

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВОДНОГО ГОСПОДАРСТВА ТА
ПРИРОДОКОРИСТУВАННЯ

Навчально-науковий інститут кібернетики, інформаційних технологій та
інженерії

Кафедра комп'ютерних наук та прикладної математики

«До захисту допущена»

Завідувач кафедри комп'ютерних наук
та прикладної математики

_____ Турбал Ю.В.

«_____» _____ 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Розробка вебзастосунку для ОСББ на основі технології Laravel»

Виконав: Сорока Володимир Романович

(прізвище, ім'я, по батькові)

(підпис)

група КН-41

Керівник: старший викладач Харів Н.О.

(науковий ступінь, вчене звання, посада, прізвище та ініціали)

(підпис)

Рівне – 2024

Навчально-науковий інститут кібернетики, інформаційних технологій та інженерії

Кафедра комп'ютерних наук та прикладної математики

Рівень вищої освіти бакалавр

Галузь знань 12 «Інформаційні технології»

Спеціальність 122 «Комп'ютерні науки»

«ЗАТВЕРДЖУЮ»

Завідувач кафедри
комп'ютерних наук та
прикладної математики д.т.н.,
професор Турбал Ю.В.

«_____» _____ 2024 року

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Сорока Володимир Романович

(прізвище, ім'я, по батькові)

1. Тема роботи «Розробка вебзастосунку для ОСББ на основі технології Laravel»
керівник роботи Харів Наталія Олексіївна, старший викладач кафедри
комп'ютерних наук та прикладної математики

затверджені наказом вищого навчального закладу від «22» квітня 2024 року

№ 525

2. Термін здачі студентом закінченої роботи 18 квітня 2024 р.

3. Вихідні дані до роботи: _____

4. Зміст розрахунково-пояснювальної записки Розділ 1. Розділ 2. Розділ 3

5. Перелік графічного матеріалу мультимедійна презентація

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
<i>Розділ 1</i>	<i>Харів Н.О., ст. викладач</i>		
<i>Розділ 2</i>	<i>Харів Н.О., ст. викладач</i>		
<i>Розділ 3</i>	<i>Харів Н.О., ст. викладач</i>		

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	<i>Опрацювання літератури на задану тему.</i>	<i>До 15 квітня 2024 року</i>	
2	<i>Визначення з усіма інструментами розробки</i>	<i>До 2 травня 2024 року</i>	
3	<i>Розробка плану реалізації та проектування</i>	<i>До 20 травня 2024 року</i>	
4	<i>Реалізація системи</i>	<i>До 1 червня 2024 року</i>	
5	<i>Підготовка звіту</i>	<i>До 10 червня 2024 року</i>	
6	<i>Створення презентації роботи</i>	<i>До 15 червня 2024 року</i>	
7	<i>Подання роботи на відгук та рецензію</i>	<i>До 18 червня 2024 року</i>	
8			
9			

Студент _____ (Сорока В.Р.)

Керівник кваліфікаційної роботи _____ (Харів Н.О.)

ЗМІСТ

РЕФЕРАТ.....	6
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	7
ВСТУП.....	8
РОЗДІЛ 1. ОПИС ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	10
1.1. Поняття терміну ОСББ.....	10
1.2. Історичні відомості.....	10
1.3. Класифікація ОСББ.....	11
1.4. Виконувані роботи ОСББ.....	11
1.5. Аналіз існуючих рішень.....	13
1.6. Переваги та недоліки сучасних сайтів ОСББ.....	13
1.7. Проміжні висновки.....	14
РОЗДІЛ 2. ОГЛЯД ВИКОРИСТАНИХ ІНСТРУМЕНТІВ ТА ТЕХНОЛОГІЙ.....	15
2.1. Бекенд.....	15
2.1.1. Мова програмування PHP.....	15
2.1.2. Фреймворк Laravel.....	16
2.1.3. СУБД MySQL.....	18
2.2 Фронтенд.....	20
2.2.1 HTML, CSS, JS.....	20
2.2.2 Фреймворк Bootstrap.....	21
РОЗДІЛ 3. ОПИС РОЗРОБЛЕНОГО ПРОДУКТУ.....	23
3.1. ОСББ «Затишний дім».....	23

ВИСНОВКИ.....	29
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	31

РЕФЕРАТ

Кваліфікаційна робота: 31 с. , 10 малюнків, 5 джерел

Мета: розробка вебзастосунку для практичного використання

Завдання: розробка власного рішення для управління ОСББ

Методи вивчення:

1. Опрацювання літератури та аналіз подібних рішень
2. Проектування вебдодатку
3. Практична реалізація
4. Тестування та узагальнення результатів

Ключові слова: вебресурс, ОСББ, управління будинком онлайн

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

Об'єднання співвласників багатоквартирного будинку (ОСББ) – це форма організації, яка об'єднує мешканців багатоквартирного будинку з метою спільного управління та вирішення питань, що стосуються утримання та експлуатації будинку.

СУБД (Система Управління Базами Даних) — це програмне забезпечення, яке використовується для створення, управління та маніпулювання базами даних.

PHP (Hypertext Preprocessor) — це скриптова мова програмування загального призначення, яка використовується для розробки веб-додатків.

Laravel – це відомий веб-фреймворк для PHP, який надає розробникам потужний інструментарій для створення сучасних веб-додатків.

MySQL – це відкрита реляційна система керування базами даних (СКБД), яка використовується для зберігання і управління структурованими даними.

PhpStorm - це інтегроване середовище розробки (IDE) для PHP, розроблене компанією JetBrains. Воно спеціалізується на підтримці і полегшенні розробки PHP-програм на основі різноманітних фреймворків і бібліотек.

Frontend (або "фронтенд") – це частина веб-додатка або веб-сайту, яка відповідає за взаємодію з користувачем і відображення інформації.

Backend (або "бекенд") – це частина веб-додатка або веб-сайту, яка відповідає за обробку логіки, доступ до бази даних, аутентифікацію користувачів та взаємодію з іншими системами.

ВСТУП

В умовах стрімкого розвитку інформаційних технологій особливу увагу привертає можливість автоматизації та оптимізації управлінських процесів у різних сферах діяльності. Однією з таких сфер є управління житлово-комунальними послугами, зокрема діяльністю об'єднань співвласників багатоквартирних будинків (ОСББ). ОСББ відіграють важливу роль у забезпеченні комфорту та якості життя мешканців багатоквартирних будинків, займаючись питаннями обслуговування будинків, фінансовими питаннями, організацією спільних заходів та іншими важливими аспектами.

Ефективне управління ОСББ потребує злагодженої комунікації між мешканцями та адміністрацією, а також чіткого обліку фінансових та організаційних аспектів. У цьому контексті вебзастосунки стають незамінними інструментами для вирішення цих завдань. Вони дозволяють забезпечити прозорість, оперативність та ефективність управлінських процесів, що в свою чергу сприяє підвищенню рівня задоволеності мешканців.

Метою даної роботи є створення вебдодатку для управління ОСББ, надання актуальної та потрібної інформації співвласникам, перенесення більшості функцій ОСББ онлайн. Створення такого додатку дасть змогу ефективно керувати будинком, комунікувати, проводити різні опитування та голосування між жителями будинку.

Основними завданнями цієї розробки є:

Автоматизація управлінських процесів: Розроблений вебзастосунок повинен дозволити ефективно керувати різними аспектами управління ОСББ, такими як фінансовий облік, обслуговування будинку, управління заявками та комунікація з мешканцями.

Покращення доступності і зручності для користувачів: Вебзастосунок має забезпечувати зручний інтерфейс для користувачів будинку, щоб вони могли легко взаємодіяти з управлінням, подавати заявки, спілкуватися з іншими мешканцями і отримувати необхідну інформацію.

Підвищення ефективності управління: Використання сучасних інформаційних технологій, зокрема фреймворка Laravel, має забезпечити швидку розробку, стабільність і масштабованість вебзастосунку, що дозволить покращити процеси управління і зменшити адміністративні витрати ОСББ.

Актуальність даної роботи обумовлена необхідністю підвищення ефективності управління житловими будинками та покращення комунікації між мешканцями та адміністрацією. Існуючі рішення для управління ОСББ часто не відповідають сучасним вимогам або є занадто складними у використанні. Використання вебзастосунку на основі Laravel дозволить забезпечити доступний, функціональний та надійний інструмент для управління ОСББ, що сприятиме зростанню прозорості, оперативності та ефективності їх роботи.

Для досягнення поставлених завдань у роботі був використаний комплекс методів дослідження, включаючи аналіз літературних джерел, системний аналіз, проектування програмного забезпечення та методи тестування. Основний акцент робиться на практичному застосуванні технології Laravel для розробки функціонального вебзастосунку, який відповідає потребам ОСББ.

Аналіз літературних джерел дозволив визначити сучасні тенденції та найкращі практики у сфері управління житлово-комунальними послугами та розробки вебзастосунків. Системний аналіз забезпечив виявлення ключових вимог до вебзастосунку та розробку його архітектури. Проектування програмного забезпечення здійснювалося з урахуванням принципів модульності, масштабованості та безпеки, що забезпечить високу якість та надійність розробленого рішення. Тестування дозволить оцінити ефективність вебзастосунку та його відповідність вимогам користувачів.

Таким чином, розробка вебзастосунку для управління ОСББ на основі технології Laravel є важливим та актуальним завданням, яке сприятиме підвищенню ефективності управлінських процесів та покращенню якості життя мешканців багатоквартирних будинків. Використання сучасних технологій та методів розробки дозволить створити надійний та функціональний інструмент для управління ОСББ, що забезпечить прозорість, оперативність та ефективність їх роботи.

РОЗДІЛ 1

ОПИС ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Поняття терміну ОСББ

Об'єднання співвласників багатоквартирного будинку (ОСББ) є правовим суб'єктом, який утворюється співвласниками квартир (приміщень) для спільного управління і вирішення питань, пов'язаних з утриманням та експлуатацією будинку. Головною метою ОСББ є забезпечення ефективного управління будинком, покращення умов проживання мешканців, зниження адміністративних витрат та забезпечення транспарентності управління.

1.2. Історичні відомості

Ідея створення ОСББ на території України виникла у 2000-х роках як відповідь на проблеми управління багатоквартирними будинками. Раніше, управління такими будинками здійснювалося державними органами або комерційними управителями, що часто призводило до неефективного використання ресурсів і зниження якості обслуговування. Закон України "Про об'єднання співвласників багатоквартирних будинків" був прийнятий у 2004 році і забезпечив правову основу для створення ОСББ. Цей закон регулює порядок організації, управління та взаємодії ОСББ з іншими органами.

На сьогоднішній день ОСББ активно функціонують у більшості міст України. Вони забезпечують ефективне управління будинками, покращення умов проживання та зменшення адміністративних витрат. Сучасні ОСББ використовують сучасні інформаційні технології для автоматизації процесів управління, ведення обліку та комунікації з мешканцями.

Одним із ключових викликів, з якими стикаються ОСББ сьогодні, є підвищення ефективності управління, забезпечення фінансової стійкості та здійснення капітального ремонту будівель. Для цього використовуються різні підходи, включаючи залучення фахівців, розробку довгострокових стратегій та впровадження сучасних технологій управління.

1.3. Класифікація ОСББ

ОСББ можна класифікувати за різними ознаками, серед яких найбільш поширені:

1. За місцем знаходження:

- **Місцеві ОСББ:** Утворюються для кожного окремого будинку чи комплексу будинків.
- **Обласні/регіональні ОСББ:** Об'єднують кілька будинків у межах одного міста чи області для спільного управління.

2. За типом будинку:

- **Житлові ОСББ:** Формуються для управління житловими будинками.
- **Комерційні ОСББ:** Можуть утворюватися для управління нежитловими приміщеннями, такими як офісні або торгові приміщення.

3. За формою власності:

- **Приватні ОСББ:** Утворюються співвласниками, які мають право на власність приміщень у будинку.
- **Державні ОСББ:** Утворюються для управління будинками, що перебувають у власності держави чи місцевих органів влади.

4. За спеціалізацією:

- **Екологічні ОСББ:** Фокусуються на зниженні впливу будинку на навколишнє середовище та енергоефективності.
- **Соціальні ОСББ:** Займаються питаннями соціальної підтримки та покращення умов проживання мешканців.

1.4. Виконувані роботи ОСББ

Функції Об'єднань співвласників багатоквартирних будинків (ОСББ) включають широкий спектр обов'язків і завдань, спрямованих на забезпечення ефективного управління житловими будівлями та поліпшення умов проживання мешканців. Основні функції ОСББ можна умовно розділити на кілька категорій:

1. Управління майном

- **Планування і координація ремонтних робіт:** Організація та контроль за поточними і капітальними ремонтами будинків і прибудинкових територій.
- **Забезпечення технічної експлуатації:** Моніторинг технічного стану будинків, систем комунікацій, водопостачання, опалення та електропостачання.
- **Управління енергоефективністю:** Впровадження заходів з енергозбереження, використання альтернативних джерел енергії та підвищення енергоефективності будинків.

2. Фінансове управління

- **Бухгалтерський облік і фінансові звіти:** Ведення обліку доходів та витрат, формування річних фінансових звітів і звітів про використання коштів.
- **Визначення розміру внесків:** Встановлення розміру внесків від співвласників для покриття витрат на утримання будинків і розвиток інфраструктури.

3. Правове та адміністративне управління

- **Ведення документації:** Забезпечення правильного оформлення усіх необхідних документів, в тому числі протоколів зборів, рішень загальних зборів, договорів з постачальниками послуг.
- **Захист прав співвласників:** Представництво і захист інтересів співвласників перед державними органами, комерційними структурами та третіми особами.

4. Комунікації та взаємодія з мешканцями

- **Інформаційна підтримка:** Забезпечення доступу мешканців до актуальної інформації про роботу ОСББ, оголошення, новини та рекомендації.
- **Проведення загальних зборів:** Організація і проведення загальних зборів співвласників для ухвалення важливих рішень і вирішення актуальних питань.

5. Соціальна взаємодія та спільнота

- **Соціальні ініціативи:** Організація соціальних заходів, заходів з підтримки співвласників, покращення соціального клімату в будинках.

1.4. Аналіз існуючих рішень

На просторах інтернету розміщено безліч платформ та сервісів які надають свої послуги у вирішенні питань, що можуть охоплювати цілі ОСББ. Проте для вирішення даної проблеми потрібне більш комплексне рішення, яке полягало б у поєднанні багатьох можливостей та послуг одразу в одному місці. Також оглядаючи уже існуючі вебзастосунки для ОСББ, розміщених у вільному доступі, можна дійти висновку, що більшість з них не мають в собі значного функціоналу для керування будинком. Тому саме ідеєю було забезпечення можливих функцій та дій, що стосуються управління ОСББ, на вебсайті, а також виправлення недоліків уже існуючих систем.

1.5. Переваги та недоліки сучасних сайтів ОСББ

Переваги сучасних сайтів ОСББ:

1. **Зручний доступ до інформації:** Мешканці можуть легко отримати доступ до актуальної інформації про управління будинком, новини, розклади ремонтних робіт та інші важливі оголошення.
2. **Ефективне управління:** Веб-сайти дозволяють здійснювати ефективне управління фінансами, виконанням ремонтних робіт, плануванням заходів з підвищення енергоефективності тощо.
3. **Онлайн сервіси:** Мешканці можуть користуватися онлайн сервісами для подання заявок на ремонт, спілкування з управлінням, перегляду особистого кабінету для оплати комунальних послуг.
4. **Зменшення бюрократії:** Впровадження цифрових сервісів спрощує процеси обслуговування мешканців і зменшує паперову роботу для управління ОСББ.
5. **Забезпечення прозорості:** Відкритий доступ до фінансових звітів, протоколів загальних зборів і рішень дозволяє забезпечити прозорість управління перед мешканцями.

Недоліки сучасних сайтів ОСББ:

1. **Технічні проблеми:** Веб-сайти можуть мати проблеми з доступністю, швидкістю завантаження та іншими технічними недоліками, що можуть обмежувати їх ефективність.
2. **Низька інформаційна грамотність:** Не у всіх мешканців може бути достатньою рівень інформаційної грамотності для користування веб-сайтами або мобільними додатками.
3. **Потреба у підтримці:** Підтримка та оновлення веб-сайтів потребують витрат часу та ресурсів з боку управління ОСББ.

1.6. Проміжні висновки

Отже, в результаті виконання роботи було проведено ранній аналіз системи ОСББ, яка функціонує в Україні. Виконаний збір інформації, щодо існуючих рішень, оцінка переваг та недоліків для подальшого планування та розробки власної системи. Внаслідок цього можна зробити висновок, що сучасні веб-сайти ОСББ є важливим інструментом для покращення управління та комунікації з мешканцями. Вони забезпечують зручний доступ до інформації, спрощують багато аспектів управління, але водночас потребують уваги до технічних аспектів і захисту даних, щоб забезпечити безпеку та ефективність їх функціонування.

Розробка вебзастосунку для ОСББ є цінним в умовах сучасних реалій життя, особливо у випадку усунення ряду існуючих недоліків. Адже коли все більше функцій переходить в цифровий формат, з'являється попит на створення та обслуговування цифрових ресурсів.

РОЗДІЛ 2. ОГЛЯД ВИКОРИСТАНИХ ІНСТРУМЕНТІВ ТА ТЕХНОЛОГІЙ

2.1. Бекенд

2.1.1. Мова програмування PHP

Основні особливості PHP:

1. **Скриптова мова:** PHP є інтерпретованою мовою, тобто код виконується на сервері при кожному запиті, без необхідності компіляції перед запуском.
2. **Спрощений синтаксис:** PHP має схожий до C синтаксис, що робить його зрозумілим для багатьох програмістів і легким для вивчення для новачків.
3. **Підтримка багатьох платформ:** PHP підтримується практично на всіх платформах, що підтримують веб-сервери, таких як Windows, Linux, macOS та інші.
4. **Широкі можливості розширення:** В PHP існує велика кількість розширень і бібліотек для різноманітних задач, що значно розширює його функціональність.
5. **Вбудовані функції:** PHP має вбудовані функції для роботи з рядками, масивами, файлами, базами даних та іншими типовими завданнями веб-розробки.

Використання PHP:

1. **Веб-розробка:** Основне призначення PHP - це створення динамічних веб-сайтів і веб-додатків. Він використовується для генерації HTML-сторінок, обробки форм, керування сесіями користувачів та інтерактивної взаємодії з базами даних.
2. **Робота з базами даних:** PHP має вбудовані функції для роботи з реляційними базами даних, такими як MySQL, PostgreSQL, SQLite та іншими, що робить його ідеальним вибором для розробки веб-додатків з потребою в зберіганні та обробці даних.
3. **Консольні застосунки:** PHP також може використовуватися для створення консольних застосунків, таких як утиліти для обробки даних або автоматизації задач.
4. **Робота з API:** PHP часто використовується для створення API, які дозволяють іншим системам та додаткам взаємодіяти з вашим веб-додатком.

Недоліки PHP:

1. **Безпека:** PHP має історію проблем з безпекою, особливо коли використовується некоректно (наприклад, з SQL-ін'єкціями або XSS-атаками).
2. **Масштабованість:** Для дуже великих проєктів PHP може вимагати додаткових умов, таких як кешування або шкальовані бази даних, для забезпечення високої продуктивності.
3. **Організація коду:** Історично PHP не має такої суворої організації коду, як інші мови, що може призводити до менш чистого коду в більших проєктах.

PHP є потужним і широко використовуваним інструментом у веб-розробці, зокрема завдяки своїм простоті і доступності для новачків, а також величезній спільноті, що підтримує його розвиток та підтримку.

2.1.2. Фреймворк Laravel

Laravel є одним із найпопулярніших PHP-фреймворків, який відзначається своєю елегантністю, простотою та потужністю. Основні концепції Laravel включають:

1. **Система маршрутизації (Routing):** Laravel використовує просту та інтуїтивну систему маршрутизації, яка дозволяє визначати маршрути для додатків за допомогою простих та зрозумілих синтаксисів.
2. **Шаблони (Blade):** Blade є потужним шаблонізатором, що дозволяє використовувати шаблони з простим синтаксисом, включаючи умовні конструкції та цикли.
3. **Моделі (Eloquent ORM):** Eloquent ORM забезпечує простий та елегантний спосіб роботи з базами даних, використовуючи Active Record патерн. Це дозволяє здійснювати CRUD операції без написання складних SQL-запитів.
4. **Міграції (Migrations):** Міграції дозволяють версіонувати базу даних та легко керувати її змінами, забезпечуючи контроль за структурою бази даних через код.
5. **Middleware:** Middleware є механізмом для фільтрації HTTP-запитів, що дозволяє здійснювати аутентифікацію, авторизацію та інші операції на рівні запитів.
6. **Dependency Injection:** Laravel активно використовує принцип інверсії управління (IoC) та залежності (DI), що сприяє підвищенню гнучкості та тестованості коду.

Laravel слідує архітектурному паттерну MVC (Model-View-Controller), який розділяє додаток на три основні компоненти:

1. **Model (Модель):** Модель відповідає за бізнес-логіку та взаємодію з базою даних. У Laravel для цього використовується Eloquent ORM, який забезпечує просту роботу з базою даних.
2. **View (Вид):** Вид відповідає за відображення даних та інтерфейс користувача. Laravel використовує Blade шаблонізатор, що дозволяє створювати динамічні та повторно використовувані шаблони.
3. **Controller (Контролер):** Контролер відповідає за обробку запитів, взаємодію з моделями та повернення відповідних видів. Контролери забезпечують логіку обробки запитів та координацію між моделями та видами.

Laravel надає широкий спектр інструментів та бібліотек, що полегшують розробку вебзастосунків:

1. **Artisan Console:** Інструмент командного рядка для виконання різних задач, таких як генерація коду, управління міграціями, тестування та інше.
2. **Queue:** Система черг для обробки фонових задач, що дозволяє відкладати виконання завдань, таких як відправка email, обробка зображень тощо.
3. **Notification:** Система сповіщень для відправки повідомлень через різні канали, такі як email, SMS, Slack та інші.
4. **Events & Listeners:** Механізм подій та обробників, що дозволяє створювати асинхронні та реагуючі на події системи.
5. **Passport:** Пакет для реалізації аутентифікації на основі OAuth2, що забезпечує безпечний доступ до API.
6. **Cashier:** Інструмент для інтеграції з платіжними системами, такими як Stripe, що дозволяє легко реалізовувати підписки та одноразові платежі.
7. **Scout:** Пакет для реалізації пошуку по моделі з використанням популярних пошукових систем, таких як Algolia та Elasticsearch.
8. **Socialite:** Інструмент для аутентифікації через соціальні мережі, такі як Facebook, Google, Twitter та інші.

Laravel має багато переваг, що роблять його одним з найкращих виборів для розробки вебзастосунків:

1. **Простота та зручність:** Завдяки зрозумілому синтаксису та документації, Laravel є легким у вивченні та використанні, що зменшує час на освоєння та впровадження.

2. **Широкий функціонал:** Laravel надає безліч вбудованих інструментів та функцій, які покривають більшість потреб розробників.
3. **Активна спільнота:** Laravel має активну та підтримуючу спільноту, що забезпечує швидке вирішення проблем та доступ до великої кількості навчальних матеріалів.
4. **Безпека:** Вбудовані механізми захисту від поширених загроз, таких як SQL-ін'єкції, CSRF, XSS, забезпечують високий рівень безпеки вебзастосунків.
5. **Масштабованість:** Laravel легко масштабується, що дозволяє створювати як невеликі проекти, так і складні системи з великою кількістю користувачів.

Хоча Laravel має багато переваг, існують також деякі недоліки та виклики, пов'язані з його використанням:

1. **Продуктивність:** Laravel може бути менш продуктивним у порівнянні з деякими іншими фреймворками або мовами програмування, особливо для дуже високонавантажених систем.
2. **Складність налаштування:** Хоча більшість задач можна виконати легко, складні налаштування та кастомізація можуть вимагати глибокого знання фреймворку.
3. **Оновлення:** Підтримка нових версій Laravel може бути викликом, особливо для великих проектів з численними залежностями.

Як видно з опису, фреймворк дуже обширний і на його вивчення потрібно багато часу, що є одним із недоліків. Тому дану тему можна розглядати для подальшого вивчення та майбутніх робіт, адже весь функціонал за період написання роботи не був освоєний.

2.1.3. СУБД MySQL

MySQL - це одна з найпопулярніших систем управління реляційними базами даних (СУБД). Вона широко використовується в різних додатках, від невеликих веб-сайтів до великих корпоративних систем. MySQL є відкритим програмним забезпеченням і підтримується багатьма операційними системами, включаючи Windows, Linux і macOS.

Основні Компоненти MySQL

1. **MySQL Server:** Серверна частина MySQL, яка обробляє всі запити до бази даних. Сервер запускається як окремий процес і взаємодіє з клієнтами через мережеві протоколи.
2. **Клієнтські програми:** Програми, які взаємодіють з сервером MySQL для виконання запитів. Це можуть бути командний рядок, графічні інтерфейси, веб-додатки тощо.
3. **Конектор/Драйвери:** Програмні бібліотеки, які дозволяють клієнтським додаткам взаємодіяти з сервером MySQL через різні мови програмування.

Переваги MySQL

1. **Відкритий код:** MySQL є відкритим програмним забезпеченням, що означає, що він безкоштовний для використання і має великий внесок від спільноти розробників.
2. **Масштабованість:** MySQL добре масштабується від невеликих додатків до великих корпоративних систем з мільйонами записів і великою кількістю користувачів.
3. **Надійність і стабільність:** MySQL широко відомий своєю надійністю і стабільністю в роботі, що робить його вибором багатьох великих компаній і проектів.
4. **Швидкість і продуктивність:** MySQL оптимізований для швидкої обробки запитів і високої продуктивності при великому навантаженні.
5. **Гнучкість налаштувань:** MySQL дозволяє гнучко налаштовувати параметри конфігурації для досягнення найкращої продуктивності і відповідності конкретним вимогам додатку.

Недоліки MySQL

1. **Обмежена підтримка складних транзакцій:** MySQL не завжди підходить для додатків, які потребують дуже складних транзакцій.
2. **Обмежені можливості для аналізу даних:** Для великих аналітичних запитів інші СУБД можуть бути більш ефективними.
3. **Обмежена підтримка сторонніх розширень:** Хоча MySQL має багато розширень, деякі функціональності можуть вимагати сторонніх рішень, що може ускладнити налаштування і підтримку.

2.2 Фронтенд

2.2.1 HTML, CSS, JS

HTML (HyperText Markup Language)

HTML - це стандартна мова розмітки для створення веб-сторінок. Вона використовується для структурування і представлення контенту в Інтернеті. Основні компоненти HTML включають:

1. **Теги:** HTML складається з елементів, які позначаються тегами. Теги зазвичай мають відкриваючий та закриваючий елементи.
2. **Атрибути:** Теги можуть мати атрибути, які надають додаткову інформацію про елемент.

CSS (Cascading Style Sheets)

CSS - це мова стилів, яка використовується для опису вигляду і форматування HTML-документів. CSS дозволяє змінювати колір, шрифт, розташування та інші аспекти вигляду веб-сторінок. Основні компоненти CSS включають:

1. **Селектори:** В CSS використовується селектори для визначення елементів HTML, до яких застосовуються стилі.
2. **Властивості і значення:** Кожен стиль в CSS визначається парою властивість-значення.
3. **Каскадність і наслідування:** CSS має властивості каскадності, тобто стилі можуть накладатися один на одного, та наслідування, коли дочірні елементи наслідують стилі від батьківських.

JavaScript (JS)

JavaScript - це мова програмування, яка використовується для створення інтерактивних і динамічних веб-сторінок. Вона дозволяє змінювати контент сторінки, обробляти події, взаємодіяти з користувачем та багато іншого. Основні компоненти JavaScript включають:

1. **Функції:** Функції є основним способом організації коду в JavaScript. Вони дозволяють групувати код для виконання певних задач.
2. **Обробка подій:** JavaScript може реагувати на події, такі як кліки миші, введення тексту або завантаження сторінки. Це дозволяє створювати динамічні і інтерактивні елементи.

3. **Маніпуляція DOM:** JavaScript дозволяє динамічно змінювати структуру HTML-документа через об'єктну модель документа (DOM).

2.2.2 Фреймворк Bootstrap

Bootstrap - це популярний фреймворк для розробки адаптивних веб-додатків та веб-сайтів. Він був створений розробниками з Twitter і випущений як проект з відкритим вихідним кодом.

Основні характеристики Bootstrap

1. **Адаптивний дизайн:** Bootstrap забезпечує вбудовану підтримку адаптивного дизайну, що дозволяє веб-сайтам автоматично підлаштовуватися під різні розміри екрану та пристрої, від мобільних телефонів до настільних комп'ютерів.
2. **Система сітки:** Bootstrap використовує гнучку систему сітки на основі 12 колонок, яка дозволяє легко організовувати макети сторінок. Розробники можуть налаштовувати кількість колонок і їх ширину для різних розмірів екранів.
3. **Готові компоненти:** Bootstrap надає великий набір готових компонентів, таких як кнопки, форми, навігаційні панелі, модальні вікна, вкладки, сповіщення та багато інших, що значно прискорює процес розробки.
4. **Стилі та теми:** Bootstrap включає набір стандартних стилів для типографіки, таблиць, форм та інших елементів інтерфейсу, а також підтримує створення користувацьких тем.
5. **JavaScript компоненти:** Bootstrap також містить JavaScript компоненти для додавання інтерактивності, такі як каруселі, випадаючі меню, модальні вікна, спливаючі підказки та багато інших.

Переваги Bootstrap

1. **Простота використання:** Bootstrap легко освоїти і використовувати завдяки добре задокументованим прикладам і широкій підтримці спільноти.
2. **Прискорення розробки:** Використання готових компонентів та стилів значно скорочує час розробки веб-сайтів і веб-додатків.
3. **Уніфікований вигляд:** Bootstrap забезпечує єдиний стиль і дизайн для всіх елементів інтерфейсу, що робить веб-сайти більш професійними та естетично привабливими.

4. **Адаптивність:** Вбудована підтримка адаптивного дизайну дозволяє створювати сайти, які добре виглядають на будь-яких пристроях.
5. **Підтримка браузерів:** Bootstrap забезпечує широку підтримку сучасних браузерів, що гарантує коректне відображення веб-сторінок у більшості користувачів.

Недоліки Bootstrap

1. **Розмір і продуктивність:** Використання Bootstrap може додати до веб-сторінки значний обсяг коду, що може вплинути на продуктивність, особливо на мобільних пристроях.
2. **Обмежена унікальність:** Через широку популярність Bootstrap, багато веб-сайтів можуть виглядати схожими один на одного, якщо не вносити індивідуальні зміни в дизайн.
3. **Вивчення для кастомізації:** Хоча Bootstrap легко використовувати з коробки, його кастомізація може вимагати глибшого розуміння структури.

РОЗДІЛ 3. ОПИС РОЗРОБЛЕНОГО ПРОДУКТУ

3.1 ОСББ «Затишний дім»

Програмна реалізація відбувалася в середовищі PhpStorm. Додатково потрібно було лише встановити фреймворк Laravel.

Було налаштовано систему авторизації. При вході на сайт потрібно здійснити реєстрацію або вхід у систему (Рис. 3.1, 3.2):



Реєстрація:

Ім'я

Володимир

Електронна адреса

sorokavova9a@gmail.com

Телефон

0689453123

Адреса

м. Рівне, вул. Відінська 35а, кв. 5

Пароль

.....

Підтвердити пароль

8-3SixZACqAew!h

Зареєструватись

Рис. 3.1. Реєстрація користувача

Вхід:

Електронна адреса

sorokavova9a@gmail.com

Пароль

.....

Запам'ятати мене

Увійти

Рис. 3.2. Вікно входу в систему

Після успішної авторизації користувача переадресовує на сторінку «Про нас» (Рис. 3.3.):

ОСББ "Затишний дім"

Про нас

Вхід | Про нас | Документи | Мій кабінет | Оголошення | Новини | Послуги | Голосування | Контакти

Що таке ОСББ "Затишний дім"?

ОСББ "Затишний дім" - це об'єднання власників квартир в багатоповерховому будинку з метою забезпечення зручних та комфортних умов проживання. Ми працюємо для того, щоб забезпечити нашим мешканцям якісні послуги утримання будинку, здійснювати поточний ремонт та планувати капітальний ремонт, а також розвивати та зміцнювати спільноту мешканців.

Наші цілі та завдання

Основними цілями та завданнями ОСББ "Затишний дім" є:

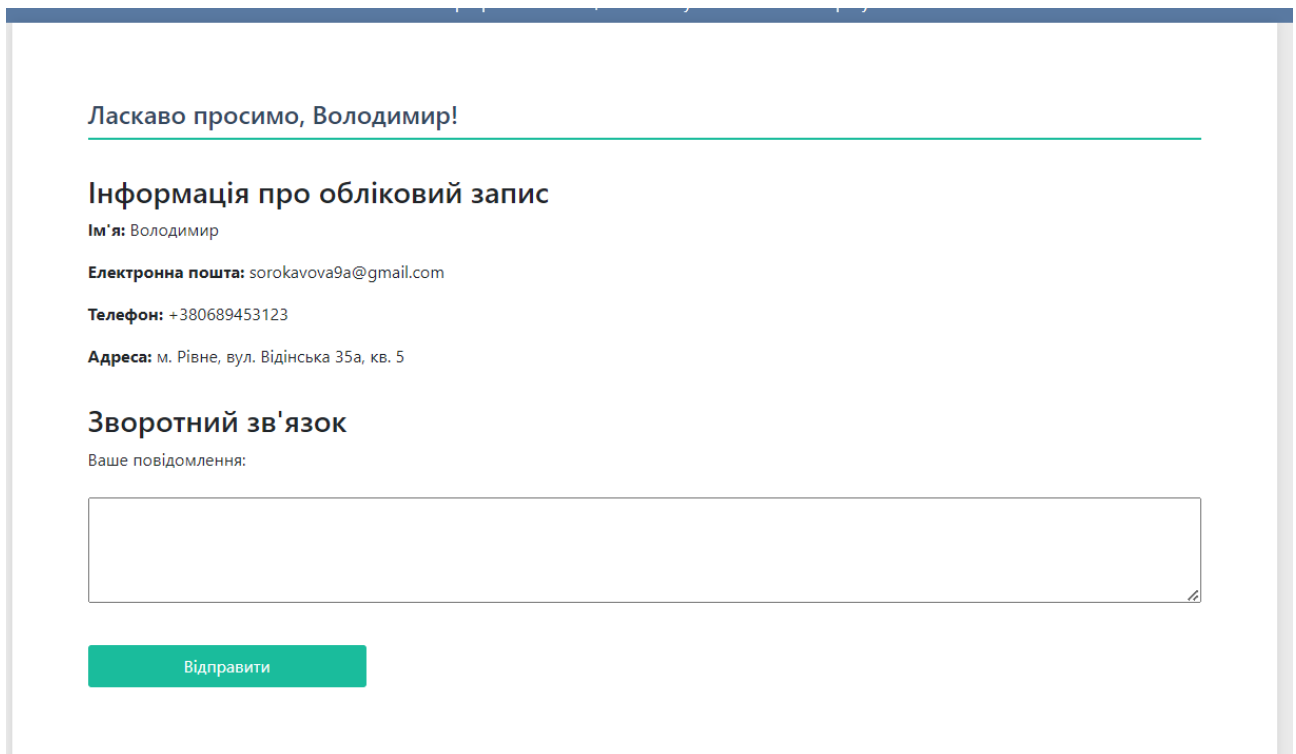
- Забезпечення якісних послуг з утримання будинку та прибудинкової території.
- Проведення поточного та планування капітального ремонту.
- Створення комфортних умов для проживання мешканців.
- Формування та зміцнення спільного відчуття власності та відповідальності серед мешканців.
- Розвиток та підтримка спільної інфраструктури будинку.

Наша команда

ОСББ "Затишний дім" складається з професійних та відповідальних працівників, які працюють на благо мешканців та розвиток нашого будинку. Наша команда включає у себе управителя будинку, технічний персонал, бухгалтера та інших спеціалістів, які забезпечують роботу ОСББ на високому рівні.

Рис. 3.3. Домашня сторінка

Є можливість зайти у свій кабінет (Рис. 3.4):



Ласкаво просимо, Володимир!

Інформація про обліковий запис

Ім'я: Володимир

Електронна пошта: sorokavova9a@gmail.com

Телефон: +380689453123

Адреса: м. Рівне, вул. Відінська 35а, кв. 5

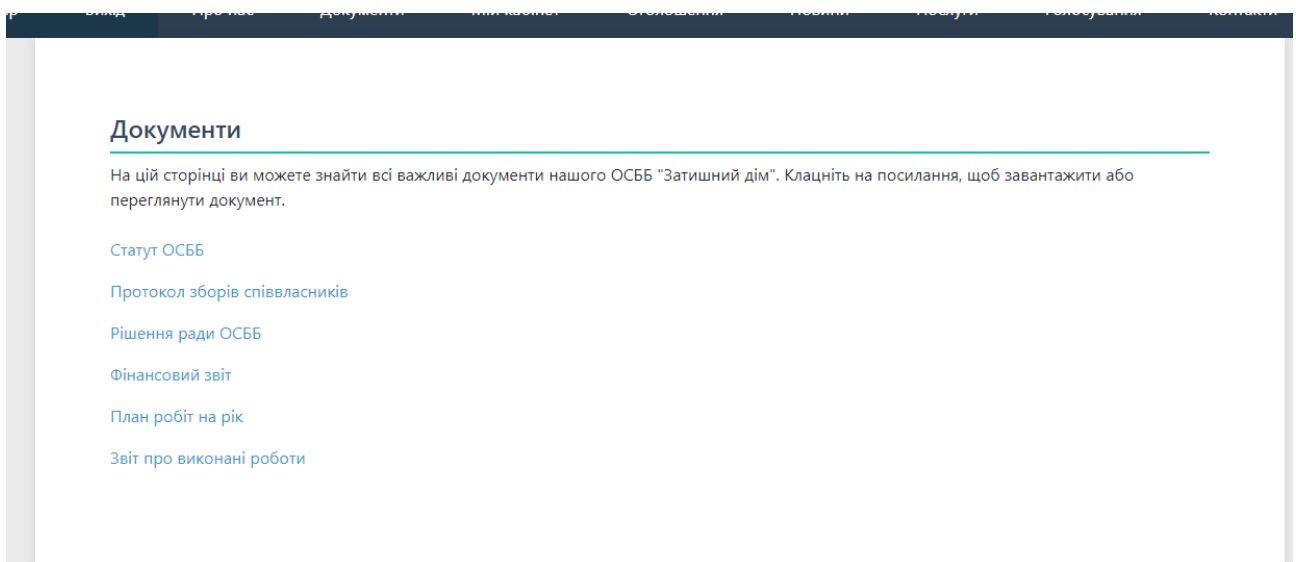
Зворотний зв'язок

Ваше повідомлення:

Відправити

Рис. 3.4. Сторінка з особистою інформацією

Можливість переглянути загальні документи ОСББ (Рис. 3.5.):



Документи

На цій сторінці ви можете знайти всі важливі документи нашого ОСББ "Затишний дім". Клацніть на посилання, щоб завантажити або переглянути документ.

- [Статут ОСББ](#)
- [Протокол зборів співвласників](#)
- [Рішення ради ОСББ](#)
- [Фінансовий звіт](#)
- [План робіт на рік](#)
- [Звіт про виконані роботи](#)

Рис. 3.5. Сторінка «Документи»

Сторінка з останніми новинами (Рис. 3.6, Рис. 3.7):



Рис. 3.6. Новини ОСББ

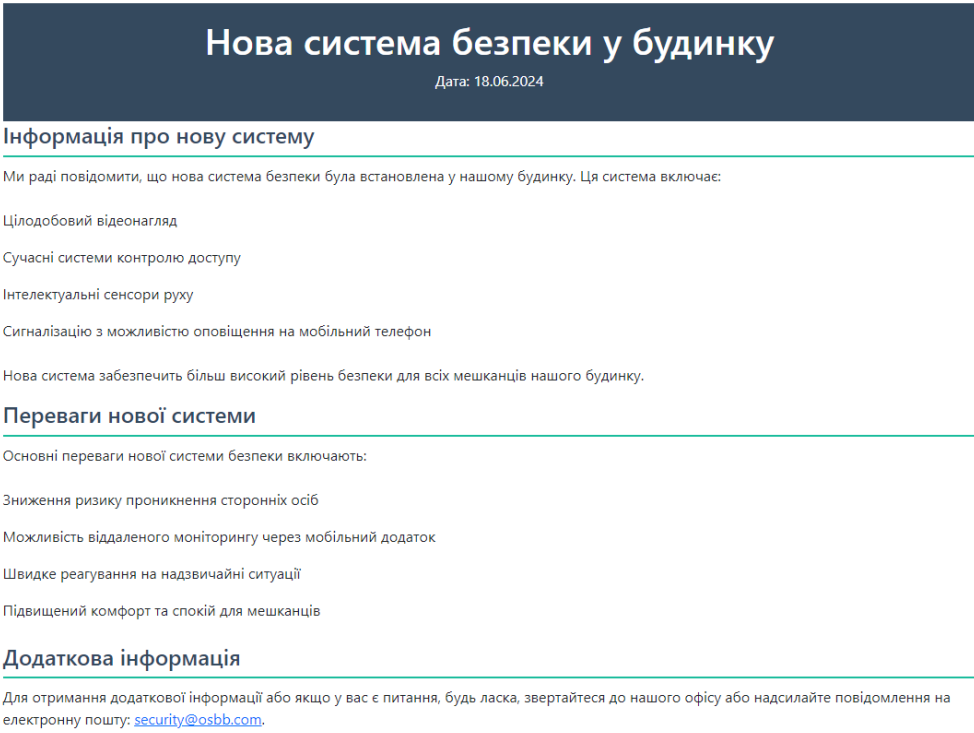


Рис. 3.7. Продовження новин

Активні голосування (Рис. 3.8.):

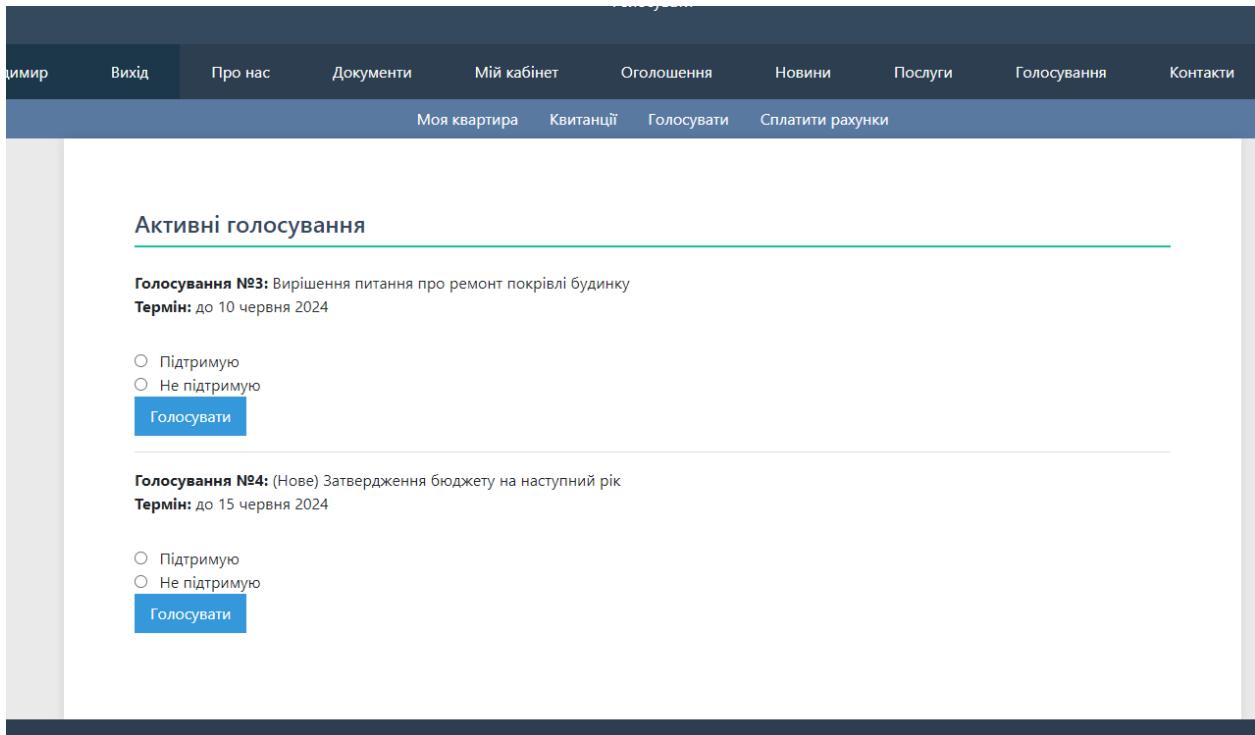


Рис. 3.8. Голосування

Контакти (Рис. 3.9.):

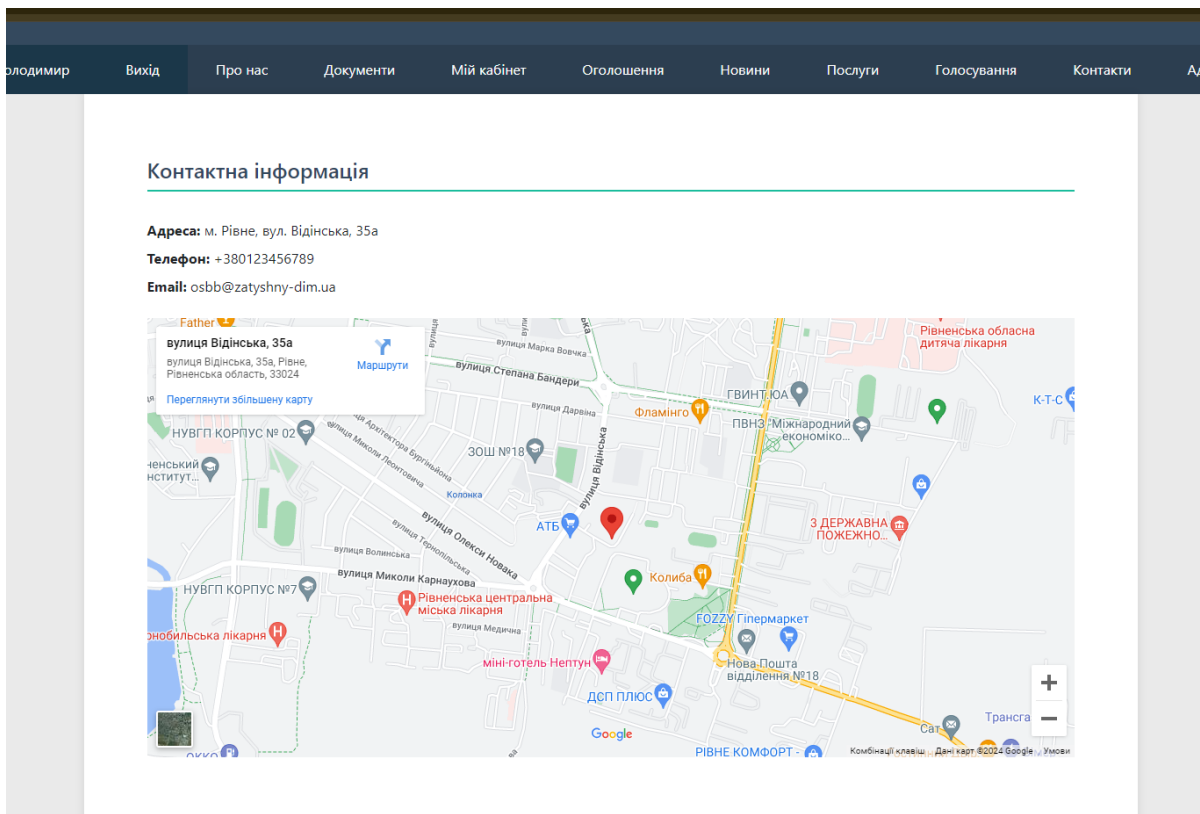


Рис. 3.9. Контактна інформація

Завантажити квитанції (Рис. 3.10):

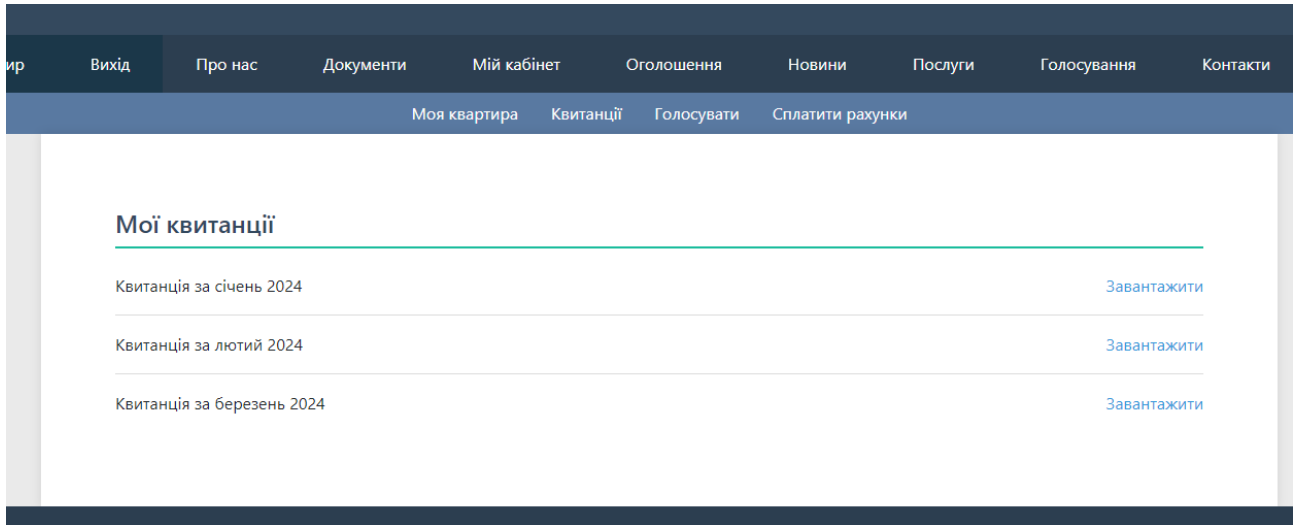


Рис. 3.10. Квитанції

ВИСНОВКИ

Аналіз результатів

Під час розробки вебзастосунку для ОСББ на основі Laravel були визначені та досягнуті ключові цілі проекту. Один з основних завдань включав у себе полегшення адміністративних процесів управління житловим будинком, зокрема управління фінансами, взаємодію з мешканцями та забезпечення доступу до важливої інформації. Вебзастосунок надав зручний інтерфейс для користувачів з можливістю відстеження фінансових витрат, внесення змін до розкладу обслуговування, а також для реєстрації та вирішення звернень мешканців.

Впровадження вебзастосунку дозволило збільшити прозорість управління фінансами та відкритість інформації, що стало ключовим аспектом для залучення довіри спільноти.

Ефективність використаних технологій

Використання Laravel, PHP та MySQL як основних технологій для розробки вебзастосунку дозволило досягти високої швидкості розробки, гнучкості та масштабованості системи. Laravel, зі своєю потужною архітектурою та великим спільнотним супроводом, забезпечив зручність у розробці та підтримці коду, що сприяло швидкому впровадженню нових функцій та виправленню помилок.

PHP, як основна мова програмування, та MySQL, як система управління базами даних, забезпечили стабільність роботи вебзастосунку та ефективне управління даними користувачів та фінансовими потоками. Використання сучасних технологій таких, як Bootstrap для веб-інтерфейсу та JavaScript для покращення взаємодії з користувачами, також виявилось критично важливим для забезпечення високоякісного користувацького досвіду.

Практичне значення та рекомендації

Розроблений вебзастосунок має значний практичний потенціал для впровадження в реальне управління ОСББ. Пропонуються наступні кроки для подальшого вдосконалення системи, включаючи розширення можливостей звітності та аналітики для забезпечення кращого управління фінансами, а також інтеграцію з платіжними системами для автоматизації процесів внесків мешканців.

Узагальнюючи, робота над проектом показала, що впровадження сучасних інформаційних технологій може значно покращити ефективність управління

ОСББ, забезпечити зручність для мешканців та сприяти розвитку громадських ініціатив у сфері житлової нерухомості.

Ці висновки дозволяють зрозуміти важливість розробки сучасних інформаційних рішень для підтримки житлових об'єднань та підкріплюють значення використання Laravel та інших сучасних технологій у реалізації подібних проектів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

Книги

1. Stauffer M. Laravel: Up and Running. A Framework for Building Modern PHP Apps. O'Reilly Media, 2016.

Технічні документації

2. Documentation Laravel: вебсайт. URL: <https://laravel.com/docs/11.x> (дата звернення: 17.06.2024).

3. Bootstrap. The most popular HTML, CSS, and JS library. вебсайт. URL: <https://getbootstrap.com/> (дата звернення: 17.06.2024).

4. Documentation PHP: вебсайт. URL: <https://www.php.net/docs.php> (дата звернення: 17.06.2024)

5. Documentation PhpStorm (IDE): вебсайт. URL: <https://www.jetbrains.com/phpstorm/resources/> (дата звернення: 18.06.2024)