

Національний університет «Полтавська політехніка імені Юрія Кондратюка»

(повне найменування вищого навчального закладу)

Навчально-науковий інститут інформаційних технологій і робототехніки

(повне найменування інституту, назва факультету (відділення))

Кафедра автоматики, електроніки та телекомунікацій

(повна назва кафедри (предметної, циклової комісії))

Пояснювальна записка

до кваліфікаційної роботи

магістр

(освітній рівень)

на тему Модернізація web-застосунку для візуалізації даних системи

«Розумний будинок»

Виконав: студент 6 курсу, групи 601-тг
спеціальності 172 «Телекомунікації та
радіотехніка»

(шифр і назва напрямку підготовки, спеціальності)

Базарний А. В.

(прізвище та ініціали)

Керівник Леві Л.І.

(прізвище та ініціали)

Рецензент Шефер О.В.

(прізвище та ініціали)

Полтава - 2022 рік

Національний університет «Полтавська політехніка імені Юрія Кондратюка»
Інститут Навчально-науковий інститут інформаційних технологій і
робототехніки
Кафедра Автоматики, електроніки та телекомунікацій
Ступінь вищої освіти Магістр
Спеціальність 172 «Телекомунікації та радіотехніка»

ЗАТВЕРДЖУЮ

Завідувач кафедри автоматики,
електроніки та телекомунікацій

_____ О.В. Шефер
“ ___ ” _____ 202_ р.

З А В Д А Н Н Я

НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Базарному Андрію Вікторовичу

1. Тема проекту (роботи) **«Модернізація web-застосунку для візуалізації даних системи «Розумний будинок»»**
керівник проекту (роботи) **Лєві Леонід Ісаакович, д.т.н., професор**
затверджена наказом вищого навчального закладу від “12” 08 2022 року № 544 фа
2. Строк подання студентом проекту (роботи) 07.12.2022 р.
3. Вихідні дані до проекту (роботи) мова програмування PHP, JavaScript, мова стилю сторінок CSS, мова розмітки гіпертексту HTML.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Обґрунтування необхідності модернізації web-застосунку для візуалізації даних системи «Розумний будинок». Огляд програмних засобів для реалізації проекту. Виконання задачі по модернізації веб-застосунку для візуалізації даних системи «Розумний будинок». Висновки по роботі.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових плакатів):
 - 1) Схема роботи веб-застосунку;
 - 2) Схема взаємодії веб-серверу з клієнтом;

- 3) Зображення роботи веб-серверу
- 4) Наявні елементи комплексу «розумний будинок»
- 5) Схематичне зображення взаємодії компонентів системи «розумний будинок»;
- 6) Схематичне зображення взаємодії веб-застосунку із комплексом «розумний будинок»;
- 7) Схематичне зображення функції отримання даних;
- 8) Інтерфейс розробленого веб-застосунку;

6. Дата видачі завдання 01.09.2022 р.

КАЛЕНДАРНИЙ ПЛАН

Пор. №	Назва етапів магістерської роботи	Термін виконання етапів роботи			Примітка (плакати)
1	Огляд принципу роботи та поняття технології “Розумний будинок” . Постановка задачі на магістерську роботу	13.09.22		15%	Пл. 1
2	Огляд принципу роботи веб-застосунків	27.09.22	I	30%	Пл. 2
3	Огляд наявних компонентів системи “Розумний будинок”	10.10.22		40%	Пл. 4
4	Визначення із програмними засобами для реалізації проєкту	17.10.22		50 %	Пл. 5
5	Визначення функціональних можливостей	25.10.22	II	60%	Пл. 6
6	Розробка серверної частини веб-застосунку	01.11.22		70%	Пл. 7
7	Розробка клієнтської частини веб-застосунку	08.11.22		90%	Пл. 8
8	Оформлення магістерської роботи	07.12.22	III	100%	

Магістрант _____ Базарний А.В.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Леві Л.І.
(підпис) (прізвище та ініціали)

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	6
ВСТУП.....	7
1. ПОНЯТТЯ ТЕХНОЛОГІЇ «РОЗУМНИЙ БУДИНОК» ТА ПРИНЦИПУ РОБОТИ ВЕБ-ЗАСТОСУНКІВ	9
1.1 Поняття та основні можливості технології «Розумний будинок»	9
1.2 Поняття та огляд принципу роботи сучасних веб-застосунків.....	16
1.3 Аналіз переваг та недоліків веб-застосунків.....	21
1.4 Висновки	25
2. ТЕХНОЛОГІЇ ДЛЯ ПОБУДОВИ ТА КЕРУВАННЯ СИСТЕМАМИ «РОЗУМНИЙ БУДИНОК».....	26
2.1 Аналіз готових комплексів для побудови системи «розумний будинок»	26
2.2 Аналіз технології зв'язку пристроїв системи «розумний будинок».....	33
2.3 Наявні комплектуючі для системи «Розумний будинок».....	38
2.4 Висновки	45
3. ВИЗНАЧЕННЯ ТА РОЗГЛЯД ЗАСОБІВ ДЛЯ РЕАЛІЗАЦІЇ ПРОЄКТУ	46
3.1 Визначення із програмними засобами для реалізації проєкту	46
3.2 Сервер HTTP.....	49
3.3 Скриптова мова програмування PHP	55
3.4 HTML – мова розмітки гіпертексту	58
3.5 CSS – каскадні таблиці стилів	60
3.6 Скриптова мова програмування JavaScript та технологія AJAX	63
3.7 Висновки	68
4. ПРОЦЕС ВИКОНАННЯ ЗАДАЧІ ПО МОДЕРНІЗАЦІЇ ВЕБ-ЗАСТОСУНКУ ДЛЯ ВИЗУАЛІЗАЦІЇ ДАНИХ СИСТЕМИ «РОЗУМНИЙ ДІМ».....	69
4.1 Загальні функціональні можливості веб-застосунку.....	69
4.2 Налаштування програмного середовища веб-серверу.....	70

4.3 Конфігурація системи управління базами даних MySQL	75
4.4 Реалізація серверної частини веб-застосунку	80
4.5 Реалізація інтерфейсу користувача	87
4.6 Висновки	91
ЗАГАЛЬНІ ВИСНОВКИ	92
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	93
Додаток А	97
Додаток Б	113
Додаток В	118

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ПК – персональний комп'ютер

ПЗ – програмне забезпечення

URL – Uniform Resource Locator – «єдиний вказівник на ресурс»

IoT – internet of things – інтернет речей

PHP – Hypertext Processor

HTML – HyperText markup Language – мова розмітки гіпертексту

HTTP – HyperText Transfer Protocol

AJAX – Asynchronous JavaScript And Xml

ОС – операційна система

СУБД – система управління базами даних

ВСТУП

Розумний будинок являє собою систему домашніх пристроїв, які у свою чергу можуть виконувати певні дії та вирішувати деякі повсякденні задачі без участі людини. Тобто усі електроприлади у будинку функціонально пов'язуються один з одним, що дає змогу централізованого керування, наприклад з пульта-дисплею. Також дані прилади можуть бути під'єднані до мережі інтернет, що надає змогу керувати ними через глобальну мережу, або у локальній мережі за допомогою ПК. Завдяки тому, що інформаційні технології інтегруються у домашні умови, всі прилади та системи виконують функції узгоджено між собою, порівнюючи задані програми та зовнішні показники [1].

Проблемою візуалізації даних із систем типу «розумний будинок» є велика кількість доступних операційних систем для пристроїв користувача, для кожної із них необхідно створювати або адаптувати окремий застосунок. Вирішенням даної проблеми є використання веб-технологій у створенні застосунку для візуалізації даних із системи «розумний будинок».

Веб-застосунок є прикладним програмним забезпеченням, яке запускається у браузері, а не локально у операційній системі пристрою. Веб-застосунки доставляються користувачам через глобальну мережу інтернет [2]. У сучасному світі наявний великий попит на веб-застосунки. Будь-який користувач має змогу використати веб-застосунок маючи лише пристрій із доступом до мережі інтернет, у якому встановлений веб-браузер, завдяки цьому веб-застосунки є невід'ємною частиною інтернет-простору. Значною перевагою у використанні веб-застосунків є те, що немає необхідності у встановленні додаткового ПЗ. Для повноцінного використання усіх функціональних можливостей веб-застосунку необхідно тільки мати зв'язок веб-браузеру та веб-серверу.

Актуальністю даної роботи є створення функціонального веб-застосунку для візуалізації даних системи «розумний будинок». Адже у сьогоднішні все більше

з'являється «розумних» пристроїв, тому є необхідність у зручному інтерфейсі візуалізації даних із таких пристроїв.

Новизна роботи полягає у модернізації веб-застосунку для візуалізації даних системи «розумний будинок», який підтримує мережеві MQTT, WebSocket, HTTP-запити. Завдяки наявній можливості обміну даними із сенсорами та пристроями системи «розумний будинок» із використанням протоколів MQTT, WebSocket, HTTP у користувача з'являється можливість під'єднання не стандартний пристрої до системи, що додає можливість додаткової гнучкості у налаштуванні системи «розумний будинок» в залежності від своїх потреб. При наявності підтримки вищезгаданих протоколів у користувача системи «розумний будинок» з'являється можливість під'єднати до системи візуалізації даних власний DIY (Do It Yourself) пристрій чи сенсор, що дозволить працювати з таким пристроєм, без необхідності у розробці додаткового інтерфейсу візуалізації.

Значною перевагою у застосуванні веб-технологій при розробці застосунку є здатність даного ПЗ виконуватись на різноманітних апаратних платформах або операційних системах, відсутня необхідність у розробці або адаптуванні розроблюваного ПЗ окремо під кожну платформу. Веб-застосунки набагато простіші у розгортанні аніж звичайні локальні застосунки, наприклад після того, як було завершено розробку ПЗ, то немає необхідності у інсталяції даного ПЗ на кожен користувацький пристрій, а необхідно лише повідомити користувачу URL-адресу. Ще однією перевагою є те, що відсутня необхідність у підтримці програмним забезпеченням старих апаратних платформ чи операційних систем. Завдяки тому, що користувачі отримують застосунок із веб-серверу, то завжди існує лише одна актуальна версія застосунку, а отже при оновленні застосунку усі користувачі автоматично отримують його нову версію навіть не помічаючи цього.

1. ПОНЯТТЯ ТЕХНОЛОГІЇ «РОЗУМНИЙ БУДИНОК» ТА ПРИНЦИПУ РОБОТИ ВЕБ-ЗАСТОСУНКІВ

1.1 Поняття та основні можливості технології «Розумний будинок»

Технологія «розумний будинок» являє собою систему домашніх пристроїв, які у свою чергу можуть виконувати певні дії та вирішувати деякі повсякденні завдання без участі у цьому людини. Можемо сказати, що усі електроприлади у приміщенні функціонально зв'язуються один з одним, це у свою чергу дає можливість централізованого керування даними приладами [1].

У зв'язку зі збільшенням обчислювальних можливостей пристроїв технологія «розумний будинок» поступово отримала своє продовження у систему «IoT – інтернет речей», згідно з якою було проведено первинну стандартизацію та окреслені основні рекомендації та правила для реалізації готового продукту, як на рівні окремих компонентів так і на рівні цілісної системи. Хоча і технологія є відносно новою, на ринку уже існують десятки готових рішень.

Основою технології «розумний будинок» є вміння розпізнання конкретних ситуацій, які відбуваються у приміщенні та відповідним чином реагування дані ситуації. Однією із основних особливостей «розумного будинку» є саме об'єднання декількох підсистем у об'єднаний керований комплекс (Рисунок 1.1).

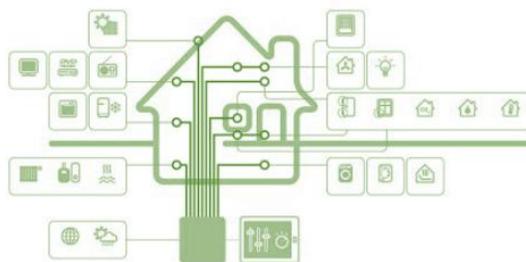


Рисунок 1.1 – Схематичне зображення системи «Розумний будинок»

Однією із найбільш важливих особливостей системи «розумний будинок», що надає їй відмінності від інших видів організації життєвого простору, є те, що дана

концепція є найпрогресивніша у взаємодії людини та житлового простору, а саме це надає можливість задати бажану обстановку лише однією простою командою, а в подальшому вже автоматичні системи відстежують роботу усіх підключених електроприладів та інженерних систем [3]. У даному випадку зникає необхідність при перегляді телебачення у використанні декількох пультів дистанційного керування, великою кількістю вимикачів для управління приладами освітлення, окремими блоками керування у керуванні системами опалення та вентиляції, системами охорони (сигналізація та відеоспостереження) та іншим. При використанні технології «розумний будинок» достатньо навіть одного натискання на пульті дистанційного керування. Після вибору необхідного сценарію система починає самостійно налаштовувати роботу систем у відповідності до вибору людини, яка вибрала сценарій або ж в залежності від зовнішніх чинників таких, як зовнішньої освітленості, вологості повітря, погоди, часу доби [4].

До концепції приміщення із використанням технологій «розумний будинок» входять положення [4]:

- Розробка інтегрованої системи керування приміщенням, тобто системи, яка має можливість забезпечити комплексну роботу усіх систем приміщення, а саме: контроль доступу, вентиляція, освітлення, опалення, водопостачання та безлічі інших.
- Вирішення поставлених сценаріїв та задач підсистемам вбудованої системи керування приміщенням та відсутність у потребі обслуговуючого персоналу. Так званий «інтелект» закладається у вищезгадані підсистеми, який являє собою певний алгоритм дій, який відповідає змінам показників наявних у системі датчиків та решти подій, наприклад позаштатних ситуацій.
- Механізм негайного вимкнення та за необхідності передачі засобів керування людині будь-якою підсистемою інтелектуальної будівлі.

Одночасно із цим повинно бути надано однаковий та зручний доступ до керування і візуалізації усіх частин системи «розумний будинок» для людини.

- Гарантування коректності роботи для конкретних підсистем при випадку відмови загальної системи керування або будь-яких інших елементів системи.
- Вартість модернізації та обслуговування систем приміщення повинна бути мінімальною, що у побудові підсистем повинно забезпечуватись використанням загальних стандартів, автоматичне виявлення та налаштування нових пристроїв у процесі додавання їх до існуючої системи.
- Існування у приміщенні готового комунікаційного середовища, яке необхідне у підключення модулів та пристроїв до неї. Окрім цього, необхідно, щоб була наявна можливість використання різних типів фізичних каналів у якості комунікаційного середовища, наприклад слабкоструміві лінії, радіоканал, силові лінії.

У системі «розумний будинок» наявні такі об'єкти автоматизації [5]:

- Сервер управління;
- Керування освітленням;
- Керування обладнанням із панелей сенсорного керування;
- Керування електроприводами;
- Контроль аварійних станів та електричних навантажень;
- Клімат-контроль;
- Системи для організації контролю доступу;
- Керування системою вентиляції;
- Охоронно-пожежна сигналізація;

- Централізоване керування системами;
- Системи відеоспостереження;
- Мультирум;

Будівля із наявною системою «розумний будинок» має велику кількість переваг. Система керування надає можливість власнику розробити процедури функціонування будь-якої складності, адже усі елементи даної системи мають змогу злагодженої та спільної праці. Прикладом розробки процедур, які зберігають ресурси [6]:

1. Забезпечення безпеки та керування доступом;
2. Управління та облік більшості параметрів систем та у разі їх критичних змін миттєве реагування.
3. Віддалений контроль та керування приміщенням, адже у такій системі є цифровими усі керуючі та інформаційні канали зв'язку.

За допомогою навіть одного дотику є можливість зробити більш комфортним свій будинок, а саме буде встановлено комфортний для конкретної людини мікроклімат, опустяться штори, наповниться ванна, буде включено освітлення. Доступна можливість навіть розігріти їжу, не встаючи зі свого місця, лише необхідно заздалегідь розмістити її на плиті. Доступна можливість керування медіа апаратурою використовуючи сенсорні панелі [6].

У керуванні освітленням наявна можливість не тільки керування яскравістю лампи при включенні а і часу протягом якого буде досягнута задана яскравість лампи. Для цього необхідно використовувати спеціальні світлорегулятори. В основному функція постійного контролю освітлюваності необхідна у використанні для приміщень офісів, що дозволяє в незалежності від зовнішніх факторів підтримувати задану освітленість у приміщенні чи якоїсь поверхні. Можливість увімкнення зовнішнього освітлення автоматично залежно від присутності людей чи часу доби може забезпечити додатковий комфорт мешканцям будинку, а також відлякувати

зловмисників. Для цих цілей необхідно використовувати спеціальні датчики світла, приклад представлено на рисунку 1.2.



Рисунок 1.2 – Датчики світла

Система «розумний будинок» на постійній основі проводить виміри температури у кожній окремій кімнаті індивідуально та виконує певні дії для підтримки її на певному заданому рівні. Для підтримки температури використовується наприклад керування клапанами радіатора або, якщо виникає необхідність автоматично керує системою вентиляції [7]. Окрім цього дана система допомагає власникам приміщення у економії коштів, що досягається завдяки різноманітності режимів роботи: нічний, комфортний або режим коли у приміщенні нікого немає. Відбуватися заміна даних режимів може як за розкладом, так і в ручному режимі за командою. Для цього достатньо задати на сенсорній панелі необхідну температуру відповідно до необхідного режиму. За наявності датчиків на вікнах наявна можливість автоматичного вимкнення системи кондиціонування чи опалення, якщо система визначить, що були відкриті вікна для провітрювання. Це дозволить також заощадити енергію та гроші. Для виконання таких завдань можна використовувати контролер, якій представлений на рисунку 1.3.



Рисунок 1.3 – Контролер для управління температурою Овен TRM 32

У теплу пору року наявна можливість автоматично регулювати жалюзі, що допомагає запобігти потраплянню зайвого сонячного світла до кімнати без зменшення світлового потоку. За рахунок цього приміщення менше нагрівається від сонячного світла та допомагає зекономити кошти за рахунок економії електроенергії, яку витрачає кондиціонер.

За час відсутності власників система «розумний будинок» складе звіт про події, які відбувались на території будинку [5]. Коли відбудеться якась позаштатна ситуація, наприклад протікання води, то система «розумний будинок» виконає необхідні дії для локалізації виниклої аварії, наприклад припинить подачу води та повідомить необхідну службу, а також господаря будинку. При проникненні у приміщення зловмисників система «розумний дім» повідомить власника приміщення всіма можливими каналами зв'язку із ним та здійснить виклик охоронної служби.

Якщо необхідно створити ефект присутності, то система «розумний будинок» також може імітувати звичний спосіб життя під час того, коли власники відсутні. Наприклад вмикати музику та освітлення.

Використання технологій «розумний будинок» надає змогу централізованого інтерактивного керування та контролю різноманітних типах приміщень, таких як офісні приміщення, житлові приміщення та будь-які інші [8].

У загальному вигляді схему системи керування можемо описати наступним виглядом [8]:

1. Головний блок керування
2. Пристрої зчитування показників (датчики освітленості, датчики температури, датчики руху та інші)
3. Пристрої для керування (інфрачервоні емітери, реле та інші)
4. Інтерфейси для керування (сенсорні панелі, веб-інтерфейси, пульти, вимикачі та інші)
5. Мережа керування, яка об'єднує між собою усі елементи, які згадані вище
6. Пристрої, якими виконується керування (світильники, системи вентиляції, медіа компоненти та інші)
7. Допоміжні мережі (для доставки відео та аудіо сигналу, телефонна, Ethernet)
8. Програмне забезпечення

До основних функціональних можливостей центрального блоку керування належить керування пристроями, які йому підпорядковуються. Для керування такими пристроями використовуються такі інтерфейси: IR, Rs-232, RS-485, Ethernet, аналоговий., цифровий та ін. Центральний блок керування містить у собі багатозадачну ОС та у окремих випадках може містити власний веб-сервер. Датчики, які використовуються у системі «розумний будинок» зв'язані між собою через проміжні пристрої або ж безпосередньо між собою єдиною мережею та розташовуються у певних місцях будівлі. Здійснення загального керування комплексом «розумний будинок» відбувається через інтерфейси керування [6].

Опишемо загальний робочий алгоритм комплексу «розумний будинок» [9]:

- До центрального блоку керування надходить інформація від пристроїв вимірювання показників, це відбувається за допомогою власної мережі керування.

- Програмне забезпечення, яке встановлено на керуючому блоці виконує обробку отриманої інформації та повертає команди до пристроїв керування.

Команди для виконання можуть надходити із власної мережі, а також і за допомогою допоміжної мережі. Склад та форма інформації для візуалізації даних, які містять інформацію про стан різноманітних систем, а також методи генерації команд закладаються на етапі розробки ПЗ, враховуючи вимоги заданого проекту.

1.2 Поняття та огляд принципу роботи сучасних веб-застосунків

Більшість користувачів персонального комп'ютеру мають уявлення про те, що являє собою звичайний застосунок, який працює у операційній системі Windows. А саме застосунок це є комп'ютерною програмою, за допомогою якої користувач ПК може вирішувати конкретні необхідні йому прикладні задачі [10]. Прикладом таких програм можуть бути медіаплеєри, веб-браузери, поштові клієнти та багато інших.

Веб-застосунок є розподіленим застосунком, адже логіка його розташовується на сервері, а у свою чергу функція, яка відображає інформацію завантажену із цього серверу полягає на веб-браузер, що встановлений на користувацькому пристрої [11]. Виконуються такі застосунки зазвичай на стороні веб-серверу за рахунок чого створюють інтерфейс між веб-сайтом та користувачем. Розробка веб-застосунків в основному відбувається із використанням скриптових мов програмування таких, як PHP, Perl та інших. Набагато рідше у розробці веб-застосунків використовують мови високого рівня, які необхідно скомпілювати для кожної конкретної операційної системи (C, C++ та інші).

Прийом запитів веб-сервером від клієнтів відбувається за допомогою протоколу HTTP. Після прийому та обробки запиту від клієнту сервер надає відповідь у вигляді HTML-сторінки чи інших даних. Для отримання доступу до необхідної сторінки на веб-сервері, необхідно знати її URL-адресу [12].

Під час розвитку всесвітньої мережі поряд із нею також і веб-застосунки вдосконалювались та розвивались досить стрімко. Однією із найважливіших особливостей веб-застосунку є можливість доступу до нього практично із будь-якого пристрою користувача та будь-якого місця де наявний доступ до інтернету. Ще однією важливою рисою веб-застосунків являється їх гнучкість, забезпечення безпеки та наявність можливостей для їх масштабування. Використання веб-технологій у розробці застосунку дає змогу отримати таку властивість застосунку, як мультиплатформність. Дана властивість програмного забезпечення дає змогу працювати із розробленим застосунком у будь-якому програмному середовищі, тому за допомогою даної властивості можна вирішити проблему із сумісністю ПЗ на різних платформах.

Як було вже з'ясовано веб-застосунок являє собою програму, що працює у веб-браузері, саме тому, для роботи такого застосунку необхідно мати доступ до мережі та мати інсталюваний на користувацькому пристрої веб-браузер. Отже функції обчислень, обробки та зберігання інформації полягають на віддалений веб-сервер, а функції так званого клієнту виконує встановлений на пристрої користувача веб-браузер. На рисунку 1.4 приведено схематично зображений принцип роботи веб-застосунку:

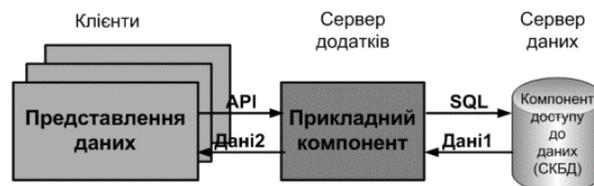


Рисунок 1.4 – Схематичне зображення принципу роботи веб-застосунку

Прикладом веб-застосунку для розгляду може бути широко відомий поштовий клієнт «Gmail». Використовуючи даний веб-застосунок користувач не відчуває принципової різниці у основних функціональних можливостях порівнюючи із класичним клієнтським ПЗ для роботи із електронною поштою, яке необхідно

інсталювати на свій ПК чи мобільний пристрій. Розроблений веб-інтерфейс даного поштового клієнту за допомогою браузерної скриптової мови програмування JavaScript. Завдяки тому, що у розробці веб-інтерфейсу було використано мову JavaScript користувач може у роботі із застосунком відправляти команди керування за допомогою клавіатури, оновлювати сторінки без повного їх перезавантаження (таку можливість надає технологія AJAX), та інші [13].

У створенні сучасних веб-застосунків більшість розробників використовують технологію JavaScript, що надає змогу виконання деяких функціональних можливостей без необхідності у виконанні запитів до віддаленого серверу, що дозволяє зменшити навантаження на сам сервер та пришвидшити виконання певних задач. У період коли мова JavaScript тільки розпочинала свій розвиток, її використовували не для виконання якихсь практичних задач, а лише просто тому, що вона існує. У часи появи даної технології її використовували лише для того, щоб змусити елементи, які містяться на веб-сторінці реагувати на рух миші, динамічно керувати стилями елементів, тощо. Тому у певний час, використання даної технології стало поганим тоном [14].

Причиною стрімкого розвитку популярності веб-застосунків у сьогоденні є початок активного використання технології JavaScript, яка існує у наші часи. Адже раніше завжди вважали недоліком у розробці веб-застосунків їх недостатню динамічність та інтерактивність при взаємодії користувача із веб-сторінкою. Розвиток JavaScript до того рівня, який наявний на сьогоднішній день повністю допомагає вирішити дану проблему.

Отже, як раніше було з'ясовано у стандартному програмному забезпеченні, яке інсталюється на ПК користувачів логіка застосунку розташовується безпосередньо у межах ПК чи іншого користувацького пристрою на відміну від веб-застосунків, де логіка застосунку розташована на віддаленому сервері. Розташування логіки застосунку на віддаленому сервері надає можливість існування єдиної версії

застосунку, яка є завжди актуальною, завдяки даній особливості, застосунок дуже легко поширюється серед усіх його користувачів. Використання веб-технологій у розробці застосунків дозволяє відійти від класичного поширення застосунку, оскільки у такому випадку немає необхідності у отриманні користувачем повної копії застосунку. Завдяки такій особливості, користувачі отримують лише ту частину, яка необхідна їм для роботи із застосунком, а саме інтерфейс програми. Виходячи із того, що сказано вище можна сказати, що при поширенні веб-застосунків існує набагато менше проблем у порівнянні із поширенням класичних застосунків, адже отримати доступ до веб-застосунку можливо із будь-якого пристрою користувача, який під'єднано до мережі інтернет.

Основним компонентом у роботі із веб-застосунком є така програма, як веб-браузер. Для початку роботи із веб-застосунком користувачеві необхідно лише в адресному рядку веб-браузеру вказати необхідну URL-адресу.

Відмінністю класичного застосунку, який користувач інсталує на свій пристрій від веб-застосунку є те, що останній розташовується на віддаленому сервері. Така особливість дає можливість користувачеві не виконувати ролі адміністратора. Наприклад у користувача немає необхідності у здійсненні інсталяції ПЗ, у нього немає необхідності у вирішенні проблем із даним ПЗ, адже роль адміністратора у випадку веб-застосунку виконує людина чи група людей, які безпосередньо виконують розробку застосунку. З одного боку така модель передбачає збільшене навантаження на людей, які розробляють застосунок але з іншого боку, модель коли невелика група програмістів, які займаються розробкою застосунку та зосереджені в одному місці є економічно набагато вигіднішим рішенням. Саме тому це є більш ефективно та більш вигідно у економічному плані, адже у такому випадку відсутня необхідність у утриманні певної команди спеціалістів, що мали б задачу у інсталяції та налаштуванні застосунків на пристроях користувачів.

У використанні веб-застосунку відсутні будь-які вимоги до користувача. У розробці веб-застосунку заздалегідь передбачається, що на пристрої користувача вже інстальований веб-браузер, адже дана програма у сучасному світі є невід'ємною частиною програмного оточення на електронних користувацьких пристроях. Основною ідеєю використання веб-застосунків є те, що з ними можна працювати практично у будь-якій ОС, що встановлена на користувацьких пристроях. Така можливість обумовлена тим, що у своїй роботі у веб-застосунку відсутні особливі вимоги до апаратних платформ. Після початку широкого розповсюдження веб-застосунків у минулому залишилась проблема із необхідністю у підтримці різних версій розроблюваного застосунку. А саме коли розробник застосунку опублікував нову його версію, то без винятку усі користувачі цього застосунку отримують останню опубліковану версію, для цього достатньо виконати перезавантаження веб-сторінки. Одразу після публікації актуальної версії застосунку, зникають усі версії веб-застосунку, які вважаються застарілими. Завдяки такій особливості користувачі застосунку можуть навіть не помітити, що вони вже почали використовувати оновлену версію веб-застосунку. Також завдяки даній особливості веб-застосунків, вирішується проблема у зворотній сумісності застосунків та виконання підтримки не актуальних версій [15].

Одним із оптимальних варіантів використання веб-застосунків є те, коли на пристроях користувача наявний обмежений об'єм накопичувача інформації. Адже очевидна перевага використання веб-застосунку на пристроях користувачів із обмеженим об'ємом накопичувача є те, що при використанні таких застосунків відсутня необхідність у завантаженні на свій пристрій повноцінного застосунку. У деяких випадках навіть відсутня необхідність у повному завантаженні інтерфейсу, а достатньо лише його частини, яка необхідна для виконання поставленого завдання. А, отже веб-застосунки займають мало місця на накопичувачі пристрою користувача, що дозволяє пришвидшити завантаження та реакцію на дії користувача.

Саме завдяки розподіленій архітектурі веб-застосунку, то для звичайного користувача його архітектура є практично невидимою, що надає можливості у масштабуванні обчислювальних можливостей програми без необхідності втручатися у роботу пристроїв користувачів. А саме адміністратор може модернізувати апаратне забезпечення серверу, який містить у собі веб-застосунок непомітно для користувача, розподілити між кількома різними серверами навантаження на застосунок та інше.

При використанні веб-застосунків необхідно розуміти, що вони не можуть мати деяких можливостей, що наявні у класичних настільних застосунків. Наприклад у користувача відсутня можливість керування апаратними компонентами пристрою за допомогою веб-застосунку, працювати зі складною графікою та іншим. У наші часи веб-технології розвиваються досить стрімко, саме тому все вищесказане є актуальним лише на сьогоднішній день та найближче майбутнє.

Спостерігаючи за розвитком веб-технологій, можемо сказати, що однією із найбільш актуальних технологій у майбутньому будуть веб-застосунки. Адже вони кожного дня стрімко розвиваються та максимально наближаються до класичного ПЗ, яке ми використовуємо практично щоденно. Для прикладу такого таких застосунків можемо розглянути ПЗ від компанії Microsoft – «Office Online». Дане ПЗ являє собою пакет офісних застосунків, які розроблені у якості веб-застосунків. Воно по суті є облегшеною версією класичного настільного ПЗ Microsoft Office, та має всі основні можливості його настільного аналога. При використанні Office Online відсутня необхідність у встановленні додаткового ПЗ на пристрої користувачів, саме це надає їм можливість працювати із документами практично із будь-якого місця, де наявне інтернет з'єднання [16].

1.3 Аналіз переваг та недоліків веб-застосунків

Перед початком розробки застосунку для розробника стоїть задача, яка полягає у визначенні із типом розроблюваного застосунку. На даному етапі у розробника наявні декілька варіантів. Одним із таких варіантів є розробка класичного застосунку,

який призначений для конкретної програмної та апаратної платформи, для реалізації якого необхідно використовувати рекомендовані для кожної платформи середовища розробки та компілятори. Альтернативним варіантом є розробка застосунку із використанням веб-технологій, у якому наявна така особливість, як те, що такий застосунок не буде залежати від програмної та апаратної платформи користувачького пристрою. Саме даний етап є одним із найголовніших, адже саме від нього залежать наступні етапи у розробці проєкту. На даній стадії розробки застосунку уже можемо зрозуміти приблизну вартість його розробки, адже на даній стадії виконується залучення працівників, що будуть розроблювати застосунок для обраної платформи.

Веб-технології, які наявні на сьогоднішній день є досить яскравим прикладом, того, що вдало створена технологія може бути застосована людьми у самих найрізноманітніших сценаріях та ситуаціях.

Принципальною різницею веб-застосунку та класичного настільного застосунку є те, що класичний застосунок у своїй роботі взаємодіє із ОС, яка встановлена на пристрої, а веб-застосунок взаємодіє лише із веб-браузером, який встановлений на пристрої користувача. Завдяки такій особливості можна значно розширити кількість користувачів, які зможуть працювати із застосунком, адже у людини яка використовує застосунок на основі веб-технологій з'являється можливість використовувати його практично на будь-якому пристрої, де наявний веб-браузер. При використанні HTML – мови розмітки гіпертексту, у розробника наявна можливість розроблювати користувачький інтерфейс, який має адаптивну розмітку. Це означає, що інтерфейс та його елементи будуть підлаштовуватись під різні розміри дисплеїв на пристроях користувачів. Але хоч і веб-застосунки мають велику перевагу через свою незалежність від програмної та апаратної платформи на яких вони працюють є і достатньо важливий їх недолік – набагато менша продуктивність у порівнянні із класичними застосунками, які є оптимізовані під кожну конкретну платформу.

Класичні застосунки відрізняються від застосунків на основі веб-технологій тим, що такий застосунок завжди зберігається на користувацьких пристроях. У сьогоденні уже більшість користувачів достатньо знайомі із інсталяцією класичних застосунків, але розробники не можуть впевнитися, що усі користувачі будуть працювати із однаковою версією розроблюваного застосунку.

Порівнюючи класичні застосунки та веб-застосунки можемо побачити багато недоліків та переваг використовуючи той чи інший метод розробки застосунку. Через, те, що класичні застосунки можуть напряду використовувати апаратне забезпечення пристрою, для розробника значно спрощується робота із такими компонентами як камера, мікрофон, служби визначення місцезнаходження та інші. Також перевагою класичних застосунків є те, що велику кількість таких застосунків можна знайти в так званих магазинах застосунків. Така можливість надає користувачеві більшої впевненості у безпеці встановлюваного застосунку, адже перед публікацією застосунку на такому майданчику, він проходить перевірку. Поряд із численними перевагами існують також і недоліки у використанні класичних застосунків. Зазвичай на розробку класичного застосунку необхідно більше коштів, що особливо актуально для тих застосунків, яким необхідна можливість працювати на різних апаратних та програмних платформах. Навіть якщо не зважати на вартість розробки такого застосунку, значно дорожче обходиться і його підтримка та оновлення, особливо якщо такий застосунок розроблено під використання на декількох платформах користувацьких пристроїв. Немало важливим є ще такий недолік, як факт того, що користувачі можуть використовувати різні версії застосунку у зв'язку із чим виникають проблеми у розробці, а також сумісності між різними версіями застосунку.

При порівнянні класичного застосунку із веб-застосунком можемо побачити, що працювати із веб-застосунками для користувача набагато простіше, адже у будь-який час та з будь-якого місця користувач може отримати доступ до актуальної версії застосунку, необхідна лише наявність доступу до мережі Інтернет.

У веб-застосунків також наявні свої переваги та недоліки. Можна виділити одну із переваг веб-застосунків це те, що у них наявна спільна кодова база для безлічі пристроїв, а також наявна можливість швидкої адаптації для роботи із неактуальним пристроями, саме через це веб-застосунки є простішими у своєму обслуговуванні. Завдяки тому, у використанні веб-застосунків відсутня необхідність у завантаженні на пристрої користувачів, то можемо сказати що це теж є їх перевагою, яка особливо актуальна для користувачів із обмеженим об'ємом накопичувача. Для розробника не має значення тип пристрою користувача та його платформа, адже веб-застосунок працює у веб-браузері, що дає можливість не зважати на це. Завдяки даній особливості у користувачів наявна можливість використовувати застосунок на будь-яких пристроях, за умови що у них наявний веб-браузер. Логіка застосунку розташовується на віддаленому веб-сервері, а користувач отримує на свій пристрій лише інтерфейс користувача. Така структура застосунку дозволяє виключити проблему сумісності різних його версій, адже дана структурна особливість надає можливість всім користувачам використовувати одну єдину актуальну версію застосунку. На сьогоднішній день уже чимало користувачів користуються постійно різними електронними пристроями, але досвідчених користувачів все ж не дуже велика частина. Саме через це ще однією перевагою використання веб-застосунків є те, що для його використання відсутня необхідність у його інсталяції та налаштування, адже інколи для правильної роботи застосунку необхідно виконувати деякі налаштування, що може бути досить не простим завданням для не досвідченого користувача.

Поряд із перевагами у використанні веб-застосунків наявні також і недоліки. У веб-застосунків наявний досить обмежений доступ до апаратних складових пристрою користувача. Окрім цього для повноцінної роботи веб-застосунку необхідне наявне інтернет з'єднання на пристрої користувача, адже вся логіка застосунку розташована на віддаленому веб-сервері, а користувач завантажує лише інтерфейс користувача.

1.4 Висновки

У даному розділі було розглянуто поняття та основні можливості технології «розумний будинок», поняття та огляд принципу роботи сучасних веб застосунків та аналіз їх переваг та недоліків.

Розглядаючи поняття та основні можливості технології «розумний будинок» можемо зробити висновок, що використання даної технології є перспективним. Завдяки використанню даної технології можна значно спростити виконання повсякденних задач для людини та зробити її проживання у будинку більш комфортним та зручним. Завдяки можливості при використанні технології «розумний будинок» оптимізації роботи різних систем, які використовують енергоресурси власник має можливість суттєво зекономити власні кошти.

Розглядаючи поняття та принципи роботи веб-застосунків було розглянуто їх архітектурну складову та принцип їх роботи. А саме веб-застосунок складається із двох ключових частин – серверна та користувацька частини. Суть користувацької частини полягає у наданні користувачеві інтерфейсу взаємодії із застосунком, тоді як логіка роботи застосунку розташована на віддаленому веб-сервері. Серверна частина містить у собі такі компоненти як програмне забезпечення веб-серверу, яке приймає запити до нього за протоколом HTTP та логіка застосунку, що написана на певній мові програмування.

У використанні веб-застосунків наявні переваги такі, як відсутня проблема у закритих фаєрволем портах, адже в основному такі порти, які необхідні для роботи веб-застосунків не закриті за допомогою фаєрвола [17]. А отже основна перевага веб-застосунків це можливість працювати на різноманітних програмних та апаратних платформах без необхідності у додатковій адаптації застосунку.

2. ТЕХНОЛОГІЇ ДЛЯ ПОБУДОВИ ТА КЕРУВАННЯ СИСТЕМАМИ «РОЗУМНИЙ БУДИНОК»

2.1 Аналіз готових комплексів для побудови системи «розумний будинок»

Керування розумним будинком за допомогою віддаленого доступу чи спеціального центрального пульта керування називають екосистемою. Існує декілька варіантів [18]:

1. Збірка схеми пристроїв системи розумний будинок своїми руками за допомогою використання різноманітних хабів; ручний підбір та налаштування конкретного свого плагіну для кожного окремого пристрою; управління із використанням саморобних застосунків.

2. Придбання готового стартового набору, потім придбання необхідних для даної системи розумний будинок пристроїв, які сумісні із придбаним стартовим набором та виконати підключення цих пристроїв до хабу, інсталювати на свій пристрій застосунок для керування функціями системи розумний будинок.

Якщо користувач скористається другим варіантом, то у такому разі він отримає уже готове рішення без необхідності прикладати додаткові зусилля для організації роботи пристроїв між собою. Взагалі сума вкладень у таку готовий набір системи розумний будинок може бути порівняно не високою. Для організації системи розумний будинок, більшості може бути достатньо придбати лише набір, який може включати в себе пристрої для контролювання пожежної безпеки, пристрої для виявлення витoku води, пристрої для виявлення несанкціонованого проникнення до приміщення.

У сьогоднішні користувач може придбати різноманітні системи управління розумним будинком, які використовують різні технології об'єднання та керування пристроями у системі. Нижче розглянемо декілька популярних готових рішень системи розумний будинок та проведемо аналіз їх основних переваг та недоліків [14].

Ajax StarterKit (рисунок 2.1). Даний стартовий набір системи розумного будинку розроблено українським брендом Ajax. У даному наборі присутні основні пристрої для впровадження контролю безпеки у приміщенні, що дозволяють сповістити власника приміщення про проникнення до будинку чи наприклад витік води. До даного стартового набору входять такі пристрої та датчики:

- Центральний контролер
- Бездротовий брелок
- Бездротовий датчик відкриття
- Бездротовий датчик руху



Рисунок 2.1 – Стартовий набір системи «розумний будинок» Ajax StarterKit

Переваги:

- Захищений радіоканал
- Просте налаштування та керування
- Швидкість оповіщення
- Наявність резервного каналу живлення блоку керування

До недоліків належить наявність лише охоронного функціоналу.

Вартість такого набору на даний момент становить 8400 грн.

Хіаомі – добре відома китайська компанія, яка випускає безліч різноманітної та високоякісної електроніки. Екосистема даного виробника насправді є досить великою та не обмежується пристроями, які наявні у базовому наборі, адже дана компанія є виробником великої кількості різних «розумних» пристроїв. Вони є досить популярними завдяки своїм доступним для покупця цінам. Така особливість даного виробника дозволяє створити систему розумного будинку з досить широкими функціональними можливостями.

Шлюз Міїа (рисунок 2.2) – багатофункціональний пристрій, що поєднує у собі як функції хабу, так і функціональні можливості нічника, радіоприймача тощо. Для відстеження несанкціонованого проникнення до приміщення використовуються датчики відкриття вікон та дверей, які надають можливість при несанкціонованому доступі до приміщення надіслати повідомлення власникові та увімкнути камеру відеоспостереження. Можна налаштувати систему так, щоб наприклад вранці при відкритті вікна автоматично вимикався кондиціонер. Бездротова кнопка (вимикач) дає можливість одним натисканням на неї виконати якусь певну дію наприклад виходячи із думівки можна одним натисканням на кнопку знеструмити електричні прилади.



Рисунок 2.2 – Набір системи «розумний будинок» Хіаомі Міїа

Також важливою складовою системи «розумний будинок» є «інтелектуальна» розетка. Після підключення такого пристрою як «інтелектуальна» розетка до шлюзу системи «розумний будинок» з'являється можливість запрограмувати сценарій вимкнення та увімкнення електричних пристроїв, які під'єднані до такої розетки. Наприклад можна налаштувати автоматичне керування живленням пристроїв за часом доби. За допомогою наявного датчика руху доступна можливість налаштувати сценарії керування наприклад освітленням в залежності від присутності людини у даному приміщенні.

Вартість такого набору на даний момент складає 2100 грн.

Переваги:

- Можливість у реалізації великої кількості сценаріїв лише при використанні базового набору
- Сумісність застосунку для керування із ОС Android та IOS
- Відносно не висока вартість

Недоліки:

- Для використання «розумних» розеток необхідно придбати перехідник на європейський стандарт розеток, адже вони розроблені під стандарти китайських розеток
- При використанні шлюзу у режимі радіоприймача він може приймати лише радіостанції Китаю.

Redmond. Відмінною рисою даного бренду являється широким вибором пристроїв, якими можна керувати із одного застосунку. Оцінюючи кількість наявного обладнання для організації системи «розумний будинок» можемо сказати, що виробник займає лідируючі позиції. У стартовий набір входить датчик руху,

«розумна» розетка та датчик відкриття дверей (рисунок 2.3). Решту необхідних користувачеві пристроїв необхідно купувати окремо.



Рисунок 2.3 – Елементи системи «розумний будинок» Redmond

Вартість даного базового комплекту системи «розумний будинок» становить 3900 грн.

Переваги:

- Велике різноманіття «розумних» пристроїв
- Наявність можливості повної автоматизації житлового приміщення
- Надійність та якість

Недоліки:

- Необхідність у обов’язковій наявності у користувача пристрою для управління системою «розумний будинок». Таким пристроєм може бути смартфон або планшет.
- Відсутня можливість керування голосом

Google – всесвітньо відомий американський бренд. Концепція керування системою «розумного будинку» являється керування за допомогою голосу. Має свій віртуальний асистент, який викликається за допомогою голосової команди «Окей Гугл», після чого асистент готовий виконати команди, які зможе розпізнати. Наприклад увімкнути музику, створити нагадування, завести будильник тощо. Керування системою відбувається за допомогою пристрою, що являє собою музичну колонку, який є єдиним пристроєм системи «розумний будинок», що виготовляє безпосередньо Google. Решта компонентів із яких складається система «розумний будинок» розробляються та виготовляються іншими відомими компаніями такими як TP-Link, Philips або ж Xiaomi. Завдяки даній особливості кількість сумісних пристроїв є достатньо велика для створення системи «розумний будинок».

У середньому вартість колонки Google Home становить 4000 грн. Для створення повноцінної системи «розумного будинку» необхідно докупляти окремо усі необхідні прилади та датчики, які виробляються сторонніми компаніями. Розрахунок вартості комплекту, який включає в себе можливість регулювати освітлення, охоронні функції та інше необхідно виконувати для кожного окремого випадку.

Переваги:

- Наявність можливості керування функціями «розумного будинку» голосом
- Велика кількість сумісного обладнання, які виготовляють різні виробники
- Наявність «розумної колонки»

Недоліком даної системи є її досить висока вартість.

Amazon. Так само, як і Google є американською компанією та є найближчим конкурентом системи «розумний будинок» Google Home. Так само як і в системі від Google у Amazon так само екосистема «розумного будинку» побудована навколо «розумної» колонки власного виробництва, а інші необхідні функціональні пристрої та елементи для побудови повноцінного комплексу «розумний будинок» виробляють

інші відомі виробники електронних пристроїв. Така особливість дозволяє користувачеві полегшити його зусилля у збиранні необхідного набору компонентів для реалізації повноцінної системи «розумний будинок», а також надає можливість у подальшому розширити можливості своєї системи за допомогою нових елементів. Завдяки високій якості, досить не високих цін, а також широкій сумісності, система від Amazon вважається однією із найгнучкіших та перспективних систем.

Концепція системи «розумний будинок» від компанії Amazon така ж як і від Google – виробник випускає лише саму «розумну» колонку, яка являє собою своєрідний хаб. У середньому ціна колонки становить 4300 грн. Решта компонентів, які можуть бути необхідними у налаштуванні системи «розумний будинок» виготовляється сторонніми виробниками.

Переваги:

- Наявність задати безліч сценаріїв для автоматизації
- Велика кількість сумісного обладнання від різних виробників
- Наявність голосового керування

До недоліків можна віднести відсутність підтримки української мови.

Порівняємо більш наочно характеристики вищеописаних систем у таблиці 2.1

Таблиця 2.1 – Характеристики базових наборів системи «розумний будинок»

Система	Ajax StarterKit	Xiaomi Mijia	Redmond	Google Home	Amazon
Простота у налаштуванні	+	+	+	+	+
Відкритість системи	-	+	-	+	+

Продовження таблиці 2.1

Мобільний застосунок	+	+	-	+	+
Веб-інтерфейс	+	+	+	+	+
Голосове керування	-	-	-	+	+
Вартість базового набору, грн	8400	2100	3900	4000	4300

Розглянувши варіанти готових програмних та апаратних рішень від різних виробників найбільш оптимальними варіантами є продукти компаній Google та Amazon. Дані продукти є найбільш оптимальними при виборі готових систем «розумного будинку» завдяки наявності у них більш ширшого функціоналу серед подібних готових систем, простоті у додаванні нових модулів та елементів системи, а також простоті їх налаштування. Але через те, що ціна на обладнання цих виробників є досить високою та через необхідність у створенні своєї системи за допомогою компонентів від різних виробників, то такі варіанти є не досконалими.

2.2 Аналіз технології зв'язку пристроїв системи «розумний будинок»

У сьогоднішній більшості популярних пристроїв, які використовуються для створення середовища системи «розумний будинок» базуються на технологіях зв'язку таких як Wi-Fi, Bluetooth, ZigBee та Z-Wave [19]. Кожна окрема така технологія зв'язку має свої переваги та недоліки, однак для компенсації недоліків однієї технології можна використовувати декілька технологій, що дозволить компенсувати недоліки кожної окремої технології. Втім у різних типах пристроїв для створення

«розумного будинку», що дозволяють виконувати різноманітні завдання використовуються різноманітні технології зв'язку.

Для прикладу розглянемо стандартну побутову техніку, таку як телевізор, холодильник та інші. У домашній побутовій техніці зазвичай використовують дві технології зв'язку – Wi-Fi та Bluetooth. Дані технології також можемо зустріти у кожному смартфоні. Причиною використання даних технологій полягає у тому, що даними побутовими приладами люди користуються в не залежності від наявності комплексної системи «розумний будинок». У випадках, коли необхідно виконати задачі з автоматизації керування системами освітлення або контролю системи клімату доцільно використовувати модулі системи «розумний будинок», які використовують технології зв'язку Z-Wave або ZigBee. Дані технології зв'язку розроблені спеціально для використання у пристроях, які будуть інтегровані із наявними світловими приладами та приладами контролю клімату, завдяки цьому вони є набагато більш енергоефективними ніж Wi-Fi та Bluetooth. Але важливо зауважити, що для роботи таких пристроїв необхідний спеціальний хаб.

1) Wi-Fi. При використанні пристроїв, які мають потребу у передачі «важкої» інформації (IP-камери, телевізори, аудіо техніка та інші) необхідно використовувати технологію зв'язку Wi-Fi, що зумовлено можливістю передавати дані за цією технологією достатньо швидко [20]. Звісно ж дану технологію можна використовувати також і у різноманітних приладах таких як датчики, вимикачі, термостати, але це не є доцільним рішенням. Пристрої, які використовують технологію зв'язку Wi-Fi мають високе споживання енергії, через що дані пристрої не зможуть довго працювати від батарейки. Недоліком використання пристроїв із даною технологією зв'язку ще є те, що кожен виробник свого пристрою також працює із своїм власним ПЗ для керування їх приладами. При використанні технології зв'язку Wi-Fi відсутні уніфіковані стандарти, які надають можливість керування «розумними» пристроями із використанням одного застосунку. Завдяки вище

згаданим недолікам можемо зробити висновок, що на даний момент немає можливості створити систему «розумний будинок» комфортну для користувача використовуючи лише технологію зв'язку Wi-Fi.

2) Bluetooth. На теперішній час актуальною версією стандарту технології зв'язку Bluetooth є версія 5.3 Low Energy [21]. Завдяки тому, що пристрої, які використовують дану технологію мають не високе енергоспоживання. Завдяки такій особливості технологія Bluetooth набула масового розповсюдження у маленьких пристроях таких бездротові навушники та різноманітні датчики, які живляться від батарейок. Проблеми у використанні даної технології такі ж як і у технології Wi-Fi, а саме відсутність єдиного стандарту у керуванні пристроями, що змушує виробників розробляти для своїх пристроїв власний застосунок, який зазвичай не сумісний із пристроями інших виробників, що викликає неприємний досвід у користувача при використанні таких систем. У версії Bluetooth 5.0 з'явилася технологія чарункової мережі (Mesh), яка є важливою для створення систем «розумний будинок», але до сьогодні вона мало де використовується у пристроях для системи «розумний будинок», але можливо у майбутньому дана технологія стане більш розвиненою та буде основою для «побудови» системи «розумний будинок».

3) ZigBee. Спочатку дана технологія зв'язку розроблялася для обміну даних у мережах із приладів, що являють собою датчики, наприклад газові лічильники, лічильники використання електроенергії та води, датчики відкриття та температури та інші. При використанні даної технології зв'язку топологія створеної мережі бути виконана у різні способи, наприклад комірчаста топологія (mesh). Дана особливість надає можливість кожному пристрою зв'язуватись із будь-яким іншим пристроєм у мережі та передавати сигнал далі, використовуючи сусідній пристрій, а саме виконувати ретрансляцію, завдяки чому збільшується загальна надійність створеної мережі. У технології зв'язку ZigBee наявний уніфікований стандарт команд для керування системою «розумний будинок», який було запроваджено у 2007 році [22].

На сьогоднішній день більшість пристроїв, які використовуються у створенні систем домашньої автоматизації використовують технологію ZigBee, наприклад лампи, замки, датчики та інші. Разом із тим у побутовій техніці такої як телевізори та холодильники дана технологія не використовується. Порівнюючи пристрої, які використовують технологію ZigBee та інші технології, то можемо сказати, що у пристроїв із ZigBee ціни на них є нижчими у порівнянні із конкурентами, які використовують інші технології зв'язку. Але незважаючи на усі переваги у використанні даної технології зв'язку, у користувача відсутня можливість «побудувати» систему «розумний будинок» із використанням лише пристроїв на технології ZigBee, адже у таких пристроїв відсутня повна сумісність при використанні модулів та хабів від різних виробників.

4Z-Wave. Дану технологію зв'язку було розроблено спеціально для використання у пристроях, які призначені для автоматизації, тобто для створення системи «розумний будинок». Однією із головних особливостей та переваг даної технології є наявність повної сумісності між різними пристроями від різних виробників. Наприклад датчик руху від одного виробника може керувати димером, який випустив інший виробник, а виконання автоматизованих задач буде забезпечувати контролер RaZberry. На сьогоднішній день у продажі наявні безліч різноманітних пристроїв, які використовують технологію зв'язку Z-Wave, що дозволяє задовольнити практично будь-які потреби у створенні системи «розумний будинок». У побудові систем «розумний будинок» у приміщенні, яке має площу до 500 м² найбільш поширена технологія Z-Wave [23]. Технологія зв'язку Z-Wave також використовує комірну топологію (Mesh), що надає можливість для ретрансляції сигналу та знаходження найбільш оптимального маршруту у автоматичному режимі. При обмеженому бюджеті на «побудову» системи «розумний будинок» недоліком буде досить висока ціна на пристрої, які використовують дану технологію. Середня

вартість пристроїв становить приблизно 70-90 доларів, що є дорожче у порівнянні із аналогами, які розроблені із використанням технології ZigBee.

У таблиці 2.2 приведемо порівняння технологій зв'язку, які використовуються у «побудові» системи «розумний будинок».

Таблиця 2.2 – порівняння технологій зв'язку системи «розумний будинок»

Технологія	Де використовується	Наявність уніфікованого стандарту керування приладами	Вартість
Wi-Fi	ІР-камери, телевізори, аудіо/медіа плеєри та інші	Відсутній	Середня
Bluetooth	Бездротові навушники, колонки та різноманітні датчики на батарейках	Відсутній	Середня
ZigBee	У мережах з датчиків (лічильники електроенергії, води, газу), датчики температури та інші	Частково	Низька
Z-Wave	У більшості приладів, які використовують у «побудові» системи «розумний будинок»	Наявна повна сумісність	Висока

2.3 Наявні комплектуючі для системи «Розумний будинок»

Перед «побудовою» системи «розумний будинок» необхідно визначити задачі, які необхідно буде вирішувати за допомогою побудованої системи. У даному випадку буде розглянуто прилади, які необхідні для створення базової системи «розумний будинок» із можливістю контролю безпеки та частковою автоматизацією.

Одним із найпростіших пристроїв із яких можна почати «побудову» системи «розумний будинок» являє собою «розумна» лампочка. У даному випадку наявний пристрій NiteBird WB4, що являє собою «розумну» лампочку, який представлений на рисунку 2.4.



Рисунок 2.4 – «Розумна» лампа NiteBird WB4

Завдяки тому, що такі лампочки мають можливість підключити до загальної електромережі приміщення це дає можливість здійснити досить просте встановлення такого пристрою, його достатньо лише вкрутити у патрон. Такі лампи дають можливість користувачеві регулювати параметри випромінюваного світла, а саме яскравість та колір світла. Завдяки використанню у лампах даного типу світлодіодних

елементів вони споживають набагато менше електроенергії та мають більший строк служби відносно звичайних ламп розжарювання. Задля того, щоб збільшити «розумні» можливості наявної лампи, можемо додати до даної системи датчик руху.

Також для отримання даних про температуру та вологість повітря передбачимо відповідні датчики. Для того, щоб не використовувати два окремих датчики візьмемо пристрій Xiaomi Mi Temperature and Humidity Monitor, який містить у собі і датчик температури і датчик вологості. На рисунку 2.5 представлено датчик вологості та температури Xiaomi.



Рисунок 2.5 – Датчик вологості та температури Xiaomi Mi Temperature and Humidity Monitor

Наступними компонентами будуть датчики відкриття. У даному випадку встановлено датчики відкриття Perenio PECWS01, який представлено на рисунку 2.6.



Рисунок 2.6 – Датчик відкриття вікон або дверей Perenio PECWS01

Даний датчик являє собою контактний сенсор, який необхідно закріпити на віконних рамах та на дверях. Основною задачею цього датчика є задача повідомити про несанкціоноване проникнення до приміщення через вікна або двері на яких даний датчик встановлений. Але за допомогою даного датчика наявна можливість не лише виконувати охоронні функції, а і наприклад можна вимкнути кондиціонер при відкритті вікна. За допомогою датчика відкриття є можливість налаштувати сповіщення власнику приміщення при несанкціонованому проникненні до нього, а також увімкнути камеру відео нагляду та активувати звукову сигналізацію для того щоб привернути увагу.

Для забезпечення протипожежної безпеки необхідно встановити у приміщенні датчик задимлення. У нашому випадку встановлено датчик задимлення Perenio PECSS01, який представлений на рисунку 2.7



Рисунок 2.7 – Датчик задимленості Perenio PECSS01

При виявленні датчиком диму задимленості у приміщенні, він подає команди у систему керування «розумного будинку», а та у свою чергу приймає рішення про подальші дії. Наприклад можна налаштувати при виявленні диму відключення усіх електроприладів та надіслати сповіщення власнику. Також даний датчик може працювати у автономному режимі. У такому режимі роботи у нього наявна можливість при виявленні задимлення подавати звукові сигнали для того, щоб привернути увагу.

Для забезпечення виявлення протікань води необхідно встановити датчики протікання. Датчики протікання доцільно встановлювати у кімнатах, де безпосередньо найбільший ризик протікань, а саме у ванній кімнаті, на кухні під мийкою, біля пральної машини. На рисунку 2.8 представлено датчик протікання води Ezviz T10.



Рисунок 2.8 – Датчик протікання води Ezviz T10

При реалізації системи «розумний будинок» важливим елементом є датчик руху. Задачею датчика руху є виявлення у приміщенні де він встановлений присутність людини чи тварини. На рисунку 2.9 представлено датчик руху Номтун MS-20-Z. Даний датчик є простий у встановленні, прикріпити до необхідної поверхні його можна лише за допомогою одного шурупа або приклеїти. Завдяки своєму низькому енергоспоживанню такий датчик може працювати приблизно 2 роки без необхідності у заміні батарейки. Перевагою даного датчика також є його компактність, що робить його практично непомітним.



Рисунок 2.9 – Датчик руху Номтун MS-20-Z

В основному задачею системи «розумний будинок» являються дії для відключення електроприладів від мережі. Через це існує необхідність у передбаченні «розумних» розеток. Зазвичай вони представлені у формі так званого перехідника від

вмонтованої у кімнаті розетки до електроприладу користувача, завдяки чому відсутня необхідність у докладанні зайвих зусиль для монтажу. Для прикладу розглянемо «розумну» розетку TP-Link HS100, яка представлена на рисунку 2.10. Дана розетка має значно ширші функціональні можливості, а ніж просто перемикання по команді від користувача. А саме при використанні даної розетки наявна можливість керувати живленням електроприладів, які підключені до неї за таймером, а також наявна функція по збору статистичних даних по електроспоживанню.



Рисунок 2.10 – «Розумна» розетка TP-Link HS100

Станом на сьогоднішній день існують також побутові прилади (чайник, мультиварка та інші) у яких наявна можливість вбудовування в систему без використання «розумних» розеток. Прикладом таких побутових приладів можуть бути прилади від виробника Redmond.

Також популярним пристроєм у системі «розумний будинок» є електрокарниз.

Приклад схеми його монтажу приведено на рисунку 2.11.

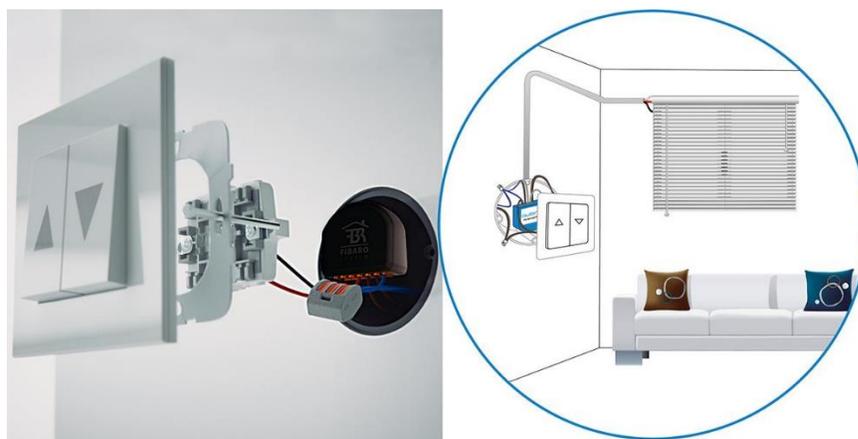


Рисунок 2.11 – Принцип встановлення електричного карнизу

Для керування шторами або жалюзі необхідний спеціальний модуль, який являє собою мотор. Для керування таким мотором необхідні спеціальні мікромодулі, які представлені різними виробниками, наприклад Zipato та Qubino. Завдяки своїй більшій функціональній можливості краще обрати модуль Qubino Shutter. У даного модуля наявна можливість у проведенні вимірів електроспоживання та наявна можливість автоматичного виявлення крайніх положень жалюзі. Також присутня можливість синхронізації із будильником.

Як уже було з'ясовано існує велика кількість виробників пристроїв для створення системи «розумний будинок» та всі вони у виробництві своїх пристроїв використовують різноманітні технології зв'язку. Для того, щоб з'явилась можливість об'єднання таких пристроїв у одну єдину систему наявна необхідність у додатковому пристрої, який являє собою шлюз. Основною задачею шлюзу являється виконувати роль так званого центру створеної мережі. Завданням шлюзу є збір різноманітних даних із сенсорів та пристроїв та передачу їх наприклад у хмару, для подальшого доступу до них із смартфона чи веб-інтерфейсу. А також шлюз виконує функції комутації команд між пристроями. Уніфікований стандарт для створення можливості об'єднання пристроїв від різних виробників в одну мережу відсутній.

2.4 Висновки

У даному розділі було виконано аналіз уже комплексно готових рішень для побудови системи «розумний будинок». Розглядаючи готові рішення програмних та апаратних комплексів можна виділити такі варіанти, як Google Home та Amazon. Дані рішення є найбільш простими у встановленні додаткових модулів та налаштуванні. Але вибір таких готових систем не є ідеальним, адже вони є достатньо коштовними та при створенні своєї системи «розумний» будинок, користувачеві необхідно комбінувати модулі від різних виробників, що не є достатньо зручним та гнучким варіантом.

Було розглянуто технології бездротового зв'язку, які використовуються при реалізації проектів «побудови» систем «розумний будинок». Було з'ясовано, що при «побудові» системи «розумний будинок» найбільш зручним рішенням є використання технології Z-Wave, адже на ринку представлено велику кількість різноманітних пристроїв від різних виробників для створення системи «розумний будинок». У даній технології наявний уніфікований стандарт, що дає повну сумісність пристроїв від різних виробників між собою.

Також було розглянуто деякі пристрої, які необхідно використовувати для побудови базової системи «розумний будинок» із функціональними можливостями охороно та часткової автоматизації.

3. ВИЗНАЧЕННЯ ТА РОЗГЛЯД ЗАСОБІВ ДЛЯ РЕАЛІЗАЦІЇ ПРОЄКТУ

3.1 Визначення із програмними засобами для реалізації проєкту

У сьогоднішній інформаційній технології розвиваються досить швидко, завдяки цьому для більшості людей стали доступними для користування такі пристрої, як смартфони, ПК та ноутбуки. Через те, що різноманітні електронні пристрої набули широкого розповсюдження у розробника застосунків з'явилася необхідність при розробці застосунків роботи їх доступними для використання на декількох видах пристроїв. Головною проблемою у розробці таких застосунків, що можуть виконуватись на різних програмних та апаратних платформах є велике різноманіття таких платформ. Дану проблему може вирішити така властивість, як кросплатформність [24].

Властивість програмного забезпечення, за допомогою якої досягається робота розроблюваного застосунку більше ніж на одній програмній та апаратній платформі називається кросплатформністю. Прикладом такого програмного забезпечення, що використовують дану властивість можуть бути застосунки, які розроблені для виконання одночасно на декількох ОС, наприклад Linux та Windows.

Розглядаючи більшість актуальних на теперішній час високорівневих мов програмування можна сказати, що вони є кросплатформні. Кросплатформність у таких мовах програмування забезпечується так званими перекладачами, що дозволяє застосунку виконуватись на різних програмних платформах. Для приведення прикладу розглянемо мови програмування такі, як C, C++ та Pascal. Дані мови програмування досягають властивості кросплатформності на рівні компіляції, тобто можемо сказати, що для різних програмних та апаратних платформ існують компілятори коду. Іншим прикладом кросплатформних мов програмування можна представити такі мови, як C# та Java, адже їх властивість кросплатформності досягається на так званому рівні виконання [25]. Така особливість означає, що для того, щоб виконати розроблене ПЗ із використанням даних мов програмування,

відсутня необхідність у перекомпіляції коду програми для різних платформ. Однак для того, щоб мати можливість виконання таких застосунків інколи наявна необхідність, щоб на пристрої користувача було встановлено додаткове програмне забезпечення. Прикладом такого додаткового програмного забезпечення може бути середовище виконання Java Virtual Machine, яке необхідно мати встановленим на пристрої користувача, щоб запускати застосунки написані із допомогою мови програмування Java.

При виконанні завдання даної дипломної роботи, будуть використані веб-технології. Завдяки тому, що при виконанні даної задачі будуть використовуватись веб-технології, розроблений в результаті застосунок буде мати змогу виконуватись на різних програмних та апаратних платформах, що дозволить людині використовувати дане ПЗ практично на будь-якому користувацькому пристрої. При використанні даного типу застосунків передбачено, що на користувацькому пристрої уже встановлений веб-браузер. У розробці веб-застосунку основною перевагою є те, що для його використання на різних платформах відсутня необхідність у проведенні компіляції коду застосунку для кожної платформи окремо. Ще однією вагомою перевагою використання веб-застосунків є те, що при його використанні у користувача відсутня необхідність у інсталяції додаткових компонентів у пристрій користувача, адже для роботи із веб-застосунком необхідний лише один компонент, тобто веб-браузер, який зазвичай інстальований за замовчуванням у практично усіх популярних ОС.

При написанні серверної частини веб-застосунку буде використано скриптову мову програмування PHP [26]. Мова програмування PHP набула своєї популярності завдяки її простому синтаксису, завдяки чому навіть початківець у програмуванні може достатньо швидко опанувати дану мову програмування. Дана мова програмування із самого початку її розвитку розроблювалася для використання при написанні веб-сторінок, а пізніше і веб-застосунків. Завдяки початковій орієнтації

даної мови програмування на веб, у ній було передбачено деякі особливості, які дозволяють більш ефективно працювати у парі із мовою розмітки HTML [27]. Для прикладу приведемо таку властивість, як те, що у розробника наявна можливість вставляти код написаний на мові PHP безпосередньо у HTML сторінку, що не буде викликати жодних проблем у обробці інтерпретатором PHP. Дана особливість надає більше можливостей для зручного використання мови програмування PHP при формуванні HTML сторінок, завдяки чому з'являється можливість надати більшій гнучкості та динамічності веб-сторінкам.

Найбільш важливим компонентом у роботі веб-застосунку являється веб-сервер. Розглядаючи апаратну складову веб-серверу, можна зрозуміти, що він по суті є таким же, як і звичайний домашній комп'ютер, який має підключення до мережі інтернет та за допомогою даної мережі виконує дії для доставлення файлів сторінок, що зберігаються на сервері до пристроїв користувачів. Здебільшого різницею веб-серверу та ПК спеціальне програмне забезпечення. Наприклад для роботи веб-серверу необхідне ПЗ, яке виконує контроль доступу користувачів веб-застосунку до файлів даного застосунку, які зберігаються на веб-сервері. Головною складовою програмного комплексу веб-застосунку є такий компонент, як HTTP-сервер. За допомогою даної програмної складової веб-сервер насамперед отримує можливість працювати із протоколом HTTP та коректно опрацьовувати запити за допомогою URL-адрес.

Для реалізації даного проєкту буде використано ПЗ веб-серверу під назвою Apache HTTP-сервер. Насамперед дане ПЗ веб-серверу являється одним із найпопулярніших у світі. Особливістю його є те, що даний продукт має відкритий початковий код, а також є безкоштовним для використання. При використанні Apache у розробника відсутня необхідність у обмеженні виборі операційної системи на якій буде працювати веб-сервер, адже він може бути встановлений на більшість популярних ОС, таких як Linux, Mac OS, Windows та інших [28].

Станом на сьогоднішній день переважна більшість веб-застосунків використовує спеціальні бази даних, для зберігання та використання необхідної інформації. Можливість використання баз даних з'явилася після появи мови програмування PHP. Така нова можливість при розробці веб-сайтів надала можливість розробникам у розділенні контенту та каркасу сторінок веб-сайту, адже тепер наявна можливість динамічно завантажувати необхідний контент за допомогою запиту до БД. Завдяки використанню такого підходу при розробці веб-застосунку для розробника стає легшим адміністрування ресурсу та подальшу його модернізацію. При використанні баз даних відбувається значне полегшення у вирішенні питань для сортування, пошуку та пагінації даних. При виконанні даного проєкту буде використано систему керування базами даних MySQL. СУБД MySQL являється однією із найбільш популярних та універсальних СУБД. Перевагами використання MySQL є її висока швидкість обробки даних, вона відкритий початковий код та є безкоштовною для використання у своїх проєктах. Також при використанні даної СУБД наявна можливість використання зручного веб-інтерфейсу для адміністрування баз даних, який має назву phpMyAdmin.

3.2 Сервер HTTP

Для роботи будь-якого веб-застосунку необхідний веб-сервер, адже він є основною складовою у роботі веб-застосунків.

Будь-який веб-сервер складається із двох складових, а саме із апаратної складової та програмної. Розглядаючи апаратну складову веб-серверу, можемо зрозуміти, що з точки зору «заліза», він являє собою звичайний ПК, який виконує функції зберігання таких складових веб-застосунку, як HTML сторінки, файли таблиць стилів CSS, зображення та інші. Але у обов'язки веб-серверу входить не лише зберігання файлів, а також їх доставка до пристроїв користувачів. Для можливості віддаленого доступу до застосунку, який міститься на веб-сервері, він має бути підключений до глобальної мережі інтернет та для зручності використання має бути

наявне для використання доменне ім'я, наприклад example.com.ua. Розглядаючи програмну складову веб-серверу, можемо побачити, що вона являється програмним комплексом, який складається із декількох компонентів, які виконують функції для контролю доступу до ресурсів, які розташовані на сервері та інші. Головною складовою у програмному комплексі веб-серверу є HTTP-сервер, що являє собою компонент, який виконує роботу із мережевим протоколом HTTP та опрацьовує URL-запити.

Основним завданням веб-серверу є надання відповіді за запитом по URL-адресі. У відповіді може міститись запитуваний файл, наприклад HTML сторінка, зображення та інше. Також може бути виконаний запит на виконання так званої програми, у результаті чого після звернення користувача до програми або скрипту на сервері він отримає у відповідь, яка містить результат виконання програмного коду, який був запитаний. Дані функції є основними частинами при роботі веб-серверу, усе інше являє собою допоміжні складові.

Для взаємодії користувача та веб-серверу, необхідно запустити веб-браузер на користувацькому пристрої та звернутись до серверу за допомогою використання URL-адреси, наприклад <https://mon.gov.ua>. Після введення URL-адреси у відповідному полі у браузері, він надсилає запит до серверу, на який вказує введений URL. Обов'язки веб-серверу полягають у готовності прийняття повідомлення від користувачів у будь-який момент, а саме тому вона має бути завжди увімкнений та підключений до глобальної мережі інтернет. На рисунку 3.1 приведено схематичне зображення взаємодії веб-серверу та клієнту:

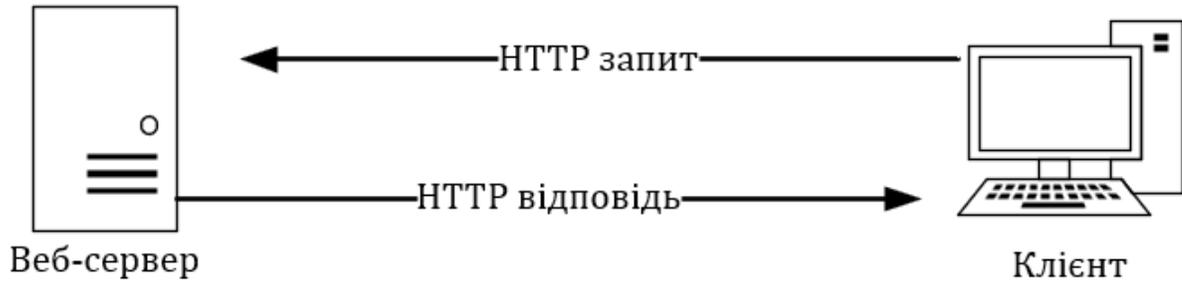


Рисунок 3.1 – Схематичне зображення взаємодії веб-серверу та клієнту

URL являється аббревіатурою від Uniform Resource Locator, що при перекладі з англійської мови означає єдиний вказівник ресурсів[]. Для розгляду структури URL-адреси для прикладу візьмемо URL <https://mon.gov.ua/>. Дана URL адреса містить у собі три складові частини [29]:

<схема>://<хост>/<шлях>

<схема> – схема звернення до серверу, в основному являє собою мережевий протокол, у даному випадку це протокол «HTTPS».

<хост> – ім'я хосту у системв доменних імен або це може бути IP-адреса серверу, у даному прикладі це «mon.gov.ua»

<шлях> – уточнює інформацію про місцезнаходження ресурсу, у даному прикладі це «/», в основному означає, що необхідно отримати головну сторінку. Розглянута найбільш поширена структура URL, але для більшого розуміння приведемо повну структуру [29].

<схема>:[//[<логін>[:<пароль>]@]<хост>[:<порт>]][/<шлях>][?<параметри>][#<якір>]

Запит, який був відправлений користувачем до віддаленого серверу зазвичай надходить до 80 порту. Даний порт являється типовим портом для протоколу HTTP. Розглядаючи структуру самого запиту, можемо побачити його чотири складові

частини: метод, ідентифікатор ресурсу, протокол та його версія, заголовки HTTP. Задачею веб-серверу є обробка отриманого запиту від клієнта. Приведемо схематичне зображення структури HTTP запиту на рисунку 3.2:

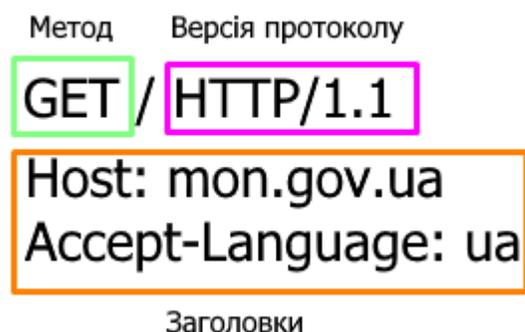


Рисунок 3.2 – Структура запиту HTTP

Для реалізації мало навантажених проєктів можна використовувати у якості веб-серверу домашній комп'ютер або ж навіть кластер комп'ютерів для високонавантажених проєктів. У обох випадках необхідно на даних комп'ютерах мати встановлене спеціалізоване програмне забезпечення, яке виконує функції роботи з мережею інтернет, а саме виконувати прийом та обробку запитів із мережі.

Раніше було з'ясовано, що для організації повноцінного веб-серверу наявна необхідність у встановленні спеціалізованого ПЗ на комп'ютер, що буде виконувати роль серверу. Через це для реалізації даної роботи необхідно визначитись із програмним забезпеченням, яке буде використано у даному проєкті. Розглянемо кілька найпоширеніших варіантів безкоштовного програмного забезпечення для веб-серверів.

Nginx – це HTTP-сервер, а також зворотній проксі-сервер, TCP/UDP проксі-сервер загального призначення, а також поштовий проксі-сервер [30]. З самого початку дане ПЗ розроблялося для популярного тоді ресурсу Rambler, адже він був високонавантажений та потребував більш продуктивного ПЗ веб-серверу. Станом на листопад 2022 року, Nginx використовують 21.21% високонавантажених ресурсів у

світі [31]. Із самого початку свого розвитку Nginx розроблювався лише для роботи із Unix подібними операційними системами, через це його було не можливо використовувати на серверах під ОС Windows. У процесі розвитку даного ПЗ воно стало доступним для використання на серверах, які використовують ОС Windows, але недоліком такого використання була нижча продуктивність у порівнянні із сервером, який працює на *nix. Найбільш оптимальним рішенням використовувати ПЗ Nginx є за наявності здебільше статичних елементів або ж файлів доступних для завантаження. Nginx використовує асинхронні сокети. Завдяки даній особливості відсутня необхідність у створенні нового процесу для обробки кожного запиту. За допомогою такої організації роботи з запитами Nginx у порівнянні із конкурентами являється більш продуктивним при роботі із статичним контентом. Завдяки тому, що один процес на ядро має змогу обробити декілька тисяч одночасних з'єднань досягається значне зниження навантаження на обчислювальну спроможність серверу. Також дане ПЗ можна використовувати не лише, як класичний HTTP-сервер, а і як інтерфейсний проксі сервер. При підтримці високонавантажених проєктів можемо часто побачити, як адміністратори використовують поєднання Nginx та Apache, що дозволяє отримати максимальну продуктивність роботи веб-серверу. На рисунку 3.3 приведемо схему взаємодії при парній роботі Nginx та Apache:

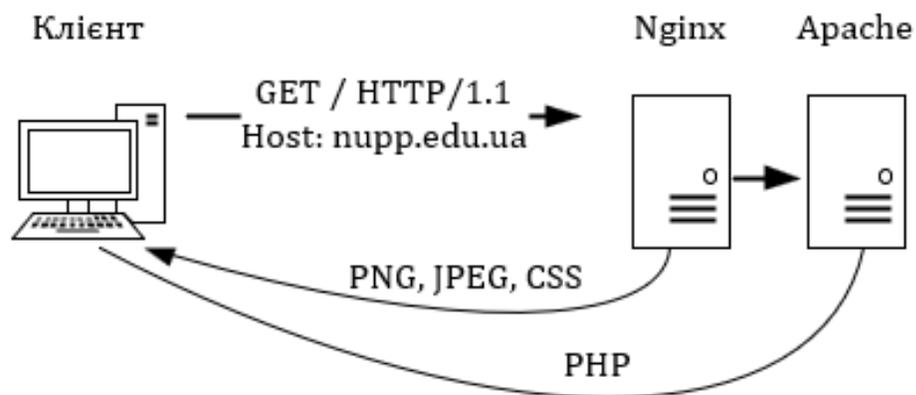


Рисунок 3.3 – Схематичне зображення парної роботи Apache та Nginx

HTTP-сервер Apache – веб-сервер, який є безкоштовним для використання та має відкритий початковий код [32]. На сьогоднішній день він являється найпопулярнішим ПЗ веб-серверу у світі [31]. Відкритість початкового коду Apache надає змогу будь-кому дослідити його. Така особливість дозволяє зробити припущення, що таке ПЗ може бути більш безпечним у порівнянні із продуктами, які мають закритий початковий код, адже у випадку з Apache, його код може дослідити будь-хто та знайти помилки. Також перевагою відкритого початкового коду є те, що при необхідності доступна можливість змінити код для своїх потреб. При використанні ПЗ Apache, у розробників наявна можливість використовувати сторонні модулі, що дає змогу розширити його базові функціональні можливості. Apache можна використовувати для будь-яких за розміром проєктів. Він підходить для запуску як і малих проєктів таких, як сайт-візитка, так і великих та високонавантажених сайтів. При порівнянні Apache та Nginx у продуктивності роботи із динамічним контентом, можемо побачити, що їх продуктивність при такому сценарії є приблизно на однаковому рівні. Якщо порівнювати роботу цих двох ПЗ при роботі із значною кількістю статичних файлів, то при такому сценарії, Apache має нижчу продуктивність. Архітектура Apache має дві складові частини – це ядро та модулі. При використанні ПЗ Apache, використання модулів не є обов’язковим, але у такому випадку багато функціональних можливостей буде досить обмежено. При необхідності додавання нових функціональних можливостей до Apache у розробника наявна можливість встановлення додаткових необхідних йому модулів, що стає можливим завдяки модульній структурній моделі ПЗ. Також звичайно присутня можливість вимкнення модулів у яких на даний момент немає потреби, що дозволяє збільшити продуктивність роботи веб-серверу. При використанні Apache у розробника наявна можливість використовувати у розробці свого проєкту не лише мову програмування PHP, а й такі, як Python, Ruby та інші. Така функціональна можливість досягається за допомогою встановлення модулів («mod_python», «apache_ruby»). При виконанні даної роботи будемо використовувати веб-сервер

Apache, та скриптову мову програмування PHP, для використання якої також необхідно додатковий модуль, який має назву «mod_php».

3.3 Скриптова мова програмування PHP

Мова програмування PHP, яка зараз є однією при розробці веб-застосунків створена у 1994 році. Із самого початку розробки даної мови, вона мала мету відслідковувати відвідувачів домашньої сторінки людини, яка розробляла мову програмування PHP. Спочатку PHP можна було розшифровувати, як Personal Home Page. Але при стрімкому розвитку даної мови її функціональність почала обростати новими можливостями. Завдяки розвитку функціональних можливостей PHP, розробники веб-сайтів почали створювати набагато складніші проекти. Після появи можливості створення складних проектів із використанням мови PHP її аббревіатуру розшифровують, як Hypertext Processor [33].

PHP надає можливість будь-кому редагувати початковий код під власні потреби. Така можливість досягається завдяки тому, що даний продукт має відкритий початковий код.

У процесі створення мови програмування PHP було запозичено велику кількість конструкції у інших мов програмування, таких як C та Perl, завдяки цьому PHP є досить схожою на інші мови програмування. У разі якщо розробник уже вивчав інші мови програмування на які схожа PHP, то для нього буде набагато легше вивчити мову програмування PHP завдяки своїй схожості на інші мови. Особливістю даної мови програмування є її простота. В незалежності від розміру коду сценарію, який написаний на мові PHP у розробника немає необхідності у використанні додаткових бібліотек [34].

Сценарії, які написані за допомогою мови програмування PHP у будь-якому випадку містяться та виконуються на веб-сервері, який у свою чергу має надіслати у відповідь до користувача сторінку HTML. В основному для виклику виконання

сценарію, користувачеві необхідно надіслати запит на віддалений сервер. Після того, як веб-сервер отримав запит від клієнту, він починає виконання PHP сценарію. Зазвичай після виконання сценарію PHP його результатом є згенерована HTML сторінка, яка містить у собі деякі динамічні дані. На рисунку 3.4 приведемо схему взаємодії клієнту та веб-серверу, який містить модуль PHP.

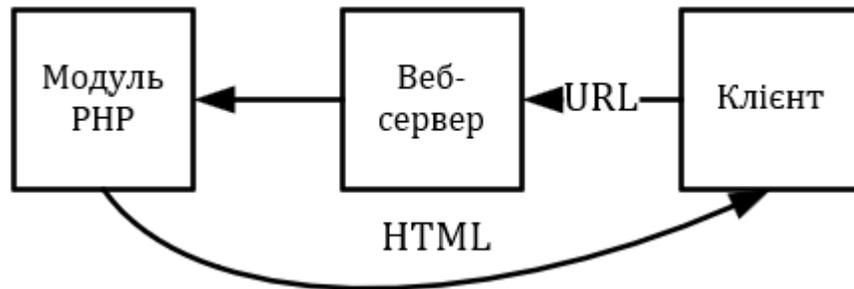


Рисунок 3.4 – Схема взаємодії клієнту, серверу та PHP

У разі, коли веб-ресурс містить лише HTML документи, то при отриманні сервером запити від клієнту у відповідь він надає статичну сторінку, яка була заздалегідь написаною. У такому випадку веб-сервер не виконує інтерпретацію даних. На рисунку 3.5 приведемо схему взаємодії клієнту та веб-серверу, який не містить модулю PHP.

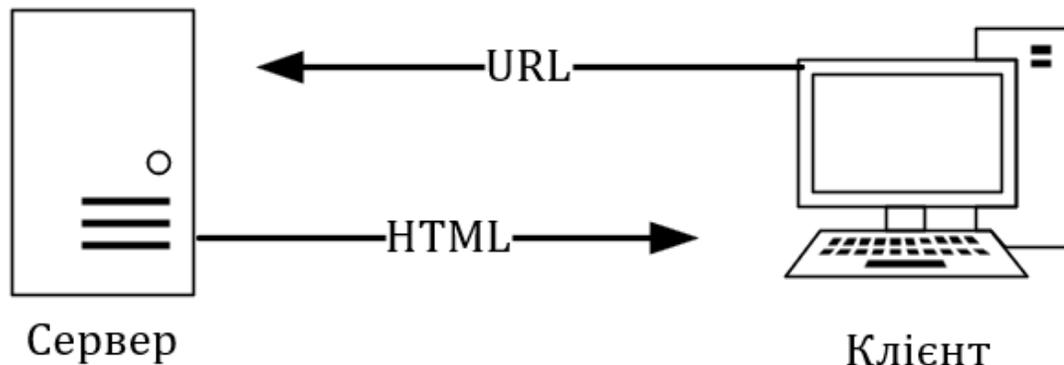


Рисунок 3.5 – Взаємодія клієнту з сервером без PHP

При розробці серверної частини веб-застосунку у розробника наявна можливість використання різних мов програмування окрім PHP, наприклад Python, Java та інші. Проте при розробці веб-застосунку перевагою використання мови програмування PHP є те, що вона із самого початку свого розвитку створювалась для створення веб-сторінок, які можуть містити динамічні елементи. А саме завдяки цьому передбачається більш продуктивне та швидке виконання таких задач у порівнянні із іншими мовами програмування, які використовують на серверній стороні веб-застосунку. Дивлячись на всі переваги мови програмування PHP, вона не завжди може забезпечити необхідні функціональні можливості для деяких специфічних проєктів.

При виборі необхідних засобів, які необхідні для розробки веб-застосунку важливими є наявні безпекові механізми. При використанні PHP у розробника наявна досить велика свобода дій та безпека, що досягається завдяки тому, що засобами безпеки керує адміністратор ресурсу [34]. Прикладом засобу безпеки може бути можливість налаштування обмеження на максимальний час, який дозволено виконуватись сценарію та обмежити використання пам'яті, що дозволяє уникнути негативного впливу на продуктивність веб-серверу. Також адміністратору ресурсу доступна можливість достатньо просто обмежити доступ до каталогів та файлів, які розташовані на веб-сервері, що дозволяє запобігти несанкціонованого доступу до конфіденційної інформації.

Перевагою використання мови програмування PHP у своїх проєктах є його незалежність роботи від ПЗ веб-серверу, яке використовується. PHP відмінно працює у поєднанні із більшістю найпопулярніших веб-серверів у світі, а саме Apache, Nginx, Microsoft IIS та багато інших. Враховуючи дану особливість, можна сказати, що мова програмування PHP є незалежною від програмної та апаратної складової серверу, а отже у розробника є змога використовувати цю мову програмування на різних серверах без необхідності у виконанні змін у кодї програми.

У створенні веб-застосунку при використанні мови програмування PHP, розробнику доступні такі переваги:

- **Продуктивність.** Мова програмування PHP створювалась насамперед для використання при розробці веб-сайтів, а отже має найбільшу ефективність роботи у такому сценарії.
- **Безпека.** За умови коректного налаштування PHP та реалізації програмного коду, від користувача буде приховано конфіденційний контент.
- **Універсальність.** Завдяки можливості роботи PHP з різноманітним ПЗ веб-серверів, наявна можливість роботи програмного коду на різних ОС, без додаткової його модифікації.
- **Простота.** PHP із самого початку призначався для використання для побудови веб-сайтів, тому наявна можливість помістити код PHP до структури HTML сторінки. Також при вивченні синтаксису даної мови, можемо побачити, що він є досить легким для розуміння людиною.
- **Підтримка.** На сьогоднішній день мова програмування PHP є однією із найпопулярнішою при створенні веб-застосунків, завдяки чому у мережі інтернет доступна велика кількість необхідної інформації для вирішення проблем.

Отже, для виконання даної роботи буде використано скриптову мову програмування PHP.

3.4 HTML – мова розмітки гіпертексту

Оскільки для повноцінної роботи веб-застосунку необхідний інтерфейс взаємодії для користувача, то для виконання даної задачі буде використано мову для розмітки гіпертексту, яка має назву HTML.

HTML – мова гіпертекстової розмітки. Дана мова є стандартизованою мовою для виконання розмітки документів [35].

Посилання, які виконують функції зв'язування веб-сторінок між собою називають гіпертекстом. Такі посилання дають можливість зв'язувати документи, які можуть бути в межах одного сайту та і посилати на інші сайти.

Для коректного відображення у браузері користувача таких елементів, як текст та зображення зазвичай необхідно використовувати мову розмітки гіпертексту HTML. Розглядаючи структуру HTML документу можна сказати, що він являє собою ASCII-файл, який додатково містить керуючі теги HTML. Для того, щоб браузер мав змогу відрізнити звичайний текст від елементу HTML, необхідно використовувати спеціальні теги. Тегом являється назва елементу, який необхідно оточити кутовими дужками «<» та «>». Використання тегів HTML у більшості випадків є інтуїтивно зрозумілими для людини, завдяки тому, що їх назви є загальноживаними англійськими словами.

Відмінністю мови розмітки гіпертексту від звичайного тексту є наявна можливість при створенні вмісту сторінки використовувати спеціальні конструкції, які містить мова розмітки гіпертексту HTML. Використовуючи такі конструкції, розробник може задати гіперпосилання до іншого документу, що дозволяє користувачеві перейти до нього.

Важливо зазначити, що мова розмітки гіпертексту HTML не являється мовою програмування. Використання даною мови передбачає лише розмітку гіпертексту. При читанні розмітки гіпертексту браузером, він виконує розпізнання параметрів розмітки, які розробник присвоїв певним елементам.

Виконання розмітки HTML доступне навіть без наявності додаткового спеціального ПЗ, виконати дану задачу можливо навіть у звичайному текстовому редакторі по типу Блокнота у ОС Windows. Але використання додаткового ПЗ для розробки може додати зручності та швидкості написання коду, а отже кожен може обрати оптимальний для себе варіант. Станом на сьогоднішній день у мережі інтернет

можна знайти велику кількість зручних редакторів коду, які доступні для вільного використання та більшість із них мають у наявності підтримку мови розмітки гіпертексту. Такі редактори бувають різних видів, а саме можуть бути у вигляді візуального редактору, що дозволяє наочно додавати та розміщувати елементи або ж спеціальні текстову редактори, які за своїми алгоритмами можуть вказати на помилки у синтаксисі та для зручності сприйняття оформити код у різних кольорах код. Також для пришвидшення розробки у редакторах, які мають підтримку мови HTML, наявна можливість автодоповнення, що дозволяє у значній мірі підвищити швидкість написання коду.

3.5 CSS – каскадні таблиці стилів

У сучасному світі при створенні користувацьких інтерфейсів неможливо використовувати лише мову розмітки гіпертексту. Необхідно ще додати стилістичне оформлення для відображення HTML елементів. Для виконання таких задач розробники використовують каскадні таблиці стилів CSS (Cascading Style Sheets), що являє собою формальну мову, яка надає можливість опису зовнішнього вигляду документу, який був написаний із використанням HTML [34].

Задачею у використанні каскадних стилів таблиць (CSS) є надання можливості для виконання розділення структури розмітки сторінки та опису того, як вона повинна виглядати на пристрої користувача. CSS в основному використовують для того, щоб задати позицію блоків на сторінці, параметри шрифту та кольори. Розділення елементів розмітки гіпертексту від опису стилів документу надає простіше сприйняття коду для людини та запобігання повторюваності вмісту.

Для того, щоб надати можливість браузеру на клієнтському пристрої поєднання стилів та розмітки, розробнику у файлі HTML необхідно вказати шлях до файлу необхідних таблиць стилів. Взагалі при розробці веб-сторінки існує кілька способів задати необхідні стилі, а саме задання стилів безпосередньо у документі HTML,

винести код стилів у окремий документ та підключити цей документ до HTML сторінки або ж наявна можливість об'єднання цих двох способів.

Для прикладу продемонструємо найбільш поширений варіант поєднання таблиці стилів із документом HTML:

- Каскадна таблиця стилів міститься у окремому файлі та підключається до HTML документу за допомогою HTML елемента `<link>` [34]:

```
<!DOCTYPE html>
<html>
  <head>
    .....
    <link rel="stylesheet" type="text/css" href="style.css">
  </head>
  <body>
    .....
  </body>
</html>
```

Елементи CSS описані безпосередньо у файлі розмітки гіпертексту використовуючи тег `<style>`:

```
<!DOCTYPE html>
<html>
  <head>
    .....
    <style>
      body {
        color: blue;
      }
    </style>
  </head>
  <body>
    .....
  </body>
</html>
```

Використання CSS дозволило заощадити зусилля, які необхідні для виконання верстки веб-сторінок, адже каскадні таблиці стилів надають можливість гнучко виконувати налаштування стилів елементів мови розмітки гіпертексту, які знаходяться у документі. Якщо у парі із HTML розміткою використовувати стилі CSS, розробник зможе досягти більш зручної можливості розробки зовнішнього вигляду веб-сторінки та покращити точність її відображення на різних девайсах. Одна із

головних характеристик зручності використання CSS є наявна можливість відокремлення коду, який відповідає за оформлення сторінки від змісту самої сторінки.

- Перевагами використання каскадних стилів таблиць CSS є:
- Проста підтримка коду
- Швидкість підключення
- Наявність можливості у використанні одного файлу, який містить таблиці стилів для безлічі різних HTML документів.

Головною рисою сучасних веб-сторінок є їх досить привабливий вигляд для користувача. Для збільшення швидкості верстки веб-сторінок почали розробляти CSS фреймворки. Фреймворком являється своєрідний каркас, який дозволяє збільшити швидкість розробки, адже основні елементи у ньому уже реалізовані, а отже для розробника відсутня необхідність у розробці якихось базових елементів. Одним із таких CSS фреймворків є Bootstrap.

Bootstrap являє собою безкоштовний продукт, який має відкритий початковий код, а саме це є набором базових інструментів, які необхідні при створенні сучасних веб-сторінок [36].

Головною особливістю даного фреймворку наявність сітки, яка складається із дванадцяти колонок. Така особливість надає розробнику можливість набагато простіше та швидше виконувати верстку веб-сторінок, які можуть адаптуватися в залежності від пристрою користувача. Для розуміння організації функцій сітки Bootstrap можемо представити її у вигляді звичайної таблиці, яка представлена у таблиці 3.1:

Таблиця 3.1 – Сітка Bootstrap

.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1
.col-md-8								.col-md-4			
.col-md-4				.col-md-4				.col-md-4			
.col-md-6						.col-md-6					
.col-md-12											

Отже, у процесі створення інтерфейсу користувача буде використано каскадні таблиці стилів CSS, адже вони дозволяють додати привабливого вигляду веб-сторінці та оформити її елементи так, що дозволить користувачеві зручно взаємодіяти із елементами сторінки. Для того, щоб спростити та пришвидшити верстку, будемо використовувати CSS фреймворк Bootstrap. Використання даного фреймворку дозволить створити максимально швидко достатньо привабливий та зручний інтерфейс для користувача, який також матиме змогу адаптуватись в залежності від пристрою користувача.

3.6 Скриптова мова програмування JavaScript та технологія AJAX

У процесі розвитку веб-технологій у розробників виникла проблема, адже статичних веб-сторінок стало вже не достатньо та необхідно було додати до них динамічності. За часів статичних веб-сайтів веб-технології почали потребувати створення простої скриптової мови, що дозволяє виконувати роботу із елементами DOM. Внаслідок чого була створена скриптова мова програмування, яка має назву JavaScript, яка вирішила дану проблему.

JavaScript – це інтерпретована мова програмування із об'єктно-орієнтованими можливостями [37]. Дана мова є схожою на мови програмування C, C++, Java через використання у ній подібного синтаксису та конструкцій. Завдяки тому, що JavaScript

являє собою нетипізовану мову, тому для розробника немає необхідності задавати типи змінних.

В основному JavaScript застосовується на пристроях користувачів за допомогою веб-браузеру. Дана особливість надає можливість реалізувати динамічність у взаємодії людини із веб-сторінкою. При використанні JavaScript з'являється можливість динамічної зміни змісту сторінки без наявної необхідності у перезавантаженні веб-сторінки.

Далі розглянемо особливості використання JavaScript на стороні клієнту, адже у даній роботі ця мова буде використана лише при створенні інтерфейсу користувача.

При використанні скриптової мови програмування JavaScript на стороні клієнту, основною задачею його являється керування поведінкою контенту, який містить веб-сторінка. Дана можливість досягається через те, що JavaScript може виконувати певні дії із об'єктною моделлю документу (DOM). На основі JavaScript також наявні деякі технології, які доступні при розробці веб-застосунку, наприклад AJAX.

Для того, щоб влаштувати JavaScript код у структуру HTML документу, використовують спеціальні HTML теги, а саме код розміщується між тегами `<script>` та `</script>`.

```
<script>  
// JavaScript сценарій  
</script>
```

Також, як і у випадку із CSS наявна можливість винесення коду JavaScript в окремий файл. Для реалізації такого варіанту приєднання необхідно при використанні тегу `<script>` використати спеціальний атрибут «src», значенням якого повинна буди URL-адреса, яка вказує на файл JavaScript коду.

```
<script src="./js/script.js"></script>
```

У кодї розмітки HTML сторінки наявна можливість розміщення необмеженої кількості об'єктів `<script>`. У випадку, коли у кодї розмітки наявні декілька сценаріїв,

то їх виконання відбувається послідовно, по мірі їх розміщення у коді веб-документу. Незважаючи на той факт, що код окремих сценаріїв виконується по мірі обробки клієнтським браузером коду розмітки гіпертексту HTML, ці окремі елементи являють собою частини однієї програми. Завдяки наявності такої особливості задані у різних файлах змінні та функції стають доступними для взаємодії між всіма сценаріями, які підключені до сторінки.

Розглядаючи об'єктну модель документу, можемо побачити, що основним об'єктом являється об'єкт Window, через який доступні для взаємодії усі інші наявні об'єкти. Наприклад, будь-який об'єкт Windows містить у собі document, що виконує посилання на об'єкт Document, який є пов'язаний із вікном та містить доступну властивість location. Windows є глобальним об'єктом, який міститься у початку ланцюга видимої та усі інші об'єкти можуть бути доступні, як властивості інших доступних користувачеві об'єктів. А саме при використанні мови JavaScript наявна своєрідна ієрархія об'єктів, яка починається із об'єкту Window. На рисунку 3.6 представимо ілюстрацію такої ієрархії:

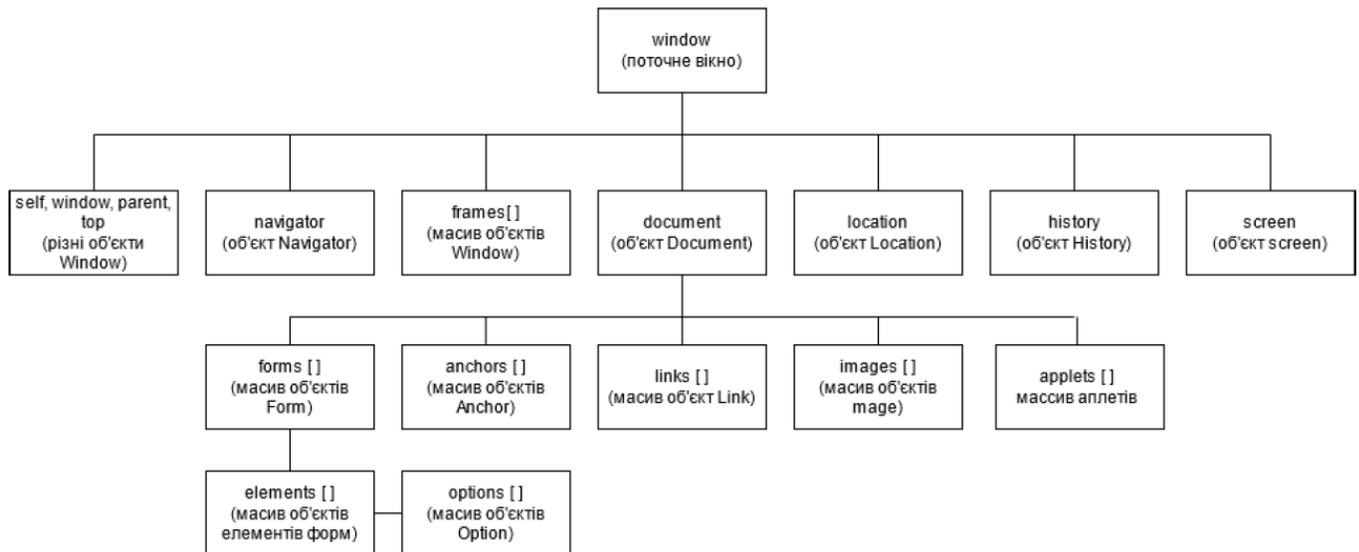


Рисунок 3.6 – Ієрархія об'єктів JavaScript

Завдяки стрімкому розвитку функціональних можливостей та збільшення популярності мови JavaScript на основі її почали створюватись технології, що базуються на використанні JavaScript. Прикладом технології основаної на мові JavaScript, може бути технологія AJAX.

Абревіатура AJAX розшифровується, як Asynchronous JavaScript and XML (англ. асинхронний JavaScript та XML) [38]. Розглядаючи, як влаштована дана технологія, можемо сказати, що вона є звичайним сценарієм JavaScript задачею якого наявній необхідності взаємодіяти із віддаленим сервером у фоновому режимі, а саме надсилати та запити та отримувати відповідь від серверу. Таку функціональну можливість зазвичай використовують для виконання окремих елементів наявних у структурі веб-сторінки, завдяки чому зникає необхідність у повному перезавантаженні сторінки, при необхідності оновлення елементів на сторінці.

Сучасне ПЗ для взаємодії із веб-контентом, а саме веб-браузери повністю містять у собі необхідний набір технологій, який необхідний у роботі технології AJAX. Це означає, що при використанні застосунків, які використовують AJAX, відсутня необхідність у інсталяції додаткового ПЗ чи компонентів.

Насамперед для того, щоб виконати запит до серверу із використанням технології AJAX, необхідно створити екземпляр об'єкту, який має назву XMLHttpRequest та містить в собі певний набір властивостей. Для розуміння наявних властивостей даного об'єкту у таблиці 3.2 приведемо опис основних властивостей XMLHttpRequest:

Таблиця 3.2 – Основні властивості об'єкту XMLHttpRequest

Метод	Опис
open("method", "URL"[, async-Flag[, "userName"[, "password"]]])	ініціалізація параметрів запиту send
send(content)	виконання HTTP запиту
status	повернути код статусу запиту
statusText	повернути інформацію про стан запиту у повідомленні
responseXML	повернути відповідь від серверу у форматі XML документу
responseText	повернути відповідь серверу у форматі текстового рядку
onreadystatechange	дає можливість встановити установки функції зворотнього виклику, яка буде обслуговувати стан запиту
readyState	повертає стан запиту: 0 = не ініціалізовано 1 = триває відправлення запиту 2 = запит успішно відправлений 3 = триває обмін 4 = завершений
setRequestHeader("label","value")	призначає пари label/value у заголовку запиту
getResponseHeader("headerLabel")	повернення заголовку однієї відповіді у вигляді рядка
abort()	виконує зупинку виконання поточного запиту

У процесі розгляду наявних у сьогоденні веб-технологій можемо зробити висновок, що сучасні веб-застосунки мають мати таку властивість, як динамічність. А тому, для створення веб-застосунку, який буде відповідати сучасним вимогам, у даному проєкті будемо використовувати технології JavaScript та AJAX. Необхідність використання даних технологій полягає у необхідності динамічного оновлення відображуваних даних на сторінці користувацького інтерфейсу.

3.7 Висновки

У даному розділі було розглянуто та обрано програмні засоби для виконання даної роботи, а саме модернізації web-застосунку для візуалізації даних системи «Розумний будинок».

При виконанні роботи по реалізації серверної частини даного веб-застосунку було зроблено вибір на користь використання безкоштовного для використання веб-серверу Apache.

Перевагою використання даного ПЗ веб-серверу є наявність можливості його роботи під керуванням різних ОС, а отже робота даного веб-застосунку буде незалежною від того, яку ОС встановлено на веб-сервері, що дозволяє за необхідності легко переносити веб-застосунок на сервер із практично будь-якою підтримуваною ОС.

Серверною мовою програмування буде слугувати скриптова мова програмування PHP. Дана мова програмування була обрана завдяки її простоті у інтеграції до документів розмітки гіпертексту HTML, яка є практично незамінною при створенні інтерфейсу користувача веб-застосунку.

Для надання привабливого вигляду та адаптивності веб-сторінкам, буде використано CSS фреймворк під назвою Bootstrap, а також для надання динамічності веб-сторінкам буде використано технології JavaScript та AJAX.

4. ПРОЦЕС ВИКОНАННЯ ЗАДАЧІ ПО МОДЕРНІЗАЦІЇ ВЕБ-ЗАСТОСУНКУ ДЛЯ ВІЗУАЛІЗАЦІЇ ДАНИХ СИСТЕМИ «РОЗУМНИЙ ДІМ»

4.1 Загальні функціональні можливості веб-застосунку

Для початку виконання поставленої задачі, наявна необхідність у визначенні із необхідними для реалізації можливостями даного веб-застосунку.

Основною задачею веб-застосунку для візуалізації даних системи «розумний будинок» є зручне візуальне представлення даних, що отримано від різноманітних сенсорів, які наявні у системі «розумний будинок».

При побудові системи «розумний будинок», важливим аспектом є можливість об'єднання пристроїв, які використовують різні протоколи передачі даних. При реалізації даного проєкту буде вирішено проблему відображення даних із сенсорів системи «розумний будинок», що використовують різні протоколи для передачі даних.

Особливістю даного веб-застосунку буде наявна можливість обміну даними із системою «розумний будинок» за допомогою використання комбінації декількох протоколів передачі даних, а саме MQTT (Message Queuing Telemetry Transport), WebSocket, SSE (Server Side Events) та HTTP-запитів.

У результаті буде розроблено веб-застосунок, який дозволяє:

- групувати пристрої по кімнатах
- візуально представляти отримані із сенсорів дані
- задання певних дій при отриманні порогових значень із сенсорів.

4.2 Налаштування програмного середовища веб-серверу

Перед початком створення веб-застосунку для візуалізації даних системи «розумний дім» є необхідність у запуску веб-серверу. Це необхідно для того, щоб у процесі розробки була можливість перевіряти функціонал PHP сценаріїв.

Для реалізації даного проєкту було обрано використання веб-серверу Apache, але для надання повноцінних функціональних можливостей наявна необхідність у інсталяції додаткових модулів. А саме необхідно інстальювати:

- ПЗ веб-серверу Apache та модуль для нього під назвою PHP
- СУБД MySQL та для зручності адміністрування її, графічний інтерфейс phpMyAdmin.

Обрана ОС для інсталяції на сервер є Linux дистрибутив Ubuntu 22.04.1 LTS, який на даний момент є останньою доступною версією даного дистрибутиву. А отже при виконанні конфігурації необхідно брати до уваги, що деякі налаштування, які були актуальні для попередніх версій дистрибутиву, можуть бути недоступними для даної версії. У процесі керування веб-сервером буде використано протокол, що надає можливість керувати віддаленим сервером та має можливість тунелювати TCP з'єднання, а саме це протокол SSH (Secure Shell).

Задля забезпечення базової безпеки насамперед необхідно брандмауер, який дозволяє забезпечити доступ до можливості підключення лише для базових служб. У якості програмного забезпечення брандмауеру інстальюємо UFW. У процесі інсталяції прикладного ПЗ для нього буде надана можливість у реєстрації профілів у UFW. Насамперед для того, щоб не втратити можливість віддаленого керування веб-сервером, необхідно надати дозвіл на прийом з'єднань SSH.

Для того, щоб додати програмне забезпечення OpenSSH у список дозволених до вхідних з'єднань необхідно у терміналі виконати таку команду:

```
ufw allow OpenSSH
```

Результатом виконання даної команди є надання доступу для хідних з'єднань відповідному ПЗ. Для впевненості, що команда спрацювала можемо перевірити це за допомогою такої команди:

```
ufw status
```

Результатом виконання команди є відображення таблиці, яка містить у собі активні правила брандмауеру:

```
clouduser@dipl-ubuntu:~$ sudo ufw status
Status: active

To Action From
-- ---
OpenSSH ALLOW Anywhere
OpenSSH (v6) ALLOW Anywhere (v6)
```

Рисунок 4.1 – Активні правила брандмауеру

Після налаштування брандмауеру, виконаємо інсталяцію веб-серверу Apache. Інсталювати Apache можна використовуючи типовий репозиторій Ubuntu за допомогою використання інструменту для управління пакетами, який наявний у даній ОС. Насамперед необхідно оновити список пакетів, які доступні для інсталяції із репозиторіїв. Виконати дану дію можна за допомогою такої команди:

```
sudo apt update
```

Виконане заздалегідь оновлення списку пакетів, надає можливість гарантовано встановити останню наявну версію необхідного ПЗ.

Для подальшої роботи, необхідно виконати інсталяцію пакету apache2, використовуючи таку команду:

```
sudo apt install apache2
```

Для того, щоб надати дозвіл на підключення до необхідних Apache веб-портів, необхідно виконати редагування конфігурації брандмауеру. У процесі інсталяції Apache, відбувається реєстрація профілів UFW, які в подальшому використовують для

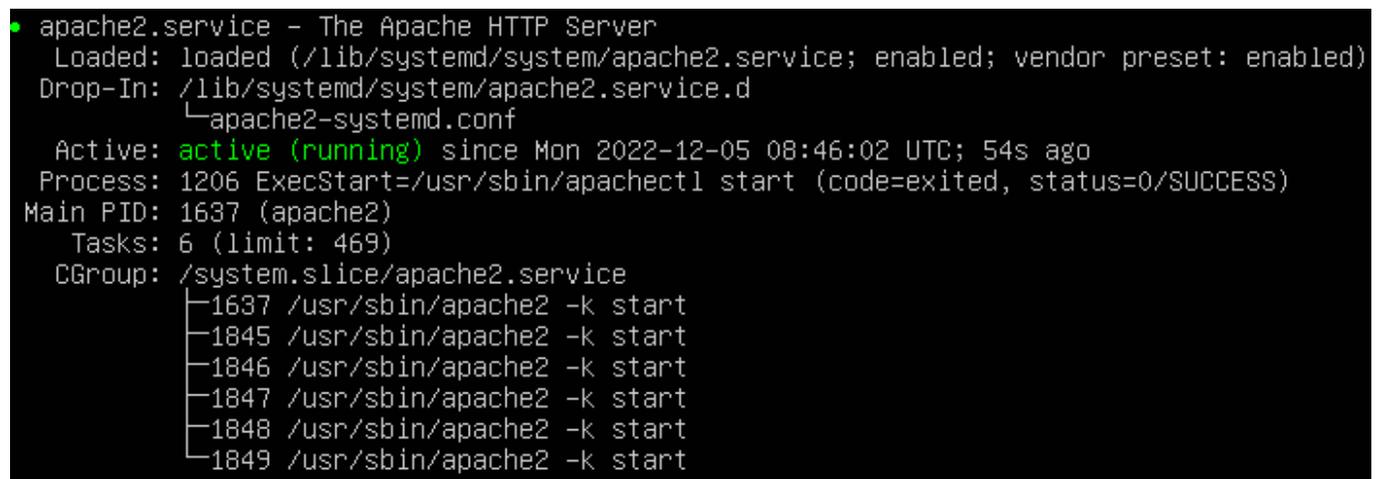
управління обмеженнями доступу до мережі. Для дозволу зовнішнього доступу до порту 80, що є стандартним портом HTTP, необхідно виконати таку команду:

```
sudo apt install apache2
```

Після того, як було налаштовано правила брандмауєру для Apache, виконаємо перевірку статусу служби Apache, що дозволяє впевнитись у коректності його роботи. Виконаємо команду:

```
sudo systemctl status apache2
```

На рисунку 4.2 приведено результат виконання команди описаної вище:

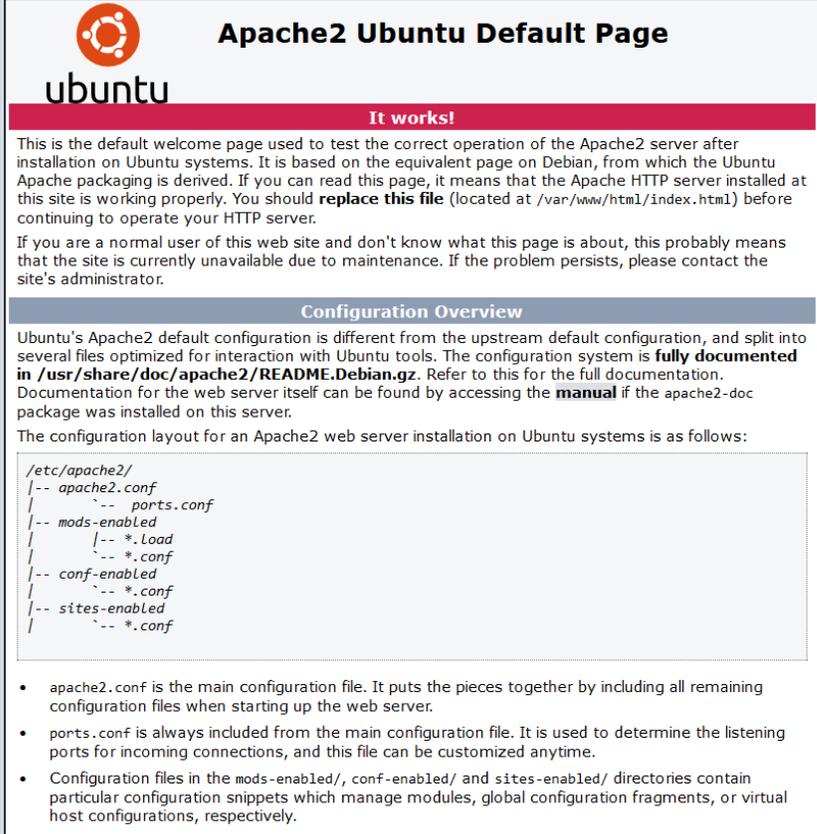


```
• apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Drop-In: /lib/systemd/system/apache2.service.d
           └─apache2-systemd.conf
   Active: active (running) since Mon 2022-12-05 08:46:02 UTC; 54s ago
   Process: 1206 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 1637 (apache2)
   Tasks: 6 (limit: 469)
   CGroup: /system.slice/apache2.service
           └─1637 /usr/sbin/apache2 -k start
             └─1845 /usr/sbin/apache2 -k start
               └─1846 /usr/sbin/apache2 -k start
                 └─1847 /usr/sbin/apache2 -k start
                   └─1848 /usr/sbin/apache2 -k start
                     └─1849 /usr/sbin/apache2 -k start
```

Рисунок 4.2 – Результат виконання перевірки статусу служби apache2

Після виконання перевірки статусу служби apache2 можемо впевнитись, що вона активна та виконується без помилок. Але задля більшої впевненості у коректності конфігурації перевіримо роботу веб-серверу за допомогою безпосереднього доступу до стандартної веб-сторінки, яка міститься на веб-сервері, а саме виконаємо запит до URL-адреси у локальній мережі <http://192.168.174.128>.

На рисунку 4.3 представлено скріншот відповіді від серверу, яка являє собою тестову веб-сторінку серверу Apache.



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```

/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf

```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.

Рисунок 4.3 – Стандартна тестова сторінка Apache

Результатом запиту до веб-серверу за URL-адресом є HTML документ, який являє собою веб-сторінку, яка у свою чергу генерується веб-сервером Apache у процесі його інсталяції. А отже можемо зробити висновок, що базова інсталяція веб-серверу Apache виконана успішно.

Керування процесом веб-серверу Apache виконується за допомогою використання терміналу. У таблиці 4.1 представлено основні команди, які необхідні для управління веб-сервером.

Таблиця 4.1 – Команди для управління службою Apache

Команда	Опис
<code>sudo systemctl stop apache2</code>	Зупинка веб-серверу
<code>sudo systemctl start apache2</code>	Запуск зупиненого веб-серверу
<code>sudo systemctl restart apache2</code>	Перезавантаження служби Apache
<code>sudo systemctl reload apache2</code>	Перезавантаження служби Apache без відключення з'єднань
<code>sudo systemctl disable apache2</code>	Деактивація автоматичного запуску служби при завантаженні серверу
<code>sudo systemctl enable apache2</code>	Активація автоматичного запуску служби при завантаженні серверу

Завдяки тому, що для виконання даної роботи було обрано скриптову мову програмування PHP, то наявна необхідність у інсталяції додаткового модуля Apache, який у свою чергу додає можливість роботи PHP на даному веб-сервері. Також буде необхідно використання БД MySQL, тому виконаємо команду для інсталяції необхідних додаткових модулів:

```
sudo apt install php libapache2-mod-php php-mysql
```

Після виконання інсталяції модулю PHP перевіримо його роботу. Для виконання цієї задачі необхідно у публічній директорії створити файл, який являє собою скрипт PHP, та буде містити у собі функцію `phpinfo()`, що дозволить вивести у відповідь на запит інформацію про PHP. У нашому випадку назва тестового файлу є `phpinfo.php`, отже виконаємо запит до серверу за допомогою URL-адреси `http://192.168.174.128/phpinfo.php`. Отримавши запит за даним URL, веб-сервер у свою чергу виконає PHP скрипт та згенерує у відповідь сторінку, яка містить інформацію

про модуль PHP, який інстальовано до Apache. На рисунку 4.4 представлено результат виконання скрипту `phpinfo.php`.



PHP Version 8.0.0	
System	Linux focal64 5.4.0-42-generic #46-Ubuntu SMP Fri Jul 10 00:24:02 UTC 2020 x86_64
Build Date	Nov 27 2020 12:26:22
Build System	Linux
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.0/fpm
Loaded Configuration File	/etc/php/8.0/fpm/php.ini
Scan this dir for additional .ini files	/etc/php/8.0/fpm/conf.d
Additional .ini files parsed	/etc/php/8.0/fpm/conf.d/10-opcache.ini, /etc/php/8.0/fpm/conf.d/10-pdo.ini, /etc/php/8.0/fpm/conf.d/20-calendar.ini, /etc/php/8.0/fpm/conf.d/20-ctype.ini, /etc/php/8.0/fpm/conf.d/20-exif.ini, /etc/php/8.0/fpm/conf.d/20-ffi.ini, /etc/php/8.0/fpm/conf.d/20-fileinfo.ini, /etc/php/8.0/fpm/conf.d/20-ftp.ini, /etc/php/8.0/fpm/conf.d/20-gettext.ini, /etc/php/8.0/fpm/conf.d/20-iconv.ini, /etc/php/8.0/fpm/conf.d/20-phar.ini, /etc/php/8.0/fpm/conf.d/20-posix.ini, /etc/php/8.0/fpm/conf.d/20-readline.ini, /etc/php/8.0/fpm/conf.d/20-shmop.ini, /etc/php/8.0/fpm/conf.d/20-sockets.ini, /etc/php/8.0/fpm/conf.d/20-sysvmsg.ini, /etc/php/8.0/fpm/conf.d/20-sysvsem.ini, /etc/php/8.0/fpm/conf.d/20-sysvshm.ini, /etc/php/8.0/fpm/conf.d/20-tokenizer.ini
PHP API	20200930
PHP Extension	20200930
Zend Extension	420200930
Zend Extension Build	API420200930,NTS
PHP Extension Build	API20200930,NTS
Debug Build	no

Рисунок 4.4 – Результат виконання `phpinfo.php`

Розглядаючи результат виконання нашого тестового скрипту, можемо зробити висновок, що модуль PHP інстальовано правильно та він коректно працює.

4.3 Конфігурація системи управління базами даних MySQL

При виконанні практичної частини даної роботи наявна необхідність у зручному зберіганні даних, які надходять від різних пристроїв. Така можливість необхідна для подальшої їх обробки та відображення у користувацькому інтерфейсі. Оптимальним вибором для вирішення даної задачі є система управління базами даних, яка має назву MySQL, адже вона є достатньо продуктивною у використанні для малих та середніх за розміром веб-проектів.

Насамперед для початку роботи наявна необхідність у інсталяції СУБД на сервер. Для реалізації даної роботи був зроблений вибір на користь MySQL, отже виконаємо команду для інсталяції:

```
sudo apt install mysql-server
```

В результаті виконання даної команди отримаємо інстальований пакет MySQL останньої версії. MySQL містить у собі скрипт безпеки, який необхідний для того, щоб виправити типові опції, адже не налаштована з точки зору безпеки СУБД є уразливою. Наприклад за замовчуванням після інсталяції даної СУБД у системі наявні тестові користувачі та інші ненадійні налаштування. Наступним кроком виконаємо команду:

```
sudo mysql_secure_installation
```

Результатом виконання даної команди є серія діалогових вікон, які дозволяють виправити типові уразливості та підвищити безпеку від несанкціонованого доступу. Здійснення налаштувань, які запропоновані даним скриптом є інтуїтивно зрозумілими, не будемо їх детально розглядати.

Виконаємо перевірку роботи щойно інстальованої СУБД MySQL. Після виконання встановлення MySQL, вона має автоматично запустись та працювати. За допомогою наступної команди перевіримо її роботу:

```
systemctl status mysql.service
```

На рисунку 4.5 приведено результат виконання команди:

```
mysql.service - MySQL Community Server
Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
Active: active (running) since Mon 2022-12-05 14:55:41 UTC; 48s ago
Process: 1469 ExecStart=/usr/sbin/mysqld --daemonize --pid-file=/run/mysqld/mysqld.pid (code=exite
Process: 1146 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, status=0/SUCCESS
Main PID: 1471 (mysqld)
Tasks: 27 (limit: 469)
CGroup: /system.slice/mysql.service
└─1471 /usr/sbin/mysqld --daemonize --pid-file=/run/mysqld/mysqld.pid
```

Рисунок 4.5 – Перевірка статусу MySQL

Отримавши результат виконання команди перевірки статусу MySQL, можемо побачити, що дана СУБД активна, а отже була успішно інстальована та відсутні критичні помилки конфігурації.

Однією із переваг використання MySQL є наявна можливість для адміністрування використовувати зручного веб-інтерфейсу, який має назву

phpMyAdmin. Тому для того, щоб полегшити адміністрування БД, виконаємо інсталяцію веб-інтерфейсу phpMyAdmin. Для інсталяції даного ПЗ та необхідних модулів для його роботи, виконаємо наступну команду:

```
sudo apt-get install phpmyadmin php-mbstring php-gettext
```

При процесі інсталяції даного ПЗ, система запропонує задати базові конфігурації, а саме буде необхідність у виборі наявного веб-серверу (Apache) та задання необхідних паролів.

Для можливості коректної роботи phpMyAdmin необхідно активувати два модулі PHP, а саме mcrypt та mbstring за допомогою команди:

```
sudo phpenmod mcrypt
```

```
sudo phpenmod mbstring
```

Після виконання дій по інсталяції даного ПЗ, необхідно перезавантажити веб-сервер Apache за допомогою наступної команди:

```
sudo systemctl restart apache2
```

Перевіримо роботу phpMyAdmin за допомогою доступу по URL-адресі <http://192.168.174.128/phpmyadmin>. Виконавши запит за даним URL, необхідно виконати авторизацію, після чого можемо побачити доступний користувацький інтерфейс phpMyAdmin, який представлено на рисунку 4.6:

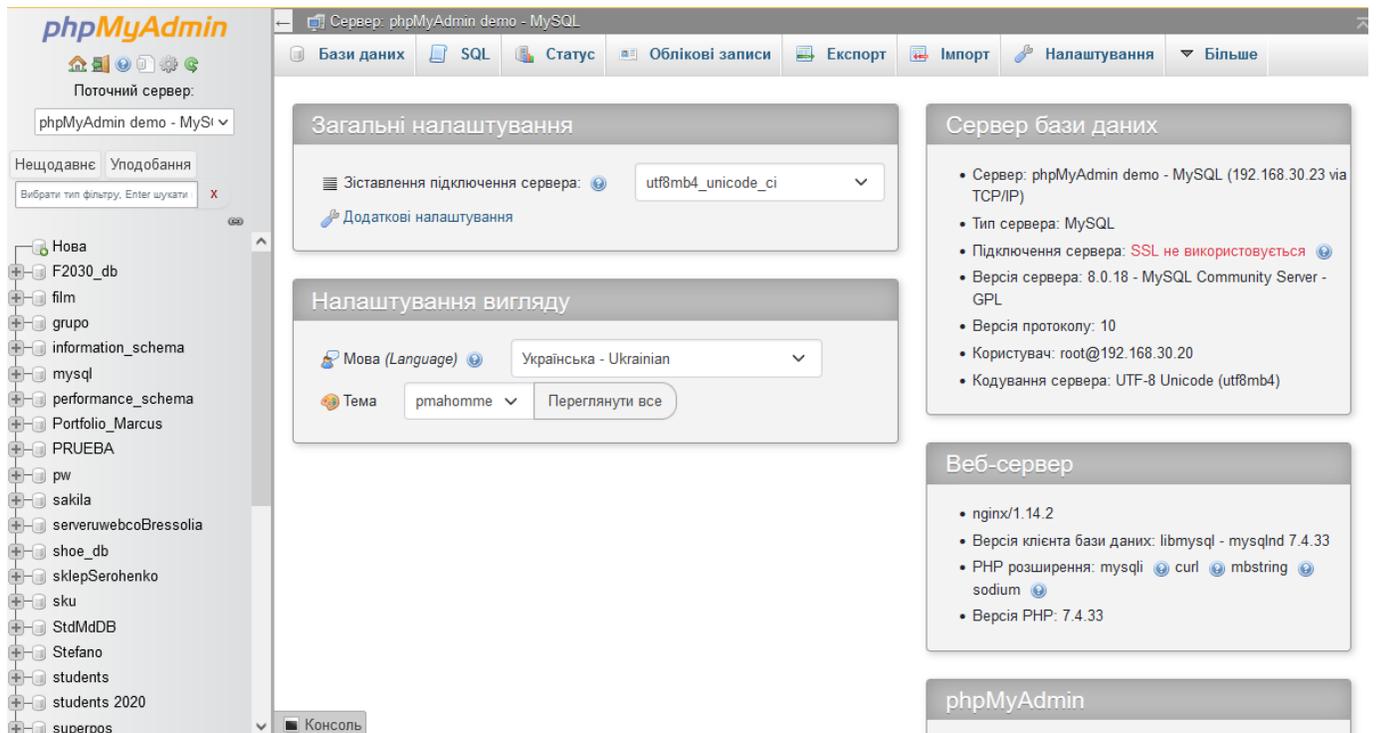


Рисунок 4.6 – Інтерфейс phpMyAdmin

У результаті інсталяція даного ПЗ надає можливості легкого та зручного керування базами даних, таблицями та користувачами. Також за допомогою використання phpMyAdmin наявна можливість у зручному та наочному додаванні та реагуванні даних, які уже містять у базі даних.

Використовуючи користувацький веб-інтерфейс керування базами даних, створимо структуру БД, яка буде необхідна при роботі нашого веб-застосунку.

Для початку розробки структури таблиць, необхідно створити базу даних. Використовуючи веб-інтерфейс phpMyAdmin створимо нову базу даних та присвоїмо їй назву «smarthouse». Після успішного створення бази даних буде надалі запропоновано системою створити нову таблицю, яка буде міститись у даній БД. Для можливості простої та зручної роботи із даними отриманими із системи «розумний будинок» наявна необхідність у створенні каркасу таблиці, яка буде містити у собі отриману інформацію із системи.

Для нашого проєкту необхідно три таблиці, а саме таблицю яка буде містити у собі список наявних пристроїв у системі, таблицю, яка буде містити історію даних, які були отримані із сенсорів та пристроїв та таблицю із налаштованими діями у відповідь на задані порогові дані із сенсорів.

Представимо структуру таблиць баз даних у вигляді звичайної таблиці 4.2.

Таблиця 4.2 – Структура таблиць баз даних

Наявні пристрої та сенсори									
Ім'я	id	vendor	time	device_type	protocol	room_id	label	units	status
Тип даних	int	varchar	timestamp	varchar	varchar	int	varchar	varchar	int
Історія отриманих даних із сенсорів									
Ім'я	id	time	protocol	mess_type	value				
Тип даних	int	timestamp	varchar	varchar	int				
Дії пристроїв									
Ім'я	id	action	id_to_action	last_activete					
Тип даних	int	int	int	timestamp					
Наявні кімнати									
Ім'я	id	descr	photo						
Тип даних	int	varchar	varchar						

4.4 Реалізація серверної частини веб-застосунку

У процесі визначення із необхідними засобами для реалізації проєкту, було обрано PHP, як серверну мову програмування. Перед тим, як приступити до виконання даної задачі, необхідно виконати конфігурацію компонентів веб-серверу та обрати доменне ім'я, яке необхідне для зручності у процесі доступу до веб-ресурсу.

При реалізації серверної частини веб-застосунку не буде використано ніяких фреймворків. Буде використано лише деякі публічні бібліотеки для реалізації підтримки обміну інформацією за протоколами MQTT та WebSocket.

Першим файлом, який буде створено, буде файл, який має ім'я index.php. Після створення даного файлу на деякий час залишимо його вміст порожнім. Перш за все необхідно реалізувати функціональну можливість обмеження доступу до веб-застосунку за допомогою авторизації заданих користувачів.

При процесі авторизації скрипт буде здійснювати порівняння даних, які були отримані від користувача та ті, що містяться у базі даних. А саме буде створено веб-сторінку, яка буде містити у собі текстові поля для введення логіну та паролю, а також кнопку, за допомогою якої ці дані будуть надіслані до скрипту на перевірку. Файл, який буде містити функціонал авторизації має назву login.php. Вміст елементів форми авторизації буде передаватися до скрипту за допомогою використання методу POST. Даний метод обраний завдяки тому, що він передає дані у прихованому від користувача вигляді. Задля підвищення безпеки паролі, які зберігаються на сервері у БД, зашифровані за допомогою алгоритму md5.

При написанні скрипту обробки форми авторизації, для вилучення переданих даних використаємо глобальну змінну `$_POST`. Використання даної змінної надає можливість отримати доступ до даних, які були передані браузером із форми. Для прикладу пароль, який був переданий формою буде міститись у масиві, де отримати доступ до саме паролю можна звернувшись до `$_POST['password']`.

Наступними діями скрипту є шифрування отриманого паролю за допомогою алгоритму md5 та виконує порівняння даних із тими, що містяться у БД. У випадку, коли отримані дані від користувача збігаються з тими, що наявні в БД, відбувається запис до глобальної змінної `$_SESSION` дані, завдяки яким, користувачеві немає необхідності у новій авторизації при виконанні переходу на наступну сторінку застосунку. Взагалі сесії використовують не тільки для виконання авторизації, а й для зберігання необхідної інформації, яка необхідно, щоб була доступна у роботі різних PHP скриптів. Надалі для перевірки на наявність доступу будемо у кожному скрипті виконувати перевірку за допомогою глобальної змінної `$_SESSION`.

Наступним кроком буде інсталяція ПЗ Mosquitto, яке являє собою сервер для взаємодії видавця та підписника на повідомлення, які передаються за допомогою протоколу MQTT. Для виконання інсталяції даного ПЗ необхідно створити користувача, від імені якого буде виконуватись процес серверу. Далі необхідно завантажити пакети для інсталяції, зробити це можемо використовуючи наступну команду:

```
wget https://mosquitto.org/files/source/mosquitto-2.0.11.tar.gz
tar zxvf mosquitto-2.0.11.tar.gz
sudo make install
```

Після успішної інсталяції пакетів створимо службовий файл `systemd` для Mosquitto та помістимо у нього наступний вміст:

```
[Unit]
Description=Insite MQTT Broker

[Service]
ExecStart=/usr/local/sbin/mosquitto -c /etc/mosquitto/mosquitto.conf
Restart=always

[Install]
WantedBy=multi-user.target
```

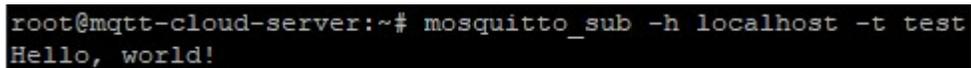
Після успішної інсталяції та конфігурації можемо почати використання щойно встановленого MQTT брокеру.

Створимо файл `mqtt_worker.php`, який буде відповідати за публікацію та отримання повідомлень за допомогою використання протоколу MQTT. Для того, щоб за допомогою використання мови PHP мати можливість обміну повідомленнями за протоколом MQTT, використаємо бібліотеку, яка має назву `php-mqtt`, що доступна на `github` для вільного використання.

Опублікуємо тестове повідомлення за допомогою PHP скрипту для перевірки правильності роботи використаної бібліотеки. Помістимо у файл `mqtt_worker.php` код для тестової публікації повідомлення:

```
$server = '192.168.174.128';
$port   = 1883;
$clientId = 'testScript';
$mqtt = new \PhpMqtt\Client\MqttClient($server, $port, $clientId);
$mqtt->connect();
$mqtt->publish('test', 'Hello World!', 0);
$mqtt->disconnect();
```

Надіслане повідомлення отримають усі, хто підписаний на дану тему. За допомогою терміналу підпишемось на тему, яка має назву `test` та перевіримо успіх публікації за допомогою PHP, якщо отримаємо повідомлення «Hello, world!», то це означає, що все працює коректно. На рисунку 4.7 зображено отримане повідомлення:



```
root@mqtt-cloud-server:~# mosquitto_sub -h localhost -t test
Hello, world!
```

Рисунок 4.7 – Прийняте повідомлення MQTT

Для реалізації можливості підтримки протоколу `Websocket` у поєднанні із PHP, будемо використовувати безкоштовну бібліотеку `Workerman`, яка є доступною на `GitHub`. Створимо файл `websocket_server.php`, який буде виконувати функції серверу, який буде приймати повідомлення від клієнтів та реагувати на них відповідно до

заданих параметрів. Для тестування роботи даної бібліотеки, створимо тестовий скрипт, який буде міститись у файлі `websocket_server.php`:

```
<?php
use Workerman\Worker;
require_once __DIR__ . '/vendor/autoload.php';
$webs_worker = new Worker('websocket://0.0.0.0:2211');
$webs_worker->onConnect = function ($connection) {
    echo "Нове з'єднання\n";
};
$webs_worker->onMessage = function ($connection, $mess) {
    $connection->send('Привіт ' . $mess);
};
$webs_worker->onClose = function ($connection) {
    echo "З'єднання розірвано\n";
};
Worker::runAll();
```

Даний скрипт виводить повідомлення у консоль в залежності від події, яка відбулася, а саме: при новому підключенні виводиться повідомлення про нове з'єднання; при отриманні повідомлення від клієнту, виводить привітання та отримане повідомлення; при розірванні з'єднання виводить відповідне повідомлення. На рисунку 4.8 представлено скріншот консолі серверу Websocket. Можемо побачити, що задані у тестовому скрипті функції виконуються вірно.

```
----- WORKERMAN -----
Workerman version:4.1.4      PHP version:8.1.12      Event-Loop:\Workerman\Events\Select
----- WORKERS -----
-----
proto  user      worker      listen      processes  status
tcp    none     websocket://192.168.174.128:2346  1           [OK]
-----
Press Ctrl+C to stop. Start success.
Нове з'єднання
Привіт, Андрій
З'єднання розірвано
```

Рисунок 4.8 – Запущений сервер Websocket

Протокол MQTT реалізовано за принципом видавець-підписник. А отже для обміну повідомлень було налаштовано на нашому сервері так званий MQTT брокер, за допомогою якого буде відбуватися обмін повідомленнями. На рисунку 4.9 наведено схематичне зображення реалізації обміну повідомленнями MQTT.

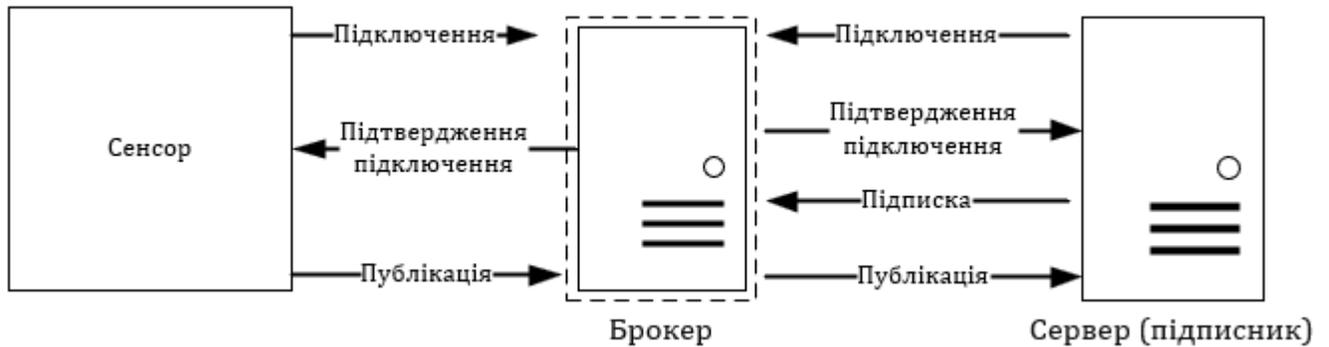


Рисунок 4.9 – Обмін повідомленнями за протоколом MQTT

У ролі підписника буде виступати наш веб-сервер, він буде підписуватись на задані топіки та отримувати повідомлення. При використанні протоколу MQTT дані із сенсорів до серверу будуть передаватись за допомогою формату JSON. Дані, які передаються матимуть такий вигляд:

```
{"id": "53", "vendor": "PST02-A", "device_type": "sensor", "value": "221", "units": "V"}
```

Далі скрипт виконує обробку отриманих даних та робить запит до БД для запису даних, що надає можливість їх зберегти та використати повторно. Для запису даних до таблиці БД, необхідно виконати такий запит:

```
INSERT INTO 'received_data' ('id', 'vendor', 'device_type', 'value', 'units')
values ('53', 'PST02-A', 'sensor', '221', 'V')
```

Для більшого розуміння приведемо на рисунку 4.10 схему взаємодії компонентів при отриманні повідомлення за протоколом MQTT.

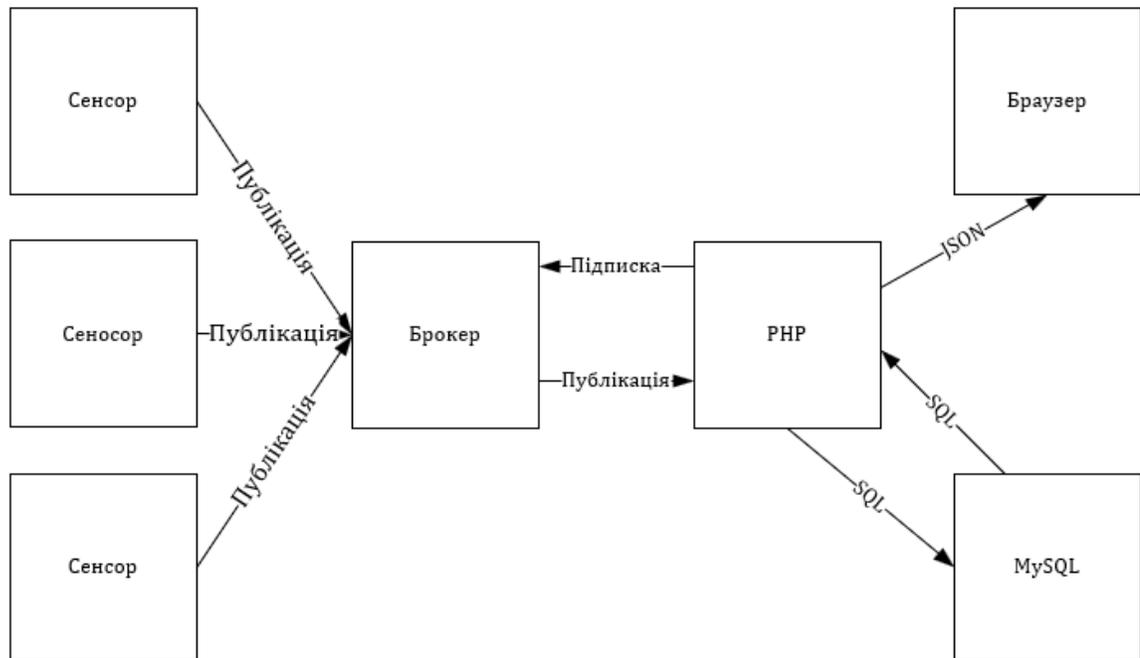


Рисунок 4.10 – Схема взаємодії при використанні протоколу MQTT

Для реалізації можливості обміну даними за протоколом WebSocket необхідно запуснути PHP скрипт, що має відповідний програмний код для керування з'єднаннями та обміну повідомленнями. Даний скрипт необхідно, щоб виконувався у фоновому режимі, а отже запустимо його за допомогою терміналу, використовуючи наступну команду:

```
php websocket_worker.php start
```

Результат виконання даної команди представлено на рисунку 4.11:

```

----- WORKERMAN -----
Workerman version:4.1.4      PHP version:8.1.12      Event-Loop:\Workerman\Events\Select
----- WORKERS -----
-----
proto  user    worker  listen                                processes  status
tcp    ba      none    websocket://192.168.174.128:2346      1          [OK]
-----
Press Ctrl+C to stop. Start success.
  
```

Рисунок 4.11 – Запуск серверу WebSocket

Для виконання ініціалізації з'єднання до WebSocket серверу необхідно сформулювати спеціальний HTTP запит, а сервер належним чином відповідає на цей запит, після чого відбувається встановлення з'єднання, тобто з'єднання TCP залишається відкритим, що дозволяє виконувати обмін даними без необхідності повторного підключення та закриття з'єднання кожного разу при необхідності передачі нових даних. На рисунку 4.12 схематично зображено процес встановлення зв'язку за протоколом WebSocket.



Рисунок 4.12 – Встановлення зв'язку WebSocket

Так само, як і при використанні протоколу MQTT, повідомлення за допомогою протоколу WebSocket будуть передаватись у форматі JSON. Після отримання даних від сенсорів також їх буде записано у БД для реалізації можливості використання у подальшому. На рисунку 4.13 зображено схематичне зображення принципу роботи скрипту, що виконує обробку даних отриманих за протоколом WebSocket.

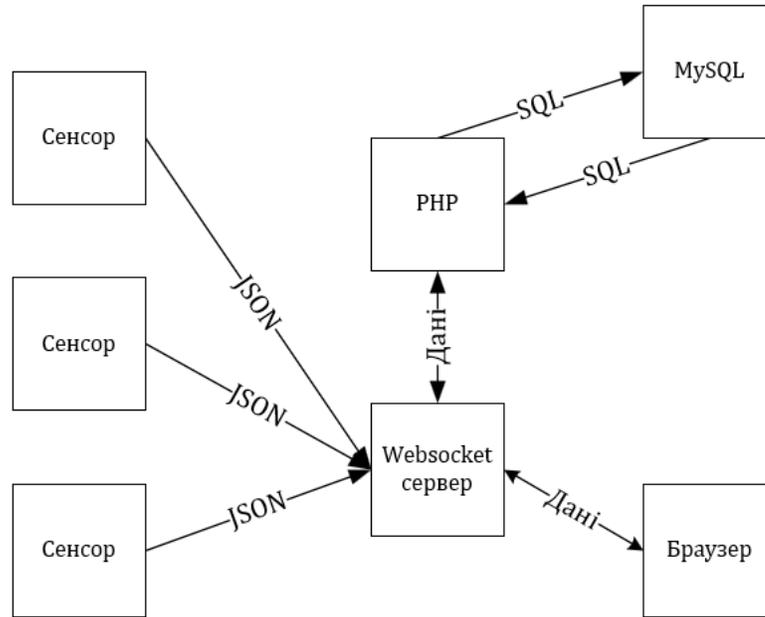


Рисунок 4.13 – Схема взаємодії компонентів при використанні Websocket

4.5 Реалізація інтерфейсу користувача

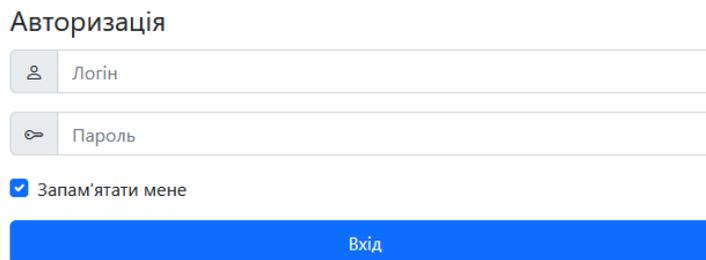
Насамперед необхідно створити файл `index.php`, який, містити у собі головну сторінку веб-застосунку. Для початку створимо у вмісті сторінки базову структуру документу HTML, яка також буде містити у собі тестовий заголовок «Hello world!» та виконаємо підключення JavaScript бібліотеки JQuery та CSS фреймворку стилів Bootstrap.

У результаті, отримаємо веб-сторінку, яка тимчасово містить у собі заголовок «Hello world!». Наступним кроком буде створення файлу `login.php`, що буде виконувати роль сторінки для авторизації. Макет сторінки буде створено за допомогою використання CSS фреймворку Bootstrap, що дозволить зробити зовнішній вигляд привабливим для людини.

Веб-сторінка для авторизації буде містити форму у якій користувач зможе ввести необхідні дані для його авторизації. Для початку створимо блок, який буде містити у собі форму авторизації. Створення блоку відбувається за допомогою тегу `<div>`, для задання стилів блоку використаємо клас `.container`. Присвоєння даного

класу для блоку задає властивість таку, як максимальна ширина, у даному випадку це 1320 пікселів.

Форма для надсилання введених користувачем даних створюється із використанням тегу `<form>`, який містить у собі спеціальні атрибути де задається URL для надсилання даних та метод. Створена форма буде містити два елементи `input`, які являють собою поля для введення даних та мають присвоєне ім'я `login` та `pass`, а також присвоєний тип даних `text` та `password`. Для того, щоб здійснити надсилання даних до серверу, форма містить кнопку відправки. Створення даної кнопки відбувається за допомогою тегу `<button>`. На рисунку 4.14 представлено скріншот створеної сторінки авторизації:



Авторизація

Логін

Пароль

Запам'ятати мене

Вхід

Рисунок 4.14 – Скріншот сторінки авторизації

Наступним кроком реалізуємо головну сторінку, яка буде містити у собі інформацію із сенсорів та вимикачі наприклад для перемикання стану реле. Структура документу буде складатись із двох елементів, а саме шапки, що містить у собі меню та контейнеру, який буде містити основний вміст сторінки. Отримувані дані із сенсорів при відображенні на головній сторінці будуть згруповані за наявними кімнатами. Для того, щоб зменшити видимий обсяг елементів для збільшення зручності перегляду необхідних даних будемо використовувати такий елемент, як акордеон. Для того, щоб створити елемент акордеон, створимо блок за допомогою тегу `<div>` та присвоїмо йому клас `. accordion`. Для додання елементів, які будуть згорнутись використовуємо тег `<div>` із присвоєним класом `. accordion-item`. Після

створення блоків, які можна буде згорнути та розгорнути, всередині створимо декілька блоків, яким присвоїмо клас `.card`. На рисунку 4.15 приведено скріншот створеної сторінки, яка містить дані отримані від сенсорів.

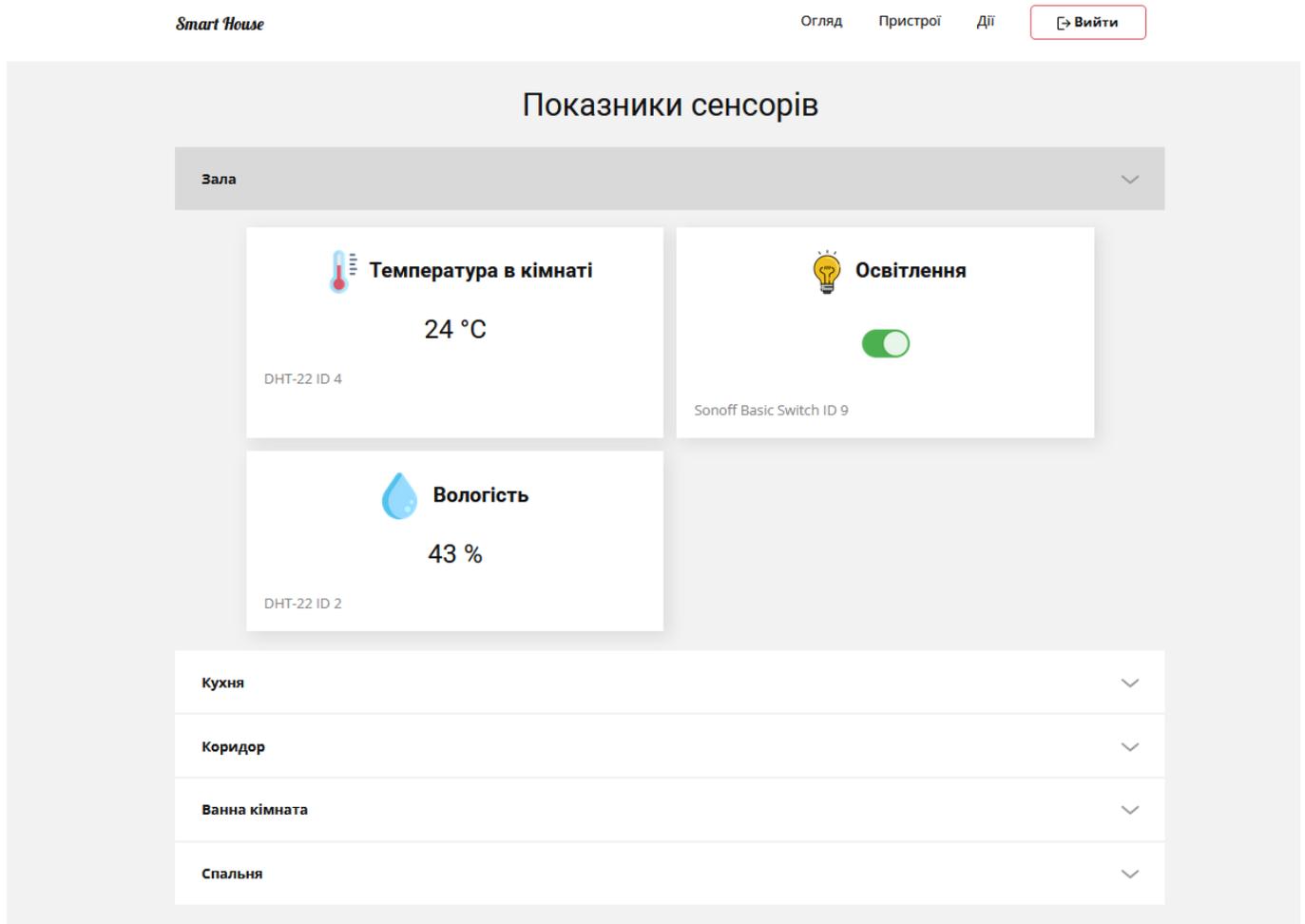


Рисунок 4.15 – Скріншот головної сторінки

Наступним етапом буде створення сторінки `devices.php`, яка буде містити у собі список наявних пристроїв та певну інформацію про них. Дані на сторінці згруповані за назвами кімнат. Для більшої зручності виконано групування за допомогою такого елемента, як вкладення. Для його реалізації необхідно створити блок `<div>` та присвоїти йому клас `.nav-tabs`. Після створення головного блоку, всередині нього створюємо необхідну кількість блоків із класом `.nav-item`, яка відповідає кількості необхідних вкладок. Далі для зручності відображення списку наявних пристроїв

використаємо тег `<table>`, який дозволяє створити таблицю та присвоїмо йому клас `.table`. Після створення таблиці із необхідною розмірністю помістимо у неї інформацію про наявні пристрої, а також декілька функціональних кнопок, за допомогою яких можна виконати певну конфігурацію наявних пристроїв та додати нові пристрої або кімнати. На рисунку 4.16 приведемо скріншот створеної сторінки.

Smart House Огляд Пристрої Дії [Вийти](#)

Зала		Кухня		Коридор		Ванна кімната		Спальня	
Статус	Модель	Тип	Протокол	Опис	Показник	На головну сторінку			
●	Parallax X-band	Сенсор	MQTT	Датчик руху	0		<input checked="" type="checkbox"/>		
●	DHT22	Сенсор	MQTT	Датчик температури	23		<input checked="" type="checkbox"/>		
●	Sonoff Basic Switch	Реле	MQTT	Реле	1		<input checked="" type="checkbox"/>		
●	HC-SR04	Сенсор	Websocket	Датчик відстані	12.1		<input type="checkbox"/>		

[Додати кімнату](#) [Додати пристрій](#)

Рисунок 4.16 – Скріншот сторінки `devices.php`

Останнім кроком буде створення сторінки `actions.php`, яка буде відповідати за налаштування певних базових дій у відповідь на задані критичні показники, що були отримані від сенсорів. Аналогічно створенню попередньої сторінки буде використано таблицю для відображення списку налаштованих дій. На рисунку 4.17 приведено скріншот створеної сторінки.

Пристрій	Якщо	То	Останнє виконання	
Parallax X-band	value = 1	value = 0, Sonoff Basic Switch id 5	06.12.2022 13:48	  
DHT22	value < 18	value = 1, Sonoff Basic Switch id 3	01.12.2022 10:11	  
Gidrolock WSU	value = 1	value = 0, Sonoff Switch id 19	10.11.2022 16:26	  

Рисунок 4.17 – Скріншот сторінки для налаштування дій

4.6 Висновки

У даному розділі було описано процес виконання поставлених задач. А саме інсталяція та конфігурація програмного середовища веб-серверу, інсталяція та конфігурація системи керування базами даних, інсталяція графічного інтерфейсу для СУДБ MySQL, опис структури бази даних, яка необхідна для реалізації функцій веб-застосунку, опис процесу реалізації серверної частини даного веб-застосунку та опис процесу реалізації клієнтської частини даного веб-застосунку.

У процесі виконання поставленої задачі було використано такі програмні компоненти:

- ПЗ веб-серверу Apache
- Система керування базами даних MySQL
- Графічний інтерфейс phpMyAdmin для СУБД MySQL
- Бібліотеки для роботи із протоколами MQTT та WebSocket
- CSS фреймворк Bootstrap

ЗАГАЛЬНІ ВИСНОВКИ

Метою даної кваліфікаційної роботи є виконання модернізації веб-застосунку для візуалізації даних системи «Розумний будинок».

У першому розділі даної роботи було розглянуто поняття та основні можливості технології «розумний будинок». А також було розглянуто поняття та принци роботи сучасних веб-застосунків, а також аналіз їх переваг та недоліків. У процесі аналізу переваг та недоліків веб-застосунків, було зроблено висновок, що їх використання є досить перспективним.

У другому розділі було проведено аналіз готових комплексних рішень, які використовуються для «побудови» системи «розумний будинок». Також у даному розділі було розглянуто наявні технології бездротового зв'язку, що використовують для систем «розумний будинок». При аналізі бездротових мереж, можна зробити висновок, що найбільш оптимальним є використання технології Z-Wave.

У третьому розділі було розглянуто та обрано програмні засоби, які необхідні для виконання модернізації веб-застосунку для візуалізації даних системи «розумний будинок».

У четвертому розділі було описано процес реалізації даного проєкту. А саме інсталяція програмних компонентів для роботи веб-серверу та баз даних, реалізація серверної частини веб-застосунку та інтерфейсу користувача.

Отже, у ході виконання даної кваліфікаційної роботи було виконано поставлені задачі, а саме модернізацію веб-застосунку для візуалізації даних системи «розумний будинок». Було створено серверну частину веб-застосунку та інтерфейс користувача, який доступний за допомогою браузера.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Розумний дім [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Розумний_дім
2. Web application [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Web_application
3. Patrascu M. Integrating Services and Agents for Control and Monitoring: Managing Emergencies in Smart Buildings. Service Orientation in Holonic and Multi-Agent Manufacturing and Robotics. / Patrascu., 2014. – 544 с.
4. Dickson B. How to prevent your IoT devices from being forced into botnet bondage [Електронний ресурс] / Dickson. – 2015. – Режим доступу до ресурсу: <https://techcrunch.com/2016/08/16/how-to-prevent-your-iot-devices-from-being-forced-into-botnet-slavery/>.
5. An Overview of Home Automation Systems [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: <https://ieeexplore.ieee.org/document/7791223/>.
6. Smart house simulation based multi-agent system and internet of things [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: <https://ieeexplore.ieee.org/abstract/document/8259717>.
7. Power Load Event Detection and Classification Based on Edge Symbol Analysis and Support Vector Machine [Електронний ресурс]. – 2014. – Режим доступу до ресурсу: <https://www.hindawi.com/journals/acisc/2012/742461/>.
8. Санітарні норми виробничого шуму, ультразвуку та інфразвуку: ДСН 3.3.6.037-99-2000.
9. Granzer W. P. Security in Building Automation Systems / Wolfgang Praus Granzer. Munich: Appress, 2018. – 578 с.
10. Застосунок [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/Застосунок>.
11. Веб-застосунок [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/Вебзастосунок>.

12. Detection of Malicious HTTP Requests Using Header and URL Features / A. Laughter, S. Omari, P. Szczurek, J. Perry, 2020. – (Future Technologies Conference). – С. 449–468.
13. Gmail [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/Gmail>
14. Duckett J. Web Design with HTML, CSS, JavaScript and jQuery Set / Jon Duckett., 2014. – 1152 с. – (Wiley).
15. Web application protection techniques: A taxonomy. // Journal of Network and Computer Applications. – 2016. – №60. – С. 95–112.
16. Office Online [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Office_Online.
17. Мережевий екран [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Мережевий_екран.
18. "Розумне місто" ("розумний будинок") та його енергетична складова: світовий досвід / Л. М. Побоченко, Ю. Е. Шваюк // Стратегія розвитку України. - 2016. - № 1. - С. 141-145.
19. Андреева О.В. Концепції розвитку безпроводних рішень для охоронних функцій цифрового будинку : збірник тез доповідей XIII Міжнародна наукова практична конференція, «Інтегровані інтелектуальні робото-технічні комплекси», 19–20 травня 2020 р. – К.: НАУ, 2020. – С. 269-271.
20. Wi-Fi [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/Wi-Fi>.
21. Seyed M. Bluetooth Low Energy Mesh Networks: A Survey / M. Seyed, C. Gomes. // Department of Network Engineering, Universitat Politècnica de Catalunya., – 2017. – С. 1–19.
22. Zigbee 101: посібник для початківців [Електронний ресурс] – Режим доступу до ресурсу: <https://dou.ua/forums/topic/38080/>.

23. Unwala, Z. Taqvi and J. Lu, "IoT Security: ZWave and Thread," 2018 IEEE Green Technologies Conference (GreenTech), 2018, pp. 176-182, doi: 10.1109/GreenTech.2018.00040.
24. Багатоплатформність [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/Багатоплатформність>.
25. The Java Virtual Machine Specification, Java SE 7 Edition (Java Series), 2013. – 608 с. – (Addison-Wesley Professional).
26. What is PHP? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.php.net/manual/en/intro-what-is.php>.
27. HTML | MDN [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/ru/docs/Web/HTML>.
28. Apache HTTP Server [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Apache_HTTP_Server.
29. URL [Електронний ресурс] – Режим доступу до ресурсу: <https://ru.wikipedia.org/wiki/URL>.
30. Clément N. Nginx HTTP Server - Second Edition / Nedelcu Clément., 2013. – 301 с.
31. November 2022 Web Server Survey [Електронний ресурс] – Режим доступу до ресурсу: <https://news.netcraft.com/archives/2022/11/10/november-2022-web-server-survey.html>.
32. About the Apache HTTP Server Project – The Apache HTTP Server Project [Електронний ресурс] – Режим доступу до ресурсу: https://httpd.apache.org/ABOUT_APACHE.html.
33. PHP [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/PHP>.
34. Learning PHP, MySQL & JavaScript: With JQuery, CSS & HTML5, 2015. – 782 с. – (O'Reilly).
35. HTML [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/HTML>.

36. Syed Fazle Rahman. Jump Start Bootstrap. — SitePoint, 2014. — 150 с.
37. Proceedings of the ACM on Programming Languages. // Association for Computing Machinery. — 2020. — С. 1–168.
38. AJAX - JavaScript довідка [Електронний ресурс] – Режим доступу до ресурсу: <http://яваскрипт.укр/AJAX>.

Додаток А

Переклад першого розділу

1. THE CONCEPT OF SMART HOME TECHNOLOGY AND THE WORKING PRINCIPLE OF WEB APPLICATIONS

1.1 Concepts and main capabilities of the "Smart House" technology

The "smart home" technology is a system of home devices that, in turn, can perform certain actions and solve some everyday tasks without the participation of a person. We can say that all electrical devices in the room are functionally connected to each other, which in turn makes it possible to centrally control these devices [1].

In connection with the increase in the computing capabilities of devices, the "smart house" technology gradually received its extension into the "IoT - Internet of Things" system, according to which the primary standardization was carried out and the main recommendations and rules for the implementation of the finished product were outlined, both at the level of individual components as well as at the level of a complete system. Although the technology is relatively new, there are already dozens of ready-made solutions on the market.

The basis of the "smart house" technology is the ability to recognize specific situations that occur in the room and respond accordingly to the given situation. One of the main features of the "smart house" is the combination of several subsystems into a unified controlled complex (Figure 1.1).

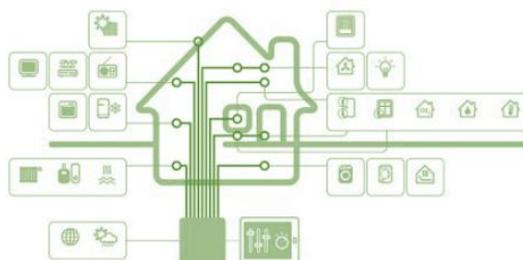


Figure 1.1 - Schematic representation of the "Smart House" system

One of the most important features of the "smart house" system, which makes it different from other types of living space organization, is that this concept is the most progressive in the interaction between a person and a living space, and this is what makes it possible to set the desired environment with just one simple command. and in the future, automatic systems monitor the operation of all connected electrical appliances and engineering systems [3]. In this case, the need to use several remote controls, a large number of switches for controlling lighting devices, separate control units for controlling heating and ventilation systems, security systems (alarm and video surveillance) and others disappears when watching television. When using the "smart home" technology, even one click on the remote control is enough. After choosing the necessary scenario, the system begins to independently adjust the system operation in accordance with the choice of the person who chose the scenario or depending on external factors such as external lighting, air humidity, weather, time of day [4].

The concept of premises using "smart house" technologies includes provisions [4]:

- Development of an integrated room management system, that is, a system capable of ensuring the integrated operation of all room systems, namely: access control, ventilation, lighting, heating, water supply and many others.
- Solving the set scenarios and tasks for the subsystems of the built-in room management system and the absence of service personnel in need. The so-called "intelligence" is embedded in the above-mentioned subsystems, which is a certain algorithm of actions that corresponds to changes in the indicators of the sensors available in the system and other events, such as emergency situations.
- A mechanism for immediate shutdown and, if necessary, transfer of controls to a person for any subsystem of an intelligent building. At the same time, the same and

convenient access to control and visualization of all parts of the "smart home" system for a person should be provided.

- Guaranteeing the correctness of work for specific subsystems in case of failure of the general control system or any other elements of the system.
- The cost of modernization and maintenance of premises systems should be minimal, and the construction of subsystems should be ensured by the use of common standards, automatic detection and configuration of new devices in the process of adding them to the existing system.
- Existence of a ready-made communication environment in the premises, which is necessary for connecting modules and devices to it. In addition, it is necessary to have the possibility of using different types of physical channels as a communication medium, for example, low-current lines, radio channel, power lines.

The "smart house" system includes the following automation objects [5]:

- Management server;
- Lighting control;
- Equipment control from touch control panels;
- Control of electric drives;
- Control of emergency conditions and electrical loads;
- Climate control;
- Systems for organization of access control;
- Management of the ventilation system;
- Security and fire alarm system;
- Centralized management of systems;
- Video surveillance systems;
- Multiroom;

A building with an existing smart home system has many advantages. The management system enables the owner to develop operating procedures of any complexity, because all elements of this system are able to work in a coordinated and joint manner. An example of the development of procedures that save resources [6]:

1. Ensuring security and access management;
2. Management and accounting of most system parameters and in case of their critical changes, instant response.
3. Remote control and management of the premises, because in such a system all control and information communication channels are digital.

With the help of even one touch, it is possible to make your home more comfortable, namely, a comfortable microclimate for a specific person will be established, the curtains will be lowered, the bath will be filled, and the lighting will be turned on. It is possible to even heat food without getting up from your seat, you just need to place it on the stove in advance. It is possible to control media equipment using touch panels [6].

In lighting control, it is possible not only to control the brightness of the lamp when it is turned on, but also the time during which the specified brightness of the lamp will be reached. For this, you need to use special light regulators. Basically, the function of constant backlighting is necessary for use in office premises, which allows you to maintain a given illumination in a room or any surface, regardless of external factors. The ability to turn on outdoor lighting automatically depending on the presence of people or the time of day can provide additional comfort to the residents of the house, as well as deter intruders. For these purposes, it is necessary to use special light sensors, an example is presented in Figure 1.2.

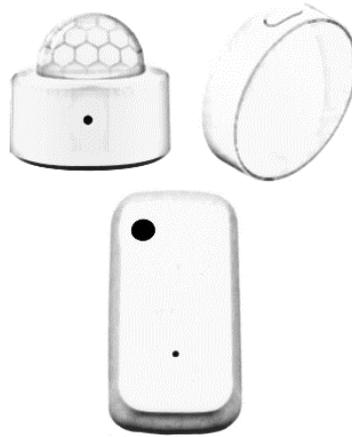


Figure 1.2 - Light sensors

The "smart home" system constantly measures the temperature in each individual room individually and takes certain actions to maintain it at a certain set level. To maintain the temperature, it is used, for example, to control the radiator valves or, if necessary, automatically controls the ventilation system [7]. In addition, this system helps the owners of the premises in saving money, which is achieved due to the variety of operation modes: night, comfort mode or mode when there is no one in the premises. These modes can be changed both according to the schedule and in manual mode by command. To do this, it is enough to set the required temperature on the touch panel according to the required mode. If there are sensors on the windows, it is possible to automatically turn off the air conditioning or heating system if the system determines that the windows have been opened for ventilation. This will also save energy and money. To perform such tasks, you can use the controller shown in Figure 1.3.



Figure 1.3 – Temperature control controller ОВЕН TPM 32

In the warm season, it is possible to automatically adjust the blinds, which helps prevent excess sunlight from entering the room without reducing the light flow. Due to this, the room heats up less from sunlight and helps to save money due to the saving of electricity consumed by the air conditioner.

During the absence of the owners, the "smart house" system will compile a report on the events that took place on the territory of the house [5]. When an extraordinary situation occurs, for example, a water leak, the "smart house" system will perform the necessary actions to localize the accident, for example, stop the water supply and notify the necessary service, as well as the owner of the house. When intruders enter the premises, the "smart home" system will notify the owner of the premises through all possible communication channels and call the security service.

If it is necessary to create the effect of presence, then the "smart house" system can also simulate the usual way of life when the owners are absent. For example, turn on music and lighting.

The use of "smart house" technologies enables centralized interactive management and control of various types of premises, such as office premises, residential premises and any others [8].

In general, the scheme of the control system can be described as follows [8]:

1. Main control unit
2. Devices for reading indicators (illumination sensors, temperature sensors, motion sensors, and others)
3. Control devices (infrared emitters, relays and others)
4. Interfaces for management (touch panels, web interfaces, remotes, switches and others)
5. A control network that combines all the elements mentioned above
6. Controlled devices (lights, ventilation systems, media components and others)

7. Auxiliary networks (for video and audio signal delivery, telephone, Ethernet)
8. Software

The main functionality of the central control unit includes the control of devices subordinate to it. The following interfaces are used to control such devices: IR, Rs-232, RS-485, Ethernet, analog, digital, etc. The central control unit contains a multitasking OS and in some cases may contain its own web server. Sensors used in the "smart house" system are connected to each other through intermediate devices or directly to each other through a single network and are located in certain places of the building. General control of the "smart house" complex is carried out through control interfaces [6].

- The central control unit receives information from the devices for measuring indicators, this happens with the help of its own control network.
- The software installed on the control unit processes the received information and returns commands to the control devices.

Commands for execution can come from the own network, as well as through the auxiliary network. The composition and form of information for data visualization, which contain information about the state of various systems, as well as command generation methods are established at the software development stage, taking into account the requirements of the given project.

1.2 Concept and overview of the working principle of modern web applications

Most PC users have an idea of what a typical application that runs on the Windows operating system is. Namely, an application is a computer program with the help of which a PC user can solve the specific application tasks he needs [10]. Examples of such programs can be media players, web browsers, mail clients, and many others.

The web application is a distributed application, because its logic is located on the server, and in turn, the function that displays the information downloaded from this server is on the web browser installed on the user device [11]. Such applications are usually executed

on the side of the web server, due to which they create an interface between the website and the user. Development of web applications mainly takes place using scripting programming languages such as PHP, Perl and others. Much less often, high-level languages that must be compiled for each specific operating system (C, C++, and others) are used in the development of web applications.

The web server receives requests from clients using the HTTP protocol. After receiving and processing the request from the client, the server provides a response in the form of an HTML page or other data. To get access to the required page on the web server, you need to know its URL [12].

During the development of the World Wide Web, along with it, web applications also improved and developed quite rapidly. One of the most important features of a web application is the ability to access it from almost any user device and from any place where Internet access is available. Another important feature of web applications is their flexibility, security and scalability. The use of web technologies in application development makes it possible to obtain such a property of the application as multi-platform. This property of the software allows you to work with the developed application in any software environment, therefore, with the help of this property, you can solve the problem of compatibility of the software on different platforms.

As already explained, a web application is a program that runs in a web browser, which is why such an application requires network access and a web browser installed on the user's device. Therefore, the functions of calculations, processing and storage of information consist of a remote web server, and the functions of the so-called client are performed by the web browser installed on the user's device. Figure 1.4 shows the schematically depicted principle of the web application:

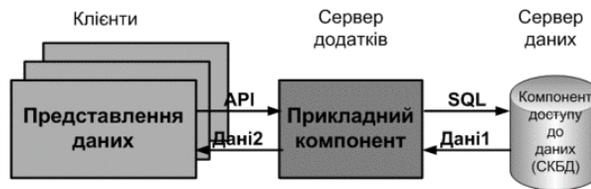


Figure 1.4 – Schematic representation of the principle of operation of the web application

An example of a web application to consider is the well-known email client Gmail. When using this web application, the user does not feel a fundamental difference in the main functionality compared to the classic client software for working with e-mail, which must be installed on your PC or mobile device. The web interface of this e-mail client was developed using the JavaScript browser scripting language. Due to the fact that the JavaScript language was used in the development of the web interface, the user can send control commands using the keyboard when working with the application, update pages without completely reloading them (this possibility is provided by AJAX technology), and others [13].

In the creation of modern web applications, most developers use JavaScript technology, which makes it possible to perform some functionality without the need to make requests to a remote server, which allows you to reduce the load on the server itself and speed up the execution of certain tasks. In the period when the JavaScript language was just beginning its development, it was not used to perform any practical tasks, but simply because it existed. In the early days of this technology, it was only used to make elements on a web page respond to mouse movements, dynamically control element styles, and so on. Therefore, at a certain time, the use of this technology became a bad tone [14].

The reason for the rapid development of the popularity of web applications today is the beginning of the active use of JavaScript technology that exists in our time. After all, in the past, their lack of dynamism and interactivity during user interaction with a web page was always considered a drawback in the development of web applications. The

development of JavaScript to the level that is available today completely helps to solve this problem.

So, as previously explained, in off-the-shelf software that is installed on users' PCs, the application logic resides directly within the PC or other user device, unlike web applications, where the application logic resides on a remote server. The location of the application logic on a remote server enables the existence of a single version of the application that is always up-to-date, thanks to this feature, the application is very easily distributed among all its users. The use of web technologies in the development of applications allows you to move away from the classic distribution of the application, since in this case there is no need for the user to receive a full copy of the application. Thanks to this feature, users get only the part they need to work with the application, namely the application interface. Based on the above, it can be said that there are much fewer problems with the distribution of web applications compared to the distribution of classic applications, because it is possible to access the web application from any user device that is connected to the Internet.

The main component in working with a web application is a program such as a web browser. To start working with the web application, the user only needs to specify the required URL in the address bar of the web browser.

The difference between a classic application that a user installs on their device and a web application is that the latter is located on a remote server. This feature allows the user not to perform the role of an administrator. For example, the user does not need to install software, he does not need to solve problems with this software, because the role of administrator in the case of a web application is performed by a person or a group of people who directly develop the application. On the one hand, this model implies an increased load on people who develop the application, but on the other hand, a model where a small group of programmers who are engaged in the development of the application and are concentrated in one place is an economically much more profitable solution. That is why it is more

efficient and more profitable in economic terms, because in this case there is no need to maintain a certain team of specialists who would have the task of installing and configuring applications on users' devices.

There are no user requirements in using the web application. In the development of a web application, it is assumed in advance that a web browser is already installed on the user's device, because this program in the modern world is an integral part of the software environment on electronic user devices. The main idea behind using web applications is that they can be used on almost any OS installed on user devices. This possibility is due to the fact that the web application does not have special requirements for hardware platforms in its work. After the beginning of the widespread distribution of web applications in the past, the problem remained with the need to support different versions of the developed application. Namely, when the developer of the application has published a new version of it, then without exception all users of this application receive the latest published version, for this it is enough to reload the web page. Immediately after the publication of the current version of the application, all versions of the web application that are considered obsolete disappear. Thanks to this feature, users of the application may not even notice that they have already started using the updated version of the web application. Also, thanks to this feature of web applications, the problem of backward compatibility of applications and implementation of support for outdated versions is solved [15].

One of the optimal use cases for web applications is when the user's devices have limited storage capacity. After all, the obvious advantage of using a web application on users' devices with limited storage capacity is that when using such applications, there is no need to download a full-fledged application to your device. In some cases, there is even no need to fully load the interface, and only a part of it, which is necessary to perform the task, is enough. And, therefore, web applications take up little space on the storage of the user's device, which allows for faster loading and response to user actions.

Thanks to the distributed architecture of the web application, its architecture is practically invisible to the average user, which provides opportunities for scaling the computing capabilities of the program without the need to interfere with the operation of the user's devices. Namely, the administrator can modernize the hardware of the server that contains the web application imperceptibly for the user, distribute the load on the application among several different servers, and more.

When using web applications, it is necessary to understand that they cannot have some features that are available in classic desktop applications. For example, the user does not have the ability to control the hardware components of the device using a web application, work with complex graphics, and more. Nowadays, web technologies are developing quite rapidly, which is why all of the above is relevant only for today and the near future.

Observing the development of web technologies, we can say that one of the most relevant technologies in the future will be web applications. After all, they develop rapidly every day and come as close as possible to the classic software that we use almost every day. For an example of such applications, we can consider software from the Microsoft company - "Office Online". This software is a package of office applications that are developed as web applications. It is essentially a lightweight version of the classic Microsoft Office desktop software, and has all the main features of its desktop counterpart. When using Office Online, there is no need to install additional software on users' devices, this gives them the opportunity to work with documents from almost any place where there is an Internet connection [16].

1.3 Analysis of advantages and disadvantages of web applications

Before starting the development of the application, the developer faces the task of defining the type of application being developed. At this stage, the developer has several options. One of these options is the development of a classic application, which is intended for a specific software and hardware platform, for the implementation of which it is

necessary to use the development environments and compilers recommended for each platform. An alternative option is the development of an application using web technologies, in which there is such a feature as the fact that such an application will not depend on the software and hardware platform of the user device. This stage is one of the most important, because the next stages in the development of the project depend on it. At this stage of the development of the application, we can already understand the approximate cost of its development, because at this stage the involvement of employees who will develop the application for the chosen platform is being carried out.

Web technologies that are available today are quite a vivid example of the fact that a well-designed technology can be used by people in a wide variety of scenarios and situations.

The fundamental difference between a web application and a classic desktop application is that the classic application interacts with the OS installed on the device, while the web application interacts only with the web browser installed on the user's device. Thanks to this feature, it is possible to significantly expand the number of users who can work with the application, because a person who uses an application based on web technologies has the opportunity to use it on almost any device with a web browser. When using HTML - the hypertext markup language, the developer has the opportunity to develop a user interface that has adaptive markup. This means that the interface and its elements will adapt to different sizes of displays on users' devices. But although web applications have a great advantage due to their independence from the software and hardware platforms on which they work, there is also a rather important drawback - much lower performance compared to classic applications that are optimized for each specific platform.

Classic applications differ from applications based on web technologies in that such an application is always stored on user devices. Today, most users are already familiar enough with installing classic applications, but developers cannot be sure that all users will work with the same version of the developed application.

Comparing classic applications and web applications, we can see many disadvantages and advantages of using one or another method of application development. Due to the fact that classic applications can directly use the hardware of the device, it is much easier for the developer to work with such components as the camera, microphone, location services and others. Also, the advantage of classic applications is that a large number of such applications can be found in so-called application stores. This possibility gives the user more confidence in the security of the installed application, because before publishing the application on such a site, it passes a check. Along with numerous advantages, there are also disadvantages in using classic applications. Usually, the development of a classic application requires more funds, which is especially relevant for those applications that need the ability to work on different hardware and software platforms. Even if you ignore the cost of developing such an application, it is much more expensive to maintain and update it, especially if such an application is designed for use on multiple platforms of user devices. Another disadvantage is the fact that users can use different versions of the application, which causes problems in development, as well as compatibility between different versions of the application.

When comparing a classic application with a web application, we can see that working with web applications is much easier for the user, because at any time and from any place the user can access the current version of the application, all that is needed is access to the Internet .

Web applications also have their advantages and disadvantages. One of the advantages of web applications is that they have a common code base for many devices, as well as the ability to quickly adapt to work with outdated devices, which is why web applications are easier to maintain. Due to this, there is no need to download web applications to users' devices, so we can say that this is also their advantage, which is especially relevant for users with limited storage space. For the developer, it doesn't matter what type of user's device or platform it is, because the web application runs in a web browser, which makes it possible to ignore it. Thanks to this feature, users have the opportunity to use the application on any

device, provided that they have a web browser. The application logic resides on a remote web server, and the user receives only the user interface on their device. This structure of the application allows you to eliminate the problem of compatibility of different versions of it, because this structural feature enables all users to use one single, up-to-date version of the application. Today, many users are constantly using various electronic devices, but the majority of experienced users are still not very large. Because of this, another advantage of using web applications is that there is no need to install and configure them, because sometimes for the correct operation of the application, it is necessary to perform some settings, which can be quite a difficult task for an inexperienced user.

Along with the advantages of using web applications, there are also disadvantages. Web applications have fairly limited access to the hardware components of the user's device. In addition, an existing Internet connection on the user's device is necessary for the full operation of the web application, because all the logic of the application is located on a remote web server, and the user only downloads the user interface.

1.4 Conclusions

In this section, the concept and main capabilities of the "smart house" technology, the concept and overview of the working principle of modern web applications and the analysis of their advantages and disadvantages were considered.

Considering the concepts and main capabilities of the "smart house" technology, we can conclude that the use of this technology is promising. Thanks to the use of this technology, it is possible to significantly simplify the performance of everyday tasks for a person and make his stay at home more comfortable and convenient. Thanks to the possibility of optimizing the operation of various systems that use energy resources when using the "smart house" technology, the owner has the opportunity to significantly save his own money.

Considering the concepts and principles of operation of web applications, their architectural component and the principle of their operation were considered. Namely, the web

application consists of two key parts - the server part and the user part. The essence of the user part is to provide the user with an interface to interact with the application, while the logic of the application is located on a remote web server. The server part includes such components as web server software that accepts requests to it using the HTTP protocol and application logic written in a specific programming language.

The advantages of using web applications are that there is no problem with ports closed by a firewall, because basically such ports, which are necessary for the operation of web applications, are not closed by a firewall [17]. Therefore, the main advantage of web applications is the ability to work on various software and hardware platforms without the need for additional adaptation of the application.

Додаток Б

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
«ПОЛТАВСЬКА ПОЛІТЕХНІКА ІМЕНІ ЮРІЯ КОНДРАТЮКА»

ЗБІРНИК НАУКОВИХ ПРАЦЬ
за матеріалами VIII Всеукраїнської науково-практичної конференції
**«ЕЛЕКТРОННІ ТА МЕХАТРОННІ СИСТЕМИ:
ТЕОРІЯ, ІННОВАЦІЇ, ПРАКТИКА»**
04 листопада 2022 року



Полтава 2022

УДК 004.89 + 681.51

Збірник наукових праць за матеріалами VIII Всеукраїнської науково-практичної конференції «Електронні та мехатронні системи: теорія, інновації, практика», 4 листопада, 2022 р. / Національний університет «Полтавська політехніка імені Юрія Кондратюка».

Редколегія: О.В. Шефер (головний редактор) та ін. – Полтава: НУ «Полтавська політехніка імені Юрія Кондратюка», 2022. – 100 с.

У збірнику представлені результати наукових досліджень та розробок в області сучасних електромеханічних систем та автоматизації, електричних машин і апаратів, моделювання та методів оптимізації, енергозбереження в електромеханічних системах, управління складними технічними системами, проблем аварійності та діагностики в електромеханічних системах та електричних машинах, інформаційно-комунікаційних технологіях та засобах управління. Призначений для наукових й інженерно-технічних працівників, аспірантів і магістрів.

Матеріали відтворено з авторських оригіналів та рекомендовано до друку VII Всеукраїнської науково-практичної конференції «Електронні та мехатронні системи: теорія, інновації, практика». Редакція не обов'язково поділяє думку автора і не відповідає за фактичні помилки, яких він припустився.

Відповідальний за випуск - д.т.н., професор О.В. Шефер.

Редакційна колегія:

О.В. Шефер – головний редактор, доктор технічних наук, професор, завідувач кафедри автоматичної електроніки та телекомунікацій Національного університету «Полтавська політехніка імені Юрія Кондратюка»;

Н.В. Єрмілова – кандидат технічних наук, доцент кафедри автоматичної електроніки та телекомунікацій Національного університету «Полтавська політехніка імені Юрія Кондратюка»;

С.Г. Кислиця – кандидат технічних наук, доцент кафедри автоматичної електроніки та телекомунікацій Національного університету «Полтавська політехніка імені Юрія Кондратюка»

Б.Р. Боряк – кандидат технічних наук, доцент кафедри автоматичної електроніки та телекомунікацій Національного університету «Полтавська політехніка імені Юрія Кондратюка».

© Національний університет
«Полтавська політехніка імені Юрія Кондратюка»

Н.В. Єрмілова, Є.О. Єндіяров ВИКОРИСТАННЯ КРОКОВИХ ЕЛЕКТРОПРИВОДІВ В УСТАНОВКАХ НАНЕСЕННЯ ТОНКИХ ПЛІВОК ПРИ ВИРОБНИЦТВІ МІКРОСХЕМ.....	29
Л.І. Леві, А.В. Базарний МОДЕРНІЗАЦІЯ WEB-ЗАСТОСУНКУ ДЛЯ ВІЗУАЛІЗАЦІЇ ДАНИХ СИСТЕМИ «РОЗУМНИЙ БУДИНОК».....	32
О.В. Шефер, О.С. Ястреба, В.С. Ястреба ОСОБЛИВОСТІ ОПТИМІЗАЦІЇ ІНФОКОМУНІКАЦІЙНИХ МЕРЕЖ.....	34
Г.М. Кожушко, С.Г. Кислиця, Д.В. Кислиця СИСТЕМИ АВТОМАТИЧНОГО КЕРУВАННЯ ОСВІТЛЕННЯМ – ЕФЕКТИВНИЙ ШЛЯХ ЕКОНОМІЇ ЕЛЕКТРОЕНЕРГІЇ.....	36
О.Г. Дрючко, Б.Р. Боряк, Р.В. Захарченко, В.І. Троянський, В.В. Жданов ТЕНДЕНЦІЇ ПОБУДОВИ МЕХАТРОННИХ СИСТЕМ СУЧАСНИХ ЕЛЕКТРОМОБІЛІВ.....	38
Ю.Р. Зоураб, Р.М. Царьков, Р.О. Єрмілов МЕТОДИКИ ТА ОСНОВНІ ЕЛЕМЕНТИ СУЧАСНИХ СИСТЕМ ТЕХНІЧНОГО ЗОРУ РОБОТІВ.....	40
С.Г. Кислиця, С.І. Демус МУЛЬТИСЕРВІСНА МЕРЕЖА ЯК ТЕХНОЛОГІЯ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ КОМУНІКАЦІЙ.....	43
О.Г. Дрючко, О.В. Шефер, О.В. Сухоребрий, Д.О. Ненич, В.П. Будім ЕФЕКТИВНІСТЬ ВВОДУ ВИПРОМІНЮВАННЯ ДЖЕРЕЛ В ОПТИЧНЕ ВОЛОКНО.....	45
О.В. Шефер, Є.М. Плутцов МОДЕРНІЗАЦІЯ СИСТЕМИ БЕЗПЕКИ ТА КОНТРОЛЮ ДОСТУПУ З ІНТЕГРАЦІЄЮ У ТЕЛЕКОМУНІКАЦІЙНУ МЕРЕЖУ ТОВ «ІНДУСТРІАЛЬНІ СИСТЕМИ АВТОМАТИЗАЦІЇ».....	48
О.І. Безверхий, В.А. Дворук, Р.Т. Азізов РОЗРОБКА ТА ВСТАНОВЛЕННЯ НА ХОСТИНГ ІГРОВОГО СЕРВЕРУ ЗА ДОПОМОГОЮ JAVA ТА ORACLE CLOUD.....	49
М.Б. Вітер, Д.В. Коровін, Г.О. Швидков АВТОМАТИЗАЦІЯ РЕКРУТЕРСЬКОЇ ДІЯЛЬНОСТІ В ІТ КОМПАНІЯХ....	50
М.К. Бороздін, Р.Р. Кирпота ЗАМІНА СИСТЕМИ ГОЛОВНОГО ПРИВОДА НА ТИРИСТОРНИЙ ПЕРЕТВОРЮВАЧ НА ПРОКАТНОМУ СТАНІ.....	52
О.І. Безверхий, В.О. Гулевич, В.В. Діхтяренко РОЗШИРЕННЯ ФУНКЦІОНАЛУ ОБРОБКИ ЗАМОВЛЕНЬ.....	55

УДК 004.42

Л.І. Леві, д.т.н., професор,

А.В. Базарний, магістрант

Національний університет «Полтавська політехніка імені Юрія Кондратюка»

МОДЕРНІЗАЦІЯ WEB-ЗАСТОСУНКУ ДЛЯ ВІЗУАЛІЗАЦІЇ ДАНИХ СИСТЕМИ «РОЗУМНИЙ БУДИНОК»

Станом на сьогоднішній день концепція системи «розумний будинок» стає все більш популярною. По мірі розвитку даної технології на ринку з'являлося дедалі більше рішень для побудови своєї системи «розумний будинок», а саме різноманітні сенсори та електронні пристрої. Функціональні можливості таких готових пристроїв обмежуються тими, що були закладені при розробці пристрою виробником, через що у користувача відсутня можливість у гнучкому налаштуванні функцій системи «розумний будинок». Для того, щоб додати необхідної гнучкості своїй системі «розумний будинок», необхідно комбінувати різні сенсори та пристрої від різних виробників та навіть використовувати так звані DIY (Do It Yourself) пристрої.

У випадках, коли існує потреба у об'єднанні обладнання від різних виробників, які для обміну даними використовують різні протоколи передачі, існує проблема у виборі засобів візуалізації даних із системи «розумний будинок».

Вирішенням даної проблеми є створення застосунку, який дозволяє провести візуалізацію даних із різних сенсорів та пристроїв системи «розумний будинок». У результаті чого буде наявна можливість зручного сприйняття людиною інформації отриманої із системи «розумний будинок».

Особливістю даного веб-застосунку є наявна можливість обміну даними із системою «розумний будинок» за допомогою використання комбінації декількох протоколів передачі даних, а саме MQTT (Message Queuing Telemetry Transport), WebSocket, SSE (Server Side Events) та HTTP-запитів.

MQTT (Message Queuing Telemetry Transport) являє собою відкритий протокол, який був спеціально розроблений для обміну даними між пристроями інтернету речей. Для реалізації процесу обміну повідомленнями використовується принцип публікація/підписка (publish/subscribe). Одними із головних аспектів при розробленні такого типу системи є мінімізація пропускну здатності комп'ютерної мережі та використання апаратних ресурсів приладів. Концепція передачі даних виглядає наступним чином. Певна кількість клієнтів підключається до так званого сервера посередника (брокера) та повідомляє, яку тему вони хочуть прослуховувати. Коли інший клієнт виконує публікацію свого повідомлення у даній темі, то брокер виконує пересилання цього повідомлення усім клієнтам, які підписані на дану тему.

WebSocket являє собою протокол, задачею якого є обмін інформацією між клієнтом та сервером у режимі реального часу. За допомогою даного протоколу

наявна можливість у забезпеченні повнодуплексного каналу зв'язку із використанням одного TCP-сокета.

На рисунку 1 представлено схематичне зображення клієнт-серверної взаємодії MQTT та станів зв'язку протоколів WebSocket та HTTP.

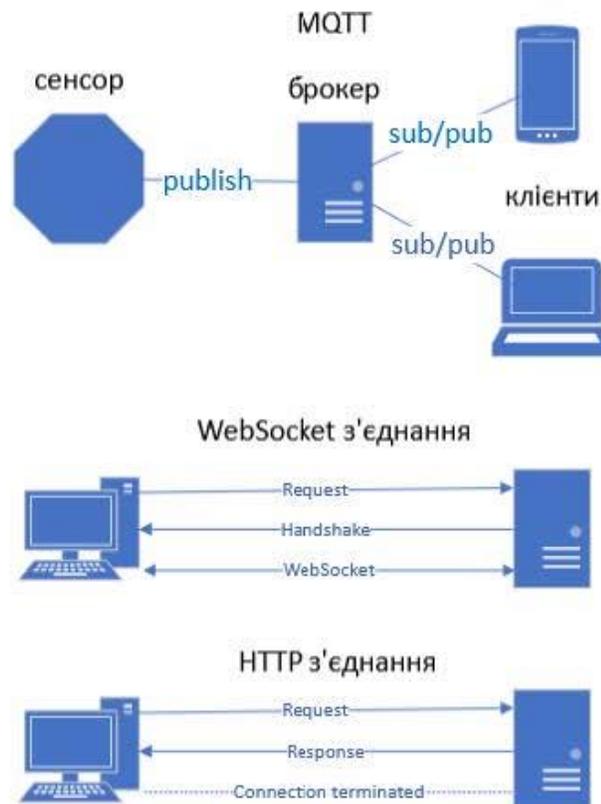


Рис. 1. Схематичне зображення клієнт-серверної взаємодії MQTT та станів зв'язку протоколів WebSocket та HTTP

ЛІТЕРАТУРА:

1. Gaston C. H. *MQTT Essentials - A Lightweight IoT Protocol* / Hillar C. Gaston., 2017. – 163 с.
2. Lombardi A. *WebSocket: Lightweight Client-Server Communications* / Andrew Lombardi., 2015. – (O'Reilly).
3. *Comparison with HTTP and MQTT In Internet of Things (IoT).* – 2019. – С. 13.

MODERNIZATION OF THE WEB APPLICATION FOR THE VISUALIZATION OF SMART HOUSE SYSTEM DATA

L. Lievi, ScD, Professor,

A. Bazarnyi, Master's student

National University «Yuri Kondratyuk Poltava Polytechnic»

Додаток В

ТЕМА: МОДЕРНІЗАЦІЯ WEB-ЗАСТОСУНКУ ДЛЯ ВІЗУАЛІЗАЦІЇ ДАНИХ СИСТЕМИ «РОЗУМНИЙ БУДИНОК»

Мета: створення програмного комплексу для візуалізації даних із системи розумний будинок на основі веб технологій.

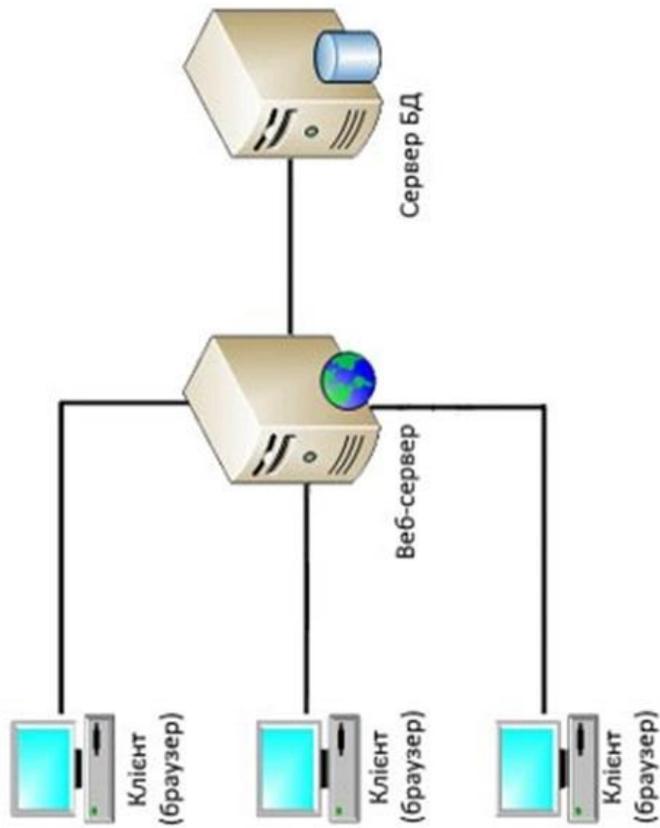
Об'єкт дослідження: необхідні програмні засоби при модернізації web-застосунку.

Предмет дослідження: принципи роботи та створення сучасних web-застосунків.

Задачі кваліфікаційної роботи:

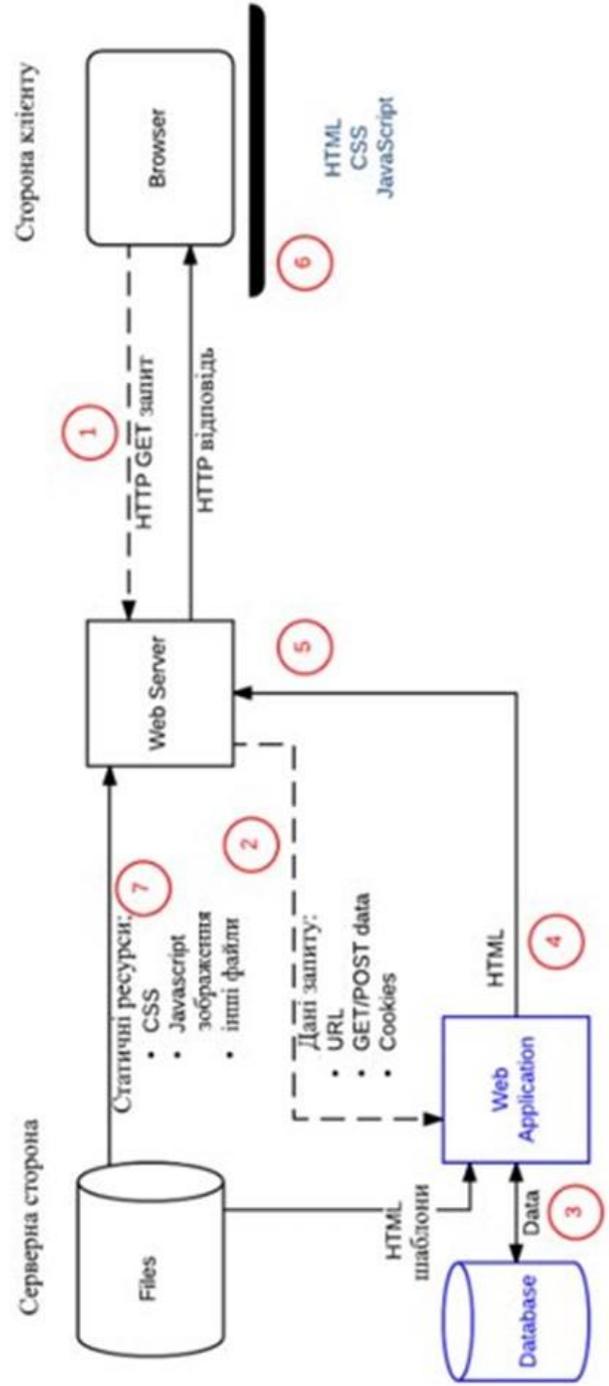
- Огляд принципу роботи та поняття технології «Розумний будинок».
- Огляд принципу роботи веб-застосунків.
- Огляд наявних компонентів системи «Розумний будинок».
- Визначення із програмними засобами для реалізації проєкту.
- Розрахунок вартості підтримки веб-застосунку.
- Розробка серверної частини веб-застосунку.
- Розробка клієнтської частини веб-застосунку.

Принцип роботи веб-застосунків

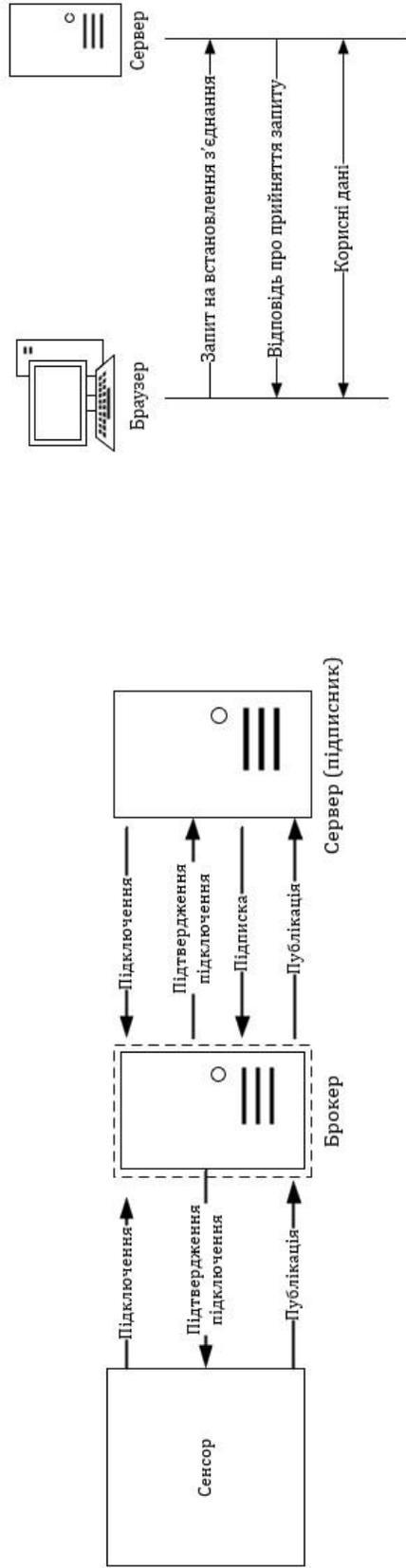


Веб-застосунок – розподілений застосунок, логіка якого зосереджується на віддаленому сервері, а функція відображення інтерфейсу полягає на веб-браузер.

Принцип роботи веб-серверу



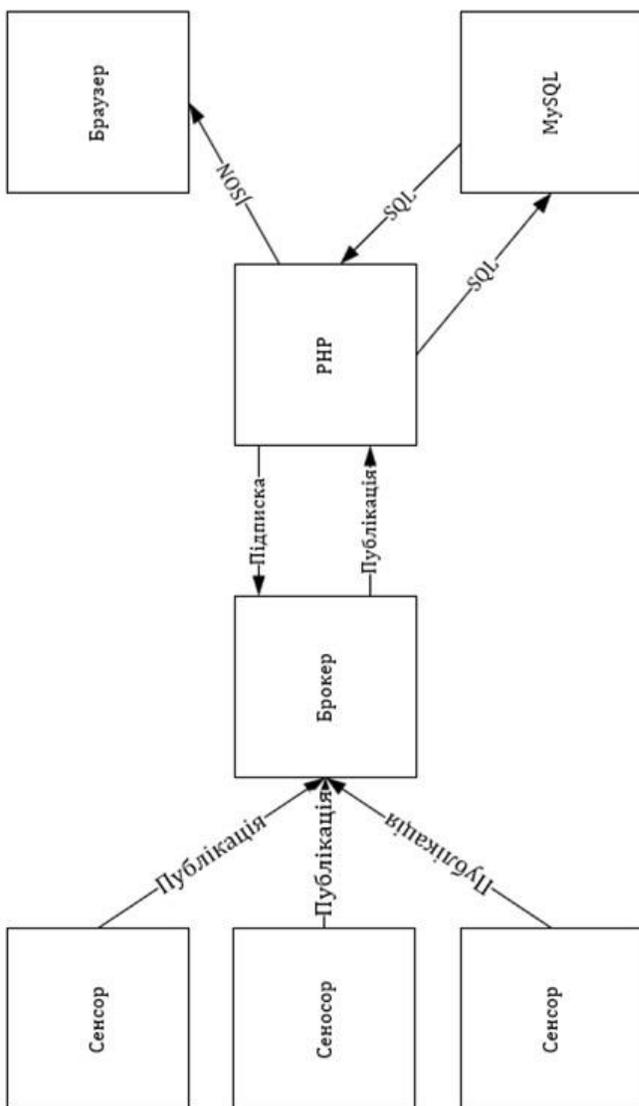
Принцип роботи протоколів MQTT та WebSocket

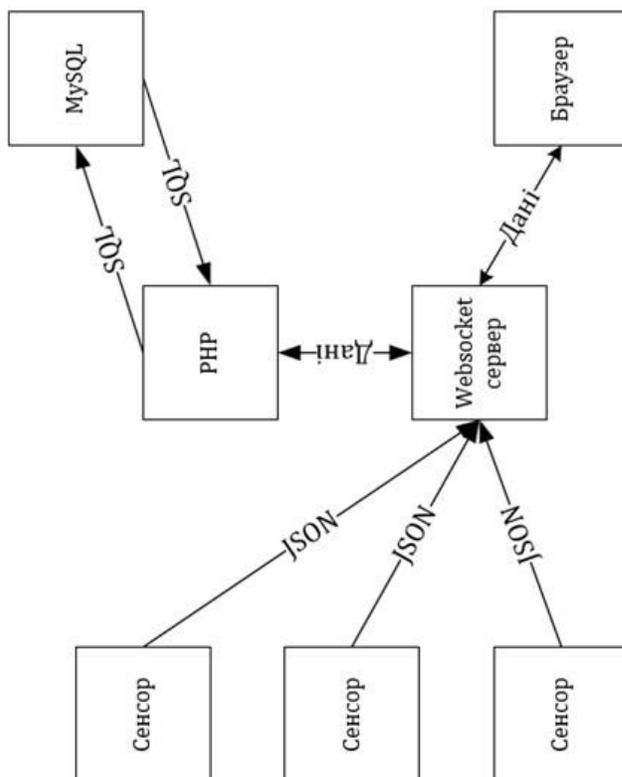


MQTT

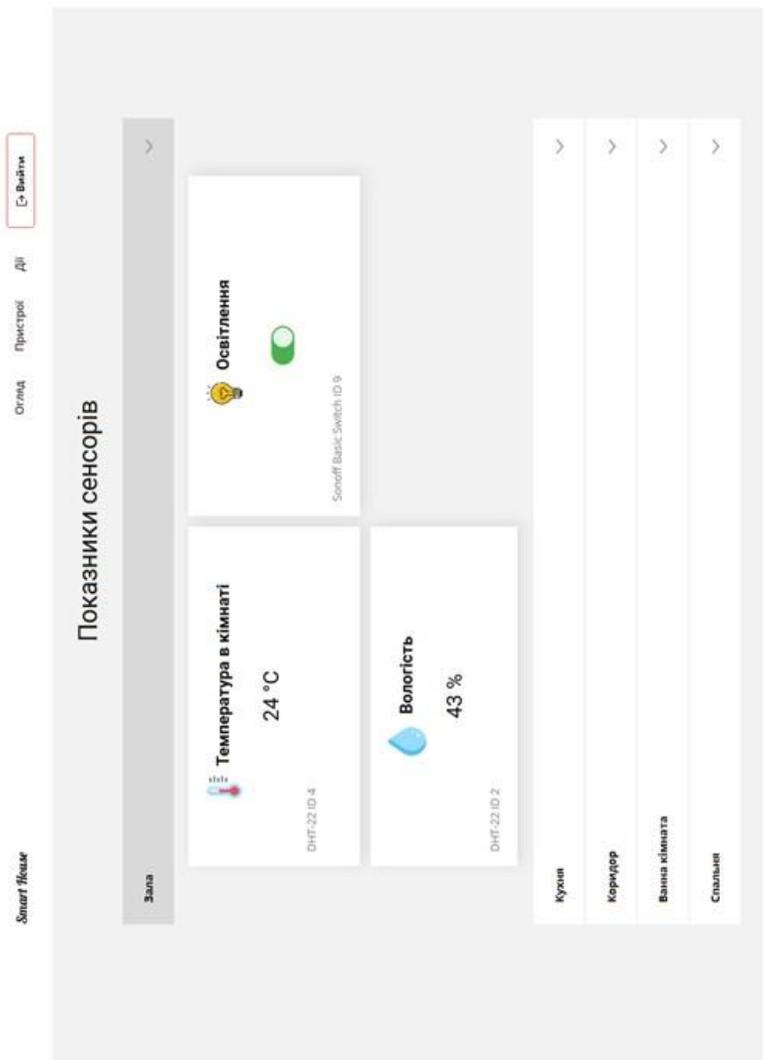
WebSocket

Взаємодія компонентів за протоколом MQTT



Взаємодія компонентів за протоколом WebSocket

Інтерфейс веб-застосунку



Головна сторінка

Інтерфейс веб-застосунку

Smart Home Огляд Пристрої Дії Вийти

Зала **Кухня** **Коридор** **Ванна кімната** **Спальня**

Статус	Модель	Тип	Протокол	Опис	Показник	На головну сторінку
●	Raallax X-band	Сенсор	MQTT	Датчик руху	0	 
●	DHT22	Сенсор	MQTT	Датчик температури	23	 
●	Sonoff Basic Switch	Реле	MQTT	Реле	1	 
●	HC-SR04	Сенсор	Websocket	Датчик відстані	12.1	 <input type="checkbox"/>

Додати кімнату Додати пристрій

Сторінка із наявними пристроями

Інтерфейс веб-застосунку

Smart Home Огляд Пристрій Дії [Вийти](#)

Пристрій	Якщо	То	Останнє виконання	
Parallax X-band	value = 1	value = 0, Sonoff Basic Switch id 5	06.12.2022 13:48	<input checked="" type="checkbox"/>  
DHT22	value < 18	value = 1, Sonoff Basic Switch id 3	01.12.2022 10:11	<input checked="" type="checkbox"/>  
Gidrolock WSU	value = 1	value = 0, Sonoff Switch id 19	10.11.2022 16:26	<input type="checkbox"/>  

Сторінка налаштування взаємодії пристроїв

Загальні висновки

Результатом виконання даної кваліфікаційної роботи магістра є розроблений веб-застосунок, який дозволяє візуалізувати дані отримані із системи «Розумний будинок».

У ході виконання роботи було виконано усі поставлені задачі, а саме:

Дослідження принципу роботи веб-застосунків

- Дослідження наявних систем «Розумний будинок»
- Дослідження програмних засобів для розробки веб-застосунків
- Визначення із необхідними програмними засобами для виконання даної роботи
- Реалізація веб-застосунку для візуалізації даних із системи «Розумний будинок»