

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики
(інститут)

Факультет інформаційних технологій
(факультет)

Кафедра Програмного забезпечення комп'ютерних систем
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня
бакалавра

(назва освітньо-кваліфікаційного рівня)

студентки *Бріжак Валерії Геннадіївни*
(ПІБ)

академічної групи *121-19-1*
(шифр)

спеціальності *121 «Інженерія програмного забезпечення»*
(код і назва спеціальності)

освітньої програми *Інженерія програмного забезпечення*
(назва освітньої програми)

на тему: *Розробка веб-інтерфейсу розумного будинку*

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	<i>проф. Лактіонов І.С.</i>	95	відмінно	
розділів:				
спеціальний	<i>проф. Лактіонов І.С.</i>			
економічний	<i>доц. Касьяненко Л.В.</i>			
Рецензент				
Нормоконтролер	<i>ст. викл. Мартиненко А.А.</i>			

Дніпро
2023

Міністерство освіти і науки України
НТУ «Дніпровська політехніка»

ЗАТВЕРДЖЕНО:

завідувач кафедри
програмного забезпечення комп'ютерних
систем

(повна назва)

(підпис)

М.О. Алексєєв
(прізвище, ініціали)

« » 2023 року

ЗАВДАННЯ
на кваліфікаційну роботу
бакалавра
(назва освітньо-кваліфікаційного рівня)

студентки 121-19-1 Бріжак Валерії Геннадіївни
(група) (прізвище та ініціали)
тема кваліфікаційної роботи Розробка веб-інтерфейсу розумного будинку

затверджена наказом ректора НТУ «ДП» від 16.05.2023 № 350-с

Розділ	Зміст виконання	Термін виконання
<i>Спеціальний</i>	<i>На основі матеріалів практик та інших науково-технічних джерел провести аналіз стану рішення проблеми та постановку задачі. Обґрунтувати вибір та здійснити реалізацію методів вирішення проблеми</i>	<i>13.05.2023 р.</i>
<i>Економічний</i>	<i>Провести розрахунок трудомісткості розробки програмного забезпечення, витрат на створення ПЗ й тривалості його розробки</i>	<i>27.05.2023 р.</i>

Завдання видав _____ Проф. Лактіонов І.С.
(підпис) (посада, прізвище, ініціали)
Завдання прийняв до виконання _____ Бріжак В.Г.
(підпис) (прізвище, ініціали)

Дата видачі завдання: 14.01.2023 р.

Термін подання кваліфікаційної роботи до ЕК: 12.06.2023 р.

РЕФЕРАТ

Пояснювальна записка: 81 с., 25 рис., 3 дод., 26 джерел.

Об'єкт розробки: веб-інтерфейс для керування пристроями та системами розумного будинку.

Мета кваліфікаційної роботи: підвищення ефективності та ергономічності процесів моніторингу й керування технічними системами розумних будинків за рахунок створення ефективного та ергономічного UX веб-інтерфейсу та можливості керування групами пристроїв.

У вступі розглядається сучасний стан проблеми та її аналіз, визначається мета кваліфікаційної роботи, а також галузь, у якій вона може бути застосована, наведено обґрунтування актуальності теми та конкретизується постановка завдання.

У першому розділі проведено аналіз предметної галузі, сформульовано актуальність завдання та призначення розробки, розкрито постановку завдання, зазначено вимоги щодо програмної реалізації, програмних засобів та технологій

У другому розділі проведено дослідження та порівняння наявних рішень, обґрунтовано вибір платформи для розробки, виконано проектування і розробка програмного забезпечення, описана робота програми, її структура та алгоритми функціонування, а також виклик та завантаження програми, вказані вхідні і вихідні дані, та описано склад параметрів технічних засобів.

В економічному розділі було встановлено трудомісткість розроблення програмного забезпечення, проведено розрахунок вартості робіт зі створення програми та визначено необхідний час для її розробки.

Практичне завдання полягає у проведенні аналізу наявних UX-рішень для веб-інтерфейсів керування розумним будинком, створення дизайну та прототипів, спираючись на дослідження, а також створення веб-інтерфейсу за дизайном.

Список ключових слів: РОЗУМНИЙ БУДИНОК, ДОСВІД КОРИСТУВАЧА, ВЕБ-ІНТЕРФЕЙС, СКРИПТ, БАЗА ДАНИХ, HTML, CSS, PHP, JS

ABSTRACT

Object of development: a web interface for managing smart home devices and systems.

The purpose of the qualification work: increasing the efficiency and ergonomics of the processes of monitoring and managing the technical systems of smart houses due to the creation of an effective and ergonomic UX web interface and the ability to manage groups of devices.

The introduction examines the current state of the problem and its analysis, defines the purpose of the qualification work, as well as the field in which it can be applied, provides the rationale for the relevance of the topic, and specifies the assignment.

In the first section, an analysis of the subject area was carried out, the relevance of the task and the purpose of the development were illustrated, the setting of the task was disclosed, the requirements for software implementation, software tools and technologies were indicated

In the second section, a study and comparison of available solutions was carried out, the choice of a development platform was justified, software design and development was carried out, the operation of the program, its structure and functioning algorithms, as well as calling and loading of the program, input and output data were indicated, and the composition of parameters was described. technical means.

In the economic section, the labor intensity of software development was determined, the cost of work on creating the program was calculated, and the time required for its development was determined.

The practical task consists of analyzing existing UX solutions for smart home control web interfaces, designing and prototyping based on research, and creating a web interface by design.

Keyword List: SMART HOME, USER EXPERIENCE, WEB INTERFACE, SCRIPT, DATABASE, HTML, CSS, PHP, JS.

ЗМІСТ

РЕФЕРАТ	3
ABSTRACT	4
ЗМІСТ	5
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	7
ВСТУП	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ	10
1.1. Загальні відомості з предметної галузі	10
1.2 Призначення розробки та галузь застосування	18
1.3. Підстава для розробки	19
1.4. Постановка завдання.....	19
1.5. Вимоги до програми або програмного виробу	20
1.5.1. Вимоги до функціональних характеристик.....	20
1.5.3. Вимоги до складу та параметрів технічних засобів	21
1.5.4. Вимоги до інформаційної та програмної сумісності.....	21
РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ	22
2.1 Функціональне призначення програми	22
2.2 Опис застосованих математичних методів.....	22
2.3 Опис використаної архітектури та шаблонів проектування.....	23
2.4 Опис використаних технологій та мов програмування	23
2.5 Опис структури програми та алгоритмів її функціонування	27
2.6 Обґрунтування та організація вхідних та вихідних даних програми.....	31
2.7 Опис розробленого програмного продукту.....	32
2.7.1 Використані технічні засоби.....	32
2.7.2 Використані програмні засоби	32
2.7.3 Виклик та завантаження програми.....	33
2.7.4 Опис інтерфейсу користувача	34
РОЗДІЛ 3. ЕКОНОМІЧНИЙ РОЗДІЛ.....	47
3.1 Розрахунок трудомісткості та вартості розробки програмного продукту	47

3.2. Розрахунок витрат на створення програми.....	51
ВИСНОВКИ.....	54
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	55
Додаток А. Лістинг програми.....	58
Додаток Б. Відгук керівника економічного розділу.....	82
Додаток В. Перелік документів на оптичному носії.....	83

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД – база даних

СУБД – система управління базами даних

FK – Foreign Key

DOM – Document Object Model

HTML – Hypertext Markup Language

CSS – Cascading Style Sheets

JS – Java Script

PHP – Hypertext Preprocessor

UX – User Experience

IT – інформаційні технології

ПК – персональний комп'ютер

ВСТУП

Актуальність дослідження та інтерфейсу полягає у покращенні користувацького досвіду, оскільки дизайн та функціонал існуючих систем керування розумним будинком потребує додаткового розвитку.

Завдання даної кваліфікаційної роботи та об'єкт її діяльності пов'язані з напрямом підготовки «Web-технології та Web-дизайн», та відповідає загальній тематиці кваліфікаційних робіт і переліку виробничих функцій, типових задач діяльності, умінь та компетенцій, необхідних для бакалаврів освітньої програми 121 "Інженерія програмного забезпечення".

Тематика даної кваліфікаційної роботи присвячена аналізу та створенню веб-інтерфейсу для керування розумним будинком.

Метою кваліфікаційної роботи є підвищення ефективності та ергономічності процесів моніторингу й керування технічними системами розумних будинків за рахунок створення ефективного та ергономічного UX веб-інтерфейсу.

Задачі кваліфікаційної роботи полягають у створенні дизайну та прототипу інтерфейсу, опираючись на результати дослідження, створення веб-інтерфейсу за допомогою мови розмітки HTML, мови стилів CSS, та мови програмування Java Script, підключення інтерфейсу до локального серверу XAMPP та обмін даними за допомогою скриптової мови програмування PHP.

Останні роки технологія розумного будинку стає все популярнішою. Ринок пристроїв збільшується з кожним днем, і пропорційно збільшується попит на новітні технології з боку власників квартир та домів, інтеграторів та спеціалістів з будівництва. Прогнозується, що до 2027 року кількість активних користувачів розумних пристроїв у світі буде становити близько 672 млн. [1]

Для того, щоб контролювати окремі пристрої або переглядати стан показників у розумному будинку, власникам таких пристроїв потрібно робити це через спеціально призначені для цього додатки.

В наш час існує багато рішень для керування розумними будинками. В основному це додатки для планшетів та телефонів.

Дуже розповсюджена проблема таких додатків - ускладнений та незрозумілий інтерфейс для звичайного користувача, а також застарілий та неактуальний дизайн. Такий інтерфейс може спровокувати негативний досвід, що в подальшому має вірогідність вплинути на враження від користування розумним будинком в цілому.

На світовому ринку та ринку України на сьогоднішній час використовуються багато варіантів додатків для керування розумним домом та розумними пристроями. Найпопулярніші представники цієї сфери у світі, у яких є свої додатки для керування розумним будинком: Amazon, Samsung, Google, Apple.

Результатом даної роботи є продуманий веб-інтерфейс, який попередньо пройшов етапи досліджень та детального проектування дизайну, і який у подальшому можна буде використати для реального програмного забезпечення для керування розумним будинком.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1. Загальні відомості з предметної галузі

Галузь розумних будинків та пристроїв є дуже популярною у сьогоденні, та кожен рік набирає обертів. За даними компанії Statista [1], кількість розумних будинків у всьому світі на 2022 рік сягає 302 млн. , а у 2025 ця кількість збільшиться до 478 млн. Тому актуальним питанням стає розробка програмного забезпечення для керування розумними будинками.

На сьогоднішній день на світовому ринку існує багато рішень стосовно керування розумними будинками та окремими смарт-девайсами. Зазвичай такі додатки розробляються для мобільних пристроїв, але є багато прикладів створення додатків такого типу для планшетів та персональних комп'ютерів (Desktop).

Керування розумним будинком стає все більш популярним і доступним для широкої аудиторії завдяки розвитку технологій та появі нових компаній, що займаються розробкою додатків для розумного будинку. Наразі одними з найкращих рішень для керування розумним будинком вважаються [2]: Amazon Alexa App, Samsung SmartThings, Google Home, Apple Home Kit App. Ці додатки відрізняються зовнішнім виглядом інтерфейсу, а також функціоналом. В основному ці додатки є самостійними, але також є рішення, які можуть взаємодіяти з голосовими помічниками, такими як Amazon Echo (працює з додатком Amazon Alexa App), Google Home (додаток Google Assistant App), Apple HomePod (додаток Apple Home Kit App).

Попереднє продумування та створення дизайну інтерфейсу для додатків такого типу має вирішальне значення для зручності використання розумного будинку [3,4]. Користувачі повинні мати змогу легко знаходити потрібні функції та керувати пристроями в домі. Це важливо не тільки з боку візуальної естетики та зручності, але й з боку безпеки. Користувачі повинні максимально

швидко отримувати повідомлення та знаходити у додатку інформацію про несправності та небезпечні ситуації у будинку (наприклад, занадто низькі/високі показники температури та вологи, протікання води або часткове затоплення, несанкціоноване відкриття дверей або вікон).

Було розглянуто деякі рішення дизайну інтерфейсу у додатку Alexa App від виробника Amazon (див. рис. 1.1). На головному екрані додатку, у верхній частині, були розміщені категорії девайсів, що дозволяє користувачеві швидко орієнтуватись у доступних опціях. Проте, що є особливо цікавим, нижче даної категорії були представлені групи пристроїв, зібрані по кімнатах. Це дозволяло користувачу здійснювати швидке керування девайсами у певній кімнаті без необхідності переходити до окремих смарт-пристроїв.

Особливо вдалим рішенням з точки зору користувацького досвіду було використання ілюстрацій на фоні груп з кімнатами. Це прибирає потребу читати назви кімнат, щоб зрозуміти, до якої групи девайсів вони відносяться. Вистачало лише поглянути на ілюстрацію, щоб миттєво розпізнати категорію. Таке рішення значно збільшувало швидкість пошуку потрібної кімнати у додатку.

Додатково, у правому верхньому куті екрану була розміщена кнопка, за допомогою якої користувач міг швидко додавати нові девайси. Це забезпечувало зручний доступ до функції додавання пристроїв і економило час користувача, який б не витрачав його на пошук відповідної опції у меню або на інші складні процедури.



Рис. 1.1. Інтерфейс програми Amazon Alexa App

Наступна програма Google Home (див. рис. 1.2) має схожі рішення на головному екрані, у розділі «Favorites». Аналогічно до попереднього прикладу, у верхній частині екрана розташовані категорії девайсів, які дозволяють користувачеві швидко зорієнтуватися у доступних опціях. Додатково, на цій сторінці присутнє швидке керування улюбленими функціями, такими як керування освітленням, пилососом або жалюзі, що дозволяє користувачу швидко взаємодіяти з часто використовуваними пристроями.

На сторінці Devices можна побачити, що всі пристрої розбиті по кімнатах, що полегшує організацію та навігацію користувача. З цієї сторінки користувач має можливість перейти до окремих пристроїв для їх подальшого налаштування. Це забезпечує зручний доступ до детальної інформації та налаштувань для кожного пристрою окремо. та налаштувань.

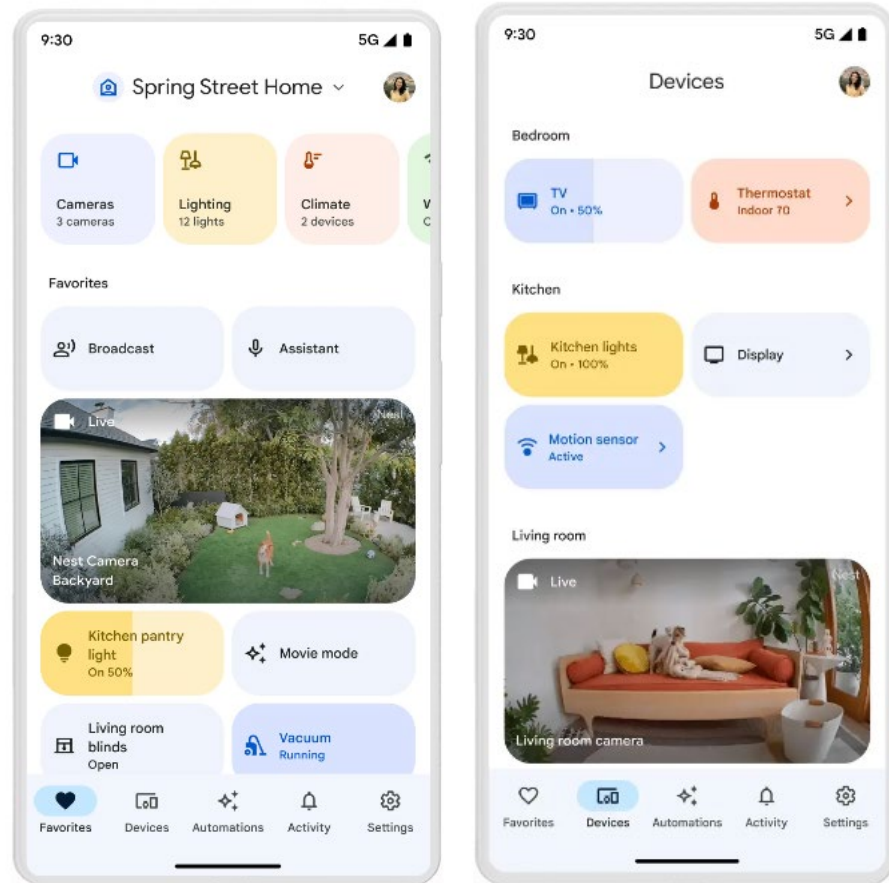


Рис. 1.2. Інтерфейс застосунку Google Home (сторінки Favorites та Devices)

Також, як приклад повнорозмірного інтерфейсу, можна розглянути головну сторінку Google Nest Hub – дисплей від компанії Google для керування розумним будинком (див. рис. 1.3).

На перший погляд, видно, що розділи для керування знаходяться зліва: тут знаходяться розділи для керування освітленням, медіапрогравачем та перехід на сторінку з кімнатами. Це дозволяє зосередитися на головній частині екрана, де знаходяться безпосередньо функції керування.

Таке розташування елементів інтерфейсу сприяє зручності взаємодії з пристроєм, оскільки користувачеві не потрібно довго шукати необхідні функції. Вони розташовані зручно і доступні з одного екрана, що дозволяє швидко переходити між різними функціями керування без зайвих зусиль.

Таким чином, засновані на принципах розташування та вигляду елементів, розглянуті приклади інтерфейсу надають користувачам зручну

можливість керувати своїм розумним будинком, забезпечуючи легку навігацію та швидкий доступ до необхідних функцій.

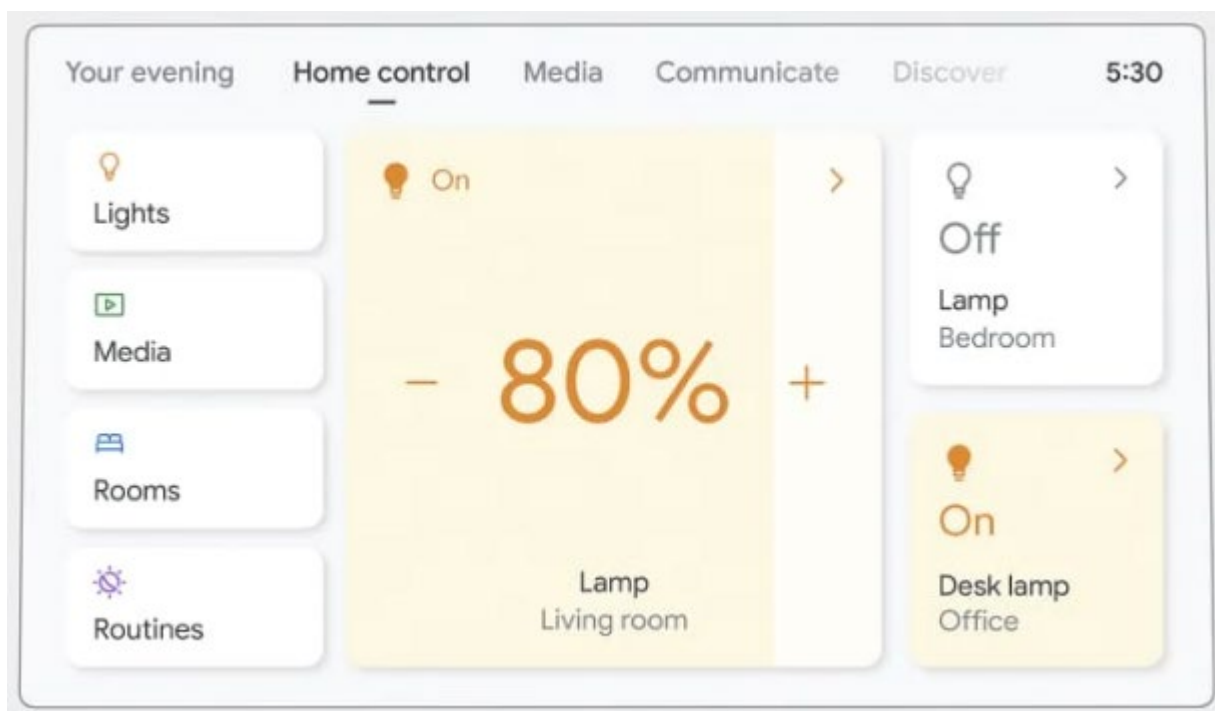


Рис. 1.3. Інтерфейс дисплею Google Nest Hub

При розгляді інтерфейсу застосунку Samsung SmartThings (див. рис. 1.4) можна помітити, що головна сторінка Favorites має свої особливості, які відрізняють її від попередніх прикладів. У верхній частині розміщені важливі повідомлення, такі як сповіщення про відкрите вікно у кімнаті або тривалість роботи пральної машини. Це надає користувачу інформацію про актуальні події в його розумному будинку.

Нижче розташовані популярні пристрої, якими можна швидко керувати прямо з головної сторінки. Це дозволяє зручно та швидко здійснювати контроль над найважливішими функціями розумного будинку без необхідності переходити на інші сторінки.

При переході на сторінку керування кондиціонером, основна функція кондиціонера відображається в блакитному блоці, виділяючи її серед інших налаштувань. Усі необхідні параметри розташовані відповідно до категорій, що спрощує їхнє знаходження та налаштування.

Такий підхід до дизайну інтерфейсу дозволяє забезпечити зручну та легку навігацію для користувачів, використовуючи категорії та виділення основних функцій. Всі необхідні налаштування знаходяться на відповідних сторінках, що дозволяє зосередитися на потрібному функціоналі без зайвих відволікань.

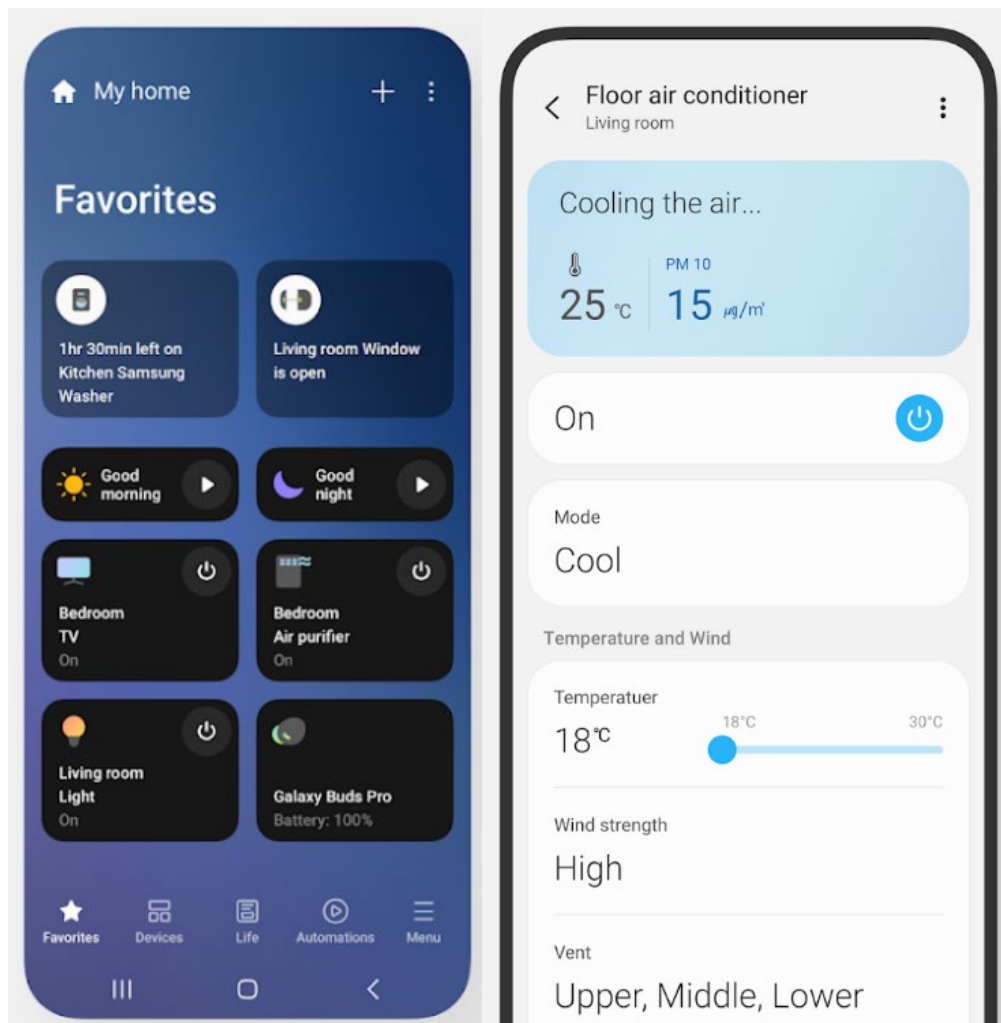


Рис. 1.4. Інтерфейс застосунку Samsung SmartThings (Сторінки Favorites та Керування кондиціонером)

При перегляді версії для персональних комп'ютерів (див. рис. 1.5) можна побачити, що за рахунок збільшеної діагоналі дисплею, панель керування розташована зліва, що забезпечує зручну навігацію та доступ до різних

функцій. Панель керування містить розділи та категорії, що дозволяють швидко знайти потрібні опції та налаштування.

У правій частині розміщені популярні девайси, що забезпечує швидкий доступ до них без необхідності переходити на інші сторінки. Користувач може швидко керувати цими пристроями та здійснювати необхідні налаштування.

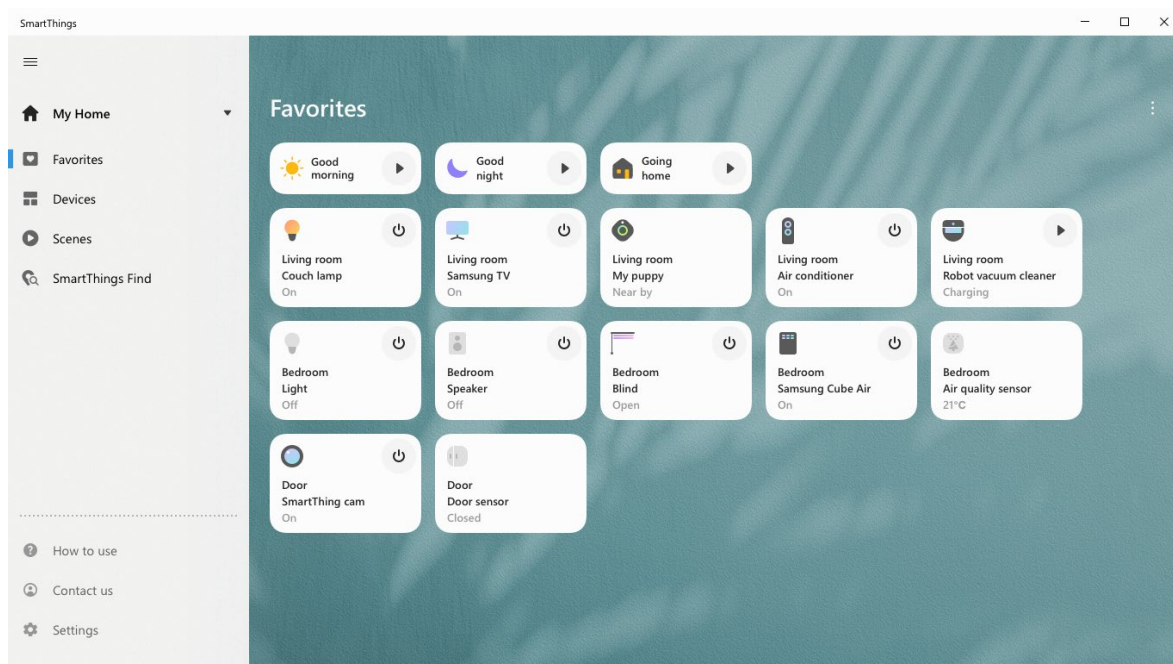


Рис. 1.5. Інтерфейс застосунку Samsung SmartThings для Desktop версії

При розгляді інтерфейсу керування пристроями у кімнаті в застосунку Apple HomeKit (див. рис. 1.6), помітно, що головною категорією є кімнати, а не окремі пристрої, що відрізняє його від попередніх прикладів. У цьому інтерфейсі основний акцент здійснюється на структуруванні та організації пристроїв за кімнатами.

У верхній частині інтерфейсу розташовано значущі рішення - швидке керування основними пристроями, а також важлива інформація, така як температура у кімнаті та стан дверей та вікон. Це надає користувачу миттєвий доступ до ключових функцій та статусу кімнати без необхідності переходити на інші сторінки.

Нижче розташовано повний список пристроїв для обраної кімнати, де користувач може здійснювати детальне керування кожним з пристроїв. Варто відзначити, що фонове зображення кімнати відображає типове оточення для обраної категорії, що створює цікаве та привабливе візуальне середовище на сторінці.

Такий підхід до інтерфейсу керування пристроями у кімнаті в застосунку Apple HomeKit дозволяє ефективно структурувати та керувати пристроями, забезпечуючи зручну та вигідну взаємодію з розумними пристроями в домашньому середовищі.

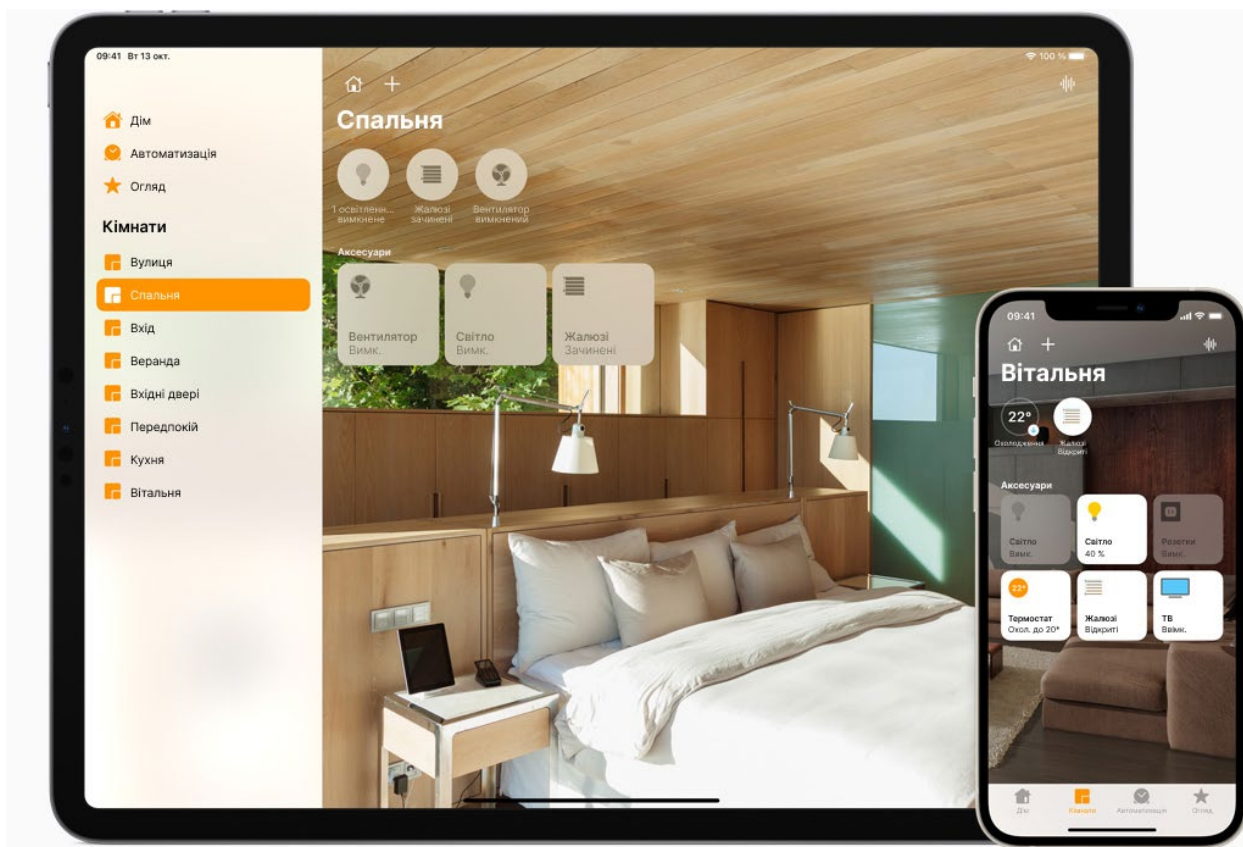


Рис. 1.6. Застосунок Apple Home Kit, Інтерфейс керування пристроями у кімнаті

Так як для аналізу наявних рішень були вибрані додатки від найпопулярніших компаній, можна побачити, що наявні рішення у сфері керування розумним будинком досить добре пропрацьовані, кожен додаток

має свої переваги, цікаві рішення з боку інтерфейсу та мають особливий набір функцій, який призначений для користувачів конкретно цієї програми.

На підставі аналізу було встановлено недоліки у програмних засобах конкурентів, а саме:

– Неактуальність розміщення пункту «Девайси» у навігаційній панелі. Цей пункт навантажує навігаційне меню, бо керування девайсами може відбуватися тільки через пункт меню «кімнати», не використовуючи пункт «Девайси». Рішенням цього недоліку у моїй розробці буде усунення пункту «Девайси» з навігаційної палені. Користувачам зможуть регулювати девайси, знаходячи їх через пункт меню «кімнати». Для швидкого доступу до девайсів, які були використовані нещодавно, буде поміщено блок «Нещодавні девайси» на головну сторінку інтерфейсу.

– Відсутність можливості швидко керувати групами девайсів з головної сторінки застосунку.

В процесі розробки інтерфейсу буде усунено вищезазначені недоліки, а також застосовані кращі практики [5,6] та рішення з інтерфейсів конкурентів, таких як:

– Округлі елементи для покращення сприйняття інформації

– Зображення на фоні у карточках кімнат

– Керування групами девайсів у кімнатах з головної сторінки застосунку.

Таким чином користувачам не буде потреби переходити до сторінки кожного пристрою окремо, тому що буде можливість регулювати важливі показники (температура, вологість) прямо з головної сторінки застосунку.

1.2. Призначення розробки та галузь застосування

Основною метою розробки даного інтерфейсу є поліпшення взаємодії користувачів з подібними застосунками та створення фундаменту для майбутньої програми, яка забезпечуватиме керування девайсами та розумними будинками. Інтерфейс буде розроблено з орієнтацією на надання

зручності, ефективності користувацького досвіду, а також на пришвидшення взаємодії користувача з елементами керування.

З метою створення більш реалістичного та інформативного середовища, інтерфейс буде підключено до локального сервера, що дасть змогу взаємодіяти з базою даних. Це дозволить користувачам додавати, видаляти та налаштовувати пристрої у віртуальному розумному будинку. Така інтеграція з локальним сервером дозволить користувачам максимально відчувати реалізм та ефективність взаємодії з віртуальним розумним будинком. Вони зможуть легко керувати своїми пристроями, налаштовувати їх параметри та спостерігати за їх станом у реальному часі. Будучи підключеними до бази даних, інтерфейс теж забезпечуватиме зручний спосіб додавання нових пристроїв до системи або видалення зайвих.

Основною перевагою такого рішення є його універсальність. Застосунок буде розроблено таким чином, щоб можна було легко адаптувати його [7] до різних типів розумних будинків та пристроїв. Це відкриває можливості для подальшого використання інтерфейсу в реальних проектах, де користувачі зможуть налаштовувати й управляти своїми девайсами в реальному житті.

1.3. Підстава для розробки

В кінці навчання, студент виконує кваліфікаційну роботу (проект). Тема роботи узгоджується з керівником проекту, випускаючою кафедрою.

Підставою для розробки кваліфікаційної роботи на тему «Розробка веб-інтерфейсу розумного будинку» є наказ по Національному технічному університету «Дніпровська політехніка» 350-с від 16.05.2023.

1.4. Постановка завдання

Метою кваліфікаційної роботи є розробка веб-інтерфейсу [8,9], який дозволить користувачам швидко та ефективно керувати своїм розумним

будинком [10]. Основний акцент робиться на комфортності та швидкості користування, щоб забезпечити приємний користувацький досвід.

Створений інтерфейс спрощує процес налаштування та моніторингу пристроїв, що присутні в розумному будинку. Для досягнення цієї мети, інтерфейс пов'язується з сервером, що дозволяє взаємодіяти з базою даних. Це важливий аспект, оскільки дозволяє користувачам легко налаштовувати, додавати та видаляти пристрої у своєму розумному будинку.

Основна ідея полягає в тому, щоб забезпечити зручну та просту взаємодію з розумним будинком, де користувачі можуть з легкістю контролювати свої пристрої та переглядати їх стан. Це допоможе зекономити час та зусилля, які зазвичай пов'язані з керуванням розумними пристроями.

1.5. Вимоги до програми або програмного виробу

1.5.1. Вимоги до функціональних характеристик

Для роботи з інтерфейсом потрібно запуснути його через кросплатформну збірку серверу XAMPP [11]. Далі авторизуватися або зареєструвати нового користувача.

Процес роботи з інтерфейсом здійснюється за допомогою натискання кнопок.

Потрібно реалізувати можливість збереження налаштувань у акаунті користувача, щоб при подальшому відкритті програми не потрібно було знов змінювати параметри.

1.5.2. Вимоги до інформаційної безпеки

Вимог до інформаційної безпеки не ставиться, оскільки інтерфейс працює тільки як прототип, і викладення на хостинг не планується. Але при втіленні інтерфейсу у вигляді програмного забезпечення для користувачів,

потрібно передбачити можливість шифрування пароллю та двоетапної автентифікації щоб доступ до розумного будинку не отримали шахраї.

1.5.3. Вимоги до складу та параметрів технічних засобів

Для функціонування веб-інтерфейсу необхідні:

- Персональний комп'ютер або планшет чи мобільний пристрій
- Клавіатура
- Миша або тачпад

Вимог до характеристик комп'ютера не висувається, так як веб-інтерфейс буде працювати на будь-якому комп'ютері, на якому встановлено інтернет-браузер.

1.5.4. Вимоги до інформаційної та програмної сумісності

Розроблюваний веб-інтерфейс сумісний зі всіма браузерами, наприклад Google Chrome (починаючи з 29 версії), Microsoft Edge (з 12 версії), Safari (починаючи з 9 версії), Firefox (з 28 версії), Opera (з 12.1 версії).

Інтерфейс не буде викладено на хостингу, тому запуск можливий тільки через програмне забезпечення ХАМРР.

РОЗДІЛ 2.

ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ

2.1 Функціональне призначення програми

Враховуючи недоліки, які можна спостерігати в інтерфейсах існуючих програм, головною метою даного проекту є розробка веб-інтерфейсу з покращеною візуальною ергономікою та зручністю.

Цей веб-інтерфейс буде здатний керувати розумними пристроями, забезпечувати їх активацію та налаштування, а також керування системами розумного будинку, такими як регулювання загальної температури або вологості повітря, а також керування освітленням, вентиляцією та аудіосистемою в окремих кімнатах.

Використання розробленого інтерфейсу покликане спростити керування розумним будинком для потенційних користувачів і прискорити процес керування за допомогою швидкодіючих кнопок.

При запуску інтерфейсу на екрані відобразиться вікно авторизації, де користувач може увійти в систему, якщо у нього вже є обліковий запис, або перейти на сторінку реєстрації, якщо такого облікового запису немає. Після успішної авторизації користувач перейде на сторінку з панеллю приладів та кімнат, де він зможе створювати нові кімнати та пристрої і налаштовувати їх параметри.

2.2 Опис застосованих математичних методів

При розробленні програмного забезпечення для кваліфікаційної роботи були використані логічні оператори для формулювання умов та прийняття рішень. Зокрема, оператор `||` (АБО) використовувався для перевірки, чи виконується хоча б одна з заданих умов, тоді як оператор `&&` (ТА) дозволяв перевірити, чи виконуються всі задані умови одночасно. Для зміни логічного

стану умови часто використовувався оператор ! (НЕ), який дозволяв інвертувати її значення. Ці логічні оператори використовувались при написанні умовних конструкцій, таких як «if» та «else if», де рішення про подальші дії програми залежить від заданих умов.

2.3 Опис використаної архітектури та шаблонів проектування

В процесі розробки даної дипломної роботи була використана клієнт-серверна архітектура з веб-орієнтованим підходом.

Клієнтська частина веб-інтерфейсу розумного будинку реалізована з використанням мови розмітки HTML [12], мови стилів CSS [13] та мови JavaScript [14], які виконуються на боці клієнта (веб-браузері).

Серверна частина реалізована з використанням мови програмування PHP [15], яка забезпечує зв'язок з сервером та базою даних. Взаємодія з базою даних відбувається за допомогою файлів-обробників (handlers).

Для досягнення кращої організації та повторного використання коду були написані універсальні функції, до яких можна звернутися з будь-якого місця у коді, підставивши потрібні параметри. Таким чином, це мінімізує потребу кожен раз прописувати у коді звернення до бази даних, бо цей процес обробляється функціями, які є блоками коду багаторазового використання.

2.4 Опис використаних технологій та мов програмування

Програмне забезпечення для кваліфікаційної роботи було розроблено за допомогою мови розмітки HTML, мови стилів CSS, мови програмування JavaScript, а також скриптової мови PHP, у середовищі розробки Microsoft Visual Studio Code 2022. Запуск веб-інтерфейсу відбувається на базі локального веб-серверу, який розгортається у програмному забезпеченні XAMPP. Адміністрування СУБД MySQL [16] відбувалося у веб-застосунку PhpMyAdmin [17].

Так як розроблюваний інтерфейс є середнім за величиною проектом, для роботи з Back-end було обрано мову програмування PHP. Вона є досить простою за синтаксисом і широко вживаною, та ідеально підходить для втілення цілей даного проекту. За допомогою PHP запитів відбувається взаємодія інтерфейсу з сервером, на якому зберігається БД. Також ще одна можливість мови PHP, яка використана при розробці інтерфейсу – збереження даних у сесії та cookie [18] браузеру.

В якості інструменту для роботи з базою даних використано MySQL – реляційна СУБД. Дані види систем управління базами даних є не дуже ефективними при роботі з великим об'ємом даних, але при цьому реляційні СУБД підходять для обробки великої кількості складних запитів, а також забезпечують зв'язок між таблицями та посилюють цілісність, що є вирішально важливим для швидкого та безпроблемного функціонування даного проекту.

Для створення розмітки сторінок веб-інтерфейсу було використано мову розмітки HTML. Головними критеріями вибору даної технології була кросплатформність – вона широко підтримується практично всіма браузерами та платформами, а також можливість розділення структури та вигляду, що дає можливість окремо контролювати структуру сторінки та її зовнішній вигляд.

Для визначення зовнішнього виду сторінок була обрана мова стилів CSS. Використання даної технології дозволяє розділити стилі від структури веб-сторінок, що полегшує редагування та зміну вигляду елементів на сторінках. Основні переваги даної мови стилів, які неодноразово використовуються при створенні інтерфейсу, є повторне використання стилів, гнучкість вигляду та адаптивний дизайн. Вид сторінок змінюється за допомогою медіа-запитів в залежності від розміру екрану пристрою. Це дає можливість використовувати функціонал інтерфейсу не тільки на ПК, а й на телефонах та планшетах.

Динамічні елементи та анімація були втілені за допомогою мови програмування JavaScript. За допомогою даної мови з'явилася можливість

маніпулювати DOM, а також змінювати інтерфейс в залежності від дій користувача на сторінці.

Мова розмітки HTML і мова стилів CSS є основними інструментами для створення веб-сторінок і керування їх виглядом та форматуванням.

HTML є стандартною мовою розмітки, яка використовується для створення структури та вмісту веб-сторінок. Вона складається з набору тегів, які вказують браузеру, яким чином треба відображати різні елементи сторінки, такі як заголовки, параграфи, зображення, посилання та багато інших. HTML є основною складовою частиною кожної веб-сторінки і визначає її структуру.

CSS є мовою стилів, яка використовується для визначення зовнішнього вигляду і форматування елементів HTML. З допомогою CSS можна контролювати кольори, шрифти, розташування, розміри, відступи та багато інших аспектів візуального представлення сторінки. CSS дозволяє розділити стиль і вигляд сторінки від її структури, що забезпечує більшу гнучкість та легкість керування зовнішнім виглядом інтерфейсу.

Обидві мови мають широку підтримку серед браузерів та є основою для багатьох фреймворків та інструментів розробки веб-інтерфейсів.

JavaScript є високорівневою, інтерпретованою мовою програмування, яка використовується для розробки веб-додатків. Вона має широке застосування і забезпечує можливості для створення динамічних, інтерактивних та високоефективних веб-сайтів і додатків. До важливих властивостей JavaScript можна віднести:

- Синтаксис, схожий на синтаксис інших мов програмування, таких як C, C++ та Java. Вона використовується для визначення поведінки об'єктів, функцій, змінних та керування потоком програми.

- Підтримка об'єктно-орієнтованого підходу до програмування. Є можливість визначати свої класи, об'єкти та методи для організації коду та забезпечення його перевикористання.

- Можливість взаємодіяти з елементами HTML-сторінок і змінювати їх вміст, стиль та розміщення. За допомогою JavaScript можна динамічно

змінювати DOM, реагувати на події користувача та оновлювати вміст сторінки без необхідності перезавантаження.

- JavaScript є однією з основних мов програмування для розробки веб-додатків, оскільки вона підтримується майже всіма сучасними веб-браузерами. Це дозволяє створювати веб-додатки, які працюють на різних платформах та пристроях.

PHP (Hypertext Preprocessor) - це скриптова мова програмування, яка переважно використовується для розробки веб-додатків та динамічних веб-сторінок [19]. PHP є однією з найпопулярніших мов програмування на боці сервера та забезпечує широкі можливості для веб-розробки. За даними Web Technology Surveys, 77% всіх веб-сайтів написані на PHP [20], тож вона є досить широко вживаною та популярною. До переваг мови PHP можна віднести наступне:

- Зосередженість на веб-розробці. PHP надає можливість створювати динамічні веб-сторінки, обробляти форми, керувати сесіями, взаємодіяти з базами даних та інші завдання, необхідні для розробки повноцінних веб-додатків.

- Легкий і зрозумілий синтаксис, що робить цю мову програмування доступним для початківців у веб-розробці. Вона також підтримує багато інструментів, які спрощують процес розробки, такі як велика кількість документації, фреймворки та бібліотеки.

- PHP є однією з найпоширеніших мов програмування на боці сервера, що означає, що вона підтримується багатьма хостинг-провайдерами та сумісна з різними веб-серверами, такими як Apache, Nginx та інші.

- Вбудована підтримка для багатьох систем управління базами даних (СУБД), таких як MySQL, PostgreSQL, SQLite та інші. Це дозволяє розробникам легко взаємодіяти з базами даних та виконувати операції збереження, отримання та маніпулювання даними.

У процесі роботи з базою даних, було використано PhpMyAdmin - безкоштовний інструмент управління базами даних MySQL через веб-

інтерфейс. PhpMyAdmin надає зручну та просту у використанні графічну оболонку, що дозволяє взаємодіяти з базами даних безпосередньо через веб-браузер. Він має широкий функціонал, включаючи можливість створення та видалення баз даних, керування таблицями, вставку, оновлення та видалення даних, а також виконання SQL-запитів. Завдяки PhpMyAdmin я зміг зручно та ефективно працювати з базою даних, виконуючи необхідні операції та забезпечуючи безперебійну роботу мого проекту.

2.5 Опис структури програми та алгоритмів її функціонування

Під час проектування програми, було ретельно вивчено та враховано різні аспекти взаємодії інтерфейсу з сервером, зовнішній вигляд та структуру бази даних разом з усіма таблицями, що вона містить. Це було зроблено з метою забезпечення ефективної та надійної роботи програми, а також для створення комфортного досвіду для потенційних користувачів.

У відношенні взаємодії інтерфейсу з сервером, було розглянуто різні аспекти, такі як передача даних між клієнтом та сервером, обробка запитів та відповідей. Було старанно сплановано та реалізовано взаємодію, щоб забезпечити швидке та безперебійне функціонування програми.

Взаємодія інтерфейсу з сервером була організована таким чином, що спочатку користувач взаємодіє з будь-яким елементом інтерфейсу, далі натиснутий елемент передає параметри до відповідного файлу, який в свою чергу опрацьовує отримані параметри, форматує їх до вигляду, який буде сприйматися сервером, далі формує sql-запит та відправляє його на сервер (див. рис. 2.1).

Схематично, взаємодію інтерфейсу з сервером можна показати наступним чином:

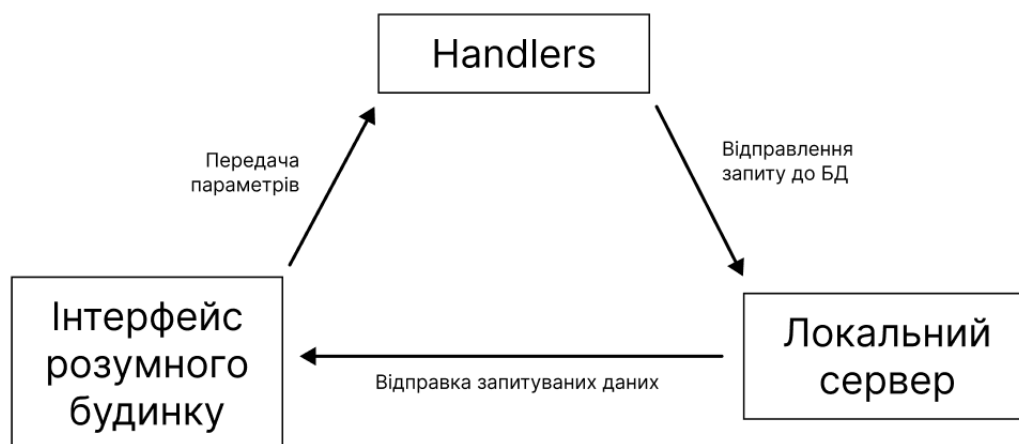


Рис. 2.1. Схема взаємодії інтерфейсу з сервером

Взаємодія відбувається завдяки універсальним функціям, які надсилають запит на сервер, і отримують результат. Під універсальними функціями маються на увазі блоки коду з декількома аргументами, які можна використовувати для будь-якої таблиці та шукати/отримувати будь-які рядки за будь-якими пошуковими параметрами. Нижче наведений приклад даних функцій:

```

function get_smth_by_smth_row(string $selectWhat, string $dbname, string
$whereWhat , string $whatValue){
    global $conn;
    $sql = "select $selectWhat from $dbname where $whereWhat =
'$whatValue'";
    $get_smth = $conn->query($sql);
    $row = mysqli_fetch_assoc($get_smth);
    return $row;
}
  
```

Завдяки даній функції можна обрати будь-який рядок (змінна \$selectWhat) у будь-якій таблиці (\$dbname), де обраний параметр (\$whereWhat) буде дорівнювати введеному значенню (\$whatValue). Таким чином, дану

функцію можна використовувати у різних цілях, для отримання значень з таблиць за параметрами і їх подальшого використання протягом виконання програми.

Одним з найважливіших етапів є проектування структури бази даних (див. рис. 2.2), так як від неї залежить коректність функціонування інтерфейсу. У дану задачу входить продумування та створення таблиць, полів в них та взаємозв'язків між таблицями. Основними таблицями у БД є «users» - містить інформацію про користувачів, їх нікнейми, паролі та дім, до якого вони прив'язані, «homes» - зберігає інформацію про показники в домі, такі як температура, вологість та електроенергія, «rooms» - зберігає інформацію про показники у кімнаті, «devices» - основну інформацію про пристрої в домі. Ці основні таблиці пов'язані між собою зовнішніми ключами (Foreign Key), за допомогою яких легко отримувати дані з будь-якої з цих таблиць.

Створено таблиці, які відповідають за тип девайсу («types»), підтип («undertypes») та виробника («brands»). Ці три таблиці пов'язані з таблицею «devices», та містять у собі список типів, підтипів та виробників девайсів, які зареєстровані у базі даних. За допомогою цих таблиць полегшується процес створення девайсів користувачем, тому що він відразу бачить перед собою список всіх доступних підтипів та виробників пристроїв.

Так як у таблиці «devices» зберігається тільки основна інформація про пристрій, постає питання – де потрібно зберігати детальну інформацію про різні пристрої, бо кожен девайс має свої особливі характеристики (наприклад, у кондиціонера це температура, режим, напрямок жалюзі, а у лічильника - число використаних кіловат). Якщо зберігати дані про всі девайси в одній таблиці, це негативно позначиться на швидкості функціонування БД. Тому було прийнято рішення всі дані про девайси винести в окремі таблиці, та під'єднати їх до таблиці з девайсами за допомогою FK. На структурі БД це таблиці «kettles», «counters», «conditioners» та «thermometers». Оскільки інтерфейс працює лише як прототип, який відображає загальну суть функцій,

на малюнку таких таблиць всього 4, але при втіленні цього інтерфейсу в реальне програмне забезпечення подібних таблиць може бути значно більше.

Важливу роль з точки зору візуальної складової інтерфейсу відіграє таблиця «deviceimages». У даній таблиці зберігаються фотографії всіх пристроїв. Девайсу з певним підтипом та виробником відповідає певне зображення, яке відображається на інтерфейсі.

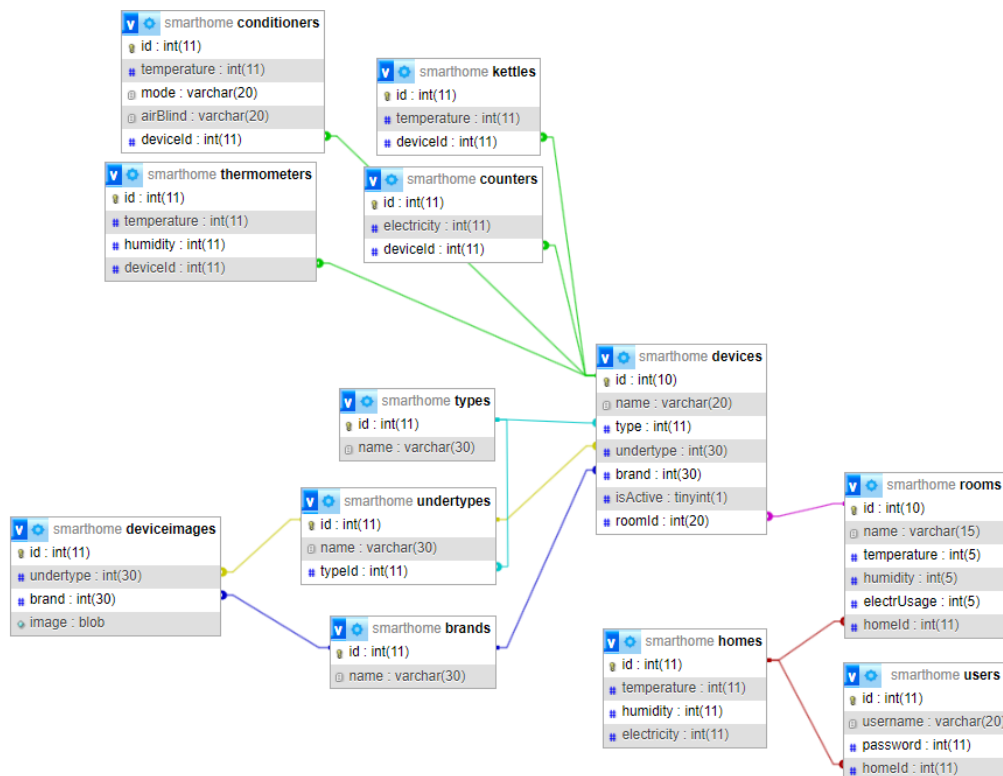


Рис. 2.2. Структура бази даних

Проектування структури інтерфейсу полягало у створенні зручного та зрозумілого маршруту користувача та блок-схеми алгоритму (див. рис. 2.3). Маршрут користувача є чітко визначеним та логічним маршрутом навігації, який допомагає користувачам швидко зорієнтуватися на сайті та легко переходити між різними сторінками. Це зменшує ймовірність, що користувачі загубляться або залишаться збентеженими, що може призвести до втрати потенційних клієнтів чи відвідувачів. Користувачі цінують зручність та ефективність використання веб-сайтів. Якщо вони можуть швидко знайти

Вихідними даними є інформація про пристрої та їх налаштування, які відображаються на інтерфейсі.

Вихідні дані:

- веб-сторінки інтерфейсу керування розумним будинком
- картки девайсів
- налаштування девайсів

Дані записуються у змінні, далі передаються до інтерфейсу у вигляді окремих рядків або їх масивів, і відображаються на сторінках-шаблонах за допомогою ідентифікаційних номерів, які передаються на сторінки у вигляді параметрів.

2.7 Опис розробленого програмного продукту

2.7.1 Використані технічні засоби

Дане програмне забезпечення є веб-інтерфейсом, тому переважно виконується у веб-браузері на локальному сервері, і для його функціонування вистачить мінімальних ресурсів ПК або ноутбуку. Основні технічні засоби для роботи ПЗ:

- Персональний комп'ютер або ноутбук чи мобільний пристрій;
- Будь-який веб-браузер із списку: Google Chrome, Microsoft Edge, Safari, Firefox, Opera
- Стабільне підключення до мережі Internet;
- Клавіатура (для ПК);
- Маніпулятор «миша» або тачпад.

2.7.2 Використані програмні засоби

Для розробки інтерфейсу використовувалися дві основні мови - мова розмітки HTML та мова стилів CSS. Ці мови відповідають за зовнішній вигляд сторінок та всіх елементів, розміщених на них. Крім того, для створення

анімацій та інтерактивних елементів [21], які реагують на дії користувача, була використана мова програмування JavaScript.

Запити до сервера, отримання та виведення інформації, що зберігається у базі даних, були реалізовані за допомогою мови програмування PHP. Ця мова дозволяє взаємодіяти з базою даних та виконувати різноманітні операції з даними [22].

У процесі розробки використовувався редактор коду Visual Studio Code 2022 від компанії Microsoft. Це потужне середовище розробки, яке спеціально призначене для кросплатформного створення веб-додатків. Воно надає широкі можливості для редагування коду та підтримує різні мови програмування та розширення.

Локальний веб-сервер, на якому працює проект, базується на кросплатформній збірці XAMPP. Ця збірка включає в себе веб-сервер Apache, реляційну систему управління базами даних MySQL та інтерпретатор скриптів PHP. Використання XAMPP дозволяє локально розгорнути та тестувати веб-додатки без необхідності підключення до зовнішнього сервера.

2.7.3 Виклик та завантаження програми

Для виклику та завантаження програми необхідно:

- Завантажити та помістити папку з ПЗ у директорію XAMPP
- Запустити додаток XAMPP
- Ввімкнути веб-сервер Apache та СУБД MySQL
- Відкрити браузер
- Ввести у строці для URL стандартне доменне ім'я приватних IP-адрес 127.0.0.1
- Вказати назву папки з ПЗ після доменного імені
- Запустити початкову сторінку Log In

2.7.4 Опис інтерфейсу користувача

Після запуску інтерфейсу відкривається сторінка авторизації (див. рис. 2.4). Якщо користувач вже зареєстрований, він може ввести свій логін та пароль, і натиснути кнопку «Log In»

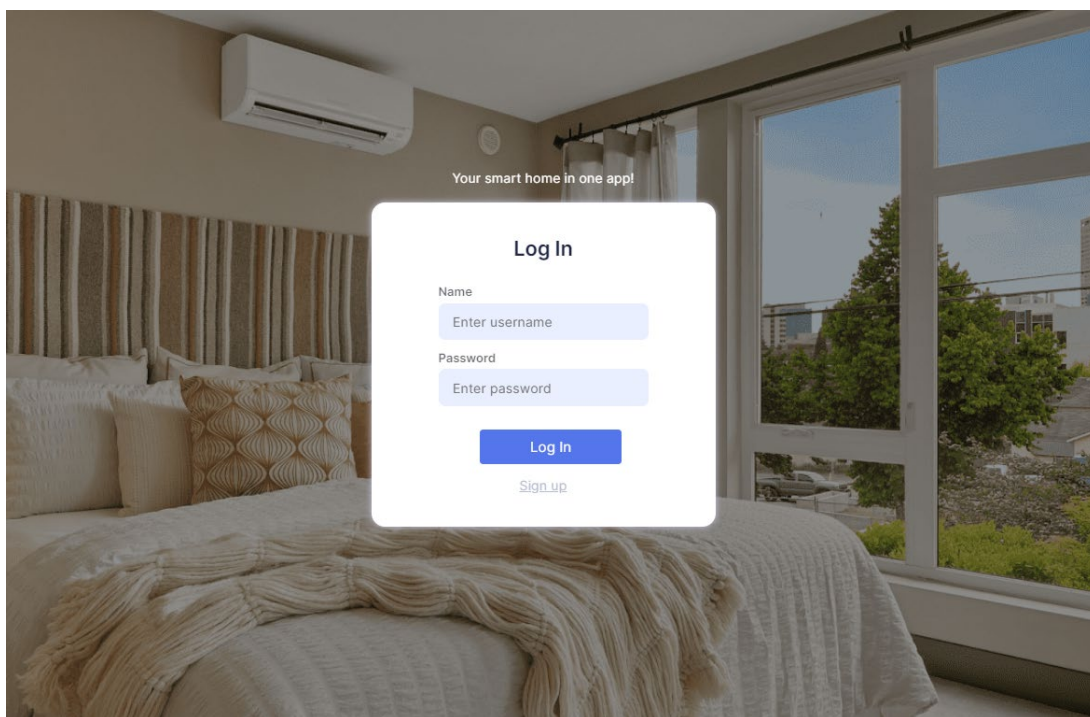


Рис. 2.4. Сторінка авторизації

У формі авторизації присутня валідація даних для забезпечення коректної роботи застосунку. При неправильно введеному логіні або паролі у формі виводиться певний надпис з помилкою, яку допустив користувач (див. рис. 2.5).

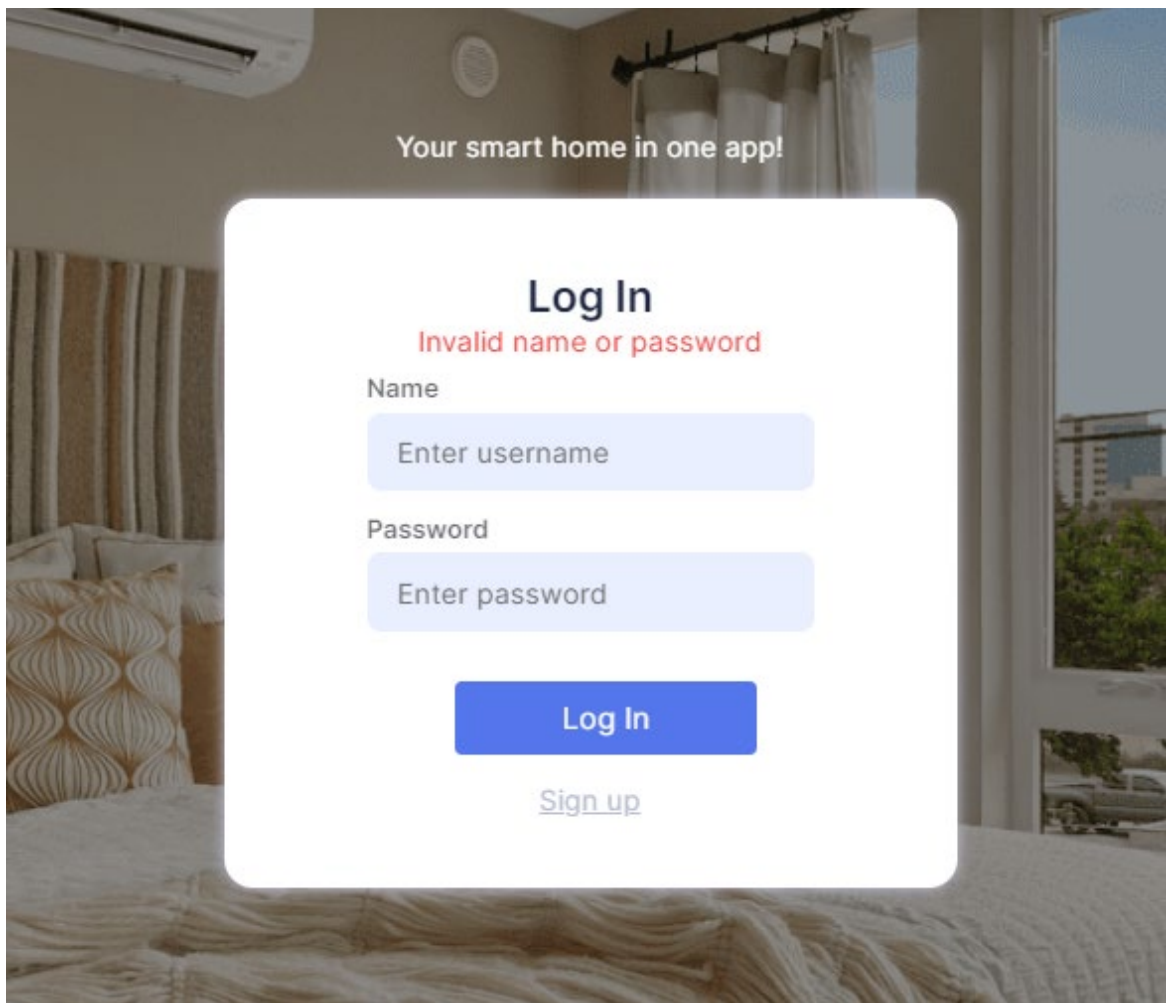


Рис. 2.5. Помилка при введенні неправильного логіну або паролю

У тому випадку, коли користувач не має аккаунту, він може натиснути кнопку «Sign Up». Це перенаправить його на сторінку реєстрації (див. рис. 2.6), де він зможе ввести логін та пароль, та зареєструватись. Якщо ж кнопка «Sign Up» була натиснута помилково, користувач може перейти на попередню сторінку, натиснувши кнопку «Log In».

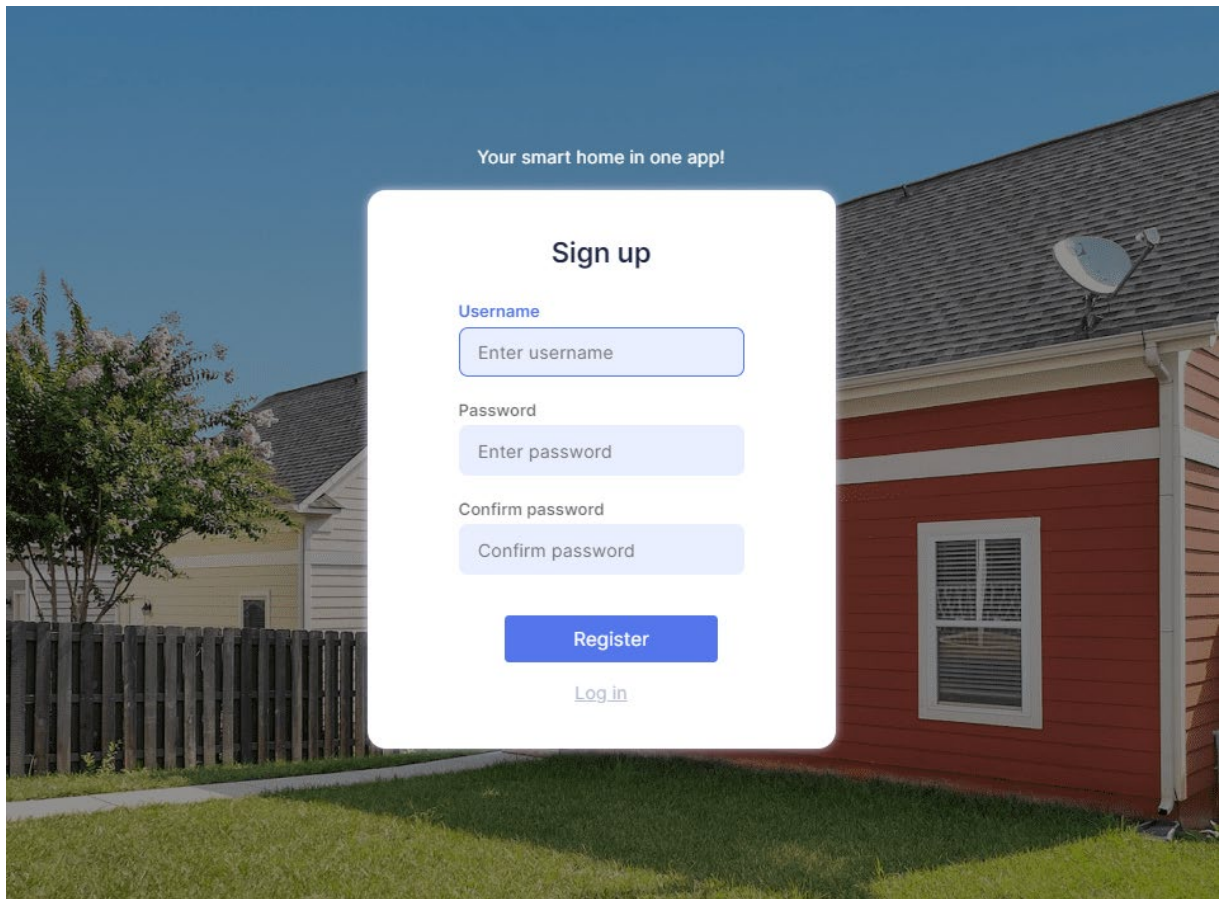


Рис. 2.6. Сторінка реєстрації

На даній сторінці також присутня валідація даних, але в ширшому вигляді, так як помилок при реєстрації можна допустити значно більше. При натисканні кнопки «Register» в першу чергу перевіряється, чи заповнені поля форми (див. рис. 2.7). Після цього йде запит до бази даних, задля перевірки унікальності нікнейму користувача (див. рис. 2.8). Якщо користувача з таким ім'ям не знайдено, перевіряється, чи збігаються паролі (див. рис. 2.9). У випадку, якщо жодна з цих помилок не виявлена, користувач успішно реєструється та перенаправляється на сторінку авторизації.

Sign up
Fields are empty

Username
Enter username

Password
Enter password

Confirm password
Confirm password

Рис. 2.7. Помилка при незаповнених полях

Sign up
Username is already taken

Username
lera

Рис. 2.8. Помилка при існуванні користувача з таким нікнеймом

Sign up
Passwords not matched

Username
lera

Password
....

Confirm password
.....

Рис. 2.9. Помилка при розбіжності паролів

Після того, як користувач успішно авторизується, він буде автоматично перенаправлений на сторінку «Dashboard», яка є центральним пунктом керування розумним будинком (див. рис. 2.10).

Зліва розташована навігаційна панель, яка дозволяє користувачеві переходити між різними сторінками та розділами системи. Також на панелі розташовані кнопки для додавання нових девайсів та кімнат, а також кнопка виходу з облікового запису.

У верхній частині головної секції відображається поточна температура у місті, де знаходиться будинок, а також актуальний час.

Нижче розташований блок «Today's overview», який надає користувачу важливу інформацію про стан будинку на даний день. В цьому блоку відображаються регулятори для налаштування температури та вологості повітря, а також загальна кількість використаної енергії в будинку.

У блоку «Rooms» перераховуються всі кімнати, які є у будинку, разом із відповідною інформацією та швидким керуванням групами девайсів. Для кожної кімнати вказується назва, кількість девайсів, наявність термометра для вимірювання температури, а також кнопки для керування вентиляцією, освітленням та звуком. Натискання на будь-яку кнопку у кімнаті миттєво ввімкне або вимкне всі пристрої відповідної групи. Це дуже зручна функція, яка позбавляє користувачів необхідності переходити до окремих налаштувань кожного пристрою та по черзі їх активувати. Таким чином, користувач може зручно керувати пристроями прямо з головного екрану, наприклад, швидко підвищити або знизити температуру в усій кімнаті або ввімкнути освітлення.

У нижній частині сторінки, завдяки використанню cookies [22], відображаються останні відвідані користувачем пристрої. Це дозволяє швидко знайти та керувати пристроями, які використовуються часто, без необхідності переходу до сторінки конкретної кімнати.

Такий дизайн головної сторінки "Dashboard" забезпечує зручну та ефективну навігацію користувача, а також швидкий доступ до основних функцій та керування розумним будинком. Він спроектований з урахуванням

зрозумілості та зручності використання, дозволяючи користувачам ефективно управляти своїм будинком.

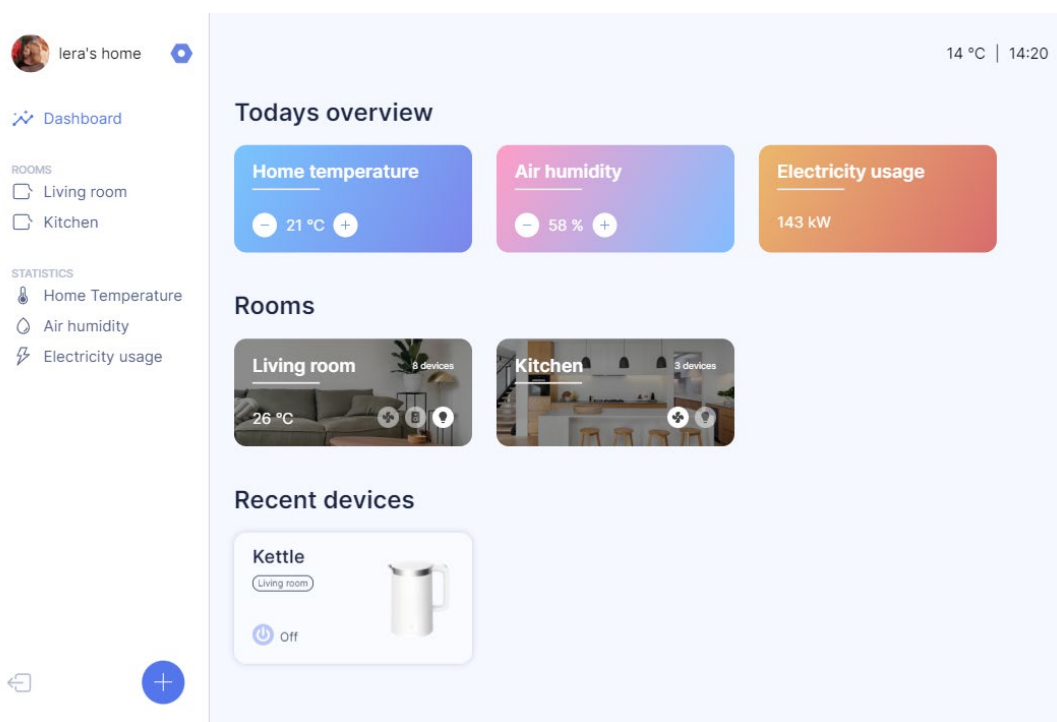


Рис. 2.10. Сторінка «Dashboard»

Завдяки використанню CSS та media-запитів, головна сторінка сайту має адаптивний дизайн, що дозволяє їй пристосовуватись до різних розмірів екрану. Наприклад, при перегляді на мобільному пристрої, відбуваються певні зміни у вигляді і розташуванні елементів.

Коли розмір екрану стає меншим за стандартний екран ноутбуку (див. рис. 2.11), навігаційна панель керування, яка знаходиться зліва, автоматично зникає, щоб не займати зайвого місця на основному інтерфейсі. Це сприяє збереженню простору та покращенню зручності користування.

Для доступу до навігаційної панелі керування, яка раніше знаходилась зліва, використовується спеціальна кнопка меню, розташована у лівому верхньому куті екрану. Користувач може натиснути на цю кнопку, щоб відкрити панель керування та отримати доступ до всіх функцій та сторінок.

Заради оптимізації простору, кнопка додавання девайсів та кімнат переміщується у правий верхній кут екрану. Це дозволяє забезпечити більше місця для важливих елементів та полегшує їх доступність для користувача.

Такий адаптивний підхід до дизайну головної сторінки дозволяє забезпечити зручну та ефективну навігацію навіть на пристроях з меншими екранами. Користувачі можуть легко отримати доступ до всіх необхідних функцій та інтерфейсу незалежно від розміру свого пристрою.

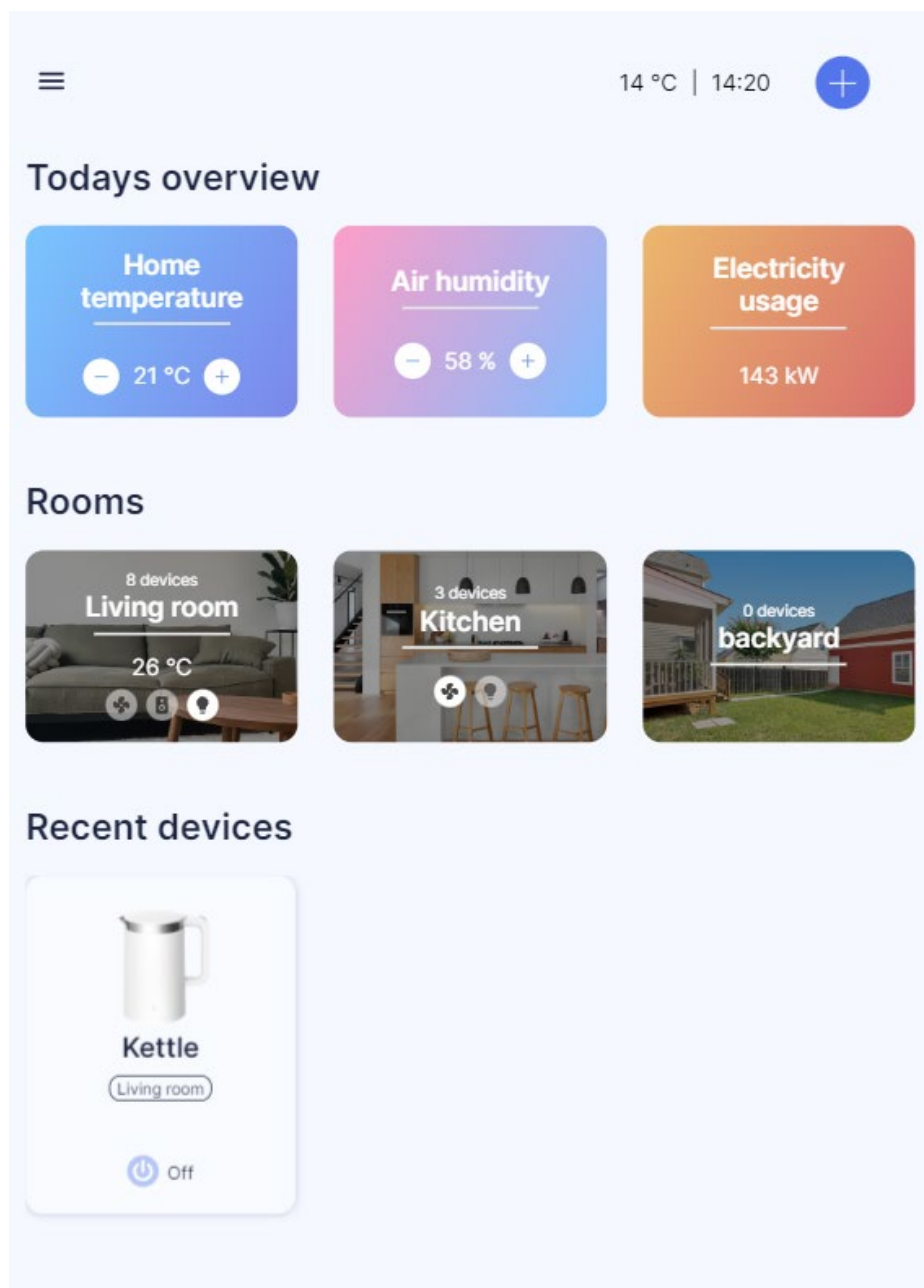


Рис. 2.11. Сторінка «Dashboard», адаптована під планшет

Коли ж екран зменшується до розмірів телефону, сторінка повністю перебудовується, всі блоки розміщуються вертикально для спрощення сприйняття інтерфейсу (див. рис. 2.12). Навігаційну панель також можна відкрити, натиснувши на меню.

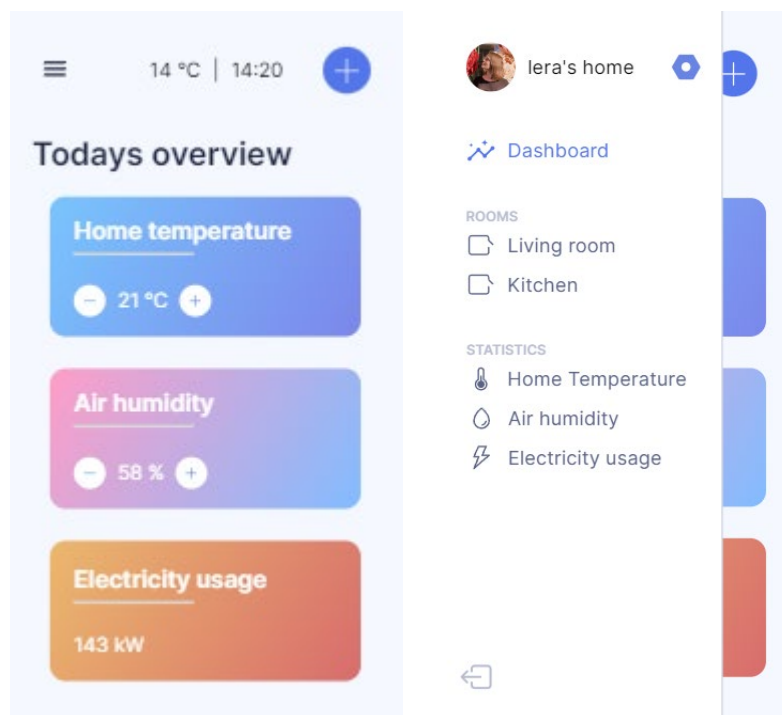


Рис. 2.12. Сторінка «Dashboard», адаптована під мобільний пристрій

З навігаційної панелі можна викликати форму створення девайсів (див. рис. 2.13). Алгоритм використання форми для девайсів простий – користувач вводить назву девайсу (якщо не вводить – тоді замість назви буде виводитись тип девайсу та виробник), і обирає у спадному меню кімнату, тип девайсу та виробника. У формі створення кімнати потрібно ввести тільки назву кімнати.

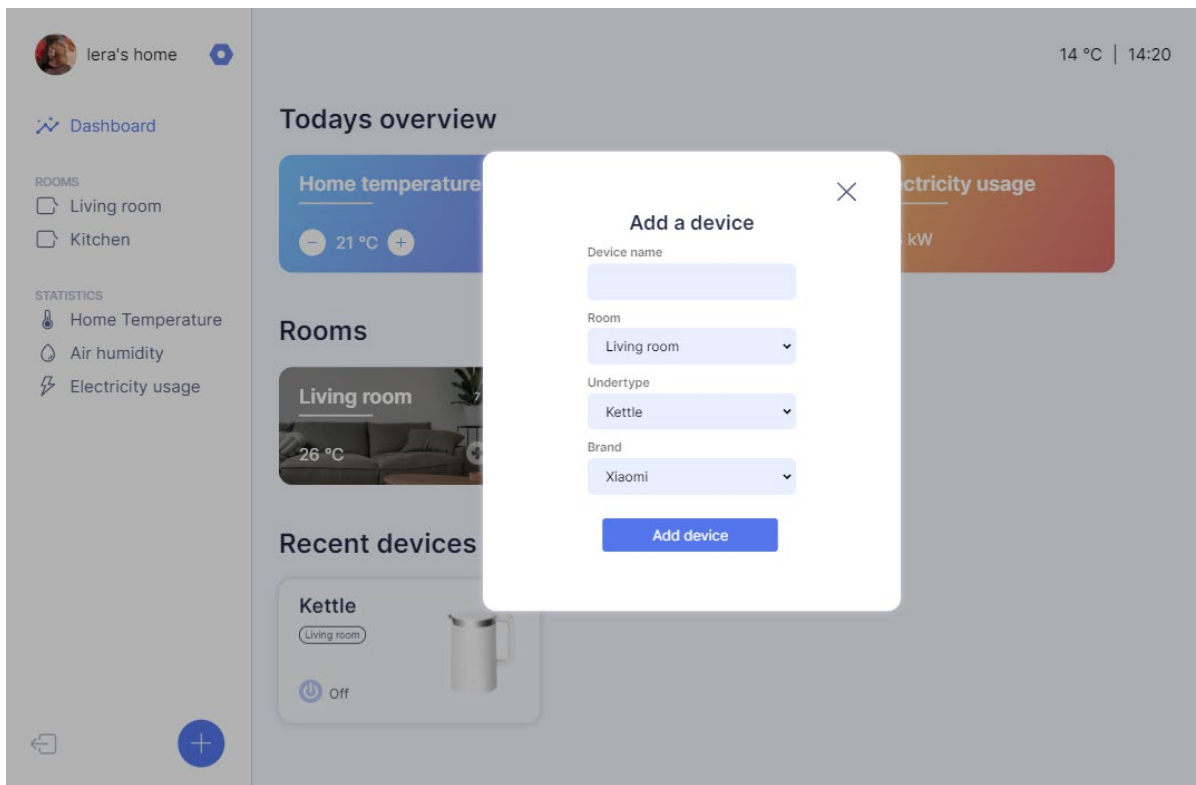


Рис. 2.13. Форма створення девайсу

Схожий функціонал має форма створення кімнат (див. рис. 2.14), але в ній потрібно ввести тільки назву кімнати та натиснути кнопку «Add room».

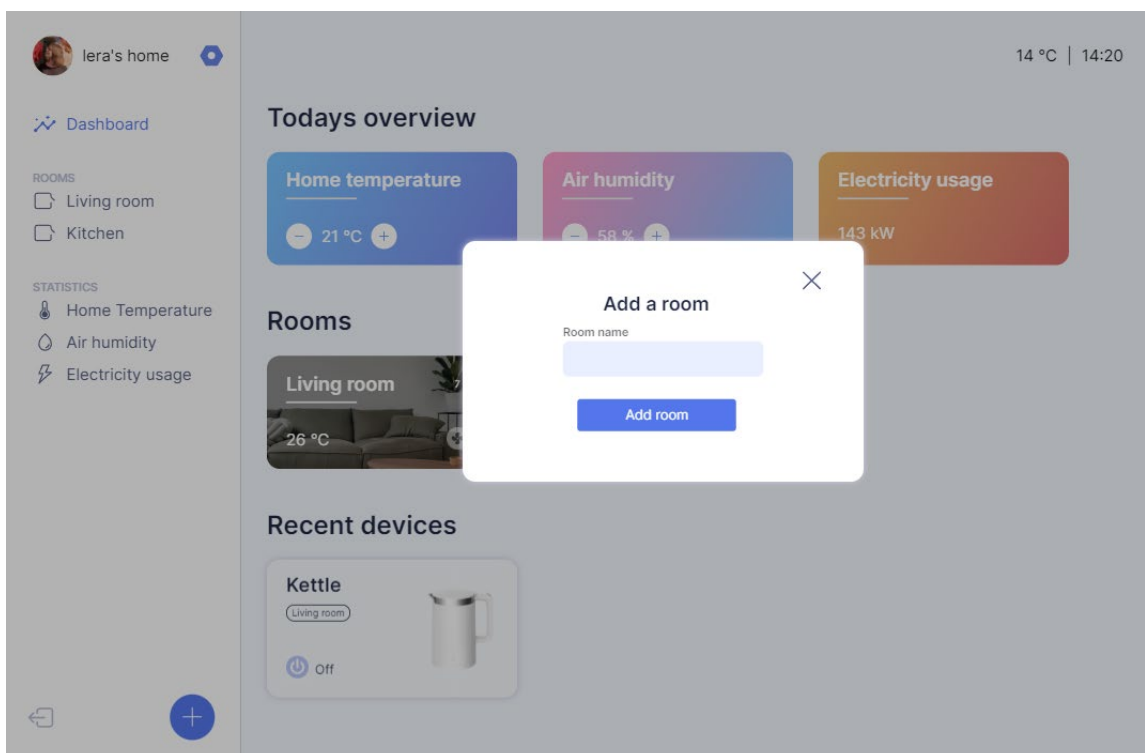


Рис. 2.14. Форма створення кімнати

Розглянемо сторінку кімнати (див. рис. 2.15). У верхньому правому куті знаходиться меню, через яке можна видалити кімнату. Нижче знаходиться блок із статистикою кімнати, такою як температура, вологість повітря та використання електроенергії на кімнату.

Коли в кімнаті працює кондиціонер, у блоці температури відображається, що температура знижується або підвищується. Якщо ж до кімнати не під'єднаний термометр та гігрометр, замість значень буде надпис «Connect thermometer» або «Connect hygrometer». Нижче знаходиться блок з девайсами, які підключені до цієї кімнати. З даної сторінки можна швидко вмикати/вимикати пристрої за допомогою кнопки.

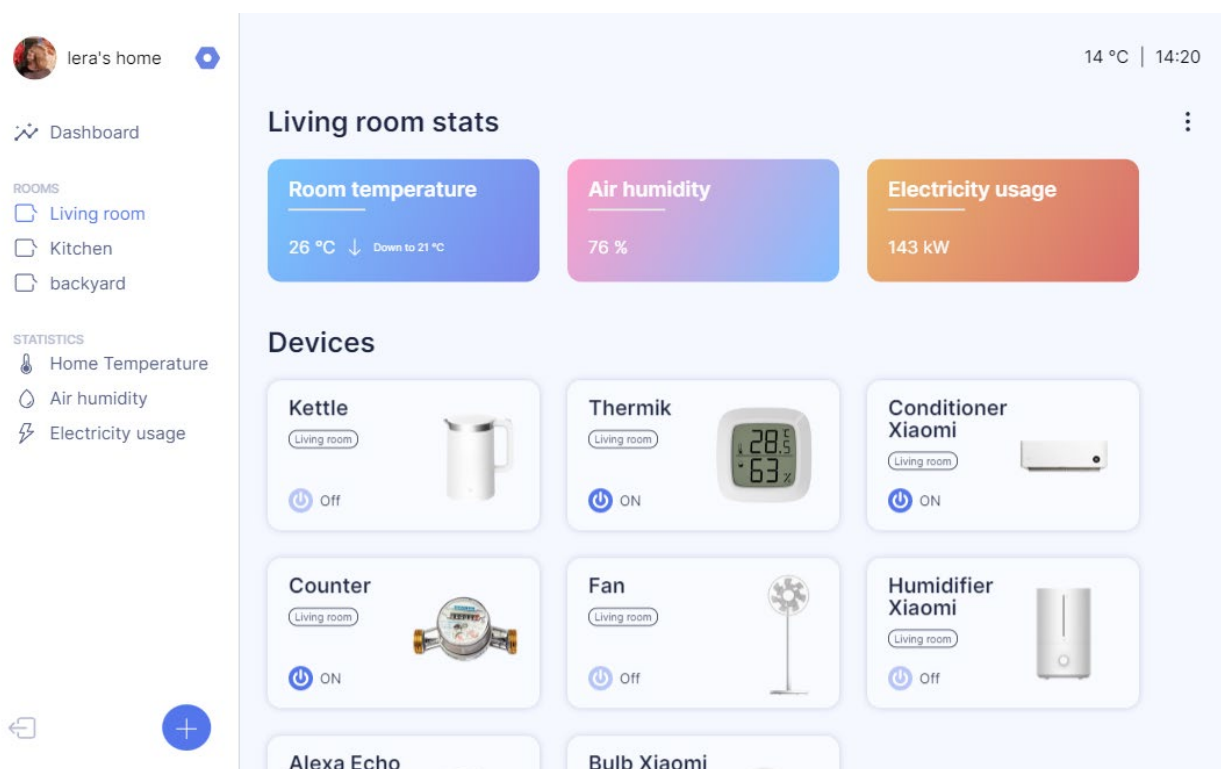


Рис. 2.15. Сторінка кімнати

Сторінка девайсу ділиться на дві частини (див. рис. 2.16). У лівій частині знаходиться всі елементи керування пристроєм, а у правій – зображення девайсу. Зверху знаходиться елемент керування «Хлібна крихта», яка перенаправляє користувача назад у кімнату, в якій знаходиться девайс. Нижче

розташована кнопка увімкнення девайсу, а також його назва та кімната. Далі знаходиться керування девайсом: налаштування температури, режиму та напряму повітряних жалюзі.

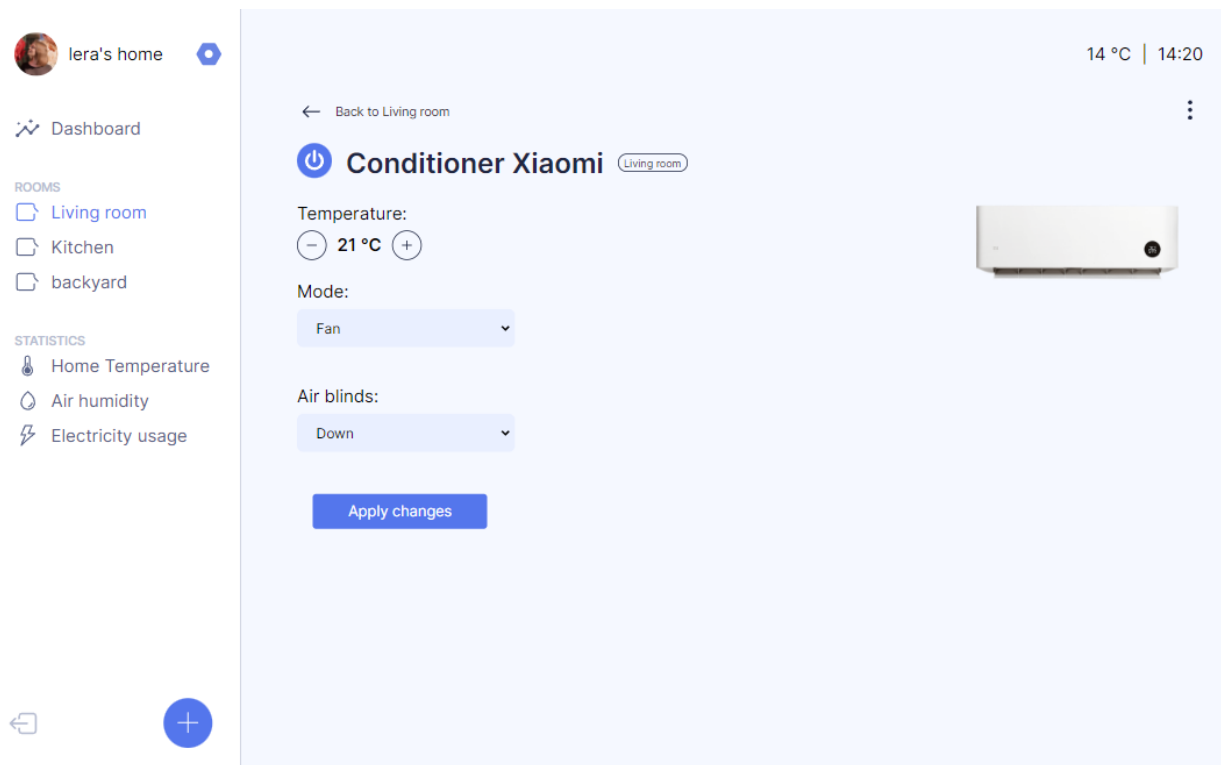


Рис. 2.16. Сторінка девайсу

Якщо перемкнутись на сторінку «Home temperature» (див. рис. 2.17), користувач опиниться на сторінці керування температурою у всьому будинку. На ній відображена температура в кожній кімнаті, а також список пристроїв, які впливають на температуру. Якщо натиснути на кнопку увімкнення глобальної температури, всі пристрої у домі увімкнуться та почнуть працювати на заданій температурі. За допомогою кнопок «+» та «-» можна регулювати значення температури.

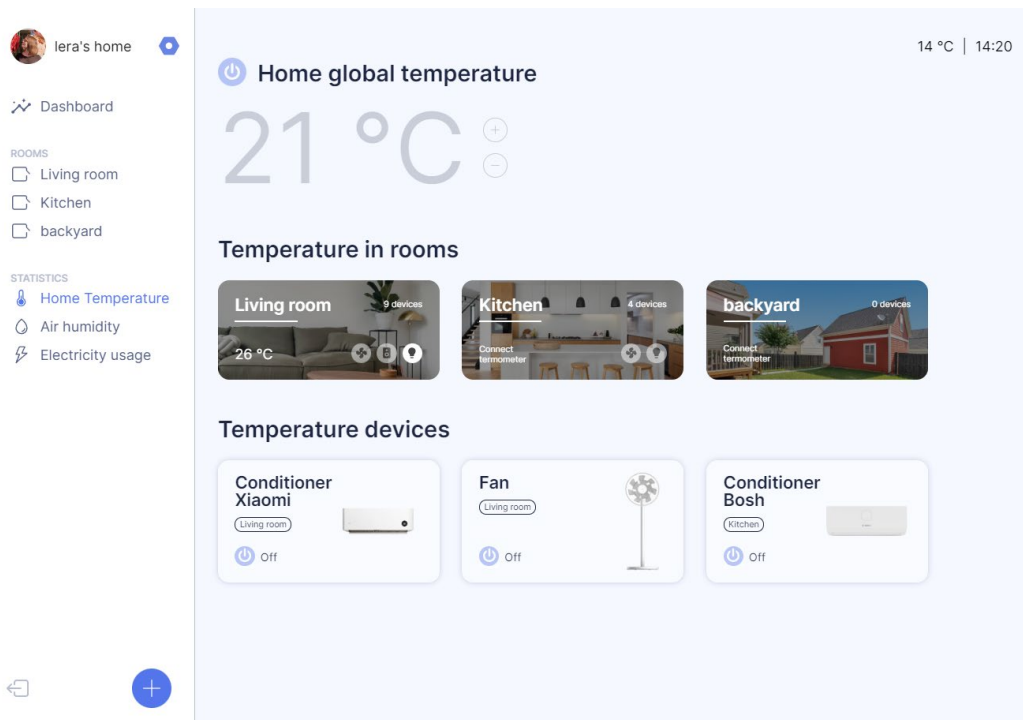


Рис. 2.17. Сторінка керування температурою у домі

На сторінці керування вологістю повітря (див. рис. 2.18) інтерфейс дуже схожий на попередню сторінку. Але є невеликі зміни – замість температури у кімнатах виводиться вологість повітря, а у блоці з девайсами виводяться тільки ті, які відповідають за вологість повітря.

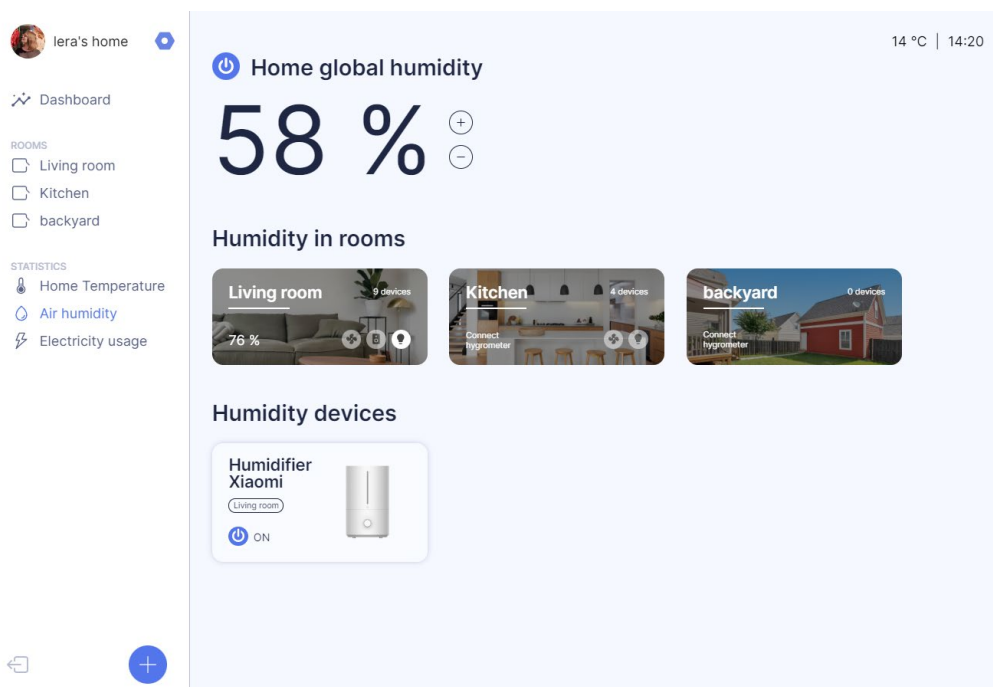


Рис. 2.18. Сторінка керування вологістю повітря у домі

Інтерфейс сторінки електроенергії (див. рис. 2.19) також схожий на попередні дві, за виключенням того, що її не можна регулювати. Дана сторінка призначена тільки для перегляду інформації про кількість вживаної електроенергії.

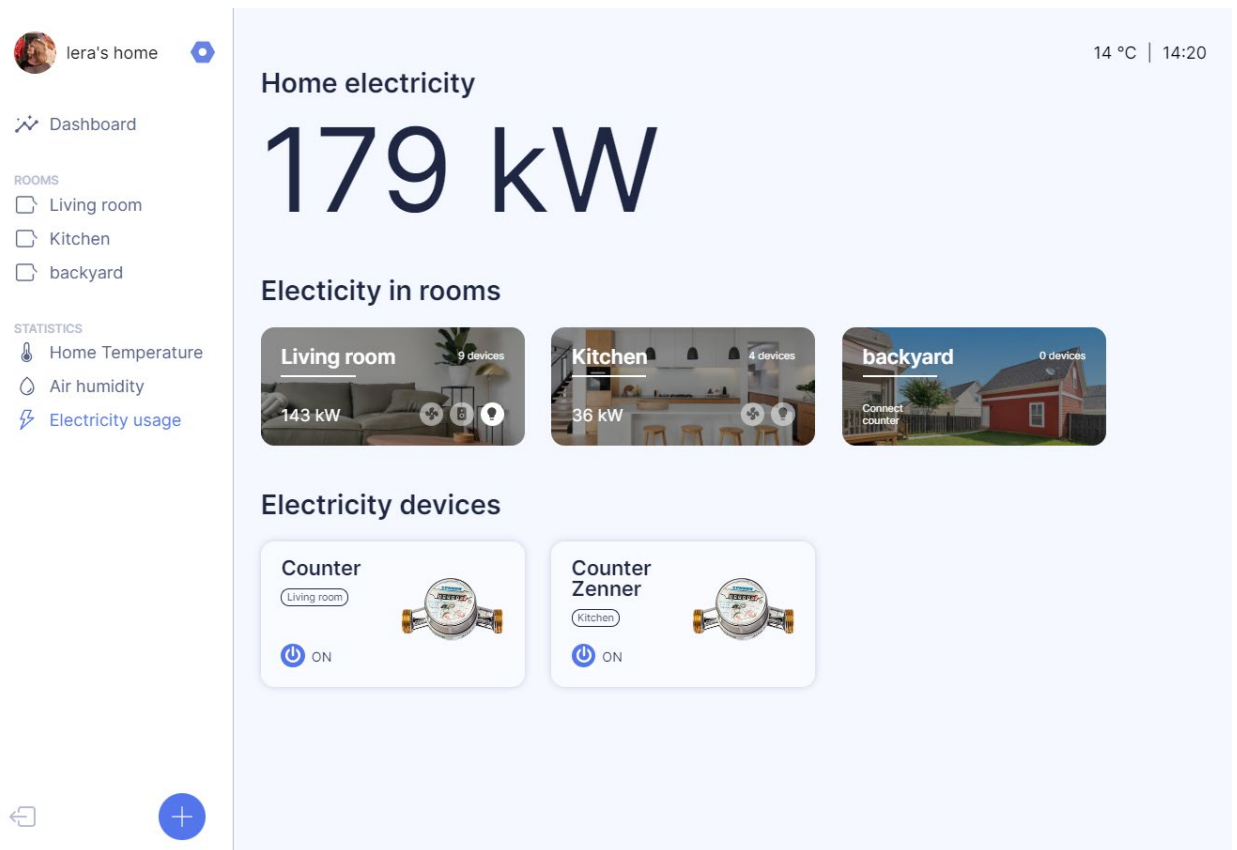


Рис. 2.19. Сторінка керування температурою у домі

Можна побачити, що на даній сторінці присутнє значення використаної електроенергії в усьому домі, а також покімнатно. Також на даній сторінці розміщені всі пристрої, які вимірюють витрати електроенергії в домі.

РОЗДІЛ 3

ЕКОНОМІЧНИЙ РОЗДІЛ

3.1 Розрахунок трудомісткості та вартості розробки програмного продукту

Початкові дані:

1. передбачуване число операторів програми – 2650;
2. коефіцієнт складності програми – 1,7;
3. коефіцієнт корекції програми в ході її розробки – 0,07;
4. годинна заробітна плата програміста – 177 грн/год;

За статистикою платформи найму "Djinni" на середину 2023 року, в середньому зарплата Junior Front End розробника в Україні коливається в діапазоні від 500\$ до 1200\$ в місяць [23]. Знаючи ці дані, можна розрахувати середню місячну заробітну плату, яка становить приблизно 850\$ у місяць. Ураховуючи офіційний курс валют Національного банку України на початок травня 2023 року, де один американський долар еквівалентний 36,56 гривням, середня місячна зарплата Junior Front End розробника в Україні складає 31076 грн. Якщо припустити стандартний робочий графік з 176 годинами на місяць, то зарплата за годину становитиме приблизно 177 гривень.

5. коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі – 1,2;
6. коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності – 1,4;
7. вартість машино-години ЕОМ – 0,8 грн/год.

Так як для розробки цього проекту використовувався потужний ПК та монітор 27 дюймів, треба враховувати вартість на енергозатрати та інтернету від провайдера «Київстар». Типовий персональний комп'ютер споживає 65-250 Вт за годину, монітор споживає 20-80 Вт [24]. Виходячи з характеристик ПК та монітору, врахуємо, що споживання електроенергії за годину виходить приблизно 280 Вт*год. За даними постачальнику електроенергії Yasno, тариф для населення на кінець травня становить 1,44 грн/кВт*год [25]. Отже,

вартість електроенергії для ПК та монітору становить $0,28 * 1,44 = 0,4$ грн/год. Домашнє інтернет-підключення від «Київстар» оплачується по тарифу «Все разом Домашній» [26] разом з телефонним зв'язком. Вартість тарифу в місяць 300 грн. Отже, година користування становить 0,4 грн. Загальна вартість машино-години ЕОМ становить 0,8 грн/год.

Нормування праці в процесі створення ПЗ істотно ускладнено в силу творчого характеру праці програміста. Тому трудомісткість розробки ПЗ може бути розрахована на основі системи моделей з різною точністю оцінки.

Трудомісткість розробки ПЗ можна розрахувати за формулою:

$$t = t_o + t_u + t_a + t_n + t_{oml} + t_{\partial}, \text{ людино-годин,} \quad (3.1)$$

де t_o - витрати праці на підготовку й опис поставленої задачі (приймається 50 людино-годин);

t_u - витрати праці на дослідження алгоритму рішення задачі;

t_a - витрати праці на розробку блок-схеми алгоритму;

t_n -витрати праці на програмування по готовій блок-схемі;

t_{oml} -витрати праці на налагодження програми на ЕОМ;

t_{∂} - витрати праці на підготовку документації.

Складові витрати праці визначаються через умовне число операторів у програмному забезпеченні, яке розробляється.

Умовне число операторів:

$$Q = q \cdot C \cdot (1 + p) \quad (3.2)$$

де q - передбачуване число операторів (100);

C - коефіцієнт складності програми (1,7);

p - коефіцієнт корекції програми в ході її розробки (0,07).

Отже, умовне число операторів в програмі має значення:

$$Q = 1,7 \cdot 2650 \cdot (1 + 0,07) = 4820,35$$

Витрати праці на вивчення опису задачі t_u визначається з урахуванням уточнення опису і кваліфікації програміста:

$$t_u = \frac{Q \cdot B}{(75..85) \cdot k}, \text{ людино-годин} \quad (3.3)$$

де B - коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі;

k - коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності. При стажі роботи від 2 до 3 років він складає 1,2.

Будемо вважати збільшення витрат праці внаслідок недостатнього опису завдання не більше 50% ($B = 1,2$). З урахуванням коефіцієнта кваліфікації $k = 1,2$, отримуємо витрати праці на вивчення опису завдання:

$$t_u = (4820,35 \cdot 1,2) / (85 \cdot 1,4) = 48,6 \text{ людино-годин.}$$

Витрати праці на розробку алгоритму рішення задачі визначаються за формулою:

$$t_a = \frac{Q}{(20...25) \cdot k}, \text{ людино-годин} \quad (3.4)$$

де Q – умовне число операторів програми;

k – коефіцієнт кваліфікації програміста.

Підставивши відповідні значення в формулу (3.4), отримуємо:

$$t_a = 4820,35 / (25 \cdot 1,4) = 137,7 \text{ людино-годин.}$$

Витрати на складання програми по готовій блок-схемі:

$$t_n = \frac{Q}{(20 \dots 25) \cdot k}, \text{ людино-годин} \quad (3.5)$$

Підставимо значення у формулу (3.5) та отримаємо витрати на складання програми по готовій блок-схемі:

$$t_n = 4820,35 / (24 \cdot 1,4) = 143,46 \text{ людино-годин.}$$

Витрати праці на налагодження програми на ЕОМ:

– за умови автономного налагодження одного завдання:

$$t_{oml} = \frac{Q}{(4 \dots 5) \cdot k}, \text{ людино-годин} \quad (3.6)$$

$$t_{oml} = 4820,35 / (5 \cdot 1,4) = 688,62 \text{ людино-годин.}$$

– за умови комплексного налагодження завдання:

$$t_{oml}^k = 1,5 \cdot t_{oml}, \text{ людино-годин} \quad (3.7)$$

$$t_{oml}^k = 1,5 \cdot 688,62 = 1032,93 \text{ людино-годин.}$$

Витрати праці на підготовку документації визначаються за формулою:

$$t_d = t_{dp} + t_{do}, \text{ людино-годин} \quad (3.8)$$

де t_{dp} -трудомісткість підготовки матеріалів і рукопису:

$$t_{dp} = \frac{Q}{(15..20) \cdot k}, \text{ людино-годин} \quad (3.9)$$

t_{do} - трудомісткість редагування, печатки й оформлення документації:

$$t_{do} = 0,75 \cdot t_{dp}, \text{ людино-годин.} \quad (3.9)$$

Підставляючи відповідні значення, отримаємо:

$$t_{dp} = 4820,35 / (20 \cdot 1,4) = 172,15 \text{ людино-годин.}$$

$$t_{do} = 0,75 \cdot 172,15 = 129,11 \text{ людино-годин.}$$

$$t_d = 172,15 + 129,11 = 301,26 \text{ людино-годин.}$$

Повертаючись до формули (3.1), отримаємо повну оцінку трудомісткості розробки програмного забезпечення:

$$t = 50 + 48,6 + 137,7 + 143,46 + 688,62 + 301,26 = 1369,64 \text{ людино-години.}$$

3.2. Розрахунок витрат на створення програми

Витрати на створення ПЗ $K_{ПО}$ включають витрати на заробітну плату виконавця програми $Z_{ЗП}$ і витрат машинного часу, необхідного на налагодження програми на ЕОМ:

$$K_{ПО} = Z_{ЗП} + Z_{МВ}, \text{ грн} \quad (3.10)$$

Заробітна плата виконавців визначається за формулою:

$$Z_{ЗП} = t \cdot C_{ПР}, \text{ грн} \quad (3.11)$$

де: t - загальна трудомісткість, людино-годин;

$C_{ПР}$ - середня годинна заробітна плата програміста, грн/година

З урахуванням того, що середня годинна зарплата програміста становить 177 грн / год, отримуємо:

$$Z_{ЗП} = 1369,64 \cdot 177 = 242313 \text{ грн.}$$

Вартість машинного часу, необхідного для налагодження програми на ЕОМ, визначається за формулою:

$$Z_{МВ} = t_{отл} \cdot C_{мч}, \text{ грн}, \quad (3.12)$$

де $t_{отл}$ - трудомісткість налагодження програми на ЕОМ, год;

$C_{мч}$ - вартість машино-години ЕОМ, грн/год (0,8 грн/год).

Підставивши в формулу (3.3) відповідні значення, визначимо вартість необхідного для налагодження машинного часу:

$$Z_{МВ} = 688,62 \cdot 0,8 = 550,9 \text{ грн.}$$

Звідси витрати на створення програмного продукту:

$$K_{ПО} = 242313 + 550,9 = 242819 \text{ грн.}$$

Очікуваний період створення ПЗ:

$$T = \frac{t}{B_k \cdot F_p}, \text{ міс.} \quad (3.13)$$

де B_k - число виконавців (дорівнює 1);

F_p - місячний фонд робочого часу (при 40 годинному робочому тижні $F_p=176$ годин).

Очікуваний період створення ПЗ:

$$T = 1369,64 / 1 \cdot 176 \approx 7,7 \text{ міс.}$$

Висновок: Розроблений інтерфейс є прототипом програмного забезпечення для керування пристроями у розумному будинку. Його основна мета – забезпечення зручного та швидкого керування пристроями та системами у розумному будинку. Вартість даного програмного забезпечення становить 242819 грн. Планується, що процес розробки займе приблизно 7,7 місяців. Цей термін обумовлений складністю поставлених задач та великим об'ємом роботи.

ВИСНОВКИ

У рамках даної кваліфікаційної роботи був розроблений інтерфейс для керування розумним будинком згідно з поставленим завданням. Основна мета цього інтерфейсу полягає у зручному відображенні статистики кімнат та швидкому керуванні окремими розумними пристроями та групами девайсів.

Цей інтерфейс має практичне застосування, оскільки надає потенційним користувачам можливість переглядати статистику важливих показників у розумних кімнатах, налаштовувати розумні пристрої, додавати або видаляти їх, змінювати параметри та керувати групами пристроїв і системами в будинку, наприклад, регулювання температури або вологості.

Структура розробленого програмного забезпечення базується на клієнт-серверній архітектурі, де розроблений інтерфейс виступає як клієнт, а локальний веб-сервер XAMPP виконує роль сервера. Взаємодія між клієнтом і сервером здійснюється за допомогою допоміжних файлів, які містять універсальні функції для відправки запитів на сервер та отримання результатів. Була розроблена база даних, яка є необхідною для правильної роботи інтерфейсу. Крім того, на етапі розробки було створено та розроблено маршрут користувача.

Для розробки даного інтерфейсу використовувалися мови розмітки HTML, мова стилів CSS, мови програмування JavaScript та PHP, а робоча середовище Microsoft Visual Studio Code 2022 використовувалося як інструмент для розробки. Веб-інтерфейс запускається на базі локального веб-серверу, який надає програмне забезпечення XAMPP. Адміністрування СУБД MySQL виконувалося через веб-інтерфейс PhpMyAdmin.

У процесі виконання кваліфікаційної роботи було визначено трудомісткість розробленої системи, а з використанням середньої зарплати розробника була розрахована вартість роботи, яка склала 242819 гривень. Також було оцінено час, необхідний для розробки інтернет-магазину, який склав 1369,64 людино-годин, що приблизно відповідає 7,7 місяцям роботи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Dr. Lasquety-Reyes J. Global: smart home number of users 2018-2027 | Statista. URL: <https://www.statista.com/forecasts/887613/number-of-smart-homes-in-the-smart-home-market-in-the-world> (дата звернення: 03.02.23)
2. Desai N. Which are the top & best smart home automation apps?. Blog Concetto Labs. URL: <https://www.concettolabs.com/blog/smart-home-automation-apps/> (дата звернення: 03.02.23)
3. Rowland C. Designing connected products: UX for the consumer Internet of things. 2015. 694 с.
4. Norman D. The Design Of Everyday Things. 2013. 368 с.
5. Олексюк Є. Швидка оцінка вашого інтерфейсного рішення без залучення цільової аудиторії. Великий гайд. UXPUВ Спільнота дизайнерів. URL: https://ux.pub/olex_world/non-user-usability-evaluation-3150 (дата звернення: 23.02.23)
6. Robbins J. N. Learning web design: A beginner's guide to HTML, CSS, JavaScript, and web graphics. 4th ed. Beijing : O'Reilly, 2012. 603 с.
7. Responsive Web Design (Brief Books for People Who Make Websites, No. 4) : навч. посіб. A Book Apart, 2011. 150 с.
8. Що таке веб додаток? | Блог WEBCASE. Webcase. URL: <https://webcase.com.ua/uk/blog/cho-takoe-web-prilozhenie-vse-vidy/> (дата звернення: 25.02.23)
9. Тестування веб-проектів: основні етапи та поради - QALight. QALight. URL: <https://qalight.ua/baza-znaniy/testuvannya-veb-proektiv-osnovni-etapi-ta-poradi/> (дата звернення: 25.02.23)
10. Spivey D. Home automation for dummies. 2015. 338 с.
11. Seidler K. O. XAMPP Hosting. URL: <https://www.apachefriends.org/hosting.html> (дата звернення: 05.03.23)
12. HTML Tutorial. W3Schools Online Web Tutorials. URL: <https://www.w3schools.com/html/default.asp> (дата звернення: 10.03.23)

13. Refsnes Data. CSS Приклади. Уроки для початківців. W3Schools українською. HTML CSS JavaScript PHP. W3Schools українською. Уроки для початківців.
URL: https://w3schoolsua.github.io/css/css_examples.html#gsc.tab=0 (дата звернення: 12.03.23)
14. Кантор І. Сучасний підручник з JavaScript. URL: <https://uk.javascript.info/> (дата звернення: 14.03.23)
15. Welling L., Thomson L. PHP and MySQL Web Development (4th Edition) (Developer's Library). Sams, 2007. 950 с.
16. Jamie Chan. SQL: Learn SQL (using MySQL) in One Day and Learn It Well. SQL for Beginners with Hands-on Project. – 2018. – 166с. ASIN : B07K374J19.
17. What is phpMyAdmin: Tutorial, Features, Installation - javatpoint. www.javatpoint.com. URL: <https://www.javatpoint.com/phpmyadmin> (дата звернення: 15.03.23)
18. PHP Cookies. W3Schools Online Web Tutorials. URL: https://www.w3schools.com/php/php_cookies.asp (дата звернення: 15.03.23)
19. Васильєв О. Книга Програмування мовою PHP : навч. посіб. Ліра-К, 2022. 368 с.
20. Usage Statistics and Market Share of PHP for Websites, May 2023. W3Techs - extensive and reliable web technology surveys. URL: <https://w3techs.com/technologies/details/pl-php> (дата звернення: 17.03.23)
21. Drasner S. SVG Animations: From Common UX Implementations to Complex Responsive Animation. O'Reilly Media, 2017. 246 с.
22. Lockhart J. Modern PHP: New Features and Good Practices. O'Reilly Media, 2015. 270 с.
23. Salary statistics JavaScript on Djinni. Djinni | Hire talent or find a job: remotely & on your own. URL: <https://djinni.co/salaries/javascript/?exp=0> (дата звернення: 16.05.23)

24. Українська правда Життя. Як зробити своє робоче або навчальне місце менш енергозатратним та більш функціональним. Українська правда Життя.

URL: <https://life.pravda.com.ua/society/2022/12/12/251751/#:~:text=Типовий%20ОПК%20та%20середньостатистичний%20монітор,використовувати%20як%20ододатковий%20Power%20Bank> (дата звернення: 17.05.23)

25. Тарифи на електроенергію для населення. YASNO.
URL: <https://yasno.com.ua/b2c-tariffs> (дата звернення: 17.05.23)

26. Мобільний зв'язок і Домашній Інтернет – всі послуги в одному тарифі з постійною економією. Мобільний зв'язок від Київстар | Національний оператор мобільного зв'язку. URL: <https://kyivstar.ua/tariffs-home-internet> (дата звернення: 17.05.23)

ЛІСТИНГ ПРОГРАМИ

Login_form.php

```

<!-- Форма авторизації -->

<?php
    //Початок сесії
    session_start();
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Smart Home LogIn</title>
    <link rel="stylesheet" href="/css/main.css">
    <link rel="stylesheet" href="/css/login.css">
</head>
<body>
    <div class="SignIn">
        <span class="app-description">Your smart home in one app!</span>
        <form method="POST" action="/handlers/login.php">
            <span class="h3">Log In</span>
            <?php
                if(isset($_COOKIE['errors'])){
                    echo '<span class="error-msg">'. $_COOKIE['errors'] . '</span>';
                    unset($_COOKIE['errors']);
                    setcookie('errors', null, -1, '/');
                } ?>

            <p class="input-group">
                <label class="label" for="text">Name</label>
                <input type="text" placeholder="Enter username" name="user" id="user" class="input"
                autocomplete="name" size="20">
            </p>

            <p class="input-group">
                <label class="label" for="text">Password</label>
                <input type="password" placeholder="Enter password" name="password" id="pass" class="input"
                autocomplete="name" size="20">
            </p>
            <input class="authButton" type="submit" name="submit" value="Log In">

            <a class="sign-up" href="/Register_form.php">Sign up</a>
        </form>
    </div>
</body>
</html>

```

Register_form.php

```

<!-- Форма реєстрації -->

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">

```

```

<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Smart Home Sign Up</title>

<link rel="stylesheet" href="/css/main.css">
<link rel="stylesheet" href="/css/login.css">

</head>
<body class="body-register">
  <div class="SignIn">
    <span class="app-description">Your smart home in one app!</span>
    <form method="POST" action="/handlers/register.php">
      <span class="h3">Sign up</span>

      <?php
        if(isset($_COOKIE['errors'])){
          echo '<span class="error-msg">'. $_COOKIE['errors'] . '</span>';
          unset($_COOKIE['errors']);
          setcookie('errors', null, -1, '/');
        } ?>

      <p class="input-group">
        <label class="label" for="text">Username</label>
        <input type="text" placeholder="Enter username" name="user" id="user" class="input"
autocomplete="name" size="20">
      </p>

      <p class="input-group">
        <label class="label" for="text">Password</label>
        <input type="password" placeholder="Enter password" name="password" id="pass" class="input"
autocomplete="name" size="20">
      </p>

      <p class="input-group">
        <label class="label" for="text">Confirm password</label>
        <input type="password" placeholder="Confirm password" name="cpassword" id="cpassword"
class="input" autocomplete="name" size="20">
      </p>

      <input class="authButton" type="submit" name="submit" value="Register">

      <a class="sign-up" href="/Login_form.php">Log in</a>
    </form>
  </div>
</body>
</html>

```

Dashboard.php

```
<!-- Сторінка дешборду -->
```

```

<?php
  session_start(); //Початок сесії
  $home_id = $_SESSION['homeId']; //Змінна, яка містить номер дому
  $user_name = $_SESSION['username']; //Змінна, яка містить номер дому

  require_once '/handlers/connection.php'; //Під'єдную сторінку зі з'єднанням до серверу
  require_once '/handlers/functions.php'; //Під'єдную сторінку з функціями

  $typesArray = select_smth_from_db("types"); //Отримую список типів
  $undertypesArray = select_smth_from_db("undertypes"); //Отримую список підтипів
  $brandsArray = select_smth_from_db("brands"); //Отримую список виробників

```

```

$roomsArray = get_rooms_by_homeId($home_id); //Отримую список кімнат

$home = get_smth_by_smth_row("?", "homes", "id", $home_id); //Отримую інформацію про дім
$counter = get_smth_by_smth_row("?", "counters", "id", 1);

$sql = "select * from rooms where homeId = $home_id";
$rooms = $conn->query($sql);
$rooms2 = $conn->query($sql);
?>

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Dashboard</title>

  <link rel="stylesheet" href="./css/main.css">
  <link rel="stylesheet" href="./css/forms.css">

</head>
<body>

  <div class="add_room_form">
    <button class="close_button"></button>
    <form method="POST" action="./handlers/add_room.php">
      <span class="h3">Add a room</span>
      <div class="input-group">
        <label class="label">Room name</label>
        <input maxlength="20" autocomplete="off" name="name" id="name" class="input" type="text">
      </div>
      <input class="authButtonDash" type="submit" name="submit" value="Add room">
    </form>
  </div>

  <div class="add_device_form">
    <button class="close_button"></button>
    <form method="POST" action="./handlers/add_device.php">
      <span class="h3">Add a device</span>
      <div class="input-group">
        <label class="label">Device name</label>
        <input maxlength="20" autocomplete="off" name="name" id="name" class="input" type="text">
      </div>

      <div class="input-group">
        <label class="label">Room</label>
        <select class="input" id="roomId" name="roomId">
          <?php
            if($rooms)
            {
              while($row = mysqli_fetch_assoc($rooms)){
                ?>
                <option value="<?php echo $row["id"];?>"><?php echo $row["name"];?></option>
              <?php
            }
          </select>
        </div>
      </div>

```

```

<div class="input-group">
  <label class="label">Undertype</label>
  <select class="input" id="undertype" name="undertype">
    <?php
      while($row = mysqli_fetch_assoc($undertypesArray)){
        ?>
        <option value="<?php echo $row["id"];?>"><?php echo $row["name"];?></option>
        <?php
          }
        ?>
      </select>
    </div>

    <div class="input-group">
      <label class="label">Brand</label>
      <select class="input" id="brand" name="brand">
        <?php
          while($row = mysqli_fetch_assoc($brandsArray)){
            ?>
            <option value="<?php echo $row["id"];?>"><?php echo $row["name"];?></option>
            <?php
              }
            ?>
          </select>
        </div>
        <input class="authButtonDash" type="submit" name="submit" value="Add device">
      </form>
    </div>

<div class="wrapper">
<nav>
  <div class="avatar-name">
    <div class="av-nm">
      <div class="avatar-image">
        
      </div>
      <?php echo '<span class="user-name body_text">' . $user_name . '\s home</span>' ?>
    </div>

    <div class="settings-btn">
      <a href="#">
        
      </a>
    </div>
  </div>

  <div class="menu-items">
    <ul class="dashboard-link">
      <li><a href="/Dashboard.php">
        <svg xmlns="http://www.w3.org/2000/svg" height="30" width="30" viewBox="0 96 960 960"
        ><path class="link_icon active" fill="#596387" d="M109.912 906 60.5 885.411 40 864.823 40 835.911 40
        807 60.494 786.5t49.273-20.5q5.233 0 10.233.5 5 .5 13 2.5t200-200q-2-8-2.5-13t-.5-10.233q0-28.779
        20.589-49.273Q371.177 476 400.089 476 429 476 449.5 496.634t20.5 49.61Q470 548 467 569l110 110q8-2 13-2.5t10-.5q5
        0 10 .5t13 2.5l160-160q-2-8-2.5-13t-.5-10.233q0-28.779 20.589-49.273Q821.177 426 850.089 426 879 426 899.5
        446.589q20.5 20.588 20.5 49.5Q920 525 899.506 545.5T850.233 566Q845 566 840 565.5q-.5-.5-13-2.5L667 723q2
        8 2.5 13t.5 10.233q0 28.779-20.589 49.273Q628.823 816 599.911 816 571 816 550.5 795.506T530 746.233q0-
        5.233.5-10.233.5-5 2.5-13L423 613q-8 2-13 2.5t-10.25.5q-1.75 0-22.75-3L177 813q2 8 2.5 13t.5 10.233q0 28.779-
        20.589 49.273Q138.823 906 109.912 906ZM160 464l-20.253-43.747L96 400l43.747-20.253L160 336l20.253
        43.747L224 400l-43.747 20.253L160 464Zm440-51-30.717-66.283L503 316l66.283-30.717L600 219l30.717
        66.283L697 316l-66.283 30.717L600 413Z"/></svg>
        <span class="link-name active">Dashboard</span>
      </li>
    </ul>
  </div>

```

```

    </a></li>
</ul>

<ul class="rooms-links">
<span class="links-description">ROOMS</span>
<?php
while($row = mysqli_fetch_assoc($roomsArray)){
?>

    <li><a href="Room_page.php?id=<?php echo $row["id"];?>"
        <svg width="30" height="30" viewBox="0 0 30 30" fill="none"
xmlns="http://www.w3.org/2000/svg"><path class="link_icon" fill-rule="evenodd" clip-rule="evenodd" d="M2.25
7.5C2.25 5.98122 3.48122 4.75 5 4.75H20.7143C22.2331 4.75 23.4643 5.98122 23.4643 7.5V9.40981C23.4643
9.74567 23.5994 10.0674 23.8393 10.3025L27.525 13.9151C27.8208 14.205 27.8256 14.6799 27.5356
14.9757C27.2457 15.2715 26.7708 15.2762 26.475 14.9863L22.7893 11.3737C22.2616 10.8565 21.9643 10.1487
21.9643 9.40981V7.5C21.9643 6.80964 21.4046 6.25 20.7143 6.25H5C4.30964 6.25 3.75 6.80964 3.75
7.5V22.5C3.75 23.1904 4.30964 23.75 5 23.75H20.7143C21.4046 23.75 21.9643 23.1904 21.9643
22.5V17.2619C21.9643 16.8477 22.3001 16.5119 22.7143 16.5119C23.1285 16.5119 23.4643 16.8477 23.4643
17.2619V22.5C23.4643 24.0188 22.2331 25.25 20.7143 25.25H5C3.48122 25.25 2.25 24.0188 2.25 22.5V7.5Z"
fill="#596387"/></svg>
        <span class="link-name"><?php echo $row["name"];?></span>
    </a></li>

<?php } ?>
</ul>

<ul class="rooms-links">
<span class="links-description">STATISTICS</span>
<li><a href="/Temperature_page.php">
        <svg width="30" height="30" viewBox="0 0 30 30" fill="none"
xmlns="http://www.w3.org/2000/svg"> <path class="link_icon" fill-rule="evenodd" clip-rule="evenodd"
d="M14.9701 3.125C13.1071 3.125 11.5968 4.63526 11.5968 6.49826V14.9711C9.77452 16.1081 8.56201 18.131
8.56201 20.4371C8.56201 23.9926 11.4444 26.875 14.9999 26.875C18.5555 26.875 21.4379 23.9926 21.4379
20.4371C21.4379 18.1059 20.1988 16.064 18.3433 14.9343V6.49826C18.3433 4.63526 16.8331 3.125 14.9701
3.125ZM12.3908 16.2437L13.0968 15.8032V14.9711V6.49826C13.0968 5.46369 13.9355 4.625 14.9701
4.625C16.0046 4.625 16.8433 5.46369 16.8433 6.49826V14.9343V15.7771L17.5632 16.2155C18.9903 17.0844
19.9379 18.6508 19.9379 20.4371C19.9379 23.1642 17.7271 25.375 14.9999 25.375C12.2728 25.375 10.062
23.1642 10.062 20.4371C10.062 18.67 10.9892 17.1182 12.3908 16.2437ZM18.766 20.4371C18.766 22.517 17.0799
24.2031 14.9999 24.2031C12.92 24.2031 11.2339 22.517 11.2339 20.4371C11.2339 18.6179 12.5237 17.1 14.2388
16.748V12.0845C14.2388 11.6806 14.5662 11.3532 14.9701 11.3532C15.374 11.3532 15.7015 11.6806 15.7015
12.0845V16.7362C17.4462 17.0649 18.766 18.5969 18.766 20.4371Z" fill="#596387"/> </svg>
        <span class="link-name">Home Temperature</span>
    </a></li>
<li><a href="/Humidity_page.php">
        <svg width="30" height="30" viewBox="0 0 30 30" fill="none"
xmlns="http://www.w3.org/2000/svg"> <path class="link_icon" d="M19.9432 7.91159C18.7492 6.51044 17.4083
5.24599 15.9443 4.14071C15.8137 4.04908 15.6587 4 15.5 4C15.3413 4 15.1863 4.04908 15.0557 4.14071C13.5917
5.24599 12.2508 6.51044 11.0568 7.91159C8.40057 11.0049 7 14.2749 7 17.3584C7 19.6503 7.89553 21.8483
9.48959 23.4689C11.0837 25.0896 13.2457 26 15.5 26C17.7543 26 19.9163 25.0896 21.5104 23.4689C23.1045
21.8483 24 19.6503 24 17.3584C24 14.2749 22.5994 11.0049 19.9432 7.91159ZM15.5 24.4288C13.6563 24.4262
11.8889 23.6805 10.5852 22.3551C9.28154 21.0297 8.54801 19.2328 8.54545 17.3584C8.54545 11.7414 13.9062
7.04743 15.5 5.77083C17.0938 7.04743 22.4545 11.7414 22.4545 17.3584C22.452 19.2328 21.7185 21.0297
20.4148 22.3551C19.1111 23.6805 17.3437 24.4262 15.5 24.4288ZM20.8318 18.2815C20.6445 19.4041 20.1178
20.4395 19.325 21.2437C18.5321 22.048 17.5126 22.5812 16.408 22.7692H16.2824C16.0877 22.7702 15.8998
22.6964 15.7563 22.5626C15.6128 22.4289 15.5243 22.2449 15.5084 22.0477C15.4925 21.8504 15.5505 21.6543
15.6707 21.4986C15.7909 21.3429 15.9645 21.239 16.1568 21.2078C16.9441 21.0738 17.6707 20.6934 18.2352
20.1195C18.7996 19.5456 19.1739 18.807 19.3057 18.0065C19.3232 17.9045 19.3603 17.807 19.415
17.7197C19.4697 17.6323 19.5408 17.5568 19.6243 17.4975C19.7078 17.4381 19.8019 17.3962 19.9014
17.374C20.0009 17.3518 20.1037 17.3499 20.204 17.3682C20.3049 17.3848 20.4015 17.4217 20.4881
17.477C20.5747 17.5322 20.6495 17.6045 20.7081 17.6897C20.7666 17.7749 20.8078 17.8712 20.829
17.9728C20.8503 18.0745 20.8512 18.1795 20.8318 18.2815Z" fill="#596387"/> </svg>
        <span class="link-name">Air humidity</span>
    </a></li>

```

```

        <li><a href="/Electricity_page.php">
            <svg width="30" height="30" viewBox="0 0 30 30" fill="none"
xmlns="http://www.w3.org/2000/svg"> <path class="link_icon" fill-rule="evenodd" clip-rule="evenodd"
d="M23.2274 11.998H17.3362L20.8481 4.04526C20.8984 3.93116 20.9189 3.8067 20.9075 3.68304C20.8961
3.55939 20.8532 3.44042 20.7828 3.3368C20.7123 3.23319 20.6165 3.14818 20.5038 3.08939C20.3912 3.03061
20.2652 2.99989 20.1373 3H12.4121C12.2608 2.99987 12.1128 3.04286 11.9865 3.12364C11.8601 3.20442 11.761
3.31943 11.7014 3.4544L7.06629 13.952C7.01589 14.0661 6.99548 14.1906 7.00687 14.3143C7.01826 14.4379
7.0611 14.5569 7.13156 14.6605C7.20201 14.7641 7.29786 14.8491 7.41052 14.9079C7.52317 14.9667 7.6491
14.9974 7.777 14.9973H11.5994L7.0601 25.9658C6.7511 26.7112 7.70439 27.3425 8.30695 26.7906L16.0692
19.6777L23.7296 13.3177C23.8496 13.2181 23.9353 13.0851 23.9751 12.9368C24.0149 12.7884 24.0069 12.6318
23.9522 12.488C23.8974 12.3442 23.7986 12.2202 23.6691 12.1325C23.5395 12.0449 23.3853 11.998 23.2274
11.998ZM16.165 13.4976H21.1354L15.0356 18.5635L9.80564 23.3549L13.4597 14.5264C13.5066 14.4126
13.5242 14.2894 13.5107 14.1676C13.4973 14.0457 13.4532 13.9289 13.3825 13.8274C13.3117 13.7258 13.2164
13.6427 13.1049 13.5852C12.9934 13.5277 12.869 13.4977 12.7428 13.4976H8.94815L12.922
4.49966H18.9662L15.4543 12.4524C15.4039 12.5665 15.3835 12.6909 15.3949 12.8146C15.4063 12.9382 15.4491
13.0572 15.5196 13.1608C15.59 13.2644 15.6859 13.3495 15.7985 13.4082C15.9112 13.467 16.0371 13.4977
16.165 13.4976Z" fill="#596387"/> </svg>
            <span class="link-name">Electricity usage</span>
        </a></li>
    </ul>
</div>

<div class="add-button-menu">
    <a class="body_text add-room-button">Add room</a>
    <a class="body_text add-device-button">Add device</a>
</div>

<div class="exit-add">
    <ul class="logout">
        <li><a href="/handlers/logout.php">
            
        </a></li>
    </ul>

    <button class="add_button">
        
    </button>
</div>
</nav>

<section class="dashboard">
    <div>
        <div class="top_info">
            <div class="menu_icon"></div>
            <div class="time_temp">
                <span class="body_text">14 °C</span>
                <span class="body_text">|</span>
                <span class="body_text">14:20</span>
                <button class="add_button">  </button>
            </div>
        </div>
        <div class="add-button-menu-adaptive">
            <a class="body_text add-room-button">Add room</a>
            <a class="body_text add-device-button">Add device</a>
        </div>
    </div>
    <div class="dash_content">
        <div class="title title_margin">
            <span class="h2">Todays overview</span>
        </div>

        <div class="boxes">
            <div class="box temperature_box first-box">

```

```

        <a class="h3" href="/Temperature_page.php">Home temperature</a>
        <hr noshade class="dash">
        <div class="regulation">
            <button
                class="regulation_button"
                onclick="location.href='/handlers/global_temp.php?action=minus&from=dashboard'"></button>
            <span class="body_text box_text"><?php echo $home["temperature"]?> °C</span>
            <button
                class="regulation_button"
                onclick="location.href='/handlers/global_temp.php?action=plus&from=dashboard'"></button>
        </div>
    </div>
    <div class="box humidity_box">
        <a class="h3" href="/Humidity_page.php">Air humidity</a>
        <hr noshade class="dash">
        <div class="regulation">
            <button
                class="regulation_button"
                onclick="location.href='/handlers/global_hum.php?action=minus&from=dashboard'"></button>
            <span class="body_text box_text"><?php echo $home["humidity"]?> %</span>
            <button
                class="regulation_button"
                onclick="location.href='/handlers/global_hum.php?action=plus&from=dashboard'"></button>
        </div>
    </div>
    <div class="box electricity_box last-box">
        <a class="h3" href="#">Electricity usage</a>
        <hr noshade class="dash">
        <div class="regulation">
            <span class="body_text box_text"><?php echo $counter["electricity"]?> kW</span>
        </div>
    </div>
</div>

<div class="title title_margin">
    <span class="h2">Rooms</span>
</div>

<div class="boxes">
    <?php
    while($row = mysqli_fetch_assoc($rooms2)){
        $roomClass = str_replace(" ", "-", strtolower($row["name"])); ?>

        <div class="box <?php echo $roomClass?> first-box" id="<?php $row["id"];?>">
            <div class="room-name">
                <a class="h3" href="Room_page.php?id=<?php echo $row["id"];?>"><?php echo
                    $row["name"];?></a>
                <span class="small-text">
                    <?php $deviceCount = get_smth_by_smth_count("*", "devices", "roomId", $row["id"]);
                    echo $deviceCount?>

                    devices
                </span>
            </div>
            <hr noshade class="dash">
            <div class="room_regulation">
                <?php

                $stermostatId = get_smth_by_2smth_row("*", "devices", "undertype", 3, "roomId", $row["id"]);
                if(isset($stermostatId))
                {
                    $roomStats = get_smth_by_smth_row("*", "thermometers", "deviceId", $stermostatId["id"]);
                    ?>
                    <span class="body_text box_text"><?php echo $roomStats["temperature"];?> °C</span>
                }
            </div>
        </div>
    }

```



```

<?php } else {?>
<span class="body_text box_text"> </span> <?php } ?>

<div class="room-regulation-items">
  <?php
    $ventIsActive = get_smth_by_3smth_count("*, "devices", "type", 2, "roomId", $row["id"],
"isActive", 1);
    $ventCount = get_smth_by_2smth_count("*, "devices", "type", 2, "roomId", $row["id"]);
    $soundIsActive = get_smth_by_3smth_count("*, "devices", "type", 6, "roomId", $row["id"],
"isActive", 1);
    $soundCount = get_smth_by_2smth_count("*, "devices", "type", 6, "roomId", $row["id"]);
    $lightIsActive = get_smth_by_3smth_count("*, "devices", "type", 7, "roomId", $row["id"],
"isActive", 1);
    $lightCount = get_smth_by_2smth_count("*, "devices", "type", 7, "roomId", $row["id"]);
  ?>

  <?php if($ventCount > 0) {
    if($ventCount > $ventIsActive) {?>
      <button
        class="room-reg"
        onclick="location.href='./handlers/system_control.php?id=<?php
$row["id"]?>&system=vent&action=activate';"></button>
      <?php }
      if($ventCount == $ventIsActive) { ?>
        <button
          class="room-reg
          active"
          onclick="location.href='./handlers/system_control.php?id=<?php
$row["id"]?>&system=vent&action=deactivate';"></button>
        <?php } }?>

    <?php if($soundCount > 0) {
      if($soundCount > $soundIsActive) {?>
        <button
          class="room-reg"
          onclick="location.href='./handlers/system_control.php?id=<?php
$row["id"]?>&system=sound&action=activate';"></button>
        <?php }
        if($soundCount == $soundIsActive) { ?>
          <button
            class="room-reg
            active"
            onclick="location.href='./handlers/system_control.php?id=<?php
$row["id"]?>&system=sound&action=deactivate';"></button>
          <?php } }?>

    <?php if($lightCount > 0) {
      if($lightCount > $lightIsActive) {?>
        <button
          class="room-reg"
          onclick="location.href='./handlers/system_control.php?id=<?php
$row["id"]?>&system=light&action=activate';"></button>
        <?php }
        if($lightCount == $lightIsActive) { ?>
          <button
            class="room-reg
            active"
            onclick="location.href='./handlers/system_control.php?id=<?php
$row["id"]?>&system=light&action=deactivate';"></button>
          <?php } }?>
      </div>
    </div>
  </div>
  <?php } ?>
</div>

```

```

<div class="title title_margin ">
  <span class="h2">Recent devices</span>
</div>

<div class="boxes">
  <?php

  if (isset($_COOKIE["recent_devices"])){
    // var_dump($_COOKIE["recent_devices"]); die();
    $recent_devices = unserialize($_COOKIE["recent_devices"]);
    // var_dump(unserialize($_COOKIE["recent_devices"])); die();

    foreach($recent_devices as &$deviceId){
      $device = get_smth_by_smth_row("*", "devices", "id", $deviceId );

    ?>
    <div class="device-box first-box">
      <div class="device-info">
        <div class="device-name-room">
          <?php $undertype = get_smth_by_smth_row("name", "undertypes", "id",
$device["undertype"])?>
          <a class="device-name" href="<?php echo $undertype["name"]?>.php?id=<?php echo
$device["id"]?>"> <?php echo $device["name"]?> </a>
          <span class="small-text device-room"><?php
            $room_dev_name = get_smth_by_smth_row("name", "rooms", "id", $device["roomId"]);
            echo $room_dev_name["name"];
          ?></span>
        </div>

        <div class="device-regulation">
          <?php
            if($device["isActive"] == 0){
              ?>
              <a class="device-button" href="/handlers/activate_device.php?id=<?php echo
$device["id"]?>&from=dashboard">  </a>
              <span class="medium-text device-action">Off</span>
              <?php }
            else{
              ?>
              <a class="device-button active" href="/handlers/deactivate_device.php?id=<?php echo
$device["id"]?>&from=dashboard">  </a>
              <span class="medium-text device-action">ON</span>
              <?php } ?>
            </div>
          </div>
          <a href="<?php echo $undertype["name"]?>.php?id=<?php echo $device["id"]?>">
          <?php
            $deviceImg = get_device_image($device["undertype"], $device["brand"]);

            echo '';
          ?>
        </a>
      </div>
    <?php
      } }
    ?>
  </div>
</div>
</section>
</div>

```

```

<script type="text/javascript" src="/scripts/main.js"></script>
</body>
</html>

```

Room_page.php

```

<!-- Шаблон сторінки кімнати -->

```

```

<?php
    session_start(); //Початок сесії
    $home_id = $_SESSION['homeId']; //Змінна, яка містить номер дому
    $user_name = $_SESSION['username']; //Змінна, яка містить номер дому

    $room_id = "0"; //З параметрів URL отримую номер кімнати
    if(isset($_GET['id'])){
        $room_id = $_GET['id'];
    }

    require_once './handlers/connection.php'; //Під'єдную сторінку зі з'єднанням до серверу
    require_once './handlers/functions.php'; //Під'єдную сторінку з функціями

    $typesArray = select_smth_from_db("*", "types");
    $undertypesArray = select_smth_from_db("*", "undertypes");
    $brandsArray = select_smth_from_db("*", "brands");

    $devicesArray = get_smth_by_smth("*", "devices", "roomId", $room_id);
    $room = get_smth_by_smth_row("*", "rooms", "id", $room_id);
    $roomsArray = get_rooms_by_homeId($home_id);
    $rooms = get_rooms_by_homeId($home_id);

    $sql = "select * from rooms where homeId = $home_id";

    //Отримую всю інформацію про температуру, електроенергію та вентиляцію
    $thermostatId = get_smth_by_2smth_row("*", "devices", "undertype", 3, "roomId", $room_id);
    if(isset($thermostatId))
    {$roomStats = get_smth_by_smth_row("*", "thermometers", "deviceId", $thermostatId["id"]);}

    $counter = get_smth_by_2smth_row("*", "devices", "undertype", 5, "roomId", $room_id);
    if(isset($counter))
    {$counterStats = get_smth_by_smth_row("*", "counters", "deviceId", $counter["id"]);}

    $condition = get_smth_by_2smth_row("*", "devices", "undertype", 2, "roomId", $room_id);
    if(isset($condition))
    {$conditionStats = get_smth_by_smth_row("*", "conditioners", "deviceId", $condition["id"]);}
    $humidifierId = get_smth_by_2smth_row("*", "devices", "undertype", 4, "roomId", $room_id);
?>

```

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Dashboard</title>

    <link rel="stylesheet" href="/css/main.css">
    <link rel="stylesheet" href="/css/roompage.css">
    <link rel="stylesheet" href="/css/forms.css">

</head>
<body>

    <div class="add_room_form">

```

```

<button class="close_button"></button>
<form method="POST" action="./handlers/add_room.php">
  <span class="h3">Add a room</span>
  <div class="input-group">
    <label class="label">Room name</label>
    <input maxlength="20" autocomplete="off" name="name" id="name" class="input" type="text">
  </div>
  <input class="authButtonDash" type="submit" name="submit" value="Add room">
</form>
</div>

<div class="add_device_form">
  <button class="close_button"></button>
  <form method="POST" action="./handlers/add_device.php">
    <span class="h3">Add a device</span>
    <div class="input-group">
      <label class="label">Device name</label>
      <input maxlength="20" autocomplete="off" name="name" id="name" class="input" type="text">
    </div>

    <div class="input-group">
      <label class="label">Room</label>
      <select class="input" id="roomId" name="roomId">
        <?php
          if($rooms)
          {
            while($row = mysqli_fetch_assoc($rooms)){
              ?>
              <option value="<?php echo $row["id"];?>"><?php echo $row["name"];?></option>
              <?php
            }
          }
          ?>
        </select>
      </div>

      <div class="input-group">
        <label class="label">Undertype</label>
        <select class="input" id="undertype" name="undertype">
          <?php
            while($row = mysqli_fetch_assoc($undertypesArray)){
              ?>
              <option value="<?php echo $row["id"];?>"><?php echo $row["name"];?></option>
              <?php
            }
          ?>
        </select>
      </div>

      <div class="input-group">
        <label class="label">Brand</label>
        <select class="input" id="brand" name="brand">
          <?php
            while($row = mysqli_fetch_assoc($brandsArray)){
              ?>
              <option value="<?php echo $row["id"];?>"><?php echo $row["name"];?></option>
              <?php
            }
          ?>
        </select>
      </div>
      <input class="authButtonDash" type="submit" name="submit" value="Add device">

```

```

        </form>
    </div>

    <div class="wrapper">
    <nav>
    <div class="avatar-name">
    <div class="av-nm">
    <div class="avatar-image">
    
    </div>
    <?php echo '<span class="user-name body_text">' . $user_name . '\s home</span>' ?>
    </div>

    <div class="settings-btn">
    <a href="#">
    
    </a>
    </div>
    </div>

    <div class="menu-items">
    <ul class="dashboard-link">
    <li><a href="/Dashboard.php">
    <svg xmlns="http://www.w3.org/2000/svg" height="30" width="30" viewBox="0 96 960 960">
    <path class="link_icon" fill="#596387" d="M109.912 906Q81 906 60.5 885.411 40 864.823 40 835.911 40 807
    60.494 786.5t49.273-20.5q5.233 0 10.233.5 5 .5 13 2.5l200-200q-2-8-2.5-13t-.5-10.233q0-28.779 20.589-
    49.273Q371.177 476 400.089 476 429 476 449.5 496.634t20.5 49.61Q470 548 467 569l110 110q8-2 13-2.5t10-.5q5
    0 10 .5t13 2.5l160-160q-2-8-2.5-13t-.5-10.233q0-28.779 20.589-49.273Q821.177 426 850.089 426 879 426 899.5
    446.589q20.5 20.588 20.5 49.5Q920 525 899.506 545.5T850.233 566Q845 566 840 565.5q-.5-.5-13-2.5L667 723q2
    8 2.5 13t.5 10.233q0 28.779-20.589 49.273Q628.823 816 599.911 816 571 816 550.5 795.506T530 746.233q0-
    5.233.5-10.233.5-.5 2.5-13L423 613q-8 2-13 2.5t-10.25.5q-1.75 0-22.75-3L177 813q2 8 2.5 13t.5 10.233q0 28.779-
    20.589 49.273Q138.823 906 109.912 906ZM160 464l-20.253-43.747L96 400l43.747-20.253L160 336l20.253
    43.747L224 400l-43.747 20.253L160 464Zm440-51-30.717-66.283L503 316l66.283-30.717L600 219l30.717
    66.283L697 316l-66.283 30.717L600 413Z"/></svg>
    <span class="link-name">Dashboard</span>
    </a></li>
    </ul>

    <ul class="rooms-links">
    <span class="links-description">ROOMS</span>
    <?php
    while($row = mysqli_fetch_assoc($roomsArray)){

    if($row["id"] == $room_id){
    ?>
    <li><a href="Room_page.php?id=<?php echo $row["id"];?>">
    <svg width="30" height="30" viewBox="0 0 30 30" fill="none"
    xmlns="http://www.w3.org/2000/svg"><path class="link_icon active" fill-rule="evenodd" clip-rule="evenodd"
    d="M2.25 7.5C2.25 5.98122 3.48122 4.75 5 4.75H20.7143C22.2331 4.75 23.4643 5.98122 23.4643
    7.5V9.40981C23.4643 9.74567 23.5994 10.0674 23.8393 10.3025L27.525 13.9151C27.8208 14.205 27.8256
    14.6799 27.5356 14.9757C27.2457 15.2715 26.7708 15.2762 26.475 14.9863L22.7893 11.3737C22.2616 10.8565
    21.9643 10.1487 21.9643 9.40981V7.5C21.9643 6.80964 21.4046 6.25 20.7143 6.25H5C4.30964 6.25 3.75 6.80964
    3.75 7.5V22.5C3.75 23.1904 4.30964 23.75 5 23.75H20.7143C21.4046 23.75 21.9643 23.1904 21.9643
    22.5V17.2619C21.9643 16.8477 22.3001 16.5119 22.7143 16.5119C23.1285 16.5119 23.4643 16.8477 23.4643
    17.2619V22.5C23.4643 24.0188 22.2331 25.25 20.7143 25.25H5C3.48122 25.25 2.25 24.0188 2.25 22.5V7.5Z"
    fill="#596387"/></svg>
    <span class="link-name active"><?php echo $row["name"];?></span>
    </a></li>
    <?php }

    else{
    ?>

```

```
<li><a href="Room_page.php?id=<?php echo $row["id"];?>">
  <svg width="30" height="30" viewBox="0 0 30 30" fill="none"
xmlns="http://www.w3.org/2000/svg"><path class="link_icon" fill-rule="evenodd" clip-rule="evenodd" d="M2.25
7.5C2.25 5.98122 3.48122 4.75 5 4.75H20.7143C22.2331 4.75 23.4643 5.98122 23.4643 7.5V9.40981C23.4643
9.74567 23.5994 10.0674 23.8393 10.3025L27.525 13.9151C27.8208 14.205 27.8256 14.6799 27.5356
14.9757C27.2457 15.2715 26.7708 15.2762 26.475 14.9863L22.7893 11.3737C22.2616 10.8565 21.9643 10.1487
21.9643 9.40981V7.5C21.9643 6.80964 21.4046 6.25 20.7143 6.25H5C4.30964 6.25 3.75 6.80964 3.75
7.5V22.5C3.75 23.1904 4.30964 23.75 5 23.75H20.7143C21.4046 23.75 21.9643 23.1904 21.9643
22.5V17.2619C21.9643 16.8477 22.3001 16.5119 22.7143 16.5119C23.1285 16.5119 23.4643 16.8477 23.4643
17.2619V22.5C23.4643 24.0188 22.2331 25.25 20.7143 25.25H5C3.48122 25.25 2.25 24.0188 2.25 22.5V7.5Z"
fill="#596387"/></svg>
  <span class="link-name"><?php echo $row["name"];?></span>
</a></li>

<?php } } ?>
</ul>

<ul class="rooms-links">
  <span class="links-description">STATISTICS</span>
  <li><a href="/Temperature_page.php">
    <svg width="30" height="30" viewBox="0 0 30 30" fill="none"
xmlns="http://www.w3.org/2000/svg"> <path class="link_icon" fill-rule="evenodd" clip-rule="evenodd"
d="M14.9701 3.125C13.1071 3.125 11.5968 4.63526 11.5968 6.49826V14.9711C9.77452 16.1081 8.56201 18.131
8.56201 20.4371C8.56201 23.9926 11.4444 26.875 14.9999 26.875C18.5555 26.875 21.4379 23.9926 21.4379
20.4371C21.4379 18.1059 20.1988 16.064 18.3433 14.9343V6.49826C18.3433 4.63526 16.8331 3.125 14.9701
3.125ZM12.3908 16.2437L13.0968 15.8032V14.9711V6.49826C13.0968 5.46369 13.9355 4.625 14.9701
4.625C16.0046 4.625 16.8433 5.46369 16.8433 6.49826V14.9343V15.7771L17.5632 16.2155C18.9903 17.0844
19.9379 18.6508 19.9379 20.4371C19.9379 23.1642 17.7271 25.375 14.9999 25.375C12.2728 25.375 10.062
23.1642 10.062 20.4371C10.062 18.67 10.9892 17.1182 12.3908 16.2437ZM18.766 20.4371C18.766 22.517 17.0799
24.2031 14.9999 24.2031C12.92 24.2031 11.2339 22.517 11.2339 20.4371C11.2339 18.6179 12.5237 17.1 14.2388
16.748V12.0845C14.2388 11.6806 14.5662 11.3532 14.9701 11.3532C15.374 11.3532 15.7015 11.6806 15.7015
12.0845V16.7362C17.4462 17.0649 18.766 18.5969 18.766 20.4371Z" fill="#596387"/> </svg>
    <span class="link-name">Home Temperature</span>
  </a></li>
  <li><a href="/Humidity_page.php">
    <svg width="30" height="30" viewBox="0 0 30 30" fill="none"
xmlns="http://www.w3.org/2000/svg"> <path class="link_icon" d="M19.9432 7.91159C18.7492 6.51044 17.4083
5.24599 15.9443 4.14071C15.8137 4.04908 15.6587 4 15.5 4C15.3413 4 15.1863 4.04908 15.0557 4.14071C13.5917
5.24599 12.2508 6.51044 11.0568 7.91159C8.40057 11.0049 7 14.2749 7 17.3584C7 19.6503 7.89553 21.8483
9.48959 23.4689C11.0837 25.0896 13.2457 26 15.5 26C17.7543 26 19.9163 25.0896 21.5104 23.4689C23.1045
21.8483 24 19.6503 24 17.3584C24 14.2749 22.5994 11.0049 19.9432 7.91159ZM15.5 24.4288C13.6563 24.4262
11.8889 23.6805 10.5852 22.3551C9.28154 21.0297 8.54801 19.2328 8.54545 17.3584C8.54545 11.7414 13.9062
7.04743 15.5 5.77083C17.0938 7.04743 22.4545 11.7414 22.4545 17.3584C22.452 19.2328 21.7185 21.0297
20.4148 22.3551C19.1111 23.6805 17.3437 24.4262 15.5 24.4288ZM20.8318 18.2815C20.6445 19.4041 20.1178
20.4395 19.325 21.2437C18.5321 22.048 17.5126 22.5812 16.408 22.7692H16.2824C16.0877 22.7702 15.8998
22.6964 15.7563 22.5626C15.6128 22.4289 15.5243 22.2449 15.5084 22.0477C15.4925 21.8504 15.5505 21.6543
15.6707 21.4986C15.7909 21.3429 15.9645 21.239 16.1568 21.2078C16.9441 21.0738 17.6707 20.6934 18.2352
20.1195C18.7996 19.5456 19.1739 18.807 19.3057 18.0065C19.3232 17.9045 19.3603 17.807 19.415
17.7197C19.4697 17.6323 19.5408 17.5568 19.6243 17.4975C19.7078 17.4381 19.8019 17.3962 19.9014
17.374C20.0009 17.3518 20.1037 17.3499 20.204 17.3682C20.3049 17.3848 20.4015 17.4217 20.4881
17.477C20.5747 17.5322 20.6495 17.6045 20.7081 17.6897C20.7666 17.7749 20.8078 17.8712 20.829
17.9728C20.8503 18.0745 20.8512 18.1795 20.8318 18.2815Z" fill="#596387"/> </svg>
    <span class="link-name">Air humidity</span>
  </a></li>
  <li><a href="/Electricity_page.php">
    <svg width="30" height="30" viewBox="0 0 30 30" fill="none"
xmlns="http://www.w3.org/2000/svg"> <path class="link_icon" fill-rule="evenodd" clip-rule="evenodd"
d="M23.2274 11.998H17.3362L20.8481 4.04526C20.8984 3.93116 20.9189 3.8067 20.9075 3.68304C20.8961
3.55939 20.8532 3.44042 20.7828 3.3368C20.7123 3.23319 20.6165 3.14818 20.5038 3.08939C20.3912 3.03061
20.2652 2.99989 20.1373 3H12.4121C12.2608 2.99987 12.1128 3.04286 11.9865 3.12364C11.8601 3.20442 11.761
3.31943 11.7014 3.4544L7.06629 13.952C7.01589 14.0661 6.99548 14.1906 7.00687 14.3143C7.01826 14.4379
7.0611 14.5569 7.13156 14.6605C7.20201 14.7641 7.29786 14.8491 7.41052 14.9079C7.52317 14.9667 7.6491
14.9974 7.777 14.9973H11.5994L7.0601 25.9658C6.7511 26.7112 7.70439 27.3425 8.30695 26.7906L16.0692
```

```

19.6777L23.7296 13.3177C23.8496 13.2181 23.9353 13.0851 23.9751 12.9368C24.0149 12.7884 24.0069 12.6318
23.9522 12.488C23.8974 12.3442 23.7986 12.2202 23.6691 12.1325C23.5395 12.0449 23.3853 11.998 23.2274
11.998ZM16.165 13.4976H21.1354L15.0356 18.5635L9.80564 23.3549L13.4597 14.5264C13.5066 14.4126
13.5242 14.2894 13.5107 14.1676C13.4973 14.0457 13.4532 13.9289 13.3825 13.8274C13.3117 13.7258 13.2164
13.6427 13.1049 13.5852C12.9934 13.5277 12.869 13.4977 12.7428 13.4976H8.94815L12.922
4.49966H18.9662L15.4543 12.4524C15.4039 12.5665 15.3835 12.6909 15.3949 12.8146C15.4063 12.9382 15.4491
13.0572 15.5196 13.1608C15.59 13.2644 15.6859 13.3495 15.7985 13.4082C15.9112 13.467 16.0371 13.4977
16.165 13.4976Z" fill="#596387"/> </svg>
    <span class="link-name">Electricity usage</span>
  </a></li>
</ul>
</div>

<div class="add-button-menu">
  <a class="body_text add-room-button">Add room</a>
  <a class="body_text add-device-button">Add device</a>
</div>

<div class="exit-add">
  <ul class="logout">
    <li><a href="/handlers/logout.php">
      
    </a></li>
  </ul>

  <button class="add_button">  </button>
</div>
</nav>

<section class="dashboard">
  <div class="top_info">
    <div class="menu_icon"></div>
    <div class="time_temp">
      <span class="body_text">14 °C</span>
      <span class="body_text">|</span>
      <span class="body_text">14:20</span>
      <button class="add_button">  </button>
    </div>
  </div>
  <div class="dash_content">
    <div class="title title_margin">
      <span class="h2"><?php echo $room["name"] ?> stats</span>
      <button class="dot-menu-room"></button>
      <a class="room-menu body_text" href="/handlers/delete_room.php?id=<?php echo
$room["name"]?>">Delete room</a>
    </div>
    <div class="boxes">
      <div class="box temperature_box first-box">
        <span class="h3">Room temperature</span>
        <hr noshade class="dash">
        <div class="regulation">
          <?php if(isset($stermostatId)){
            if(isset($condition)) { ?>
              <span class="body_text box_text"><?php echo $roomStats["temperature"];?> °C</span>
              <?php if($condition["isActive"]==1 && $conditionStats["temperature"] >
                $roomStats["temperature"]){ ?>
                
                <span class="small-text">Up to <?php echo $conditionStats["temperature"]?> °C</span>
              <?php }

```

```

        if($condition["isActive"]==1      &&      $conditionStats["temperature"]      <
$roomStats["temperature"]){ ?>

        
        <span class="small-text">Down to <?php echo $conditionStats["temperature"]?>
°C</span>

        <?php } }
        else{ ?>
        <span class="body_text box_text"><?php echo $roomStats["temperature"];?> °C</span>
        <?php } }
        else{ ?>
        <span class="small-text">Connect <br> termometer</span>
        <?php } ?>
    </div>
</div>
<div class="box humidity_box">
    <span class="h3">Air humidity</span>
    <hr noshade class="dash">
    <div class="regulation">
        <?php if(isset($termostatId)){
            if(isset($humidifierId)) { ?>
                <span class="body_text box_text"><?php echo $roomStats["humidity"];?> %</span>
                <?php if($humidifierId["isActive"]==1) {?>
                    
                <?php } } }
                else{ ?>
                <span class="small-text">Connect <br> hygrometer</span>
                <?php } ?>
            </div>
        </div>
    <div class="box electricity_box last-box">
        <span class="h3">Electricity usage</span>
        <hr noshade class="dash">
        <div class="regulation">
            <?php if(isset($counter)){
                if(isset($counterStats)){ ?>
                    <span class="body_text box_text"><?php echo $counterStats["electricity"];?> kW</span>
                <?php } }
                else{ ?>
                <span class="small-text">Connect <br> counter</span>
                <?php } ?>
            </div>
        </div>
    </div>
</div>

<div class="title title_margin ">
    <span class="h2">Devices</span>
</div>

<div class="boxes room-devices">
    <?php

    while($row = mysqli_fetch_assoc($devicesArray)){

    ?>
    <div class="device-box first-box">
        <div class="device-info">
            <div class="device-name-room">
                <?php $undertype = get_smth_by_smth_row("name", "undertypes", "id",
$room["undertype"])?>
                <a class="device-name" href="<?php echo $undertype["name"]?>.php?id=<?php echo
$room["id"]?>"><?php echo $row["name"]?> </a>

```



```

        <span class="small-text device-room"><?php
        $room_dev_name = get_smth_by_smth_row("name", "rooms", "id", $row["roomId"]);
        echo $room_dev_name["name"];
        ?></span>
    </div>

    <div class="device-regulation">
        <?php
        if($row["isActive"] == 0){
            ?>
            <a class="device-button" href="/handlers/activate_device.php?id=?php echo
            $row["id"]?>&from=roomPage">  </a>
            <span class="medium-text device-action">Off</span>
            <?php }
            else{
                ?>
                <a class="device-button active" href="/handlers/deactivate_device.php?id=?php echo
                $row["id"]?>&from=roomPage">  </a>
                <span class="medium-text device-action">ON</span>
                <?php } ?>
            </div>
        </div>
        <a href=?php echo $undertype["name"]?>.php?id=?php echo $row["id"]?>">
        <?php
        $deviceImg = get_device_image($row["undertype"], $row["brand"]);

        echo ';
        ?>
        </a>
    </div>
    <?php
    }
    ?>
</div>

</section>
</div>
<script type="text/javascript" src="/scripts/main.js"></script>

</body>
</html>

```

login.php

```

<!-- Обробка авторизації -->

<?php
session_start();
include("connection.php");
include("functions.php");

$error = null;

if(isset($_POST['submit'])){
    $username = $_POST['user'];
    $password = $_POST['password'];

    $sql = "select * from users where username = '$username' and password = '$password'";
    $result = mysqli_query($conn, $sql);
    $row = mysqli_fetch_array($result, MYSQLI_ASSOC);
    $count = mysqli_num_rows($result);

```

```

if($count==1){
    $home_id = find_home_id_by_username($username);

    $_SESSION['username'] = $_POST['user'];
    $_SESSION['homeId'] = $home_id;
    header("Location:../Dashboard.php");
}

else{

    if($username == "" || $password == ""){
        $error = "Fields are empty";

        setcookie('errors', $error, 0, '/');
        header('location:../Login_form.php');
    }

    else{
        $error = "Invalid name or password";

        setcookie('errors', $error, 0, '/');
        header("Location:../Login_form.php");
    }
}
}
?>

```

add_device.php

<!-- Сторінка додавання нового пристрою -->

```

<?php
session_start();

include("connection.php"); //Під'єдную файл зі з'єднанням до БД
include("functions.php"); //Під'єдную файл з функціями

$error = null;

//Отримую значення, передані з форми
if(isset($_POST['submit'])){
    $name = $_POST['name'];
    $undertype = $_POST['undertype'];
    $roomId = $_POST['roomId'];
    $brand = $_POST['brand'];

    $typeId = get_smth_by_smth_row("*, "undertypes", "id", $undertype);
    $typeName = get_smth_by_smth_row("name", "types", "id", $typeId["typeId"]);

    $brandName = get_smth_by_smth_row("*, "brands", "id", $brand);

    $type = $typeId["typeId"];

    //Якщо поля пусті
    if ($type == "" || $undertype == "" || $brand == ""){
        $error = 'Fields are empty';

        setcookie('errors', $error, 0, '/');
        header('location:../Dashboard.php');
    }

    //Якщо поля заповнені
    else{

```

```

//Якщо назва пристрою не заповнена, замість неї буде відобразитись тип та назва виробника
if($name == ""){
    $name = "{TypeId["name"]} {BrandName["name"]}";
}

//Додавання девайсу до БД
$insert = "insert into devices(name, type, undertype, brand, isActive, roomId) values('$name', '$type',
'$undertype', '$brand', 0, '$roomId')";
mysqli_query($conn, $insert);
header('location:../Dashboard.php');
}
}

```

?>

functuons.php

<!-- Функції взаємодії клієнта з сервером -->

<?php

```

include("connection.php");
require 'connection.php';

```

```

//Отримати номер дому за нікнеймом користувача
function find_home_id_by_username(string $username){
    global $conn;

```

```

    $findHomeId = "select homeId from users where username = '$username'";
    $get_homeId = $conn->query($findHomeId);
    $row = mysqli_fetch_assoc($get_homeId);
    return $row["homeId"];
}

```

```

//Отримати всі значення з таблиці
function get_all_from_table(string $tablename){
    global $conn;

```

```

    $get_all_from_table = "select * from $tablename";
    $get_all = $conn->query($get_all_from_table);
    return $get_all;
}

```

```

//Отримати список кімнат з певного будинку (Для виведення у масиві)
function get_rooms_by_homeId(string $home_id){
    global $conn;

```

```

    $sql = "select * from rooms where homeId = $home_id";
    $get_rooms = $conn->query($sql);
    return $get_rooms;
}

```

```

//Отримати будь-яке значення з будь-якої таблиці (Для виведення у масиві)
function get_smth_by_smth(string $selectWhat, string $dbname, string $whereWhat , string $whatValue){
    global $conn;

```

```

    $sql = "select $selectWhat from $dbname where $whereWhat = '$whatValue'";
    $get_smth = $conn->query($sql);
    return $get_smth;
}

```

```

//Отримати будь-яке значення з будь-якої таблиці за 1 параметром (Для виведення одного значення за раз)
function get_smth_by_smth_row(string $selectWhat, string $dbname, string $whereWhat , string $whatValue){
    global $conn;

```

```

    $sql = "select $selectWhat from $dbname where $whereWhat = '$whatValue'";
    $get_smth = $conn->query($sql);
    $row = mysqli_fetch_assoc($get_smth);
    return $row;
}

//Отримати будь-яке значення з будь-якої таблиці за 2 параметрами (Для виведення одного значення за раз)
function get_smth_by_2smth_row(string $selectWhat, string $dbname, string $whereWhat , string $whatValue, string $whereWhat2 , string $whatValue2){
    global $conn;

    $sql = "select $selectWhat from $dbname where $whereWhat = '$whatValue' and $whereWhat2 = '$whatValue2'";
    $get_smth = $conn->query($sql);
    $row = mysqli_fetch_assoc($get_smth);
    return $row;
}

//Отримати будь-яке значення з будь-якої таблиці за 2 параметрами (Для виведення у масиві)
function get_smth_by_2smth(string $selectWhat, string $dbname, string $whereWhat , string $whatValue, string $whereWhat2 , string $whatValue2){
    global $conn;

    $sql = "select $selectWhat from $dbname where $whereWhat = '$whatValue' and $whereWhat2 = '$whatValue2'";
    $get_smth = $conn->query($sql);
    return $get_smth;
}

//Отримати кількість значень з будь-якої таблиці за 2 параметрами
function get_smth_by_2smth_count(string $selectWhat, string $dbname, string $whereWhat , string $whatValue, string $whereWhat2 , string $whatValue2){
    global $conn;

    $sql = "select $selectWhat from $dbname where $whereWhat = '$whatValue' and $whereWhat2 = '$whatValue2'";
    $get_smth = $conn->query($sql);
    $count = mysqli_num_rows($get_smth);
    return $count;
}

//Отримати кількість значень з будь-якої таблиці за 3 параметрами
function get_smth_by_3smth_count(string $selectWhat, string $dbname, string $whereWhat , string $whatValue, string $whereWhat2 , string $whatValue2, string $whereWhat3 , string $whatValue3){
    global $conn;

    $sql = "select $selectWhat from $dbname where $whereWhat = '$whatValue' and $whereWhat2 = '$whatValue2' and $whereWhat3 = '$whatValue3'";
    $get_smth = $conn->query($sql);
    $count = mysqli_num_rows($get_smth);
    return $count;
}

//Отримати кількість значень з будь-якої таблиці за 1 параметром
function get_smth_by_smth_count(string $selectWhat, string $dbname, string $whereWhat , string $whatValue){
    global $conn;

    $sql = "select $selectWhat from $dbname where $whereWhat = '$whatValue'";
    $get_smth = $conn->query($sql);
    $count = mysqli_num_rows($get_smth);
    return $count;
}

```

```

}

//Отримати будь-яке значення з таблиці
function select_smth_from_db(string $selectWhat, string $dbname){
    global $conn;

    $sql = "select $selectWhat from $dbname";
    $get_smth = $conn->query($sql);
    return $get_smth;
}

//Отримати картинку девайсу
function get_device_image(string $undertype, string $brand){
    global $conn;

    $sql = "select image from deviceimages where undertype = '$undertype' and brand = '$brand'";
    $get_smth = $conn->query($sql);
    $row = mysqli_fetch_assoc($get_smth);
    return $row;
}

//Контроль системи (вентиляція/звук/світло)
function system_control(int $id, string $action, int $type){
    global $conn;

    if($action == 'activate'){
        $devices = get_smth_by_2smth("*", "devices", "type", $type, "roomId", $id);

        while($row = mysqli_fetch_assoc($devices)){
            $deviceId = $row["id"];
            $sql = "UPDATE `devices` SET `isActive` = '1' WHERE `devices`.`id` = $deviceId;";
            $conn->query($sql);
        }
    }

    if($action == 'deactivate'){
        $devices = get_smth_by_2smth("*", "devices", "type", $type, "roomId", $id);

        while($row = mysqli_fetch_assoc($devices)){
            $deviceId = $row["id"];
            $sql = "UPDATE `devices` SET `isActive` = '0' WHERE `devices`.`id` = $deviceId;";
            $conn->query($sql);
        }
    }
}

//Зміна температури у будинку
function update_home_temp(int $homeTemp, int $home_id){
    global $conn;

    $sql = "UPDATE `homes` SET `temperature` = '$homeTemp' WHERE `homes`.`id` = $home_id;";
    $conn->query($sql);
}

//Зміна температури у всіх девайсах в домі
function update_devices_temp($tempDevices, int $homeTemp, int $activateCond){
    global $conn;
    if($activateCond==1)
    {
        while($row = mysqli_fetch_assoc($tempDevices)){
            $deviceId = $row["id"];
            $sql = "UPDATE `devices` SET `isActive` = '1' WHERE `devices`.`id` = $deviceId;";
            $conn->query($sql);
        }
    }
}

```

```

        if($row["undertype"]==2){
            $sql = "UPDATE `conditioners` SET `temperature` = '$homeTemp' WHERE `conditioners`.`deviceId`
= $deviceId;";
            $conn->query($sql);
        }
    }
}

else{
    while($row = mysqli_fetch_assoc($tempDevices)){
        $deviceId = $row["id"];
        $sql = "UPDATE `devices` SET `isActive` = '0' WHERE `devices`.`id` = $deviceId;";
        $conn->query($sql);
    }
}
}
?>

```

global_temp.php

```

<!-- Налаштування температури в домі -->

<?php
session_start();

$home_id = $_SESSION['homeId'];
$user_name = $_SESSION['username'];

include("connection.php");
include("functions.php");

$error = null;

$action = $_GET['action'];
$from = $_GET['from'];

$tempDevices = get_smth_by_smth("*", "devices", "type", 2);
$conditioners = get_smth_by_smth("*", "devices", "undertype", 2);

$home = get_smth_by_smth_row("*", "homes", "id", $home_id);
$homeTemp = $home["temperature"];

if($action == 'activate'){

    update_devices_temp($tempDevices, $homeTemp, 1);
}

else if($action == 'deactivate'){

    update_devices_temp($tempDevices, $homeTemp, 0);
}

else if($action == 'plus'){
    $homeTemp = $homeTemp+1;

    update_home_temp($homeTemp, $home_id);

    update_devices_temp($tempDevices, $homeTemp, 1);
}

else if($action == 'minus'){
    $homeTemp = $homeTemp-1;
}

```

```

    update_home_temp($homeTemp, $home_id);

    update_devices_temp($tempDevices, $homeTemp, 1);
}

if($from=='dashboard'){
    header("Location:../Dashboard.php");
}

if($from=='tempPage'){
    header("Location:../Temperature_page.php");
}

?>

```

global_hum.php

```

<!-- Налаштування вологості в домі -->

<?php
session_start();

$home_id = $_SESSION['homeId'];
$user_name = $_SESSION['username'];

include("connection.php");
include("functions.php");

$error = null;

$action = $_GET['action'];
$from = $_GET['from'];

$humDevices = get_smth_by_smth("*", "devices", "type", 4);

$home = get_smth_by_smth_row("*", "homes", "id", $home_id);
$homeHum = $home["humidity"];

if($action == 'activate'){

    while($row = mysqli_fetch_assoc($humDevices)){
        $deviceId = $row["id"];
        $sql = "UPDATE `devices` SET `isActive` = '1' WHERE `devices`.`id` = $deviceId;";
        $conn->query($sql);
    }
}

else if($action == 'deactivate'){

    while($row = mysqli_fetch_assoc($humDevices)){
        $deviceId = $row["id"];
        $sql = "UPDATE `devices` SET `isActive` = '0' WHERE `devices`.`id` = $deviceId;";
        $conn->query($sql);
    }
}

else if($action == 'plus'){
    if($homeHum < 100)
        $homeHum = $homeHum + 1;

    $sql = "UPDATE `homes` SET `humidity` = '$homeHum' WHERE `homes`.`id` = $home_id;";
    $conn->query($sql);
}

```

```

while($row = mysqli_fetch_assoc($humDevices)){
    $deviceId = $row["id"];
    $sql = "UPDATE `devices` SET `isActive` = '1' WHERE `devices`.`id` = $deviceId;";
    $conn->query($sql);
}
}

else if($action == 'minus'){
    if($homeHum>0)
        $homeHum = $homeHum-1;

    $sql = "UPDATE `homes` SET `humidity` = '$homeHum' WHERE `homes`.`id` = $home_id;";
    $conn->query($sql);

    while($row = mysqli_fetch_assoc($humDevices)){
        $deviceId = $row["id"];
        $sql = "UPDATE `devices` SET `isActive` = '1' WHERE `devices`.`id` = $deviceId;";
        $conn->query($sql);
    }
}

if($from=='dashboard'){
    header("Location:../Dashboard.php");
}

else if($from=='humPage'){
    header("Location:../Humidity_page.php");
}
}

```

system_control.php

```

<!-- Обработка значений систем -->

<?php
session_start();

include("connection.php");
include("functions.php");

$error = null;

$id = $_GET['id'];
$system = $_GET['system'];
$action = $_GET['action'];

if($system=='vent'){
    system_control($id, $action, 2);
}

if($system=='sound'){
    system_control($id, $action, 6);
}

if($system=='light'){
    system_control($id, $action, 7);
}

header("Location:../Dashboard.php");

?>

```


temp_regulation.php

```
<!-- Регулювання температури -->

<?php
session_start();

include("connection.php");
include("functions.php");

$error = null;

$id = $_GET['id'];
$action = $_GET['action'];

$stmts = get_smth_by_smth_row("*", "conditioners", "deviceId", $id);
$device = get_smth_by_smth_row("*", "devices", "id", $id);

$temp = $stmts["temperature"];

if($action=="minus")
{
    $temp = $temp-1;
}

if($action=="plus")
{
    $temp = $temp+1;
}

if($device["isActive"] == "0"){
    $sql = "UPDATE `devices` SET `isActive` = '1' WHERE `devices`.`id` = $id;";
    $result = $conn->query($sql);
}

$sql = "UPDATE `conditioners` SET `temperature` = '$temp' WHERE `conditioners`.`deviceId` = $id;";
$result1 = $conn->query($sql);

if ($result1) {
    header("Location:../Conditioner.php?id={$id}");
}

?>
```

ВІДГУК КЕРІВНИКА ЕКОНОМІЧНОГО РОЗДІЛУ

ПЕРЕЛІК ДОКУМЕНТІВ НА ОПТИЧНОМУ НОСІЇ

| Ім'я файлу | Опис |
|------------------------|--|
| Пояснювальні документи | |
| ПЗ_Бріжак.doc | Пояснювальна записка до кваліфікаційної роботи. Документ Word. |
| ПЗ_Бріжак.pdf | Пояснювальна записка до кваліфікаційної роботи в форматі PDF |
| Програма | |
| SmartHome.rar | Архів. Містить коди програми і откомпільовану програму |
| Презентація | |
| Презентація_Бріжак.ppt | Презентація кваліфікаційної роботи |