

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ПРИРОДОКОРИСТУВАННЯ
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

другого (магістерського) рівня вищої освіти

на тему:

**«РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ
ОХОРОННИМИ ЗАСОБАМИ ВІДЕОСПОСТЕРЕЖЕННЯ»**

Виконав: здобувач групи ІТ-61
спеціальності 126 «Інформаційні системи та
технології»

_____ Силін О. І.

(прізвище та ініціали)

Керівник: _____ Пташник В. В.

(прізвище та ініціали)

Рецензент _____ Коробка С. В.

(прізвище та ініціали)

ДУБЛЯНИ-2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ПРИРОДОКОРИСТУВАННЯ
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Другий (магістерський) рівень вищої освіти
Спеціальність 126 «Інформаційні системи та технології»

ЗАТВЕРДЖУЮ
Завідувач кафедри

(підпис)

д.т.н., професор, Тригуба А. М.

(вч. звання, прізвище, ініціали)

“ ” _____ 2024 року

**З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Силін Олег Ігорович

(прізвище, ім'я, по батькові)

1. Тема роботи «Розробка інформаційної системи управління охоронними засобами відеоспостереження»

керівник роботи к. т. н., доцент., Пташник В. В.

(наук. ступінь, вч. звання, прізвище, ініціали)

затверджені наказом Львівського НУП від 28.04.2023 року № 133/к-с

2. Строк подання студентом роботи 15 січня 2024 року

3. Вихідні дані до роботи: технічна документація пристроїв; специфікація систем та пристроїв на основі технології інтернету речей; науково-технічна і довідкова література.

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

Вступ

1. Аналіз предметної області

2. Вибір програмних та апаратних засобів проектування

3. Проектування системи

4. Охорона праці та безпека в надзвичайних ситуаціях

5. Опис старту

Висновки

Список використаних джерел

5. Перелік графічного матеріалу

Графічний матеріал подається у вигляді презентації

6. Консультанти розділів

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата		Відмітка про виконання
		завдання видав	завдання прийняв	
1, 2, 3, 5	<i>Пташник В. В., к.т.н., доцент</i>			
4	<i>Городецький І. М., к.т.н., доцент</i>			

7. Дата видачі завдання 28 квітня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Відмітка про виконання
1	<i>Складання інженерної характеристики об'єкту проектування</i>	<i>28.04.2023 – 02.08.2023</i>	
2	<i>Розробка стартап проекту</i>	<i>02.08.2023 – 26.08.2023</i>	
3	<i>Розробка та програмування сайту системи відеоспостереження</i>	<i>27.08.2023 – 27.09.2023</i>	
4	<i>Проектування та програмування апаратної платформи Raspberry Pi</i>	<i>28.09.2023 – 02.11.2023</i>	
5	<i>Розгляд питань з охорони праці та безпеки у надзвичайних ситуаціях</i>	<i>03.11.2023 – 16.11.2023</i>	
6	<i>Завершення оформлення розрахунково-пояснювальної записки та презентаційного матеріалу</i>	<i>17.12.2023 – 31.12.2023</i>	
7	<i>Завершення роботи в цілому. Підготовка до захисту кваліфікаційної роботи</i>	<i>01.01.2024 – 22.01.2024</i>	

Здобувач

_____ *Силін О. І.*
 (підпис) (прізвище та ініціали)

Керівник роботи

_____ *Пташник В. В.*
 (підпис) (прізвище та ініціали)

Розробка інформаційної системи управління охоронними засобами відеоспостереження. Силін О. І. , Кафедра інформаційних технологій – Дубляни, Львівський національний університет природокористування, 2024. Кваліфікаційна робота: 57 сторінок текстової частини, 44 рисунки, 29 джерел літератури, 2 таблиці.

Для реалізації управління охоронними засобами розроблено сайт, де відбувається управління приладом відеоспостереження. Приладом відеоспостереження є спеціально налаштований Raspberry Pi. На сайті є кнопка «зробити фото», яка подає сигнал пристрою, і пристрій через Wi-Fi та Інтернет закидає фото на сайт.

Сайт виконаний у стилі стартапу. Реалізована можливість створення заявки на купівлю приладу користувачем, обробки та підтвердження її модератором. Після того як модератор активує пристрій, користувач має змогу управляти пристроєм і віддалено робити фотографії. Ці фотографії можна буде переглянути у кабінеті користувача. Код сайту і скрипти Raspberry Pi розміщені у Github.

Метою кваліфікаційної роботи є розробка інформаційної системи управління охоронними засобами відеоспостереження.

Об'єктом дослідження є алгоритми, програми та обладнання, необхідні для розробки інформаційної системи управління охоронними засобами відеоспостереження.

Предмет дослідження вивчає особливості проектування та використання інформаційної системи управління охоронними засобами відеоспостереження. Описано питання охорони праці.

Ключові слова: розумний будинок, відеоспостереження, інтернет речей, охоронні засоби.

ВСТУП

Одним із аспектів актуальності даної теми є зростаючий попит на розумний будинок і інтернет речей і загальна цифровізація всього. На ринку дуже багато різних приладів для розумного будинку, зокрема різні прилади з камерами. Через велику конкуренцію ціни на ці прилади знижуються, і розумний будинок стає масовим і доступнішим для користувачів. Метою кваліфікаційної роботи є розробка інформаційної системи управління охоронними засобами відеоспостереження на базі доступних на ринку комплектуючих таких як Raspberry Pi та камера до неї, а інформаційної системи управління буде розроблений сайт.

Для досягнення мети роботи необхідно виконати наступні завдання:

- Проаналізувати ринок розумних приладів для відеоспостереження
- Обрати обладнання
- Обрати технології для розробки
- Розробити архітектуру спілкування між сервером і приладом
- Розробити сайт
- Розробити скрипти для управління камерою на приладі

ПОСИЛАННЯ НА РЕСУРСИ

QR Code посилання на сайт:



Посилання на сайт:

<https://lnup-thesis-it-61-os.com>

Посилання на код у Github:

https://github.com/sirius256/thesis_raspberry_pi

<https://github.com/sirius256/thesis>

ЗМІСТ

ВСТУП.....	5
ПОСИЛАННЯ НА РЕСУРСИ	6
ЗМІСТ	7
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	9
1.1 Інтернет речей	9
1.2 Приклади готових рішень на ринку	10
РОЗДІЛ 2 ВИБІР ПРОГРАМНИХ ТА АПАРАТНИХ ЗАСОБІВ ПРОЕКТУВАННЯ	12
2.1 Платформа Raspberry Pi	12
2.2 Інструменти для розробки сайту	15
РОЗДІЛ 3 ПРОЕКТУВАННЯ СИСТЕМИ	22
3.1 Розробка архітектури системи	22
3.2 Розробка сайту	24
3.3 Розробка на Raspberry Pi.....	28
3.3.1 Збірка	28
3.3.2 Налаштування середовища і розробка скриптів	30
3.4 Підсумки розробки	33
РОЗДІЛ 4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ	45
4.1 Перелік небезпечних і шкідливих виробничих факторів діючих у робочій зоні	45
4.2. Технічні та організаційні заходи по зменшенню рівня впливу небезпечних та шкідливих виробничих факторів	46
4.3. Інструкція з охорони праці при обслуговуванні кліматичних системи ...	50
4.4 Забезпечення пожежної та вибухової безпеки при обслуговуванні кліматичних систем.....	51
РОЗДІЛ 5 ОПИС СТАРТАПУ	52

	8
5.1 Комерція.....	52
5.2 Перспективи і майбутній розвиток.....	52
ВИСНОВКИ	55
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	56

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Інтернет речей

Інтернет речей (Internet of Things, IoT) – концепція мережі, що складається із взаємозв'язаних фізичних пристроїв. Вони зв'язані програмним забезпеченням, що дозволяє обмінюватись між ними даними і керувати приладами. Це автоматизує багато рутинних дій людини чи видає якусь інформацію з датчиків і обробляє їх.

Через велику кількість приладів і інтерфейсів передачі даних, є проблема сумісності цих приладів і програмного забезпечення. І щоб вирішити цю проблему, придумали стандартизовані інтерфейси, щоб покращити сумісність. І великі компанії такі як Amazon, Google систему управління з всіма приладами з одного програмного інтерфейсу. Це в рази покращує сумісність і юзер експірієнс. Також ці хаби можуть керуватись голосом за допомогою Алекси чи Google асистента та інших. На рис. 1.1 зображено управління всіма приладами з одного гаджета.



Рисунок 1.1 – Інтернет речей (Internet of Things, IoT)

1.2 Приклади готових рішень на ринку

На ринку є велика кількість камер, і приладів з камерами, які можна підключити до телефону через Wi-Fi, інтернет, блютуз чи кабель. До прикладу якщо глянути в інтернет магазин розетка, то можна побачити близько 50-ти тисяч різних пропозицій від різних виробників. На рис. 1.1 зображено магазин розетка, прилади відеоспостереження які можна купити.

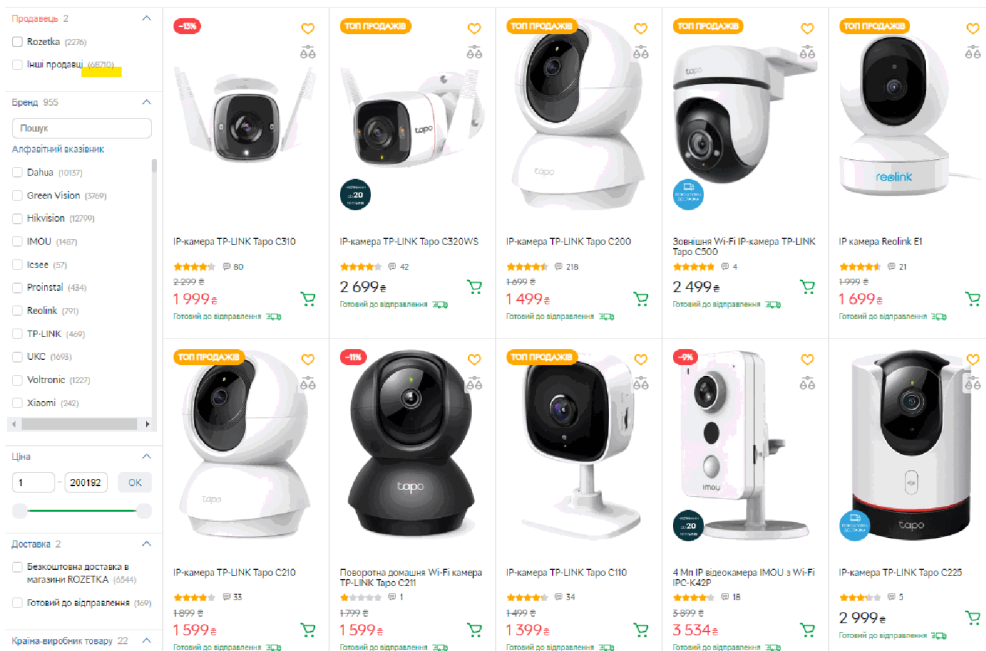


Рисунок 1.2 – Магазин розетка, прилади відеоспостереження

Для масштабних систем відеоспостереження використовуються спеціальні сервери та системи збереження для обробки та збереження великого обсягу даних відеоспостереження. Деякі відеокамери можуть бути підключені до системи "хмарного" зберігання, що дозволяє дистанційно зберігати та отримувати доступ до відеоматеріалів через Інтернет.

Одним з прикладом таких є продукти компанії Аїах. Вони надають повний спектр приладів для різних рішень поставлених задач: виявлення пожежі, захист

від вторгнення відеоспостереження, запобігання потопу, комфорт та продуктивність. На рис. 1.2 зображено сайт компанії Ажах.

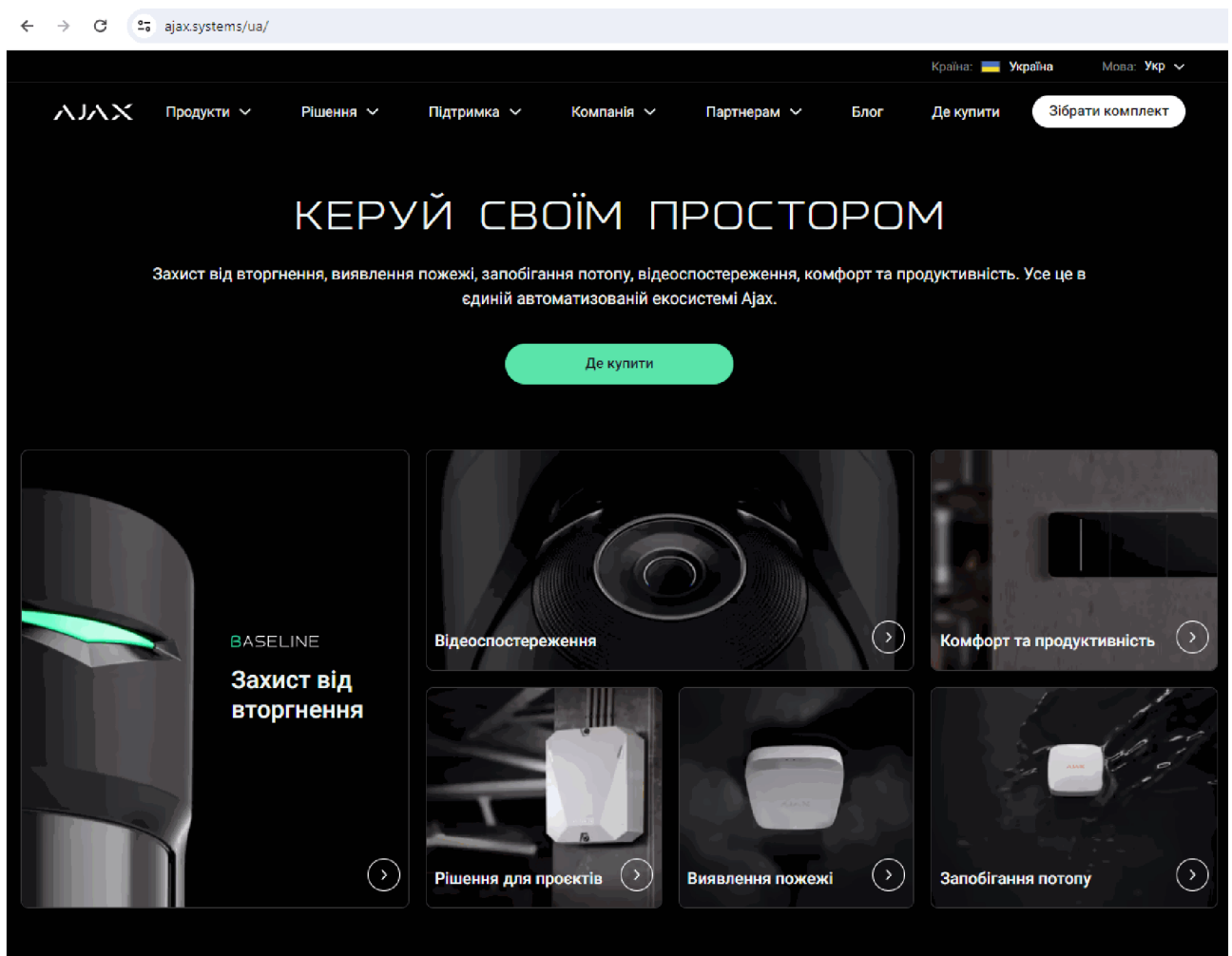


Рисунок 1.3 – Продукти компанії Ажах

Прилади відеоспостереження використовуються в різних сферах, таких як безпека громадських місць, бізнес-об'єкти, виробництва, транспортні системи, інфраструктура та житлові будівлі. Застосування відеоспостереження допомагає виявляти і вирішувати проблеми безпеки, сприяє превентивним заходам та надає можливість реагувати на події в реальному часі.

РОЗДІЛ 2 ВИБІР ПРОГРАМНИХ ТА АПАРАТНИХ ЗАСОБІВ ПРОЕКТУВАННЯ

2.1 Платформа Raspberry Pi

Raspberry Pi (укр. – малиновий пиріг) – одноплатний комп'ютер, розроблений британським фондом Raspberry Pi Foundation. Для є їхня юнікс-подібна операційна система Raspberry Pi OS. До Raspberry Pi, є багато приладів, датчиків, які підтримуються платформою «з під коробки». Також є багато документації, уроків, статей, написаного коду і велике ком'юніті. Компанія постійно розвиває свої продукти і розширює асортимент. Через ці вище написані причини я вибрав платформу Raspberry.

А саме вибрав Raspberry PI 4 Model B і камеру Raspberry Pi V3 12 MP. Цей прилад оснащений 4-ох ядерним процесором на частоті 1.5 ГГц, 8 GB оперативної пам'яті, Wi-Fi модулем, блютузом, 2 портами HDMI, 2 портами usb 2.0 і 2 портами USB 3. На рис. 2.1 зображено камеру Raspberry Pi V3 12 MP. На рис. 2.2 доступні порти, інтерфейси, виходи входи у Raspberry Pi. На рис. 2.3 зображено піддослідний екземпляр Raspberry PI 4 Model B.

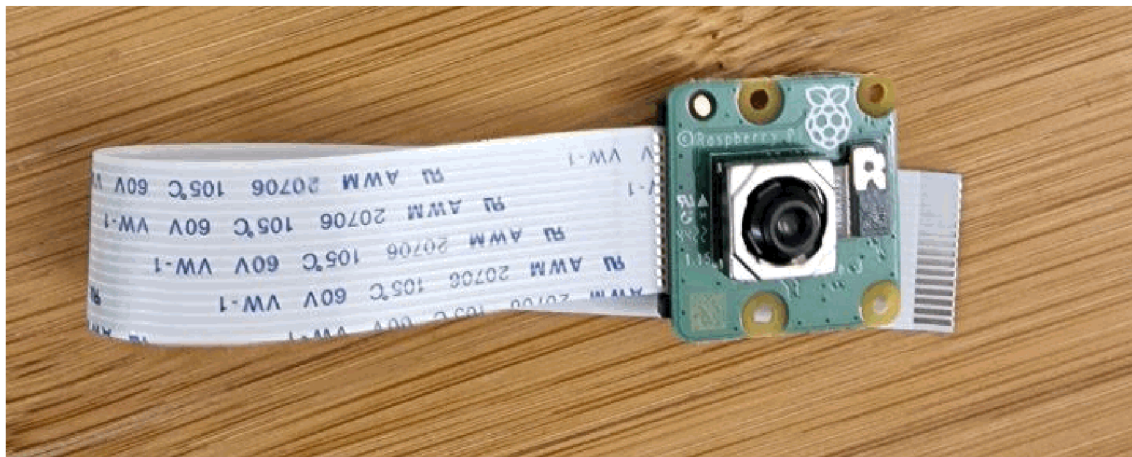


Рисунок 2.1 – Камера Raspberry Pi V3 12 MP

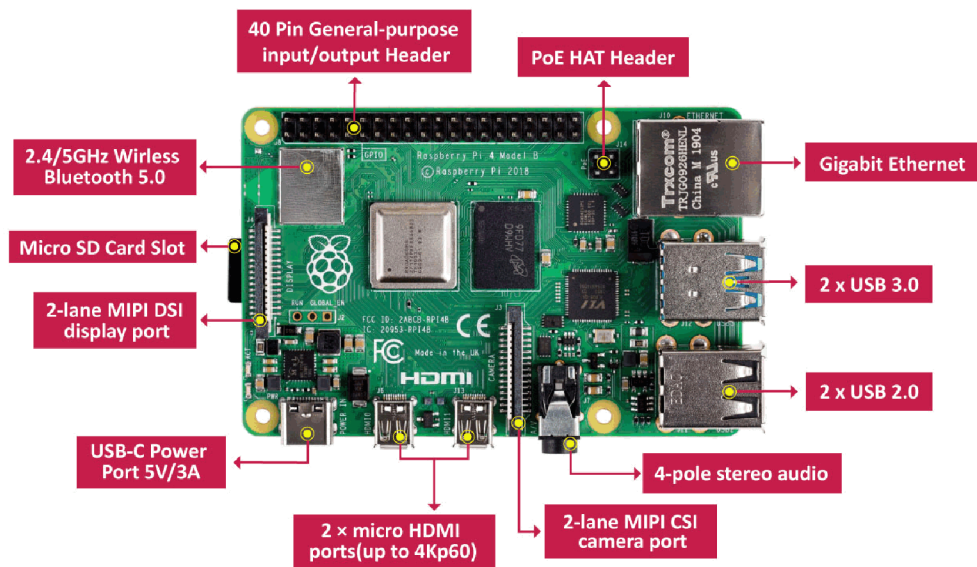


Рисунок 2.2 – Raspberry Pi: порти, інтерфейси, виходи входи

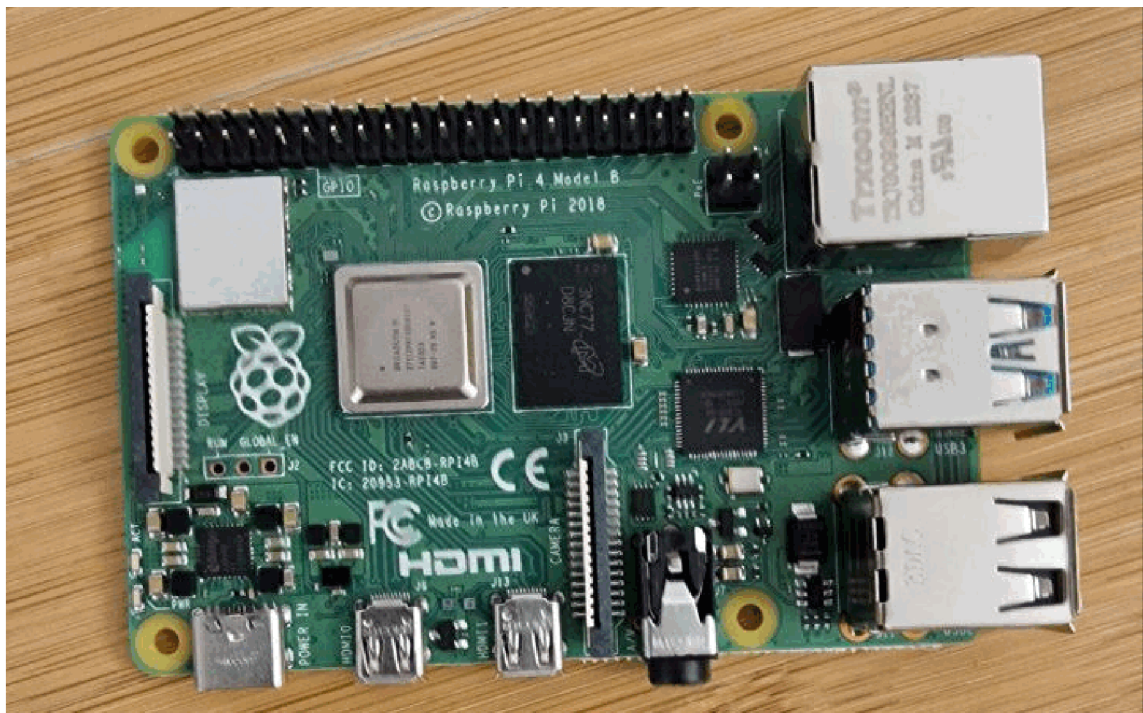


Рисунок 2.3 – Raspberry Pi 4 Model B 8 GB

Живиться пристрій через Type-C порт. Можна жити від павербанку. Для зручності я так і використовував павербанк для живлення, це збільшує мобільність пристрою. Хоч цей процесор є енергоефективний і його теплопакет

є дуже малим, але все рівно довелось придбати радіатор. Вибрав корпус-радіатор, він прижимається зверху і знизу за допомогою болтів, і в комплекті були термопрокладки, для щільного прилягання металу до чіпів пристрою. На рис. 2.4 зображено Корпус-радіатор для Raspberry Pi 4 Model B з алюмінієвого Сплаву.



Рисунок 2.4 – Корпус-радіатор для Raspberry Pi 4 Model B з алюмінієвого Сплаву

Для встановлення операційної системи, ще потрібно MicroSD. Обрано було карту розміром 128 гб. Це більше ніж потрібно, але ціна на 1 Гб пам'яті була кращою ніж моделі з меншим об'ємом.

Ще в комплекті був кабель для живлення з кнопкою вимикання/включення Type-C => USB. Це дуже зручно, бо не потрібно постійно витягувати і засовувати кабель живлення.

Щоб приєднати пристрій до екрану, потрібно перехідник HDMI => Micro HDMI, або зразу відповідний кабель. Я придбав зразу готовий кабель, оскільки цільний кабель краще ніж кабель з перехідником. Більш надійний варіант. Цей кабель можна побачити на рис. 2.5. На ньому є навіть є логотип Raspberry Pi. Я

не впевнений, що це оригінальний кабель. Є 2 порти для виводу екрану, Raspberry Pi підтримує 4к екрани.



Рисунок 2.5 – кабель HDMI => Micro HDMI

Для встановлення операційної системи (Raspberry Pi OS), потрібно встановити програму Raspberry Pi Imager. Яка виконає збірку та конфігурацію образу і закине його на MicroSD картку.

Для зберігання проекту використовується Github. Лінк на репозиторій з кодом сайту: https://github.com/sirius256/thesis_raspberry_pi

2.2 Інструменти для розробки сайту

Мова програмування. На серверній частині сайту була обрана мова програмування PHP 8.1 через такі плюси:

- легкість розробки;

- велике ком'юніті;
- постійне вдосконалення;
- наявність готових рішень, бібліотек, фреймворків;
- хороша документація;
- велика швидкість розробки;
- покриває нашу задачу.

Ця мова програмування є високорівневою, інтерпретованою, яка використовується переважно для розробки веб-застосунків та динамічних веб-сайтів. PHP створений для вбудованого в HTML код та використовується для генерації веб-сторінок на сервері. Одна з головних переваг PHP полягає в тому, що вона легко інтегрується з HTML, що полегшує створення динамічних веб-сайтів та обробку форм.

Мова має велику та активну спільноту розробників, а також широкую документацію. Вона підтримує різні бази даних, включаючи MySQL, PostgreSQL та інші, що робить її популярним вибором для веб-розробки. PHP має вбудовані функції для роботи з файлами, роботи з URL, роботи з зображеннями, обробки форм та багато іншого.

Однак, PHP також має свої недоліки, зокрема це історичні проблеми з безпекою та архітектурою мови. Незважаючи на це, PHP залишається однією з найпопулярніших мов програмування для серверної розробки веб-додатків та знаходить широке застосування в індустрії веб-технологій.

На рис 2.6 зображено порівняння швидкодії різних версій PHP.

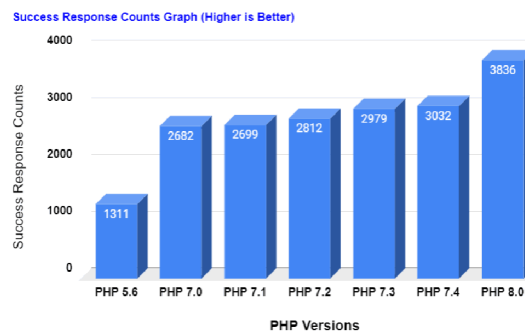


Рисунок 2.6 – порівняння швидкодії різних версій PHP

Також був обраний фреймворк Laravel. Оскільки він легкий в розробці, гнучкий, хороша документація, доволі швидкий. Постійно розвивається. Використовується Laravel 10. На рис 2.8 зображено порівняння популярності Laravel vs Symfony.

Це високорівневий веб-фреймворк, написаний на мові програмування PHP, який зосереджений на роботі з розробкою веб-застосунків та забезпечує ефективні та елегантні рішення для завдань рутинної розробки. Розробка Laravel була започаткована Тейлором Отвеллом і швидко стала однією з найпопулярніших та широко використовуваних платформ для створення сучасних веб-застосунків. Однією з ключових особливостей Laravel є елегантний та зрозумілий синтаксис, який полегшує роботу розробникам. Фреймворк включає в себе багато вбудованих функцій, таких як система маршрутизації, ORM (Eloquent), система шаблонів Blade, та інші. Laravel також пропонує потужні можливості для роботи з базами даних, міграцій, аутентифікації та авторизації. Ще однією перевагою Laravel є його активна та велика спільнота розробників, яка сприяє швидкому вирішенню проблем та поширенню знань. Фреймворк регулярно оновлюється, що дозволяє використовувати найновіші технології та підходи в розробці веб-застосунків.

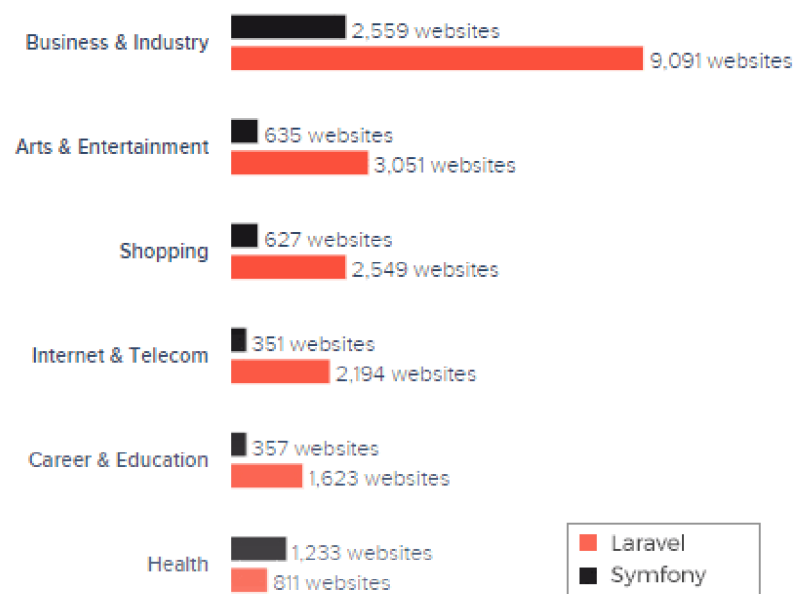


Рисунок 2.7 – порівняння популярності Laravel vs Symfony

Фронтенд частина. Буде використовуватися HTML, CSS, JavaScript, Blade, Bootstrap і збірщик Vite. Нище буде опис кожної з них.

HTML (HyperText Markup Language) є основною мовою розмітки для створення та відображення веб-сторінок. Визначаючи структуру та зміст документа, HTML використовує теги для опису різних елементів, таких як заголовки, параграфи, зображення, посилання та інші. Завдяки HTML, веб-браузери можуть інтерпретувати та відображати сторінки в Інтернеті так, як їх бачить користувач. HTML є основою веб-розробки і часто комбінується з CSS (Cascading Style Sheets) та JavaScript для створення динамічних та стильових веб-сайтів.

CSS (Cascading Style Sheets) є мовою стилізації, яка використовується в веб-розробці для задання зовнішнього вигляду веб-сторінок, написаних на HTML або XML. CSS дозволяє розробникам змінювати відображення елементів сторінок, контролюючи такі аспекти, як кольори, шрифти, відступи, розміри та розташування елементів. За допомогою CSS можна створювати привабливі та структуровані веб-сайти, а також забезпечувати їх адаптацію до різних пристроїв і розмірів екранів. CSS працює в парі з HTML, де визначається структура сторінки, щоб надати веб-документу зовнішній вигляд та стиль.

JavaScript є високорівневою, інтерпретованою мовою програмування, яка використовується для створення динамічних та інтерактивних веб-сайтів. Вона використовується в основному для взаємодії з користувачем на стороні клієнта, дозволяючи змінювати вміст та структуру веб-сторінок без перезавантаження сторінки. JavaScript також широко використовується для розробки різноманітних веб-додатків, включаючи ігри, анімацію, обробку форм, взаємодію з API та багато іншого. Вона є ключовою технологією веб-розробки, сприяючи створенню сучасних та інноваційних веб-застосунків.

Blade – це шаблонізатор, вбудований в веб-фреймворк Laravel, який дозволяє розробникам легко створювати та управляти шаблонами HTML у їх веб-додатках. Blade надає простий та елегантний синтаксис, який дозволяє вставляти змінні, умови, цикли та інші елементи в шаблони, що полегшує роботу

з представленнями в Laravel-додатках. Один із основних принципів Blade - це спрощення вставки PHP-коду в HTML, забезпечуючи при цьому зручність та читабельність коду. Шаблонізатор Blade став популярним серед розробників завдяки своїй ефективності та зручності використання при створенні різноманітних інтерфейсів веб-застосунків на базі Laravel.

Bootstrap - це популярний та відкритий веб-фреймворк, розроблений командою Twitter, який надає широкий набір готових компонентів та інструментів для швидкої та ефективної розробки адаптивних та стильних веб-інтерфейсів. Основний акцент Bootstrap робить на розробці зручних та красивих веб-сайтів, забезпечуючи консистентний дизайн та можливості адаптації до різних пристроїв і розмірів екранів.

Vite – це інноваційний і швидкий інструмент для розробки веб-додатків, який використовується для створення сучасних та продуктивних веб-інтерфейсів. Відомий своєю високою швидкістю, Vite використовує концепцію "блискавичного" старту, що дозволяє розробникам швидко запускати та реалізовувати зміни в коді без помітного часу перекомпіляції. Заснований на сучасних технологіях, таких як ES Module (ESM) та Rollup, Vite пропонує високоефективне середовище розробки для проектів, забезпечуючи оптимізовану швидкість завантаження та робочий процес, що дозволяє зосередитися на створенні якісного веб-додатку.

База даних – MySQL. Це реляційна система управління базами даних (RDBMS), яка використовується для зберігання та управління даними у веб-додатках та інших програмах. Вона надає ефективний механізм зберігання та операцій з базами даних, підтримує мову структурованого запиту SQL (Structured Query Language) і взаємодіє з різноманітними програмами та серверами. MySQL використовується великою кількістю веб-розробників та організацій завдяки своїй надійності, продуктивності та відкритому джерелу. Вона дозволяє створювати та управляти базами даних для різних застосувань, включаючи блоги, електронні комерції, соціальні мережі та багато іншого.

Система контролю версій – Git. Це розподілена система керування версіями, яка дозволяє розробникам ефективно відслідковувати та керувати змінами в коді проектів. Розроблена Лінусом Торвальдсом, Git надає можливість працювати з репозиторіями, що зберігають версії проектів, та здійснювати операції, такі як коміти, гілки, злиття та відкати. Його гнучкість, швидкість та здатність до роботи як локально, так і на віддалених серверах роблять Git стандартом у сфері керування версіями для розробників усього світу. Лінк на репозиторій з кодом сайту <https://github.com/sirius256/thesis>

Де купити сервер. Для розгортання сайту використовується сервер, орендований у на компанії hostinger. Посилання на їх сайт: <https://www.hostinger.com.ua/> Доменне ім'я **lnup-thesis-it-61-os.com** теж там куплене.

Локальний сервер. Вибрано OpenServer, це локальний сервер для розробки веб-додатків, який надає зручне та легке середовище для запуску та тестування веб-сайтів на локальному комп'ютері. Зокрема популярний серед веб-розробників, OpenServer включає в себе веб-сервер Apache, базу даних MySQL, а також підтримку PHP та других технологій. Завдяки його простоті встановлення та конфігурації, він дозволяє розробникам легко налаштовувати локальне середовище для виконання веб-додатків перед їх публікацією на сервері.

Додаткові інструменти:

1) Сайт для розробки діаграм <https://app.diagrams.net/>. Він дозволяє створення діаграм, схем, графіків та інших векторних зображень. Заснований на технології відкритих вихідних кодів, цей інструмент дозволяє користувачам створювати різноманітні візуальні представлення для проектів, бізнес-процесів, архітектурних схем, організаційних діаграм та багато іншого. Зручний інтерфейс, багатий функціонал та можливість працювати онлайн без необхідності встановлення програмного забезпечення роблять app.diagrams.net популярним вибором для тих, хто шукає простий та ефективний інструмент для створення візуальних матеріалів.

2) Postman – це потужний інструмент для тестування та розробки API (інтерфейсів програмування застосунків). Відомий своєю простотою використання та багатофункціональністю, Postman надає інтерфейс для створення, тестування та документування HTTP-запитів. Розробники використовують Postman для ефективного взаємодії з API, перевірки правильності відповідей сервера, автоматизації тестування та зручного спілкування з різними сервісами та ресурсами в мережі. Його інтуїтивний інтерфейс та набір корисних функцій роблять його необхідним інструментом для розробників, які працюють з веб-сервісами.

3) Font Awesome – це відома бібліотека іконок, яка дозволяє веб-розробникам легко використовувати векторні іконки у своїх проектах. Заснована на CSS та шрифтах, Font Awesome надає великий набір стилізованих ікон, які можуть бути вставлені безпосередньо в HTML-код. Завдяки цій бібліотеці, розробники можуть ефективно вдосконалювати зовнішній вигляд своїх веб-сайтів або додавати іконки до кнопок, меню, форм та інших елементів інтерфейсу. Font Awesome також підтримує можливість використання іконок у різних розмірах та кольорах, роблячи його популярним вибором для стилізації веб-додатків.

РОЗДІЛ 3 ПРОЕКТУВАННЯ СИСТЕМИ

3.1 Розробка архітектури системи

Сайт буде спілкуватися з приладом через API. Raspberry Pi містить в собі модуль Wi-Fi, що дає змогу використовувати API веб сайту. Кожен прилад має певний унікальний хеш, який використовує як ключ до API сайту. І по цьому ж хешу на стороні сайту можна визначити власника приладу і чи є запити від користувача на створення фото.

Прилад надсилає запит на get actions API, і якщо у відповіді сервера, є задача зробити фото, тоді прилад робить фото і надсилає це фото на сайт через send photo API. На рис 3.1 зображено архітектуру спілкування між сайтом і сервером.

REST API є архітектурним стилем для створення веб-сервісів, який дозволяє ефективно та стандартизовано взаємодіяти з різними компонентами програмного забезпечення через мережу. REST API використовує HTTP методи, такі як GET, POST, PUT та DELETE.



Рисунок 3.1 – Архітектура системи

Розглянемо детально розроблені запити до API. Ці запити візуалізовані через Postman, їх можна побачити на рис. 3.2 і 3.3.

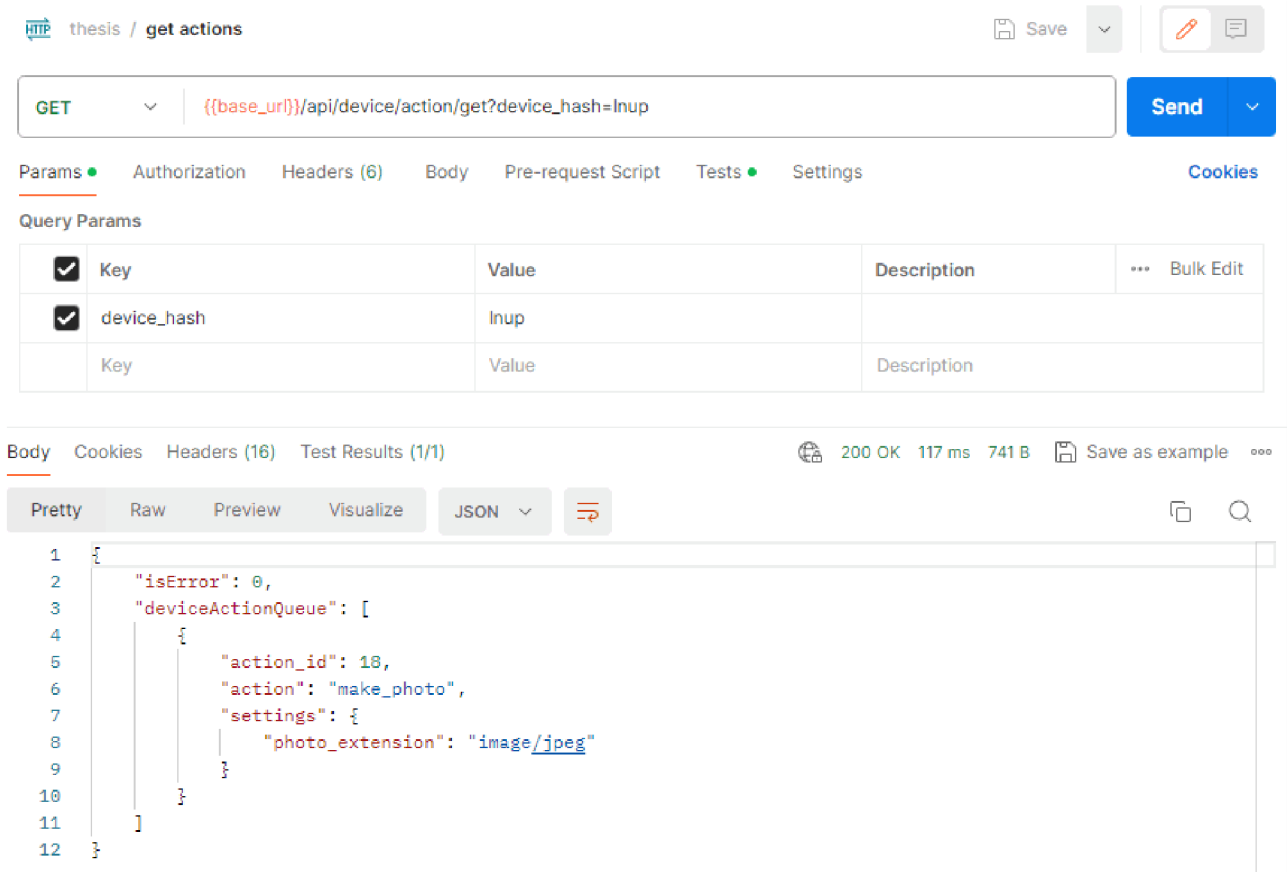


Рисунок 3.2– Запит на get actions API

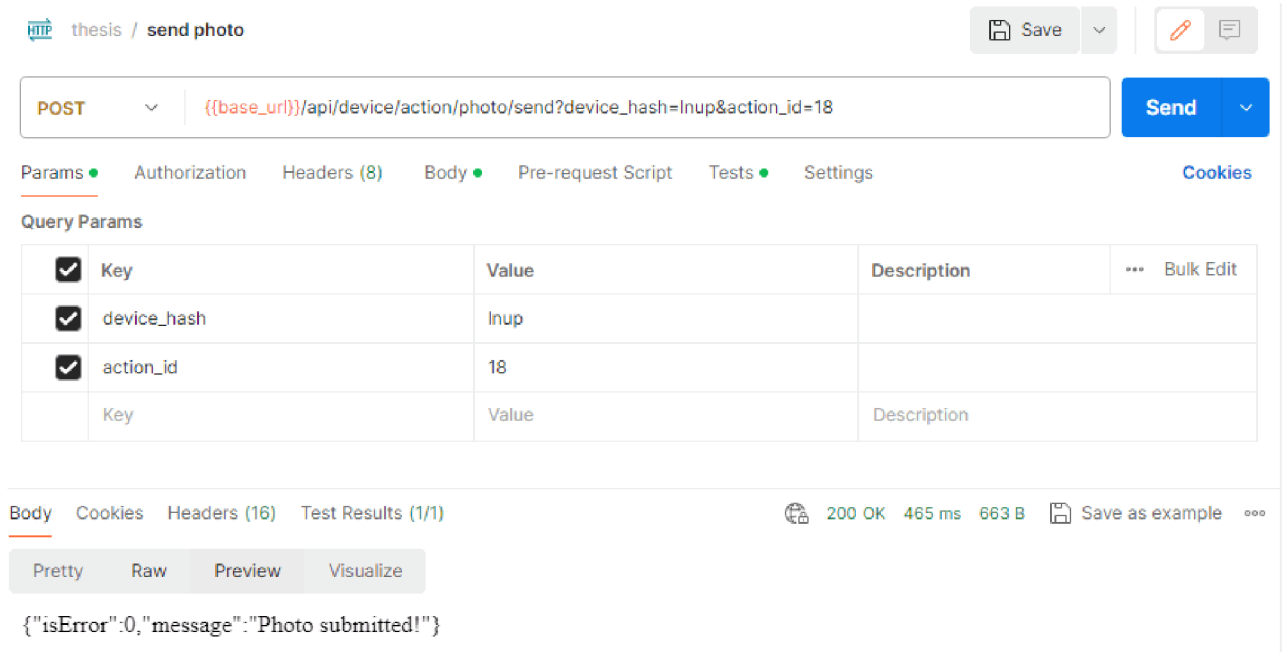


Рисунок 3.3 – Запит на send photo API

В body запиту **send photo API** передається фото. В обидвох запитах «device_hash» є по суті своїй ключем до цього API. Якщо хеш приладу не вірний, чи прилад не активований, API верне помилку.

Експортована колекція API запитів у файл можна скачати з репозиторію у Github: https://github.com/sirius256/thesis_raspberry_pi

3.2 Розробка сайту

Розробка почалась з формування технічного завдання, тобто опис функціоналу який потрібно буде розробити. Максимально описати все що буде на сайті, весь функціонал і логіку. Звісно все важко передбачити і під час самої розробки технічне завдання оновлюється і удосконалюється.

Далі був вибір інструментів розробки. Що було описано в розділі вище. Коротко що вибрали: серверну мову PHP 8.1, MySQL – як базу даних, Laravel – фреймворк.

Далі розробка архітектури бази даних. Це назви таблицок і їх поля, зв'язки між таблицями. Таблички і їх поля можна побачити на рис. 3.4.

Перелік основних сутностей в базі даних:

- Сутність «User» (користувач) представляє таблиця «users».
- Сутність «Device» (прилад) представляє таблиця «devices».
- Сутність «DeviceSetting» (налаштування приладу) представляє таблиця «device_settings».
- Сутність «DeviceModel» (модель приладу) представляє таблиця «device_models».
- Сутність «Status» (статус) представляє таблиця «statuses».
- Сутність «DeviceGallary» (галерея пристрою) представляє таблиця «device_galleries».

- Сутність «DeviceGallaryImage» (картинка галереї пристрою) представляє таблиця «device_gallery_images».
- Сутність «Order» (замовлення) представляє таблиця «orders».
- Сутність «DeviceActionQueue» (список завдань для приладу) представляє таблиця «device_action_queue».
- Сутність «OrderLog» (нотатка замовлення) представляє таблиця «order_logs».

Кожна сутність «User» (користувач), може містити безмежну кількість сутностей «Order» (замовлення), для цього таблиці «orders» є поле «user_id». Це тип зв'язку «One-to-many»(один до багатьох).

Кожна сутність «User» (користувач), може містити безмежну кількість сутностей «Device»(прилад), для цього таблиці «devices» є поле «user_id». Це тип зв'язку «One-to-many» (один до багатьох).

Кожна сутність «Device» (прилад), може містити безмежну кількість сутностей «DeviceSetting» (налаштування приладу), для цього таблиці «device_settings» є поле «device_id». Це тип зв'язку «One-to-many» (один до багатьох).

Кожна сутність «Device» (прилад), може містити безмежну кількість сутностей «DeviceActionQueue» (список завдань для приладу), для цього таблиці «device_action_queue» є поле «device_id». Це тип зв'язку «One-to-many» (один до багатьох).

Кожна сутність «Device» (прилад), може містити безмежну кількість сутностей «DeviceGallary» (галерея пристрою), для цього таблиці «device_galleries» є поле «device_id». Це тип зв'язку «One-to-many» (один до багатьох).

Кожна «DeviceGallary»(галерея пристрою), може містити безмежну кількість сутностей «DeviceGallaryImage»(картинка галереї пристрою), для цього таблиці «device_gallery_images» є поле «gallery_id». Це тип зв'язку «One-to-many» (один до багатьох).

Кожна сутність «Order» (замовлення), може містити безмежну кількість сутностей «OrderLog» (нотатка замовлення), для цього таблиці «order_logs» є поле «order_id». Це тип зв'язку «One-to-many» (один до багатьох).

Table Name	Fields
devices	id: bigint(20) unsigned, name: varchar(255), model_id: varchar(255), hash: varchar(255), is_active: tinyint(1), owner_user_id: int(11), created_at: timestamp, updated_at: timestamp
device_settings	id: bigint(20) unsigned, name: varchar(60), label: varchar(255), value: text, description: text, device_id: int(11), created_at: timestamp, updated_at: timestamp
device_models	id: bigint(20) unsigned, title: text, name: varchar(255), image_url: varchar(255), description: text, created_at: timestamp, updated_at: timestamp
statuses	id: bigint(20) unsigned, name: varchar(255), group_name: varchar(255), created_at: timestamp, updated_at: timestamp
device_gallery_images	id: bigint(20) unsigned, gallery_id: int(11), original_image_url: varchar(255), low_size_image_url: varchar(255), created_at: timestamp, updated_at: timestamp
orders	id: bigint(20) unsigned, status_id: int(11), user_id: int(11), device_id: int(11), comment_for_user: text, comment_for_moderator: text, created_at: timestamp, updated_at: timestamp
device_action_queue	id: bigint(20) unsigned, action: varchar(255), device_id: int(11), status: varchar(16), user_id: int(11), settings: text, created_at: timestamp, updated_at: timestamp
users	id: bigint(20) unsigned, name: varchar(255), email: varchar(255), email_verified_at: timestamp, password: varchar(255), remember_token: varchar(100), created_at: timestamp, updated_at: timestamp, role: varchar(16)
order_logs	id: bigint(20) unsigned, order_id: int(11), auto_comment: text, extra_comment_for_user: text, created_at: timestamp, updated_at: timestamp
device_galleries	id: bigint(20) unsigned, name: varchar(255), user_id: int(11), device_id: int(11), created_at: timestamp, updated_at: timestamp

Рисунок 3.4 – Таблиці в базі даних

Список усіх сторінок можете побачити на рис. 3.5.

```

public routes:
GET|HEAD / ..... public.home > HomeController@home
GET|HEAD login ..... login > Auth\AuthenticatedSessionController@create
POST login ..... generated::x30A1zT82sJpJ0xa > Auth\AuthenticatedSessionController@store

administration moderator routes:
GET|HEAD administration/moderator/dashboard administration.moderator.dashboard > ModeratorContr...
POST administration/moderator/order/complete/{orderId} administration.moderator.order.complete

administration user routes:
GET|HEAD administration/moderator/order/view/{orderId} administration.moderator.order.view > Or...
GET|HEAD administration/user/device/gallery/photo/view/{photoId} administration.user.device.gal...
GET|HEAD administration/user/device/gallery/photos/{galleryId} administration.user.device.galle...
GET|HEAD administration/user/device/list administration.user.device.list > DeviceController@use...
GET|HEAD administration/user/device/make/photo/{deviceId} administration.user.device.make.photo...
GET|HEAD administration/user/device/settings/{deviceId} administration.user.device.settings > D...
POST administration/user/device/settings/{deviceId} administration.user.device.settings.sub...
GET|HEAD administration/user/device/shop administration.user.device.shop > DeviceController@use...
GET|HEAD administration/user/order/list administration.user.order.list > OrderController@userLi...
GET|HEAD administration/user/order/summary/{modelId} administration.user.order.summary > OrderC...
POST administration/user/order/summary/{modelId} administration.user.order.summary.submit ...
GET|HEAD administration/user/order/view/{orderId} administration.user.order.view > OrderControl...
GET|HEAD dashboard ..... admin.user.dashboard > DashboardController@dashboard
POST logout ..... logout > Auth\AuthenticatedSessionController@destroy
GET|HEAD profile ..... profile.edit > ProfileController@edit
PATCH profile ..... profile.update > ProfileController@update
DELETE profile ..... profile.destroy > ProfileController@destroy
GET|HEAD profile/view ..... administration.user.profile.view > ProfileController@view

api routes
GET|HEAD api/device/action/get .. api.device.action.get > DeviceApiController@apiDeviceActionGet
POST api/device/action/photo/send api.device.action.photo.send > DeviceApiController@apiDev...

```

Рисунок 3.5 – Перелік всіх сторінок(роутів на сайті)

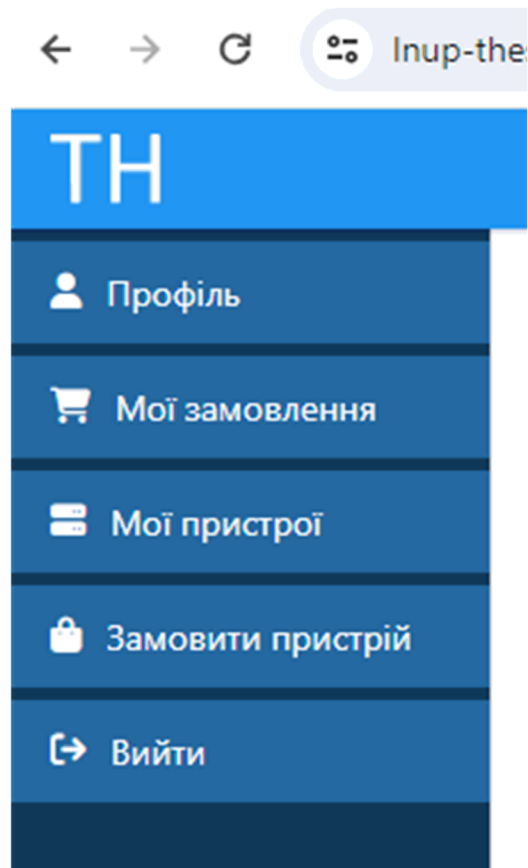


Рисунок 3.6 – Меню у кабінеті користувача

В дизайні вирішив обрати синій колір. Навігаційне меню у кабінеті користувача вирішив зробити зліва, щоб було більше простору. Багато стилів є стандартними для CSS фреймворка bootstrap. Це дуже спростило розробку дизайну. Іконки взято з безплатного перліку на сайті <https://fontawesome.com/icons>. Приклад дизайну можна побачити на рис. 3.6.

Код сайту є у відкритому доступі у Github: <https://github.com/sirius256/thesis>. Розробка велась на OpenServer – це локальний сервер на платформі Windows.

Для розгортання сайту використовується сервер, орендований у на компанії hostinger. Посилання на їх сайт: <https://www.hostinger.com.ua/>. Доменне ім'я **lnup-thesis-it-61-os.com** теж куплене у них.

3.3 Розробка на Raspberry Pi

3.3.1 Збірка

Спочатку потрібно було все зібрати до купи. Почитати документацію, щоб правильно все підключити. На рис 3.7 можна побачити розібраний прилад. На рис 3.8 зображено схему підключення камери до плати. На рис 3.9 зображено зібраний прилад.



Рисунок 3.7 – Розібраний прилад

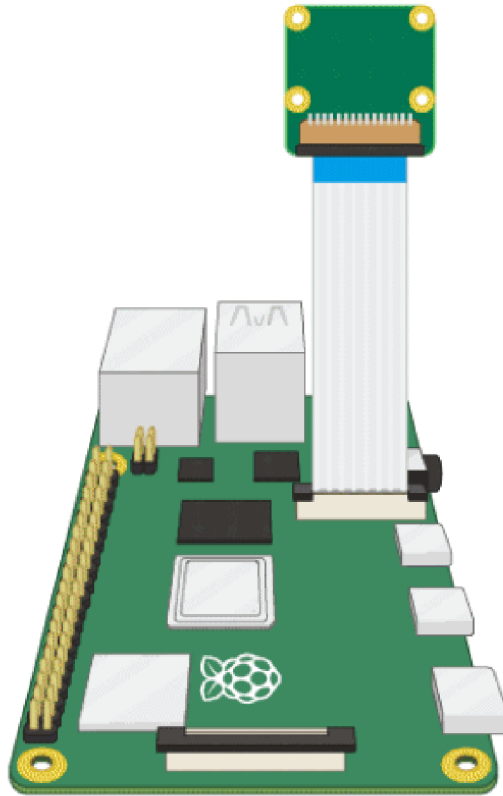


Рисунок 3.8 – Схема підключення камери до плати



Рисунок 3.9 – Зібраний прилад

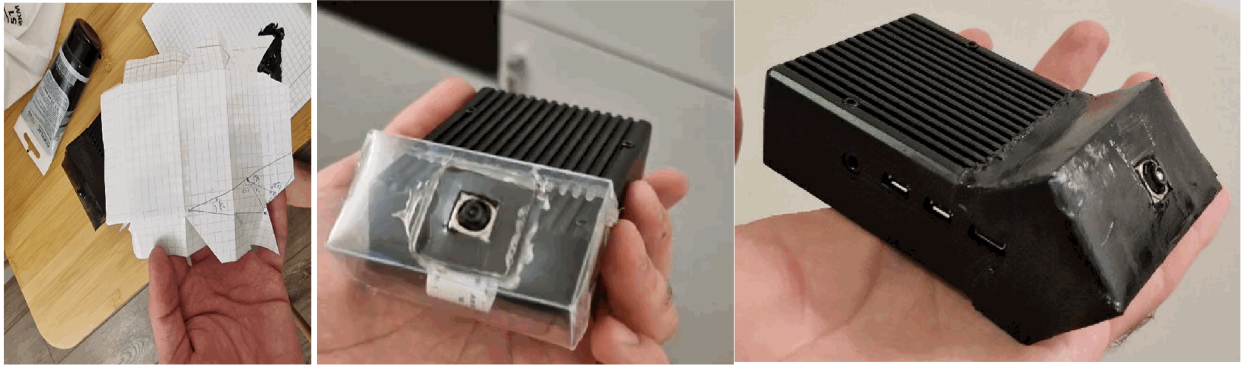


Рисунок 3.10 – Розробка корпусу

Коли прилад було зібрано, то камера не була закріплена, і могла легко пошкодитись. По паперовій моделі зліплено з тонкого пластику верхню захисну оболонку для модуля камери. Прикріплено клейовим пістолетом, і пофарбовано акриловою чорною фарбою. Це зображено на рис. 3.10.

3.3.2 Налаштування середовища і розробка скриптів

- скачати інсталятор <https://www.raspberrypi.com/software/>
- встановити на мікро сд картку Операційну систему
- вставити картку в raspberry
- під'єднати екран, мишу, клавіатуру, живлення, камеру
- бажано під'єднати радіатор до чіпів на raspberry, бо гріється
- запустити і слідувати інструкції встановлення, під час встановлення приєднатися до вайфаю
- відкрити термінал
- встановити PHP:
- `sudo apt-get install php -y`
- встановити розширення до PHP, щоб робити запити до api
- `sudo apt install php8.2 php8.2-cli php8.2-{bz2,curl,mbstring,intl}`
- робимо все по інструкції, щоб включити камеру: <https://projects.raspberrypi.org/en/projects/getting-started-with-picamera/2>

- якщо камери немає у списку девайсів, то її включати не потрібно, вона вже увімкнена по замовчуванню.
- `git clone https://github.com/sirius256/thesis_raspberry_pi.git`
- `cd thesis_raspberry_pi`
- `chmod +x make_photo.sh`
- `chmod +x main.sh`
- `sudo crontab -e`
- вставити там код:
- `@reboot . /home/[тут назва користувача]/Desktop/thesis_raspberry_pi/main.sh &`
- перезагрузити пристрій

На рис. 3.11 зображено інсталятор для операційної системи. На рис. 3.12 зображено перший крок, щоб увімкнути камеру. На рис. 3.13 зображено другий крок, щоб увімкнути камеру.

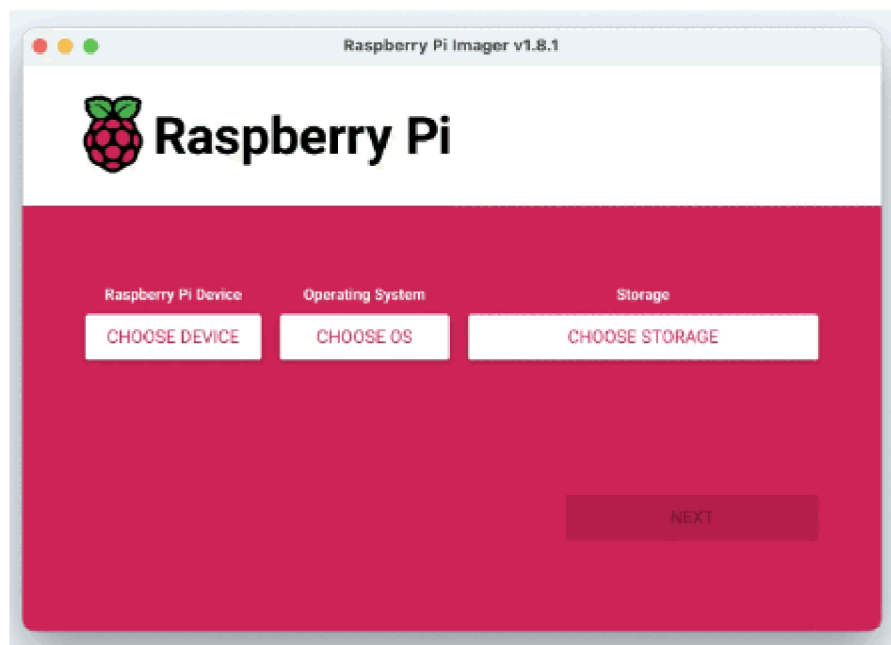


Рисунок 3.11 – Raspberry Pi Imager

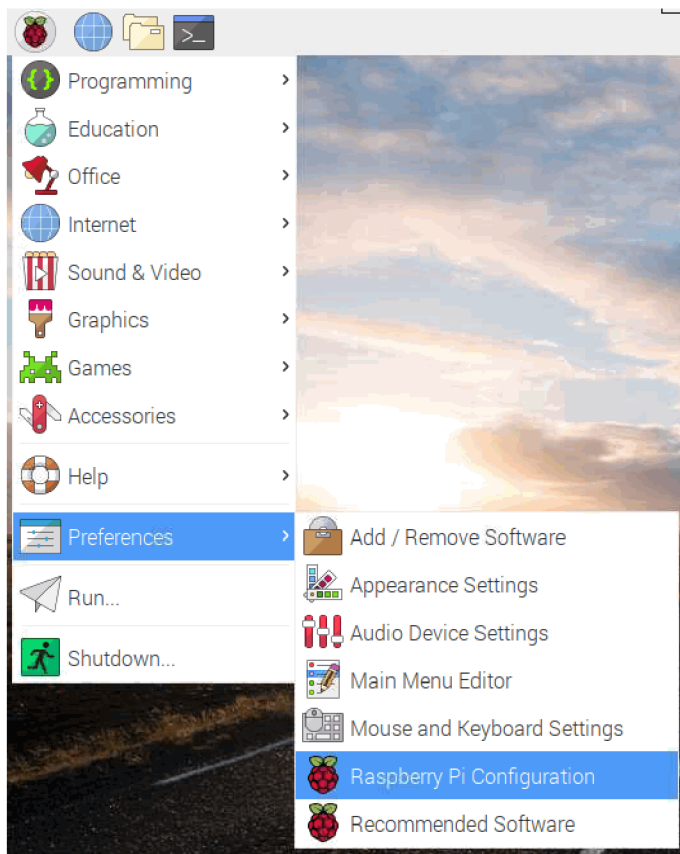


Рисунок 3.12 – Налаштування Raspberry

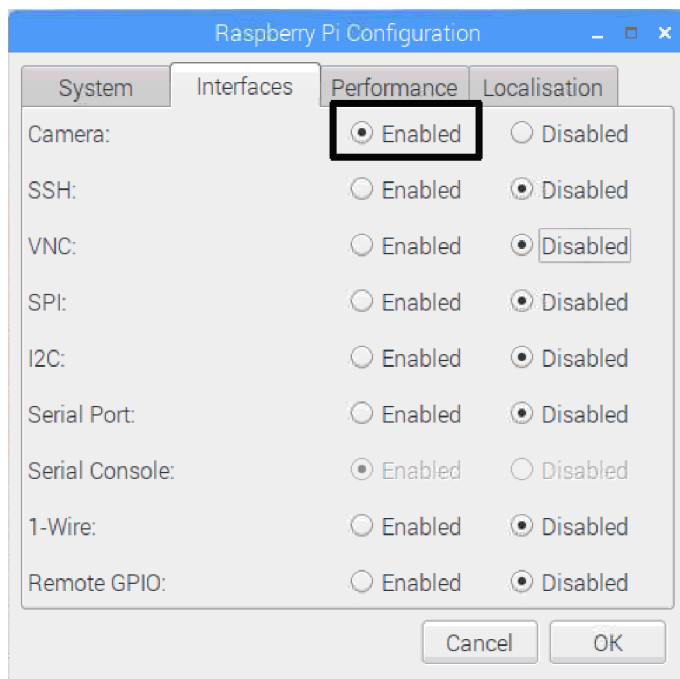


Рисунок 3.13 – Налаштування Raspberry, включення камери

```
1   while true
2   do
3       php /home/oleg/Desktop/thesis_raspberry_pi/getActions.php
4       sleep 1
5   done
```

Рисунок 3.14 – main.sh

Головна точка запуску це main.sh. Цей файл мусить запуснитись при старті приладу. Головне не забувати підключати до вайфаю прилад, інакше скрипт не працюватиме. Код main.sh зображено на рис 3.14.

Код скриптів є у відкритому доступі у Github: https://github.com/sirius256/thesis_raspberry_pi

3.4 Підсумки розробки

В результаті, створено сайт де відбувається управління приладом відеоспостереження. На сайті є кнопка «зробити фото», яка подає сигнал пристрою, і пристрій через Wi-Fi і інтернет закидає фото на сайт.

Сайт виконаний у стилі стартапу. Реалізована можливість створення заявки на купівлю приладу користувачем, обробки та підтвердження її модератором. Після того як модератор активує пристрій, користувач має змогу управляти пристроєм і віддалено робити фотографії. Ці фотографії можна буде переглянути у кабінеті користувача. Код сайту і скрипти Raspberry Pi розміщені у Github. Далі детальніше пройдемося по функціоналу з точки зору користувача.

Почнемо з головної сторінки. Шлях: <https://lnup-thesis-it-61-os.com/> Вона є публічною, є список доступних приладів для замовлення і виведена їх доступна кількість. Також там є 2 посилання на сторінку реєстрації і залогіювання. Хедер і футер спільні для всіх публічних сторінок. На рис 3.15 зображено головну сторінку сайту.

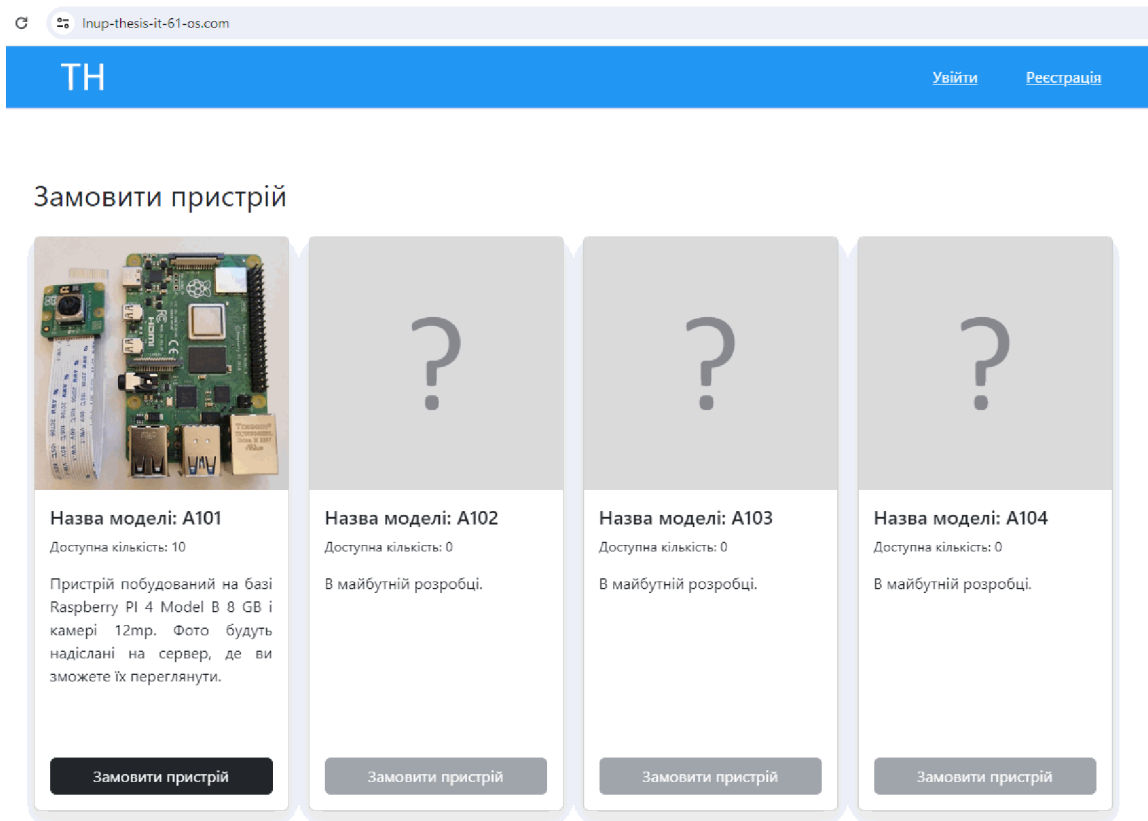


Рисунок 3.15 – головна сторінка

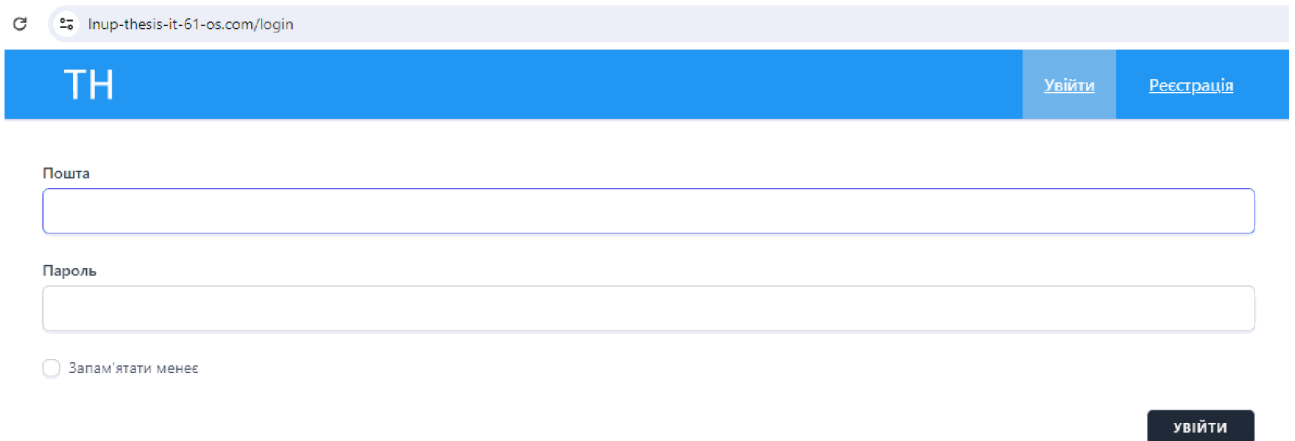


Рисунок 3.16 – Сторінка для входу у свій кабінет

На рис. 3.16 зображено сторінку, на котрій користувач може увійти у свій кабінет. Також це може зробити модератор.

Рисунок 3.17 – Сторінка реєстрації

А тут може зареєструватися будь-який користувач. Але модератор не може бути створеним через цю сторінку. Це може зробити тільки адміністратор. На рис. 3.17 зображено сторінку реєстрації.

Користувач попадає на головну сторінку. Вона є публічною, є список доступних приладів для замовлення і виведена їх доступна кількість. Коли користувач нажме на кнопку «замовити пристрій», то він перейде на сторінку реєстрації, і туди буде передано ід моделі пристрою, яка знадобить, для показу наступного кроку. На рис 3.18 зображено сторінка реєстрації, яка є першим кроком до створення замовлення.

Рисунок 3.18 – Сторінка реєстрації, яка є першим кроком до створення замовлення.

Далі користувач попаде у свій кабінет, а саме на сторінку підтвердження замовлення. Для цього і була потрібна ід моделі приладу, яку ми передавали з головної сторінки на сторінку реєстрації і тепер сторінку підтвердження замовлення. На рис 3.19 зображено сторінка підтвердження замовлення.

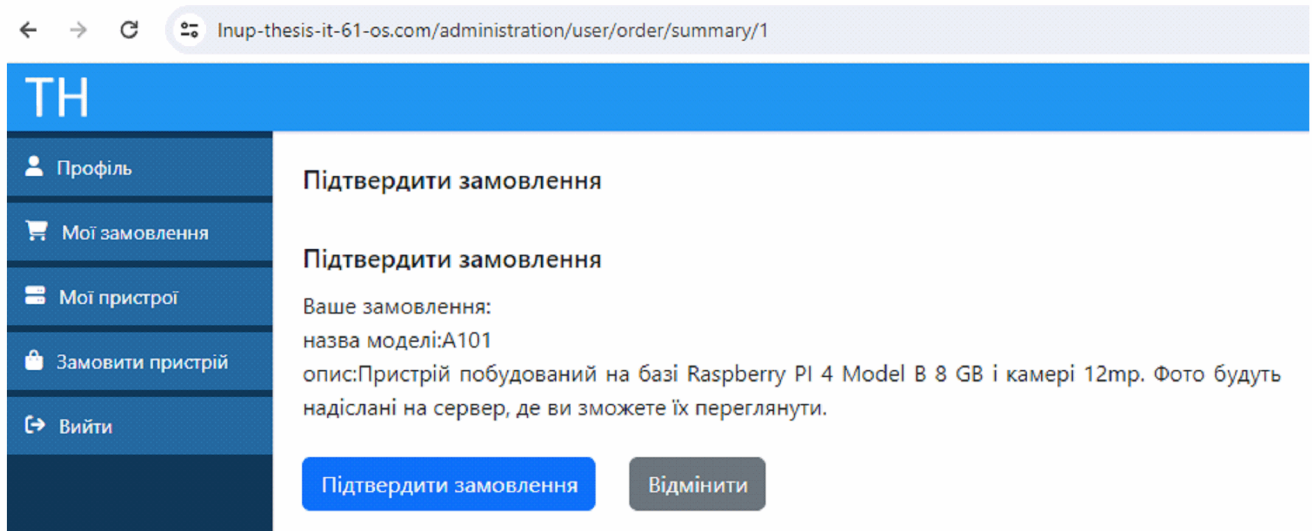


Рисунок 3.19 – Сторінка підтвердження замовлення

І далі користувач попаде на сторінку з усіма його замовленнями, на якій може переглянути деталі замовлення. На рис 3.20 можна побачити список всіх замовлень користувача. На рис 3.21 можна побачити деталі певного замовлення користувача.

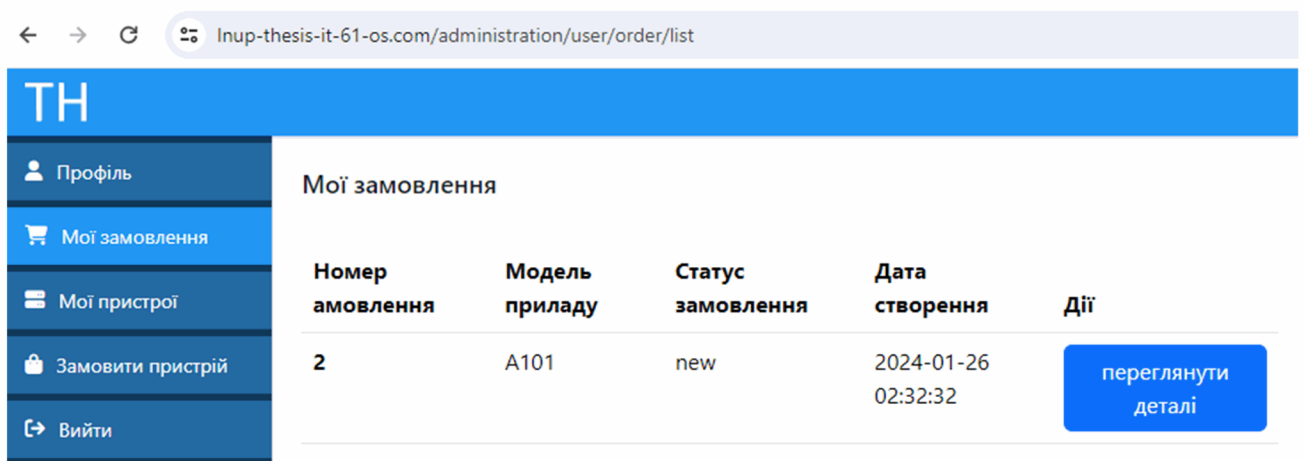


Рисунок 3.20 – Сторінка з замовлення користувача

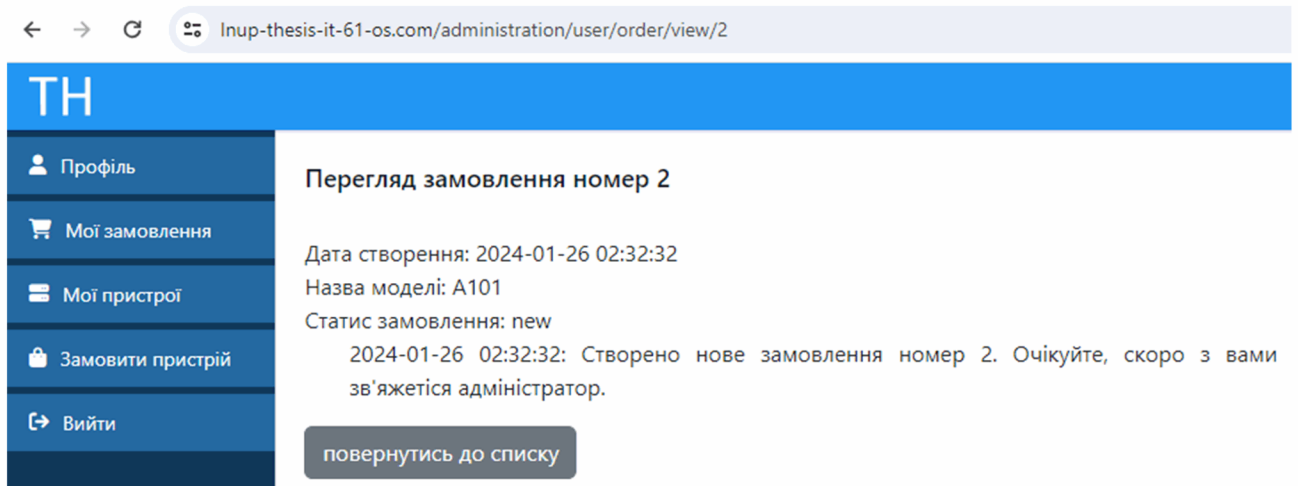


Рисунок 3.21 – Сторінка з деталями замовлення користувача

Після цього етапу модератор зв'язується з користувачем, домовляється про оплату і доставку, і потім активує пристрій через кабінет модератора. І тоді уже пристрій стане доступним для користування. До цих сторінок має доступ тільки модератор. На його головній сторінці відображаються всі замовлення всіх користувачів. Він може їх завершувати чи переглянути деталі. На рис 3.22 зображено головну сторінку модератора.

Номер амовлення	Модель приладу	Статус замовлення	Дата створення	Дії
1	A101	complete	2024-01-25 12:30:12	переглянути деталі Активувати пристрій і завершити замовлення
2	A101	new	2024-01-26 02:32:32	переглянути деталі Активувати пристрій і завершити замовлення

Рисунок 3. 22 – Модератора головна сторінка

Коли модератор активує пристрій, то він відобразатиметься у пункті меню «Мої пристрої». Тут користувач може вже надсилати сигнал пристрою для того, щоб зробити фотографію. Через якийсь час фото з'явиться у галереї. Звісно, що пристрій мусить бути увімкненим і підключеним до Wi-Fi. На рис 3.23 зображено

сторінку «мої пристрої», де можна керувати своїми пристроями. На рис 3.24 зображено приклад сторінки галереї приладу.

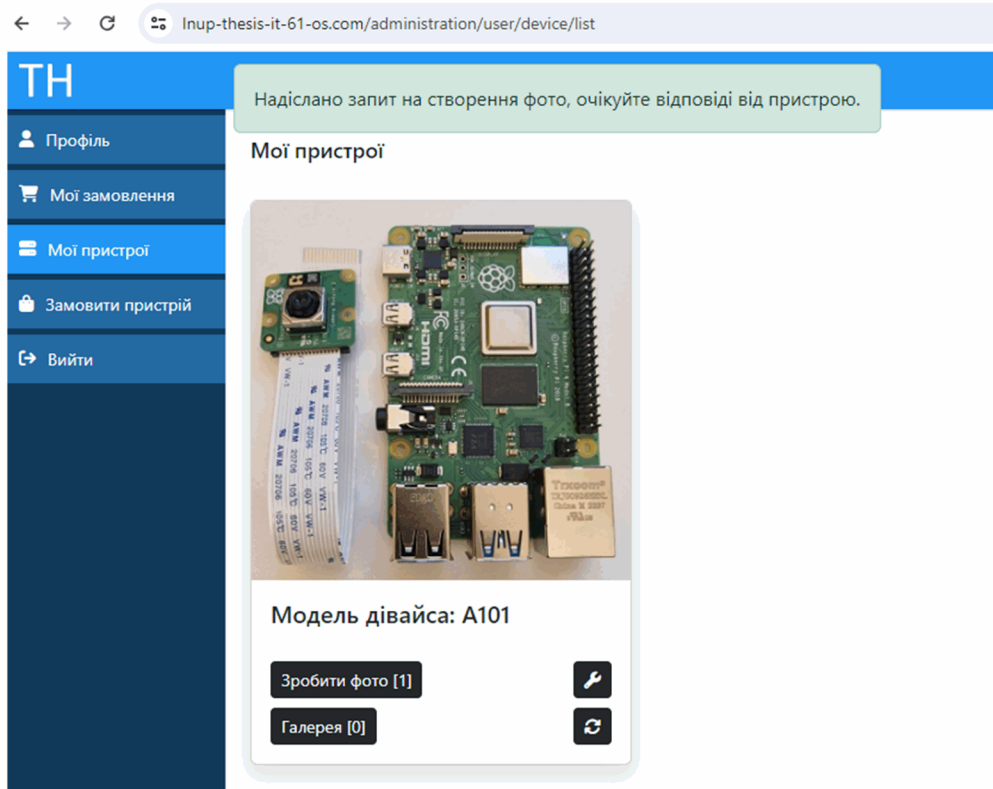


Рисунок 3.23 – Сторінка користувача, де можна керувати своїми пристроями

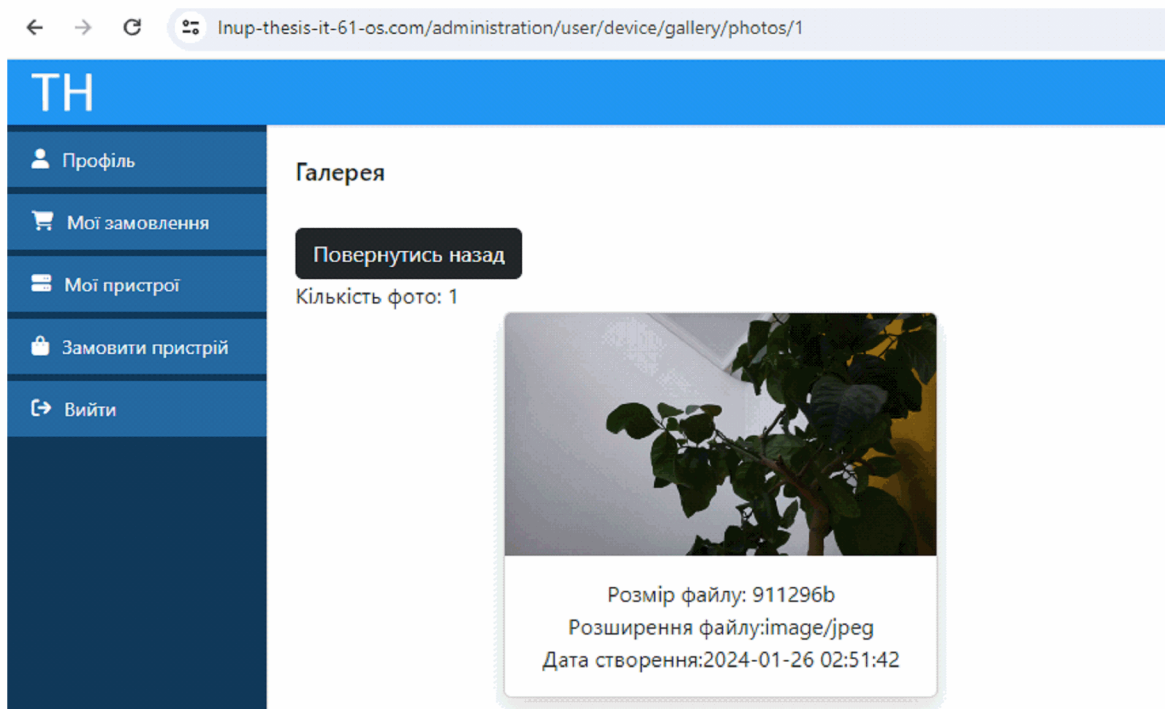


Рисунок 3.24 – Галерея приладу

В галереї відображаються тільки ті фото, які належать саме цьому користувачу. Підібрати ід картинки не вийде.

На рис 3.25 зображено магазин користувача. Тут користувач може замовити собі ще один пристрій.

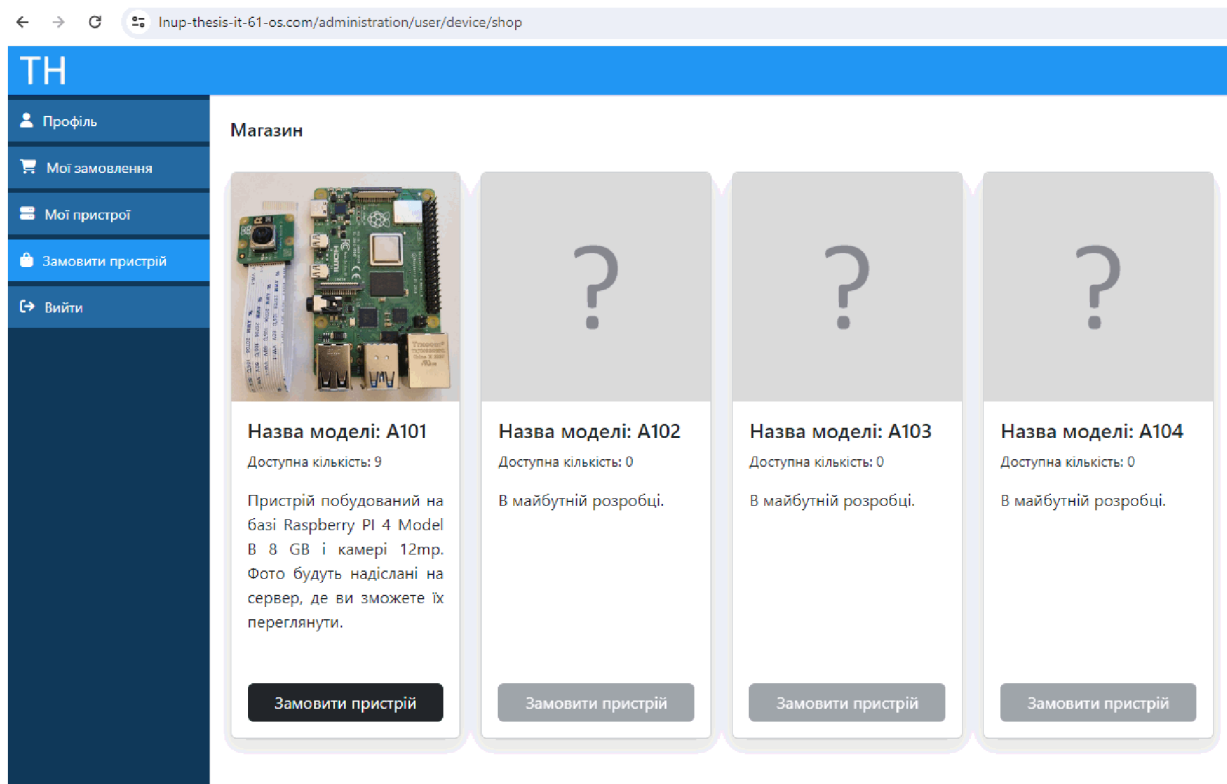


Рисунок 3.25 – Магазин користувача

Оскільки мобільні телефони дуже популярні, тому я розробив мобільну версію сайту. На рис 3.26, 3.27, 3.28, 3.29 можна побачити приклади адаптації верстки під мобільні телефони. Мобільна версія сайту в сучасному веб-середовищі є критично важливою для успішної онлайн-присутності. Зростання використання мобільних пристроїв, таких як смартфони та планшети, визначає необхідність оптимізації веб-сайтів для мобільних платформ. Мобільна версія стає стратегічною необхідністю для бізнесів та веб-проектів, оскільки вона сприяє розширенню аудиторії та забезпеченню конкурентоспроможності в онлайн-середовищі.



Рисунок 3.26 – Мобільна версія сторінки «мої пристрої»



Рисунок 3.27 – Мобільна версія сторінки «мої пристрої»

The screenshot shows a mobile browser interface for the login page. At the top, the address bar displays 'inup-thesis-it-61-os.com/login'. Below the address bar is a blue navigation bar with the logo 'TH' on the left and two buttons: 'Увійти' (Login) and 'Реєстрація' (Registration). The main content area contains a form with the following elements: a 'Пошта' (Email) label above an empty text input field; a 'Пароль' (Password) label above an empty text input field; a radio button labeled 'Запам'ятати менес' (Remember me); and a dark blue button labeled 'УВІЙТИ' (Login).

Рисунок 3.28 – Мобільна версія сторінки «Увійти»

The screenshot shows a mobile browser interface for the registration page. At the top, the address bar displays 'inup-thesis-it-61-os.com/register'. Below the address bar is a blue navigation bar with the logo 'TH' on the left and two buttons: 'Увійти' (Login) and 'Реєстрація' (Registration). The main content area contains a form with the following elements: a 'Нікнейм' (Nickname) label above an empty text input field; a 'Пошта' (Email) label above an empty text input field; a 'Пароль' (Password) label above an empty text input field; a 'Введіть пароль ще раз' (Enter password again) label above an empty text input field; a link labeled 'Уже зареєстровані?' (Already registered?); and a dark blue button labeled 'ЗАРЕЄСТРУВАТИСЬ' (REGISTER).

Рисунок 3.29 – Мобільна версія сторінки «реєстрація»

Крім цього користувач може оновити пароль, дані користувача, і взагалі стерти свій акаунт з сайту.

Також є сторінка де користувача може переглянути свої дані профілю. На рис 3.30 зображено сторінку профілю користувача. На ній є кнопка «редагувати профіль». Натиснувши на неї користувач попаде на форму оновлення даних профілю таких як нікнейм і пошта. Це зображено на рис 3.31.

Нижче буде форма оновлення парою. Щоб оновити пароль потрібно ввести теперішній пароль і новий пароль 2 рази(щоб не допустити помилки). Це можна побачити на рис 3.32.

Ще нижче буде форма видалення акаунту. Її зображено на рис 3.33. Видалити акаунт випадково не вийде. Потрібно спочатку буде ввести пароль до акаунту, зображено рис 3.34.

В майбутньому, для зручності користувача, можна розділити ці 3 форми на три окремі сторінки.

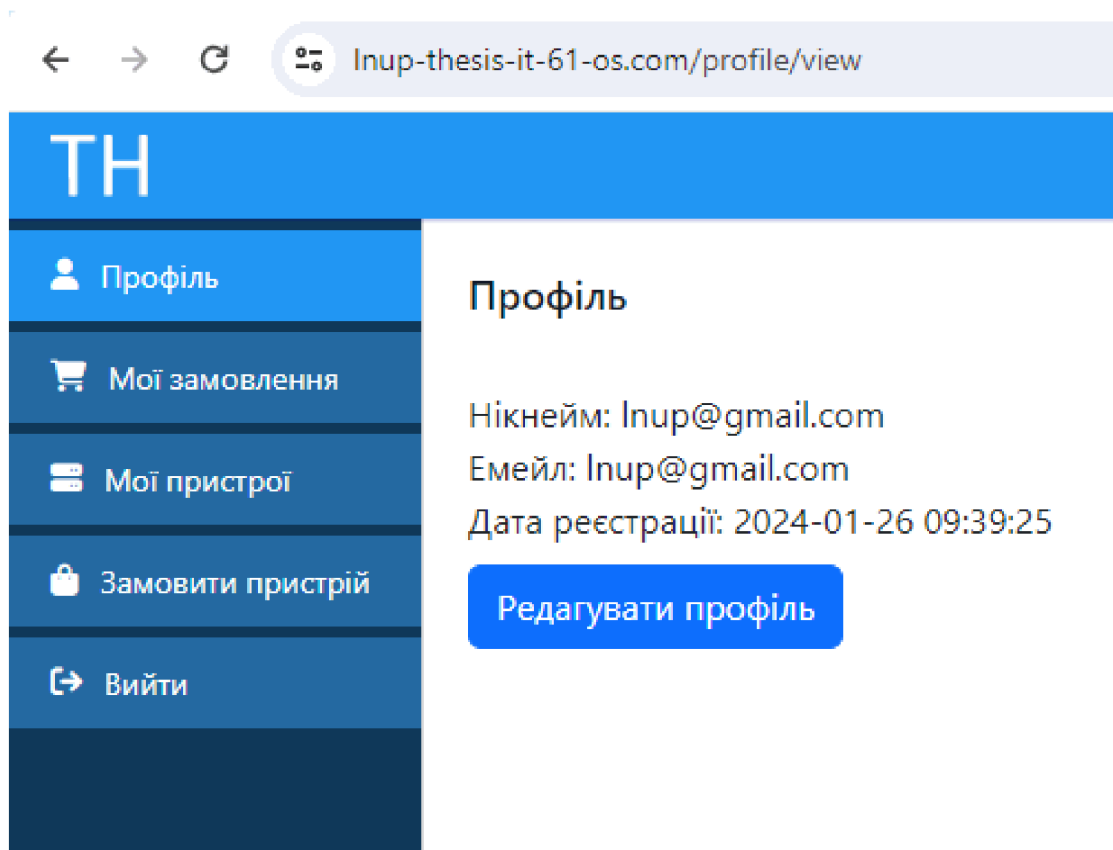


Рисунок 3.30 – Сторінка профілю користувача

← → ↻ Inup-thesis-it-61-os.com/profile

ТН

Профіль

Мої замовлення

Мої пристрої

Замовити пристрій

Вийти

Редагування профілю

Profile Information

Update your account's profile information and email address.

Name

Email

SAVE

Рисунок 3.31 – Сторінка редагування профілю, форма оновлення нікнейму і пошти

Update Password

Ensure your account is using a long, random password to stay secure.

Current Password

New Password

Confirm Password

SAVE

Рисунок 3.32 – Сторінка редагування профілю, форма оновлення паролю

Delete Account

Once your account is deleted, all of its resources and data will be permanently deleted. Before deleting your account, please download any data or information that you wish to retain.

DELETE ACCOUNT

Рисунок 3.33 – Сторінка редагування профілю, форма видалення акаунту

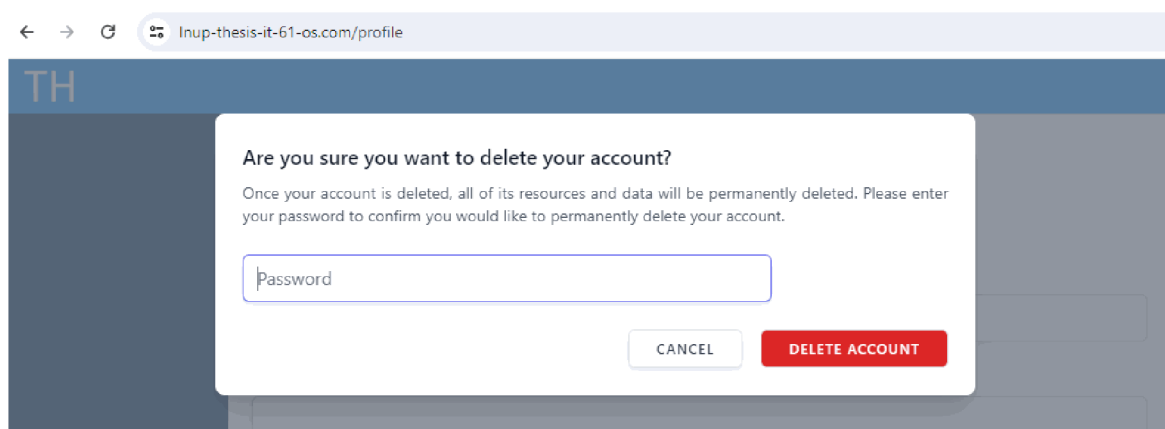


Рисунок 3.34 – Сторінка редагування профілю, підтвердження видалення акаунту

В результаті, створено сайт у стилі стартапу де відбувається управління Raspberry Pi і підключеною камерою. На сайті є кнопка «зробити фото», яка подає сигнал пристрою, і пристрій через Wi-Fi і інтернет закидає фото на сайт. Ці фотографії можна буде переглянути у кабінеті користувача. Реалізована можливість створення заявки на купівлю приладу користувачем, обробки та підтвердження її модератором. Після того як модератор активує пристрій, користувач має змогу управляти пристроєм.

РОЗДІЛ 4

ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Перелік небезпечних і шкідливих виробничих факторів діючих у робочій зоні

Відповідно до ДСТУ 12.0.003-74 інженер під час роботи у робочій зоні може знаходитися під впливом наступних небезпечних і шкідливих виробничих факторів:

- підвищена або понижена температура повітря в робочій зоні;
- підвищена або понижена температура поверхонь обладнання та матеріалів;
- низька освітленість робочої зони;
- підвищена або понижена вологість повітря;
- підвищена або понижена циркуляція повітря;
- підвищений рівень шуму на робочому місці;
- підвищений рівень електромагнітного випромінювання;
- підвищений рівень статичної електрики.

Розглянемо детально деякі з цих факторів.

Підвищений рівень статичної електрики. Рівень напруженості електростатичних полів на робочому місці інженера не повинен перевищувати: за тривалістю впливу 1 година - 60 кВ / м, і до 9 годин - 20 кВ/м. Поверхневий електростатичний потенціал не повинен перевищувати 500 В.

Підвищена або понижена вологість повітря; підвищена або понижена циркуляція повітря. Відповідно до ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень» у виробничих приміщеннях при виконанні робіт з більш високим емоційним навантаженням, оптимальні значення температури повітря становлять 22–24 °С і відносної вологості в 40–

60 % мають бути збережені, разом з циркуляцією повітря зі швидкістю не більше 0,1 м/с.

Підвищений рівень шуму. Основні характеристики і граничні рівні шуму на робочих місцях, визначені у ДСН 3.3.6.037-99 «Шум. Загальні вимоги безпеки». Прийнятний рівень звукового тиску для аналітичних та вимірювальних робіт становить 60 дБА.

Низька освітленість робочої зони. Робота з електричними приладами вимірювання належить класу V зорової роботи - висока точність. Відповідно до ДБН В.2.5-28:2018 «Природне і штучне освітлення» для цього типу роботи рівень штучного освітлення унормований такими характеристиками: комплексне освітлення – 300–400 лк (у тому числі в загальній складності 200 люкс); в загальній системі освітленості 200–300 лк. ВДТ на основі електронно-променевих трубок є джерелом декількох видів електромагнітного випромінювання, зокрема мікрохвиль нетеплової інтенсивності.

4.2. Технічні та організаційні заходи по зменшенню рівня впливу небезпечних та шкідливих виробничих факторів

Небезпечні та шкідливі фактори, що перераховані вище, так чи інакше впливають на самопочуття людини, що призводить до зниження продуктивності її праці. Заходами щодо зниження впливу шкідливих факторів можуть бути:

- 1) для підвищеного рівню статичної електрики: при підвищеному рівні напруженості електростатичного поля час роботи повинен бути скорочений, також повинен використовуватися захисний екран з тонкої сітки або скла, який приймає на себе електростатичний заряд, також людина не повинна бути розташована на відстані 5–10 см від екрану монітора, тому що рівень інтенсивності може досягати 140 В/м для електричних компонентів, що значно перевищує допустимі значення ДСанПіН 3.3.2-007-98;

2) до несприятливого мікроклімату: потрібно дотримуватись правил і норм, зазначених в ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень»;

3) до підвищеного рівню запиленості робочої зони: необхідно часто робити вологе прибирання робочої зони;

4) до підвищеного рівню шуму: зниження рівня шуму на шляху його поширення (за допомогою місцевої і загальної звукоізоляції, шумових екранів і поглинаючих фільтрів).

Вимоги до температури повітря в робочій зоні. У кабінах, на пультах і постах керування технологічними процесами, в залах обчислювальної техніки та інших виробничих приміщеннях при виконанні робіт операторського типу, пов'язаних з нервово-емоційною напругою, повинні дотримуватися оптимальні величини температури повітря 22-24 ° С.

Вимоги до температури поверхонь обладнання та матеріалів. При забезпеченні оптимальних показників мікроклімату температура внутрішніх поверхонь конструкцій, що огороджують робочу зону (стіни, підлоги, стелі та ін), або пристроїв (екранів і т.п.), а також температура зовнішніх поверхонь технологічного обладнання або огорожувальних його пристроїв не повинні виходити більш ніж на 2 ° С за межі оптимальних величин температури повітря для окремих категорій робіт. При температурі поверхонь конструкцій нижче або вище оптимальних величин температури повітря робочі місця повинні бути віддалені від них на відстань не менше 1 м.

При забезпеченні допустимих показників мікроклімату температура внутрішніх поверхонь конструкцій, що огороджують робочу зону (стіни, підлоги, стелі та ін), або пристроїв (екранів і т.п.) не повинна виходити за межі допустимих величин температури повітря.

При забезпеченні оптимальних і допустимих показників мікроклімату в холодний період року слід застосовувати засоби захисту робочих місць від радіаційного охолодження від закслених поверхонь віконних прорізів, у теплий період року - від потрапляння прямих сонячних променів.

Низька освітленість робочої зони. Вимоги до освітлення. Мінімальна освітленість встановлюється залежно від характеру зорової роботи за найменшим розміром об'єкта розрізнення, контрастом об'єкта з фоном і характеристикою фону. Враховується система робочого освітлення (загальне або комбіноване) та джерела світла (лампи розжарювання або газорозрядні).

Вимоги до вологості повітря. Параметри мікроклімату можуть мінятися в широких межах, тоді як необхідною умовою життєдіяльності людини є підтримка постійності температури тіла завдяки терморегуляції, тобто здібності організму регулювати віддачу тепла в оточуюче середовище. Принцип нормування мікроклімату – створення оптимальних умов для теплообміну тіла людини з навколишнім середовищем.

Вимоги до рівня шуму. В Україні і в міжнародній організації зі стандартизації застосовується принцип нормування шуму на основі граничних спектрів (граничні допустимих рівнів звукового тиску) в октавних смугах частот.

Граничні величини шуму на робочих місцях регламентуються ДСН 3.3.6.037-99 "Шум. Загальні вимоги безпеки". В ньому закладено принцип встановлення певних параметрів шуму, виходячи з класифікації приміщень за їх використання для трудової діяльності різних видів.

Рівень шуму на робочому місці математиків-програмістів і операторів відеоматеріалів не повинен перевищувати 50 дБ, а в залах обробки інформації на обчислювальних машинах – 65 дБ. Для зниження рівня шуму стіни і стелі приміщень, де встановлені комп'ютери, можуть бути фанеровані звукопоглинальними матеріалами. Рівень вібрації в приміщеннях обчислювальних центрів може бути понижений шляхом встановлення устаткування на спеціальні вібро-ізолятори.

Вимоги до циркулювання повітря. Для забезпечення комфортних умов використовуються як організаційні методи (раціональна організація проведення робіт залежно від пори року і доби, чергування праці і відпочинку), так і технічні засоби (вентиляція, кондиціонування повітря, опалювальна система) (табл. 4.1, 4.2).

Таблиця 4.1 – Вимоги до швидкості руху повітря для приміщень

	Період року	Величина
1	Холодний	до 0,1 м/с
2	Теплий	0,1...0,2 м/с

Таблиця 4.2 – Норми подачі свіжого повітря в приміщення

	Характеристика приміщення, об'єм на одну людину	Об'ємна витрата свіжого повітря, що подається в приміщення, м ³ на одну людину в годину
1	до 20 м ³	не менше 30
2	20...40 м ³	не менше 20
3	понад 40 м ³	природна вентиляція

Вимоги до рівня електромагнітного випромінювання та заходи щодо його зниження. Допустимі рівні електромагнітного поля на робочих місцях при роботі з джерелами електромагнітних випромінювань встановлюються відповідно до вимог ГОСТу 12.1.006-84 ССБТ

У ближній зоні, яка має фізичне значення при частотах до 300 МГц, нормуються напруженості електричної і магнітної складових полів.

У дальній зоні в діапазоні частот 300 МГц - 300 ГГц, у якому, як правило, і перебуває персонал, що обслуговує джерела ЕМВ із довжиною хвилі менше метра, нормується густина потоку енергії та енергетичне навантаження.

Вимоги до рівня електростатичного поля. Заряди статичної електрики на робочому місці, де знаходиться електронне обладнання, можуть виникати на незаземлених металевих і діелектричних поверхнях устаткування, на покриттях підлоги, на панелях стін. Крім ушкоджень електронних компонентів, існують розряди статичної електрики, що виникають між іншими працівниками й елементами устаткування, що знижують продуктивність праці.

4.3. Інструкція з охорони праці при обслуговуванні кліматичних системи

Згідно з вимогами НПАОП 0.00-4.15-98 «Положення про розробку інструкцій з охорони праці» (Наказ Держнаглядохоронпраці від 29.01.1998 р. №9) розробимо типову інструкцію.

Загальні вимоги.

1) До роботи з приладом допускається інженерно-технічний склад, що вивчив об'єкт, інструкцію з технічної експлуатації, діючу інструкцію, а також склав залік з технічної безпеки та пожежної безпеки;

2) Ремонт та наладку мають виконувати не менше, ніж два спеціаліста. При цьому інструмент має бути справним, джерело живлення відключеним;

3) Робоче місце або ділянка має бути обладнана засобами захисту від пожежі – вогнегасниками порошкового або іншого типу;

Вимоги безпеки перед початком роботи.

- прилад правильно підключений і має заземлення;
- перед запуском не залишилось зайвих незакріплених предметів;
- всі прилади, що досліджуються, закріплені належним чином;
- усі з'єднувальні кабелі та місця комутації справні.

Вимоги безпеки під час роботи.

- використовувати тільки справний інструмент і за призначенням;
- слідкувати, щоб на робочому місці не було зайвих предметів, що відволікають увагу і можуть призвести до травмування;
- при появі іскріння, короткого замикання, запаху гару, диму прилад негайно відключити та виявити причини можливого виникнення пожежі.

Вимоги безпеки після закінчення роботи.

- вимкнути прилад, коли спеціаліст залишає своє місце;
- прибрати своє робоче місце;
- перевірити наявність всього інструменту згідно опису;
- повідомити керівника робіт про виявлені недоліки в роботі приладу.

Вимоги безпеки в аварійних ситуаціях.

- у випадку виникнення пожежі негайно викликати пожежну команду. До її приїзду приступити до тушіння пожежі підручними засобами, а також спасінню людей та надання їм допомоги;
- у випадку ураження електричним струмом відключити живлення, прийняти необхідні міри по наданню першої медичної допомоги;
- в робочому приміщенні працівники мають бути ознайомлені з планом та порядком евакуації з приміщення, розміщеним на видному місці.

4.4 Забезпечення пожежної та вибухової безпеки при обслуговуванні кліматичних систем

У відповідності до Закону України «Про пожежну безпеку» та вимог НАПБ А.01.001-2004 «Правила пожежної безпеки в Україні» по запобіганню пожежі та пожежного захисту розглянемо необхідні заходи для забезпечення пожежної та вибухової безпеки. Пожежна та вибухова безпека – це стан об'єкту, при якому виключається виникнення пожежі і вибуху, а у випадку появи мінімізується чи ліквідується дія на людей небезпечних факторів пожежі і вибуху, а також забезпечується захист і збереження матеріальних цінностей. Під час обслуговування розробленої кліматичної підсистеми розумного будинку, пожежа може виникнути у випадку перевантаження електричного обладнання при обслуговуванні клімат-контролю, внаслідок пошкодження ізоляції, неякісного з'єднання електричної проводки чи короткого замикання. Для уникнення таких ситуацій усе електричне обладнання слід комутувати через автомат захисту на випадок перевантаження та короткого замикання. Крім того періодично проводять перевірку стану ізоляції проводів.

РОЗДІЛ 5

ОПИС СТАРТАПУ

5.1 Комерція

Прилад вийшов досить дорогим, але натомість, легким в розробці і потужним з запасом функціональності, і великим спектром можливих девайсів. Хороша програмна база.

Ціни:

- Raspberry Pi Model B 8GB – 4999грн
 - Камера Raspberry Pi V3 12 MP – 1650грн
 - Алюмінієвий корпус-радіатор – 555грн
 - MicroSD карта 128 GB – 279грн
 - Кабель кабель HDMI => Micro HDMI – 253грн
- Загальна сума $4999 + 1650 + 555 + 279 + 253 = 7736$ грн.

5.2 Перспективи і майбутній розвиток

У проекті можна реалізувати наступні покращення:

- збільшити модельний ряд з різними фішками і ціновими діапазонами;
- здешевити базу;
- помістити батарею в середину корпусу пристрою. І додати до неї керуючу плату, щоб вона могла працювати і від мережі і портативно. Ще варіант зробити живлення від батарейок;
- ще як варіант додати шторку для камери, щоб її можна було відкрити тільки фізично, але щоб закрити можна було з сайту;

- додати датчики руху, щоб можна було робити фотографію чи відео при будь-якому русі зафіксованому датчиком;
- зробити можливість знімання відео. Як постійно так і в певному проміжку часу;
- збільшити кількість налаштувань пристрою в кабінеті користувача. Додати статистику в кабінеті адміністратора. Додати можливість редагувати і додавати моделі з адміністратором сайту;
- додати більше полів для користувача такі як телефон (обов'язковий) і адреса доставки (не обов'язкові);
- додати можливість створення коментарів до замовлення;
- додати соціальні мережі на сайт;
- зробити відео-огляд на ютубі;
- створити корпус під цей пристрій;
- зробити модель з інфрачервоною камерою;
- зробити водонепроникний корпус;
- протестувати в екстремальних умовах;
- покращити швидкість фотографування на стороні пристрою;
- зробити маленькі прев'ю фотки, щоб воно швидше грузилось для користувача і використовували менше трафіку і серверного часу;
- удосконалити обмін даними між пристроєм і сайтом;
- протестувати безпеку, можливо додати ще додаткову авторизацію для пристроїв. Забезпечити детальну перевірку файлів, які надсилаються на сервер;
- додати параметри фотографій, такі як формат фото, розширення вихідного фото. Те саме зробити і для відео;
- додати мікрофон для запису звуку;
- можна ще додати індикатор того, що пристрій увімкнений і що, камера саме зараз працює;

- коли буде батарея додати індикатор заряду. Це можна зробити за допомогою малого екрану чи світлодіодів;
- зробити сайт мультимовним;
- зробити рефактор і покращення деяких частин на сайті які вже реалізовані;
- додати капчу на сайт;
- удосконалити головну сторінку для кращої індексації гуглом;
- підключити мейлінг;
- підключити різні види оплати;
- додати гарну візуальну статистику для користувача;
- зробити різні тарифні плани.

ВИСНОВКИ

Розроблено систему управління охоронними засобами відеоспостереження, зокрема пристрій і програмне забезпечення до нього. Протестовано на стабільність і сумісність групу технологій, при розробці системи управління охоронними засобами відеоспостереження. Розроблено архітектуру обміну даними між приладом і сервером.

Визначено оптимальний стек технологій для розробки серверної частини системи. Обрано PHP, MySQL, Laravel, JavaScript, HTML, CSS. Визначено оптимальні хардверні складові для розробки приладу.

Протестовано повний цикл розробки готового продукту, від проектування приладу, до використання користувачем. Розроблено список можливих удосконалень для приладу і для сайту. І подальшого можливого розвитку стартапу.

У подальшому для приладу, можна здешевити базу і взяти за основу простіший і дешевший варіант. Також можна додати більше налаштувань для самого приладу, таких як формат фотографій (png, jpeg та інші) або їх розширення. Додати індикатори зйомки і включення приладу. Зробити прилад більш мобільним, додавши батарею і її контролер. Додати додаткові датчики, такі як датчик руху, і мікрофон чи інфрачервону камеру. І можна удосконалити та пришвидшити комунікацію між приладом і сайтом.

Для самого сайту варто зробити вивід статистики для адміністратора, розширити функціонал з керуванням замовлення модератором, удосконалити дизайн, розробити можливість додавати моделі і девайси адміністратором. Розробити можливість видалення фотографії користувачем. Та оптимізувати головну сторінку, щоб вона краще індексувалась пошуковими роботами. Зняти відео огляд пристрою.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Вікіпедія Raspberry Pi. URL: https://uk.wikipedia.org/wiki/Raspberry_Pi (дата звернення: 06.01.2024).
2. Інтернет магазин розетка. URL: <https://rozetka.com.ua/ua/videonablyudenie> (дата звернення: 03.01.2024).
3. Основи інтернету речей. URL: <http://edu.asu.in.ua/mod/book/view.php?id=112> (дата звернення: 01.01.2024).
4. Основи охорони праці. В. Ц. Жидецький, В. С. Джигирей, О. В. Мельников — Вид. 4-е, Львів: Афіша, 2015.
5. Сайт компанії Ајах. URL: <https://rozetka.com.ua/ua/videonablyudenie> (дата звернення: 04.01.2024).
6. Сайт компанії Raspberry Pi. URL: <https://www.raspberrypi.org> (дата звернення: 05.01.2024).
7. A Complete Guide to Flexbox. URL: <https://css-tricks.com/snippets/css/a-guide-to-flexbox/> (дата звернення: 09.01.2024).
8. Bootstrap documentation. URL: <https://getbootstrap.com/docs/5.0/getting-started/introduction/> (дата звернення: 09.01.2024).
9. CSS documentation. URL: <https://developer.mozilla.org/en-US/docs/Web/CSS> (дата звернення: 08.01.2024).
10. Diagrams app. URL: <https://app.diagrams.net/> (дата звернення: 09.01.2024).
11. Fontawesome documentation. URL: <https://fontawesome.com/icons> (дата звернення: 10.01.2024).
12. Git documentation. URL: <https://git-scm.com/doc> (дата звернення: 09.01.2024).
13. Hostinger. URL: <https://www.hostinger.com.ua/documentation> (дата звернення: 07.01.2024).
14. JavaScript documentation. URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> (дата звернення: 07.01.2024).

15. Laravel artisan documentation. URL: <https://laravel.com/docs/10.x/artisan> (дата звернення: 10.01.2024).
16. Laravel documentation. URL: <https://laravel.com/docs/10.x/readme> (дата звернення: 10.01.2024).
17. Laravel eloquent documentation. URL: <https://laravel.com/docs/10.x/eloquent> (дата звернення: 10.01.2024).
18. Laravel database documentation. URL: <https://laravel.com/docs/10.x/database> (дата звернення: 10.01.2024).
19. Laravel installation documentation. URL: <https://laravel.com/docs/10.x/installation> (дата звернення: 10.01.2024).
20. Laravel routing documentation. URL: <https://laravel.com/docs/10.x/routing> (дата звернення: 10.01.2024).
21. Laravel views documentation. URL: <https://laravel.com/docs/10.x/views> (дата звернення: 10.01.2024).
22. Laravel database migrations documentation. URL: <https://laravel.com/docs/10.x/migrations> (дата звернення: 10.01.2024).
23. Laravel database seeds documentation. URL: <https://laravel.com/docs/10.x/seeding> (дата звернення: 10.01.2024).
24. Libcamera documentation. URL: <https://docs.arducam.com/Raspberry-Pi-Camera/Native-camera/Libcamera-User-Guide/>
25. MySQL documentation . URL: <https://dev.mysql.com/doc/> (дата звернення: 09.01.2024).
26. PHP documentation . URL: <https://www.php.net/> (дата звернення: 08.01.2024).
27. Raspberry Pi Documentation. URL: <https://www.raspberrypi.com/documentation> (дата звернення: 07.01.2024).
28. Raspberry Pi OS. URL: <https://www.raspberrypi.com/software> (дата звернення: 07.01.2024).
29. Vite Documentation. URL: <https://www.php.net/> (дата звернення: 08.01.2024).