

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: «Розробка програмного забезпечення»

Група: 4РП-07

# Дипломний проєкт

здобувача освіти денної форми навчання

РП.07.07.000.ДП

**КУЧЕРЕНКО СЕРГІЯ  
ЮРІЙОВИЧА**

м. Одеса  
2024 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 121 «Комп'ютерна інженерія»

Освітньо-професійна програма: «Інженерія програмного забезпечення»

Група: 4РП-07

**ПОЯСНЮВАЛЬНА ЗАПИСКА**

до дипломного проекту на тему:


**Розробка автоматизованої  
веб-системи для ОСББ**

Проектний матеріал складається з пояснювальної записки на 79 сторінках та графічного (презентаційного) матеріалу на 12 аркушах (слайдах).

Дипломник  (Кучеренко С. Ю.)

Керівник  (Жадан А. С.)

**Консультанти:**

з економічного розділу  (Іванченков В. С.)

з розділу охорони праці та техніки безпеки  (Чорновол Н. І.)

з нормоконтролю  (Петрашова В. І.)

старший консультант  (Кривченко Ю. В.)

**До захисту допущений**

Голова циклової комісії  (Кривченко Ю. В.)

Завідувач відділення  (Скорнякова О. В.)

Захист «25» 06 2024 р.

Протокол **ЕК** № 2

Оцінка **ЕК** 4 (добре) / 205.

Секретар **ЕК** 

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Відділення комп'ютерних систем Комісія КТ та ПІ  
Спеціальність 121 «Інженерія програмного забезпечення»  
Освітня програма «Розробка програмного забезпечення»

ЗАТВЕРДЖУЮ:

Заст. дир. з НВР Беркань І. В.  
« 15 » 01 2024 року

**ЗАВДАННЯ**

**на дипломний проєкт**

Здобувачеві освіти Кучеренко Сергію Юрійовичу

1. Тема проєкту Розробка автоматизованої веб-системи для ОСББ

Затверджена наказом по коледжу від « 02 » листопада 2023 р., наказ № 244-А2-ОД

2. Термін здачі закінченого проєкту 10.06.2024

3. Вихідні дані до проєкту \_\_\_\_\_

1. Реалізувати мінімалістичний графічний інтерфейс засобами фреймворку Bootstrap

2. Створити базу даних засобами СУБД MySQL

3. Розробити веб-застосунок на базі MVC-Micro-framework засобами мови PHP

4. Застосовувати принципи ООП з метою підтримки коду

4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити)

1. Аналіз предметної області. 2. Технології та засоби розробки (проєктування).

3. Проєктування веб-дизайну. 4. Проєктування архітектури веб-застосунку.

5. Розробка MVC-застосунку. 6. Тестування створеного веб-застосунку.

7. Економічний розрахунок. 8. Аспекти охорона праці та техніки безпеки

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

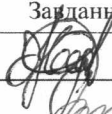
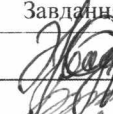
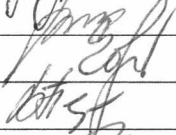
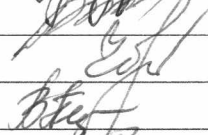

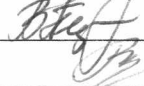




Презентація Power Point – 12 слайдів

(Предметна область; Обрані технології розробки; Розділи веб-системи; Схема бази даних;

Архітектура веб-системи; Обробка запитів; Приклад дошки; Налаштування адміністратора;

Реєстрація за ключем)

6. Консультанти по проекту, із зазначенням розділів проекту, що їх стосується

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Основний розділ	Жадан А. С.		
Економічний розділ	Іванченков В. С.		
Розділ охорони праці	Чорновол Н. І.		
Нормоконтроль	Петрашова В. І.		
Старший консультант	Кривченко Ю. В.		

7. Дата видачі завдання

15.04.24

Керівник

Жадан А. С.

  
(підпис)

Завдання прийняв до виконання

Кучеренко С. Ю.

(підпис)

КАЛЕНДАРНИЙ ПЛАН

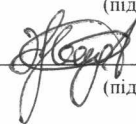
№ з/р	Назва етапів дипломного проекту (роботи)	Термін виконання етапів дипломного проекту (роботи)	Відмітка про виконання
1	Формування вступу	29.04.24	виконано
2	Аналіз предметної області	10.05.24	виконано
3	Підбір технічної літератури	19.05.24	виконано
4	Вибір технологій та засобів розробки (проектування)	20.05.24	виконано
5	Проектування веб-інтерфейсу веб-застосунку	22.05.24	виконано
6	Проектування архітектури веб-застосунку	24.05.24	виконано
7	Розробка MVC-застосунку	27.05.24	виконано
8	Тестування створеного веб-застосунку	29.05.24	виконано
9	Оформлення пояснювальної записки	31.05.24	виконано
10	Оформлення графічної (презентаційної) частини	01.06.24	виконано
11	Економічний розрахунок	02.06.24	виконано
12	Опис охорони праці та техніки безпеки	09.06.24	виконано
13	Аналіз результатів проектування	13.06.24	виконано
14	Підготовка доповіді для захисту	16.06.24	виконано

Дипломник



(підпис)

Керівник

  
(підпис)



# ЗМІСТ

ВСТУП .....	7
1 ОСНОВНИЙ РОЗДІЛ .....	9
1.1 Аналіз предметної області .....	9
1.1.1 Наявні рішення на ринку .....	9
1.1.1 Обрані засоби та технології розробки .....	15
1.2 Проєктування веб-системи .....	20
1.2.1 Технічне завдання на розробку .....	20
1.2.2 Проєктування дизайну веб-системи .....	21
1.2.3 Проєктування бази даних веб-системи .....	23
1.2.4 Проєктування архітектури веб-системи .....	27
1.3 Реалізація веб-системи .....	30
1.3.1 Запуск серверу .....	30
1.3.2 Ядро MVC-застосунку .....	31
1.3.3 Кредити БД .....	32
1.3.4 Роутинг MVC-застосунку .....	32
1.3.5 Моделі (Models) MVC-застосунку .....	33
1.3.6 Шаблонізатор .....	35
1.3.7 Представлення (View) у MVC-застосунку .....	37
1.3.8 Контролери (Controllers) у MVC-застосунку .....	40
1.4 Мануальне тестування (Manual QA) .....	45
1.4.1 Огляд головної сторінки .....	46
1.4.2 Огляд сторінки реєстрації та авторизації .....	46
1.4.3 Огляд кабінету адміністратора .....	47
1.4.4 Огляд карток .....	48
1.4.5 Огляд сторінок з дошками повідомлень .....	50
2 ЕКОНОМІЧНИЙ РОЗДІЛ .....	52
2.1 Резюме .....	52
2.2 Визначення трудомісткості розробки програмного забезпечення .....	52

2.3 Розрахунок ціни програмного продукту .....	55
<b>3 РОЗДІЛ ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ .....</b>	<b>58</b>
3.1 Вступ .....	58
3.2 Аналіз дії небезпечних та шкідливих чинників на працівника під час роботи .....	58
3.3 Вимоги до виробничого середовища .....	58
3.3.1 Мікrokлімат .....	58
3.3.2 Шум .....	59
3.3.3 Освітлення .....	60
3.3.4 Електробезпека .....	60
3.3.5 Організація робочого місця програміста .....	61
3.4 Пожежна безпека .....	62
<b>ВИСНОВКИ .....</b>	<b>63</b>
<b>ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ .....</b>	<b>64</b>
ДОДАТОК А. Програмний код основної логіки веб-застосунку .....	65
ДОДАТОК Б. Слайди мультимедійної презентації .....	74

## ВСТУП

У сучасних умовах еволюції технологій управління багатоквартирними будинками (ОСББ), автоматизовані веб-системи стають важливими інструментами, що активно привертають увагу учасників до впровадження, використання, оновлення та підтримки новаторських рішень для оптимізації управління житловим фондом. Збільшення популярності таких веб-систем породжує потребу в ефективних засобах для управління ОСББ. Веб-орієнтовані платформи грають ключову роль у спрощенні процесів, забезпечуючи комфортність, швидкість та ефективність у взаємодії з ОСББ.

Ці веб-платформи можуть містити різноманітні модулі, такі як системи для автоматизованого внесення звітності, управління фінансами, спільного доступу до документів тощо. Ці інструменти допомагають учасникам ОСББ ефективно управляти житловим фондом, підвищуючи продуктивність та забезпечуючи високий рівень обслуговування мешканців.

Автоматизація через веб-системи дає можливість членам ОСББ оперативно реагувати на зміни в управлінні, оптимізувати процеси та прискорювати прийняття рішень. Це сприяє підвищенню конкурентоспроможності та стабільності в управлінні мультиквартирними будинками.

Системи управління ОСББ можуть включати функції для збору та аналізу даних, автоматизації процесів прийняття рішень, а також забезпечення доступу до інформації та сервісів для різних категорій користувачів.

Метою даного дослідження є створення автоматизованої веб-системи для ОСББ, яка використовуватиме передові технології для оптимізації управління багатоквартирними будинками. Система має надавати зручні та ефективні інструменти для управління, забезпечення доступу до інформації та підтримки членів ОСББ.

У ході дослідження основних проблем управління багатоквартирними будинками, аналізу аналогів та інструментів розробки був застосований

					<i>РП 07. 07 000. 00 ДП ПЗ</i>	Арк.
						7
Ізм.	Лист	№ докум.	Підпис	Дата		

системний підхід. Для створення бази даних використовувалася сучасна технологія реляційних баз даних. Для розробки програмного продукту використовувалися передові методи об'єктно-орієнтованого програмування на мовах програмування PHP та JavaScript.

Практичне значення отриманих результатів полягає в тому, що розроблена система буде корисною для членів ОСББ, допомагаючи їм ефективно управляти багатоквартирними будинками та забезпечувати високий рівень обслуговування мешканців.

					<i>РП 07. 07 000. 00 ДП ПЗ</i>	Арк.
						8
Ізм.	Лист	№ докум.	Підпис	Дата		

# 1 ОСНОВНИЙ РОЗДІЛ

## 1.1 Аналіз предметної області

### 1.1.1 Наявні рішення на ринку

Розглянемо кілька альтернатив, які можна знайти на першій сторінці результатів у Google під запитом "ОСББ".

Перший аналог – “МОЄОСББ”.

"МОЄОСББ" - це онлайн сервіс, призначений для об'єднань співвласників багатоквартирних будинків (ОСББ) та управляючих компаній. Цей сервіс забезпечує широкий спектр можливостей, включаючи бухгалтерський облік, взаємодію з мешканцями, облік доходів та витрат, нарахування комунальних платежів та багато іншого. МОЄОСББ робить управління будинком більш ефективним та зручним, а також сприяє підвищенню прозорості роботи та довіри мешканців до управління будинком. Він доступний для користування на різних пристроях, таких як комп'ютери, телефони та планшети, і не вимагає встановлення додаткового програмного забезпечення [1].

На рис. 1.1 зображено стартову сторінку “МОЄОСББ”.

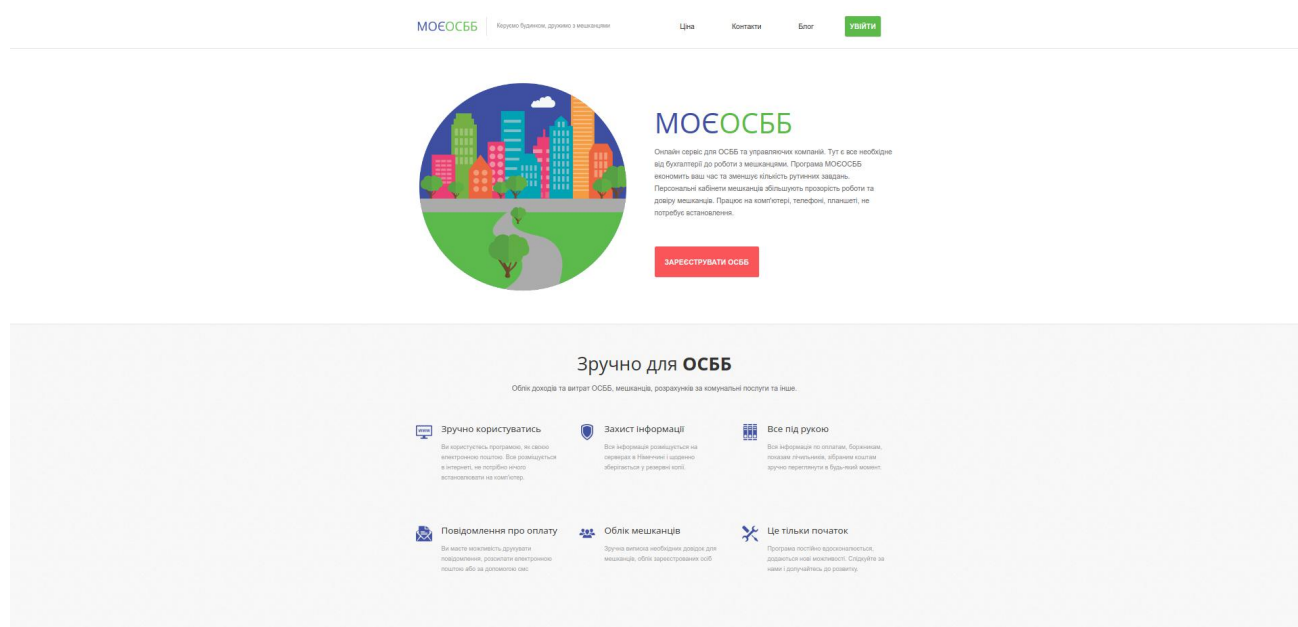


Рисунок 1.1. Стартова сторінка “МОЄОСББ”

Ізм.	Лист	№ докум.	Підпис	Дата

Переваги:

1. Комплексність сервісу: “МОЄОСББ” надає широкий спектр функцій, включаючи облік доходів і витрат, комунальних платежів, звітності для мешканців тощо.
2. Зручність користування: Система легко доступна через Інтернет і не вимагає встановлення додаткового програмного забезпечення.
3. Захист інформації: Інформація зберігається на серверах в Німеччині, що забезпечує певний рівень конфіденційності та безпеки даних.
4. Доступність: Мешканці мають доступ до інформації про своє ОСББ та розрахунки за комунальні послуги у будь-який час і з будь-якого пристрою з Інтернет-підключенням.

Недоліки:

1. Необхідність Інтернет-підключення: Для користування системою потрібний доступ до Інтернету, що може бути недоступним у деяких обставинах.
2. Відсутність нативних застосунків: Хоча система доступна на різних пристроях, відсутність нативних застосунків може зробити користування менш зручним для деяких користувачів.
3. Можливість технічних проблем: Залежність від Інтернет-з'єднання може призвести до недоступності системи у випадку технічних проблем або перебоїв в мережі.
4. Обмеження в регіональному плані: Система може бути менш ефективною в країнах або регіонах, де доступ до Інтернету обмежений або ненадійний.

Другий аналог – “ОСББ-online”.

"ОСББ-online" – це веб-орієнтована система, призначена для ведення бухгалтерського обліку та управління об'єднаннями співвласників багатоквартирних будинків (ОСББ). Вона надає широкий спектр функцій, включаючи облік доходів та витрат, нарахування внесків, ведення звітності, а також можливості комунікації з мешканцями. Система доступна через веб-

браузер, що дозволяє користувачам отримувати доступ до неї з будь-якого пристрою з підключенням до Інтернету [2].

На рис. 1.2 зображено стартову сторінку “ОСББ-online”.

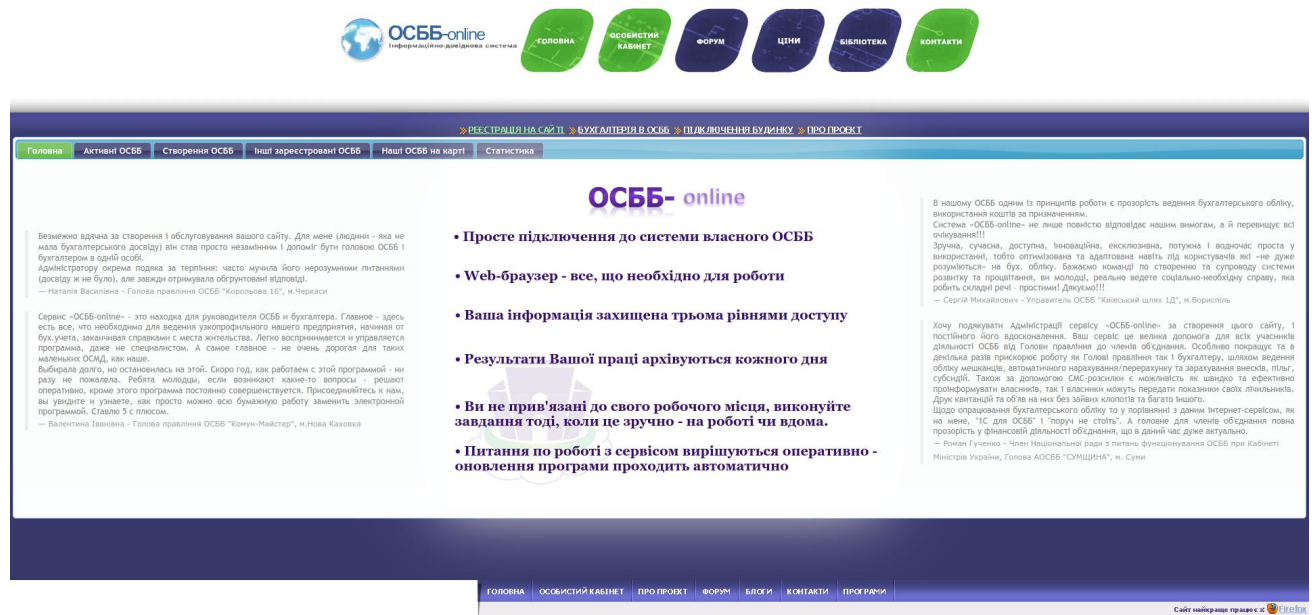


Рисунок 1.2. Стартова сторінка “ОСББ-online”

### Переваги:

1. Комплексність та зручність використання: Сайт "ОСББ-online" надає широкий спектр функцій для ведення бухгалтерського обліку та управління ОСББ, і при цьому легкий у використанні навіть для користувачів без бухгалтерського досвіду.
2. Операційна підтримка та оновлення: Адміністрація сайту надає оперативну підтримку та забезпечує постійне вдосконалення сервісу.
3. Захист даних та доступність: Інформація захищена трьома рівнями доступу, а робота з сайтом можлива з будь-якого пристрою через веб-браузер.

### Недоліки:

1. Залежність від Інтернет-підключення: Для користування сайтом потрібне постійне підключення до Інтернету, що може створити проблеми в умовах обмеженого доступу до мережі.

					<b>РП 07. 07 001. 00 ДП ПЗ</b>	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		11

2. Можливість технічних проблем: Залежність від роботи веб-сервера може призвести до недоступності сайту у випадку технічних проблем або перебоїв в роботі сервера.

3. Обмеження в функціональності: Сайт може не мати деяких функцій, які можуть бути доступні в інших системах бухгалтерського обліку.

Третій аналог – “Мій Дім Online”.

"Мій Дім Online" – це веб-орієнтована система, яка надає послуги для ОСББ (Об'єднання співвласників багатоквартирного будинку) та ЖБК (Житлово-будівельні кооперативи). Вона спрямована на автоматизацію обліку, звітності та комунікаційних процесів, пов'язаних з управлінням житловими будинками. Сервіс надає можливості для бухгалтерів, керівників ОСББ та мешканців для зручного та ефективного ведення обліку, взаємодії та спілкування [3].

На рис. 1.3 зображено стартову сторінку “Мій дім Online”.

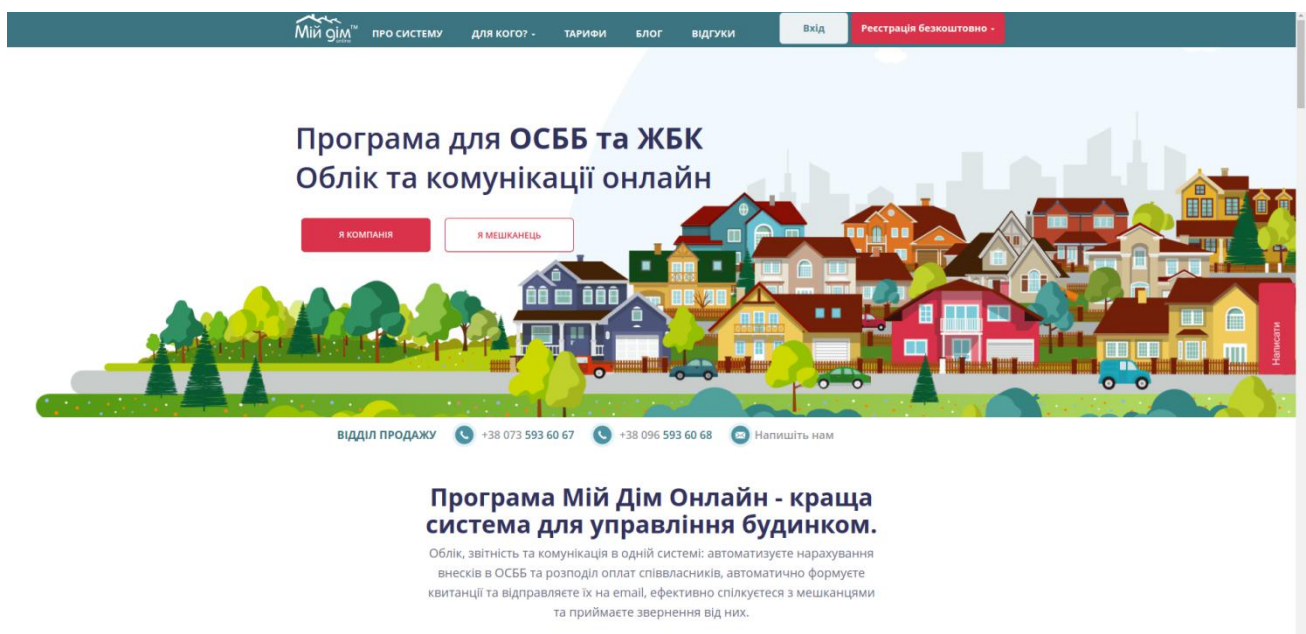


Рисунок 1.3. Стартова сторінка “Мій дім Online”

Переваги:

1. Автоматизація обліку: Система дозволяє автоматизувати розрахунки внесків у ОСББ та розподіл оплат між співвласниками, що значно спрощує бухгалтерську роботу.

					РП 07. 07 001. 00 ДП ПЗ	Арк.
						12
Ізм.	Лист	№ докум.	Підпис	Дата		

2. Зручна комунікація: Мешканці можуть легко спілкуватися з управлінням будинку та подавати звернення через онлайн диспетчерську, що забезпечує ефективну взаємодію.
3. Різні рівні доступу: Система надає різні рівні доступу до даних для різних користувачів, що забезпечує захист конфіденційності та можливість індивідуального керування правами доступу.
4. Економія часу: Автоматизація процесів та доступність системи 24/7 дозволяють заощадити час управлінням та мешканцям.

Недоліки:

1. Необхідність навчання: Для ефективного використання системи може знадобитися час на навчання користувачів, особливо якщо вони не мають попереднього досвіду роботи з подібними системами.
2. Залежність від інтернету: Використання системи передбачає постійний доступ до Інтернету, що може бути проблематичним у випадку недоступності мережі.
3. Можливість помилок: Автоматизація процесів може призвести до помилок у випадку неправильного налаштування системи або невірною введення даних.

Проведемо порівняльний аналіз аналогів.

У табл. 1.1 зображено порівняння аналогів.

Таблиця 1.1. Порівняння аналогів

Характеристика	МОЄОСББ	Мій Дім Онлайн	ОСББ-online
Цільова аудиторія	ОСББ та управляючі компанії	ОСББ та ЖБК	ОСББ та ЖБК
Основні функції	Облік, комунікація, бухгалтерія	Облік, комунікація, бухгалтерія	Облік, комунікація, бухгалтерія

Доступність	Інтернет-сервіс	Інтернет-сервіс	Інтернет-сервіс
Прозорість	Так	Так	Так
Зручність користування	Так	Так	Так
Можливість автоматизації	Так	Так	Так
Захист даних	Так	Так	Так
Підтримка	Так	Так	Так
Мобільний доступ	Так	Так	Так
Реєстраційна плата	Безкоштовно	Безкоштовно	Безкоштовно
Додаткові можливості	Персональні кабінети мешканців, економія часу	Не вказано	Автоматичний розподіл оплати з реєстрів та банківських виписок, SMS-розсилка, опитування, мобільний застосунок

Цілком можливо, що користувачі можуть сприймати переваги та недоліки цієї веб-орієнтованої веб-системи по-різному в залежності від їхніх конкретних потреб та умов. Порівнюючи сервіси Мій Дім Online, ОСББ-online та МОЄОСББ, можна відзначити, що вони дійсно спрямовані на полегшення управління та обліку для ОСББ та управляючих компаній. Однак, слід зазначити, що деякі з них можуть мати обмеження в функціональності або складнощі в інтеграції з

іншими системами. Наприклад, один з недоліків може полягати в обмеженій підтримці або технічних проблемах, які можуть виникнути при використанні платформи. Також важливо враховувати можливість нестабільності або недоліків у забезпеченні безпеки даних, що може стати проблемою для користувачів.

### 1.1.1 Обрані засоби та технології розробки

Розглянемо топ мов програмування, які можна знайти на першій сторінці результатів у Google під запитом "Top programming languages".

Візьмемо статистику з сайту "ZTM".

"Zero to Mastery" – це веб-сайт та платформа для навчання програмуванню, створена Андреем Неагоеєм, що пропонує різноманітні курси з програмування та розробки. За допомогою відеоуроків, практичних завдань та спільноти, учасники можуть вивчити різні галузі програмування "з нуля до майстерності" [16].

На рис. 1.4 зображено топ мов програмування на думку ZTM за критерієм вакансій у США.

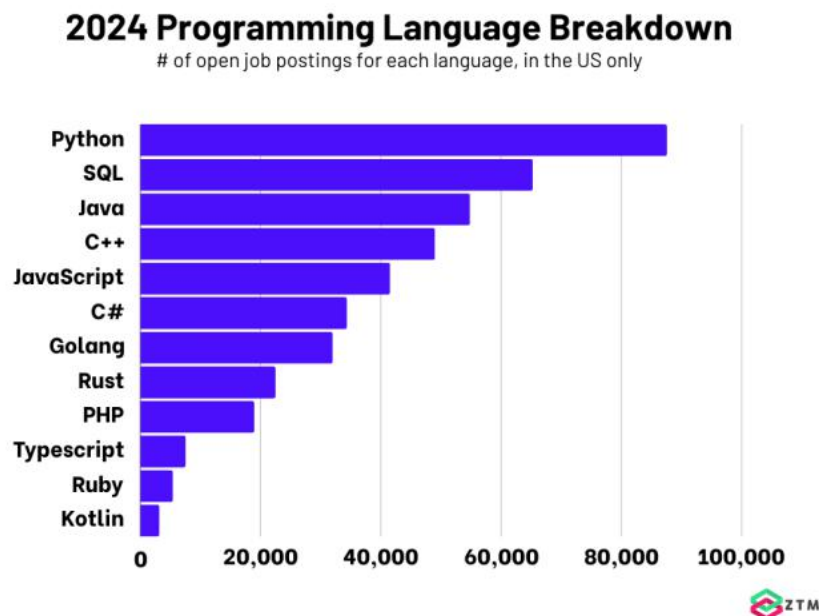


Рисунок 1.4. Топ мов програмування на думку ZTM за критерієм вакансій у США

Мова програмування PHP залишається важливою у світі програмування, враховуючи її присутність у топ-10 по кількості вакансій, що свідчить про поширеність та популярність цієї технології в ринку праці.

Розглянемо обраний стек розробки.

UI/ UX засоби.

В рамках цієї роботи, UI/UX засоби включають HTML, CSS та Bootstrap. HTML відповідає за створення структури та вмісту веб-сторінок, CSS – за їх зовнішній вигляд, а Bootstrap спрощує розробку адаптивних та привабливих сайтів [4, 5, 6, 7].

Front-End засоби.

В рамках цієї роботи, Front-End засоби для клієнтської сторони включають JavaScript та бібліотеку jQuery. JavaScript забезпечує можливість створення інтерактивності на сторінках та маніпулювання елементами HTML, а jQuery спрощує роботу з HTML, обробку подій [8, 9, 10].

Засоби БД.

В рамках цієї роботи, БД засоби включають SQL мову та СУБД MySQL. SQL відповідає за взаємодію з реляційними базами даних, а СУБД MySQL – за зберігання даних у веб-застосунках [11, 12, 13].

Back-End засоби.

В рамках цієї роботи, Back-End засоби для серверної сторони включають мову програмування PHP. PHP забезпечує можливість створення динамічного контенту та роботи з базами даних [14, 15].

На рис. 1.4 зображено схему обраних засобів та технологій розробки.



Рисунок 1.4. Обранці засоби та технології розробки

В рамках цієї роботи основою мовою програмування буде PHP.

Переваги:

1. Простота вивчення: PHP має простий і зрозумілий синтаксис, що дозволяє швидко навчитися розробці веб-застосунків для початківців програмістів.
2. Велика спільнота: PHP має велику та активну спільноту розробників, яка постійно підтримує мову, створює нові бібліотеки та фреймворки, а також надає допомогу у вирішенні проблем.
3. Широке застосування: PHP використовується для створення різноманітних веб-застосунків, від невеликих сайтів до великих корпоративних проєктів.
4. Інтеграція з базами даних: PHP має вбудовану підтримку для роботи багатьма різновидами баз даних, що робить його зручним для розробки динамічних веб-сайтів.

Недоліки:

1. Змішана структура коду: PHP допускає вбудовання PHP-коду безпосередньо в HTML, що може призводити до складності і погіршення читабельності коду.
2. Неоднорідність: Історично PHP розвивався швидко, іноді без чіткого плану, що призвело до наявності різних підходів та функцій, які можуть виявитися застарілими.
3. Проблеми з безпекою: Неправильно написаний код на PHP може призвести до серйозних проблем з безпекою, таких як вразливості SQL-ін'єкцій або вразливості типу "Cross-Site Scripting" (XSS).
4. Низька продуктивність: У порівнянні з деякими іншими мовами програмування, такими як Java або C++, PHP може працювати повільніше, особливо при обробці великих обсягів даних.

Порівняння мови PHP з мовою Python.

У таблиці 1.2 зображено порівняння аналогів.

Характеристика	PHP	Python
Синтаксис	Зорієнтований на HTML, менш зручний для вивчення	Простий та чистий, легко вивчити та читати
Типізація	Слабка, динамічна	Сильна, динамічна
Тип виконання	Використовується як мова з інтерпретатором та компілятором (залежно від веб-сервера)	Інтерпретована
Парадигми	Процедурне, об'єктно-орієнтоване, функціональне	Об'єктно-орієнтоване, процедурне, функціональне

Використання	Веб-розробка, особливо для серверної частини	Різноманітність застосування, від веб-розробки до наукових досліджень
Швидкодія	Швидший для обробки веб-запитів	Зазвичай повільніший ніж PHP
Веб-розробка	Має багато вбудованих функцій для взаємодії з вебом	Є багато фреймворків, таких як Django та Flask
Розширення	Велика кількість розширень та бібліотек для різних задач	Має також широкий вибір бібліотек та модулів
Спільнота	Велика та активна	Також велика, але зосереджена на різних галузях
Використання в наукових дослідженнях	Не так поширений для цієї мети	Широко використовується у наукових дослідженнях та аналізі даних
Робота з базами даних	Має вбудовану підтримку баз даних, таких як MySQL	Має також різні бібліотеки для роботи з базами даних

За допомогою таблиці порівняння мов програмування PHP та Python можна зробити кілька висновків. По-перше, PHP частіше використовується для веб-розробки, зокрема для створення серверної частини веб-застосунків, оскільки має вбудовані функції для взаємодії з вебом та базами даних. По-друге, Python є більш універсальною мовою, яка знаходить застосування в різних галузях, включаючи наукові дослідження та аналіз даних, а також веб-розробку

за допомогою різноманітних фреймворків. Нарешті, хоча PHP може бути швидшим для обробки веб-запитів, Python зазвичай має більш чистий та простий синтаксис, що полегшує розробку та підтримку коду.

## 1.2 Проєктування веб-системи

### 1.2.1 Технічне завдання на розробку

Технічне завдання включає в себе наступні аспекти:

1. Мінімалістичний графічний інтерфейс (GUI): Система повинна мати простий та зрозумілий інтерфейс, який дозволить користувачам легко навігувати та виконувати потрібні дії без зайвих складнощів.
2. Простий користувальницький досвід (UI): Користувальницький інтерфейс повинен бути зорієнтований на простоту та зручність використання, з мінімальною кількістю необхідних кроків для виконання потрібних завдань.
3. Наявність інформативних карток (користувача та будинку): Система повинна надавати інформацію у вигляді карток, які відображатимуть основні дані про користувача (наприклад, особисті дані, історія платежів) та про будинок (наприклад, інформація про споживання енергії, розмір комунальних платежів тощо).
4. Наявність дошок з повідомленнями за категоріями (скарги, новини тощо): Система повинна мати можливість відображення повідомлень користувачам у вигляді категоризованих дошок, де можуть бути розміщені новини, скарги, оголошення тощо.
5. CMS система адміністратора: Для зручного управління системою потрібно реалізувати Content Management System (CMS) для адміністраторів. Ця система повинна дозволяти додавати, видаляти та редагувати контент на сайті, керувати правами доступу користувачів та забезпечувати інші адміністративні функції.

На рис. 1.6 зображено майбутню систему навігації по застосунку.

					<b>РП 07. 07 001. 00 ДП ПЗ</b>	Арк.
						20
Ізм.	Лист	№ докум.	Підпис	Дата		

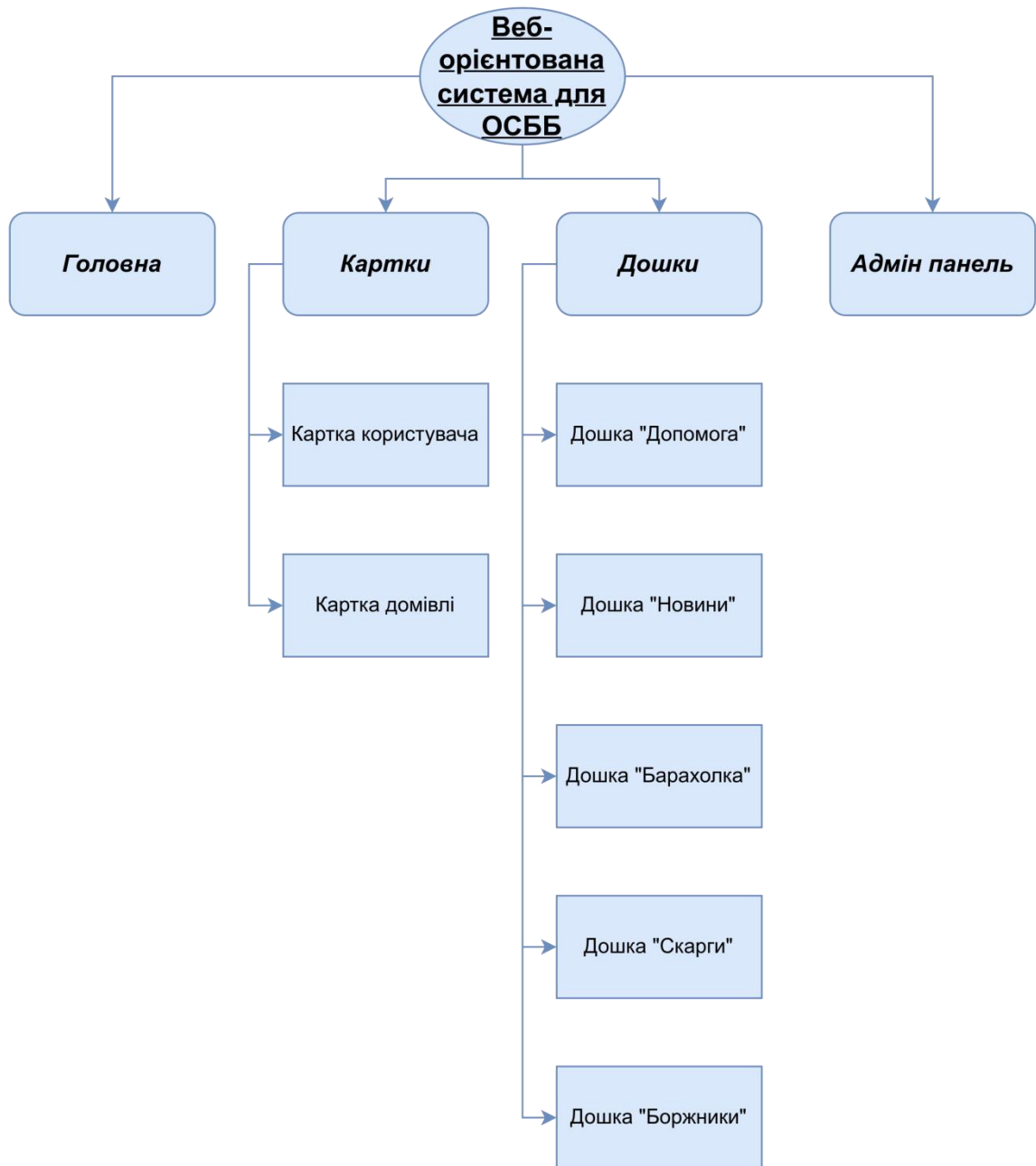


Рисунок 1.6. Майбутня навігація по веб-системі

### 1.2.2 Проєктування дизайну веб-системи

Розглянемо філософію графічного інтерфейсу кінцевої веб-системи. Графічний інтерфейс представлено у мінімалістичному стилі.

Мінімалізм являє собою дизайнерський підхід або стиль, що використовує мінімальну кількість елементів та деталей для досягнення максимального виразності та функціональності.

Користувальницький досвід кінцевої веб-системи представлено у вигляді навігаційної панелі у вигляді сайдбару та контентної частини.

Аспекти UI/ UX.

Сайдбар: бічний блок на веб-сторінці, який зазвичай містить навігаційні посилання, важливі віджети або іншу додаткову інформацію, яка не є основним контентом сторінки.

Контент: основна частина веб-сторінки, яка містить основний інформаційний або візуальний матеріал, такий як текст, зображення, відео, таблиці тощо. Це та частина сторінки, яка привертає увагу користувача та надає йому основну інформацію або функціональність [4].

На рис. 1.7 зображено дизайн навігаційного меню веб-системи адаптований під десктоп.

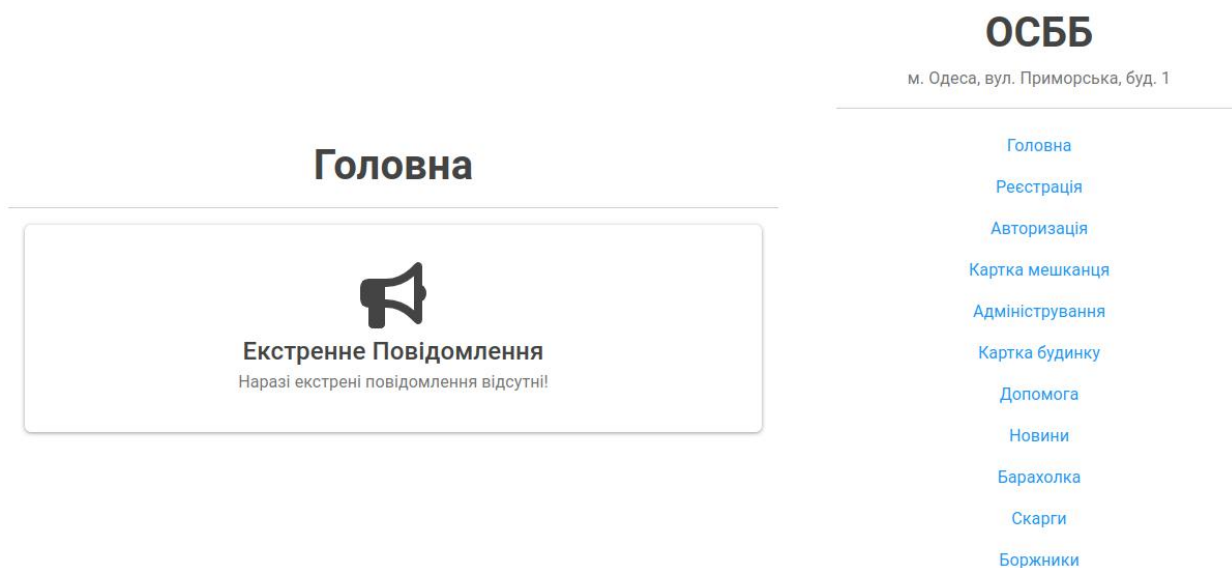


Рисунок 1.7. Дизайн навігаційного меню веб-системи під десктоп

На рис. 1.8 зображено дизайн навігаційного меню веб-системи адаптований під смартфон.

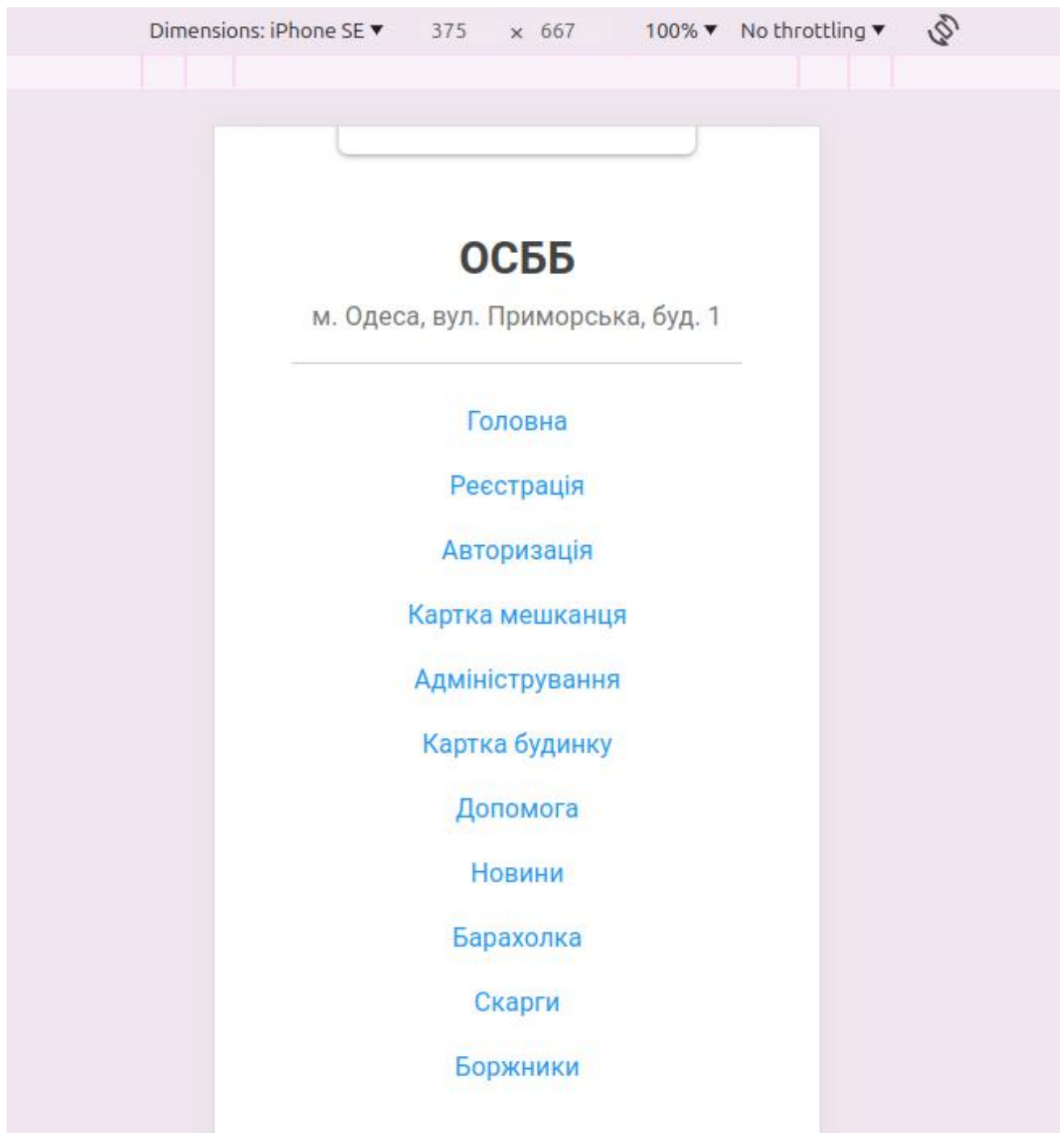


Рисунок 1.8. Дизайн навігаційного меню веб-системи під смартфон

### 1.2.3 Проєктування бази даних веб-системи

Розглянемо підхід до проєктування бази даних. В даному проєкті будуть використовуватись технології регуляційних баз даних. Реляційна база даних (РБД) являє собою бази даних, що ґрунтується на моделі даних, яка використовує таблиці для збереження та організації даних у вигляді рядків і стовпців.

					<i>РП 07. 07 001. 00 ДП ПЗ</i>	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		23

Аспекти реляційних баз даних включають:

1. Ідентифікатори: Кожен запис у таблиці має унікальний ідентифікатор, який дозволяє однозначно ідентифікувати цей запис. Ідентифікатор може бути одним або кількома полями, які утворюють унікальний ключ.
2. Зв'язки: Реляційні бази даних можуть встановлювати зв'язки між таблицями, що дозволяє створювати складні структури даних. Зв'язки визначаються за допомогою зовнішніх ключів, які вказують на зв'язані записи у інших таблицях, і дозволяють здійснювати операції зв'язування та звертання даних між таблицями [11].

На рис. 1.9 зображено схему бази даних кінцевої веб-системи.

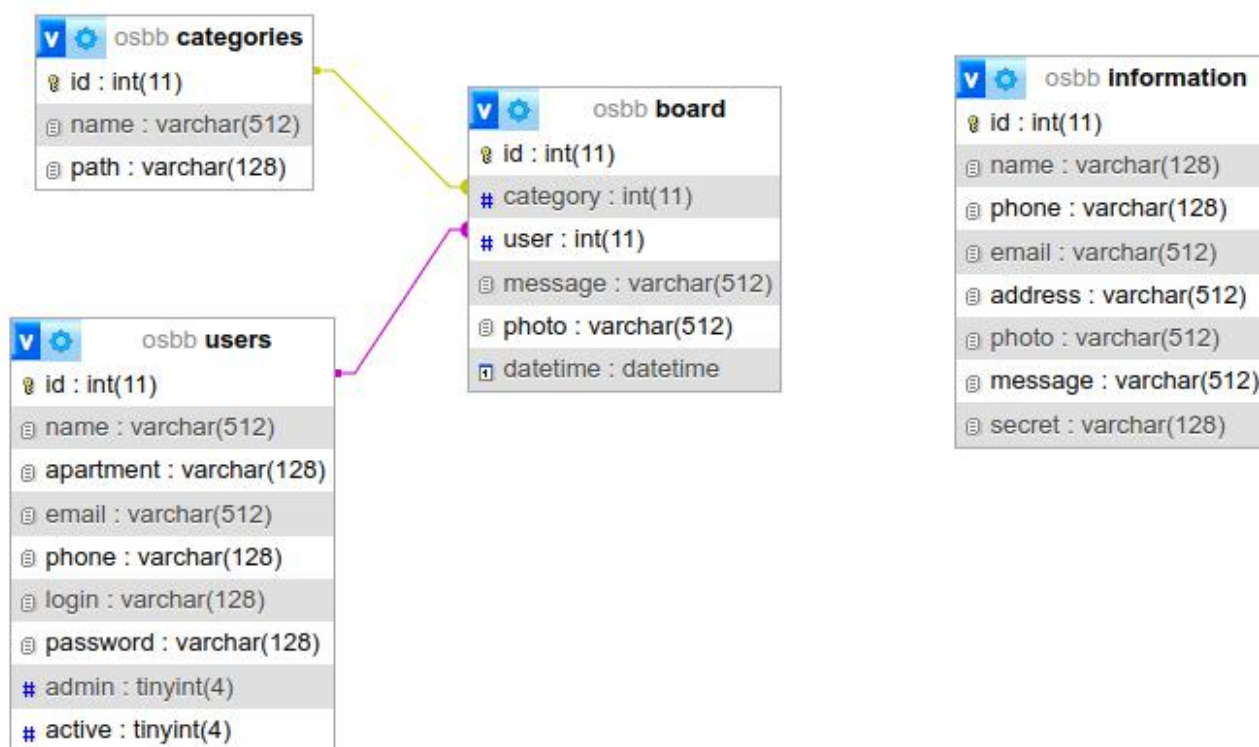


Рисунок 1.9. Схема бази даних

Розглянемо таблиці кінцевої веб-системи.

Таблиця “categories” зберігає інформацію категорії дошок.

У табл. 1.3 зображено опис таблиці “categories”.

Таблиця 1.3. Таблиця “categories”

Назва		
categories		
Поле	Тип	Опис
id	INT	Ідентифікатор категорії
name	VARCHAR	Назва категорії

Таблиця “users” зберігає інформацію про мешканців.

У табл. 1.4 зображено опис таблиці "users".

Таблиця 1.4. Таблиця "users"

Назва		
users		
Поле	Тип	Опис
id	INT	Ідентифікатор користувача
name	VARCHAR	ПІБ
apartment	VARCHAR	Номер квартири
email	VARCHAR	Email користувача
phone	VARCHAR	Телефон користувача
login	VARCHAR	Логін користувача
password	VARCHAR	Хеш пароля користувача
admin	TINYINT	Прапорець адміністратора
active	TINYINT	Статус активності облікового запису користувача

Таблиця “board” зберігає інформацію про повідомлення у дошках за категоріями.

У табл. 1.5 зображено опис таблиці “board”.

					<b>РП 07. 07 001. 00 ДП ПЗ</b>	Арк.
						25
Ізм.	Лист	№ докум.	Підпис	Дата		

Таблиця 1.5. Таблиця “board”

Назва		
board		
Поле	Тип	Опис
id	INT	Ідентифікатор повідомлення
category	INT	Ідентифікатор категорії
user	INT	Ідентифікатор користувача
message	VARCHAR	Текст повідомлення
photo	VARCHAR	Шлях до зображення повідомлення
datetime	DATETIME	Дата та час створення повідомлення

Таблиця “information” зберігає загальну інформацію про ОСББ.

У табл. 1.6 зображено опис таблиці “information”.

Таблиця 1.6. Таблиця “information”

Назва		
information		
Поле	Тип	Коментар
id	INT	Ідентифікатор запису.
name	VARCHAR	Назва закладу.
phone	VARCHAR	Телефон закладу.
email	VARCHAR	Email закладу.
address	VARCHAR	Адреса закладу.
message	VARCHAR	Повідомлення від закладу.

На рис. 1.12 зображено створені таблиці у phpMyAdmin.

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> board	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	48.0 KiB	-
<input type="checkbox"/> categories	★ Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> information	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> users	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	16.0 KiB	-
4 tables	Sum	7	InnoDB	utf8mb4_general_ci	96.0 KiB	0 B

Рисунок 1.10. Створені таблиці в phpMyAdmin

## 1.2.4 Проктування архітектури веб-системи

Ця веб-система будується на основі монолітної архітектури, де клієнтська і серверна частини працюють як єдине ціле, а не як окремі компоненти. Такий підхід дозволяє застосовувати об'єктно-орієнтоване програмування (ООП) та впроваджувати шаблон проєктування "Модель-Представлення-Контролер" (MVC).

Аспекти MVC:

1. **Модель:** Відповідає за взаємодію з базою даних, обробляючи SQL-запити та виконуючи CRUD-операції. Вона є посередником між даними та іншими компонентами системи.
2. **Представлення:** Відповідає за візуальну складову інтерфейсу, реалізуючи її за допомогою HTML, CSS та JavaScript. Це те, що бачить і з чим взаємодіє користувач.
3. **Контролер:** Забезпечує інтеракцію між моделями та представленнями. Він обробляє запити від користувача, використовує HTTP-методи GET і POST для передачі даних та визначає, які моделі і представлення слід використовувати для обробки цих запитів[14].

Завдяки монолітній структурі веб-застосунок реалізує мультисторінковий підхід, де кожен розділ сайту представлений окремою сторінкою, що призводить до перезавантаження сторінок під час навігації. Ця архітектурна модель дозволяє зручно організувати логіку застосунку та забезпечує ефективне управління даними та інтерфейсом.

На рис. 1.13 зображено схему архітектуру веб-системи.

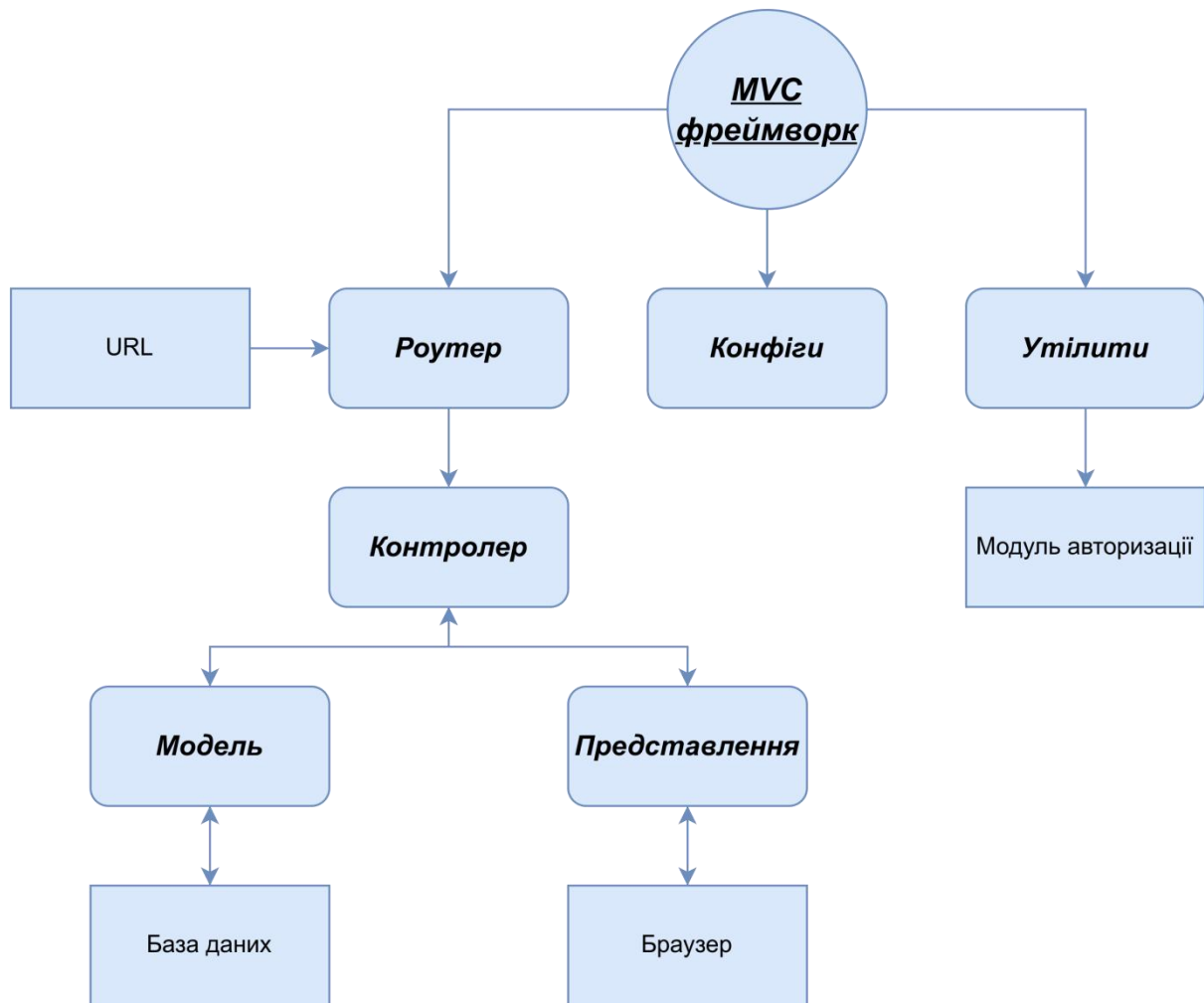


Рисунок 1.11. Архітектура веб-системи

Для реалізації даної веб-системи використовується мікро-фреймворк PHP з наступною файловою структурою.

Файлова архітектура PHP-MVC-мікро-фреймворку зображена на рис. 1.12.

Ізм.	Лист	№ докум.	Підпис	Дата

РП 07. 07 001. 00 ДП ПЗ

Арк.

28

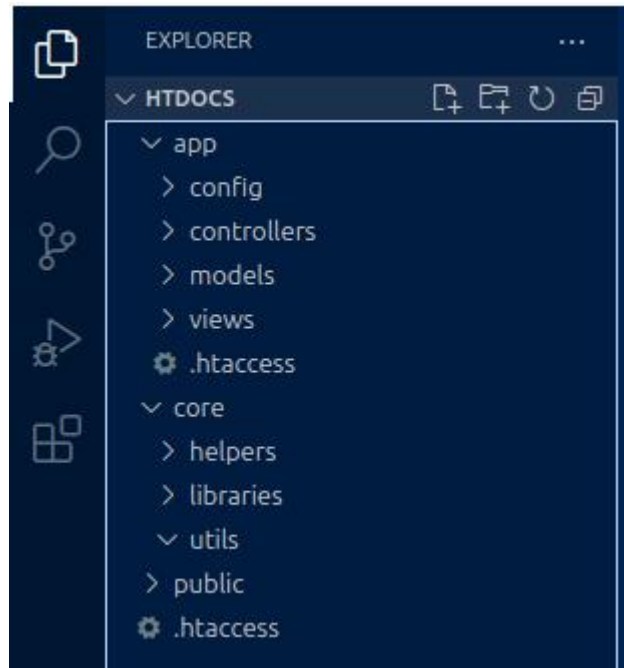


Рисунок 1.12. Файлова архітектура PHP-MVC-мікро-фреймворку

Розглянемо кожну теку більше детально.

1. /core: Ця тека містить основні компоненти фреймворку. Вона включає різноманітні допоміжні класи, функції та бібліотеки, що забезпечують роботу фреймворку.
2. /core/helpers: Ця тека зберігає утилітарні функції або класи, призначені для полегшення процесу розробки та надання додаткових можливостей.
3. /core/utils: Ця тека включає допоміжні інструменти або класи, що надають корисний функціонал, який можна використовувати в різних частинах фреймворку.
4. /core/libraries: Ця тека розташовує зовнішні бібліотеки або компоненти, які використовуються у веб-застосунку.
5. /app: Ця тека містить основну логіку веб-застосунку. Вона включає всі моделі, контролери та представлення, необхідні для роботи програми.
6. /app/config: Ця тека зберігає файли конфігурації веб-застосунку. Тут розміщені налаштування маршрутизації, баз даних та інші параметри конфігурації.

7. /app/views: Ця тека містить файли представлення (View) для шаблонів MVC. Вони зазвичай включають HTML, CSS та JavaScript файли, які відповідають за відображення даних користувачам.
8. /app/models: Ця тека містить моделі (Model) в рамках патерну MVC, які відповідають за взаємодію з базою даних та обробку даних. Тут зберігаються файли, що представляють різні об'єкти даних та виконують логіку доступу до даних.
9. /app/controllers: Ця тека зберігає контролери (Controller) в патерні MVC. Вони обробляють запити користувачів, викликають відповідні методи моделей та забезпечують відображення результатів через представлення.
- 10./public: Ця тека містить ресурси, доступні публічно через веб-сервер. Це можуть бути файли CSS, JavaScript, зображення та інші статичні ресурси.
- 11./public/uploads: Ця тека зберігає файли, які завантажують користувачі на сервер.
- 12./public/assets: Ця тека розташовує статичні ресурси, такі як стилі, скрипти та інші файли, які використовуються у клієнтському коді.
- 13./public/db: Ця тека може містити файли резервних копій бази даних або інші файли, пов'язані з базою даних.

Ця структура тек забезпечує логічне розділення різних аспектів розробки, що робить процес створення та підтримки веб-застосунків більш організованим та ефективним.

## 1.3 Реалізація веб-системи

### 1.3.1 Запуск серверу

Для розробки та запуску проєкту використовується XAMPP.

XAMPP являє собою безкоштовний популярний локальний сервер для розробки, який включає в себе кілька ключових компонентів:

MySQL Database: Система управління базами даних, що дозволяє зберігати, обробляти та керувати даними для веб-застосунків.

					<b>РП 07. 07 001. 00 ДП ПЗ</b>	Арк.
						30
Ізм.	Лист	№ докум.	Підпис	Дата		

ProFTPD: FTP-сервер, який забезпечує безпечну та ефективну передачу файлів між комп'ютером розробника та сервером.

Apache Web Server: Веб-сервер, який обробляє HTTP-запити та доставляє веб-сторінки користувачам, підтримуючи різноманітні модулі та налаштування.

На рис. 1.13 зображено запуск серверу через XAMP.

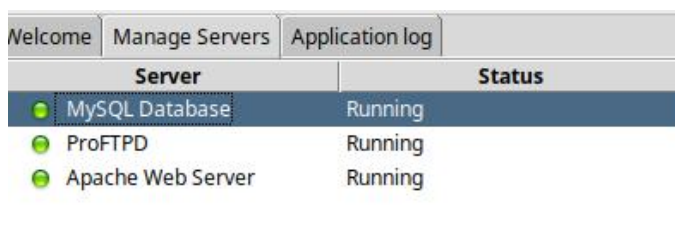


Рисунок 1.13. Запущений сервер XAMP

### 1.3.2 Ядро MVC-застосунку

Розглянемо ядро у веб-застосунку.

Файл `index.php` у PHP MVC фреймворку слугує головною точкою входу для всіх запитів, ініціалізуючи базові налаштування, завантажуючи необхідні компоненти та маршрутизуючи запити до відповідних контролерів для обробки.

Ничже зображено код файлу `public/index.php`.

```
<?php
// Check errors
error_reporting(E_ALL);
ini_set('display_errors', 1);

// Load config
require_once($_SERVER['DOCUMENT_ROOT'] . '/app/config/config.php');

// Load helpers
require_once($_SERVER['DOCUMENT_ROOT'] . '/core/helpers/session_helper.php');
require_once($_SERVER['DOCUMENT_ROOT'] . '/core/helpers/url_helper.php');

// Load core
require_once($_SERVER['DOCUMENT_ROOT'] . '/core/libraries/Autoload.php');

// Instance router
$route = new Router();
$route->run();
```

Цей код є основним файлом у PHP застосунку, який спочатку включає перевірку помилок, виводячи всі помилки для налагодження. Потім завантажуються конфігураційні файли та допоміжні функції для сесій і URL-

адрес. Нарешті, завантажується автозавантажувач класів і створюється екземпляр маршрутизатора, який запускає обробку запитів.

### 1.3.3 Кредити БД

Розглянемо конфігурацію БД у веб-застосунку.

Файл `config.php` являє собою файл, який містить налаштування конфігурації для веб-застосунку, такі як параметри бази даних, налаштування маршрутизації та інші ключові параметри.

Нижче зображено код файлу `/app/config/config.php`.

```
define('DB_HOST', '127.0.0.1');
define('DB_USER', 'root');
define('DB_PASS', '');
define('DB_NAME', '');
```

Цей код визначає константи для налаштувань підключення до бази даних, включаючи хост, користувача, пароль і ім'я бази даних.

### 1.3.4 Роутинг MVC-застосунку

Розглянемо механізм роутингу у веб-застосунку.

Механізм роутингу у веб-системах визначає, який код буде виконуватися відповідно до URL-адреси запиту, що надсилається серверу користувачем.

Нижче зображено код файлу `app/config/routes.php`.

```
<?php
// Routes array
return array(
    '' => 'home/index',
    'home' => 'home/index',
    'information' => 'information/index',
    'login' => 'login/index',
    'login/signUp' => 'login/signUp',
    'login/signIn' => 'login/signIn',
    'login/signOut' => 'login/signOut',
    'board/createMessage' => 'board/createMessage',
    'help/([0-9]+)' => 'board/index/1/$1',
    'news/([0-9]+)' => 'board/index/2/$1',
    'sells/([0-9]+)' => 'board/index/3/$1',
    'reports/([0-9]+)' => 'board/index/4/$1',
    'debtors/([0-9]+)' => 'board/index/5/$1',
    'help' => 'board/index/1/1',
    'news' => 'board/index/2/1',
    'sells' => 'board/index/3/1',
    'reports' => 'board/index/4/1',
    'debtors' => 'board/index/5/1',
    'user' => 'user/index',
    'admin' => 'admin/index',
```

Ізм.	Лист	№ докум.	Підпис	Дата

```
'admin/updateInformation' => 'admin/updateInformation',  
'board/deleteMessage/([0-9]+)' => 'board/deleteMessage/$1'  
);
```

Цей код визначає масив маршрутів для веб-застосунку, де ключі представляють URL-шляхи, а значення вказують на відповідні контролери та дії, які повинні виконуватися. Він включає маршрути для різних частин сайту, таких як домашня сторінка, сторінки входу, створення повідомлень на дошці оголошень, а також допоміжні сторінки з параметрами. Деякі маршрути мають динамічні сегменти, що дозволяють обробляти запити з числовими параметрами, які будуть передані відповідним методам контролерів.

### 1.3.5 Моделі (Models) MVC-застосунку

Розглянемо моделі у веб-застосунку.

Модель у шаблоні проектування MVC (Model-View-Controller) відповідає за представлення даних та бізнес-логіку програми. Вона взаємодіє з базою даних, виконує запити та маніпулює даними для їхнього подальшого відображення чи обробки. Модель допомагає відокремити логіку застосунку від його візуальної частини, що забезпечує більшу гнучкість та повторне використання коду.

На рис. 1.14 зображено створені моделі для веб-системи.

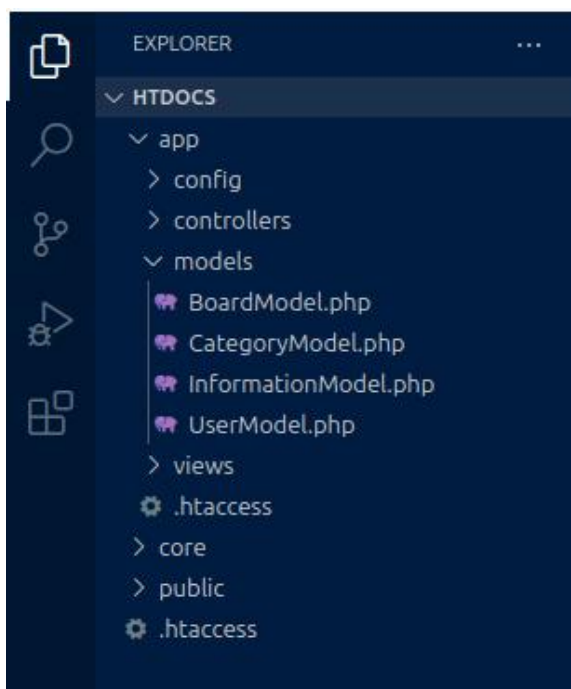


Рисунок 1.14. Створені моделі для веб-системи

Розглянемо клас моделі для таблиці “board” (дошок) у веб-застосунку.

Нижче зображено код файлу app/models/BoardModel.php.

```
<?php
class BoardModel
{
    private $db;

    public function __construct()
    {
        $this->db = new Database;
    }

    public function selectByCategoryAndPage($page, $perPage, $category)
    {
        // Calculate the offset based on the current page and items per page
        $offset = ($page - 1) * $perPage;

        // Query with pagination
        $this->db->query('SELECT board.*, users.name AS `name` FROM board INNER
JOIN users ON board.user = users.id WHERE board.category = :category LIMIT :perPage
OFFSET :offset');

        // Bind values
        $this->db->bind(':category', $category);
        $this->db->bind(':perPage', $perPage);
        $this->db->bind(':offset', $offset);

        // Execute
        $results = $this->db->resultSet();

        // Return result
        return $results;
    }

    public function total($category)
    {
        // Query
        $this->db->query('SELECT COUNT(*) as total FROM board WHERE category
= :category');

        // Bind values
        $this->db->bind(':category', $category);

        // Execute
        $result = $this->db->single()->total;

        // Return result
        return $result;
    }

    public function insert($data)
    {
        // Query
        $this->db->query('INSERT INTO board (category, user, message, photo)
VALUES(:category, :user, :message, :photo)');

        // Bind values
        $this->db->bind(':category', $data['category']);
    }
}
```

					<b>РП 07. 07 001. 00 ДП ПЗ</b>	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		34

```

        $this->db->bind(':user', $data['user']);
        $this->db->bind(':message', $data['message']);
        $this->db->bind(':photo', $data['photo']);

        // Execute
        if($this->db->execute())
        {
            return true;
        }
        else
        {
            return false;
        }
    }

    public function delete($id)
    {
        // Query
        $this->db->query('DELETE FROM board WHERE id = :id');

        // Bind values
        $this->db->bind(':id', $id);

        // Execute
        if($this->db->execute())
        {
            return true;
        }
        else
        {
            return false;
        }
    }
}
?>

```

Цей код визначає клас BoardModel, який представляє модель для дошки оголошень у веб-застосунку. У конструкторі класу встановлюється з'єднання з базою даних через об'єкт класу Database. Метод selectByCategoryAndPage виконує запит до бази даних для вибору оголошень за певною категорією та певною сторінкою з використанням пагінації. Метод total повертає загальну кількість оголошень у вказаній категорії. Метод insert додає нове оголошення до бази даних. Метод delete видаляє оголошення за його ідентифікатором з бази даних. Код використовує підготовлені запити для запобігання SQL-ін'єкціям та інших безпекових загроз.

### 1.3.6 Шаблонізатор

Розглянемо шаблонізатор у веб-застосунку.

					<i>РП 07. 07 001. 00 ДП ПЗ</i>	Арк.
						35
Ізм.	Лист	№ докум.	Підпис	Дата		

Шаблонізатор відповідає за відображення структурних елементів сторінки, таких як навігаційна панель (навбар), верхній блок (хедер), або нижній блок (футер). Він інтегрується з основною логікою застосунку, обробляє дані та відтворює їх у відповідному форматі для подальшого відображення користувачеві. Шаблонізатор допомагає відокремити представлення від бізнес-логіки, що сприяє більшій гнучкості та повторному використанню коду.

На рис. 1.15 зображено створені шаблони для веб-системи.

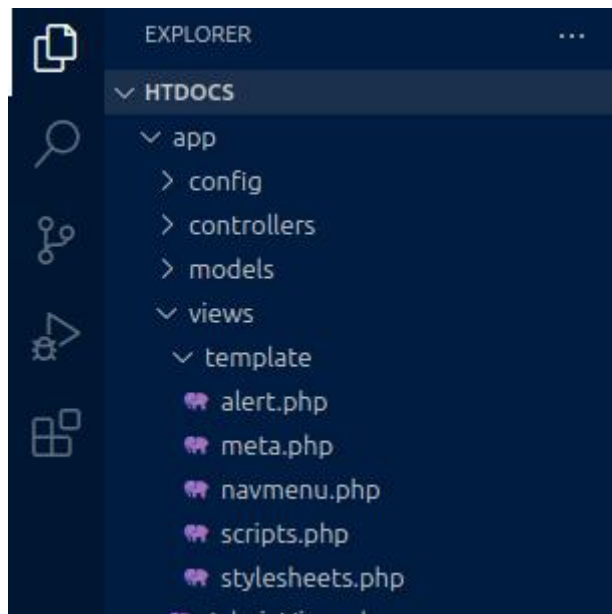


Рисунок 1.15. Створені шаблони для веб-системи

Розглянемо шаблон навігаційної панелі у веб-застосунку.

Нижче зображено код файлу app/views/template/navmenu.php.

```

<div>
  <div class="row">
    <div class="col text-center">
      <h1 class="text-center"><strong>ОСББ</strong></h1>
      <p class="text-muted"><?php echo $data['information']-
>name; ?></p>
      <hr>
    </div>
  </div>
</div>
<ul class="nav flex-column nav-fill">
  <li class="nav-item"><a class="nav-link" href="<?php echo
URLROOT; ?>/">Головна</a></li>
  <li class="nav-item"><a class="nav-link" href="<?php echo
URLROOT; ?>/login">Реєстрація</a></li>
  <li class="nav-item"><a class="nav-link" href="<?php echo
URLROOT; ?>/login">Авторизація</a></li>
  <li class="nav-item"><a class="nav-link" href="<?php echo
URLROOT; ?>/user">Картка мешканця</a></li>

```

```

        <li class="nav-item"><a class="nav-link" href="<?php
URLROOT; ?>/admin">Адміністрування</a></li>
        <li class="nav-item"><a class="nav-link" href="<?php
URLROOT; ?>/information">Картка будинку</a></li>
        <li class="nav-item"><a class="nav-link" href="<?php
URLROOT; ?>/help">Допомога</a></li>
        <li class="nav-item"><a class="nav-link" href="<?php
URLROOT; ?>/news">Новини</a></li>
        <li class="nav-item"><a class="nav-link" href="<?php
URLROOT; ?>/sells">Барахолка</a></li>
        <li class="nav-item"><a class="nav-link" href="<?php
URLROOT; ?>/reports">Скарги</a></li>
        <li class="nav-item"><a class="nav-link" href="<?php
URLROOT; ?>/debtors">Боржники</a></li>
    </ul>

```

Цей код відображає навігаційну панель сторінки веб-системи. Він складається з двох основних елементів: перша частина відповідає за відображення заголовку і інформаційного блоку про ОСББ (Організацію спільного володіння будинком), а друга частина містить список посилань на різні розділи системи, такі як "Головна", "Реєстрація", "Авторизація" тощо. Кожне посилання має відповідну URL-адресу, яка генерується змінною URLROOT. Такий код дозволяє користувачам навігувати між різними частинами веб-застосунку шляхом кліку на посилання в навігаційній панелі.

### 1.3.7 Представлення (View) у MVC-застосунку

Розглянемо представлення у веб-застосунку.

Представлення відповідають за відображення структурних елементів сторінки, таких як навігаційна панель (навбар), верхній блок (хедер), або нижній блок (футер). Вони інтегруються з основною логікою застосунку, отримують дані та відтворюють їх у відповідному форматі для подальшого відображення користувачеві. Представлення допомагають відокремити представлення від бізнес-логіки, що сприяє більшій гнучкості та повторному використанню коду.

На рис. 1.16 зображено створені представлення для веб-системи.

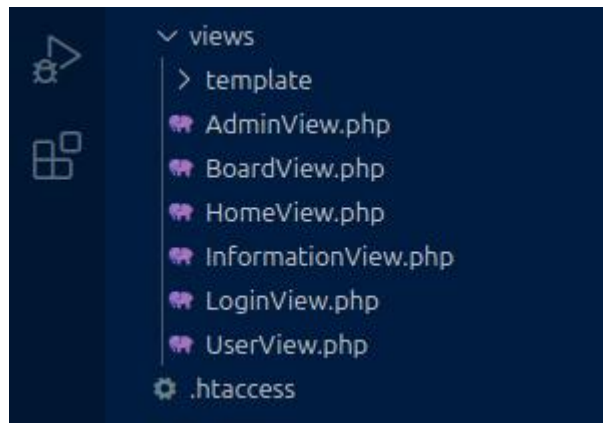


Рисунок 1.16. Створені представлення для веб-системи

Розглянемо представлення для відображення дошок у веб-застосунку.

Нижче зображено код файлу app/views/BoardView.php.

```

<!DOCTYPE html>
<html data-bs-theme="light" lang="en">

<head>
  <?php require_once APPROOT . '/views/template/meta.php'; ?>
  <?php require_once APPROOT . '/views/template/stylesheets.php'; ?>
</head>

<body class="bg-white">
  <?php require_once APPROOT . '/views/template/scripts.php'; ?>

  <section class="d-xxl-flex m-auto justify-content-xxl-center align-items-xxl-
center">
    <div class="container p-5">
      <div class="row justify-content-center align-items-center">
        <div class="col text-center m-3">
          <section>
            <h1><?php echo $data['category']->name; ?></h1>
            <hr>
            <form class="m-3" method="post" enctype="multipart/form-
data">
              <input type="hidden" name="category" value="<?php echo
$data['category']->id; ?>">
              <textarea class="form-control mt-3 mb-3"
name="message" placeholder="Повідомлення" required=""></textarea>
              <input class="form-control mt-3 mb-3" type="file"
name="photo">
              <button class="btn btn-primary w-100 mt-3 mb-3"
type="submit" formaction="<?php
URLROOT; ?>/board/createMessage">Відправити</button>
            </form>
            <?php if (empty($data['board'])) : ?>
              <?php require_once APPROOT .
'/views/template/alert.php'; ?>
            <?php else : ?>
              <div class="row row-cols-1">
                <?php foreach ($data['board'] as $item) : ?>
                  <div class="col">
                    <div class="card m-3">

```

```

        <div class="card-body">
        <div class="row row-cols-2 justify-
content-center align-items-center">
        <?php if ($item->photo) : ?>
        <div class="col">
            
            </div>
        <?php endif; ?>
        <div class="col">
            <h4 class="m-1"><?php echo
$item->user; ?></h4>
            <p class="m-1"><?php echo
$item->message; ?></p>
            </div>
        </div>
        <?php if (isLoggedInAdmin()) : ?>
        <a class="btn btn-danger w-100 m-1"
href="<?php echo URLROOT; ?>/board/deleteMessage/<?php echo $item->id; ?>">
            Видалити
        </a>
        <?php endif; ?>
        </div>
    </div>
    <?php endforeach; ?>
</div>
<!-- Pagination -->
<div class="pagination">
    <?php for ($p = 1; $p <= $data['totalPages'];
$p++) : ?>
        <a href="<?php echo URLROOT . '/' .
$data['category']->path . '/' . $p; ?>" class="page-link"><?php echo $p; ?></a>
    <?php endfor; ?>
</div>
<?php endif; ?>
</section>
</div>
<div class="col-md-4 m-3">
    <?php require_once APPROOT . '/views/template/navmenu.php'; ?>
</div>
</div>
</div>
</section>
</body>
</html>

```

Цей код відображає сторінку для відображення повідомлень у веб-застосунку. Він містить форму для відправлення повідомлень, список вже існуючих повідомлень з можливістю видалення для адміністраторів, а також навігаційне меню. Кожне повідомлення включає інформацію про користувача, текст повідомлення та, якщо наявне, прикріплене зображення. Якщо користувач не авторизований або відсутні повідомлення, відображається відповідне

					<b>РП 07. 07 001. 00 ДП ПЗ</b>	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		39

повідомлення. Навігаційне меню містить посилання на різні частини веб-застосунку.

### 1.3.8 Контролери (Controllers) у MVC-застосунку

Розглянемо використання контролерів у веб-застосунку.

Контролери відповідають за обробку запитів користувача та взаємодію з моделями даних. Вони приймають вхідні дані від користувача, обробляють їх згідно з бізнес-логікою застосунку та відправляють відповіді або рендерять відповідні представлення.

Контролери інтегруються з основною структурою застосунку, отримують дані з вхідних запитів та передають їх моделям для обробки. Вони також можуть виконувати авторизацію, перевірку доступу та інші аспекти бізнес-логіки. Використання контролерів допомагає відокремити логіку обробки запитів від представлення, що робить код більш гнучким та легким для розширення.

На рис. 1.17 зображено створені контролери для веб-системі.

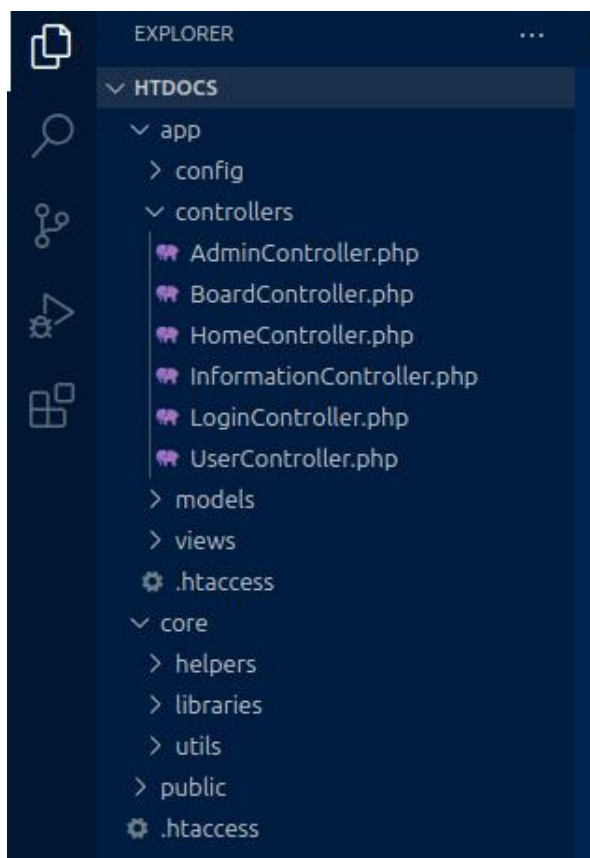


Рисунок 1.17. Створені контролери для веб-системі

Розглянемо клас контролеру для відображення дошок у веб-застосунку.

Нижче зображено код файлу app/controllers/BoardController.php.

```
<?php
class BoardController extends Controller {
    private $boardModel;
    private $categoryModel;

    public function __construct() {
        if(!isLoggedInUser()) {
            redirect('login');
        }
        $this->boardModel = $this->model('Board');
        $this->categoryModel = $this->model('Category');
    }

    public function index($id, $page = 1) {
        $category = $this->categoryModel->selectById($id);
        $perPage = 2; // Define $perPage before using it
        $board = $this->boardModel->selectByCategoryAndPage($page, $perPage,
$category->id);
        $total = $this->boardModel->total($category->id);

        // Calculate the total number of pages
        $totalPages = ceil($total / $perPage);

        $data = [
            'board' => $board,
            'category' => $category,
            'currentPage' => $page,
            'perPage' => $perPage,
            'totalPages' => $totalPages
        ];

        $this->view('Board', $data);

        flash('message');
    }

    public function createMessage() {
        try {
            // Check if the user is an admin and if it's a POST request
            if ($_SERVER['REQUEST_METHOD'] == 'POST') {
                // Sanitize input data
                $_POST = filter_input_array(INPUT_POST,
FILTER_SANITIZE_FULL_SPECIAL_CHARS);

                // Check if photo is uploaded
                if (isset($_FILES['photo']) && $_FILES['photo']['error'] ==
UPLOAD_ERR_OK) {
                    $uploads_dir = 'uploads/';
                    $file = uniqid() . '.' . pathinfo($_FILES["photo"]["name"],
PATHINFO_EXTENSION);
                    move_uploaded_file($_FILES['photo']['tmp_name'],
$uploads_dir.$file);
                } else {
                    // Handle case where photo is not uploaded
                    $file = ''; // Set photo field to empty string if no photo
                    uploaded
                }
            }
        }
    }
}
```

					<i>РП 07. 07 001. 00 ДП ПЗ</i>	Арк.
						41
Ізм.	Лист	№ докум.	Підпис	Дата		

```

    }

    // Prepare data for insertion
    $data = [
        'category' => trim($_POST['category']),
        'user' => $_SESSION['id'], // Assuming user_id is stored in
session
        'message' => trim($_POST['message']),
        'photo' => $file, // Uploaded file name
    ];

    // Call the insert method of the board model to add the message
    if ($this->boardModel->insert($data)) {
        // Message added successfully
        flash('message', 'Операція успішна! ');
        redirectBack();
    } else {
        // Failed to add message
        flash('message', 'Операція не успішна! ');
        redirectBack();
    }
} catch (Exception $e) {
    // Handle exceptions
    flash('message', 'Виникла помилка: ' . $e->getMessage());
    redirectBack();
}
}
public function deleteMessage($id) {
    try {
        if ($_SESSION['admin'] == true) {
            if ($this->boardModel->delete($id)) {
                flash('message', 'Операція успішна!');
                redirectBack();
            }
        }
    } catch (Exception $e) {
        flash('message', 'Операція не успішна!');
        redirectBack();
    }
}
}
?>

```

Цей код представляє клас BoardController, який відповідає за обробку запитів, пов'язаних з дошками веб-застосунка. У конструкторі відбувається перевірка наявності входу користувача, і якщо користувач не увійшов, відбувається перенаправлення на сторінку входу. Контролер також ініціалізує об'єкти моделей дошок і категорій. Метод index приймає параметри \$id та \$page, де відбувається вибірка дошок за категорією та сторінкою, обчислення загальної кількості сторінок та передача даних у відповідне представлення. Метод createMessage призначений для додавання повідомлень на дошку. Він перевіряє,

					<i>РП 07. 07 001. 00 ДП ПЗ</i>	Арк.
						42
Ізм.	Лист	№ докум.	Підпис	Дата		

чи користувач відправляє POST-запит, санітизує вхідні дані, завантажує фотографію, якщо така є, і додає повідомлення до бази даних через модель дошки. Метод deleteMessage видаляє повідомлення за заданим ідентифікатором, проте ця операція доступна тільки адміністраторам, перевірка яких відбувається перед видаленням. У випадку виникнення помилки виникає відповідне повідомлення про помилку, а користувач перенаправляється на попередню сторінку.

Розглянемо клас контролеру для системи реєстрації та авторизації.

Нижче зображено код файлу app/controllers/LoginController.php.

```
<?php
class LoginController extends Controller {
    private $userModel;
    private $informationModel;

    public function __construct() {
        if(isLoggedInAdmin()) {
            redirect('admin');
        }
        else if (isLoggedInUser()) {
            redirect('user');
        }

        $this->userModel = $this->model('User');
        $this->informationModel = $this->model('Information');
    }

    public function index() {
        $information = $this->informationModel->select();

        $data = [
            'information' => $information,
        ];

        $this->view('Login', $data);

        flash('message');
    }

    public function signIn() {
        // Check for POST
        if ($_SERVER['REQUEST_METHOD'] == 'POST') {
            // Process form

            // Sanitize POST data
            $_POST = filter_input_array(INPUT_POST,
FILTER_SANITIZE_FULL_SPECIAL_CHARS);

            // Init data
            $data = [
                'login' => trim($_POST['login']),
                'password' => trim($_POST['password'])
            ]
        }
    }
}
```

					<i>РП 07. 07 001. 00 ДП ПЗ</i>	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		43

```

];

    $loggedInUser      =      $this->userModel->select($data['login'],
$data['password'], 1);

    // Validate user
    if($loggedInUser) {
        setSession($loggedInUser);
        redirect('login');
    }
    else
    {
        flash('message', 'Операція неуспішна! Можливо, не вірні дані.
Спробуйте інші дані. ');
        redirect('login');
    }
}

}

public function signUp() {
    // Check for POST
    if ($_SERVER['REQUEST_METHOD'] == 'POST') {
        // Process form
        // Sanitize POST data
        $_POST          =          filter_input_array(INPUT_POST,
FILTER_SANITIZE_FULL_SPECIAL_CHARS);

        // Check if the provided secret matches the secret in the information
model
        $secret = trim($_POST['secret']);

        $info = $this->informationModel->select();

        if ($info && $info->secret != $secret)
        {
            flash('message', 'Неправильний секрет! Перевірте і спробуйте ще
раз. ');
            redirect('login');
        }

        // Init data
        $data = [
            'name' => trim($_POST['name']),
            'apartment' => trim($_POST['apartment']),
            'email' => trim($_POST['email']),
            'phone' => trim($_POST['phone']),
            'login' => trim($_POST['login']),
            'password' => trim($_POST['password'])
        ];

        // Hash password
        $data['password'] = password_hash($data['password'], PASSWORD_DEFAULT);

        // Validate action
        if($this->userModel->insert($data)) {
            flash('message', 'Операція успішна! ');
            redirect('login');
        }
        else {
            flash('message', 'Операція неуспішна! Можливо, не вірні дані.
Спробуйте інші дані. ');

```

					<i>РП 07. 07 001. 00 ДП ПЗ</i>	Арк.
						44
Ізм.	Лист	№ докум.	Підпис	Дата		

```

        redirect('login');
    }
}

public function signOut() {
    unsetSession();

    redirect('home');
}
}
?>

```

Цей код відповідає за обробку запитів, пов'язаних із входом користувачів у веб-застосунок. У конструкторі перевіряється, чи увійшов користувач з правами адміністратора чи звичайний користувач, та відбувається перенаправлення на відповідні сторінки. Контролер ініціалізує об'єкти моделей користувачів та інформації. Метод `index` відображає сторінку входу. Метод `signIn` обробляє дані форми входу, перевіряє їх правильність та встановлює сесію за допомогою функції `setSession($loggedInUser)`. Метод `signUp` обробляє дані форми реєстрації, перевіряє правильність введеного секрету та зберігає нового користувача у базі даних. Після успішної реєстрації також встановлюється сесія користувача. Метод `signOut` видаляє сесію користувача за допомогою функції `unsetSession()` та перенаправляє його на головну сторінку.

## 1.4 Мануальне тестування (Manual QA)

Проведемо мануальне тестування.

Ручне тестування являє процес випробування програмного забезпечення людиною без застосування автоматизованих інструментів. Цей етап важливий у життєвому циклі програмного продукту, оскільки дозволяє виявити помилки, які можуть залишитися непоміченими автоматизованими тестами, а також оцінити зручність користування з точки зору реального користувача.

Ручне тестування основних веб-сторінок розробленої веб-орієнтованої системи буде проведено в подальших розділах. Це допоможе виявити можливі недоліки та забезпечити високу якість кінцевого продукту.

					<b>РП 07. 07 001. 00 ДП ПЗ</b>	Арк.
						45
Ізм.	Лист	№ докум.	Підпис	Дата		

### 1.4.1 Огляд головної сторінки

Розглянемо головну веб-сторінку у веб-системі.

Головна веб-сторінка може бути корисна для швидкого доступу до важливих повідомлень та оновлень.

На рис. 1.18 зображено головну сторінку веб-системи.

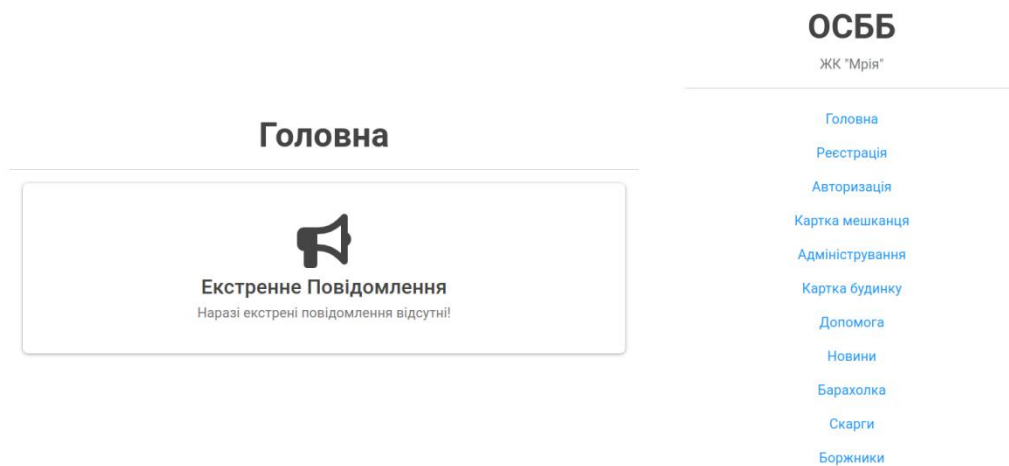


Рисунок 1.18. Головна веб-сторінка у веб-системі

### 1.4.2 Огляд сторінки реєстрації та авторизації

Розглянемо веб-сторінку реєстрації та авторизації у веб-системі.

Реєстрація дозволяє користувачам створити обліковий запис у веб-системі, що забезпечує їм доступ до персоналізованих функцій та можливість зберігати дані.

Авторизація впевнюється, що лише авторизовані користувачі отримують доступ до конфіденційної інформації та функціоналу, зберігаючи безпеку та конфіденційність даних.

На рис. 1.19 зображено веб-сторінку реєстрації та авторизації у веб-системі.

## Авторизація та Реєстрація

ПІБ

Квартира

Електронна пошта

Мобільний телефон

Логін

Пароль

Ключ реєстрації

ЗАРЕЄСТРУВАТИСЬ

реєстрація лише для мешканців будинку

Логін

Пароль

АВТОРИЗУВАТИСЬ

авторизація лише для мешканців будинку

## ОСББ

ЖК "Мрія"

[Головна](#)

[Реєстрація](#)

[Авторизація](#)

[Картка мешканця](#)

[Адміністрування](#)

[Картка будинку](#)

[Допомога](#)

[Новини](#)

[Барахолка](#)

[Скарги](#)

[Боржники](#)

Рисунок 1.19. Сторінка реєстрації та авторизації у веб-системі

### 1.4.3 Огляд кабінету адміністратора

Розглянемо веб-сторінку кабінету адміністратора у веб-системі.

Кабінет адміністратора для ОСББ є ключовим інструментом управління веб-системою, де адміністратор може змінювати адреси об'єктів, оновлювати фотографії для актуалізації інформації, а також додавати нові повідомлення для мешканців щодо важливих питань та подій. Крім того, через цей кабінет адміністратор може видаляти застарілі повідомлення, щоб зберегти актуальність інформації та забезпечувати чистоту спільного простору. Це дозволяє забезпечити ефективне комунікаційне середовище між учасниками ОСББ та забезпечити вчасне оновлення інформації про спільні справи та події. Власники можуть впевнено користуватися цим кабінетом для керування важливими аспектами їхнього спільного життя та співпраці з адміністрацією.

На рис. 1.20 зображено сторінку з кабінетом адміністратора у веб-системі.

					<i>РП 07. 07 001. 00 ДП ПЗ</i>	Арк.
						47
Ізм.	Лист	№ докум.	Підпис	Дата		

## Адміністрування

Ім'я
ЖК "Мрія"
Адреса
м. Одеса, вул. Приморська, буд. 1
Телефон
+380669996699
E-Mail
name@domain.com
Повідомлення
Наразі екстрені повідомлення відсутні!
Фото
Choose FileNo file chosen
Максимальний розмір: 5MB
Секрет
2024
<b>ЗБЕРЕГТИ</b>
<b>ВИХІД</b>

## ОСББ

ЖК "Мрія"

- Головна
- Реєстрація
- Авторизація
- Картка мешканця
- Адміністрування
- Картка будинку
- Допомога
- Новини
- Барахолка
- Скарги
- Боржники

Рисунок 1.20. Сторінка кабінету адміністратора у веб-системі

### 1.4.4 Огляд карток

Розглянемо картку мешканця у веб-системі.

Картка мешканця у веб-системі дозволяє переглянути особисту інформацію кожного мешканця, таку як контактні дані та дані про нерухомість, сприяючи відстеженню даних. Це забезпечує зручний доступ до важливої інформації для адміністрації та інших мешканців спільної власності. Крім того, на картці мешканця може бути функція швидкого виходу з системи, що забезпечує безпечну та зручну управління доступом до особистої інформації. Такий підхід дозволяє підвищити рівень конфіденційності та безпеки даних, що є важливим для користувачів веб-системи.

На рис. 1.21 зображено картку мешканця у веб-системі.

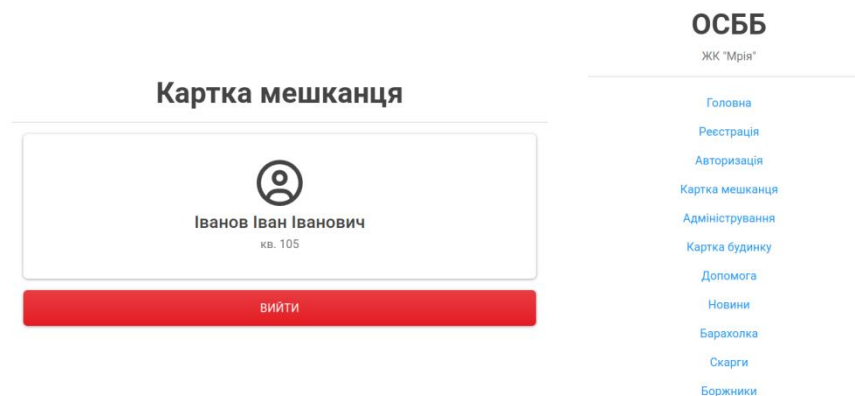


Рисунок 1.21. Сторінка кабінету користувача у веб-системі

Розглянемо картку будинку у веб-системі.

Картка будинку у веб-системі містить фотографії будинку, його назву та точну фізичну адресу, що забезпечує користувачам швидкий доступ до важливої інформації про об'єкт нерухомості. Це дозволяє мешканцям та адміністрації спостерігати за станом будинку та забезпечує чітку ідентифікацію його місце знаходження. При цьому зображення та деталі адреси допомагають у візуальному та легкому розпізнанні будинку для всіх зацікавлених сторін.

На рис. 1.22 зображено картку будинку у веб-системі.

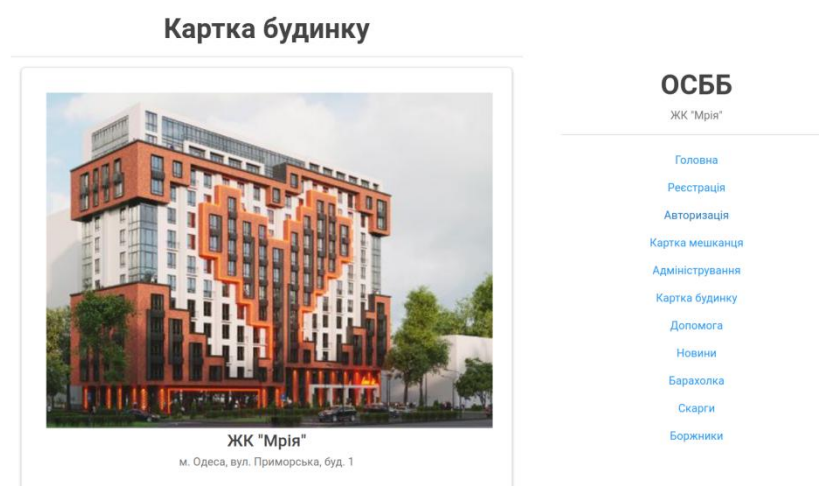


Рисунок 1.22. Сторінка кабінету користувача у веб-системі

Ізм.	Лист	№ докум.	Підпис	Дата

РП 07. 07 001. 00 ДП ПЗ

Арк.

49

### 1.4.5 Огляд сторінок з дошками повідомлень

Розглянемо сторінки з дошками повідомлень по категоріям.

Дошки повідомлень у веб-системі розподілені за категоріями, такими як “Допомога”, “Новини”, “Барахолка”, “Скарги” та “Боржники”, що дозволяє користувачам швидко знаходити та переглядати інформацію, що відповідає їхнім потребам та інтересам. Кожна категорія функціонує як окрема секція, яка полегшує організацію та навігацію по різноманітній інформації. Ця структура сприяє ефективному обміну інформацією та співпраці між учасниками спільноти або спільної власності.

Наприклад, на рис. 1.22 зображено дошку “Допомога”.

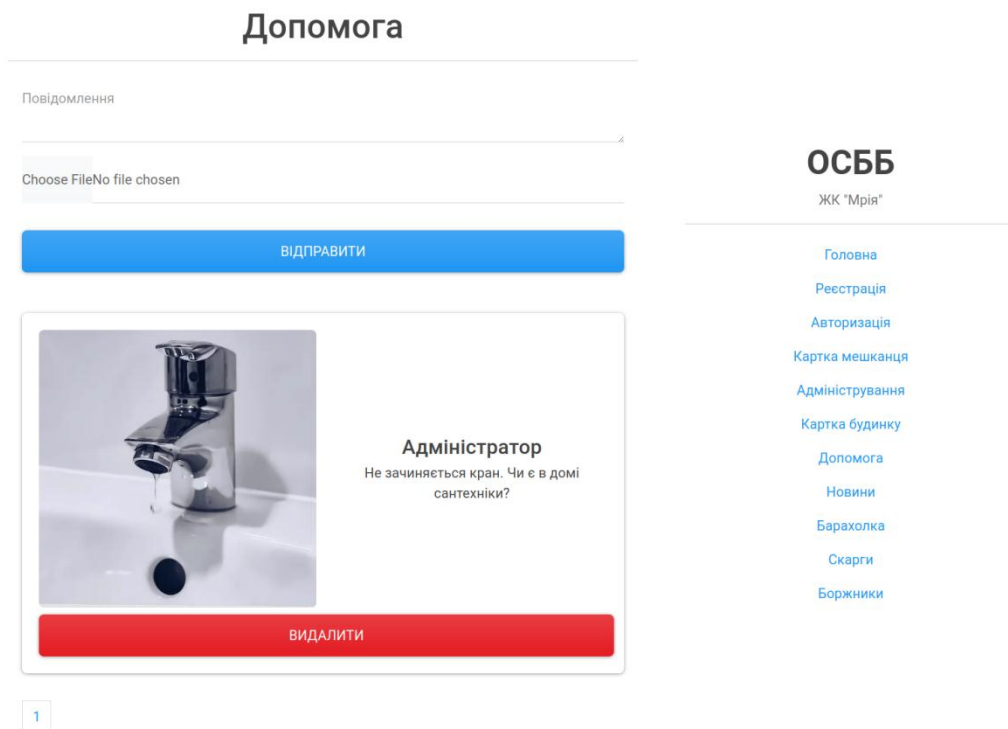


Рисунок 1.23. Дошка “Допомога” у веб-системі

Наприклад, на рис. 1.23 зображено дошку “Новини”.

На рис. 1.23 зображено сторінку з колекціями NFT по категоріям.

## Новини

Повідомлення

Choose File No file chosen

ВІДПРАВИТИ

Інформація наразі відсутня!

## ОСББ

ЖК "Мрія"

[Головна](#)

[Реєстрація](#)

[Авторизація](#)

[Картка мешканця](#)

[Адміністрування](#)

[Картка будинку](#)

[Допомога](#)

[Новини](#)

[Барахолка](#)

[Скарги](#)

[Боржники](#)

Рисунок 1.23. Дошка "Новини" у веб-систем

					<i>РП 07. 07 001. 00 ДП ПЗ</i>	Арк.
						51
Ізм.	Лист	№ докум.	Підпис	Дата		

## 2 ЕКОНОМІЧНИЙ РОЗДІЛ

### 2.1 Резюме

Темою даного дипломного проекту є розробка веб-платформи, спрямованої на автоматизацію управління ОСББ. Застосування передових програмних рішень і баз даних, адаптованих до вимог сучасного ринку, виявилось стратегічно важливим. Проведені дослідження довели, що впровадження такої системи сприятиме оптимізації процесів управління багатоквартирними будинками, надаючи конкурентні переваги та підвищуючи ефективність функціонування. Використання веб-орієнтованих інструментів дозволить оперативно реагувати на зміни у вимогах споживачів, оптимізувати робочі процеси та забезпечити високу якість обслуговування спільноти.

У створеній системі реалізовано широкий спектр функціоналу, включаючи інструменти для автоматизації обліку ресурсів, моніторингу стану будинків, а також забезпечення зв'язку між мешканцями та управителем.

Ефективність програмного продукту визначається його якістю та ефективністю процесу розробки. Якість ПП визначається наступними складовими: з точки зору користувача; з позиції використання ресурсів; виконання вимог до програмного забезпечення. Оцінка якості програмного продукту включає визначення трудомісткості і вартості його створення.

Проведемо розрахунки визначення трудомісткості розробки даного програмного продукту.

### 2.2 Визначення трудомісткості розробки програмного забезпечення

Тривалість розробки програмного продукту залежить від його обсягу, трудомісткості розробки, кваліфікації виконавців, а також планових термінів, визначених умовами ринку. Методом структурної аналогії по відповідних каталогах аналогів програмного забезпечення визначається обсяг програмних засобів, у тисячах умовних машинних команд програми аналога.

					<i>РП 07. 07 002. 00 ДП ПЗ</i>	Арк.
						52
Ізм.	Лист	№ докум.	Підпис	Дата		

Таблиця 2.1. Каталог аналогів

Найменування ПП	Обсяг функції ПП – $V_0$ , усл. машинних командах
1. ПП СУБД	2500 – 9800
2. Комплексні системи ведення БД	950 – 7430
3. ПП введення інформації	1060 – 5750

У табл. 2.1 представлені аналоги програмного забезпечення, функції яких, у більшому або меншому ступені, виконує розроблений програмний продукт. Для нашого варіанта виділено сірим кольором.

Вибравши аналог ПП, що містить  $V_0$  в умовних машинних командах, трудомісткості визначати на основі табл. 2.2.

Таблиця 2.2. Обсяг команд

Обсяг ПП, тис.умов.машинних команд	Норма часу, люд/год
1.00	229
2.00	244
3.00	262

На підставі отриманого значення, по довіднику, визначається укрупнена норма часу на розробку аналога програмного забезпечення (коректується поправочним коефіцієнтом враховуючої умови розробки ПП, тобто в умовах комп'ютера,  $K_k=0,7 \div 0,8$ ):  $T_{ар}=229 \times 0,8=183,2$  (люд/годин).

Трудомісткість програмного продукту визначається по кожному етапу розробки окремо на підставі трудомісткості аналога з урахуванням складності розробки, ступеня новизни і ступеня використання в розробці стандартних модулів на підставі формул:

$$T_{ТЗ} = T^a p \leftarrow \mathcal{L}_1 \leftarrow \mathcal{K}_H \quad (2.1)$$

$$T_{ПП} = T^a p \leftarrow \mathcal{L}_2 \leftarrow \mathcal{K}_H \quad (2.2)$$

$$T_{РП} = T^a p \leftarrow \mathcal{L}_3 \leftarrow \mathcal{K}_H \leftarrow \mathcal{K}_T \quad (2.3)$$

Для розрахунку необхідні наступні коефіцієнти:

					<b>РП 07. 07 002. 00 ДП ПЗ</b>	Арк.
						53
Ізм.	Лист	№ докум.	Підпис	Дата		

$L_i$  – питома вага  $i$ -го етапу розробки (див. табл. 2.3);

$K_n$  – поправочний коефіцієнт, що враховує ступінь новизни (див. табл. 2.4);

$K_t$  – поправочний коефіцієнт, що враховує ступінь використання в розробці типових програм (див. табл. 2.5).

Таблиця 2.3. Значення питомих коефіцієнтів трудомісткості стадії в загальній трудомісткості розробки ПП

Код стадії	Ступінь новизни		
	А	Б	В
ТЗ ( $L_1$ )	0,15	0,12	0,12
ТП ( $L_2$ )	0,16	0,15	0,11
РП ( $L_3$ )	0,55	0,58	0,61

Для нашого варіанта виділено сірим кольором.

Таблиця 2.4. Значення поправочного коефіцієнта, що враховує ступінь новизни

Код ступеня новизни	Ступінь новизни	Значення $K_n$
А	Принципово нові ПО	1,75 – 1,2
Б	ПО – розвиток визначеног параметричного ряду	1,0 – 0,8
В	ПО маючий аналог	0,7

Для нашого варіанта виділено сірим кольором.

Таблиця 2.5. Значення коефіцієнта ступеня використання в розробці типових програм

Ступінь охоплення реалізованих функцій розроблювального ПО типовими програмами, %	Значення $K_t$
60 і вище	0,6
40-60	0,7
20-40	0,8
До 20	0,9

Тепер розраховуємо трудомісткість по кожному етапу окремо:

Трудомісткість технічного завдання

$$T_{ТЗ}=T_a * L_1 * K_n = 183,2 * 0,12 * 0,7 = 15,39 \text{ (люд/годин)} \quad (2.1)$$

Трудомісткість розробки технічного проєкту

$$T_{ТП}=T_a * L_2 * K_n = 183,2 * 0,11 * 0,7 = 17,42 \text{ (люд/годин)} \quad (2.2)$$

Трудомісткість розробки робочого проєкту

$$T_{РП}=T_a * L_3 * K_n * K_T = 183,2 * 0,61 * 0,7 * 0,7 = 54,76 \text{ (люд/годин)} \quad (2.3)$$

Для подальших розрахунків визначили кількість папера, витраченого на кожен етап: технічне завдання  $N_{ТЗ}=2$  (стр), розробка ТП  $N_{ТП}=44$ (стр), розробка робочого проєкту  $N_{РП}=10$  (стр), пояснювальна записка відповідно  $N_{ПЗ}=20$  (стр).

Розрахунок зведений у табл. 2.6.

Таблиця 2.6. Розрахунок трудомісткості ПП

Найменування етапів	Розрахунок, годин.		
1.ТЗ	$T_{РТЗ}=15,39$	$T_{КК}=0,7 * N_{ТЗ}= 0,7 * 2=1,4$	$T_{НК}=0,15 * N_{ТЗ}=0,15 * 2=0,30$
2.Розробка ТП	$T_{РТП}=14,12$	$T_{КК}=0,7 * N_{ТП}=0,7 * 44=30,8$	$T_{НК}=0,15 * N_{ТП}=0,15 * 44=6,6$
3.Розробка РП	$T_{РРП}=54,76$	$T_{КК}=0,7 * N_{РП}=0,7 * 10=7,0$	$T_{НК}=0,15 * N_{РП}=0,15 * 10=1,5$
4.Розробка ПЗ	$T_{ПЗ}=1,5 * N_{ПЗ}= 1,5 * 20 =30$	$T_{КК}=0,7 * N_{ТЗ}=0,7 * 20=14$	$T_{НК}=0,15 * N_{ПЗ}=0,15 * 20 =3,0$
Усього, в т.ч.:	178,87		
- на розробку	$T_p=114,27$		
- контроль		$T_{КК}=53,2$	
- нормоконтроль			$T_{НК}=11,4$

### 2.3 Розрахунок ціни програмного продукту

У цьому розділі для визначення ціни розраховуємо основну заробітну плату виконавців, матеріальні витрати, загальні витрати на розробку ПП. Розрахунок основної заробітної плати виконавців приведений у табл. 2.7. Відповідно до статті 8 «Закону про Державний бюджет України на 2024»

встановлено мінімальну заробітну плату у місячному розмірі з 1 квітня 2024 року - 8000 гривень; мінімальну погодинну тарифну ставку – 46,00 грн.

Таблиця 2.7. Розрахунок основної заробітної плати виконавців

Найменування робіт	Трудомісткість робіт, години	Погодинна тарифна ставка, грн.	Розрахунок, грн.
1.Розробка ПП	114,27	60,00	6 856,20
2.Контроль керівника	53,20	90,00	4 788,00
3.Нормоконтроль	11,4	80,00	912,00
Усього	-	-	30= 12 556,2

Зробимо розрахунок матеріальних витрат на розробку ПП. Розрахунок зведемо в табл. 2.8.

Таблиця 2.8. Розрахунок матеріальних витрат на розробку ПО

Найменування матеріальних витрат	Тип, модель	Кількість	Ціна одиниці, грн.	Вартість, грн.
Папір	Лист А4	75	4.00	300,0
Транспортно – заготівельні Витрати (10%)				$V_{тр\_з} = 0,1 \times V_{м1} = 0,1 \times 300 = 30,0$
Усього				$V_{м} = V_{мі} + V_{тр\_з} = 330.0$

На підставі отриманих даних по окремих статтях витрат складена калькуляція планової собівартості в цілому ПП за формою, приведеною в табл. 2.9.

Таблиця 2.9. Розрахунок статей витрат планової собівартості

Стаття витрат	Значення, грн.	Формула розрахунку
1. Матеріали	330.0	$V_M$ (див. табл. 2.8)
2. Основна заробітна плата	12 556,2	$Z_o$ (див. табл. 2.7)
3. Додаткова заробітна плата	12 55,62	$3d = 0,1 \leftrightarrow Z_o = 12\ 556,2 \times 0,1$
4. Відрахування до єдиного фонду соціального внеску	3 038,60	$Вс.с.в. = 0,22 \leftrightarrow (Z_o + 3d) = 0,22 \times (12\ 556,2 + 12\ 55,62)$
5. Накладні витрати	5 022,48	$Внак. = 0,4 \leftrightarrow Z_o = 0,4 \times 12\ 556,2$
6. Повна собівартість	22 202,9	$C_{пов} = V_M + Z_o + 3d + Вс.с.в. + Внак. = 330.0 + 12\ 556,2 + 12\ 55,62 + 3\ 038,60 + 4307,77$

Розмір прибутку, що включається в ціну, визначаємо по наступній формулі:

$$П = (C_{пов} * P) / 100 = (22\ 202,9 * 10) / 100 = 22\ 20,29 \text{ грн.} \quad (2.4)$$

Де  $p$  – плановий рівень рентабельності (10-20%).

Оптова ціна (кошторисна вартість) визначається по формулі:

$$Ц_o = C_{п} + П = 22\ 202,9 + 2220,29 = 24\ 423,19 \text{ грн.} \quad (2.5)$$

Виходячи з отриманих даних, ціна реалізації розробленого програмного продукту на основі наступної формули, становитиме:

$$Ц_p = Ц_o + ПДВ = 24\ 423,19 + 24\ 423,19 * 0,2 = 29\ 307,82 \text{ грн.} \quad (2.6)$$

## **3 РОЗДІЛ ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ**

### **3.1 Вступ**

Безпека праці на підприємстві може бути на належному рівні тільки тоді, коли всебічно відповідає вимогам трудового законодавства, державним стандартам України, норм і правил, розроблених для збереження здоров'я працюючих. Важливе місце при цьому належить виконанню організаційних вимог з охорони праці, а також трудовій та виробничій дисципліні працюючих.

В розділі дипломного проекту «Охорона праці» розглядається питання розробки автоматизованої веб- системи для ОСББ. Тому для розгляду вибираємо робоче місце програміста.

### **3.2 Аналіз дії небезпечних та шкідливих чинників на працівника під час роботи**

Забезпечення безпечних і здорових умов праці в значній мірі залежить від правильної оцінки небезпечних та шкідливих виробничих факторів. Однакові по складності зміни в організмі людини можуть бути викликані різними причинами. Це можуть бути фактори виробничого середовища (підвищений рівень шуму, підвищена температура зовнішнього середовища, відсутність або недостатня освітленість робочої зони, електричний струм, статична електрика тощо), надмірне фізичне і розумове навантаження, нервово-емоційна напруга, а також різне сполучення цих причин.

### **3.3 Вимоги до виробничого середовища**

На робочому місці програміста повинні бути створені умови для безпечної та високопродуктивної праці. Безпечні умови праці досягаються за рахунок забезпечення безпеки виробничих процесів, які обґрунтовані і прийняті в технологічній частині дипломного проекту.

#### **3.3.1 Мікроклімат**

					<i>РП 07. 07 003. 00 ДП ПЗ</i>	Арк.
						58
Ізм.	Лист	№ докум.	Підпис	Дата		

На функціональному стані людини негативно позначається висока температура повітря. Хоч генерація теплоти дисплеєм досягає критичного рівня тільки у саму теплу пору року, необхідно створювати комфортні теплові умови постійно.

Оптимальні та допустимі мікрокліматичні параметри у приміщеннях повинні враховувати специфіку технологічного процесу при використанні комп'ютерів. Згідно з діючими у нашій країні нормативними документами ( ДСанПіН 3.3.2-007-98 у холодні періоди року температура повітря, швидкість його руху та відносна вологість повітря повинні відповідно складати: 22-24<sup>o</sup>C; 0,1 м/с; 40-60%. Температура повітря може коливатись у межах від 21 до 25<sup>o</sup>C при збереженні інших параметрів мікроклімату.

В теплі періоди року температура повітря, його рухливість та відносна вологість повинні відповідно становити: 23-25<sup>o</sup>C; 0,1-0,2 м/с; 40-60 %.

Оптимальним рівнем аероіонізації у зоні дихання користувача вважається вміст легких аерофонів обох знаків від 150 до 5000 у 1 см<sup>3</sup> повітря.

Вплив на склад повітря робочої зони справляють примусова вентиляція, захисні екрани ( оснащені заземленням) та застосування іонізаторів.

### 3.3.2 Шум

Джерелами цілого ряду звуків, що містять як коливання, які можна почути, так і коливання ультразвукового діапазону є деякі ВДТ. Цей шум справляє негативний вплив на стан користувача, особливо при тривалому впливі. У користувача, діяльність якого пов'язана з переробкою інформації це виражається у зниженні розумової працездатності, зростає кількість помилок, розвиток зорового стомлення, зміні відчуття кольорів, появі головного болю, послаблення уваги.

Нормованим параметром шуму на робочих місцях є рівень 50 дБ. Заходами боротьби з шумом є усунення або ослаблення причин шуму в самому його джерелі у процесі проектування, використання засобів звукопоглинання, раціональне планування виробничих приміщень.

					<b>РП 07. 07 003. 00 ДП ПЗ</b>	Арк.
						59
Ізм.	Лист	№ докум.	Підпис	Дата		

### 3.3.3 Освітлення

Освітлення у приміщеннях з ПК має бути змішаним – природним та штучним. Природне освітлення повинно здійснюватись у вигляді бічного освітлення та відповідати нормам ДБН В.2.5-28-2006 «Природне і штучне освітлення».

Наявність сонцезахисних засобів, що знижують перепади яскравостей між природним світлом та свіченням екрана ПК передбачаються при природному освітленні. Це можуть бути плівки з металізованим покриттям або жалюзі з вертикальними ламелями, що регулюються.

У вигляді комбінованої системи освітлення з використанням люмінесцентних джерел світла у світильниках загального освітлення. На робочих місцях має бути забезпечена рівномірна освітленість за допомогою переважно відбитого або розсіяного світлорозподілу. Світлових відблисків з клавіатури, екрана та від інших частин ПК у напрямку очей користувача не повинно бути.

Для штучного освітлення у приміщенні використовуються люмінесцентні лампи типу ЛБ, які в порівнянні з лампами розжарювання мають ряд істотних переваг: за спектральним складом світла вони близькі до природного світла, мають підвищену світлову віддачу (у 2-5 разів вищу, ніж у ламп розжарювання); мають триваліший термін служби – до 10 тис годин. Допускається застосування ламп розжарювання у світильниках місцевого освітлення. Норма освітленості на робочих місцях складає 300-500лк.

### 3.3.4 Електробезпека

Значення сили струму, що проходить через організм людини, залежить від напруги, під якою перебуває людина й від опору ділянки тіла, до якого прикладена ця напруга. Джерелом живлячої напруги є мережа змінного струму з напругою 229В, на яку поширюється ГОСТ 25861-83.

Для попередження поразок електричним струмом необхідно чітко й у повному обсязі виконувати правила провадження робіт і правил технічної

					<b>РП 07. 07 003. 00 ДП ПЗ</b>	Арк.
						60
Ізм.	Лист	№ докум.	Підпис	Дата		

експлуатації. Необхідно виключити можливість доступу оператора до частин устаткування, що працює під небезпечною напругою, до неізольованим частинам, призначеним для роботи при малій напрузі й не підключеним до захисного заземлення, а також підводити електроживлення до ПЕОМ від розетки за допомогою спеціальної вилки із заземлюючим контактом.

### **3.3.5 Організація робочого місця програміста**

Відповідність конструкцій всіх елементів робочого місця та їх взаємного розташування, ергономічним вимогам, з урахуванням характеру і особливостей трудової діяльності (ДСанПіН 3.3.2.-007-98) мають забезпечувати обладнання і організація робочого місця з ПК .

Конструкція робочого місця й взаємне розташування всіх його елементів (сидіння, органи керування, засобу відображення інформації) відповідають антропометричним, фізіологічним і психологічним вимогам, а також характеру роботи. Конструкція робочих меблів дає можливість забезпечувати можливість індивідуального регулювання їх відповідно до потреб працівника для підтримки зручної пози. Робочий стіл повинен бути пофарбований матовою фарбою. Дисплей розташований так, що його верхній край перебуває на рівні очей, на відстані близько 70 см, що укладається в припустимі рамки від 60 до 90 см. Частота мерехтіння екрана дорівнює 100 Гц, що відповідає умові більше 70 Гц.

Доцільно впроваджувати виконання комплексу вправ, які наведені у Державних санітарних правилах і нормах роботи з візуальними терміналами електронно-обчислювальних машин ДСанПіН 3.3.2.007-98 для зниження нервово-емоційного напруження, стомлювання, поліпшення мозкового кровообігу, подолання несприятливих наслідків гіподинамії, запобігання втомі

Працюючі з ПК підлягають обов'язковим медичним оглядам: попереднім – при влаштуванні на роботу і періодичним – протягом трудової діяльності, відповідно до наказу МОЗ України № 45.

Основними критеріями оцінки придатності до роботи з ПК мають бути показники стану органів зору: гострота зору, показники рефракції, стану

					<b>РП 07. 07 003. 00 ДП ПЗ</b>	Арк.
						61
Ізм.	Лист	№ докум.	Підпис	Дата		

біноккулярного апарату ока тощо. При цьому необхідно враховувати також стан організму в цілому.

### **3.4 Пожежна безпека**

Пожежі руйнують виробничі будівлі. знищують матеріали і готову продукцію, приводять в негідність обладнання, на тривалий час припиняють роботи в цехах. Протипожежний захист приміщення забезпечується застосуванням автоматичної установки пожежної сигналізації, наявністю засобів пожежогасіння, застосуванням основних будівельних конструкцій будинку з регламентованими межами вогнестійкості, організацією своєчасної евакуації людей.

До засобів гасіння пожежі відносяться внутрішні пожежні водопроводи (крани-ПК), вогнегасники (вуглекислотні та порошкові), сухий пісок тощо.

Громадяни України мають право на захист свого життя і здоров'я від аварій, катастроф, пожеж, стихійних лих та інших небезпечних ситуацій. Держава як гарант цього права здійснює свою політику в сфері цивільного захисту населення.

					<i>РП 07. 07 003. 00 ДП ПЗ</i>	Арк.
						62
Ізм.	Лист	№ докум.	Підпис	Дата		

## ВИСНОВКИ

В ході виконання дипломного проєкту було розроблено веб-платформу, спрямовану на автоматизацію управління ОСББ. Застосування передових програмних рішень і баз даних, адаптованих до вимог сучасного ринку, виявилось стратегічно важливим. Проведені дослідження довели, що впровадження такої системи сприятиме оптимізації процесів управління багатоквартирними будинками, надаючи конкурентні переваги та підвищуючи ефективність функціонування. Використання веб-орієнтованих інструментів дозволить оперативно реагувати на зміни у вимогах споживачів, оптимізувати робочі процеси та забезпечити високу якість обслуговування спільноти.

Проєкт реалізовувався з використанням різноманітних технологій, включаючи MySQL для управління базами даних та мови програмування PHP та JavaScript для обробки даних. Цей комплекс інструментів дозволив створити високоефективний продукт, готовий відповідати потребам сучасного управління ОСББ. У створеній системі реалізовано широкий спектр функціоналу, включаючи інструменти для автоматизації обліку ресурсів, моніторингу стану будинків, а також забезпечення зв'язку між мешканцями та управителем.

Пояснювальна записка охоплює всі аспекти, визначені технічним завданням, включаючи аналіз сучасного стану управління ОСББ, детальний опис використаних технологій та розрахунок ефективності впровадження системи. Окремо розглядаються питання безпеки праці та зображено перелік використаних джерел, що підтверджують науковий підґрунток та актуальність проєкту.

					<i>РП 07. 07 000. 00 ДП ПЗ</i>	Арк.
						63
Ізм.	Лист	№ докум.	Підпис	Дата		

# ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. МОСОСББ. [Веб-сайт]. URL: <https://osbb.online/>.
2. ОСББ-online. [Веб-сайт]. URL: <https://www.osbb-online.com/>.
3. Мій Дім Online. [Веб-сайт]. URL: <https://miydimonline.com.ua/>.
4. К. В. Двірничук, Д. О. Вацек. Веб-програмування та веб-дизайн: Навчальний посібник. – «Чернівецький національний університет імені Юрія Федьковича», 2022.
5. HTML Docs. [Веб-сайт]. URL: <https://developer.mozilla.org/en-US/docs/Web/HTML>.
6. CSS Docs. [Веб-сайт]. URL: <https://developer.mozilla.org/en-US/docs/Web/CSS>.
7. Bootstrap Docs. [Веб-сайт]. URL: <https://getbootstrap.com/docs/5.3/getting-started/introduction/>.
8. В. В. Босько, Л. В. Константинова, К. М. Марченко, О. С. Улічев. Web-програмування. Частина 1 (Frontend): Навчальний посібник. – «Кропивницький центральноукраїнський національний технічний університет», 2022.
9. JavaScript Docs. [Веб-сайт]. URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/>.
10. jQuery Docs. [Веб-сайт]. URL: <https://api.jquery.com/>.
11. С. І. Доценко. Організація та системи керування базами даних: Навчальний посібник. – «Український державний університет залізничного транспорту», 2023.
12. SQL Tutorial. [Веб-сайт]. URL: [https://www.w3schools.com/sql/sql\\_intro.asp](https://www.w3schools.com/sql/sql_intro.asp).
13. MySQL Docs. [Веб-сайт]. URL: <https://dev.mysql.com/doc/>.
14. О. С. Бунке. Серверні WEB-технології: Навчальний посібник. – «КПІ ім. Ігоря Сікорського», 2023.
15. PHP Docs. [Веб-сайт]. URL: <https://www.php.net/docs.php/>.
16. ZeroToMastery. [Веб-сайт]. URL: <https://zerotomastery.io/>.

					<b>РП 07. 07 000. 00 ДП ПЗ</b>	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		64

# ДОДАТОК А. Програмний код основної логіки веб-застосунку

```
// BoardModel.php

<?php
class BoardModel
{
    private $db;

    public function __construct()
    {
        $this->db = new Database;
    }

    public function selectByCategoryAndPage($page, $perPage, $category)
    {
        // Calculate the offset based on the current page and items per page
        $offset = ($page - 1) * $perPage;

        // Query with pagination
        $this->db->query('SELECT board.*, users.name AS `name` FROM board INNER JOIN
users ON board.user = users.id WHERE board.category = :category LIMIT :perPage
OFFSET :offset');;

        // Bind values
        $this->db->bind(':category', $category);
        $this->db->bind(':perPage', $perPage);
        $this->db->bind(':offset', $offset);

        // Execute
        $results = $this->db->resultSet();

        // Return result
        return $results;
    }

    public function total($category)
    {
        // Query
        $this->db->query('SELECT COUNT(*) as total FROM board WHERE category
= :category');;

        // Bind values
        $this->db->bind(':category', $category);

        // Execute
        $result = $this->db->single()->total;

        // Return result
        return $result;
    }

    public function insert($data)
    {
        // Query
        $this->db->query('INSERT INTO board (category, user, message, photo)
VALUES(:category, :user, :message, :photo)');
```

```

// Bind values
$this->db->bind(':category', $data['category']);
$this->db->bind(':user', $data['user']);
$this->db->bind(':message', $data['message']);
$this->db->bind(':photo', $data['photo']);

// Execute
if($this->db->execute())
{
    return true;
}
else
{
    return false;
}
}

public function delete($id)
{
    // Query
    $this->db->query('DELETE FROM board WHERE id = :id');

    // Bind values
    $this->db->bind(':id', $id);

    // Execute
    if($this->db->execute())
    {
        return true;
    }
    else
    {
        return false;
    }
}
}
?>

```

```
// navmenu.php
```

```

<div>
    <div class="row">
        <div class="col text-center">
            <h1 class="text-center"><strong>ОСББ</strong></h1>
            <p class="text-muted"><?php echo $data['information']->name; ?></p>
            <hr>
        </div>
    </div>
</div>
<ul class="nav flex-column nav-fill">
    <li class="nav-item"><a class="nav-link" href="<?php echo
URLROOT; ?>/">Головна</a></li>
    <li class="nav-item"><a class="nav-link" href="<?php echo
URLROOT; ?>/login">Реєстрація</a></li>
    <li class="nav-item"><a class="nav-link" href="<?php echo
URLROOT; ?>/login">Авторизація</a></li>
    <li class="nav-item"><a class="nav-link" href="<?php echo
URLROOT; ?>/user">Картка мешканця</a></li>
    <li class="nav-item"><a class="nav-link" href="<?php echo
URLROOT; ?>/admin">Адміністрування</a></li>

```



```

        <div class="col">
            <h4 class="m-1"><?php echo $item-
>name; ?></h4>
            <p class="m-1"><?php echo $item-
>message; ?></p>
        </div>
    </div>
    <?php if (isLoggedInAdmin()) : ?>
    <a class="btn btn-danger w-100 m-1"
href="<?php echo URLROOT; ?>/board/deleteMessage/<?php echo $item->id; ?>">
        Видалити
    </a>
    <?php endif; ?>
</div>
</div>
    <?php endforeach; ?>
</div>
<!-- Pagination -->
<div class="pagination m-3">
    <?php for ($p = 1; $p <= $data['totalPages']; $p++) : ?>
        <a href="<?php echo URLROOT . '/' .
$data['category']->path . '/' . $p; ?>" class="page-link"><?php echo $p; ?></a>
    <?php endfor; ?>
</div>
<?php endif; ?>
</section>
</div>
<div class="col-md-4 m-3">
    <?php require_once APPROOT . '/views/template/navmenu.php'; ?>
</div>
</div>
</div>
</section>

```

```
</body>
```

```
</html>
```

```
// BoardController.php
```

```

<?php
class BoardController extends Controller {
    private $boardModel;
    private $categoryModel;
    private $informationModel;

    public function __construct() {
        if(!isLoggedInUser()) {
            redirect('login');
        }
        $this->boardModel = $this->model('Board');
        $this->categoryModel = $this->model('Category');
        $this->informationModel = $this->model('Information');
    }

    public function index($id, $page = 1) {
        $information = $this->informationModel->select();
        $category = $this->categoryModel->selectById($id);
        $perPage = 2; // Define $perPage before using it
        $board = $this->boardModel->selectByCategoryAndPage($page, $perPage, $category-
>id);
    }
}

```

```

$total = $this->boardModel->total($category->id);

// Calculate the total number of pages
$totalPages = ceil($total / $perPage);

$data = [
    'information' => $information,
    'board' => $board,
    'category' => $category,
    'currentPage' => $page,
    'perPage' => $perPage,
    'totalPages' => $totalPages
];

$this->view('Board', $data);

flash('message');
}

public function createMessage() {
    try {
        // Check if the user is an admin and if it's a POST request
        if ($_SERVER['REQUEST_METHOD'] == 'POST') {
            // Sanitize input data
            $_POST = filter_input_array(INPUT_POST,
FILTER_SANITIZE_FULL_SPECIAL_CHARS);

            // Check if photo is uploaded
            if (isset($_FILES['photo']) && $_FILES['photo']['error'] ===
UPLOAD_ERR_OK) {
                $uploads_dir = 'uploads/';
                $file = uniqid() . '.' . pathinfo($_FILES["photo"]["name"],
PATHINFO_EXTENSION);
                move_uploaded_file($_FILES['photo']['tmp_name'], $uploads_dir.$file);
            } else {
                // Handle case where photo is not uploaded
                $file = ''; // Set photo field to empty string if no photo uploaded
            }

            // Prepare data for insertion
            $data = [
                'category' => trim($_POST['category']),
                'user' => $_SESSION['id'], // Assuming user_id is stored in session
                'message' => trim($_POST['message']),
                'photo' => $file, // Uploaded file name
            ];

            // Call the insert method of the board model to add the message
            if ($this->boardModel->insert($data)) {
                // Message added successfully
                flash('message', 'Операція успішна! ');
                redirectBack();
            } else {
                // Failed to add message
                flash('message', 'Операція не успішна! ');
                redirectBack();
            }
        }
    } catch (Exception $e) {
        // Handle exceptions
        flash('message', 'Виникла помилка: ' . $e->getMessage());
        redirectBack();
    }
}

```

```

    }
}
public function deleteMessage($id) {
    try {
        if ($_SESSION['admin'] == true) {
            if ($this->boardModel->delete($id)) {
                flash('message', 'Операція успішна!');
                redirectBack();
            }
        }
    }
    catch (Exception $e) {
        flash('message', 'Операція не успішна!');
        redirectBack();
    }
}
}
?>

```

// LoginController.php

```

<?php
class LoginController extends Controller {
    private $userModel;
    private $informationModel;

    public function __construct() {
        if(isLoggedInAdmin()) {
            redirect('admin');
        }
        else if (isLoggedInUser()) {
            redirect('user');
        }

        $this->userModel = $this->model('User');
        $this->informationModel = $this->model('Information');
    }

    public function index() {
        $information = $this->informationModel->select();

        $data = [
            'information' => $information,
        ];

        $this->view('Login', $data);

        flash('message');
    }

    public function signIn() {
        // Check for POST
        if ($_SERVER['REQUEST_METHOD'] == 'POST') {
            // Process form

            // Sanitize POST data
            $_POST = filter_input_array(INPUT_POST, FILTER_SANITIZE_FULL_SPECIAL_CHARS);

            // Init data
            $data = [
                'login' => trim($_POST['login']),
                'password' => trim($_POST['password'])
            ];
        }
    }
}

```

```

];

    $loggedInUser = $this->userModel->select($data['login'], $data['password'],
1);

    // Validate user
    if($loggedInUser) {
        setSession($loggedInUser);
        redirect('login');
    }
    else
    {
        flash('message', 'Операція неуспішна! Можливо, не вірні дані. Спробуйте
інші дані.');
```

```

        redirect('login');
    }
}

public function signUp() {
    // Check for POST
    if ($_SERVER['REQUEST_METHOD'] == 'POST') {
        // Process form
        // Sanitize POST data
        $_POST = filter_input_array(INPUT_POST, FILTER_SANITIZE_FULL_SPECIAL_CHARS);

        // Check if the provided secret matches the secret in the information model
        $secret = trim($_POST['secret']);

        $info = $this->informationModel->select();

        if ($info && $info->secret != $secret)
        {
            flash('message', 'Неправильний секрет! Перевірте і спробуйте ще раз.');
```

```

            redirect('login');
        }

        // Init data
        $data = [
            'name' => trim($_POST['name']),
            'apartment' => trim($_POST['apartment']),
            'email' => trim($_POST['email']),
            'phone' => trim($_POST['phone']),
            'login' => trim($_POST['login']),
            'password' => trim($_POST['password'])
        ];

        // Hash password
        $data['password'] = password_hash($data['password'], PASSWORD_DEFAULT);

        // Validate action
        if($this->userModel->insert($data)) {
            flash('message', 'Операція успішна! ');
            redirect('login');
        }
        else {
            flash('message', 'Операція неуспішна! Можливо, не вірні дані. Спробуйте
інші дані.');
```

```

            redirect('login');
        }
    }
}
}

```

```

        public function signOut() {
            unsetSession();

            redirect('home');
        }
    }
?>

// AdminController.php

<?php
class AdminController extends Controller {
    private $informationModel;

    public function __construct() {
        if(!isLoggedInAdmin()) {
            redirect('login');
        }
        $this->informationModel = $this->model('Information');
    }

    public function index() {
        $information = $this->informationModel->select();

        $data = [
            'information' => $information,
        ];

        $this->view('Admin', $data);

        flash('message');
    }

    public function updateInformation() {
        try {
            if ($_SESSION['admin'] == true) {
                if($_SERVER['REQUEST_METHOD'] == 'POST') {
                    $_POST = filter_input_array(INPUT_POST,
FILTER_SANITIZE_FULL_SPECIAL_CHARS);

                    // Check if photo is uploaded
                    if (isset($_FILES['photo']) && $_FILES['photo']['error'] ===
UPLOAD_ERR_OK) {
                        $uploads_dir = 'uploads/';
                        $file = uniqid() . '.' . pathinfo($_FILES["photo"]["name"],
PATHINFO_EXTENSION);
                        move_uploaded_file($_FILES['photo']['tmp_name'], $uploads_dir.$file);
                    } else {
                        // Handle case where photo is not uploaded
                        $file = ''; // Set photo field to empty string if no photo uploaded
                    }

                    $data = [
                        'id' => trim($_POST['id']),
                        'name' => trim($_POST['name']),
                        'phone' => trim($_POST['phone']),
                        'email' => trim($_POST['email']),
                        'address' => trim($_POST['address']),
                        'message' => trim($_POST['message']),
                        'photo' => $file,
                    ];
                }
            }
        } catch (Exception $e) {
            flash('error', $e->getMessage());
        }
    }
}

```



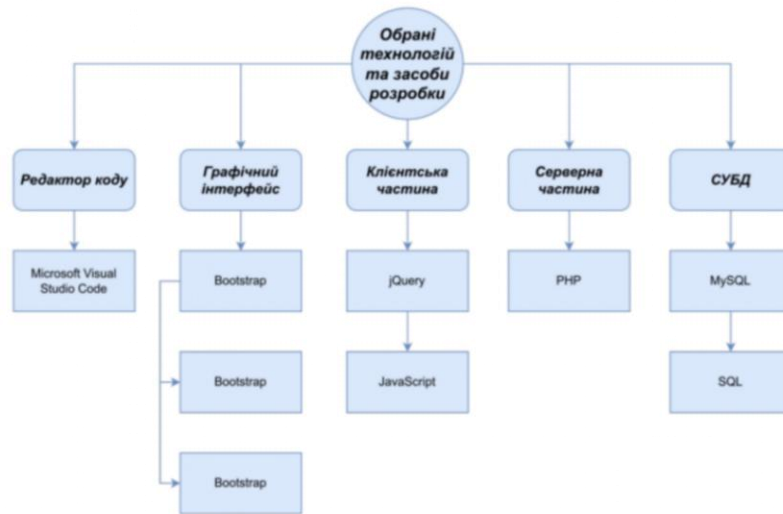
# Розробка автоматизованої веб-системи для ОСББ

Кучеренко Сергій Юрійович

## Предметна область

- Автоматизовані веб-системи для управління багатоквартирними будинками (ОСББ) забезпечують комфортність, швидкість та ефективність взаємодії з мешканцями, підвищуючи продуктивність та рівень обслуговування.
- Метою дипломного проєктування є створення веб-системи для ОСББ, яка оптимізує управління житловим фондом за допомогою інформаційних технологій.
- Практичне значення системи полягає в її корисності для членів ОСББ, допомагаючи ефективно управляти багатоквартирними будинками.

# Обрані технології розробки

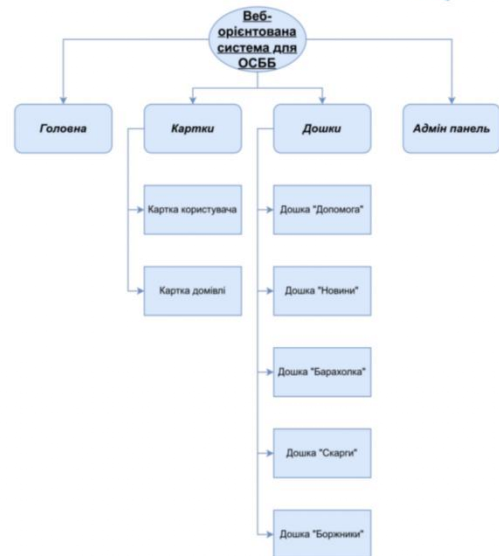


# Розділи веб-системи

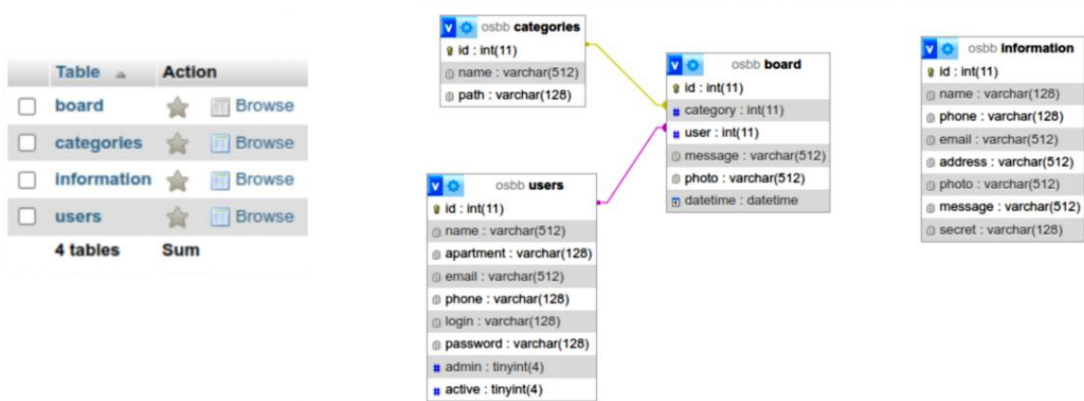
## ОСББ

м. Одеса, вул. Приморська, буд. 1

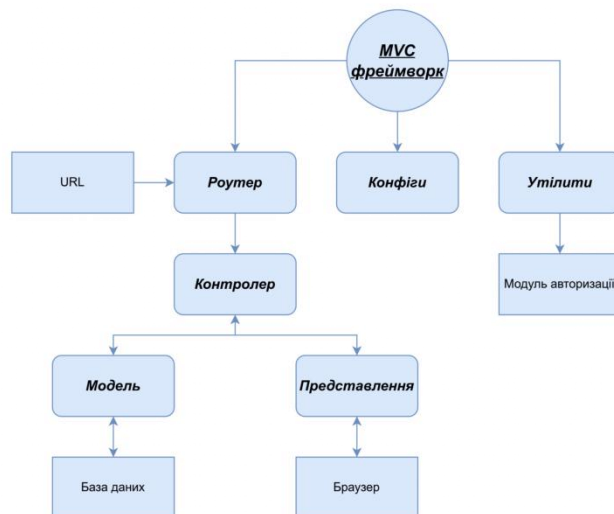
- Головна
- Реєстрація
- Авторизація
- Картка мешканця
- Адміністрування
- Картка будинку
- Допомога
- Новини
- Барахолка
- Скарги
- Боржники



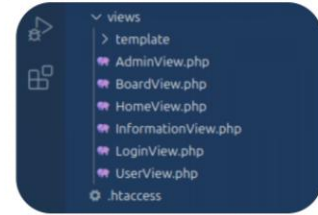
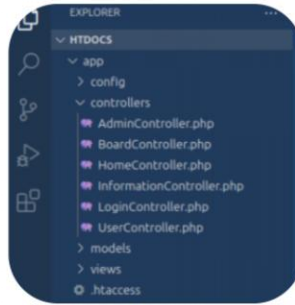
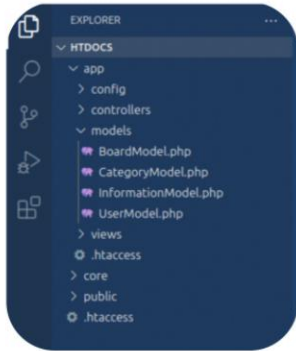
# Схема бази даних



# Архітектура веб-системи



# Розробка веб-системи



Server		Status
MySQL Database	Running	
ProFTPD	Running	
Apache Web Server	Running	

# Обробка запиту на створення повідомлення

```
public function createMessage() {
    try {
        // Check if the user is an admin and if it's a POST request
        if ($_SERVER['REQUEST_METHOD'] == 'POST') {
            // Sanitize input data
            $_POST = filter_input_array(INPUT_POST,
            FILTER_SANITIZE_FULL_SPECIAL_CHARS);

            // Check if photo is uploaded
            if (!isset($_FILES['photo']) || $_FILES['photo']['error']
            === UPLOAD_ERR_OK) {
                $uploads_dir = 'uploads/';
                $file = uniqid() . '.' .
                pathinfo($_FILES["photo"]["name"], PATHINFO_EXTENSION);
                move_uploaded_file($_FILES['photo']['tmp_name'],
                $uploads_dir.$file);
            } else {
                // Handle case where photo is not uploaded
                $file = ''; // Set photo field to empty string if no
                photo uploaded
            }
        }
    }
```

```
// Prepare data for insertion
$data = [
    'category' => trim($_POST['category']),
    'user' => $_SESSION['id'], // Assuming user_id is
    stored in session
    'message' => trim($_POST['message']),
    'photo' => $file, // Uploaded file name
];

// Call the insert method of the board model to add the
message
if ($this->boardModel->insert($data)) {
    // Message added successfully
    flash('message', 'Операція успішна! ');
    redirectBack();
} else {
    // Failed to add message
    flash('message', 'Операція не успішна! ');
    redirectBack();
}
} catch (Exception $e) {
    // Handle exceptions
    flash('message', 'Виникла помилка: ' . $e->getMessage());
    redirectBack();
}
}
```

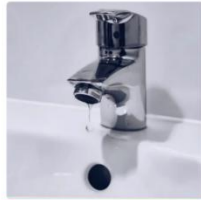
# Приклад дошки "Допомога"

## Допомога

Повідомлення

Choose FileNo file chosen

ВІДПРАВИТИ



### Адміністратор

Не зачиняється кран. Чи є в дні сантехніка?

ВИДАЛИТИ

1

## ОСББ

ЖК "Мрія"

- Головна
- Реєстрація
- Авторизація
- Картка мешканця
- Адміністрування
- Картка будинку
- Допомога
- Новини
- Барахолка
- Скарги
- Боржники

# Налаштування ОСББ адміністратором

## Адміністрування

Ім'я	ЖК "Мрія"
Адреса	м. Одеса, вул. Приморська, буд. 1
Телефон	+380669996699
Е-Май	name@domain.com
Повідомлення	Наразі екстрені повідомлення відсутні!
Фото	Choose FileNo file chosen
Максимальний розмір: 5MB	Секрет
2024	
<p>ЗБЕРЕГТИ</p> <p>ВИХІД</p>	

## ОСББ

ЖК "Мрія"

- Головна
- Реєстрація
- Авторизація
- Картка мешканця
- Адміністрування
- Картка будинку
- Допомога
- Новини
- Барахолка
- Скарги
- Боржники

## Картка будинку



ЖК "Мрія"

м. Одеса, вул. Приморська, буд. 1

## ОСББ

ЖК "Мрія"

- Головна
- Реєстрація
- Авторизація
- Картка мешканця
- Адміністрування
- Картка будинку
- Допомога
- Новини
- Барахолка
- Скарги
- Боржники

# Реєстрація за секретним ключем

## Авторизація та Реєстрація

Ім'я

Прізвище

Спеціальність/назва

Навчальний заклад

Стать

Телефон

Електронна пошта

[РЕГІСТРАЦІЯ](#)  
реєстрація лише для існуючих користувачів

Ім'я

Прізвище

[РЕГІСТРАЦІЯ](#)  
реєстрація лише для існуючих користувачів

## ОСББ

- Головна
- Реєстрація
- Авторизація
- Користувачі
- Адміністратори
- Користувачі
- Додатки
- Налаштування
- Сторінка
- Сторінка
- Сторінка

Дякую за увагу!

Буду рад відповісти на Ваші запитання

**ВІДГУК**

керівника на дипломний проект здобувача (здобувачки) освіти  
відділення комп'ютерних систем

*Кучеренко Сергія Юрійовича*

(прізвище, ім'я та по батькові)

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: «Розробка програмного забезпечення»

Тема дипломного проекту: Розробка автоматизованої веб-системи для ОСББ

**ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ**

а) обсяг і якість виконання проекту (графічного матеріалу і розрахунково-пояснювальної записки) Дипломний проект виконано відповідно технічному завданню. Пояснювальна записка до дипломного проекту містить 79 сторінок. У пояснювальній записці описано етапи розробки автоматизованої веб-системи для ОСББ засобами PHP та MySQL. Графічна частина складається з окремих слайдів, оформлених у вигляді презентації, передбачених технічним завданням. Якість виконання пояснювальної записки та слайдів добра.

б) самостійність роботи над проектом: Протягом виконання дипломного проекту здобувач освіти Кучеренко Сергій поступово та послідовно виконував всі етапи, проявляв ініціативу в створенні загальної концепції та реалізації роботи. Всі роботи здобувач освіти виконував самостійно, з оглядом на рекомендації керівника.

в) теоретична підготовка випускника (випускниці): Здобувач освіти Кучеренко Сергій під час роботи над дипломним проектом вивчив достатньо багато літературних та інтернет-джерел за даною тематикою.

Вважаю, що теоретична підготовка дипломника достатня і він готовий до захисту проекту.

г) вміння розв'язувати виробничі та конструкторські питання Під час виконання дипломного проекту здобувач освіти Кучеренко Сергій показав вміння організовано працювати над поставленим завданням, застосовувати знання у галузі програмування та математики, розробляти, встановлювати та налаштовувати спеціалізоване програмне забезпечення, оформлювати слайди та складати презентації, користуючись сучасними комп'ютерними програмними засобами, такими як MS VS Code, PHP, MySQL, MS PowerPoint, MS Visio та ін.

Оцінка розрахункової частини Відмінно

Оцінка графічної частини Добре

Загальна оцінка Відмінно

Прізвище, ім'я, по батькові керівника дипломного проекту \_\_\_\_\_

Жадан Артур Сергійович

Місце роботи і посада керівника дипломного проекту ВСП «Одеський технічний фаховий коледж ОНТУ», викладач спецдисциплін циклової комісії комп'ютерної техніки та програмної інженерії

Підпис \_\_\_\_\_

«10» 06 2024 р.

## РЕЦЕНЗІЯ

на дипломний проект здобувача (здобувачки) освіти  
відділення комп'ютерних систем

*Кучеренко Сергія Юрійовича*

(прізвище, ім'я та по батькові)

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма «Розробка програмного забезпечення»

Керівник дипломного проекту (роботи) Жадан Артур Сергійович

(прізвище, ім'я та по батькові)

Тема дипломного проекту (роботи) Розробка автоматизованої веб-системи для ОСББ

Обсяг розрахунково-пояснювальної записки 79 сторінок

Обсяг графічної (презентаційної) частини 12 аркушів (слайдів)

### ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ (РОБОТИ)

а) заключення про ступінь відповідності виконаного дипломного проекту завданню

*Представлений на рецензію дипломний проект відповідає затвердженій темі та виконаний відповідно технічному завданню. Дипломний проект присвячений проблемі управління багатоквартирними будинками та складається з пояснювальної записки, додатку з програмним кодом та мультимедійної презентації, що містить приклади роботи програми.*

б) характеристика виконання кожного розділу дипломного проекту \_\_\_\_\_

*Пояснювальна записка складається з основного розділу (аналізу предметної області, проектування застосунку, реалізації застосунку, тестування застосунку), економічного розділу, розділу охорони праці та додатків. Перелічені розділи поетапно охоплюють розробку, виконані докладно та обґрунтовано. Розділ охорони праці містить загальну інформацію та вимоги до техніки безпеки оператора КТ. Економічний розділ проекту містить розрахунок витрат на НДР та реалізацію проекту.*

в) оцінка якості виконання пояснювальної записки та графічної частини дипломного проекту

*Графічна частина складається з 12 слайдів мультимедійної презентації, виконаної у програмному продукті MS PowerPoint, які містять ілюстративні схеми, скриншоти роботи програмного застосунку, передбачені технічним завданням. Пояснювальна записка виконана акуратно та у відповідності до норм. Якість виконання графічної частини проекту та пояснювальної записки відмінна, розробку виконано у повному обсязі.*

г) перелік позитивних якостей дипломного проекту Реалізовано автоматизовану веб-систему для ОСББ, що дозволяє управляти багатоквартирними будинками засобами категоризованих інформаційних дошок.

Стилізація графічного інтерфейсу відповідає тематиці обраної предметної області.

Розроблена веб-система добре підходить для невеликих будинків.

д) основні недоліки дипломного проекту

1. У роботі не представлено блок-схем алгоритмів для створених скриптів;

2. При впровадженні веб-системи для великих будинків, можуть виникнути певні складнощі у зв'язку із обмеженнями веб-системи;

3. Наявні деякі помилки оформлення тексту пояснювальної записки

Оцінка розрахункової частини Добре

Оцінка графічної частини Добре

Загальна оцінка Добре

Прізвище, ім'я, по батькові рецензента Васіліу Євген Вікторович

Місце роботи і посада рецензента Державний університет інтелектуальних технологій і зв'язку, д.т.н., проф. кафедри КБ та ТЗІ



06

2024 р.

Ім'я користувача:  
Катерина Григоріївна Краснокутська

ID перевірки:  
1016336426

Дата перевірки:  
08.06.2024 21:39:51 EEST

Тип перевірки:  
Doc vs Internet + Library

Дата звіту:  
08.06.2024 21:46:11 EEST

ID користувача:  
100011688

Назва документа: 4РП-07\_Кучеренко\_С

Кількість сторінок: 48 Кількість слів: 6587 Кількість символів: 51168 Розмір файлу: 2.74 МВ ID файлу: 1016137192

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

**6.32%**  
**Схожість**

Найбільша схожість: 2.17% з Інтернет-джерелом ([https://repository.bsi.ac.id/index.php/unduh/item/336164/1\\_HKI\\_SITAW](https://repository.bsi.ac.id/index.php/unduh/item/336164/1_HKI_SITAW))

6.32% Джерела з Інтернету

448

Сторінка 50

Не знайдено джерел з Бібліотеки

**0% Цитат**

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

**0%**  
**Вилучень**

Немає вилучених джерел

**Модифікації**

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Підозріле форматування

11  
сторінок

**ДОЗВІЛ  
НА РОЗМІЩЕННЯ  
ВИПУСКНОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ  
(ДИПЛОМНОГО ПРОЕКТУ)  
В ЕЛЕКТРОННОМУ РЕПОЗИТАРІЇ ВСП «ОТФК ОНТУ»**

Ми, що нижче підписалися,

**Кучеренко Сергій Юрійович,**  
здобувач освіти гр. 4РП-07, та

**Жадан Артур Сергійович,**  
керівник дипломного проекту,

не заперечуємо щодо розміщення електронного варіанту пояснювальної записки до дипломного проекту фахового молодшого бакалавра на тему:

**«Розробка автоматизованої веб-системи для ОСББ» (автор роботи – Кучеренко С.Ю., керівник роботи – Жадан А.С.)**

виконаного у ВСП «Одеський технічний фаховий коледж Одеського національного технологічного університету» в 2024 році, у повному обсязі в електронному репозитарії ВСП «ОТФК ОНТУ» для вільного доступу через мережу Інтернет.

Несемо відповідальність за ідентичність електронного та друкованого варіантів випускної кваліфікаційної роботи і даємо згоду на обробку персональних даних.

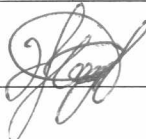
Виконавець



---

/ Кучеренко С.Ю. /

Керівник



---

/ Жадан А.С. /

«10» червня 2024 р.