

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 систем та мереж  
(повна назва кафедри)

## КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

*бакалавр*

(назва освітнього ступеня)

на тему: *Комп'ютерна система локальних мережевих сервісів для  
укриттів з обробкою даних на базі Dell OptiPlex 3050 Micro*

Виконав: студент 4 курсу, групи СІс-41  
спеціальності 123 «Комп'ютерна інженерія»

(шифр і назва спеціальності)

Омельченко К.Р.

(підпис)

(прізвище та ініціали)

Керівник

Стрембіцький М.О.

(підпис)

(прізвище та ініціали)

Нормоконтроль

Тиш Є.В.

(підпис)

(прізвище та ініціали)

Завідувач кафедри

Осухівська Г.М.

(підпис)

(прізвище та ініціали)

Рецензент

Стоянов Ю.М.

(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 систем та мереж  
(повна назва кафедри)

ЗАТВЕРДЖУЮ  
Завідувач кафедри  
Осухівська Г.М.  
(підпис) (прізвище та ініціали)  
«25» квітня 2026 р.

**ЗАВДАННЯ**  
**НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня бакалавр  
(назва освітнього ступеня)

за спеціальністю 123 «Комп'ютерна інженерія»  
(шифр і назва спеціальності)

студента Омельченка Кіріла Романовича  
(прізвище, ім'я, по батькові)

1. Тема роботи Комп'ютерна система локальних мережевих сервісів для укриттів з обробкою даних на базі Dell OptiPlex 3050 Micro

Керівник роботи Стрембіцький М. О.  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 24 » квітня 2026 року № 4/9-189

2. Термін подання студентом завершеної роботи 17.06.2026

3. Вихідні дані до роботи Технічне завдання

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

Вступ. 1. Аналіз технічного завдання

2. Проєктна частина

3. Практична частина

4. Безпека життєдіяльності, основи охорони праці.

Висновки

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

1. Блок-схема алгоритму

2. Структурна схема системи

3. Структурна схема бази даних

4. Схема роботи системи

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

6. Консультанти розділів роботи

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

7. Дата видачі завдання 25.04.2026 р.

**КАЛЕНДАРНИЙ ПЛАН**

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	<i>Розробка технічного завдання</i>	<i>26.01 – 02.02</i>	<i>Викон.</i>
2.	<i>Робота над першим розділом «Аналіз технічного завдання»</i>	<i>03.02 – 15.02</i>	<i>Викон.</i>
3.	<i>Робота над другим розділом «Проектна частина»</i>	<i>20.04 – 25.04</i>	<i>Викон.</i>
4.	<i>Робота над третім розділом «Проектна частина»</i>	<i>26.04 – 05.05</i>	<i>Викон.</i>
5.	<i>Робота над четвертим розділом «Безпека життєдіяльності, основи охорони праці»</i>	<i>07.05 – 25.05</i>	<i>Викон.</i>
6.	<i>Оформлення пояснювальної записки і графічного матеріалу</i>	<i>26.05 – 7.06</i>	<i>Викон.</i>
7.	<i>Перевірка на академічний плагіат, перевірка керівником та консультантами</i>	<i>8.06 – 14.06</i>	<i>Викон.</i>
8.	<i>Попередній захист кваліфікаційної роботи бакалавра</i>	<i>15.06 – 21.06</i>	<i>Викон.</i>
9.	<i>Захист кваліфікаційної роботи бакалавра</i>	<i>25.06</i>	<i>Викон.</i>

Студент

\_\_\_\_\_ (підпис)

Омельченко К. Р.

\_\_\_\_\_ (прізвище та ініціали)

Керівник роботи

\_\_\_\_\_ (підпис)

Стрембіцький М. О.

\_\_\_\_\_ (прізвище та ініціали)

## АНОТАЦІЯ

Омельченко К. Р. Комп'ютерна система локальних мережевих сервісів для укриттів з обробкою даних на базі Dell OptiPlex 3050 Micro: робота на здобуття кваліфікаційного ступеня бакалавра: спец. 123 — комп'ютерна інженерія. Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2026.

Ключові слова: комп'ютерна система, локальна мережа, укриття, Dell OptiPlex 3050 Micro, Flask, SQLite, автономний доступ.

Кваліфікаційна робота присвячена розробленню комп'ютерної системи локальних мережевих сервісів для укриттів з обробкою даних на базі Dell OptiPlex 3050 Micro. У роботі розглянуто побудову локальної мережі Shelter-Net, розгортання серверної частини, організацію бази даних SQLite, реалізацію користувацького інтерфейсу та адміністративних функцій. Система забезпечує локальний доступ до повідомлень, чату, інструкцій, ресурсів, мапи укриття, обліку користувачів і журналу дій. Практична реалізація підтверджує можливість роботи системи в локальному середовищі навіть за відсутності зовнішнього інтернету.

## ANNOTATION

Omelchenko K. R. Computer System of Local Network Services for Shelters with Data Processing Based on Dell OptiPlex 3050 Micro: Bachelor's Graduation Thesis: speciality 123 — computer engineering. Ternopil: Ternopil Ivan Puluj National Technical University, 2026.

Keywords: computer system, local network, shelter, Dell OptiPlex 3050 Micro, Flask, SQLite, offline access.

The bachelor thesis is devoted to the development of a computer system of local network services for shelters with data processing based on Dell OptiPlex 3050 Micro. The work considers the creation of the Shelter-Net local network, deployment of the server part, organization of the SQLite database, user interface and administrative functions. The system provides local access to messages, chat, instructions, resources, shelter map, user registration and event log. The practical implementation confirms that the system can operate in a local environment even without external Internet access.

## ЗМІСТ

ВСТУП.....	9
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ВИМОГ ДО СИСТЕМИ .....	11
1.1 Призначення локального порталу для укриття .....	11
1.2 Особливості роботи інформаційної системи в автономному середовищі .....	13
1.3 Функціональні вимоги до системи .....	14
1.4 Нефункціональні вимоги та обмеження .....	16
1.5 Аналіз технічної основи локальних мережевих сервісів .....	17
1.6 Постановка задачі кваліфікаційної роботи.....	18
РОЗДІЛ 2 ПРОЄКТУВАННЯ КОМП'ЮТЕРНОЇ СИСТЕМИ ЛОКАЛЬНИХ СЕРВІСІВ .....	20
2.1 Загальна архітектура системи .....	20
2.2 Проєктування локальної мережевої інфраструктури .....	22
2.3 Вибір програмних компонентів .....	23
2.4 Проєктування структури бази даних.....	25
2.5 Проєктування веб-інтерфейсу.....	27
2.6 Проєктування адміністративних функцій та журналу дій.....	28
2.7 Алгоритм роботи системи в онлайн- та офлайн-режимах.....	30
РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА ПЕРЕВІРКА РОБОТИ СИСТЕМИ .	32
3.1 Підготовка апаратної платформи Dell OptiPlex 3050 Micro .....	32
3.2 Налаштування локальної мережі та доступу до порталу.....	33
3.3 Розгортання серверної частини.....	35
3.4 Реалізація входу користувачів і обліку присутніх.....	36
3.5 Реалізація розділу повітряної тривоги .....	37
3.6 Реалізація оголошень, чату та інструкцій.....	39
3.7 Реалізація мапи укриття та ресурсів.....	41
3.8 Реалізація адміністративної панелі та журналу дій.....	42

					<b>КС КРБ 123.270.00.00 ПЗ</b>			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Омельченко К.Р.			<i>Комп'ютерна система локальних мережевих сервісів для укриттів з обробкою даних на базі Dell OptiPlex 3050 Micro</i>	Літ.	Арк.	Аркушів
Перевір.		Стрембиський М.О					6	1
Реценз.		Стоянов Ю.М.				ТНТУ, каф. КС, гр. СІс-41		
Н. Контр.		Тиш Є.В						
Затверд.		Осухівська Г.М.						

3.9	Перевірка працездатності системи .....	44
3.10	Особливості демонстрації системи на захисті .....	45
РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ .....		47
4.1	Загальні вимоги безпеки з охорони праці для користувачів ПК.....	47
4.2	Вимоги електробезпеки для користувачів ПК .....	49
4.3	Пожежна безпека з точки зору охорони праці .....	51
ВИСНОВКИ.....		54
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....		56
Додаток А Технічне завдання .....		60
Додаток Б Лістинг програмного коду .....		70

					КС КРБ 123.270.00.00 ПЗ	
Змн.	Арк.	№ докум.	Підпис			7

## СПИСОК СКОРОЧЕНЬ

ПК — персональний комп'ютер

БД — база даних

ЛОМ — локальна обчислювальна мережа

IP — мережевий протокол

Wi-Fi — бездротова мережа

HTTP — протокол передавання даних

API — програмний інтерфейс

HTML — мова розмітки веб-сторінок

CSS — таблиці стилів

SQLite — система керування базою даних

Flask — веб-фреймворк Python

					КС КРБ 123.270.00.00 ПЗ	
Змн.	Арк.	№ докум.	Підпис			8

## ВСТУП

Укриття під час тривалих повітряних тривог або інших небезпечних ситуацій стає не лише фізичним місцем перебування людей, а й невеликим організаційним простором. У такому просторі важливо швидко передавати повідомлення, зберігати інструкції, вести облік присутніх, показувати актуальний стан системи та надавати доступ до службової інформації без зайвої залежності від зовнішнього інтернету. Саме тому локальна комп'ютерна система для укриття має практичне значення: вона підтримує порядок у середовищі, де звичайні онлайн-сервіси можуть бути недоступними або нестабільними [11;12].

Обрана тема кваліфікаційної роботи пов'язана з розробленням комп'ютерної системи локальних мережевих сервісів для укриттів з обробкою даних на базі Dell OptiPlex 3050 Micro. У роботі розглядається не будівельна частина укриття, а інформаційна інфраструктура, яка може працювати всередині локальної мережі Shelter-Net. Користувачі підключаються до Wi-Fi, відкривають портал у браузері та отримують доступ до повідомлень, чату, інструкцій, плану укриття, ресурсів і стану повітряної тривоги. Адміністратор, у свою чергу, має окремий режим керування й може редагувати наповнення порталу, бачити список користувачів і переглядати журнал дій.

Актуальність роботи пояснюється тим, що в умовах обмеженого або відсутнього інтернет-з'єднання звичайний обмін інформацією ускладнюється. Укриття може мати електроживлення й локальну мережу, але не мати стабільного доступу до зовнішніх ресурсів. За таких обставин доцільно розмістити веб-портал і базу даних безпосередньо на локальному сервері. Це дає змогу не втрачати доступ до основних даних, навіть якщо зовнішня мережа тимчасово недоступна [13-15].

Технічною основою розробки є компактний неттоп Dell OptiPlex 3050 Micro. Його продуктивності достатньо для виконання ролі локального сервера,

					КС КРБ 123.270.00.00 ПЗ	
Змн.	Арк.	№ докум.	Підпис			9

розміщення веб-застосунку, бази SQLite та обробки запитів від клієнтських пристроїв. У поєднанні з Wi-Fi маршрутизатором пристрій формує локальну інформаційну мережу, до якої можуть підключатися телефони, ноутбуки та інші клієнти. Такий підхід є зручним для демонстрації, масштабування та подальшого вдосконалення [4;17;18].

Об'єктом дослідження є процес організації локальних мережевих сервісів у середовищі укриття. Предметом дослідження є комп'ютерна система, яка забезпечує автономний веб-доступ, локальну обробку даних, облік користувачів і адміністрування інформаційного наповнення. Мета роботи полягає у створенні та обґрунтуванні локальної інформаційної системи, яка працює на базі Dell OptiPlex 3050 Micro та забезпечує доступ до необхідних сервісів через локальну мережу.

Для досягнення мети потрібно проаналізувати вимоги до локальної системи укриття, визначити її функціональні модулі, спроектувати архітектуру мережі й програмної частини, реалізувати веб-портал, налаштувати базу даних, додати механізм обліку користувачів та перевірити працездатність системи в локальному режимі. Окрема увага приділяється тому, щоб система була зрозумілою для звичайного користувача і водночас мала службові інструменти для адміністратора.

Практична цінність роботи полягає в тому, що отриманий результат можна використовувати як прототип локального порталу для укриття, навчального стенда або автономної інформаційної точки. Система не вимагає складної серверної інфраструктури, може працювати на доступному компактному комп'ютері та підтримує основні функції, потрібні для організації людей у замкненому локальному просторі.

# РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ВИМОГ ДО СИСТЕМИ

## 1.1 Призначення локального порталу для укриття

Локальний портал для укриття варто розглядати як внутрішню інформаційну систему, яка не замінює зовнішні офіційні ресурси, але забезпечує доступ до важливих відомостей безпосередньо в межах локальної мережі. У найпростішому варіанті така система виконує роль сторінки з оголошеннями та інструкціями. У більш розвиненому варіанті вона вже містить базу даних, адміністрування, облік користувачів, локальний чат, журнал подій та інформаційні модулі, які можуть оновлюватися за наявності інтернету [1].

Головна особливість такого порталу полягає в автономності. Коли люди перебувають в укритті, вони не завжди мають стабільний мобільний зв'язок. Навіть якщо зовнішній інтернет працює, він може бути перевантаженим або обмеженим. Локальна система в такій ситуації дає змогу зберегти мінімально потрібний інформаційний простір: люди бачать стан тривоги, план приміщення, перелік ресурсів, інструкції та повідомлення від адміністратора.

У межах цієї роботи портал розглядається не як звичайний сайт у глобальній мережі, а як сервіс, доступний лише всередині локальної мережі. Це важлива відмінність, оскільки користувачеві не потрібно вводити складні зовнішні адреси або залежати від віддаленого хостингу. Достатньо підключитися до Wi-Fi мережі Shelter-Net та відкрити локальну адресу порталу.

					КС КРБ 123.270.00.00 ПЗ			
Змн.	Арк.	№ докум.	Підпис					
Розроб.		.Омельченко К.Р			Аналіз предметної області та вимог до системи	Літ.	Арк.	Аркушів
Перевір.		Стрембський М.О					11	
Реценз.						ТНТУ, каф. КС,		
Н. Контр.		Гиш Є.В.				гр. СІс-41		
Затверд.		Осухівська Г.М.						

Такий підхід спрощує використання системи для людей, які можуть не мати технічної підготовки.

У практичному сенсі портал виконує кілька ролей одночасно. Він є довідковим центром, оскільки містить інструкції та план укриття.

Він є організаційним інструментом, тому що адміністратор може створювати оголошення та бачити користувачів у системі. Він є каналом короткого зв'язку завдяки локальному чату. Також він є демонстраційною комп'ютерною системою, яка показує роботу веб-сервера, бази даних і локальної мережі в єдиному комплексі.

Для укриття важливо, щоб інформація була не тільки доступною, а й зрозумілою. Тому інтерфейс системи має бути простим: окремі вкладки, чіткі назви, помірні кольори, зрозумілі кнопки та відсутність зайвого технічного шуму. Саме такий підхід був використаний під час розробки portalу: звичайний користувач бачить тільки потрібні розділи, а службові можливості відкриваються лише після входу адміністратора [5] (рис. 1.1).

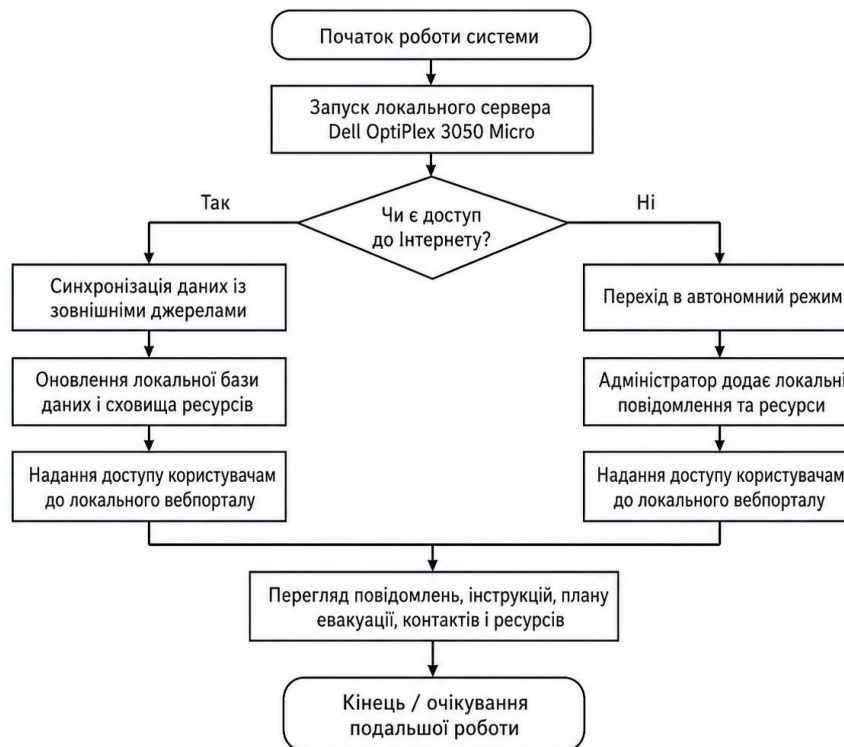


Рисунок 1.1 — Блок схема алгоритму роботи

## 1.2 Особливості роботи інформаційної системи в автономному середовищі

Середовище укриття має низку особливостей, які відрізняють його від звичайної офісної або навчальної мережі. Насамперед користувачі можуть підключатися з різних пристроїв: смартфонів, ноутбуків або планшетів. Частина людей перебуває в мережі короткий час, частина залишається довше, а частина може від'єднатися без натискання кнопки виходу. Тому система повинна не лише зберігати факт входу, а й оновлювати активність користувача, щоб адміністратор бачив, хто дійсно знаходиться у системі, а хто вже офлайн.

Інша особливість полягає в тому, що адміністратор не повинен працювати з файлами вручну для кожної дрібної зміни. Зручніше, коли оголошення, інструкції та службові дані змінюються безпосередньо через веб-інтерфейс. Це зменшує ризик помилок і робить систему придатною для практичного використання. Якщо потрібно змінити текст інструкції або прибрати старе оголошення, адміністратор виконує це з порталу, а не редагує внутрішні файли сервера.

Система також повинна враховувати, що частина даних має різний рівень доступу. Наприклад, план укриття, чат, ресурси та інструкції потрібні всім користувачам. Натомість список користувачів, журнал дій, видалення повідомлень і редагування оголошень мають бути доступними тільки адміністратору. Такий поділ дозволяє не перевантажувати звичайний інтерфейс і зберігати службову інформацію окремо.

Для автономної системи важливо не створювати в користувача хибного враження, що вона завжди отримує актуальні дані ззовні. Якщо інтернет недоступний, портал повинен чесно показувати, що працює локально, і використовувати останній збережений стан. Такий підхід є зрозумілим і безпечнішим, ніж спроба приховати втрату зовнішнього зв'язку.

					КС КРБ 123.270.00.00 ПЗ	
Змн.	Арк.	№ докум.	Підпис			13

У розробленій системі цей принцип реалізовано через відображення режиму роботи та часу останньої синхронізації.

Окремо слід врахувати психологічний аспект інтерфейсу. Укриття може бути середовищем підвищеної напруги, тому кольори й повідомлення не мають бути надмірно агресивними. Критичні стани повинні привертати увагу, але загальний вигляд порталу має залишатися спокійним і зрозумілим. Саме тому під час доопрацювання інтерфейсу були пом'якшені службові кнопки, зменшена зайва яскравість та упорядковані блоки на головній сторінці.

### 1.3 Функціональні вимоги до системи

Функціональні вимоги до локального порталу формувалися з урахуванням того, що система повинна працювати на одному компактному сервері та обслуговувати користувачів через браузер. Основною вимогою є надання доступу до порталу в локальній мережі без потреби в зовнішньому хостингу. Це означає, що веб-застосунок, база даних і статичні файли мають бути розміщені на сервері Dell.

Першим обов'язковим модулем є вхід користувача. Перед відкриттям порталу людина вводить ім'я та номер телефону в міжнародному форматі. Така перевірка потрібна не для складної ідентифікації, а для базового обліку присутніх. Якщо адміністратор видаляє запис користувача, на його пристрої форма входу з'являється повторно, що дає змогу контролювати актуальність списку.

Другим важливим модулем є відображення стану повітряної тривоги. За наявності зовнішньої мережі система може оновити стан із зовнішнього джерела, після чого результат зберігається в локальній базі. Якщо інтернет зникає, портал продовжує показувати останні збережені дані. Для користувача це виглядає просто: він бачить мапу, статус, час оновлення та коротке пояснення.

					КС КРБ 123.270.00.00 ПЗ	
Змн.	Арк.	№ докум.	Підпис			14

Третім модулем є оголошення адміністратора. Вони поділяються на звичайні, важливі та критичні. Такий поділ допомагає виділити повідомлення за пріоритетністю, але не перетворює сторінку на хаотичний набір кольорових блоків. Адміністратор може додавати, редагувати і видаляти оголошення, а всі відповідні дії мають фіксуватися в журналі.

Четвертий модуль - локальний чат. Він призначений для короткого обміну повідомленнями в межах укриття. Повідомлення надсилається від імені користувача, який увійшов до порталу, тому окреме поле для введення імені в чаті не потрібне. Адміністратор може видаляти некоректні або застарілі повідомлення, а дія видалення також записується до журналу.

Крім цього, система повинна містити ресурси укриття, інструкції, план приміщення і службову адмін-панель. Ресурси показують наявність води, аптечки, зарядних пристроїв і вільних місць. Інструкції зберігають короткі правила поведінки та підключення. Основні функціональні вимоги до порталу узагальнено в таблиці 1.1.

Таблиця 1.1 — Основні функціональні вимоги до локального порталу

Компонент	Призначення	Результат використання
Flask	Обробка HTTP-запитів і API-маршрутів	Локальний веб-сервер порталу
SQLite	Зберігання структурованих даних у файлі	База користувачів, оголошень, чату та журналу
HTML	Структура сторінок порталу	Зрозумілі вкладки й блоки інтерфейсу
CSS	Візуальне оформлення сторінок	Спокійний і впорядкований вигляд
JavaScript	Динамічна взаємодія з сервером	Оновлення даних без ручного перезавантаження

## 1.4 Нефункціональні вимоги та обмеження

Окрім переліку функцій, для системи важливими є нефункціональні вимоги. До них належать автономність, простота використання, стабільність, зрозуміла структура інтерфейсу та помірні вимоги до апаратного забезпечення. Система не повинна вимагати окремого дата-центру або складного адміністрування. Її можна розгорнути на компактному комп'ютері, який виконує роль локального сервера.

Автономність передбачає, що базові функції порталу працюють навіть без доступу до інтернету. Користувачі можуть відкривати сторінки, читати інструкції, переглядати ресурси, користуватися чатом і бачити останній збережений стан тривоги. Зовнішнє джерело потрібне тільки для оновлення певних даних, але не для існування самого порталу.

Простота інтерфейсу має не менше значення, ніж технічна працездатність. Якщо система виглядає перевантаженою або має занадто яскраві кольори, користувачі швидше втомлюються й гірше орієнтуються в розділах. Тому в проєкті використано спокійну колірну схему, розділення звичайних і службових вкладок, а також компактне розміщення інформації на головній сторінці.

Стабільність системи залежить від коректної роботи веб-сервера, бази даних і локальної мережі. Веб-застосунок має обробляти помилки без повного зупинення. Якщо зовнішнє джерело недоступне, це не повинно блокувати роботу інших розділів. Якщо користувач вийшов із мережі, його статус має змінитися на офлайн без ручного втручання.

Ще одним обмеженням є безпека даних. У системі зберігаються імена та номери телефонів користувачів, тому така інформація не повинна показуватися всім відвідувачам. Вона доступна лише адміністратору. Для демонстраційного прототипу використано простий пароль адміністратора, але

					КС КРБ 123.270.00.00 ПЗ	
Змн.	Арк.	№ докум.	Підпис			16

сама логіка розділення доступу вже закладена в структуру системи і може бути розвинена в майбутньому.

Система також має бути зрозумілою для захисту кваліфікаційної роботи. Важливо показати, що це не просто сайт і не просто роутер, а комплексна локальна комп'ютерна система. Роутер забезпечує підключення, сервер Dell обробляє дані, база SQLite зберігає інформацію, а Flask-застосунок надає користувацький і адміністративний інтерфейси.

### 1.5 Аналіз технічної основи локальних мережевих сервісів

Локальні мережеві сервіси дають змогу організувати роботу інформаційної системи без зовнішнього інтернету. У межах даного проекту головним сервісом є веб-доступ до порталу. Користувачі не встановлюють окремі програми, а відкривають сторінку через браузер. Це спрощує експлуатацію, адже сучасні смартфони й ноутбуки вже мають потрібне програмне забезпечення.

Другим важливим елементом є локальна адресація. Для зручності користувач може відкривати портал не лише за IP-адресою, а й за зрозумілим іменем `portal.shelter.local`. Такий підхід робить систему більш завершеною, тому що адреса виглядає як внутрішній ресурс мережі. У практичному стенді також передбачалася запасна адреса за IP, щоб доступ можна було перевірити навіть у разі проблем з локальним DNS.

Третім елементом є файлове сховище. Укриття може потребувати локальних інструкцій, таблиць, документів або файлів, які мають бути доступними без інтернету. Для цього на сервері можна створити спільну папку, доступну в локальній мережі через SMB. У веб-порталі такі файли можуть доповнювати основні інструкції або використовуватися адміністратором для зберігання службових матеріалів.

					КС КРБ 123.270.00.00 ПЗ	
Змн.	Арк.	№ докум.	Підпис			17

База даних SQLite обрана як простий і зручний спосіб локального зберігання інформації. Вона не потребує окремого серверного процесу, зберігає дані в одному файлі й достатня для прототипу, де кількість користувачів і запитів є помірною. Для розробки системи укриття цього достатньо, оскільки основне завдання полягає не в обробці великих масивів даних, а в надійному збереженні структурованої інформації.

Веб-застосунок реалізовано на основі Flask. Такий варіант підходить для навчального і практичного прототипу, тому що дає змогу швидко створити маршрути API, підключити базу даних, організувати статичні файли й реалізувати логіку адміністратора. Важливо, що сервер запускається локально і може приймати запити з інших пристроїв у мережі, якщо працює на адресі 0.0.0.0 та потрібному порту.

Таким чином, технічна основа системи складається з кількох взаємопов'язаних компонентів: Dell-сервер, Wi-Fi маршрутизатор, локальна адреса, Flask-застосунок, база SQLite, статичні HTML/CSS/JavaScript-файли та клієнтські браузер. Разом вони утворюють завершену комп'ютерну систему, а не набір окремих налаштувань.

## 1.6 Постановка задачі кваліфікаційної роботи

На основі аналізу предметної області задача кваліфікаційної роботи формулюється як створення локальної комп'ютерної системи, яка забезпечує інформаційне обслуговування користувачів укриття в умовах нестабільного зовнішнього інтернету. Система має бути розгорнута на базі Dell OptiPlex 3050 Micro, працювати в локальній мережі Shelter-Net та надавати доступ через веб-браузер.

У межах роботи потрібно спроектувати загальну архітектуру системи, визначити мережеву схему, розробити структуру бази даних і створити веб-інтерфейс. Окремо потрібно реалізувати адміністративний режим, у якому

					КС КРБ 123.270.00.00 ПЗ	
Змн.	Арк.	№ докум.	Підпис			18

можна керувати користувачами, оголошеннями, чатом, інструкціями та переглядати журнал дій. Службові можливості не мають заважати звичайному користувачеві, тому вони відокремлюються від основних вкладок.

Практична реалізація має показати не лише зовнішній вигляд порталу, а й обробку даних. До бази мають записуватися користувачі, їх активність, оголошення, повідомлення чату, інструкції, ресурси, журнальні записи та стан тривоги. Такий набір даних дає змогу продемонструвати реальну взаємодію між клієнтським інтерфейсом, серверною логікою та локальним сховищем.

Результатом виконання роботи має стати працездатний прототип, який можна показати на захисті. Демонстрація передбачає підключення пристрою до Wi-Fi мережі, відкриття порталу, вхід користувача, надсилання повідомлення в чат, перегляд стану тривоги, перехід до плану укриття та вхід адміністратора. Також має бути показано, що адміністратор може редагувати наповнення порталу, а зміни зберігаються в базі даних.

Отже, подальші розділи роботи присвячені проєктуванню системи та опису практичної реалізації. У другому розділі розглядаються архітектурні рішення, структура модулів і база даних. У третьому розділі описується розгортання системи, налаштування локальної мережі, реалізація веб-порталу та перевірка роботи основних функцій [18-20].

## РОЗДІЛ 2 ПРОЄКТУВАННЯ КОМП'ЮТЕРНОЇ СИСТЕМИ ЛОКАЛЬНИХ СЕРВІСІВ

### 2.1 Загальна архітектура системи

Проектowana система побудована за клієнт-серверним принципом, але в межах локальної мережі. Серверна частина розміщується на Dell OptiPlex 3050 Micro. Клієнтськими пристроями є телефони, ноутбуки або планшети користувачів, які підключаються до Wi-Fi мережі Shelter-Net. Зовнішній інтернет не є обов'язковою умовою для доступу до порталу, оскільки сторінки, сценарії, стилі та база даних знаходяться на локальному сервері [2-5].

Архітектура системи передбачає чітке розділення ролей. Wi-Fi маршрутизатор відповідає за підключення пристроїв до мережі. Сервер Dell відповідає за обробку HTTP-запитів, взаємодію з базою даних і видачу сторінок порталу. Клієнтський браузер відповідає за відображення інтерфейсу та надсилання дій користувача через API. Такий поділ дозволяє пояснити систему як цілісний комплекс, де кожен компонент має своє призначення.

У системі передбачено два режими взаємодії: звичайний користувач і адміністратор. Звичайний користувач після входу бачить основні розділи: головну сторінку, тривогу, оголошення, чат, ресурси, мапу та інструкції. Адміністратор після введення пароля отримує додаткову панель керування. У цій панелі службові функції згруповані так, щоб не дублювати звичайні вкладки й не створювати перевантаження.

Серверна логіка реалізує маршрути для читання і зміни даних. Частина маршрутів є відкритою для звичайного користувача, наприклад отримання оголошень або надсилання повідомлення в чат.

					КС КРБ 123.270.00.00 ПЗ			
Змн.	Арк.	№ докум.	Підпис		Проектвання комп'ютерної системи локальних сервісів	Літ.	Арк.	Аркушів
Розроб.		Омельченко К.Р						
Перевір.		Стрембський О.М					20	
Реценз.								
Н. Контр.		Тиш Є.В.						
Затверд.		Осухівська Г.М.				ТНТУ, каф. КС, гр. СІс-41		

Інша частина потребує адміністративного пароля, зокрема видалення користувача, редагування оголошення, видалення повідомлення або перегляд журналу.

Така схема є простою, але достатньою для демонстраційного прототипу.

Важливо, що база даних не є пасивним додатком до сайту. Вона зберігає стан системи й забезпечує зв'язок між різними модулями. Коли користувач входить, його запис потрапляє до таблиці відвідувачів. Коли він надсилає повідомлення, воно зберігається в чаті від його імені. Коли адміністратор змінює оголошення, дія фіксується в журналі. Саме через базу даних портал набуває властивостей інформаційної системи [3] (рис. 2.1).

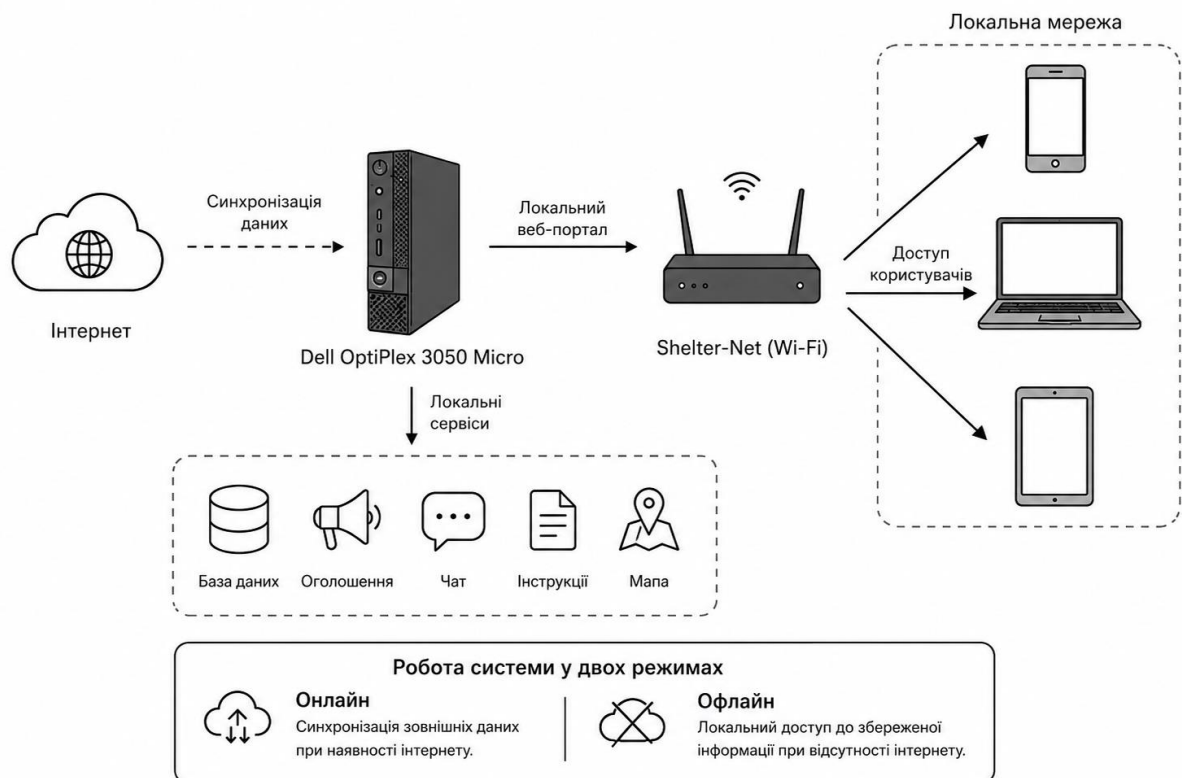


Рисунок 2.1 — Структурна схема

## 2.2 Проектування локальної мережевої інфраструктури

Локальна мережа проектується таким чином, щоб користувачеві було достатньо підключитися до Wi-Fi та відкрити адресу порталу. У практичній реалізації використовується мережа Shelter-Net, а сервер має локальну адресу 192.168.137.1. Для зручності також використовується доменне ім'я portal.shelter.local:5000, яке краще сприймається користувачем і виглядає як адреса внутрішнього ресурсу [18;20].

Порт 5000 використовується для роботи Flask-застосунку. У демонстраційному прототипі це зручно, оскільки сервер запускається без складного додаткового веб-сервера. У подальшому систему можна перенести на стандартний порт або розмістити за проксі-сервером, однак для кваліфікаційної роботи важливіше показати працездатну логіку локального доступу.

У мережевій схемі Dell виконує роль центру обробки даних. Він не обов'язково має роздавати Wi-Fi самостійно, оскільки цю функцію виконує маршрутизатор. Проте саме Dell є сервером системи: він приймає запити, працює з базою даних і зберігає файли. Роутер у такій схемі не є основною системою, а лише забезпечує канал підключення користувачів.

Для захисту роботи важливо підкреслити, що мережа не зводиться до звичайного підключення до маршрутизатора. Якщо прибрати сервер, користувачі матимуть Wi-Fi, але не матимуть локального порталу, чату, обліку людей і адміністративної панелі. Якщо прибрати маршрутизатор, сервер залишиться працездатним, але користувачам буде складніше підключатися. Отже, саме поєднання апаратної платформи, мережі та програмного забезпечення утворює систему.

Під час проектування також враховується можливість роботи без зовнішнього інтернету. Якщо Dell має тимчасовий доступ до зовнішньої мережі, адміністратор може оновити стан тривоги. Якщо такого доступу немає, користувачі все одно відкривають портал і бачать останній збережений

					КС КРБ 123.270.00.00 ПЗ	
Змн.	Арк.	№ докум.	Підпис			22

стан. Така поведінка є важливою для укриття, де стабільність локального доступу має більший пріоритет, ніж постійна онлайн-синхронізація (рис. 2.2).

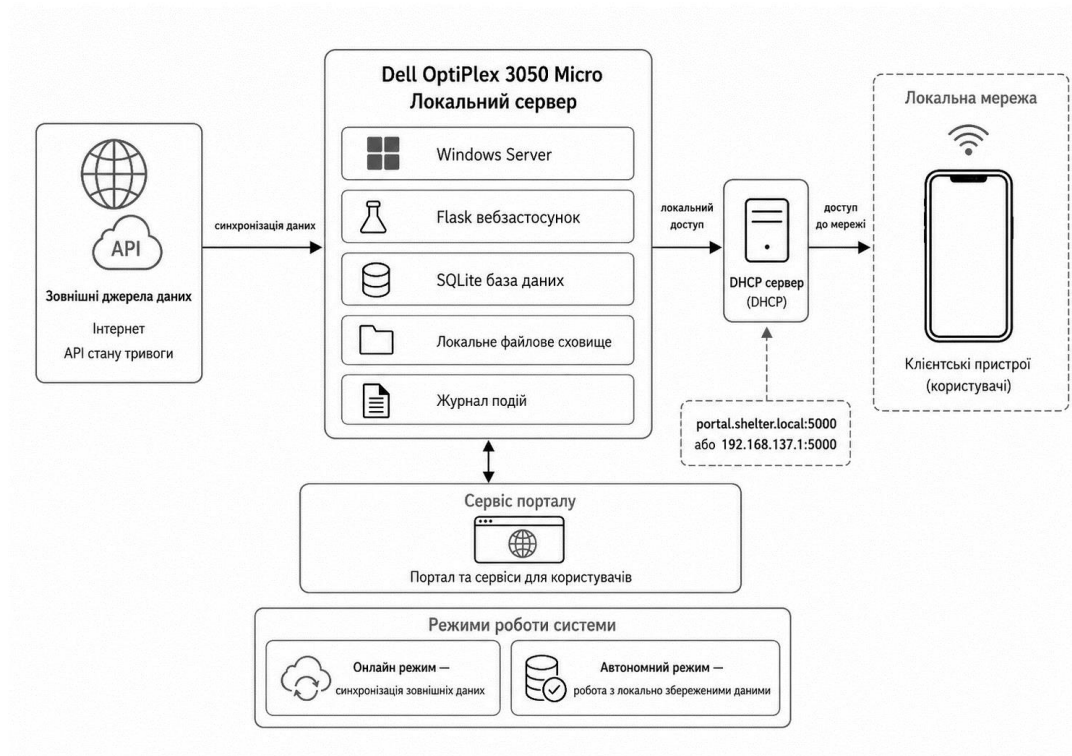


Рисунок 2.2 — Локальна мережева схема порталу укриття

### 2.3 Вибір програмних компонентів

Програмна частина системи складається з серверного застосунку, бази даних і клієнтського інтерфейсу. Для серверної логіки обрано Flask, оскільки він дозволяє швидко створити веб-застосунок з API-маршрутами та роботою зі статичними файлами. Для бази даних використано SQLite, оскільки вона проста в розгортанні й не потребує окремого сервера баз даних. Клієнтська частина складається з HTML, CSS і JavaScript.

Такий набір технологій добре відповідає умовам кваліфікаційної роботи. Система є достатньо зрозумілою для пояснення, не потребує складної інфраструктури та дає змогу показати взаємодію між усіма рівнями. Flask обробляє запити, SQLite зберігає інформацію, JavaScript оновлює сторінки без

повного перезавантаження, а CSS відповідає за зручний і спокійний вигляд порталу.

Вибір SQLite пояснюється ще й тим, що локальна система укриття не потребує одночасної роботи тисяч користувачів. Для демонстраційного стенда й невеликого приміщення достатньо компактної бази. Водночас структура таблиць залишається типовою для інформаційної системи: є користувачі, повідомлення, оголошення, інструкції, ресурси, журнал подій і стан тривоги [3].

Клієнтський JavaScript відповідає за динамічну поведінку порталу. Він надсилає запити на сервер, отримує JSON-дані, будує картки повідомлень, показує модальні підтвердження, оновлює чат і перевіряє активність користувача. Завдяки цьому сторінка не виглядає статичною, а реагує на дії користувача майже одразу.

CSS має не лише декоративну роль. Від нього залежить, чи зручно користуватися порталом у стресовій ситуації. Тому під час проектування колірну схему було зроблено помірною, без надмірної яскравості. Критичні елементи залишаються помітними, але не переважають над основним змістом. Призначення основних програмних компонентів наведено в таблиці 2.1.

Таблиця 2.1 — Призначення основних програмних компонентів

Компонент	Призначення	Результат використання
Flask	Обробка HTTP-запитів і API-маршрутів	Локальний веб-сервер порталу
SQLite	Зберігання структурованих даних у файлі	База користувачів, оголошень, чату та журналу
HTML	Структура сторінок порталу	Зрозумілі вкладки й блоки інтерфейсу
CSS	Візуальне оформлення сторінок	Спокійний і впорядкований вигляд
JavaScript	Динамічна взаємодія з сервером	Оновлення даних без ручного перезавантаження

## 2.4 Проектування структури бази даних

База даних є центральним елементом системи, оскільки саме в ній зберігаються всі змінні дані порталу. На відміну від статичних HTML-сторінок, база дозволяє адміністратору змінювати наповнення без редагування файлів. Якщо створюється нове оголошення, воно записується до таблиці. Якщо користувач надсилає повідомлення, воно також потрапляє в базу. Якщо адміністратор видаляє запис, система фіксує цю дію в журналі.

Таблиця `visitors` призначена для обліку користувачів. Вона містить ім'я, номер телефону, IP-адресу, час входу, час виходу, час останньої активності та статус. Завдяки полю `last_seen` система може визначити, чи користувач ще активний у мережі. Якщо активність не оновлюється протягом певного часу, користувач відображається як офлайн.

Таблиця `chat_messages` зберігає локальні повідомлення. Вона містить ім'я користувача, текст повідомлення, час створення і зв'язок з відвідувачем. Така структура дозволяє показувати повідомлення від імені людини, яка увійшла до порталу, а не від довільно введеного імені. Це робить чат більш контрольованим і придатним для середовища укриття.

Таблиця `announcements` відповідає за оголошення адміністратора. Кожне оголошення має заголовок, текст, тип важливості, ознаку активності та час створення. Видалення оголошення може бути реалізоване як приховування, що дає змогу не втрачати дані фізично, а лише прибрати їх із відображення для користувача.

Таблиця `instructions` зберігає інструкції. Їх можна додавати, редагувати і видаляти через адмін-панель. Таблиця `event_log` фіксує службові дії: запуск сервера, вхід користувача, вхід адміністратора, редагування, видалення та інші операції. Завдяки журналу система стає прозорішою для адміністратора і для демонстрації на захисті (рис. 2.3). Логічну структуру бази даних узагальнено в таблиці 2.2.

Таблиця 2.2 — Логічна структура бази даних

Таблиця	Основні дані	Призначення
system_status	режим, стан інтернету, час синхронізації, джерело	Зберігання загального стану системи
external_alerts	регіон, статус тривоги, повідомлення, час оновлення	Збереження стану повітряної тривоги
visitors	ім'я, телефон, IP, час входу, last_seen, статус	Облік користувачів і визначення онлайн/офлайн
chat_messages	користувач, текст, час, visitor_id	Локальний чат укриття
announcements	заголовок, текст, важливість, активність	Оголошення адміністратора
instructions	назва, текст	Інструкції для користувачів
event_log	тип події, опис, час	Журнал дій системи й адміністратора

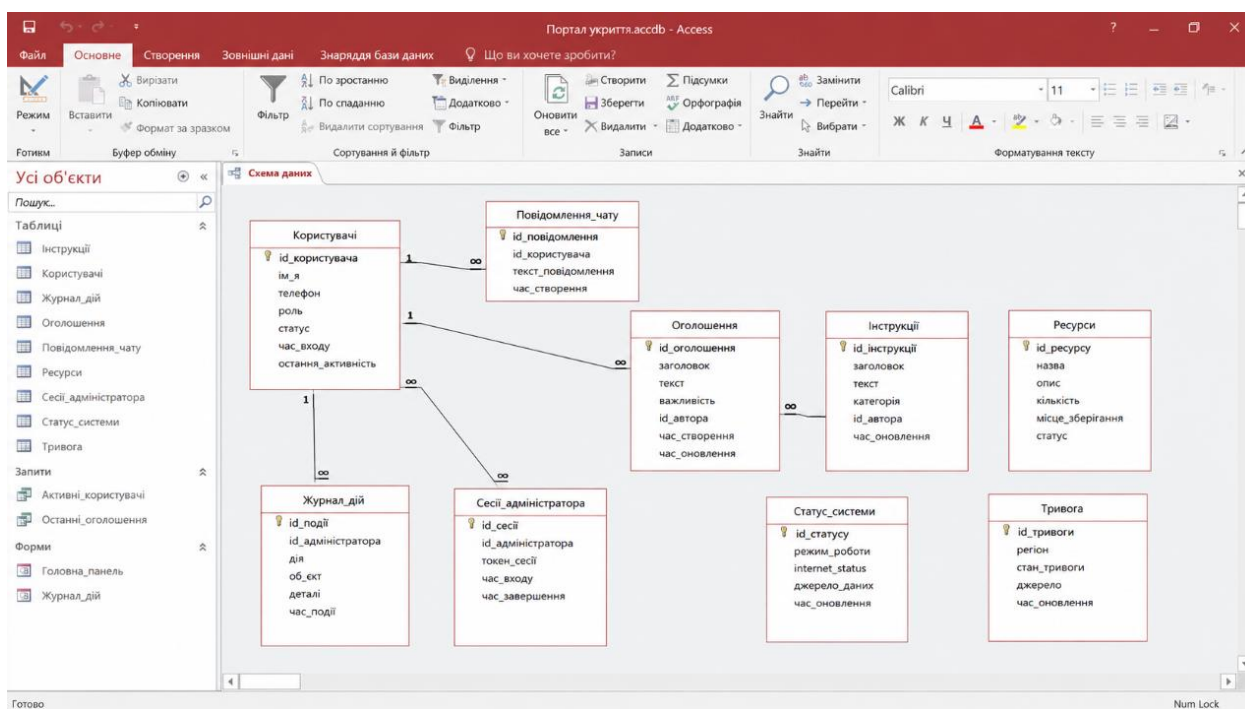


Рисунок 2.3 — Структура бази даних порталу

## 2.5 Проектування веб-інтерфейсу

Веб-інтерфейс порталу проектувався з урахуванням двох різних сценаріїв: використання звичайною людиною в укритті та використання адміністратором. Для звичайного користувача важливо швидко побачити потрібну інформацію. Тому навігація складається з простих вкладок: головна, тривога, оголошення, чат, ресурси, мапа та інструкції. Вкладка статусу була вилучена з основної навігації, оскільки її інформація дублювалася на головній сторінці [5].

Головна сторінка має виконувати роль спокійного стартового екрана. На ній розміщено коротке пояснення призначення системи, адресу порталу, назву Wi-Fi мережі, QR-код для швидкого доступу та компактний блок стану системи. При цьому з головної сторінки прибрано зайві кнопки, які дублювали навігацію. Це робить сторінку менш перевантаженою і більш придатною для першого ознайомлення.

Розділ повітряної тривоги проектувався як компактний інформаційний блок. Основним елементом є мапа з виділеним регіоном, поруч із якою показано статус, час оновлення, регіон і джерело. Замість технічного формулювання про зовнішній API використовується більш зрозумілий напис «зовнішнє джерело». Це краще підходить для користувачів, які не повинні розбиратися в деталях роботи API.

Розділ чату має показувати історію повідомлень вище поля введення. Така структура є логічнішою: користувач спочатку бачить, що вже написали інші, а потім вводить своє повідомлення. Поле введення зроблене достатньо широким, щоб не виглядати випадковим маленьким блоком у центрі сторінки. Повідомлення надсилаються від імені користувача, який увійшов на портал.

Розділ мапи укриття перероблено у вигляді плану будівлі, а не набору окремих блоків. На плані показано вхід, коридор, місця для людей, воду, аптечку, зарядну станцію, адміністратора та запасний вихід. Такий вигляд

					КС КРБ 123.270.00.00 ПЗ	
Змн.	Арк.	№ докум.	Підпис			27

краще відповідає призначенню системи, оскільки користувач сприймає його як схему евакуації, а не як декоративну діаграму.

Адмін-панель має іншу логіку. Після входу адміністратора частина звичайних вкладок ховається, щоб не дублювати інформацію. Усі службові дії зосереджені в одному розділі, де є внутрішні кнопки для користувачів, оголошень, чату, інструкцій і журналу дій. Відкриття одного блоку замінює інший, тому панель не виглядає перевантаженою (рис. 2.4).

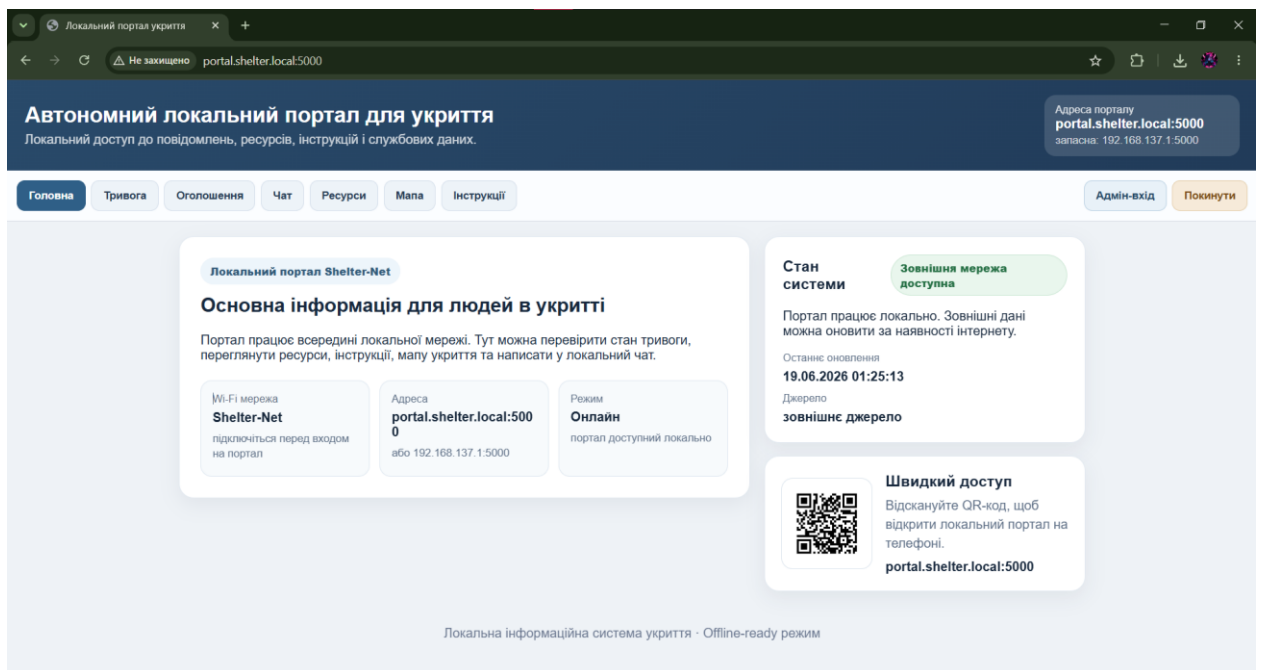


Рисунок 2.4 — Схема структури веб-інтерфейсу користувача

## 2.6 Проектування адміністративних функцій та журналу дій

Адміністративна частина системи потрібна для того, щоб відповідальна особа могла підтримувати актуальність порталу без доступу до коду. Адміністратор входить через окрему кнопку і вводить пароль. Після цього кнопка входу зникає, натомість з'являється можливість вийти з адмін-режиму. Якщо адміністратор не користується панеллю протягом години, сесія вважається завершеною і пароль потрібно ввести повторно [13;15].

У панелі адміністратора передбачено керування користувачами. Адміністратор бачить ім'я, номер телефону, час входу та статус людини. Якщо користувач активний у мережі, він відображається як «У системі». Якщо він вийшов або не оновлює активність, показується статус «Офлайн». Адміністратор може видалити користувача, після чого на пристрої цієї людини при наступній перевірці з'явиться форма входу.

Керування оголошеннями передбачає повне редагування запису, а не лише зміну заголовка. Адміністратор може змінити заголовок, текст і тип важливості. Це важливо, тому що зміст оголошення може змінюватися повністю, а не тільки в назві. Модальні вікна підтвердження використовуються для видалення та інших важливих дій, щоб уникнути випадкових операцій.

Керування чатом дозволяє видаляти окремі повідомлення. Це потрібно для підтримання порядку в локальному середовищі, особливо якщо хтось помилково надіслав зайвий або неактуальний текст. Керування інструкціями дозволяє додавати нові правила, редагувати наявні й видаляти застарілі. Усі ці функції реалізовані через API-маршрути, захищені адміністративним паролем.

Журнал дій є окремим службовим механізмом. До нього записуються запуск сервера, вхід користувачів, вхід адміністратора, видалення користувачів, редагування оголошень, видалення повідомлень і зміни інструкцій. Такий журнал корисний не лише для контролю, а й для пояснення роботи системи під час захисту. Він демонструє, що система не просто показує сторінки, а веде послідовний облік подій.

Для кращого вигляду підтвердження дій оформлені як власні модальні вікна, а не стандартні браузерні повідомлення. Це робить інтерфейс цілісним і професійнішим. Користувач бачить назву дії, коротке пояснення, додаткові деталі й кнопки підтвердження або скасування. Такий підхід зменшує ризик випадкових змін у системі (рис. 2.5).

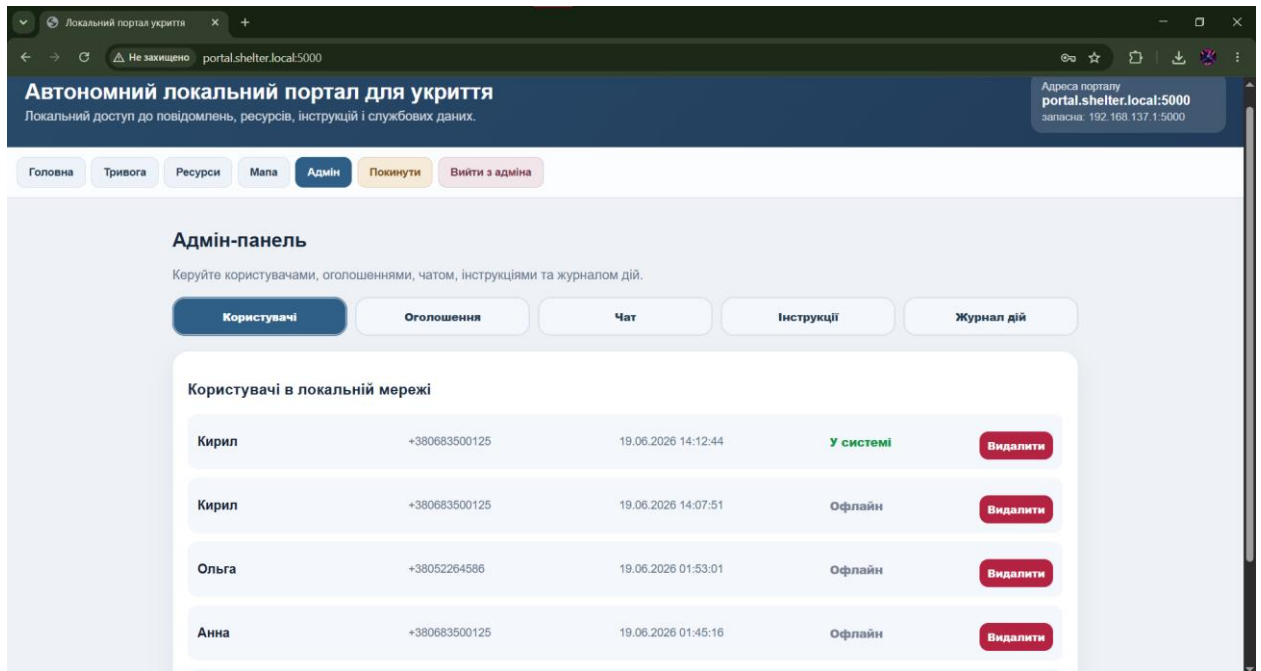


Рисунок 2.5 — Структура адміністративної панелі

## 2.7 Алгоритм роботи системи в онлайн- та офлайн-режимах

Система повинна однаково впевнено працювати як за наявності зовнішнього інтернету, так і без нього. У звичайному локальному режимі користувач відкриває портал, вводить ім'я та номер телефону, після чого отримує доступ до сторінок. Сервер записує користувача в базу даних, а клієнтський сценарій періодично надсилає сигнал активності. Завдяки цьому адміністратор бачить, хто ще перебуває у системі [2;3].

Якщо сервер має доступ до зовнішньої мережі, адміністратор може натиснути кнопку оновлення стану тривоги. Сервер виконує запит до зовнішнього джерела, визначає стан для Тернопільської області, зберігає результат у таблицю `external_alerts` і показує оновлену інформацію користувачам. У журналі може бути зафіксовано факт синхронізації.

Якщо зовнішня мережа недоступна, сервер не зупиняє роботу порталу. Він використовує останній збережений запис і показує його як актуальний для локального перегляду з відповідним часом оновлення. Для користувача

важливо, що портал не зникає і не показує порожню сторінку. Він продовжує працювати як локальний інформаційний центр.

У разі виходу користувача з portalу натискається кнопка «Покинути». Перед цим система показує підтвердження. Якщо користувач погоджується, сервер змінює його статус, а на пристрої знову з'являється форма входу. Якщо користувач просто від'єднався від Wi-Fi або закрив браузер, активність перестає оновлюватися, і через певний час він відображається як офлайн.

Адміністративна сесія працює окремо від звичайного користувацького входу. Навіть якщо людина увійшла на портал як відвідувач, для доступу до адмін-панелі потрібно ввести пароль. Після входу в журнал записується, хто саме відкрив адміністративний режим. Це дозволяє простежити службові дії та пояснює зв'язок між користувачем portalу й адміністраторською активністю.

					КС КРБ 123.270.00.00 ПЗ	
Змн.	Арк.	№ докум.	Підпис			31

## РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА ПЕРЕВІРКА РОБОТИ СИСТЕМИ

### 3.1 Підготовка апаратної платформи Dell OptiPlex 3050 Micro

Практична реалізація системи починається з підготовки апаратної платформи. У роботі використано Dell OptiPlex 3050 Micro з процесором Intel Core i5-6500T, 16 ГБ оперативної пам'яті та накопичувачем 500 ГБ. Для локального порталу таких характеристик достатньо, оскільки сервер обслуговує невелику кількість клієнтів і виконує типові задачі: запуск веб-застосунку, робота з SQLite, зберігання статичних файлів та обробка API-запитів [17].

Компактний форм-фактор пристрою є важливою перевагою. Неттоп легко розмістити біля маршрутизатора або в службовій зоні укриття. Він не займає багато місця, споживає менше електроенергії, ніж повнорозмірний системний блок, і має достатню кількість ресурсів для навчального та практичного стенда. Це робить його зручним вибором для демонстрації локальної системи.

На сервері створено структуру каталогів, яка відокремлює серверну частину, веб-файли, дані та файлове сховище. Такий поділ полегшує обслуговування системи. Серверний файл застосунку розміщується в каталозі backend, HTML/CSS/JavaScript - у каталозі web, база даних - у каталозі data, а локальні файли для обміну можуть зберігатися в storage.

Під час підготовки важливо переконатися, що Python встановлений коректно, команда запуску працює з потрібної папки, а файл app.py відкривається без помилок.

					КС КРБ 123.270.00.00 ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Омельченко К.Р			Практична реалізація та перевірка роботи системи	Літ.	Арк.	Аркушів
Перевір.		Стрембський М.О					32	
Реценз.		Стоянов Ю.М.				ТНТУ, каф. КС, гр. СІс-41		
Н. Контр.		Тиш Є.В.						
Затверд.		Осухівська Г.М.						

У процесі налаштування можливі типові проблеми: запуск із неправильного каталогу, помилки відступів у Python, відсутність таблиць у базі або застарілий кеш браузера. Усі ці помилки вирішуються послідовною перевіркою шляху, коду, структури бази та оновленням сторінки через Ctrl+F5.

Після підготовки апаратної платформи Dell виконує роль локального сервера. Це означає, що він має бути ввімкнений і на ньому повинен працювати Flask-застосунок. Клієнтські пристрої не запускають програму самостійно, а тільки відкривають веб-інтерфейс. Такий принцип спрощує використання системи для кінцевих користувачів (рис. 3.1).

					КС КРБ 123.270.00.00 ПЗ	
Змн.	Арк.	№ докум.	Підпис			33

 **Dell OptiPlex 3050 Micro**  
локальний сервер

	<b>ПРОЦЕСОР</b>	Intel Core i5-6500T
	<b>ОПЕРАТИВНА ПАМ'ЯТЬ</b>	16 GB DDR4
	<b>НАКОПИЧУВАЧ</b>	500 GB HDD
	<b>МЕРЕЖА</b>	Ethernet LAN (10/100/1000 Мбіт/с)
	<b>ОПЕРАЦІЙНА СИСТЕМА</b>	Windows Server 2019 Standard Evaluation
	<b>РОЛЬ У СИСТЕМІ</b>	локальний сервер порталу укриття



Рисунок 3.1 — Апаратна платформа Dell OptiPlex 3050 Micro

### 3.2 Налаштування локальної мережі та доступу до порталу

Для доступу користувачів до системи створюється локальна Wi-Fi мережа Shelter-Net. Сервер Dell підключається до цієї мережевої схеми й отримує локальну адресу 192.168.137.1. Саме за цією адресою клієнти можуть відкрити портал у браузері. Додатково використовується зрозуміла адреса `portal.shelter.local:5000`, яка вказує на той самий сервер.

На практиці доступ перевіряється у два способи. Спочатку портал відкривається безпосередньо на сервері за адресою `127.0.0.1:5000`. Це підтверджує, що Flask-застосунок запущений. Після цього портал відкривається з іншого пристрою в локальній мережі за адресою `192.168.137.1:5000` або `portal.shelter.local:5000`. Якщо сторінка відкривається з телефону чи ноутбука, можна вважати, що локальний доступ працює.

Важливо, щоб серверний застосунок запускався не тільки на локальному інтерфейсі, а й приймав запити з мережі. Для цього в налаштуваннях запуску використовується адреса `0.0.0.0`. У такому разі Flask слухає всі доступні мережеві інтерфейси, а не лише `localhost`. Це є типовою умовою для доступу з інших пристроїв.

Для зручності користувача на головній сторінці додано QR-код. Його можна відсканувати телефоном, щоб швидко відкрити адресу порталу. Це особливо корисно, коли користувач не хоче вручну вводити локальну адресу з портом. QR-код не є обов'язковим для роботи системи, але покращує зручність доступу (рис. 3.2).

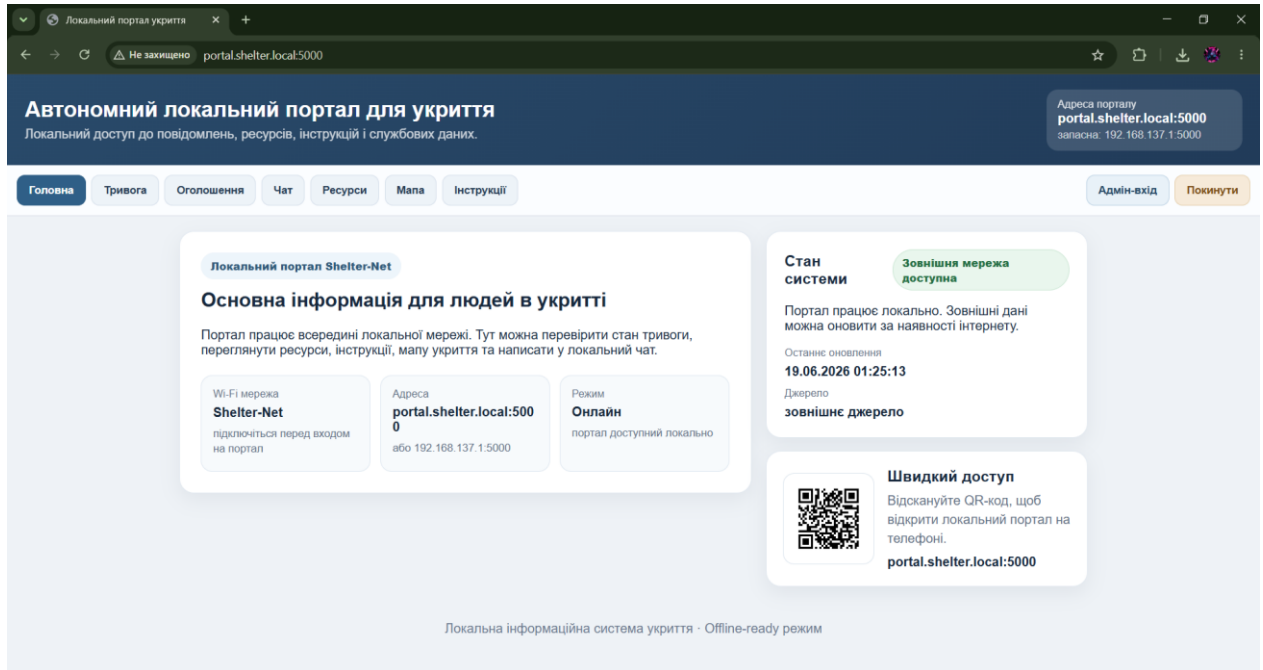


Рисунок 3.2 — Перевірка доступу до порталу з клієнтського пристрою

### 3.3 Розгортання серверної частини

Серверна частина системи реалізована у файлі `app.py`. У ньому створюється Flask-застосунок, визначаються шляхи до каталогів `web` і `data`, підключається база `SQLite` та описуються API-маршрути. Під час запуску застосунок ініціалізує базу даних, створює потрібні таблиці, додає початкові записи й відкриває порт для локальних клієнтів [2;3;4].

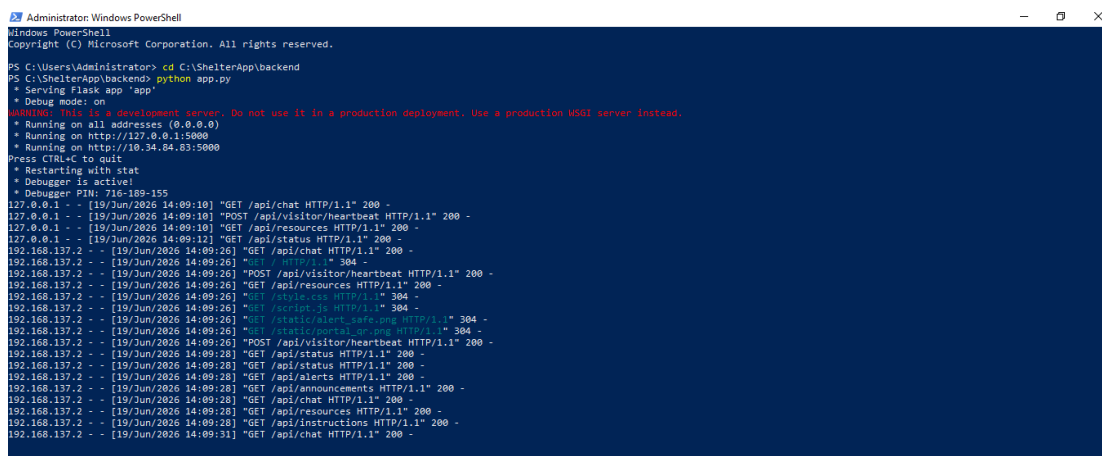
Важливою частиною серверної логіки є функція ініціалізації бази даних. Вона перевіряє наявність таблиць і створює їх у разі потреби. Якщо в майбутньому до таблиці додається нове поле, застосунок може додати його без повного видалення бази. Це зручно під час розробки, тому що структура

системи поступово уточнювалася: додавалися last\_seen для користувачів, visitor\_id для чату, поля для журналу та інші елементи.

Серверна частина також відповідає за перевірку номера телефону. Користувач має ввести номер у міжнародному форматі, наприклад з кодом країни. Така перевірка не є складною авторизацією, але не дозволяє вводити випадковий набір цифр без знаку плюс. Для прототипу це достатньо, щоб дані виглядали впорядковано.

Для адміністративних дій використовується окрема перевірка пароля через заголовок запиту. Якщо пароль відсутній або неправильний, сервер не дозволяє виконати службову операцію. До таких операцій належать видалення користувача, видалення повідомлення, редагування оголошення, редагування інструкції та перегляд журналу. Отже, серверна частина не покладається лише на приховування кнопок у браузері, а додатково перевіряє доступ на рівні API.

Окрему роль відіграє механізм журналу дій. Сервер має функцію запису подій, яка додає до таблиці event\_log тип події, опис і час. Завдяки цьому в журналі можна побачити запуск сервера, вхід користувача, вхід адміністратора та зміни, виконані через панель керування. Така інформація корисна під час тестування і демонстрації (рис. 3.3).



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> cd C:\ShelterApp\backend
PS C:\ShelterApp\backend> python app.py
 * Serving Flask app "app"
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on https://10.34.84.83:5000
Press CTRL-C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 716-189-155
127.0.0.1 - - [19/Jun/2026 14:09:10] "GET /api/chat HTTP/1.1" 200 -
127.0.0.1 - - [19/Jun/2026 14:09:10] "POST /api/visitor/heartbeat HTTP/1.1" 200 -
127.0.0.1 - - [19/Jun/2026 14:09:10] "GET /api/resources HTTP/1.1" 200 -
127.0.0.1 - - [19/Jun/2026 14:09:12] "GET /api/status HTTP/1.1" 200 -
192.168.137.2 - - [19/Jun/2026 14:09:26] "GET /api/chat HTTP/1.1" 200 -
192.168.137.2 - - [19/Jun/2026 14:09:26] "GET /api/status HTTP/1.1" 200 -
192.168.137.2 - - [19/Jun/2026 14:09:26] "POST /api/visitor/heartbeat HTTP/1.1" 200 -
192.168.137.2 - - [19/Jun/2026 14:09:26] "GET /api/resources HTTP/1.1" 200 -
192.168.137.2 - - [19/Jun/2026 14:09:26] "GET /static/css HTTP/1.1" 304 -
192.168.137.2 - - [19/Jun/2026 14:09:26] "GET /static/js HTTP/1.1" 304 -
192.168.137.2 - - [19/Jun/2026 14:09:26] "GET /static/alert_safe.png HTTP/1.1" 304 -
192.168.137.2 - - [19/Jun/2026 14:09:26] "GET /static/portal_loading HTTP/1.1" 304 -
192.168.137.2 - - [19/Jun/2026 14:09:26] "POST /api/visitor/heartbeat HTTP/1.1" 200 -
192.168.137.2 - - [19/Jun/2026 14:09:28] "GET /api/status HTTP/1.1" 200 -
192.168.137.2 - - [19/Jun/2026 14:09:28] "GET /api/status HTTP/1.1" 200 -
192.168.137.2 - - [19/Jun/2026 14:09:28] "GET /api/alerts HTTP/1.1" 200 -
192.168.137.2 - - [19/Jun/2026 14:09:28] "GET /api/announcements HTTP/1.1" 200 -
192.168.137.2 - - [19/Jun/2026 14:09:28] "GET /api/chat HTTP/1.1" 200 -
192.168.137.2 - - [19/Jun/2026 14:09:28] "GET /api/resources HTTP/1.1" 200 -
192.168.137.2 - - [19/Jun/2026 14:09:28] "GET /api/instructions HTTP/1.1" 200 -
192.168.137.2 - - [19/Jun/2026 14:09:31] "GET /api/chat HTTP/1.1" 200 -
```

Рисунок 3.3 — Запуск серверної частини Flask

### 3.4 Реалізація входу користувачів і обліку присутніх

Перед доступом до порталу користувач бачить форму входу. У ній потрібно ввести ім'я та номер телефону. Після натискання кнопки система надсилає дані на сервер, сервер перевіряє номер, створює запис у таблиці visitors і повертає ідентифікатор користувача. Цей ідентифікатор зберігається в браузері, щоб наступні дії були пов'язані з конкретною людиною (рис. 3.4).

Облік користувачів має практичне значення для укриття. Адміністратор бачить, хто зайшов на портал, коли це сталося і чи користувач ще активний. Стан «У системі» означає, що браузер продовжує надсилати сигнали активності. Якщо таких сигналів немає, користувач переходить у стан «Офлайн». Це вирішує проблему ситуації, коли людина просто закрила сторінку або вийшла з Wi-Fi, але не натиснула кнопку виходу.

Кнопка «Покинути» реалізує контрольований вихід із порталу. Перед виходом користувач бачить підтвердження. Якщо він погоджується, сервер змінює його статус, а локальні дані входу очищуються. Після цього для повторного доступу потрібно знову ввести ім'я та телефон. Така поведінка зрозуміла і для користувача, і для адміністратора.

Адміністратор може видалити користувача зі списку. У такому разі запис зникає з бази, а на пристрої користувача при наступному оновленні активності система отримує відповідь, що потрібно увійти повторно. Це дає адміністратору можливість очистити некоректні або тестові записи без ручного редагування бази.

Вільні місця в ресурсах розраховуються відносно кількості активних користувачів. Якщо місткість укриття умовно становить 30 місць, система може показувати, скільки місць залишилося. Це демонструє не просто статичне відображення ресурсів, а зв'язок між обліком людей і інформаційним наповненням порталу.

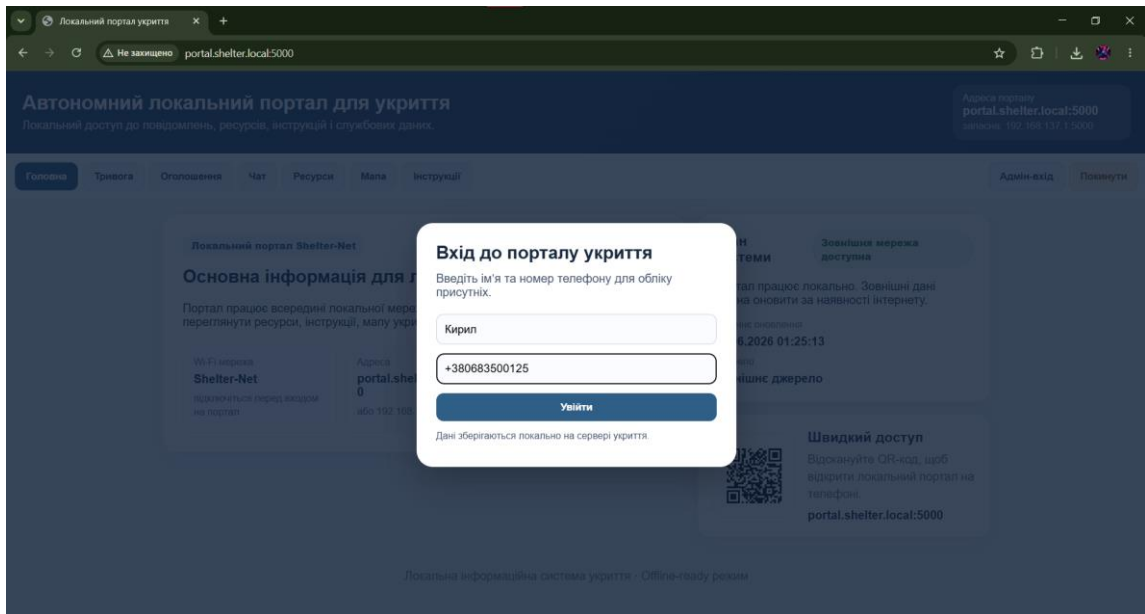


Рисунок 3.4 — Форма входу користувача до локального порталу

### 3.5 Реалізація розділу повітряної тривоги

Розділ повітряної тривоги призначений для швидкого перегляду стану для Тернопільської області. Він містить карту України з виділеним регіоном, великий текстовий статус, коротке повідомлення, час оновлення та джерело. Під час доопрацювання блок було скомпоновано компактніше, щоб він не займав надмірно багато місця і не виглядав розтягнутим [5].

Серверна логіка отримує дані із зовнішнього джерела тільки тоді, коли є доступ до інтернету. Після оновлення результат записується в таблицю `external_alerts`. Для користувача джерело показується не технічним формулюванням, а як «зовнішнє джерело». Це робить інтерфейс більш зрозумілим і не перевантажує його технічними деталями.

Якщо інтернет відсутній, система не припиняє роботу. Вона показує останній збережений стан і час його отримання. Такий підхід відповідає загальній логіці локального порталу: зовнішня синхронізація є корисною, але не є обов'язковою для функціонування системи. Користувачі все одно можуть переглядати інструкції, карту, чат і оголошення.

Історія оновлень обмежена чотирма записами. Це зроблено для того, щоб сторінка не ставала безкінечно довгою і не накопичувала зайву

					КС КРБ 123.270.00.00 ПЗ	38
Змн.	Арк.	№ докум.	Підпис			

інформацію. Для поточної задачі достатньо бачити останні зміни, а повна службова історія відображається в журналі для адміністратора.

Візуально блок тривоги має бути помітним, але не надмірним. Якщо тривоги немає, використовується зелений статус. Якщо тривога активна, використовується червоний акцент. Однак решта інтерфейсу залишається спокійною, щоб сторінка не виглядала перевантаженою кольорами (рис. 3.5).

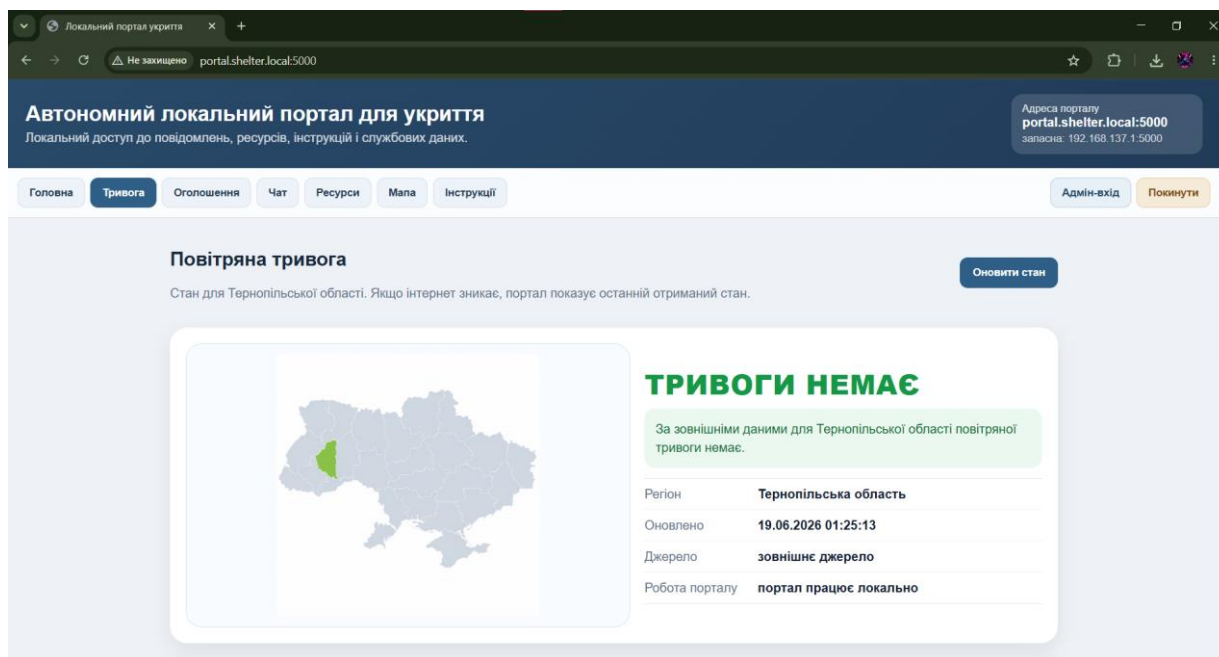


Рисунок 3.5 — Відображення стану повітряної тривоги

### 3.6 Реалізація оголошень, чату та інструкцій

Оголошення адміністратора реалізовані як окремий інформаційний модуль. Кожне оголошення має заголовок, основний текст і тип важливості. Звичайні оголошення використовуються для нейтральної інформації, важливі - для повідомлень, на які потрібно звернути увагу, а критичні - для термінових вказівок. При цьому кольори підібрані так, щоб важливі повідомлення були помітними, але не надто яскравими.

Адміністратор може редагувати оголошення повністю. Це означає, що змінюється не лише заголовок, а й текст та тип важливості. Така можливість

потрібна, тому що оголошення в укритті можуть швидко втрачати актуальність або вимагати уточнення. Видалення оголошення супроводжується красивим підтвердженням, а дія записується до журналу.

Локальний чат використовується для коротких повідомлень між користувачами. Він не залежить від зовнішніх месенджерів і працює всередині локальної мережі. Після доопрацювання історія повідомлень розташована вище, а поле введення - нижче. Це відповідає звичайній логіці спілкування: спочатку користувач читає історію, потім пише власне повідомлення.

Повідомлення чату прив'язане до користувача, який увійшов на портал. Це дозволяє не вводити ім'я в чаті окремо і зменшує кількість випадкових підписів. Якщо адміністратор бачить некоректне повідомлення, він може його видалити. Так само, як і інші службові дії, видалення записується до журналу (рис. 3.6).

Інструкції зберігають короткі правила користування системою та поведінки в укритті. Адміністратор може додавати, редагувати або видаляти інструкції. Це зручно, оскільки в реальному середовищі правила можуть змінюватися. Наприклад, можна додати інструкцію щодо підключення до мережі, дій при втраті інтернету або пошуку ресурсів у приміщенні.

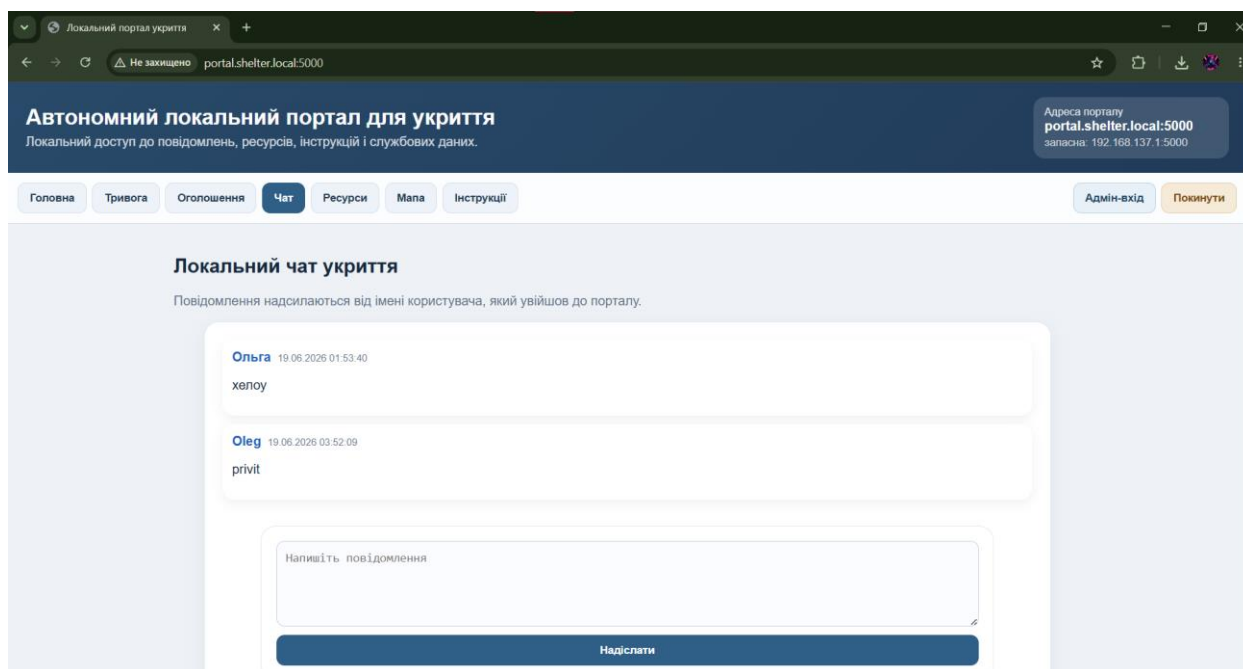


Рисунок 3.6 — Локальний чат укриття

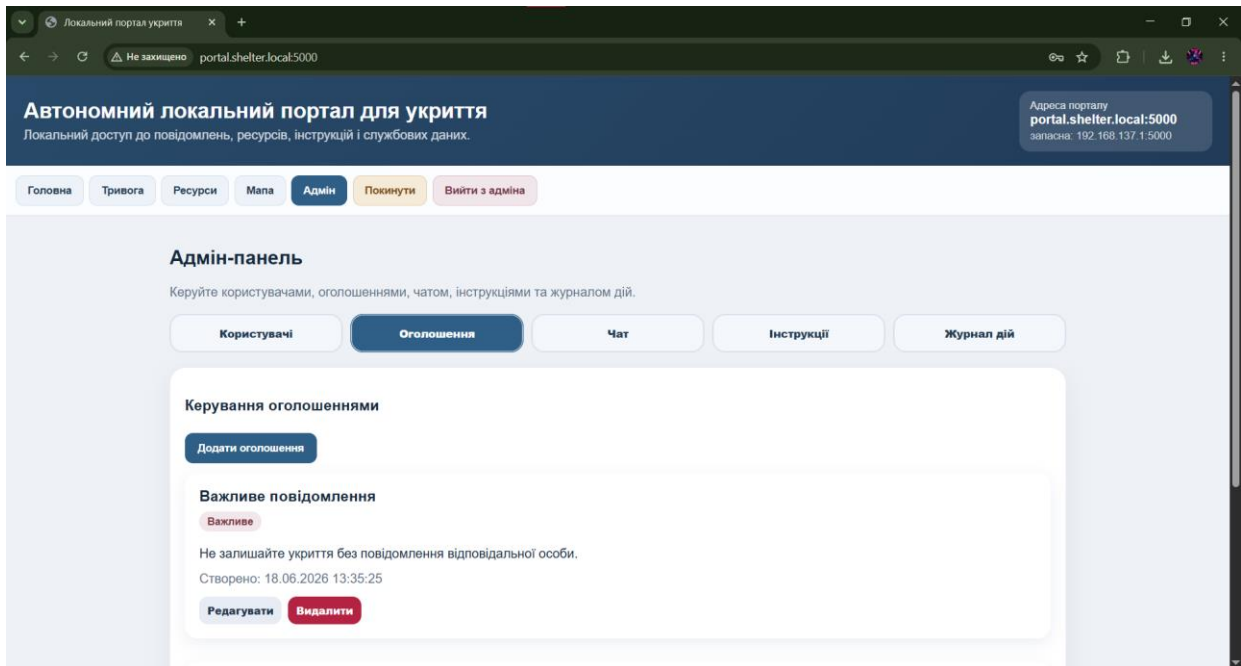


Рисунок 3.7 — Редагування оголошення адміністратором

### 3.7 Реалізація мапи укриття та ресурсів

Мапа укриття в системі виконує роль локального плану евакуації. Початковий варіант у вигляді окремих блоків було змінено на більш схожий на план будівлі. На схемі показані приміщення та функціональні зони: вхід, коридор, зона адміністратора, місця для людей, вода, аптечка, зарядна станція та запасний вихід. Це робить мапу кориснішою для користувача.

План має бути простим і зрозумілим, а не надмірно деталізованим. Його задача - дати швидке уявлення про розташування основних зон. У реальному використанні такий план можна замінити на точну схему конкретного укриття. У межах кваліфікаційної роботи він демонструє, що портал може зберігати й показувати локальну навігаційну інформацію навіть без інтернету.

Розділ ресурсів показує наявні матеріали та допоміжні засоби: воду, аптечку, павербанки, ліхтарики і вільні місця. Частина даних може бути статичною, а частина розраховується автоматично. Зокрема вільні місця пов'язані з кількістю активних користувачів у системі. Це показує, що модулі порталу можуть взаємодіяти між собою.

Оформлення ресурсів виконане у вигляді карток. Такий формат зручний для швидкого перегляду, оскільки кожна позиція має назву, кількість, стан і час оновлення. У майбутньому цей модуль можна розширити, додавши редагування ресурсів через адмін-панель або імпорт даних із локальної таблиці.

Для користувачів мапа і ресурси є одними з найпрактичніших розділів. Вони не потребують зовнішніх джерел, не залежать від стану інтернету й залишаються доступними в будь-якому режимі роботи порталу. Саме такі функції підсилюють основну ідею локальної інформаційної системи для укриття (рис. 3.8).

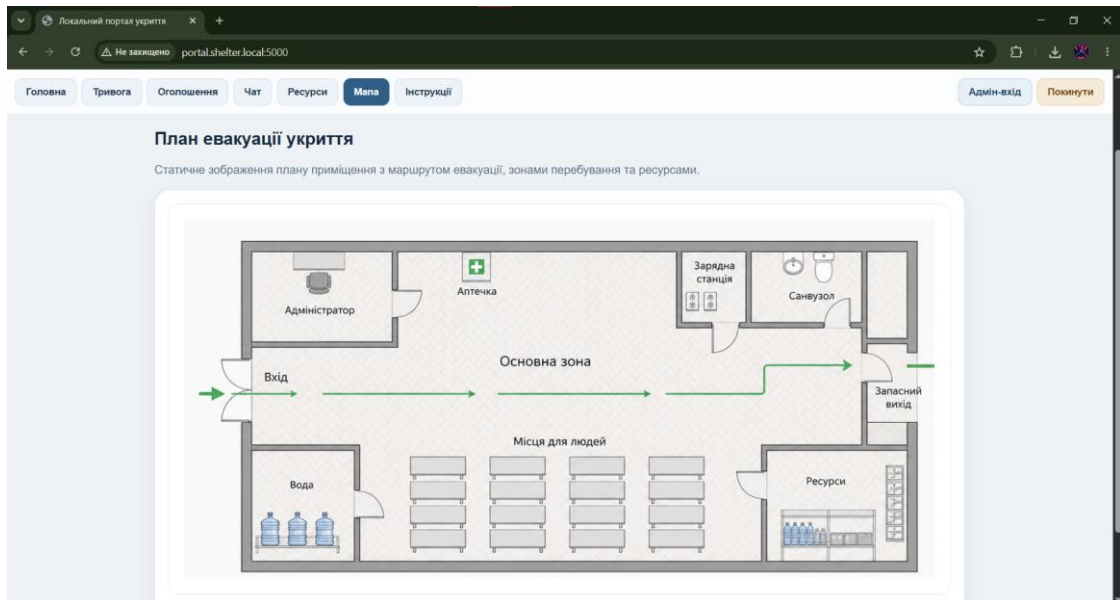


Рисунок 3.8 — План укриття у вигляді схеми евакуації

### 3.8 Реалізація адміністративної панелі та журналу дій

Адміністративна панель є службовою частиною порталу. Вона відкривається після введення пароля й приховує дублюючі вкладки, які вже доступні всередині панелі. Такий підхід робить інтерфейс чистішим: адміністратор не бачить зайві однакові розділи, а переходить між внутрішніми блоками керування.

У панелі є розділ користувачів, де показано ім'я, телефон, час входу та статус. Статуси «У системі» і «Офлайн» формуються на основі останньої активності. Тут же можна видалити користувача. Після видалення його пристрій повинен пройти вхід повторно. Це корисно під час тестування і в ситуаціях, коли в списку з'явилися зайві записи.

Розділ оголошень у панелі дозволяє переглядати, редагувати й видаляти повідомлення адміністратора. Розділ чату дозволяє видаляти окремі повідомлення. Розділ інструкцій дає змогу створювати нові записи, змінювати наявні й прибирати застарілі. Усі ці дії виконуються через модальні вікна, які мають однаковий стиль і не вибиваються з дизайну порталу (рис. 3.7).

Журнал дій відображає події, важливі для контролю системи. У ньому фіксується вхід адміністратора з прив'язкою до користувача, який перебуває в порталі. Також записуються редагування, видалення, додавання, запуск сервера та інші службові операції. Завдяки цьому адміністратор може бачити історію змін і пояснити, що саме відбувалося в системі.

У практичній реалізації журнал має значення ще й для захисту роботи. Він демонструє, що система має не лише зовнішній інтерфейс, а й внутрішню логіку обліку подій. Це підтверджує, що розробка є комп'ютерною системою з обробкою даних, а не набором статичних сторінок (рис. 3.9; рис. 3.10).

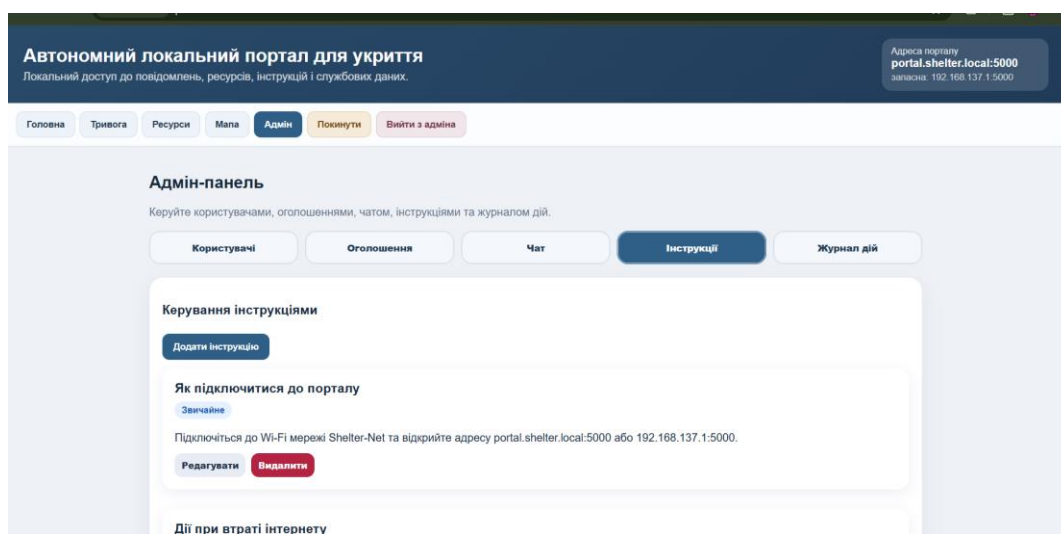


Рисунок 3.9 — Адміністративна панель portalу

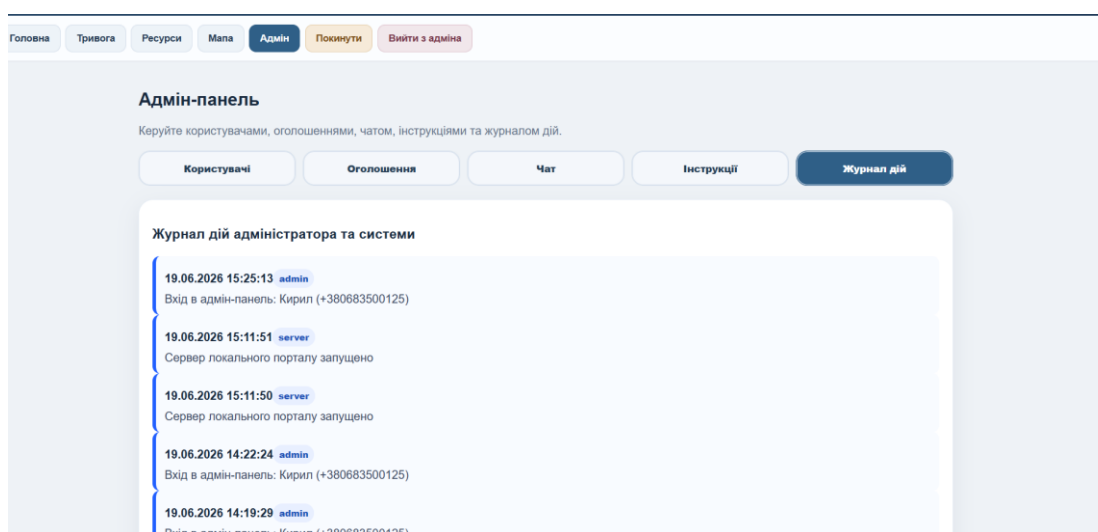


Рисунок 3.10 — Журнал дій адміністратора

### 3.9 Перевірка працездатності системи

Після реалізації основних функцій виконано перевірку роботи системи за кількома сценаріями. Спочатку перевіряється запуск серверної частини на Dell. Якщо в PowerShell відображаються локальні адреси запуску, це означає, що Flask-застосунок працює. Далі портал відкривається на самому сервері та з іншого пристрою, підключеного до Shelter-Net.

Наступним кроком перевіряється вхід користувача. Якщо ім'я введено, а номер має міжнародний формат, сервер створює запис у базі, форма входу зникає, і користувач потрапляє на головну сторінку. Якщо номер некоректний, система повинна показати повідомлення про помилку. Це підтверджує, що дані не приймаються без базової перевірки.

Далі перевіряється чат. Користувач надсилає повідомлення, після чого воно з'являється в історії від його імені. Потім виконується вхід адміністратора і перевіряється можливість видалення повідомлення. Після видалення повідомлення зникає з чату, а в журналі з'являється відповідна подія.

Окремо перевіряються оголошення та інструкції. Адміністратор створює новий запис, редагує його, змінює текст і тип важливості, а потім видаляє. Кожна дія повинна відображатися на сторінці та фіксуватися в журналі. Це підтверджує працездатність CRUD-функцій і правильний зв'язок між інтерфейсом, API та базою даних.

Стан тривоги перевіряється у двох варіантах: з доступом до зовнішньої мережі та без нього. За наявності інтернету система оновлює дані й зберігає запис. Без інтернету портал продовжує показувати останній збережений стан. Така перевірка підтверджує автономність системи та коректну роботу офлайн-режиму.

Під час тестування також перевіряється вигляд сторінок. Головна не повинна дублювати зайві кнопки, блок тривоги має бути компактним, чат повинен мати поле введення знизу, а мапа - вигляд плану будівлі. Це важливо,

тому що готова система оцінюється не лише за технічною працездатністю, а й за зручністю використання. Основні результати перевірки функцій системи наведено в таблиці 3.1.

Таблиця 3.1 — Перевірка основних функцій системи

Сценарій перевірки	Очікуваний результат	Стан
Відкриття порталу з телефона	Сторінка відкривається за локальною адресою	Виконується
Вхід користувача	Створюється запис у visitors, форма зникає	Виконується
Надсилання повідомлення в чат	Повідомлення з'являється від імені користувача	Виконується
Вхід адміністратора	З'являється адмін-панель, дія записується в журнал	Виконується
Редагування оголошення	Змінюється заголовок, текст і важливість	Виконується
Офлайн-режим	Портал відкривається і показує локально збережені дані	Виконується

### 3.10 Особливості демонстрації системи на захисті

Для демонстрації системи на захисті достатньо підготувати Dell-сервер, маршрутизатор із мережею Shelter-Net та один або два клієнтські пристрої. На сервері запускається команда `python app.py` з каталогу `backend`. Після запуску потрібно переконатися, що сервер слухає локальну адресу та порт 5000. На клієнтському пристрої відкривається адреса `portal.shelter.local:5000` або запасна адреса `192.168.137.1:5000`.

Першим демонстраційним кроком доцільно показати вхід користувача. Це одразу пояснює, що система веде облік людей. Після входу можна відкрити

головну сторінку, показати QR-код, стан системи та основні вкладки. Далі варто перейти до тривоги, показати мапу та останній збережений стан.

Наступним кроком можна показати чат. Повідомлення надсилається від імені користувача, який щойно увійшов. Це демонструє зв'язок між формою входу, базою даних і функцією чату. Після цього можна перейти до ресурсів і мапи укриття, пояснивши, що ці дані залишаються доступними без зовнішнього інтернету.

Окремо потрібно показати адміністративний режим. Після введення пароля відкривається адмін-панель, а кнопка входу зникає. У журналі має з'явитися подія про вхід адміністратора. Потім можна відредагувати оголошення, видалити тестове повідомлення в чаті або додати інструкцію. Кожна дія повинна бути зафіксована в журналі.

У поясненні варто наголосити, що система є локальною й автономною. Роутер створює мережу, але основну цінність забезпечує сервер Dell: він обробляє дані, зберігає інформацію та надає веб-портал. Саме тому розробка відповідає темі комп'ютерної системи локальних мережевих сервісів для укриттів.

					КС КРБ 123.270.00.00 ПЗ	
Змн.	Арк.	№ докум.	Підпис			47

## РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

У розділі наведено конкретні вимоги безпеки для приміщення укриття, у якому експлуатується комп'ютерна система локальних мережевих сервісів. Розглянуто робоче місце адміністратора, користувачів ПК, Dell OptiPlex 3050 Micro як локальний сервер, Wi-Fi роутер, електроживлення, кабелі, засоби пожежогасіння та порядок дій у разі несправності.

Питання сформовано відповідно до напрямів, зазначених у завданні з охорони праці: загальні вимоги безпеки для користувачів ПК, електробезпека під час роботи з комп'ютерною технікою та пожежна безпека приміщення. Вимоги подано у прикладному вигляді, тобто з урахуванням конкретного обладнання і реального розміщення його в укритті.

Нормативною основою розділу є НПАОП 0.00-7.15-18 щодо роботи з екранними пристроями, НПАОП 40.1-1.21-98 щодо безпечної експлуатації електроустановок споживачів, Правила пожежної безпеки в Україні та Правила експлуатації і типові норми належності вогнегасників [7–10].

### 4.1 Загальні вимоги безпеки з охорони праці для користувачів ПК

У приміщенні укриття користувачі працюють з екранними пристроями: ноутбуками, смартфонами, планшетами або монітором адміністратора. Згідно з НПАОП 0.00-7.15-18 екранний пристрій призначений для відтворення графічної або алфавітно-цифрової інформації, тому робоче місце адміністратора та місця тимчасового використання ПК розглядаються як місця роботи з екранними пристроями.

					КС КРБ 123.270.00.00 ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Омельченко К.Р			Безпека життєдіяльності, основи охорони праці	Літ.	Арк.	Аркушів
Перевірив		Стрембський М.О					48	
Консульт.		Гурик О.Я.				ТНТУ, каф. КС, гр. СІс-41		
Н. Контр.		Тиш Є.В.						
Затверд.		Осухівська Г.М.						

До початку роботи адміністратор перевіряє справність екрана, клавіатури, миші, кабелю живлення, блока живлення та доступ до кнопки вимкнення. На робочій поверхні не повинно бути рідин, зайвих кабелів, відкритих дротів, нестійко розміщених предметів або матеріалів, які можуть перекривати вентиляцію обладнання.

Робоче місце організовується так, щоб екран був добре видимий, зображення було чітким і стабільним, а яскравість та контрастність можна було відрегулювати відповідно до освітлення приміщення. Стіл повинен мати достатню площу для розміщення пристрою, миші, телефону та короткої інструкції дій. Крісло або місце сидіння має забезпечувати стійке положення користувача.

Під час роботи користувачі не втручаються у розміщення Dell-сервера, роутера, кабелів, розеток і подовжувачів. Вони не відкривають корпуси пристроїв, не перемикають живлення, не переносять обладнання і не підключають власні зарядні пристрої до подовжувача, який використовується для серверного вузла.

Після завершення роботи користувач закриває відкриті сторінки, від'єднує власний зарядний пристрій, якщо він більше не потрібний, і не залишає кабелі на проході. Адміністратор після завершення роботи перевіряє стан екрана, кабелів, блока живлення і розеток. Загальні вимоги безпеки та заборонені дії користувачів наведено в таблицях 4.1 і 4.2.

Таблиця 4.1 — Загальні вимоги безпеки для користувачів ПК

Елемент	Вимога	Виконання
Екран	Чітке зображення	Налаштувати яскравість
Робоче місце	Без зайвих предметів	Прибрати рідини і кабелі
Кабель	Ціла ізоляція	Не використовувати пошкоджений
Вентиляція	Отвори відкриті	Не накривати пристрої
Крісло	Стійке положення	Не працювати довго стоячи
Завершення	Вимкнення за потреби	Перевірити стан пристроїв

Таблиця 4.2 — Заборонені дії користувачів ПК

Заборонена дія	Причина
Відкривати корпус обладнання	Ризик ураження струмом
Торкатися розетки мокрими руками	Ризик електротравми
Ставити воду біля техніки	Ризик короткого замикання
Прокладати кабель через прохід	Ризик падіння людей
Самостійно переносити сервер	Ризик зупинки системи
Залишати перегрітий пристрій увімкненим	Ризик займання

#### 4.2 Вимоги електробезпеки для користувачів ПК

Електробезпека під час користування комп'ютерною технікою в укритті передбачає безпечну експлуатацію ПК, ноутбуків, моніторів, зарядних пристроїв, розеток, подовжувачів, Wi-Fi роутера та локального сервера Dell OptiPlex 3050 Micro. Основною метою є недопущення ураження електричним струмом, короткого замикання та перегріву обладнання.

Відповідно до НПАОП 40.1-1.21-98 роботи в електроустановках повинні виконуватися з дотриманням правил безпечної експлуатації. У межах цієї роботи користувачі не виконують ремонт і не обслуговують електроустановки. Вони користуються лише справними екранними пристроями та не мають доступу до відкритих електричних контактів [10].

Живлення комп'ютерної техніки допускається тільки від справної розетки. Не допускається використання розеток зі слідами плавлення, потемніння, іскріння, нестійкого контакту або нагрівання. Забороняється застосовувати саморобні перехідники, скрутки проводів, пошкоджені подовжувачі та кабелі з відкритими жилами.

Подовжувачі використовуються лише заводського виготовлення і в межах допустимого навантаження. До одного подовжувача не підключають одночасно надмірну кількість зарядних пристроїв, ноутбуків або інших

споживачів. Подовжувач не повинен лежати у проході, біля води або в місцях, де його можуть притиснути дверима чи меблями.

Серверний вузол розміщується так, щоб звичайні користувачі не мали потреби торкатися блока живлення, розетки, кабелів або роутера. Основні вимоги електробезпеки та прокладання кабелів наведено в таблицях 4.3 і 4.4.

Таблиця 4.3 — Вимоги електробезпеки до обладнання

Об'єкт	Вимога	Дія
Розетка	Без нагрівання	Не використовувати несправну
Кабель живлення	Ціла ізоляція	Не згинати і не перетискати
Блок живлення	Без перегріву	Не накривати
Подовжувач	Заводський, справний	Не перевантажувати
Корпус ПК	Без пошкоджень	Не відкривати користувачам
Мокрі руки	Робота заборонена	Не торкатися вилки

Таблиця 4.4 — Прокладання кабелів у приміщенні

Місце	Ризик	Рішення
Під столом	Перегин кабелю	Фіксація без натягу
Прохід	Спотикання	Кабель-канал або стіна
Біля сервера	Перегрів блока	Не накривати БЖ
Зарядна зона	Перевантаження	Обмежити кількість зарядок
Розподільна коробка	Відкриті контакти	Кришка закрита

У разі появи запаху горілого, іскріння, нагрівання розетки, блока живлення або подовжувача робота з обладнанням негайно припиняється. Якщо це можна зробити без ризику, несправний пристрій від'єднується від електромережі. Повторне увімкнення дозволяється тільки після огляду відповідальною особою.

### 4.3 Пожежна безпека з точки зору охорони праці

Пожежна безпека у приміщенні укриття під час експлуатації комп'ютерної техніки є складовою охорони праці, оскільки стосується захисту життя і здоров'я людей, справності електрообладнання, утримання шляхів евакуації та наявності первинних засобів пожежогасіння.

Основними джерелами пожежної небезпеки є блоки живлення ПК, зарядні пристрої, подовжувачі, розетки, кабелі живлення, Wi-Fi роутер, Dell-сервер і горючі матеріали, розміщені поруч із технікою. Небезпека може виникнути через перевантаження подовжувача, пошкоджену ізоляцію, нагрівання розетки, несправний блок живлення або перекриття вентиляційних отворів.

Згідно з Правилами пожежної безпеки в Україні у приміщенні має бути встановлено порядок користування електроспоживачами, порядок відключення обладнання після роботи або у разі аварійної ситуації, а також порядок утримання шляхів евакуації. Електрообладнання не розміщується біля паперу, картону, тканини, одягу, м'яких поверхонь або відкритих ємностей з водою. Обладнання не встановлюється в закриту нішу без вентиляції. Вентиляційні отвори комп'ютера, блока живлення і роутера мають залишатися відкритими. Шляхи евакуації повинні бути вільними. Основні пожежонебезпечні фактори наведено в таблиці 4.5.

Таблиця 4.5 — Пожежонебезпечні фактори у приміщенні

Фактор	Причина небезпеки	Профілактичний захід
Блок живлення	Перегрівання	Не накривати
Розетка	Іскріння	Використовувати справну
Подовжувач	Перевантаження	Не підключати зайве
Кабель	Пошкодження ізоляції	Прокладати без натягу
Вентиляція	Накопичення тепла	Не закривати отвори
Рідини	Коротке замикання	Не ставити біля електрики

Для приміщення з комп'ютерною технікою необхідно передбачити первинні засоби пожежогасіння. Для офісних приміщень з оргтехнікою передбачається один газовий вогнегасник із зарядом вогнегасної речовини 3 кг і більше на 20 м<sup>2</sup> площі підлоги [9]. Вимоги до розміщення вогнегасника та евакуаційних шляхів наведено в таблицях 4.6 і 4.7.

Таблиця 4.6 — Розміщення вогнегасника

Параметр	Вимога
Тип	Газовий ВВК-2
Кількість	Не менше 1 шт.
Місце	Біля виходу або на видному місці
Висота	Нижній торець до 1,5 м
Позначення	Знак на висоті 2–2,5 м
Доступ	Не перекривати меблями

Таблиця 4.7 — Вимоги до евакуаційних шляхів

Елемент	Вимога
Основний прохід	Вільний від кабелів і речей
Вихід	Не заблокований
Запасний вихід	Доступний
Кабелі	Прокладені вздовж стіни
План евакуації	Доступний для перегляду
Вогнегасник	Видимий і доступний

У разі появи запаху горілого, диму, іскріння, сильного нагрівання блока живлення, подовжувача або розетки адміністратор припиняє роботу з обладнанням. Якщо це можна зробити без ризику для людини, несправний пристрій відключається від електромережі. Користувачів не допускають до небезпечної зони.

Гасіння електрообладнання водою не допускається, оскільки це створює ризик ураження електричним струмом. Для електронної техніки застосовується вогнегасник відповідного типу. Якщо загоряння неможливо

ліквідувати без небезпеки для людей, першочерговим є безпечне виведення людей із приміщення та повідомлення відповідальної особи. Порядок дій у разі пожежної небезпеки наведено в таблиці 4.8.

Таблиця 4.8 — Порядок дій у разі пожежної небезпеки

Дія	Мета
Припинити роботу	Не погіршити ситуацію
Відключити живлення	Усунути джерело небезпеки
Попередити людей	Не допустити травмування
Звільнити прохід	Забезпечити евакуацію
Підготувати вогнегасник	Локалізувати загоряння
Не вмикати повторно	Дочекатися огляду

## ВИСНОВКИ

У кваліфікаційній роботі розглянуто створення комп'ютерної системи локальних мережевих сервісів для укриттів з обробкою даних на базі Dell OptiPlex 3050 Micro. У результаті аналізу предметної області визначено, що для укриття важливо мати не лише фізичну мережу, а й локальний інформаційний простір, який залишається доступним у разі нестабільного або відсутнього зовнішнього інтернету.

У першому розділі обґрунтовано призначення локального порталу, визначено функціональні та нефункціональні вимоги, розглянуто особливості автономної роботи та сформульовано задачу кваліфікаційної роботи. Показано, що система має забезпечувати доступ до оголошень, тривоги, чату, ресурсів, інструкцій, мапи укриття та адміністративних функцій.

У другому розділі спроектовано архітектуру системи. Визначено роль Dell-сервера, Wi-Fi маршрутизатора, Flask-застосунку, SQLite-бази та клієнтських браузерів. Описано структуру бази даних, логіку веб-інтерфейсу, адміністративну панель, журнал дій і алгоритм роботи в онлайн- та офлайн-режимах. Особливу увагу приділено тому, щоб звичайний користувач бачив простий інтерфейс, а службові функції були зосереджені в адмін-панелі.

У третьому розділі описано практичну реалізацію системи. Розглянуто підготовку апаратної платформи, налаштування локального доступу, запуск серверної частини, облік користувачів, роботу чату, оголошень, інструкцій, повітряної тривоги, мапи укриття та журналу дій. Показано, що система може працювати в локальній мережі, зберігати дані в SQLite та продовжувати роботу без зовнішнього інтернету.

Практичний результат роботи полягає у створенні працездатного прототипу локального порталу укриття. Система демонструє взаємодію апаратної платформи, локальної мережі, веб-сервера, бази даних і користувацького інтерфейсу.

					КС КРБ 123.270.00.00 ПЗ	55
Змн.	Арк.	№ докум.	Підпис	Дата		

Вона може бути використана як навчальний стенд, основа для подальшого вдосконалення або приклад локальної інформаційної системи для автономних умов.

Подальший розвиток системи може передбачати складнішу авторизацію адміністратора, шифрування збережених персональних даних, редагування ресурсів через адмін-панель, автоматичне резервне копіювання бази даних, підтримку декількох укриттів і розширення механізмів синхронізації із зовнішніми джерелами. Проте вже реалізований прототип підтверджує основну ідею: локальний сервер може забезпечити корисні інформаційні сервіси для укриття навіть без постійного доступу до інтернету.

					КС КРБ 123.270.00.00 ПЗ	
Змн.	Арк.	№ докум.	Підпис	Дата		56

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Методичні вказівки до виконання кваліфікаційної роботи бакалавра для здобувачів першого (бакалаврського) рівня вищої освіти за спеціальністю 123 «Комп'ютерна інженерія» усіх форм навчання / уклад.: Р. О. Жаровський, Н. С. Луцик, Г. М. Осухівська, А. М. Паламар, Є. В. Тиш. Тернопіль: ТНТУ, 2024. 39 с.
2. Flask Documentation. Pallets Projects. URL: <https://flask.palletsprojects.com/> (дата звернення: 12.02.2026).
3. SQLite Documentation. SQLite Consortium. URL: <https://www.sqlite.org/docs.html> (дата звернення: 21.03.2026).
4. Python 3 Documentation. Python Software Foundation. URL: <https://docs.python.org/3/> (дата звернення: 23.03.2026).
5. Microsoft. Windows Server documentation. URL: <https://learn.microsoft.com/windows-server/> (дата звернення: 01.05.2026).
6. MDN Web Docs. HTML, CSS, JavaScript. URL: <https://developer.mozilla.org/> (дата звернення: 03.02.2026).
7. НПАОП 0.00-7.15-18. Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями: наказ Міністерства соціальної політики України від 14.02.2018 № 207. URL: <https://zakon.rada.gov.ua/laws/show/z0508-18> (дата звернення: 05.05.2026).
8. Правила пожежної безпеки в Україні: наказ Міністерства внутрішніх справ України від 30.12.2014 № 1417. URL: <https://zakon.rada.gov.ua/laws/show/z0252-15> (дата звернення: 05.05.2026).
9. Правила експлуатації та типові норми належності вогнегасників: наказ Міністерства внутрішніх справ України від 15.01.2018 № 25. URL: <https://zakon.rada.gov.ua/laws/show/z0225-18> (дата звернення: 04.03.2026).
10. Правила безпечної експлуатації електроустановок споживачів: наказ Комітету по нагляду за охороною праці України від 09.01.1998 № 4. URL: <https://zakon.rada.gov.ua/laws/show/z0093-98> (дата звернення: 02.05.2026).

					КС КРБ 123.270.00.00 ПЗ	
Змн.	Арк.	№ докум.	Підпис	Дата		57

11. Закон України «Про захист персональних даних» від 1.04.2010 № 2297-VI. URL: <https://zakon.rada.gov.ua/laws/show/2297-17> (дата звернення: 15.05.2026).

12. ДБН В.2.2-5:2023. Захисні споруди цивільного захисту. Київ: Міністерство розвитку громад, територій та інфраструктури України, 2023.

13. Закон України «Про інформацію» від 02.10.1992 № 2657-XII. URL: <https://zakon.rada.gov.ua/laws/show/2657-12> (дата звернення: 11.05.2026).

14. Закон України «Про захист інформації в інформаційно-комунікаційних системах» від 05.07.1994 № 80/94-ВР. URL: <https://zakon.rada.gov.ua/laws/show/80/94-%D0%B2%D1%80> (дата звернення: 03.05.2026).

15. Закон України «Про електронні комунікації» від 16.12.2020 № 1089-IX. URL: <https://zakon.rada.gov.ua/laws/show/1089-20> (дата звернення: 02.04.2026).

16. Закон України «Про основні засади забезпечення кібербезпеки України» від 05.10.2017 № 2163-VIII. URL: <https://zakon.rada.gov.ua/laws/show/2163-19> (дата звернення: 10.02.2026).

17. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання. Київ: ДП «УкрНДНЦ», 2016.

18. ДСТУ 8302:2015. Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання. Київ: ДП «УкрНДНЦ», 2016.

19. ДСТУ 7299:2013. Дизайн і ергономіка. Робоче місце оператора. Київ: Мінекономрозвитку України, 2014.

20. ДСТУ 8604:2015. Робоче місце для виконання робіт у положенні сидячи. Загальні ергономічні вимоги. Київ: ДП «УкрНДНЦ», 2016.

21. Microsoft. Internet Information Services (IIS) documentation. URL: <https://learn.microsoft.com/iis/> (дата звернення: 12.03.2026).

22. Microsoft. Dynamic Host Configuration Protocol (DHCP) overview. URL: <https://learn.microsoft.com/windowsserver/networking/technologies/dhcp/dhcp-top> (дата звернення: 16.03.2026).

					КС КРБ 123.270.00.00 ПЗ	
Змн.	Арк.	№ докум.	Підпис	Дата		58

23. Microsoft. Domain Name System (DNS) overview. URL: <https://learn.microsoft.com/windows-server/networking/dns/dns-top> (дата звернення: 11.05.2026).

24. Cisco. What is a local area network (LAN)? URL: <https://www.cisco.com/c/en/us/products/switches/what-is-a-lan-local-area-network.html> (дата звернення: 12.04.2026).

25. Dell Technologies. OptiPlex 3050 Micro Owner's Manual. URL: <https://www.dell.com/support/home/uk-ua/product-support/product/optiplex-3050-micro/docs> (дата звернення: 10.02.2026).

26. Шингера Н. Я., Щербак Л. М. Конспект лекцій з дисципліни «Адміністрування комп'ютерних мережевих систем» для студентів денної форми навчання спеціальності 7.05010201 «Комп'ютерні системи та мережі». Тернопіль: ТНТУ, 2016. URL: <http://elartu.tntu.edu.ua/handle/123456789/17633> (дата звернення: 13.04.2026).

27. Чайковський А. В., Жаровський Р. О., Лецишин Ю. З. Методичні вказівки до виконання лабораторних робіт з дисципліни «Дослідження і проєктування комп'ютерних систем та мереж» для здобувачів другого (магістерського) рівня вищої освіти спеціальності 123 «Комп'ютерна інженерія» усіх форм навчання. Тернопіль: ТНТУ, 2021. 94 с. URL: <https://elartu.tntu.edu.ua/handle/lib/42882> (дата звернення: 15.04.2026).

28. Жаровський Р. О., Чайковський А. В., Лецишин Ю. З., Варавін А. В. Методичні вказівки для виконання курсового проєкту з дисципліни «Дослідження і проєктування комп'ютерних систем та мереж» для здобувачів другого (магістерського) рівня вищої освіти спеціальності 123 «Комп'ютерна інженерія» усіх форм навчання. Тернопіль: ТНТУ, 2023. 31 с. URL: <https://elartu.tntu.edu.ua/handle/lib/42879> (дата звернення: 12.03.2026).

Додаток А  
Технічне завдання

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії

Кафедра комп'ютерних систем та мереж

**“Затверджую”**

Завідувач кафедри КС

\_\_\_\_\_ Осухівська Г.М.

“ 2 ” лютого 2026 р.

Комп'ютерна система локальних мережевих сервісів для укриттів з обробкою  
даних на базі Dell OptiPlex 3050 Micro

**ТЕХНІЧНЕ ЗАВДАННЯ**

на 10 листках

**Вид робіт:**

Кваліфікаційна робота

На здобуття освітнього ступеня «Бакалавр»

**Спеціальність 123 «Комп'ютерна інженерія»**

«УЗГОДЖЕНО»

«ВИКОНАВЕЦЬ»

Керівник кваліфікаційної роботи

Студент групи Сіс-41

\_\_\_\_\_ к.т.н., доц. Стрембіцький М.О

\_\_\_\_\_ Омельченко К.Р

“ 2 ” лютого 2026 р.

“ 2 ” лютого 2026 р.

**Тернопіль 2026**

## 1 Загальні відомості

### 1.1 Повна назва та її умовне позначення

Повна назва теми кваліфікаційної роботи: «Комп'ютерна система локальних мережевих сервісів для укриттів з обробкою даних на базі Dell OptiPlex 3050 Micro».

Умовне позначення кваліфікаційної роботи: КС КРБ 123.270.00.00.

### 1.2 Виконавець

Студент групи СІ-41, факультету комп'ютерно-інформаційних систем і програмної інженерії, кафедри комп'ютерних систем та мереж, Тернопільського національного технічного університету імені Івана Пулюя, Омельченко Кіріл Романович.

### 1.3 Підстава для виконання роботи

Підставою для виконання кваліфікаційної роботи є наказ по університету №4/9-189 від 24.04.2026 р., а також затверджена тема кваліфікаційної роботи бакалавра за спеціальністю 123 «Комп'ютерна інженерія».

### 1.4 Планові терміни початку та завершення роботи

Плановий термін початку виконання кваліфікаційної роботи – 26.01.2026 р.

Плановий термін завершення виконання кваліфікаційної роботи – 21.06.2026 р.

## 1.5 Порядок оформлення та пред'явлення результатів роботи

Порядок оформлення пояснювальної записки та графічного матеріалу здійснюється у відповідності до чинних норм та правил ISO, ЄСКД, ЄСПД, ДСТУ та методичних вказівок кафедри комп'ютерних систем та мереж.

Пред'явлення проміжних результатів роботи з виконання кваліфікаційної роботи здійснюється у відповідності до графіка, затвердженого керівником роботи. Попередній захист кваліфікаційної роботи відбувається при готовності пояснювальної записки, графічного матеріалу та програмно-апаратної демонстрації системи.

Пред'явлення результатів кваліфікаційної роботи відбувається шляхом захисту на відповідному засіданні ЕК з демонстрацією основних результатів за допомогою графічного матеріалу та прикладу роботи локальних мережевих сервісів.

## 2 Призначення і цілі створення системи

### 2.1 Призначення системи

Комп'ютерна система локальних мережевих сервісів для укриттів призначена для організації автономного доступу користувачів до інформаційного веб-порталу та локальних сервісів у межах захищеного приміщення або укриття.

Система, що розробляється, призначена для:

- надання користувачам доступу до локального веб-інтерфейсу через внутрішню мережу;
- збереження та відображення актуальної інформації для осіб, які перебувають в укритті;
- підтримки роботи в умовах наявності та відсутності доступу до Інтернету;

– адміністрування локальних повідомлень, інструкцій, контактів, ресурсів і журналу подій.

## 2.2 Мета створення системи

Метою кваліфікаційної роботи є розроблення та налаштування комп'ютерної системи локальних мережевих сервісів для укриттів з обробкою даних на базі Dell OptiPlex 3050 Micro, яка забезпечує роботу локального інформаційного порталу, файлового доступу та базових мережевих служб у автономному режимі.

## 2.3 Характеристика об'єкта

Об'єктом розроблення є локальна інформаційна система укриття, яка поєднує сервер Dell OptiPlex 3050 Micro, мережеве обладнання, веб-сервер, локальне сховище даних і клієнтські пристрої користувачів. Система повинна забезпечувати доступ до інформаційних ресурсів через браузер без необхідності встановлення додаткового програмного забезпечення на пристрої користувачів.

## 3 Вимоги до системи

### 3.1 Вимоги до системи в цілому

#### 3.1.1. Вимоги до структури та функціонування системи

Комп'ютерна система локальних мережевих сервісів повинна складатись з:

- серверної частини на базі Dell OptiPlex 3050 Micro;
- операційної системи Windows Server;
- веб-сервера IIS для розміщення локального веб-порталу;

- локального сховища даних для файлів, таблиць, повідомлень та інструкцій;
- мережевого обладнання для підключення користувацьких пристроїв;
- адміністративного інтерфейсу для оновлення інформації та керування контентом.

### 3.1.2. Вимоги до способів та засобів зв'язку між компонентами системи

Основна вимога, яка ставиться до способів та засобів інформаційного обміну, полягає у забезпеченні стабільної роботи локальної мережі та узгодженості обміну даними між сервером, роутером і клієнтськими пристроями.

Доступ користувачів до веб-порталу повинен здійснюватися через локальну IP-адресу або локальне доменне ім'я. Обмін файлами може здійснюватися через мережеве сховище, доступне в межах локальної мережі.

### 3.1.3. Вимоги до режимів функціонування системи

Для системи визначено такі режими функціонування:

- онлайн-режим;
- офлайн-режим;
- аварійний режим.

В онлайн-режимі система повинна забезпечувати синхронізацію або оновлення інформації із зовнішніх джерел за наявності доступу до Інтернету. На веб-порталі має відображатися позначка про актуальність інформації або час останнього оновлення.

В офлайн-режимі система повинна забезпечувати автономну роботу локального веб-порталу, доступ до останніх збережених даних, локальних інструкцій, контактів, планів евакуації, повідомлень адміністратора та інших ресурсів, необхідних для користувачів укриття.

Аварійний режим характеризується частковою відмовою програмних або апаратних компонентів. У цьому режимі система повинна зберігати доступ

до базових інформаційних матеріалів та забезпечувати можливість відновлення після усунення несправності.

#### 3.1.4. Вимоги по діагностуванню системи

Для діагностування системи повинні використовуватись інструменти моніторингу операційної системи Windows Server, журнал подій, засоби перевірки роботи ІІS, мережевих параметрів, доступності локального веб-порталу та мережевого сховища.

Інструменти діагностування повинні забезпечувати можливість перевірки стану сервера, мережевого підключення, доступності веб-сторінок, прав доступу до спільних папок і коректності роботи адміністративних функцій.

#### 3.1.5. Перспективи розвитку, проектування системи

Система може бути розширена шляхом додавання бази даних, модуля авторизації користувачів, журналу підключень, локального чату, інтеграції з резервним живленням, автоматичного створення резервних копій і використання альтернативних апаратних платформ, зокрема Raspberry Pi або іншого міні-комп'ютера.

### 3.2 Показники призначення

Система повинна передбачати можливість масштабування та адаптації під різні типи укриттів. Можливості масштабування повинні забезпечуватися засобами використовуваного базового програмного і технічного забезпечення.

Основними показниками призначення системи є:

- доступність локального веб-порталу для користувачів у межах внутрішньої мережі;
- можливість автономної роботи без доступу до Інтернету;
- збереження останньої актуальної інформації на сервері;
- простота адміністрування та оновлення локального контенту;

– підтримка підключення з телефонів, ноутбуків та інших клієнтських пристроїв.

### 3.3 Вимоги до надійності

Система повинна забезпечувати працездатність та відновлення своїх функцій при виникненні таких ситуацій:

- збій у системі електропостачання апаратної частини;
- відмова або перезавантаження роутера чи серверного обладнання;
- помилки в роботі операційної системи або веб-сервера;
- тимчасова відсутність доступу до Інтернету;
- порушення доступу до окремих файлів або каталогів локального сховища.

Для захисту апаратури від стрибків напруги і комутаційних завад повинні застосовуватись мережеві фільтри або джерела безперебійного живлення.

### 3.4 Вимоги до безпеки

Зовнішні елементи технічних засобів системи, що перебувають під напругою, повинні мати захист від випадкового дотику, а самі технічні засоби повинні експлуатуватися відповідно до вимог безпеки електрообладнання.

Система електроживлення повинна забезпечувати захисне вимикання при перевантаженнях і коротких замиканнях, а також можливість аварійного ручного вимикання.

Загальні вимоги пожежної безпеки повинні відповідати чинним нормам. У разі виникнення пожежі після зняття електроживлення має бути доступне застосування первинних засобів пожежогасіння.

### 3.5 Вимоги до експлуатації, технічного обслуговування, ремонту і зберігання компонентів системи

Мікроклімат у приміщеннях, де експлуатується серверне обладнання, повинен відповідати нормам виробничого мікроклімату. Рекомендовані умови експлуатації:

- температура повітря в межах від +10 °С до +35 °С;
- відносна вологість повітря в межах від 30 % до 80 %;
- відсутність прямого потрапляння вологи на сервер, мережеве обладнання та кабельні з'єднання;
- достатня вентиляція місця розміщення Dell OptiPlex 3050 Micro.

Періодичне технічне обслуговування використовуваних технічних засобів має проводитися відповідно до вимог технічної документації, але не рідше ніж один раз на рік.

### 3.6 Вимоги до захисту інформації від несанкціонованого доступу

Система повинна забезпечувати захист від несанкціонованого доступу на рівні локальної мережі, операційної системи та адміністративного інтерфейсу. Доступ до функцій редагування інформації повинен бути дозволений лише адміністратору.

Компоненти підсистеми захисту повинні забезпечувати:

- ідентифікацію адміністратора;
- перевірку повноважень при роботі з адміністративними функціями;
- розмежування доступу між адміністратором і користувачем;
- обмеження доступу до службових файлів і налаштувань сервера;
- захист локального сховища від випадкового або несанкціонованого видалення даних.

### 3.7 Вимоги по збереженню інформації при аваріях

Інформація при виникненні аварійних ситуацій повинна бути збережена на локальному носії або резервному носії. Для важливих даних повинно

передбачатися резервне копіювання конфігураційних файлів, локальних повідомлень, інструкцій, таблиць ресурсів і журналів подій.

### 3.8 Вимоги по стандартизації і уніфікації

Система повинна відповідати вимогам ергономіки і зручності користування за умови комплектування якісним обладнанням, що має необхідні сертифікати відповідності та безпеки. Інтерфейс користувача повинен бути зрозумілим, простим і придатним для використання з мобільних пристроїв.

### 3.9 Вимоги до функцій, що виконуються системою

Система повинна виконувати такі функції:

- надання доступу до локального веб-порталу;
- відображення повідомлень адміністратора;
- збереження та показ інструкцій, контактів, ресурсів і планів дій в укритті;
- робота в офлайн-режимі без доступу до Інтернету;
- підтримка оновлення інформації адміністратором;
- забезпечення доступу до локального мережевого сховища;
- підтримка базових мережевих сервісів для локальної мережі.

## 4 Вимоги до документації

Документація повинна відповідати вимогам ЄСКД, ЄСПД, ДСТУ та методичним вказівкам до виконання кваліфікаційної роботи бакалавра.

Комплект документації повинен складатися з:

- пояснювальної записки;
- технічного завдання;
- графічного матеріалу;
- демонстраційних матеріалів для захисту.

До графічного матеріалу доцільно включити:

- структурну схему комп'ютерної системи локальних мережевих сервісів;
- блок-схему роботи локального веб-порталу;
- схему підключення обладнання в локальній мережі;
- схему взаємодії користувача, адміністратора та сервера.

\*Примітка: у комплект документації можуть вноситися зміни та доповнення в процесі розробки.

## 5 Стадії та етапи проектування

Таблиця 1 – Стадії та етапи виконання кваліфікаційної роботи бакалавра

№ етапу	Назва етапу виконання кваліфікаційної роботи бакалавра	Термін виконання
1	Розробка технічного завдання	26.01 – 02.02
2	Робота над першим розділом «Аналіз технічного завдання»	03.02 – 15.02
3	Робота над другим розділом «Проектна частина»	20.04 – 25.04
4	Робота над третім розділом «Практична частина»	26.04 – 05.05
5	Робота над четвертим розділом «Безпека життєдіяльності, основи охорони праці»	07.05 – 25.05
6	Оформлення пояснювальної записки і графічного матеріалу	26.05 – 07.06
7	Перевірка на академічний плагіат, перевірка керівником та консультантами	08.06 – 14.06
8	Попередній захист кваліфікаційної роботи бакалавра	15.06 – 21.06
9	Захист кваліфікаційної роботи бакалавра	25.06

## 6 Додаткові умови виконання кваліфікаційної роботи

Під час виконання кваліфікаційної роботи у дане технічне завдання можуть вноситися зміни та доповнення, які не суперечать затвердженій темі, меті роботи та вимогам до оформлення кваліфікаційної роботи бакалавра.

## Додаток Б

### Лістинг програмного коду

```
from flask import Flask, render_template, request, redirect, url_for,
session, send_from_directory
import sqlite3
import os
from datetime import datetime

app = Flask(__name__)
app.secret_key = "local_shelter_system_secret_key"

BASE_DIR = os.path.dirname(os.path.abspath(__file__))
DB_PATH = os.path.join(BASE_DIR, "shelter.db")
STORAGE_DIR = os.path.join(BASE_DIR, "storage")

ADMIN_LOGIN = "admin"
ADMIN_PASSWORD = "admin123"

def get_connection():
    """Створення підключення до локальної бази даних SQLite."""
    conn = sqlite3.connect(DB_PATH)
    conn.row_factory = sqlite3.Row
    return conn

def init_db():
    """Ініціалізація таблиць бази даних."""
    conn = get_connection()
    cur = conn.cursor()

    cur.execute("""
        CREATE TABLE IF NOT EXISTS messages (
            id INTEGER PRIMARY KEY AUTOINCREMENT,
            title TEXT NOT NULL,
            body TEXT NOT NULL,
            created_at TEXT NOT NULL,
            source TEXT NOT NULL
        )
    """)

    cur.execute("""
        CREATE TABLE IF NOT EXISTS system_status (
            id INTEGER PRIMARY KEY AUTOINCREMENT,
            mode TEXT NOT NULL,
            last_sync TEXT NOT NULL
        )
    """)

    cur.execute("SELECT COUNT(*) FROM system_status")
    count = cur.fetchone()[0]

    if count == 0:
        cur.execute(
            "INSERT INTO system_status (mode, last_sync) VALUES (?, ?)",
            ("offline", datetime.now().strftime("%d.%m.%Y %H:%M"))
        )

    conn.commit()
    conn.close()
```

```

@app.route("/")
def index():
    """Головна сторінка локального вебпорталу укриття."""
    conn = get_connection()
    messages = conn.execute(
        "SELECT * FROM messages ORDER BY id DESC"
    ).fetchall()

    status = conn.execute(
        "SELECT * FROM system_status ORDER BY id DESC LIMIT 1"
    ).fetchone()

    conn.close()

    resources = []
    if os.path.exists(STORAGE_DIR):
        resources = os.listdir(STORAGE_DIR)

    return render_template(
        "index.html",
        messages=messages,
        status=status,
        resources=resources
    )

@app.route("/login", methods=["GET", "POST"])
def login():
    """Авторизація адміністратора системи."""
    error = None

    if request.method == "POST":
        login_value = request.form.get("login")
        password_value = request.form.get("password")

        if login_value == ADMIN_LOGIN and password_value == ADMIN_PASSWORD:
            session["admin"] = True
            return redirect(url_for("admin_panel"))
        else:
            error = "Неправильний логін або пароль"

    return render_template("login.html", error=error)

@app.route("/logout")
def logout():
    """Вихід адміністратора з системи."""
    session.pop("admin", None)
    return redirect(url_for("index"))

@app.route("/admin", methods=["GET", "POST"])
def admin_panel():
    """Панель адміністратора для створення локальних повідомлень."""
    if not session.get("admin"):
        return redirect(url_for("login"))

    if request.method == "POST":
        title = request.form.get("title")
        body = request.form.get("body")
        created_at = datetime.now().strftime("%d.%m.%Y %H:%M")

        conn = get_connection()

```

```

        conn.execute(
            "INSERT INTO messages (title, body, created_at, source) VALUES
            (?, ?, ?, ?)",
            (title, body, created_at, "Адміністратор")
        )
        conn.commit()
        conn.close()

        return redirect(url_for("admin_panel"))

    conn = get_connection()
    messages = conn.execute(
        "SELECT * FROM messages ORDER BY id DESC"
    ).fetchall()
    conn.close()

    return render_template("admin.html", messages=messages)

@app.route("/delete/<int:message_id>")
def delete_message(message_id):
    """Видалення повідомлення адміністратором."""
    if not session.get("admin"):
        return redirect(url_for("login"))

    conn = get_connection()
    conn.execute("DELETE FROM messages WHERE id = ?", (message_id,))
    conn.commit()
    conn.close()

    return redirect(url_for("admin_panel"))

@app.route("/status", methods=["POST"])
def change_status():
    """Зміна режиму роботи системи: онлайн або офлайн."""
    if not session.get("admin"):
        return redirect(url_for("login"))

    mode = request.form.get("mode")
    last_sync = datetime.now().strftime("%d.%m.%Y %H:%M")

    conn = get_connection()
    conn.execute(
        "INSERT INTO system_status (mode, last_sync) VALUES (?, ?)",
        (mode, last_sync)
    )
    conn.commit()
    conn.close()

    return redirect(url_for("admin_panel"))

@app.route("/resources/<path:filename>")
def download_resource(filename):
    """Надання доступу до локальних файлів ресурсу."""
    return send_from_directory(STORAGE_DIR, filename, as_attachment=False)

if __name__ == "__main__":
    if not os.path.exists(STORAGE_DIR):
        os.makedirs(STORAGE_DIR)

    init_db()
    app.run(host="0.0.0.0", port=5000, debug=True)

```

## Лістинг Б.2 – Шаблон головної сторінки templates/index.html

```
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <title>Локальний портал укриття</title>
  <link rel="stylesheet" href="/static/style.css">
</head>
<body>
  <header>
    <h1>Локальний інформаційний портал укриття</h1>

    {% if status.mode == "online" %}
      <p class="online">Режим роботи: онлайн</p>
    {% else %}
      <p class="offline">Режим роботи: офлайн</p>
    {% endif %}

    <p>Останнє оновлення даних: {{ status.last_sync }}</p>
  </header>

  <main>
    <section>
      <h2>Повідомлення</h2>

      {% for message in messages %}
        <article class="card">
          <h3>{{ message.title }}</h3>
          <p>{{ message.body }}</p>
          <small>
            Джерело: {{ message.source }} |
            Дата: {{ message.created_at }}
          </small>
        </article>
      {% else %}
        <p>Повідомлення відсутні.</p>
      {% endfor %}
    </section>

    <section>
      <h2>Ресурси укриття</h2>

      <ul>
        {% for file in resources %}
          <li>
            <a href="/resources/{{ file }}">{{ file }}</a>
          </li>
        {% else %}
          <li>Файли ресурсів відсутні.</li>
        {% endfor %}
      </ul>
    </section>

    <section>
      <h2>Інструкції</h2>
      <p>У разі відсутності інтернету користувачі можуть отримувати останню збережену інформацію через локальну мережу.</p>
      <p>Для доступу потрібно підключитися до локальної Wi-Fi мережі укриття та відкрити адресу сервера у браузері.</p>
    </section>
  </main>

  <footer>
    <a href="/login">Вхід адміністратора</a>
```

```
</footer>
</body>
</html>
```

### Лістинг Б.3 – Шаблон сторінки авторизації templates/login.html

```
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <title>Авторизація адміністратора</title>
  <link rel="stylesheet" href="/static/style.css">
</head>
<body>
  <main class="login-box">
    <h1>Вхід адміністратора</h1>

    {% if error %}
      <p class="error">{{ error }}</p>
    {% endif %}

    <form method="post">
      <label>Логін</label>
      <input type="text" name="login" required>

      <label>Пароль</label>
      <input type="password" name="password" required>

      <button type="submit">Увійти</button>
    </form>

    <p><a href="/">Повернутися на головну</a></p>
  </main>
</body>
</html>
```

### Лістинг Б.4 – Шаблон панелі адміністратора templates/admin.html

```
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <title>Панель адміністратора</title>
  <link rel="stylesheet" href="/static/style.css">
</head>
<body>
  <header>
    <h1>Панель адміністратора</h1>
    <a href="/logout">Вийти</a>
  </header>

  <main>
    <section>
      <h2>Створення повідомлення</h2>

      <form method="post">
        <label>Заголовок повідомлення</label>
        <input type="text" name="title" required>

        <label>Текст повідомлення</label>
        <textarea name="body" rows="5" required></textarea>

        <button type="submit">Опублікувати</button>
      </form>
    </section>
  </main>
</body>
</html>
```

```

</section>

<section>
  <h2>Режим роботи системи</h2>

  <form method="post" action="/status">
    <select name="mode">
      <option value="online">Онлайн</option>
      <option value="offline">Офлайн</option>
    </select>

    <button type="submit">Змінити режим</button>
  </form>
</section>

<section>
  <h2>Опубліковані повідомлення</h2>

  {% for message in messages %}
    <article class="card">
      <h3>{{ message.title }}</h3>
      <p>{{ message.body }}</p>
      <small>{{ message.created_at }}</small>
      <br>
      <a href="/delete/{{ message.id }}">Видалити</a>
    </article>
  {% else %}
    <p>Повідомлення відсутні.</p>
  {% endfor %}
</section>
</main>
</body>
</html>

```

### Лістинг Б.5 – Файл стилів static/style.css

```

body {
  margin: 0;
  font-family: Arial, sans-serif;
  background: #f2f4f7;
  color: #222;
}

header {
  background: #1f2937;
  color: white;
  padding: 20px;
  text-align: center;
}

main {
  width: 90%;
  max-width: 1000px;
  margin: 20px auto;
}

section {
  background: white;
  padding: 20px;
  margin-bottom: 20px;
  border-radius: 8px;
}

.card {

```

```

    border: 1px solid #d1d5db;
    border-radius: 8px;
    padding: 15px;
    margin-bottom: 15px;
    background: #ffffff;
}

.online {
    color: #22c55e;
    font-weight: bold;
}

.offline {
    color: #ef4444;
    font-weight: bold;
}

.error {
    color: #ef4444;
    font-weight: bold;
}

form {
    display: flex;
    flex-direction: column;
    gap: 10px;
}

input,
textarea,
select {
    padding: 10px;
    border: 1px solid #9ca3af;
    border-radius: 5px;
}

button {
    padding: 10px;
    background: #2563eb;
    color: white;
    border: none;
    border-radius: 5px;
    cursor: pointer;
}

button:hover {
    background: #1d4ed8;
}

a {
    color: #2563eb;
    text-decoration: none;
}

.login-box {
    max-width: 400px;
    margin-top: 80px;
}

```