPath = $_FILES['document']['tmp_name'];
            $fileName = $_FILES['document']['name'];
            $fileExtension   =   strtolower(pathinfo($fileName,
    PATHINFO_EXTENSION));
```

```
            $allowedExtensions = ['jpg', 'jpeg', 'png', 'pdf'];
            if (in_array($fileExtension, $allowedExtensions)) {
                $documentName = 'doc_' . time() . '_' .
md5($fileName) . '.' . $fileExtension;
                $uploadFileDir = './uploads/documents/';

                if(!is_dir($uploadFileDir)){
                    mkdir($uploadFileDir, 0755, true);
                }
                move_uploaded_file($fileTmpPath, $uploadFileDir
. $documentName);
            } else {
                $error = "Недозволений формат файлу! Дозволено
лише JPG, PNG, PDF.";
            }
        } else {
            $error = "Будь ласка, завантажте документ, що
підтверджує проживання в громаді.";
        }

        if (empty($error)) {
            $hash = password_hash($password, PASSWORD_DEFAULT);
            $stmt = $pdo->prepare("INSERT INTO users (name,
email, password, document, role, is_verified) VALUES (?, ?, ?,
?, 'user', 'pending')");
            if ($stmt->execute([$name, $email, $hash,
$documentName])) {
                header("Location: login.php?msg=reg_success");
                exit;
            } else {
                $error = "Помилка реєстрації";
            }
        }
    }
}
?>
<!DOCTYPE html>
<html lang="uk">
<head>
    <title>Реєстрація | Gromada.Connect</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <div class="container" style="max-width: 500px; margin-top:
50px;">
        <h2 style="text-align: center; margin-bottom:
25px;">Реєстрація в Gromada.Connect</h2>

        <?php if($error): ?><div class="alert alert-error"
style="display:block"><?php echo $error; ?></div><?php endif; ?>

        <form method="post" enctype="multipart/form-data"
class="card" style="padding: 30px;">
```

```
                <div class="form-group">
                    <label>ПІБ (повністю)</label>
                    <input          type="text"          name="name"
    placeholder="Іванов Іван Іванович" required>
                </div>
                <div class="form-group">
                    <label>E-mail</label>
                    <input         type="email"         name="email"
    placeholder="example@gmail.com" required>
                </div>
                <div class="form-group">
                    <label>Пароль</label>
                    <input      type="password"      name="password"
    required>
                </div>
                <div class="form-group">
                    <label>Підтвердження   проживання   в   громаді
    (Паспорт / Довідка ВПО / Витяг про реєстрацію)</label>
                    <input        type="file"        name="document"
    accept=".jpg,.jpeg,.png,.pdf"   required   style="padding:   8px
    15px;">
                    <small     style="color:     var(--text-muted);
    display: block; margin-top: 5px;">Формати: JPG, PNG, PDF. Файл
    перевірить модератор.</small>
                </div>
                <button  type="submit"  class="btn"  style="width:
    100%; margin-top: 15px;">Надіслати заявку на реєстрацію</button>
            </form>
            <p   style="text-align:   center;"><a   href="index.php"
    style="color:   var(--primary-color);   font-weight:   600;">На
    головну</a> | <a href="login.php">Вже зареєстровані?</a></p>
        </div>
    </body>
</html>

// script.js
$(document).ready(function() {
    $('#vote-btn').click(function() {
        var petitionId = $(this).data('id');
        var btn = $(this);

        $.ajax({
            url: 'vote.php',
            type: 'POST',
            data: { petition_id: petitionId },
            dataType: 'json',
            success: function(response) {
                if(response.status === 'success') {
                    var    currentVotes    =    parseInt($('#votes-
count').text());
                    $('#votes-count').text(currentVotes + 1);
                    btn.addClass('btn-disabled').text('Вже
підтримано').prop('disabled', true);
```

```
                    alert('Ваш голос зараховано!');
                } else {
                    alert(response.message);
                }
            },
            error: function() {
                alert('Помилка зв\'язку з сервером');
            }
        });
    });
});
```