

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕЛЕКТРОНИКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК
СЕКЦІЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРОЕКТУВАННЯ

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: «Інформаційна система моніторингу показників стану повітря в приміщенні»

за спеціальністю 122 «Комп'ютерні науки»,
освітньо-професійна програма «Інформаційні технології проектування»

Виконавець роботи: студент групи ІТ-61 Палажченко Євген Володимирович

**Кваліфікаційна робота бакалавра
захищена на засіданні ЕК
з оцінкою**

_____ «__» _____ 2020 р.

Науковий керівник

(підпис)

к.т.н., доц., Ващенко С. М.

(науковий ступінь, вчене звання, прізвище та ініціали)

Голова комісії

(підпис)

Шифрін Д. М.

(науковий ступінь, вчене звання, прізвище та ініціали)

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів
без відповідних посилань.

Студент _____
(підпис)

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук
Секція інформаційних технологій проектування
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ

Зав. секцією ІТП

В. В. Шендрик

«___»_____2020 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ**

Палажченко Євген Володимирович

1 Тема роботи Інформаційна система моніторингу показників стану повітря в приміщенні

керівник роботи Ващенко Світлана Михайлівна, к.т.н., доцент,

затверджені наказом по університету від « 17 » травня 2020 р. № 0834-III

2 Строк подання студентом роботи «1» червня 2020 р.

3 Вхідні дані до роботи Показання датчиків, дані про періоди роботи приладів

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1) Аналіз предметної області

2) Проектування інформаційної системи

3) Розробка інформаційної системи

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Актуальність роботи, мета і задачі, порівняння систем-аналогів, функціональні вимоги, моделювання роботи системи, діаграма варіантів використання, ER діаграма, засоби реалізації, архітектура інформаційної системи, розробка теми, використання можливостей WordPress, демонстрація інформаційної системи, висновки, оприлюднення результатів.

6. Консультанти розділів роботи:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Дослідження предметної області	До 18.03.2020	
2.	Постановка задачі	До 01.04.2020	
3.	Вибір засобів реалізації	До 10.04.2020	
4.	Планування робіт	До 24.04.2020	
5.	Проектування роботи інформаційної системи	До 25.04.2020	
6.	Практична реалізація системи	До 25.05.2020	
7.	Оформлення документації	До 31.05.2020	

Студент

(підпис)

Палажченко Є.В.

Керівник роботи

(підпис)

к.т.н., доц., Ващенко С.М.

РЕФЕРАТ

Тема бакалаврської роботи: «Інформаційна система моніторингу показників стану повітря в приміщенні».

Кваліфікаційна робота бакалавра присвячена розробці інформаційної системи моніторингу показників стану повітря в приміщенні.

Перед початком розробки інформаційної системи було проведено огляд останніх досліджень і публікацій, аналіз програмних продуктів-аналогів та постановку задачі. Також було розроблено технічне завдання на виконання роботи та проведено планування робіт.

Результатом проектування роботи інформаційної системи є діаграми процесів інформаційної системи у нотації IDEF0, діаграми варіантів використання системи. Для проектування структури бази даних було створено діаграму «сутність-зв'язок» та концептуальну модель даних.

Результатом виконаної роботи є інформаційна система, яка має функціонал для надання послуг моніторингу показників стану повітря в приміщенні та візуалізації зібраних даних.

Інформаційна система була розроблена у вигляді теми, розробленої під систему керування контентом WordPress.

Практичне значення роботи полягає в тому, що система дозволяє централізовано надавати послуги моніторингу показників стану повітря в приміщенні великій кількості користувачів.

Пояснювальна записка містить вступ, 3 розділи, висновки, додатки та список літератури з 20 найменувань. Загальний обсяг роботи – 79 сторінок: 40 сторінок основного тексту, 2 сторінки списку використаних джерел, 37 сторінок додатків.

Ключові слова: ІНФОРМАЦІЙНА СИСТЕМА, МОНІТОРИНГ ПОКАЗНИКІВ СТАНУ ПОВІТРЯ, WORDPRESS.

ЗМІСТ

ВСТУП.....	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	8
1.1 Огляд останніх досліджень і публікацій.....	8
1.2 Аналіз програмних продуктів – аналогів.....	9
1.3 Постановка задачі.....	10
2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	13
2.1 Діаграми нотації IDEF0	13
2.2 Use-case діаграма.....	16
2.3 Проектування моделі бази даних	18
3 РОЗРОБКА ІНФОРМАЦІЙНО СИСТЕМИ	22
3.1 Архітектура інформаційної системи	22
3.2 Програмна реалізація	23
3.3 Використання програмного додатку	32
ВИСНОВКИ	40
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	41
Додаток А	43
Додаток Б.....	52
Додаток В	60
Додаток Г.....	64

ВСТУП

Потреба в системах моніторингу показників стану повітря існує в багатьох аспектах життя та праці людини. Для комфортного проживання та відпочинку вдома та гарантування оптимальних умов праці людини на робочому місці необхідно підтримувати температуру, рівень вологості та CO₂ в межах встановленої норми. В промисловості та сільському господарстві відстеження та регулювання показників стану повітря необхідне для більш тривалого зберігання товарів на складах або для забезпечення оптимальних умов дозрівання овочів та фруктів у теплицях і рослин у ботанічних садах. Також для багатьох, особливо екзотичних, тварин необхідно створювати мікроклімат всередині вольєру або тераріуму.

Актуальність проекту полягає у створенні інструменту керування процесом надання послуг моніторингу показників стану повітря в приміщенні та візуалізації зібраних даних, що дасть змогу полегшити сприйняття та аналіз отриманої інформації.

Об'єктом роботи є інформаційна підтримка процесу моніторингу показників стану повітря в приміщенні.

Предметом дослідження є системи та інструменти візуалізації показників стану повітря в приміщенні.

Метою даної роботи є створення системи візуалізації показників стану повітря в приміщенні.

Для досягнення поставленої мети було визначено **основні задачі**:

- проаналізувати програмні продукти–аналоги;
- здійснити огляд останніх досліджень і публікацій;
- сформулювати технічне завдання;
- провести планування етапів роботи;
- спроектувати інформаційну систему;
- розробити систему.

Результат дипломного проекту – інформаційна система моніторингу показників стану повітря в приміщенні. Практичне значення системи полягає у можливості застосування у промисловості, сільському господарстві та приватному використанні для моніторингу показників стану повітря в приміщенні.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд останніх досліджень і публікацій

Бажання автоматизувати рутинну роботу та зробити життя більш комфортним привело до створення різноманітних систем домашньої автоматизації. Такі системи спрощують, автоматизують та централізують управління освітленням, опаленням, кондиціонуванням, системою безпеки. Системи безпеки повідомляють користувача про проникнення злочинців, протікання води та про високий рівень диму або чадного газу.

Технології домашньої автоматизації розвиваються у декількох напрямках.

Інтернет речей — це мережа, що складається з фізичних об'єктів, які можуть здійснювати взаємодію з оточуючим середовищем, приймати дані ззовні, передавати іншим пристроям відомості про свій поточний стан [1].

Розумний дім або розумний будинок — це будь-яке жиле приміщення або приміщення комерційного призначення, які мають «розумні» системи забезпечення та систему multi-room. Використовуючи систему multi-room можна пов'язати між собою усі електроприлади будівлі, що дає можливість централізованого керувати. Усі прилади можуть бути під'єднані до комп'ютерної мережі, що надає можливість віддаленого керування всією системою або керування за допомогою персонального комп'ютера [2].

Під поняттям «розумний дім» необхідно розуміти централізовану систему управління всіма приладами та підсистемами в будинку. Це виключає необхідність наявності окремих панелей керування системою безпеки, водопостачання, опалення, кондиціонування та іншими. Така система має відслідковувати на будь-які внутрішні та зовнішні зміни, що можуть порушити комфортне перебування вдома, та відповідним чином реагувати на них [3].

Такі системи дозволяють економити кошти, підтримуючи оптимально-комфортні умови для перебування в приміщенні. Якщо ж господарів немає вдома, то систему можна перевести в режим енергозбереження.

Система самостійно збирає інформацію про показники стану повітря в кожній кімнаті окремо і підтримує ці показники на заданому рівні. Регуляція показників стану повітря здійснюється за рахунок керування заслінками системи вентиляції та радіаторів [4].

Після повернення господарів будинку з довготривалої подорожі «розумний будинок» може представити звіт про всі події, що сталися в ньому за час їх відсутності. Всі дії тих хто заходив до будинку буде збережено в архіві систем [5].

1.2 Аналіз програмних продуктів – аналогів

На даний момент основними конкурентами даної системи на ринку є системи «розумного будинку» та «розумної системи безпеки». «Розумні» системи пропонують декілька великих виробників, серед яких Xiaomi (Mi Smart Home Security Kit), Zipato (Zipato Zipatile), AJAX (Ajax StarterKit Plus) Nomi (Nomi Smart Home), TRUMP Electronics (DOMINTELL) та інші.

Основна частина таких систем орієнтована на клієнта business-to-consumer (B2C), які можуть регулювати показники повітря і не тільки. Але не існує системи орієнтованої як на клієнта (B2C) так і на бізнес business to business (B2B).

Оскільки сайти вищезазначених систем [6-10], є не достатньо інформативними, було проведено опитування співробітників служби підтримки щодо функціональних можливостей їх систем.

На основі проведеного опитування було створено порівняльну таблицю з зазначенням переваг та недоліків продуктів аналогів.

Порівняльну характеристику систем-аналогів наведено в таблиці 1.1.

Таблиця 1.1 — Порівняльна характеристика систем-аналогів

№	назва системи характеристика	Mi Smart Home Security Kit	Zipato Zipatile	Ajax StarterKit Plus	Nomi Smart Home	Domintell
1	Безпроводний протокол	+	+	+	+	–
2	Управління зі смартфону	+	+	+	+	+
3	Віддалене управління системою	+	+	+	+	+
4	Можливість розширення системи	+	–+	+	–	+
5	Допустимий радіус розстановки датчиків	75	15	2000	70	∞
6	Самодостатня система	+	+	+	+	+
7	Відкрита система	+	+	–	–	+
8	Фіксований початковий набір датчиків	–	+	–	+	–
9	Можливість застосування в промисловості	–	–	+	–	+
10	Максимально можлива кількість пристроїв	100	200	200	8	600

1.3 Постановка задачі

Інформаційна система повинна бути реалізована як сайт, що є інструментом керування процесом надання послуг моніторингу показників стану повітря в приміщенні. Даний підхід дозволяє централізовано обслуговувати велику кількість користувачів. Користувач може увійти в систему з будь-якого місця та будь-якого девайсу, де є доступ до мережі Internet.

Користувач, що не авторизований в системі повинен мати доступ до головної сторінки, вікна авторизації та вікна подачі заявки на реєстрацію в системі.

Повідомлення про нову заявку на реєстрацію має надсилатися на електронну адресу адміністратора системи.

Авторизований користувач повинен мати доступ до свого особистого кабінету. На сторінці особистого кабінету користувачу має бути надана можливість переглядати список всіх доступних йому пристроїв, групувати та редагувати свої пристрої. Також користувач повинен мати можливість перейти до сторінки перегляду даних, зібраних обраним пристроєм. Дані мають відображатися у графічному вигляді. Передбачити можливість перегляду архівних даних за попередні періоди.

Можливість реєструвати нових користувачів у системі повинен мати лише адміністратор. Керувати типами датчиків та одиницями вимірювання кожного типу повинен також тільки адміністратор. Необхідно передбачити можливість переходу зі сторінки перегляду списку користувачів особистого кабінету обраного користувача. Після додавання нового пристрою в систему, він має відображатися у списку «Вільні модулі» – такі, що не прив'язані ні до одного з користувачів. Дозволити адміністратору редагувати даний список модулів та список модулів будь-якого користувача.

Схему інформаційної системи моніторингу показників стану повітря в приміщенні наведено на рисунку 1.3.



Рисунок 1.3 — Схема роботи інформаційної системи

Технічне завдання на розробку інформаційної системи наведено в додатку А.

Було прийнято рішення створити інформаційну систему на основі CMS (Content Management System) WordPress. У WordPress вже реалізовано системи реєстрації, авторизації та розділення прав користувачів. Базу даних даної системи керування контентом легко доповнювати та керувати нею через вбудовані інструменти. Крім того, під WordPress існує велика кількість плагінів для вирішення різного плану задач.

Для візуалізації контенту в WordPress використовуються теми. WordPress тема створюється на мові програмування PHP з використанням функцій та класів WordPress. Для розмітки та стилізації сторінок будуть використані HTML та CSS. З метою полегшення роботи при верстці сторінок та для забезпечення адаптивності під різні пристрої буде використано Bootstrap 4. Для прискорення сприйняття інформації та покращення дизайну сайту буде використано набір web-іконок Font Awesome.

Для візуалізації зібраних даних було прийнято рішення використовувати Google Line Chart оскільки даний інструмент має стильний дизайн, є безкоштовним, його легко налаштовувати, та з ним зручно працювати.

Виходячи з встановлених вимог та обраних засобів реалізації було проведено планування робіт. Результати виконаного планування робіт наведено в додатку Б.

2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Діаграми нотації IDEF0

Для графічного опису процесів інформаційної системи моніторингу показників стану повітря в приміщенні використовуємо методологію IDEF0. Вона строго описує бізнес-процеси інформаційної системи, що сприяє розумінню принципів функціонування та взаємодії з навколишнім середовищем для представленої інформаційної системи [11].

Структурно-функціональна модель була створена з використанням програмного додатку draw.іо.

Контексту діаграму основного процесу інформаційної системи зображено на рисунку 2.1.



Рисунок 2.1 — Контекстна діаграма

Контекстна діаграма складається з одного блоку «Моніторинг та регулювання показників стану повітря в приміщенні», що є основним і максимально загальним процесом інформаційної системи, що моделюється. Вхідна стрілка «Запит на виконання роботи» є вхідними даними для перетворення системою. Вихідними даними для системи є «Графічно візуалізовані дані», «Архівні дані» і «Поточні дані». Стрілки «Інформаційна система smartbuild» та «Користувач» є механізмами для перетворення вхідних даних і входять в блок знизу. Стрілки «Граничні значення показників стану повітря» і «дані з бази даних» є інструментами для перетворення вхідних даних і входять в блок зверху.

Для кращого розуміння процесів, що відбуваються в середині системи, контекстну діаграму потрібно деталізувати.

Деталізація основного процесу наведено на рисунку 2.2.

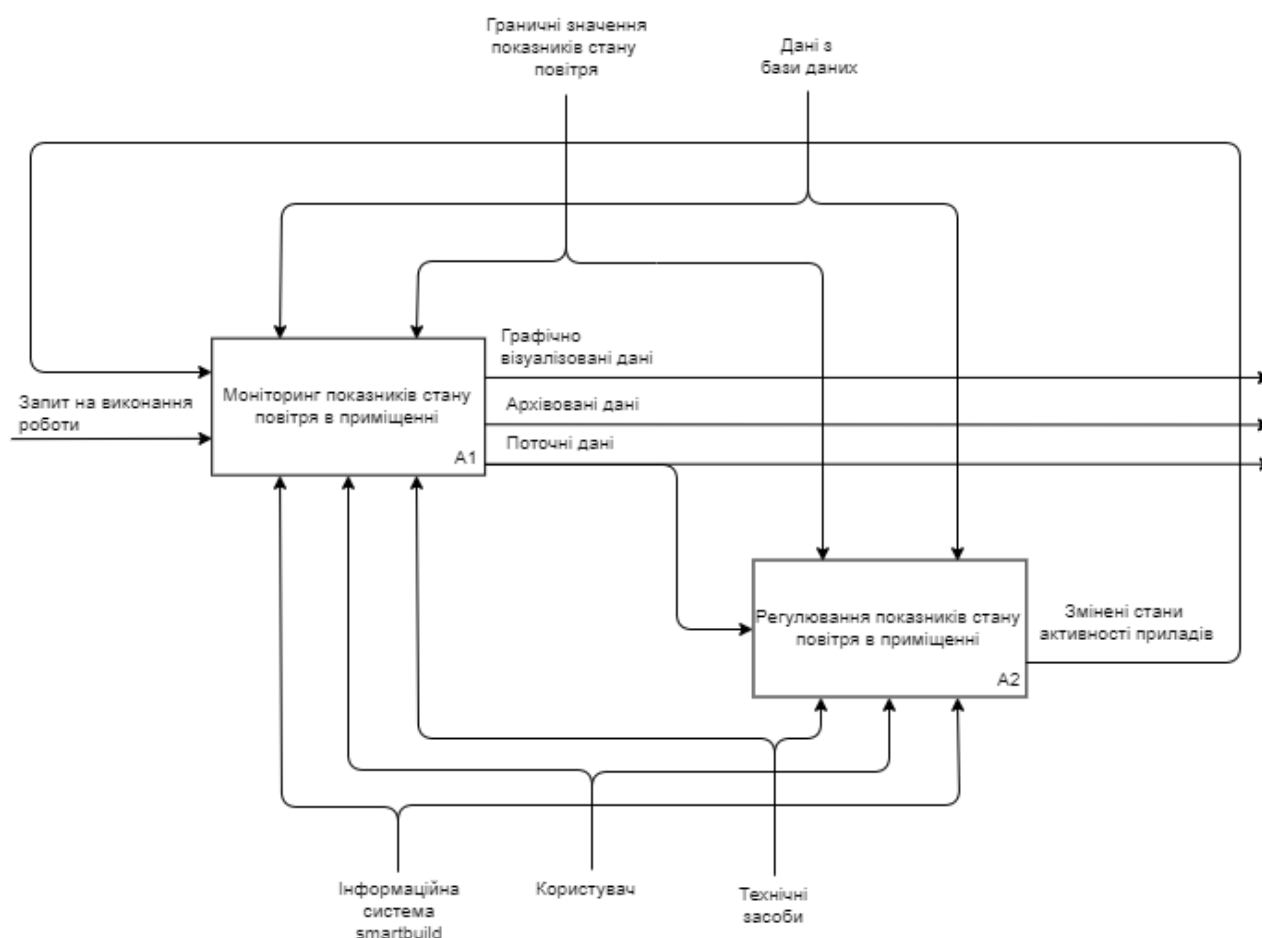


Рисунок 2.2 — Декомпозиція першого рівня

В діаграмі декомпозиції першого рівня виділено два процеси: «Моніторинг показників стану повітря в приміщенні» та «Регулювання показників стану повітря в приміщенні». Вхідними даними для блоку A2 є поточні дані, які є вихідними даними для блоку A1. На виході з блоку A2 отримуємо змінні стани активності приладів.

Перший рівень декомпозиції не може продемонструвати як відбувається процес регулювання показників стану повітря, тому було прийнято рішення деталізувати блок A2.

Деталізацію процесу регулювання показників стану повітря в приміщенні наведено на рисунку 2.3.

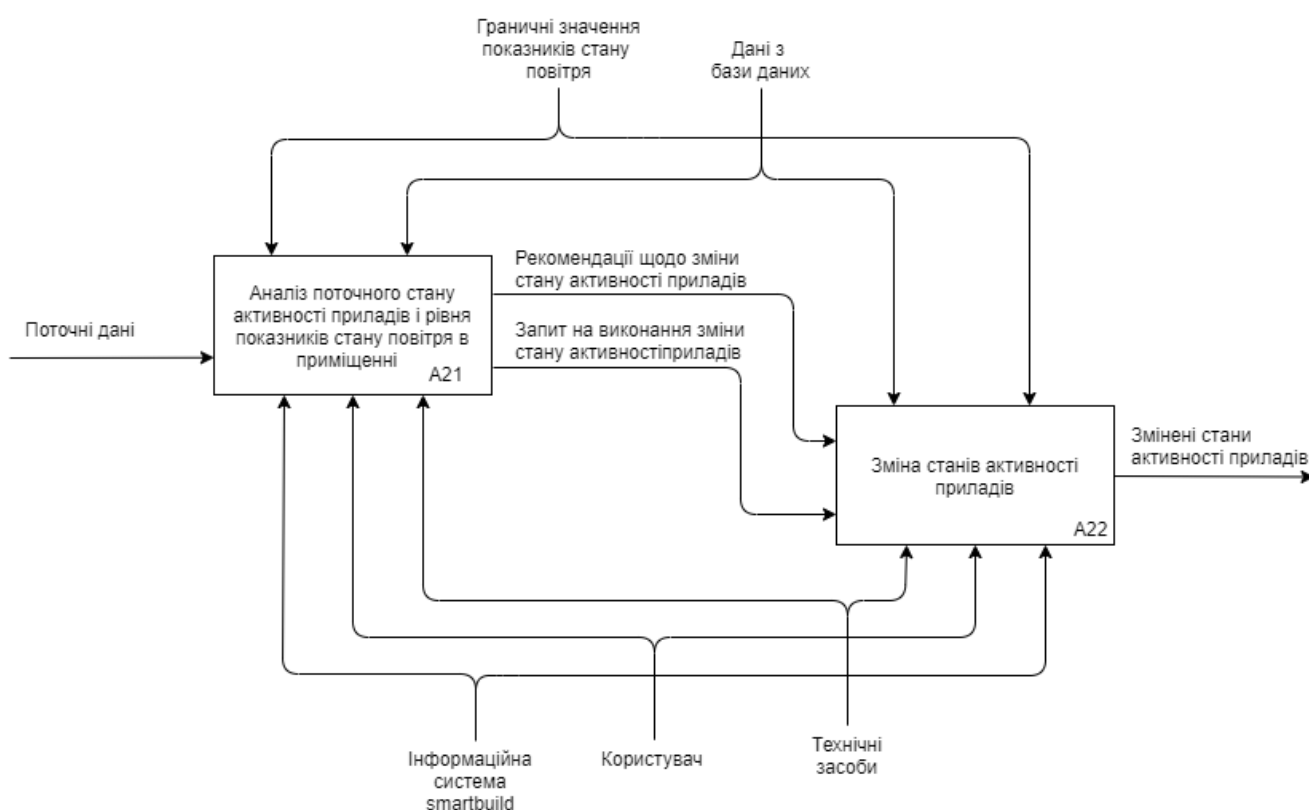


Рисунок 2.3 — Деталізація процесу регулювання

В діаграмі декомпозиції блоку A2 виділено два процеси: «Аналіз поточного стану активності приладів і рівня показників стану повітря в приміщенні» та «Зміна станів активності приладів». Результатом проведеного аналізу є рекомендації щодо зміни стану активності приладів і запит на виконання стану активності приладів. Вихідними даними процесу A22 є змінні стани активності приладів.

2.2 Use-case діаграма

Діаграма сценаріїв використання описує поведінку системи у відповідь на зовнішні запити користувача. Іншими словами, різновид використання описує, хто і що може зробити з розглянутою системою [12].

Діаграму сценаріїв використання для користувача наведено на рисунку 2.4

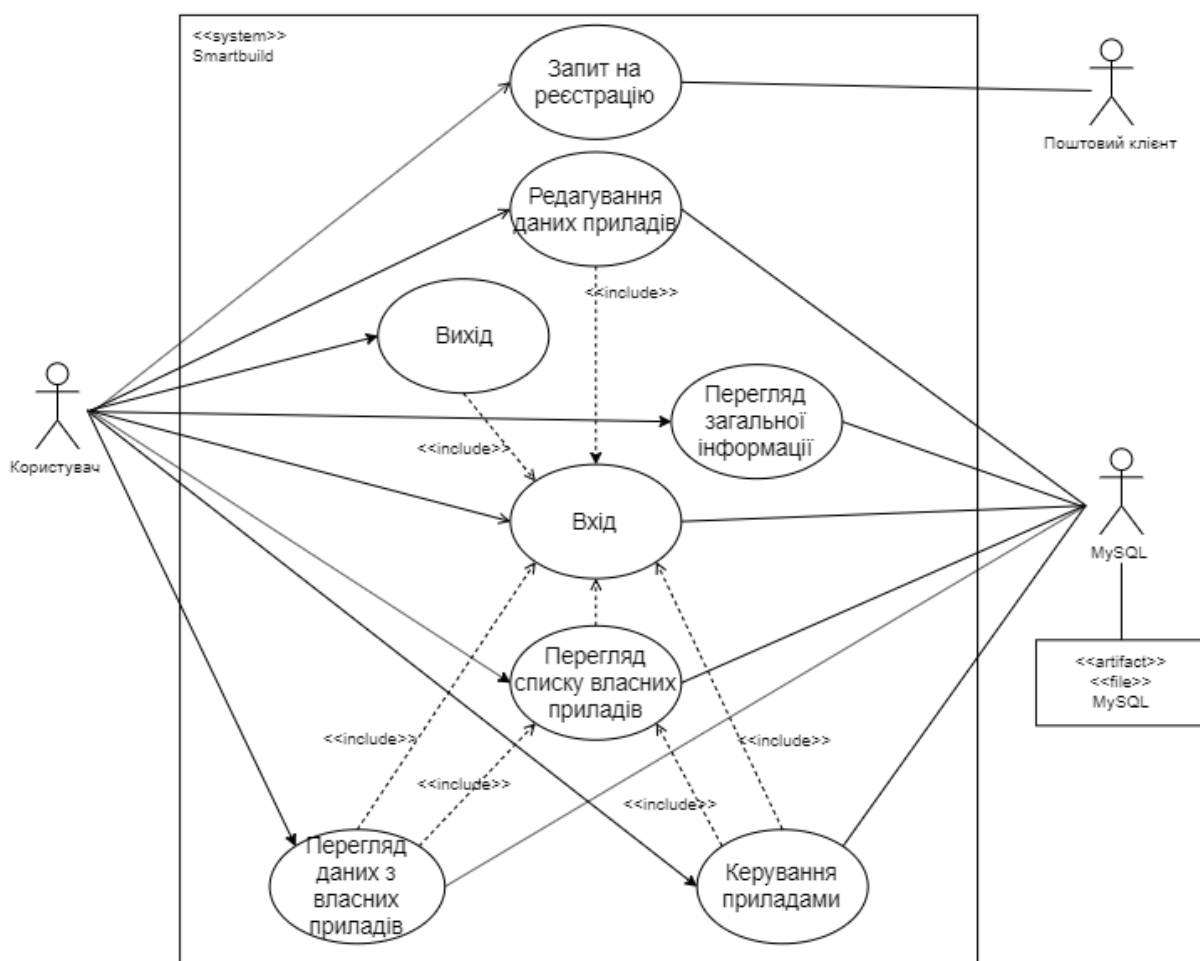


Рисунок 2.4 — Варіанти використання для актора «Користувач»

На рисунку представлено наступні варіанти використання (ВВ):

- ВВ перегляд загальної інформації – перегляд головної сторінки сайту та новин сайту (доступний без авторизації);

- ВВ запит на реєстрацію – доступний без авторизації, дає можливість не зареєстрованому користувачу подати заявку на реєстрацію;
- ВВ вхід – актор авторизується в системі використовуючи унікальний логін та пароль, що дає йому доступ до розділів системи в залежності від його прав доступу;
- ВВ перегляд списку власних приладів – передбачає перегляд всіх доступних для авторизованого користувача приладів;
- ВВ керування приладами – передбачає керування групами приладів та переміщення приладів між існуючими групами;
- ВВ редагування даних приладів – передбачає зміну параметрів приладів (назви, меж для приладів, поточного стану активності);
- ВВ перегляд даних з власних приладів – перегляд поточних та архівних даних з приладів.
- ВВ вихід – завершення сесії для поточного користувача і вихід з системи. Діаграму сценаріїв використання для адміністратора наведено на рисунку 2.5. На рисунку представлено наступні варіанти використання:
 - ВВ перегляд загальної інформації – перегляд головної сторінки сайту та новин сайту (доступний без авторизації);
 - ВВ вхід – актор авторизується в системі використовуючи унікальний логін та пароль, що дає йому доступ до розділів системи в залежності від його прав доступу;
 - ВВ перегляд списку користувачів – Перегляд списку всіх приладів з можливістю перейти до особистого кабінет будь-якого користувача.
 - ВВ перегляд приладів користувача – передбачає перегляд всіх приладів, доступних обраному користувачу;
 - ВВ керування приладами – передбачає керування приладами, які не належать жодному з користувачів;
 - ВВ перегляд даних з приладів користувача – перегляд поточних та архівних даних з приладів обраного користувача.

- ВВ адміністрування сайту – доступ до адміністративної частини сайту та керування типами приладів.
- ВВ вихід – завершення сесії для поточного користувача і вихід з системи.

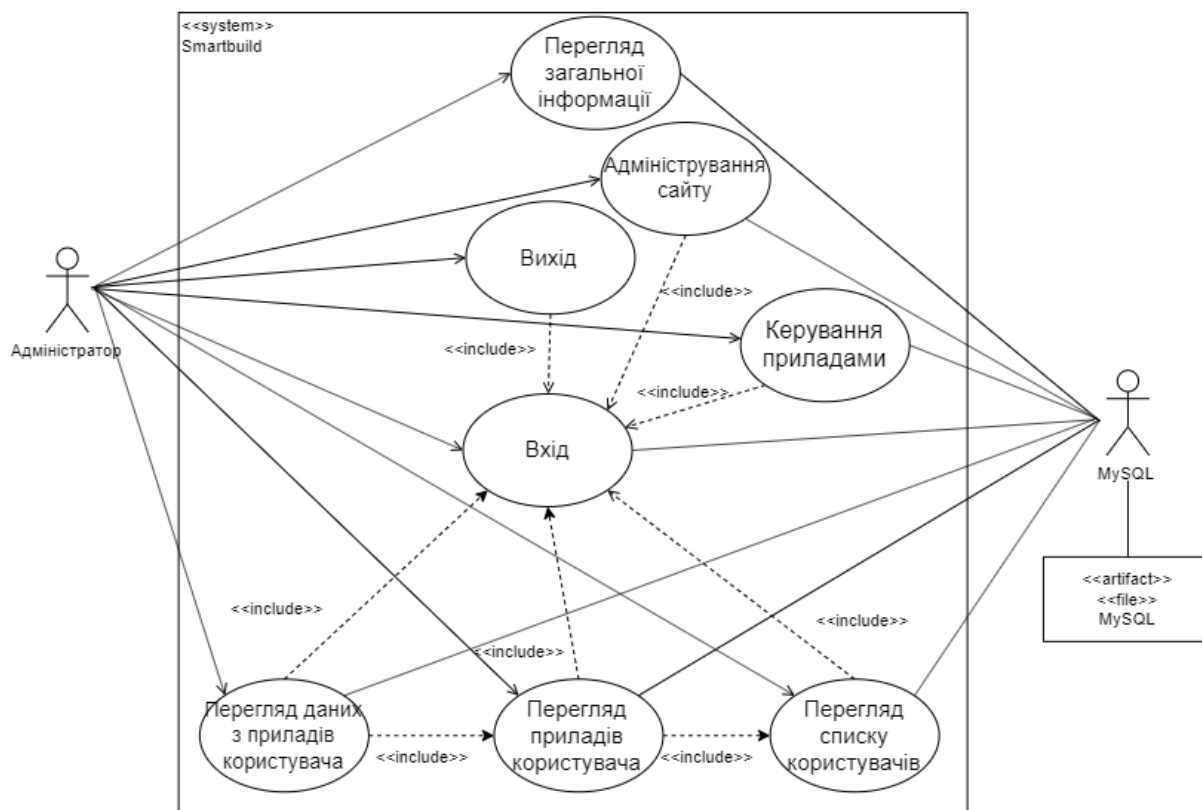


Рисунок 2.5 — Варіанти використання для актора «Адміністратор»

2.3 Проектування моделі бази даних

Діаграма сутність-зв'язок відображає відносини між об'єктами, що зберігаються в базі даних. Кожна сутність має атрибути, що визначають її властивості [13].

Для функціонування системи необхідно створити 8 таблиць, що можуть зберігати необхідні сутності. Також необхідно правильно розставити взаємозв'язки між створеними сутностями.

Проаналізувавши сутності, що використовуються в моделі ІС, перейдемо до реалізації структури БД.

Модель бази даних наведено на рисунку 2.6.

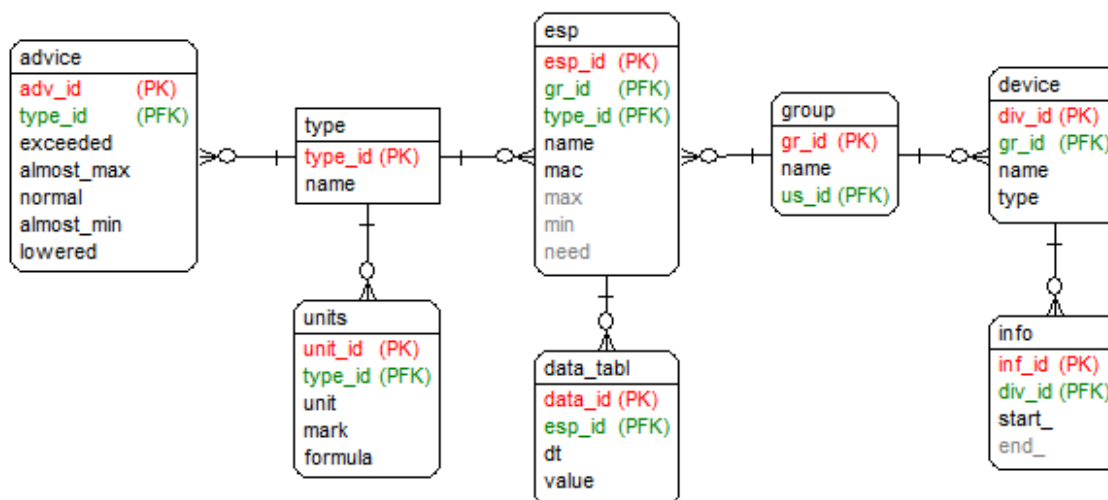


Рисунок 2.6 — ER-модель бази даних

Для кращого представлення структури бази даних наведемо імена необхідних таблиць, атрибутів, типів, їх призначення та обмеження в таблиці 2.1.

Таблиця 2.1 — Концептуальна модель даних

Таблиця	Поле	Зміст	Тип	Ключі	Обмеження
group	gr_id	Ідентифікатор групи	bigint(20)	PK	Не пустий, унікальний
	us_id	Ідентифікатор користувача	bigint(20)	FK	Не пустий
	name	Назва групи	varchar(80)		Не пустий
type	type_id	Ідентифікатор типу	bigint(20)	PK	Не пустий, унікальний
	name	Назва типу	varchar(20)		Не пустий
units	unit_id	Ідентифікатор одиниць вимірювання	bigint(20)	PK	Не пустий, унікальний
	type_id	Ідентифікатор типу	bigint(20)	FK	Не пустий
	unit	Назва одиниці вимірювання	varchar(30)		Не пустий
	mark	Скорочене позначення	varchar(10)		Не пустий
	formula	Дата заняття	varchar(20)		Не пустий
esp	esp_id	Ідентифікатор esp	bigint(20)	PK	Не пустий, унікальний
	gr_id	Ідентифікатор групи	bigint(20)	FK	Не пустий
	type_id	Ідентифікатор типу	bigint(20)	FK	Не пустий
	name	Назва приладу	varchar(20)		Не пустий

Продовження таблиці 2.1 — Концептуальна модель даних

esp	mac	MAC адреса пристрою	varchar(20)		Не пустий, унікальний
	max	Максимальний допустимий показник	bigint(20)		Не пустий
	min	Мінімальний допустимий показник	bigint(20)		Не пустий
advice	adv_id	Ідентифікатор поради	bigint(20)	PK	Не пустий, унікальний
	type_id	Ідентифікатор типу	bigint(20)	FK	Не пустий
	exceeded	Порада при перевищенні	text		Не пустий
	almost_max	Порада при наближенні до максимуму	text		Не пустий
	normal	Порада при нормальних показниках	text		Не пустий
	almost_min	Порада при наближенні до мінімуму	text		Не пустий
	lowered	Порада при заниженні граничних показників	text		Не пустий
data	data_id	Ідентифікатор запису даних	bigint(20)	PK	Не пустий, унікальний
	esp_id	Ідентифікатор esp	bigint(20)	FK	Не пустий
	dt	Дата та час запису	datetime		Не пустий
	value_	Значення	double		Не пустий
device	div_id	Ідентифікатор приладу	bigint(20)	PK	Не пустий, унікальний
	type_id	Ідентифікатор типу	bigint(20)	FK	Не пустий
	name	Назва приладу	varchar (80)		Не пустий
	type	Тип приладу	varchar (80)		Не пустий
info	inf_id	Ідентифікатор запису даних	bigint(20)	PK	Не пустий, унікальний
	div_id	Ідентифікатор приладу	bigint(20)	FK	Не пустий
	start	Дата початку роботи приладу	datetime		Не пустий
	end	Дата кінця приладу	datetime		

Таблиця «group» необхідна для зберігання інформації про групи датчиків користувачів. Ця таблиця є проміжною ланкою між структурою таблиць WordPress та доповненою нами структурою. Поле «us_id» посиляється на унікальний ідентифікатор користувач WordPress в таблиці «users».

Таблиця «esp» зберігає інформацію про наявні в системі пристрої. Жоден пристрій не може існувати в системі без групи. У кожного пристрою мусить бути тип (датчик вологості, температури і т.д.). Інформація про типи пристроїв зберігається у таблиці «type». У кожного типу датчиків може бути декілька одиниць вимірювання,

інформація про які зберігається у таблиці «units». Також для кожного типу передбачені поради, що будуть виводитися користувачу в залежності від фактичного значення показників стану повітря в приміщенні. Поради щодо регулювання показників стану повітря в приміщенні зберігаються у таблиці «advice».

Вся інформація про показники стану повітря, з вказанням ідентифікатора приладу, часу та значення відповідного показника, по всім приладам зберігається в таблиці «data».

3 РОЗРОБКА ІНФОРМАЦІЙНО СИСТЕМИ

В якості хостинг-провайдера було обрано компанію HostPro. Вони надають надійний хостинг з захистом від DDoS-атак та щоденним архівуванням даних. Також вони надають 5 GB пам'яті дискового простору 1 GB, що повністю покриває вимоги до хостингу.

3.1 Архітектура інформаційної системи

Діаграма архітектури – це графічне зображення сукупності понять, що входять до складу архітектури, включаючи їх принципи, елементи та компоненти [14].

На рисунку 3.1 зображено архітектуру інформаційної системи SmartBuild.

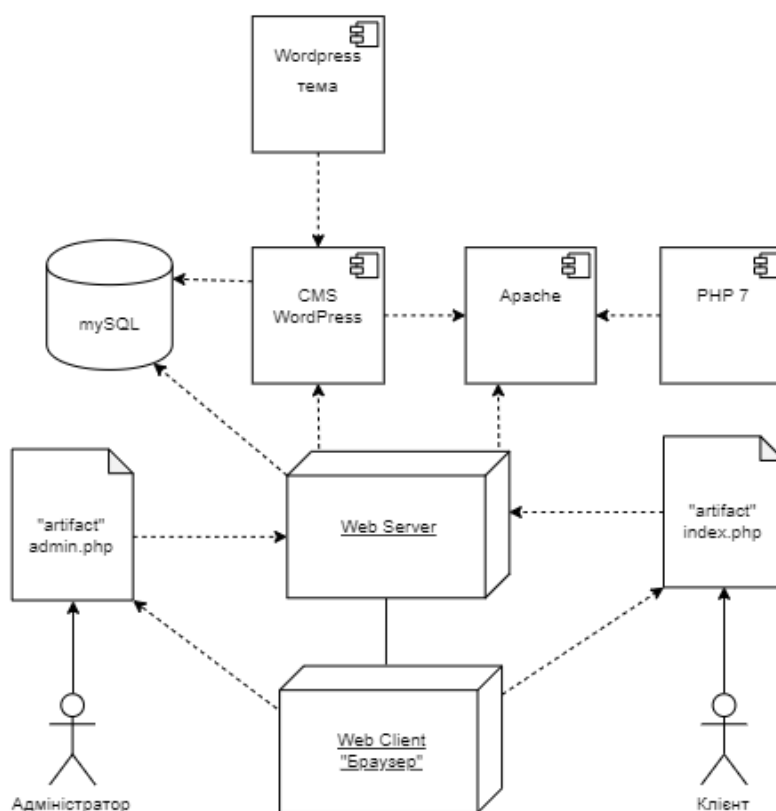


Рисунок 3.1 — Архітектура інформаційної системи

На веб сервері встановлено CMS WordPress. Інтерпретація PHP коду здійснюється за допомогою Apache HTTP Server. WordPress також взаємодіє з базою даних. WordPress містить тему, яка відповідає за відображення контенту сайту.

Користувачі взаємодіють з системою через web-браузер. Адміністративна частина інформаційної системи (admin.php) надає інтерфейс для керування контентом на сайті. Модуль для клієнтів (index.php) є користувацьким інтерфейсом для взаємодії з системою. Користувачі отримують доступ до системи після авторизації, в залежності від їх статусу.

3.2 Програмна реалізація

3.2.1 Створення бази даних

Для того щоб повноцінно використовувати можливості WordPress, як блогу та мати можливість обробляти дані про показники стану повітря в приміщенні, було розширено базу даних WordPress. Для доповнення стандартної бази даних WordPress було використано інструменту імпорту в phpMyAdmin. Скрипти для створення таблиць наведено в додатку В.

WordPress надає зручні та безпечні інструменти управління своєю базою даних за рахунок php класу wpdb. Цей клас дозволяє виконувати будь-які запити до бази даних WordPress: додавати, оновлювати, отримувати та видаляти дані. Код створення об'єкту класу wpdb{} з зазначенням параметрів підключення до бази даних знаходиться в файлі wp-config.php. Звернення до методів класу відбувається через глобальну змінну \$wpdb. Після доповнення бази даних не потрібно створювати новий об'єкт, цей клас може працювати з користувацькими таблицями в базі даних WordPress [15].

Деякі приклади використання методів вбудованого WordPress класу wpdb{} наведено на рисунку 3.2.

```

global $wpdb;
$wpdb->insert(
    'wp_custom_group',
    array( 'us_id' => $user_id, 'name' => 'Моя група' ),
    array( '%d', '%s' )
);

return $wpdb->get_results(
    $wpdb->prepare("SELECT type_id, name FROM wp_custom_type;"),
    ARRAY_A );

$wpdb->delete(
    'wp_custom_group',
    array( 'gr_id' => $_POST['dell_group'] ),
    array( '%d' );

$wpdb->query($wpdb->prepare(
    "INSERT INTO wp_custom_info( dev_id, start_ ) VALUES (%d,now());",
    $dev_id
));

```

Рисунок 3.2 — Приклади використання класу `wpdb`

Метод `insert` призначений для додавання записів у зазначену таблицю і має три параметри. Перший параметр – назва таблиці, другий – асоціативний масив елементів для вставки в таблицю, третій – масив форматів для вставки. Змінна може бути тільки трьох типів: рядок – ‘%s’ (за замовчуванням), ціле число ‘%d’ та число з плаваючою комою – ‘%f’.

Метод `delete` призначений для видалення даних з вказаної таблиці. Він має такий же набір параметрів, як і метод `insert`.

Метод `get_results` повертає результат запиту у вигляді масиву, кожен елемент якого є одним записом у базі даних. Дозволяє виконувати підзапити будь-якої складності.

Метод `query` дозволяє виконувати довільні запити в базу даних. В прикладі наведено запит на додавання запису з використанням функції `now()`. Метод `insert` не надає можливості виконати такий запит.

Клас `wpdb` має інструменти захисту від sql-ін’єкцій. Методи `insert` та `delete` мають вбудований механізм захисту, а в методах `query` та `get_results` з цією метою використано метод `prepare`. Це гарантує надійність розробленої системи та захищає її від краху.

3.2.2 Створення теми

WordPress теми складаються з файлів і стилів, що визначають зовнішній вигляд сайту. Теми розділяють представлення та контент, що дає можливість змінити дизайн сайту без зміни контенту. WordPress тема має відповідати певній структурі [16].

Оскільки жодна з існуючих WordPress тем не має необхідного функціоналу, було розроблено власну WordPress тему «Smartbuild».

Всі файли теми «SmartBuild» наведено на рисунку 3.3.

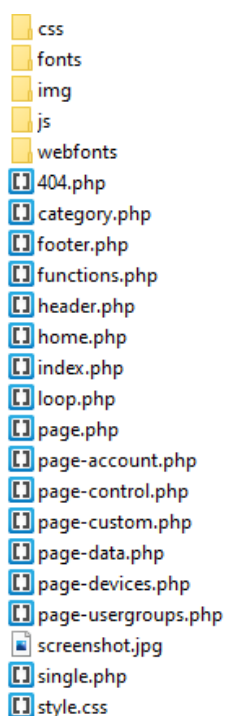


Рисунок 3.3 — Список файлів теми SmartBuild

Файли `header.php` і `footer.php` містять шапку сайту та підвал сайту відповідно. Файл `home.php` містить код, що формує контент на головній сторінці сайту. Файл `category.php` відповідає за відображення категорій на сайті, а файл `single.php` є шаблоном для відображення всіх записів у будь-якій категорії. Файл `page.php` є шаблоном у відповідності до якого виводиться контент на всіх сторінках сайту. Якщо потрібно прописати додаткову логіку чи змінити спосіб відображення контенту, то необхідно створити додатковий файл з частиною посилання обраної сторінки. Файл `function.php` містить всі необхідні функції та хуки, що необхідні для функціонування теми. Файл `style.css` є головним файлом стилів теми, а також містить інформацію про

тему та її розробника. Файл `screenshot.jpg` є основним зображенням теми, яке відображається в панелі адміністратора при перегляді списку встановлених тем.

Для формування каркасу сайту було використано бібліотеку Bootstrap v4.3.1. Дана бібліотека надає широкі можливості для створення адаптивних, сучасних та інтерактивних сайтів. Бібліотека містить набір стилів та JavaScript плагінів, що значно полегшують розробку сайту [17].

Сторінки сайту створені з використанням сітки, меню, слайдера, модальних вікон та інших компонентів bootstrap, що застосовувались для стилістичного оформлення HTML елементів.

WordPress має сторінку реєстрації користувачів в панелі адміністратора та форму авторизації, що знаходиться за url-адресою `domain_name/wp-login.php`.

Стандартну форму авторизації користувачів зображено на рисунку 3.4.

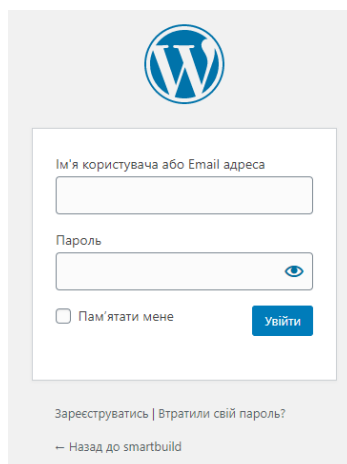


Рисунок 3.4 — Стандартна форма авторизації WordPress

Стандартна форма авторизації була кастомізована і винесена в діалогове вікно, що дає можливість більш швидкого і більш зручного доступу до функції авторизації. Кастомізована форма авторизації має дизайн, що відповідає загальному дизайну сайту. Кастомізована форма авторизації користувачів зображено на рисунку 3.5.

Рисунок 3.5 — Кастомізована форма авторизації користувачів

Такий підхід дає можливість і далі користуватися системою розділення прав авторизованих користувачів. В залежності від статусу користувача, що увійшов на сайт, він має доступ до різних сторінок, а отже потрібно виводити різні меню. Код, що відповідає за виведення пункту меню в залежності від статусу користувача, наведено на рисунку 3.6.

```

<div class="collapse navbar-collapse" id="navbarNavDropdown">
  <?php
  if ( is_user_logged_in() ){
    if(curent_user_is_admin()){

      $args = array(
        'theme_location' => 'admin',
        'container'=> false,
        'menu_id' => 'bottom-nav',
        'items_wrap' => '<ul id="%1$s" class="navbar-nav %2$s"%3$s</ul>',
        'menu_class' => 'top-menu',
        'walker' => new bootstrap_menu(true)
      );
      wp_nav_menu($args);

    } else {

      $args = array(
        'theme_location' => 'custom',
        'container'=> false,
        'menu_id' => 'bottom-nav',
        'items_wrap' => '<ul id="%1$s" class="navbar-nav %2$s"%3$s</ul>',
        'menu_class' => 'top-menu',
        'walker' => new bootstrap_menu(true)
      );
      wp_nav_menu($args);

    } } else {

      $args = array(
        'theme_location' => 'top',
        'container'=> false,
        'menu_id' => 'top-nav-ul',
        'items_wrap' => '<ul id="%1$s" class="navbar-nav %2$s"%3$s</ul>',
        'menu_class' => 'top-menu',
        'walker' => new bootstrap_menu(true)
      );
      wp_nav_menu($args);
    }
  }
  >>
</div>
<?php if ( is_user_logged_in() ){ echo "<a class='\"btn btn btn-outline-primary\"' href='\".wp_logout_url().\"'>Вийти</a>"; } else {?>
<button type="button" class="btn btn btn-outline-primary" data-toggle="modal" data-target="#exampleModal">Увійти</button>
<?php }?>

```

Рисунок 3.6 — Виведення пункту меню

Якщо користувач не авторизований, то виводимо top-меню. Якщо користувач авторизований і має статус адміністратора, то виводимо admin-меню. Якщо ж користувач авторизований, але не є адміністратором, виводимо custom-меню. В панелі адміністратора WordPress можна налаштувати відображення пунктів меню для кожного зареєстрованого в темі меню.

Для того щоб отримати дані з бази даних, вивести записи в категорії, обробити дані чи вивести пагінацію на сторінці необхідно створити функцію. З теми можна використовувати вбудовані функції WordPress для вирішення ряду задач чи створити власну функцію. Всі користувацькі функції, необхідні для функціонування теми, зберігаються у файлі `functions.php`. Для розробленої теми було створено 47 функцій, що виконують різноманітні задачі. Програмний код декількох розроблених функцій наведено на рисунку 3.7.

```

/*
 * Перевіряє чи є поточний користувач адміністратором
 */
if (!function_exists('current_user_is_admin')) {
function current_user_is_admin() {
    return get_userdata(get_current_user_id())->user_level ==10;
}
}

/*
 * Витягнути всі групи користувача
 */
if (!function_exists('select_cusom_groups')) {
function select_cusom_groups( $user_id ) {
global $wpdb;
return $wpdb->get_results(
    $wpdb->prepare("SELECT name, gr_id FROM wp_custom_group WHERE us_id= %d", $user_id)
    , ARRAY_A );
}
}

/*
 * Витягнути всі типи приладів
 */
if (!function_exists('select_cusom_types')) {
function select_cusom_types() {
global $wpdb;
return $wpdb->get_results(
    $wpdb->prepare("SELECT type_id, name FROM wp_custom_type;")
    , ARRAY_A );
}
}

```

Рисунок 3.7 — Програмний код деяких розроблених функцій

На рисунку наведено код трьох функцій. Функція `current_user_is_admin()` повертає значення «істина», якщо поточний користувач має статус адміністратора. Функція `select_custom_groups($user_id)` повертає масив записів про групи, що

належать вказаному користувачу. Функція `select_custom_types()` робить вибірку з таблиці типів та повертає масив типів приладів.

Також WordPress має велику кількість вбудованих функцій, що використовуються для вирішення стандартних задач. Наприклад, вивести запис в категорії, отримати список користувачів, отримати інформацію про користувача з його ідентифікатором та зробити ще багато іншого можна скориставшись вбудованими функціями WordPress.

На рисунку 3.8 наведено приклад використання стандартних функцій WordPress для виведення записів категорії.

```
<?php if (have_posts()) : while (have_posts()) : the_post(); ?>
<div class="row post py-4">
  <div class="col-xl-2 col-lg-3 col-md-4">
    <p><i class="fas fa-calendar-alt fa-x2"></i> <?php the_time(get_option('date_format')); ?></p>
    <?php echo the_post_thumbnail( 'spec_thumb' ); ?>
  </div>
  <div class="col-xl-10 col-lg-9 col-md-8">
    <a href="<?php the_permalink(); ?>" title="<?php the_title(); ?>">
      <h2> <?php the_title(); ?> </h2>
    </a>
    <?php $content = get_the_content();
    Stripped_content = wp_trim_words( $content, 40, '<a href="'. get_permalink() .'"> ...Read More</a>' );
    echo Stripped_content; ?>
  </div>
</div>
<?php endwhile;
else: echo '<p>Записи відсутні.</p>'; endif; ?>
```

Рисунок 3.8 — Програмний код виведення записів у категорії

На рисунку 3.8 наведено відображення записів у категорії. У даному фрагменті коду використано лише WordPress функції, він є повністю функціональним і не потребує розробки додаткових функцій.

Всі дані з форм, які вводить користувач і адміністратор, передаються методом `post`.

Метод `post` ніколи не кешується, не зберігається в історії браузера та не має обмеження в об'ємі даних, що передаються [18]. Це дає можливість безпечно передавати та обробляти дані. WordPress має власний механізм обробки даних з форм, що були передані методом `post`.

На рисунку 3.9 наведено код форми для створення нової групи в особистому кабінеті користувача.

```

<form action="<?php echo esc_url( admin_url('admin-post.php') ); ?>" method="post">
  <div class="modal-body">
    <input type="hidden" name="action" value="add_admin_group">
    <input type="hidden" name="user_id" value="<?php echo get_current_user_id(); ?>">
    <input type="hidden" name="redirect" value="/account">
    <label for="new_gr">Назва групи</label>
    <input class="form-control" type="text" placeholder="Введіть назву групи" name="name" id="new_gr">
  </div>
  <div class="modal-footer">
    <button type="reset" class="btn btn-outline-secondary" data-dismiss="modal">Закрити</button>
    <button type="submit" class="btn btn-outline-success">Додати</button>
  </div>
</form>

```

Рисунок 3.9 — Форма додавання групи користувача

Дані з форми для обробки необхідно передати на сторінку `admin-post.php`. Окрім полів, які заповнює користувач, форма також має містити приховане поле «action» з назвою дії, що передбачена для обробки даної форми. Поле «redirect» містить частину посилання сторінки, на яку потрібно перейти після закінчення процесу обробки даних з форми. Для успішної обробки форми необхідно створити функцію для обробки даних та додати 2 хуки, що будуть викликати створену функцію у разі виконання зазначеної дії.

Програмний код обробки даних з наведеної форми зображено на рисунку 3.10.

```

/*
 * Додавання групи (post)
 */
function add_admin_group() {
    global $wpdb;
    $wpdb->insert(
        'wp_custom_group',
        array( 'name' => $_POST['name'], 'us_id' => $_POST['user_id'] ),
        array( '%s', '%d' )
    );
    wp_safe_redirect( site_url( $_POST['redirect'] ) );
}
add_action( 'admin_post_nopriv_add_admin_group', 'add_admin_group' );
add_action( 'admin_post_add_admin_group', 'add_admin_group' );

```

Рисунок 3.10 — Програмний код обробки форми додавання групи

В даному випадку було створено функцію `add_admin_group()`, що додає запис в таблицю «`wp_custom_group`». Після виконання функції перенаправляємо сервер на потрібну сторінку за допомогою функції `wp_safe_redirect()`.

Дуже важливою частиною виконаної роботи є візуалізація даних про показники стану повітря в приміщенні. Дані відображаються на двовірному графіку Google Line Chart.

Google Line Chart використовує HTML5, SVG та JavaScript для створення графіків. Він легко налаштовується та надає можливість відображення даних у режимі реального часу [19].

Скрипт для створення графіку наведено на рисунку 3.11

```

<script>
google.charts.load('current', {'packages': ['line', 'corechart']});
google.charts.setOnLoadCallback(drawChart);
function drawChart() {
    var button = document.getElementById('change-chart');
    var chartDiv = document.getElementById('chart_div');
    var data = new google.visualization.DataTable();
    data.addColumn('string', 'Month');
    data.addColumn('number', "Фактичний <?php echo $type; ?>");
    data.addColumn('number', "Мінімально допустимий");
    data.addColumn('number', "Максимально допустимий");
    data.addColumn('number', "Необхідний показник");
    data.addRows([
        <?php foreach ( select_cusom_data($esp_id, $start, $end,
            $unit_arr['formula']) as $data ){ ?>
        ["<?php echo $data['d']; ?>",<?php echo $data['v']; ?>,<?php echo
            $min_limit; ?>,<?php echo $max_limit; ?>,<?php echo $need_limit; ?
            >],<?php } ?>
    ]);
    var classicOptions = {
        title: 'Графік з <?php echo $start; ?> по <?php echo $end; ?>',
        width: 900, height: 500,
        series: {0: {color: '#c028f8' },1: { color: '#6771e8' }, 2:
            {color: '#e2431e' }, 3: { color: '#2be71b'},},
        vAxes: { 0: {title: '<?php echo $unit_arr['unit']; ?> (<?php
            echo $unit_arr['mark']; ?>)'}},
        vAxis: {viewWindow: { max: <?php echo $max_limit+5; ?>,min: <?
            php echo $min_limit-5; ?>}}
    };
    function drawClassicChart() {
        var classicChart = new
            google.visualization.LineChart(chartDiv);
        classicChart.draw(data, classicOptions);
    }
    drawClassicChart();
}
</script>

```

Рисунок 3.11 — Програмний код обробки форми додавання групи

В наведеному фрагменті коду здійснюється налаштування параметрів відображення графіку та формування масиву даних для відображення.

Код основних файлів теми наведено в додатку Г.

3.3 Використання програмного додатку

Доступ до розробленої інформаційної системи можна отримати за посиланням <http://smartbuild.ml>. Не авторизованому користувачу доступна лише загальна інформація про систему.

Зображення головної сторінки наведено на рисунку 3.12.

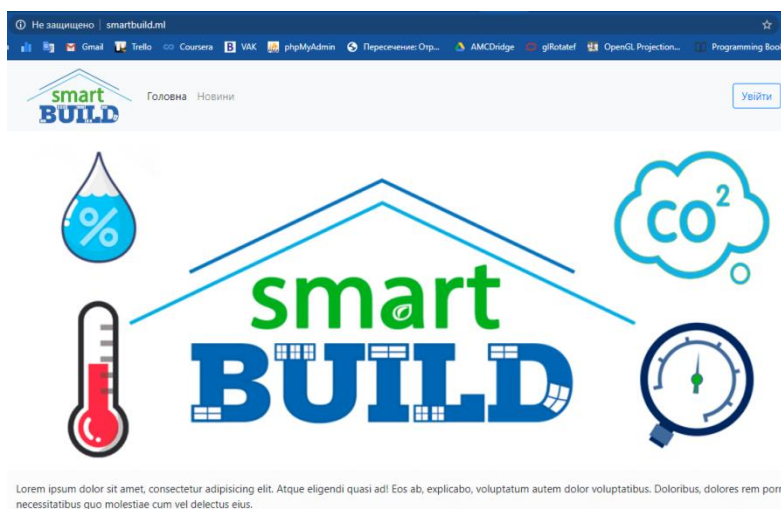


Рисунок 3.12 — Головна сторінка

Не авторизованому користувачу також доступний категорія «Новини». На рисунку 3.13 зображено категорію «Новини».

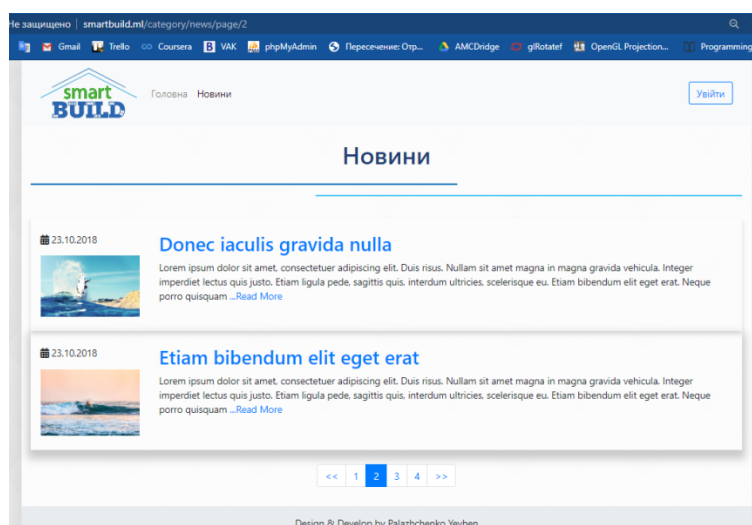


Рисунок 3.13 — Категорія «Новини»

Для того щоб отримати доступ до інформаційної системи необхідно подати заявку на реєстрацію.

На рисунку 3.14 зображено форму подачі заявки на реєстрацію.

Рисунок 3.14 — Форма подачі заявки на реєстрацію

Форма подачі заявки містить три обов'язкових до заповнення поля: «Ваше ім'я», «Ваш email» та «Ваш номер». Дані з форми надходять на пошту адміністратору системи, після чого адміністратор зв'язується з клієнтом для обговорення деталей.

Після особистого спілкування та видачі клієнту пристроїв збору інформації про показники стану повітря в приміщенні адміністратор реєструє нового клієнта через адміністративну панель WordPress.

Така процедура передбачена для того, щоб в систему не міг потрапити користувач, який не має пристроїв збору інформації про показники стану повітря в приміщенні.

Адміністратор передає клієнту унікальні логін та пароль зручним для клієнта способом. Після чого користувач може авторизуватися в системі і потрапити до свого особистого кабінету.

На рисунку 3.15 зображено особистий кабінет користувача.

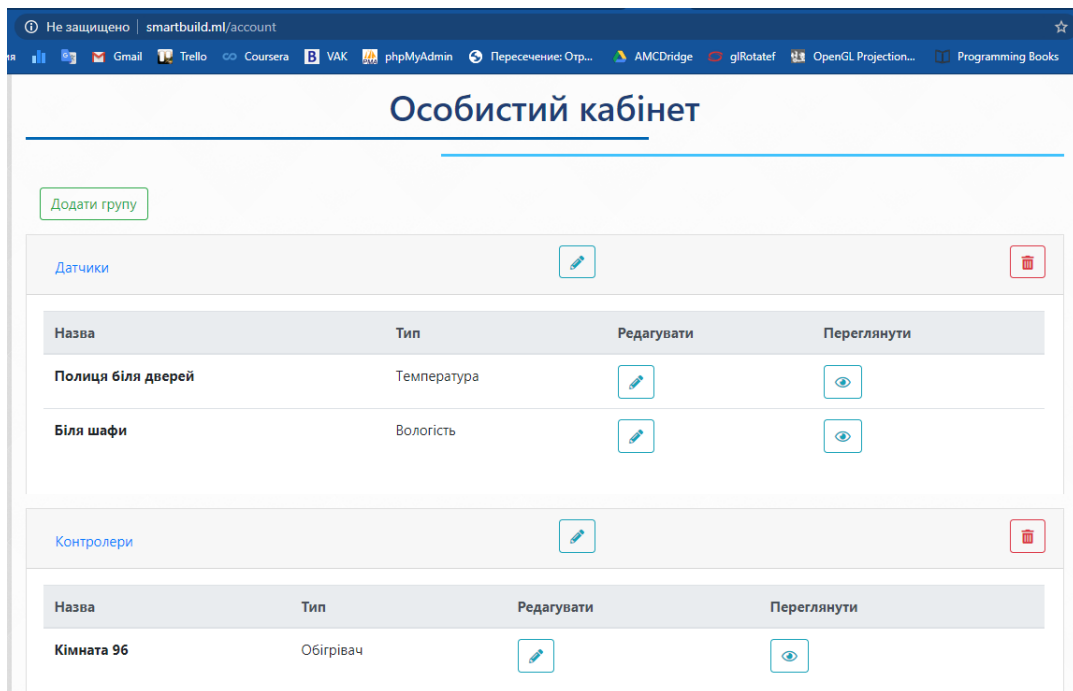


Рисунок 3.15 — Особистий кабінет користувача

В особистому кабінеті користувач отримує повний контроль над своїми приладами. Можна створювати, редагувати та видаляти групи для приладів, змінювати параметри пристроїв.

Форма зміни параметрів пристрою наведена на рисунку 3.16.

Рисунок 3.16 — Форма зміни параметрів пристрою

Як видно з рисунку можна змінити назву пристрою та перемістити обраний пристрій в іншу групу.

Для того щоб переглянути дані з пристрою потрібно натиснути на кнопку з іконкою ока.

На рисунку 3.17 зображено сторінку перегляду даних з приладу для вимірювання температури.

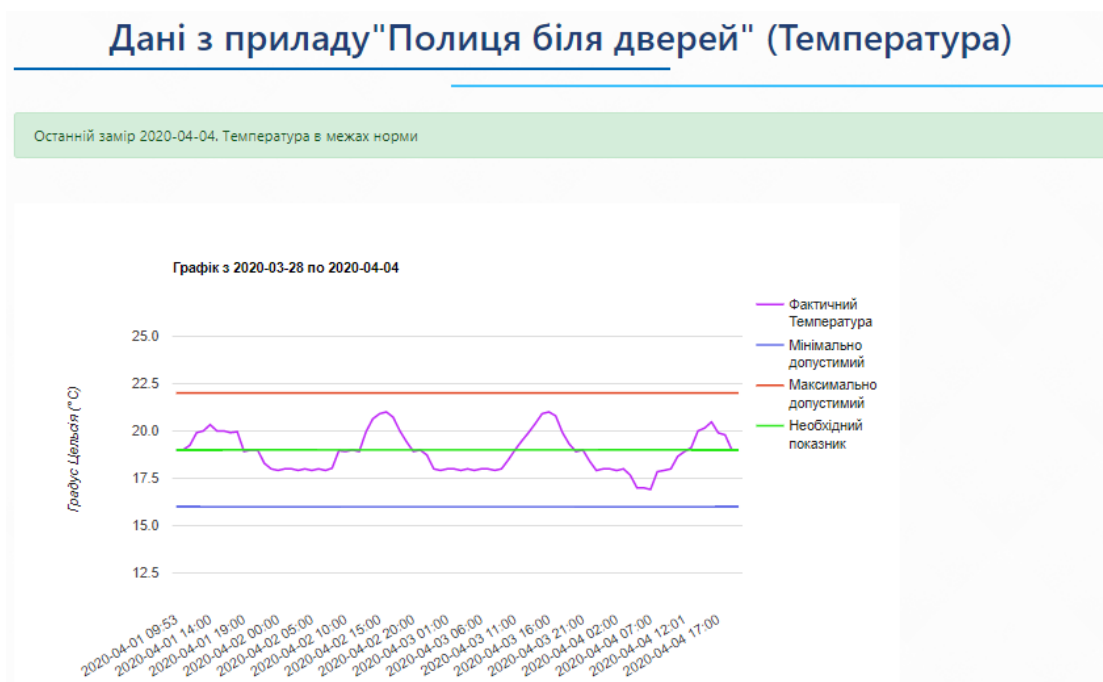


Рисунок 3.17 — Сторінка перегляду даних з приладу

На рисунку зображено графік зміни температури за період з 28.03.2020 по 04.04.2020. Над графіком вказано дату коли був здійснений останній замір температури та рекомендацію щодо її регулювання. В даному випадку температура знаходиться в межах встановленої норми.

Користувач може змінювати параметри перегляду графіку.

На рисунку 3.18 наведено форми зміни параметрів перегляду графіку та зміни гранично-допустимих значень для обраного пристрою.

Початок інтервалу	Кінець інтервалу	Одиниці вимірювання
<input type="text" value="28.03.2020"/>	<input type="text" value="04.04.2020"/>	<input type="text" value="Градус Цельсія (°C)"/>
<input type="button" value="Повторний запит"/>		
Мінімум	Максимум	Необхідний показник
<input type="text" value="16"/>	<input type="text" value="22"/>	<input type="text" value="19"/>
<input type="button" value="Змінити границі"/>		

Design & Develop by Palazhchenko Yevhen

Рисунок 3.18 — Форми зміни параметрів перегляду графіку

Користувач може обирати період перегляду даних та одиниці вимірювання. Також може змінювати гранично допустимі значення для обраного пристрою. Від вказаних граничних значень залежать рекомендації, які видає система.

На сторінці перегляду даних з приладу «Обігрівач» графік відображає періоди активності приладу та має інший вигляд.

Сторінку перегляду даних з приладу «Обігрівач» наведено на рисунку 3.19.

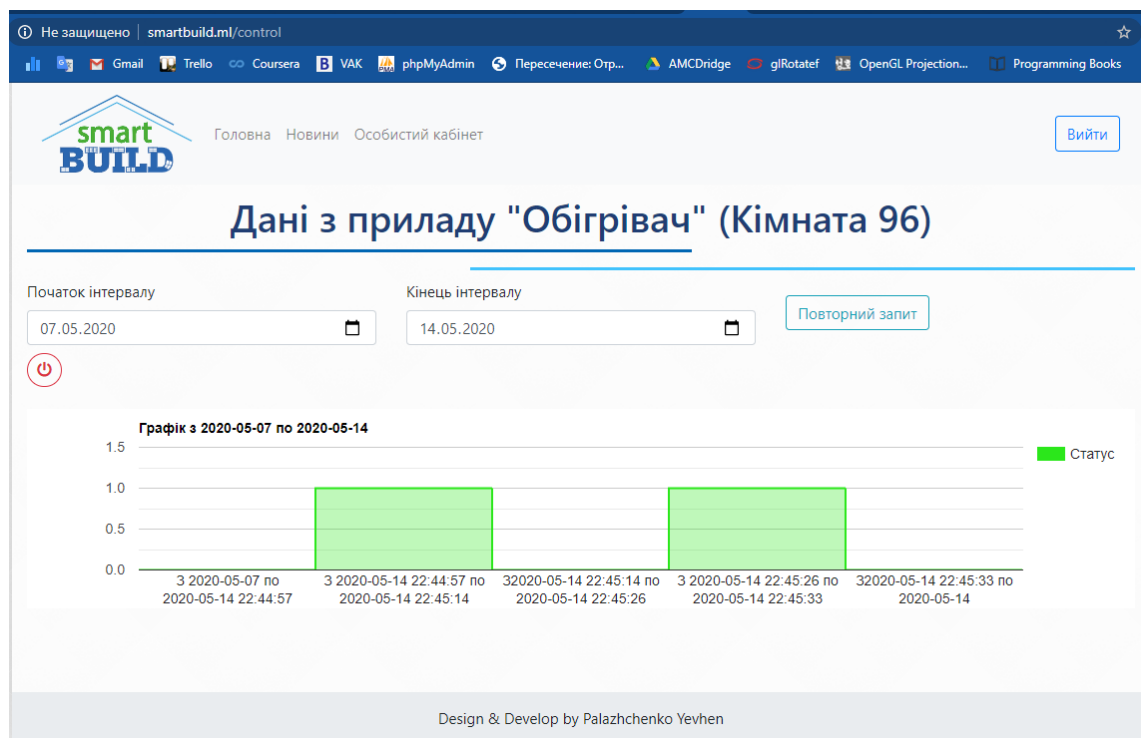


Рисунок 3.19 — Сторінка перегляду даних з приладу «Обігрівач»

На графіку зеленим кольором зображено періоди активності обігрівача. Над графіком знаходиться форма зміни інтервалу перегляду даних, що дозволяє переглядати архівні дані.

Поточний стан активності приладу можна змінити, використовуючи кнопку вмикання/вимикання приладу, що розташована над графіком.

Якщо авторизований користувач має статус адміністратора системи, то він має розширені можливості в системі. Після авторизації адміністратор потрапляє на сторінку перегляду списку користувачів.

Сторінку перегляду користувачів зображено на рисунку 3.20.

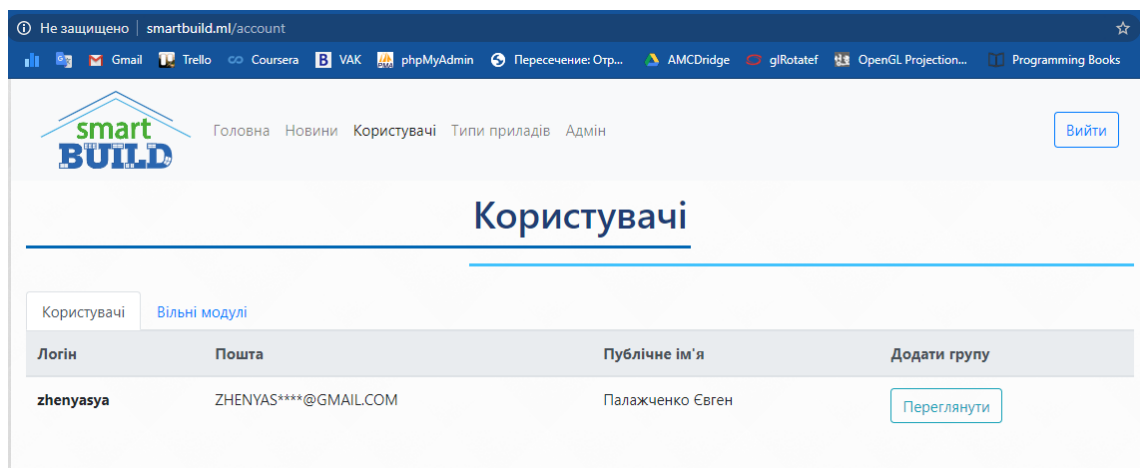


Рисунок 3.20 — Сторінку перегляду користувачів

На цій сторінці адміністратор може переглядати список користувачів та перейти в особистий кабінет будь-якого користувача.

Сторінка перегляду груп обраного користувача наведено на рисунку 3.21.

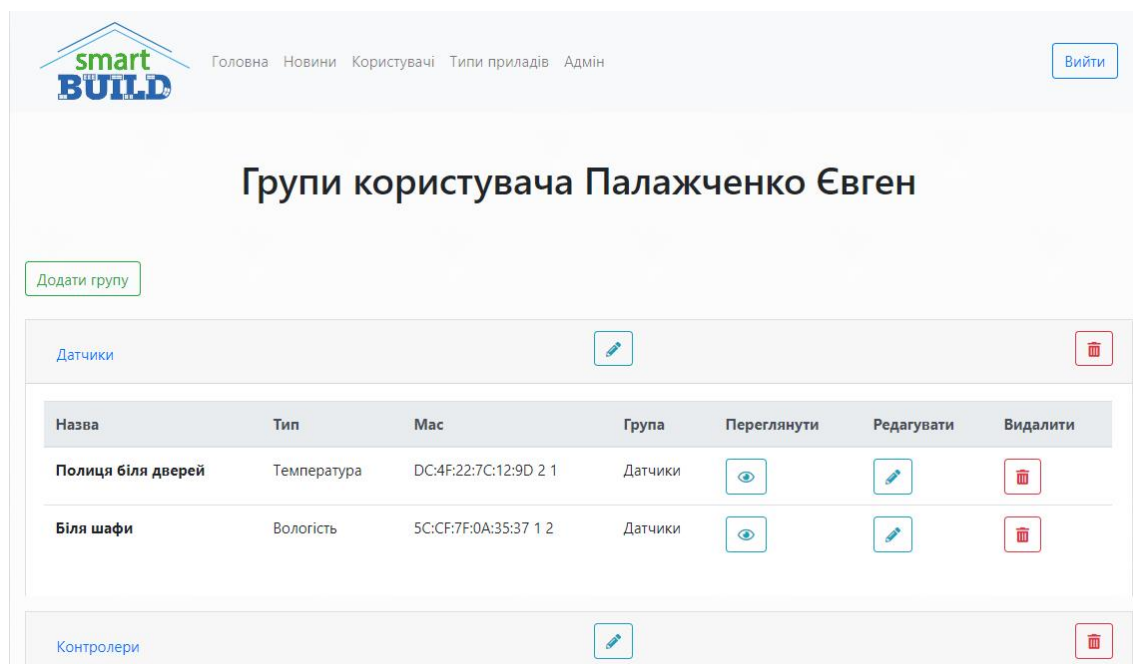


Рисунок 3.21 — Сторінка перегляду груп Палажченко Євгена

Адміністратор, окрім дій доступних звичайному користувачу, також може видалити пристрій з групи користувача та змінити тип обраного пристрою. Адміністратор має такі ж права доступу до перегляду даних з приладів як і користувач.

Адміністратору системи також доступна сторінка «Типи приладів», яку зображено на рисунку 3.22.

The screenshot shows the 'Типи приладів' (Types of devices) page in the smartbuild.ml/admin interface. The page has a navigation bar with 'Головна', 'Новини', 'Користувачі', 'Типи приладів', and 'Адмін'. A 'Вийти' (Logout) button is in the top right. The main content area is titled 'Типи приладів' and contains a form for adding a new device type. The form has four input fields: 'Назва нового типу' (Name of the new type), 'Перевищення норми' (Norm violation), 'Майже максимум' (Near maximum), and 'Норма' (Norm). Below these are two more input fields: 'Майже мінімум' (Near minimum) and 'Нижче допустимого мінімуму' (Below the acceptable minimum). A 'Додати тип' (Add type) button is to the right. Below the form are two tables, one for 'Температура' (Temperature) and one for 'Вологість' (Humidity). Each table has a header with '#', 'Назва' (Name), and 'Змінити' (Change). The 'Температура' table has two rows: '1 Градус Цельсія (°C)' and '2 Фаренгейта (F)'. The 'Вологість' table has one row: '1 Відсотки (%)'. Each row has a pencil icon for editing and a trash icon for deleting.

Рисунок 3.22 — Сторінка «Типи приладів»

Тут адміністратор системи отримує CRUD права для керування типами приладів, одиницями вимірювання до кожного типу та порадами щодо регулювання показників стану повітря в приміщенні.

На рисунку 3.23 наведено форму редагування одиниці вимірювання.

The screenshot shows a dialog box titled 'Редагування одиниці вимірювання (Фаренгейта)'. The dialog has a close button in the top right. It contains three input fields: 'Одиниці вимірювання' (Measurement unit) with the value 'Фаренгейта', 'Позначка для типу' (Type label) with the value 'F', and 'Формула для переведення в даний тип' (Conversion formula) with the value '(c*9/5)+32'. At the bottom of the dialog are two buttons: 'Закрити' (Close) and 'Змінити' (Change).

Рисунок 3.23 — форму редагування одиниці вимірювання «Фаренгейт»

У формі редагування можна змінити назву типу, позначку для обраного типу та формулу для переведення показника стану повітря з основного показника стану повітря в обраний. У наведеній формулі відбувається переведення з градусів Цельсія в градуси Фаренгейта. Позначка «cf» у наведеній формулі використовується для позначення початкового значення.

Перейшовши до пункт меню «Адмін», адміністратор потрапляє в панель адміністратора WordPress.

На рисунку 3.24 наведено зображення панелі адміністратора WordPress.

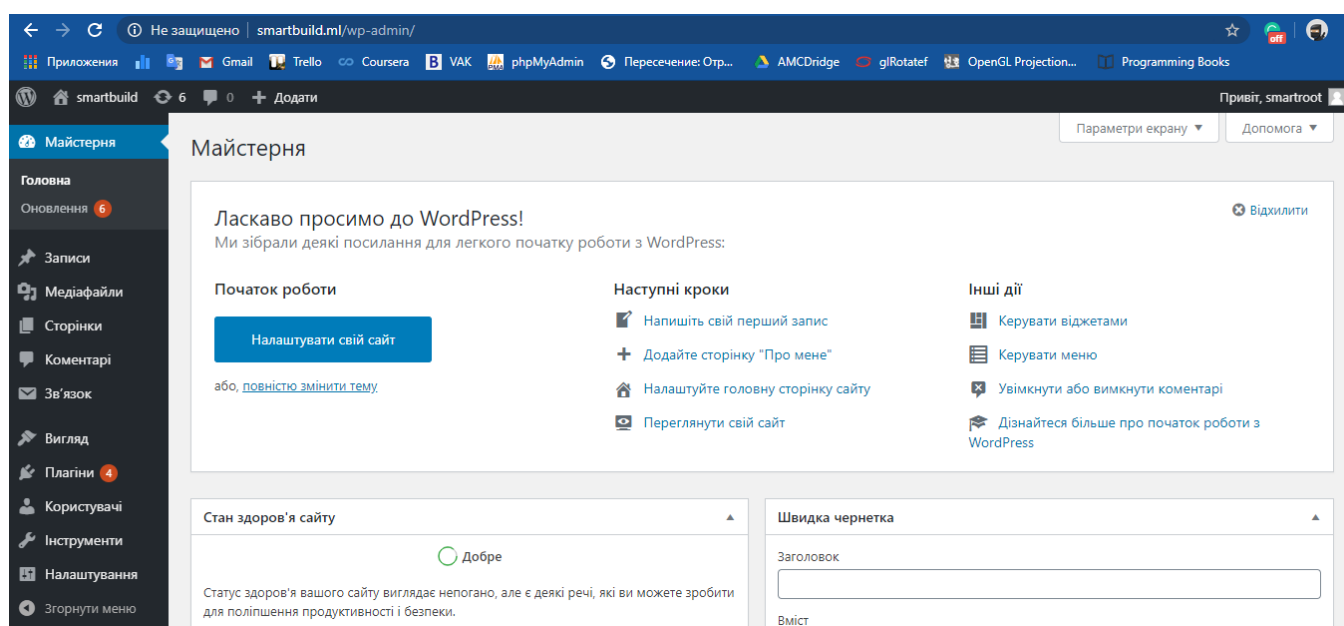


Рисунок 3.24 — Панель адміністратора WordPress

ВИСНОВКИ

У ході аналізу предметної області було досліджено літературні джерела з даної теми та існуючі системи-аналоги. Аналіз літературних джерел дав загальне уявлення про тенденції та темпи розвитку «розумних» систем. В результаті проведеного дослідження систем-аналогів, отримали чітке уявлення щодо їх можливостей, основних принципів роботи, переваг і недоліків.

На основі таблиці порівнянь систем-аналогів нами було сформовано вимоги до роботи системи моніторингу показників стану повітря в приміщенні та її окремих компонентів.

У ході вирішення задачі проектування роботи інформаційної системи було графічно описано процеси інформаційної системи у вигляді IDEF діаграми та створено діаграму варіантів використання (use-case). Також було проведено проектування бази даних та архітектури системи. Дані діаграми надають повне уявлення про структуру та механізм роботи інформаційної системи, що дає змогу перейти до розробки інформаційної системи.

Усі етапи виконання роботи були завершені згідно з календарним планом. Розробка інформаційної системи здійснювалась з використанням CMS WordPress.

Розроблена інформаційна система всім поставленим функціональним вимогам, а отже завдання було виконано в повному обсязі.

Результати роботи були представлені на міжнародній науково-технічній конференції «Інформатика, математика, автоматизація – 2020» (м. Суми) [20].

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Інтернет речей (Internet of Things, IoT) [Електронний ресурс] // IT-Enterprise – Режим доступу до ресурсу: <https://www.it.ua/knowledge-base/technology-innovation/internet-veschej-internet-of-things-iot>.
2. Розумний дім [Електронний ресурс] // Вікіпедія – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/%D0%A0%D0%BE%D0%B7%D1%83%D0%BC%D0%BD%D0%B8%D0%B9_%D0%B4%D1%96%D0%BC.
3. Dickson B. How to prevent your IoT devices from being forced into botnet bondage [Електронний ресурс] / Dickson. – 2015. – Режим доступу до ресурсу: <https://techcrunch.com/2016/08/16/how-to-prevent-your-iot-devices-from-beingforced-into-botnet-slavery/>.
4. Power Load Event Detection and Classification Based on Edge Symbol Analysis and Support Vector Machine [Електронний ресурс]. – 2014. – Режим доступу до ресурсу: <https://www.hindawi.com/journals/acisc/2012/742461/>.
5. An Overview of Home Automation Systems [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: <https://ieeexplore.ieee.org/document/7791223/>
6. Zipato — Make Your Home Smart [Електронний ресурс] – Режим доступу до ресурсу: <https://www.zipato.com/>.
7. Domintell — Smart Living Experience [Електронний ресурс] – Режим доступу до ресурсу: <https://www.domintell.com/en/>.
8. Ajax Systems | Офіційний сайт [Електронний ресурс] – Режим доступу до ресурсу: <https://ajax.systems/ua/>.
9. Nomi Smart Home — набір датчиків для вашого дома [Електронний ресурс] – Режим доступу до ресурсу: <http://nomi-electronics.com/nomi-smart-home-nabor-datchikov-dlya-vashego-doma/>.

10. Комплект системи безпеки Xiaomi Mi Smart Home Security Kit [Електронний ресурс] – Режим доступу до ресурсу: <https://miot.ua/nabir-datchikiv-smart-home-set-ytc4023cn/p17640/>.
11. IDEF0 Diagram [Електронний ресурс] // vitechcorp.com – Режим доступу до ресурсу: <http://www.vitechcorp.com/resources/core/onlinehelp/desktop/Views/IDEF0.htm>.
12. UML Use Case Diagrams [Електронний ресурс] – Режим доступу до ресурсу: <https://www.uml-diagrams.org/use-case-diagrams.html>.
13. Entity Relationship Diagram (ERD) - What is an ER Diagram? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.smartdraw.com/entity-relationship-diagram>.
14. Architecture Diagram Definition [Електронний ресурс] – Режим доступу до ресурсу: <https://www.dragon1.com/terms/architecture-diagram-definition>.
15. wpdb{ } [Електронний ресурс] // wp-кама – Режим доступу до ресурсу: <https://wp-kama.ru/function/wpdb>.
16. Theme Development [Електронний ресурс] – Режим доступу до ресурсу: https://codex.wordpress.org/Theme_Development.
17. Bootstrap | The most popular HTML, CSS, and JS library in the world. [Електронний ресурс] – Режим доступу до ресурсу: <https://getbootstrap.com>.
18. HTTP Methods GET vs POST [Електронний ресурс] // w3schools.com – Режим доступу до ресурсу: https://www.w3schools.com/tags/ref_httpmethods.asp.
19. Chart | Google Developers [Електронний ресурс] – Режим доступу до ресурсу: <https://developers.google.com/chart>.
20. Палажченко Є.В. Інформаційна система моніторингу та контролю показників стану повітря в приміщенні. / Палажченко Є.В., Ващенко С.М. // Інформатика, математика, автоматика: матеріали та програма науково-технічної конференції, м. Суми, 2020 р. – Суми : СумДУ, 2020. – 110 с.

Додаток А

ТЕХНІЧНЕ ЗАВДАННЯ на розробку інформаційної системи моніторингу показників стану повітря в приміщенні

Суми 2020

1 Призначення й мета створення інформаційної системи

1.1 Призначення інформаційної системи

Інформаційна система призначена для зберігання й візуалізації інформації про показники стану повітря в приміщенні, а також для віддаленої зміни станів активності приладів, що регулюють ці показники.

1.2 Мета створення інформаційної системи

Полегшення сприйняття інформації про показники стану повітря в приміщенні
Забезпечення зберігання та візуалізації та автоматизованого регулювання показників стану повітря в приміщенні.

1.3 Цільова аудиторія

У цільовій аудиторії інформаційної системи можна виділити наступні групи:

1. Власники квартир та будинків.
2. Приватні підприємці.
3. Фермери.

2 Вимоги до інформаційної системи

2.1 Вимоги до інформаційної системи в цілому

2.1.1 Вимоги до структури й функціонування інформаційної системи

Інформаційна система повинна бути реалізована як сайт, що є інструментом керування процесом надання послуг моніторингу показників стану повітря в приміщенні та перегляду даних, доступного в мережі Інтернет під доменним іменем www.smartbuild.ml. Сайт повинен складатися із взаємозалежних розділів із чітко розділеними функціями.

2.1.2 Вимоги до персоналу

Для підтримки сайту й експлуатації веб-інтерфейсу системи керування контентом (CMS) персонал має володіти базовими навичками роботи з персональним комп'ютером і веб-браузером.

2.1.3 Вимоги до збереження інформації

Дані про користувачів та про показники стану повітря користувачів мають зберігатися в системі до тих пір доки не буде видалено користувача. Також має бути

передбачена система щоденного резервного копіювання даних для забезпечення швидкого відновлення даних у випадку.

2.1.4 Вимоги до розмежування доступу

Інформація, розташовувана на сайті, є загальнодоступною.

Користувачів сайту можна розділити на 3 групи відповідно до прав доступу:

- Відвідувачі
- Користувачі
- Адміністратор

Відвідувачі мають доступ тільки до загальнодоступної частини сайту (головна сторінка та сторінка новин сайту). Також відвідувачі можуть подати заявку на реєстрацію в системі.

Користувачі мають доступ до особистого кабінету, де можуть керувати своїми датчиками. Користувачі можуть переглядати інформацію про показники стану повітря в приміщенні та дані про періоди активності приладів. Перехід до сторінки перегляду інформації з приладів має здійснюватися з особистого кабінету користувача.

Адміністратор може керувати користувачами сайту та здійснювати класифікацію, розподілення та привласнення датчиків користувачам. Адміністратор має доступ до особистого кабінету всіх користувачів і, відповідно, до всіх приладів користувачів. Адміністратор повинен мати можливість перейти до панелі адміністратора WordPress без додаткової авторизації.

Доступ до адміністративної та клієнтської частини сайту повинен здійснюватися з використанням унікального логіна й пароля.

2.2 Вимоги до функцій, виконуваних сайтом

2.2.1 Основні вимоги

2.2.1.1 Структура сайту

Клієнтська частина сайту повинна складатися з наступних розділів:

Головна – виводиться довідкова інформацію про інформаційну систему.

Новини – категорія з виведенням новин сайту.

Особистий кабінет – перегляд датчиків користувача.

Адміністративна частина сайту повинна складатися з наступних розділів:

Головна – виводиться довідкова інформацію про інформаційну систему.

Новини – категорія з виведенням новин сайту.

Користувачі – перегляд списку користувачів з можливістю перейти в особистий кабінет користувача.

Типи приладів – сторінка керування типами приладів.

Панель адміністратора – перехід до панелі адміністратора WordPress.

Клієнтська частина сайту повинна складатися з наступних розділів:

Головна – виводиться довідкова інформацію про інформаційну систему.

Новини – категорія з виведенням новин сайту.

2.2.1.2 Навігація

Користувацький інтерфейс сайту повинен забезпечувати наочне, інтуїтивно зрозуміле представлення структури розміщеної на ньому інформації, швидкий і логічний перехід до розділів і сторінок. Навігаційні елементи повинні забезпечувати однозначне розуміння користувачем їх змісту: посилання на сторінки повинні мати заголовок, умовні позначки, що відповідають загальноприйнятим. Графічні елементи навігації повинні мати альтернативний підпис.

Система повинна забезпечувати навігацію по всіх доступних користувачеві ресурсам і відображати відповідну інформацію. Навігації повинна бути реалізована у формі контент-меню. Меню повинне являти собою список гіперпосилань у верхній частині сторінки.

2.2.1.3 Наповнення сайту (контент)

Сторінки всіх розділів сайту повинні формуватися програмним шляхом на підставі інформації з бази даних на сервері.

Модифікація вмісту розділів повинна здійснюватися за допомогою адміністративної частини сайту, який без застосування спеціальних навичок програмування (без використання програмування й спеціального кодування або форматування) повинен передбачати можливість редагування інформаційного вмісту сторінок сайту. Наповнення інформацією повинне проводитися з використанням шаблонів сторінок сайту.

2.2.1.4 Система навігації (карта сайту)

Взаємозв'язок між розділами й підрозділами сайту (карта сайту) для відвідувача представлено на рисунку А.1.

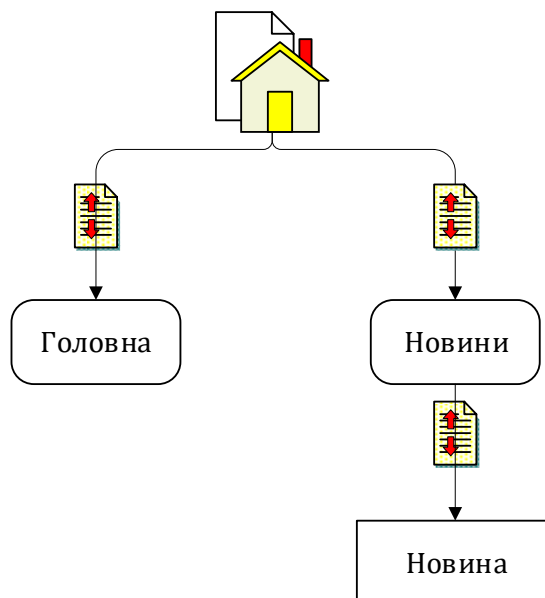


Рисунок А.1 – Карта сайту для відвідувача

Взаємозв'язок між розділами й підрозділами сайту (карта сайту) для користувача представлено на рисунку А.2.

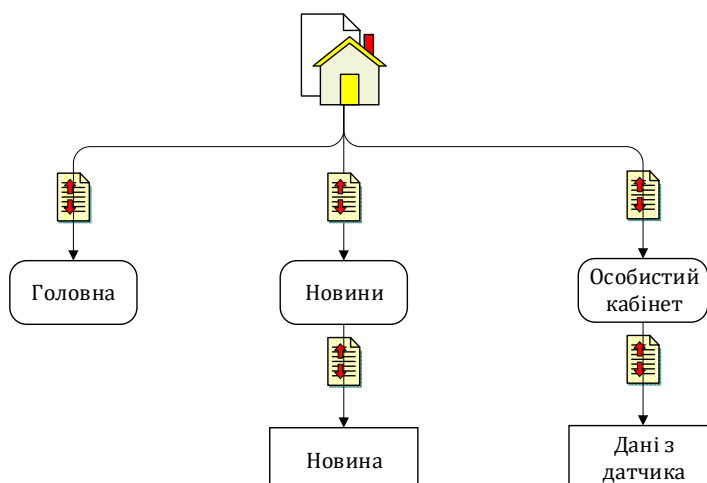


Рисунок А.2 – Карта сайту для користувача

Взаємозв'язок між розділами й підрозділами сайту (карта сайту) для адміністратора представлено на рисунку А.3.

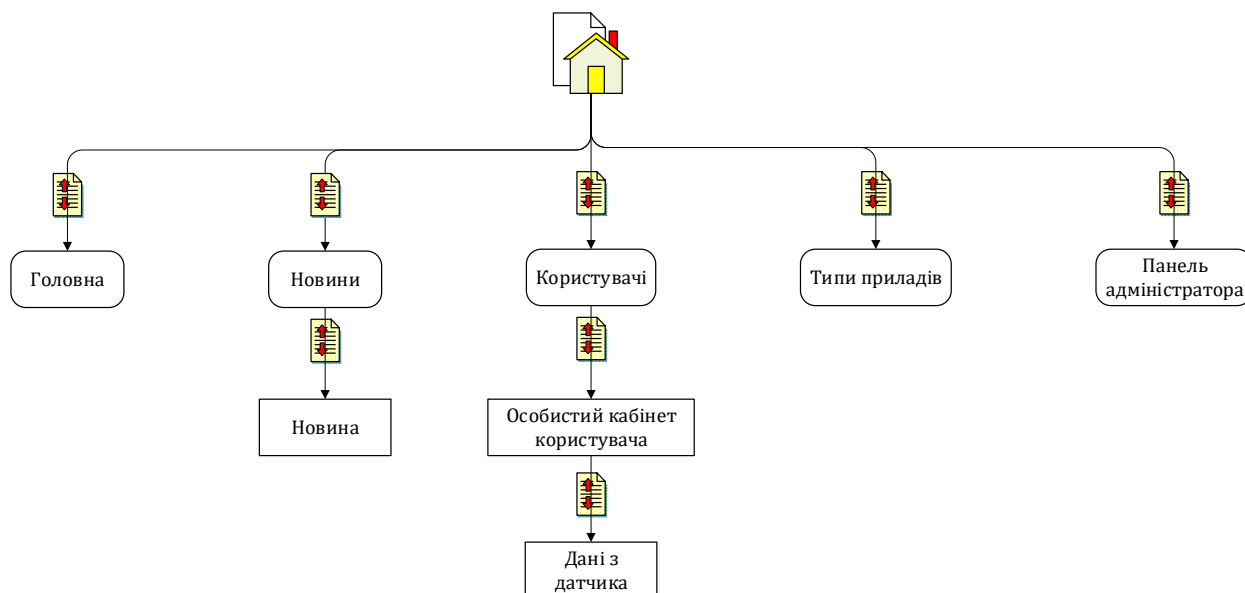


Рисунок А.3 – Карта сайту для адміністратора

2.2.2 Вимоги до функціональних можливостей

Система керування контентом (адміністративна частина сайту) повинна надавати можливість додавання, редагування й видалення вмісту статичних і динамічних сторінок сайту. Реалізувати гнучкий конструктор для типів датчиків. Користувач повинен мати можливість керувати своїм особистим кабінетом. Також повинна бути передбачена можливість додавання записів в розділ «Новини» без відображення на сайті.

2.2.2.1 Функціональні можливості розділів

Головна сторінка буде носити виключно інформативний характер.

В категорії «Новини» будуть висвітлюватися новини сайту.

В розділі «Особистий кабінет» користувач може переглядати та групувати свої прилади.

В розділі «Користувачі» адміністратор повинен мати можливість перейти до особистого кабінету користувачів та керувати модулями що не належать жодному з користувачів.

2.2.2.2 Загальні вимоги

Сайт повинен мати легкий і простий дизайн, а також світлий фон на якому буде добре видно текст та графічні елементи.

Інформація на сайті має подаватись у зручній для сприйняття формі. Інформація про показники стану повітря в приміщенні має відображатися у вигляді графіка. Для оформлення використовувати стандартні кольори бібліотеки bootstrap.

Всі сторінки сайту створені за шаблоном і мають однаковий дизайн. Змінюється тільки контент, а шапка та підвал сайту залишаються незмінними.

Розміщення основних елементів на типовій сторінці сайту схематично показано на рисунку А.4.

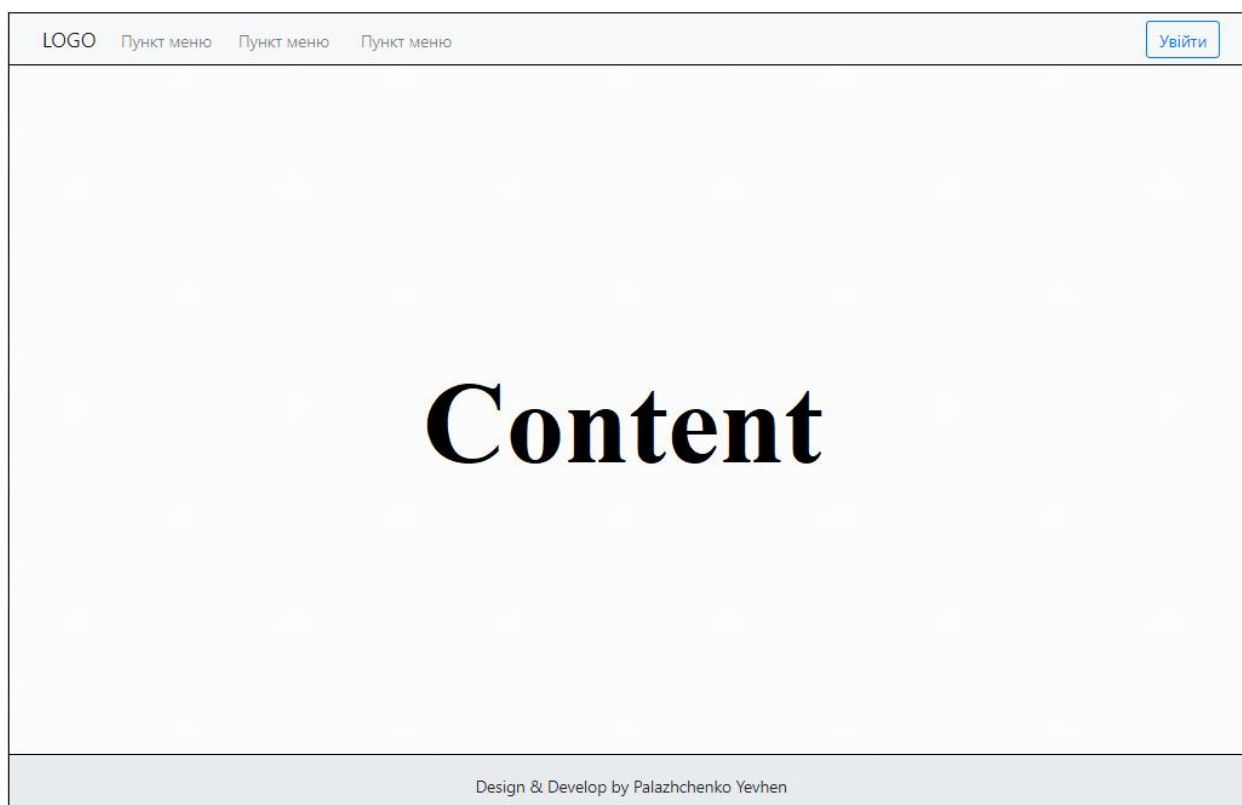


Рисунок А.4 – Типова сторінка

2.2.2.3 Типові навігаційні й інформаційні елементи

1. Шапка сайту
2. Основне поле контенту
3. Підвал сайту

2.2.2.4 Шапка сайту

Шапка сайту повинна містити логотип, назву сайту та меню. Логотип є посиланням на головну сторінку сайту.

2.2.2.5 Основне поле контенту

Основне поле контенту повинне розташовуватися в центрі сторінки. У цьому полі відображається основний зміст обраного розділу. Стильове оформлення матеріалів і їх елементів має відповідати загальним вимогам до стилістичного оформлення сторінок інформаційної системи.

2.3 Вимоги до видів забезпечення

2.3.1 Вимоги до інформаційного забезпечення

Реалізація сайту відбувається з використанням:

- Apache 2.4
- PHP 7.2
- MySQL 5.6
- WordPress 5.4
- PHPMyAdmin
- Bootstrap
- CSS3
- HTML5
- JavaScript
- Google Developers Carts

2.3.2 Вимоги до лінгвістичного забезпечення

Сайт повинен бути виконаний українською мовою.

2.3.3 Вимоги до програмного забезпечення

Програмне забезпечення клієнтської частини повинне задовольняти наступним вимогам:

- Веб-браузер: Internet Explorer 7.0 і вище, або Firefox 3.5 і вище, або Opera 9.5 і вище, або Safari 3.2.1 і вище, або Chrome 2 і вище;
- Включена підтримка javascript, Flash і cookies.

2.3.4 Вимоги до апаратного забезпечення

Апаратне забезпечення серверної частини повинне задовольняти наступним вимогам:

- Веб-сервер Apache з модулем MOD_REWRITE;
- Не менш 1000 МБ вільного місця на диску.

Апаратне забезпечення клієнтської частини повинне забезпечувати підтримку програмного забезпечення клієнтської частини, зазначеного в пункті 2.2.3

3 Склад і зміст робіт зі створення сайту

Докладний опис етапів роботи зі створення сайту наведено в таблиці А.1.

Таблиця А.1 – Етапи створення сайту

№	Склад і зміст робіт	Строк розробки (у робочих днях)
1	Аналіз об'єктів	1
2	Розробка ERD діаграми	1
3	Розробка бази даних	3
4	Встановлення та налаштування CMS WordPress	1
5	Встановлення та налаштування необхідних плагінів	1
6	Створення шаблону теми	2
7	Кастомізація системи реєстрації та входу WordPress	1
8	Захист від SQL-ін'єкцій	4
9	Розробка шаблонів унікальних сторінок	15
	Загальна тривалість робіт (з урахуванням резервного строку на налагодження й виправлення помилок) і строк закінчення проекту	29

4 Вимоги до складу й змісту робіт із введення сайту в експлуатацію

Для створення умов функціонування, при яких гарантується відповідність створюваного сайту вимогам ТЗ і можливість його ефективної роботи, в організації.

Для переносу сайту на хостинг необхідно, щоб параметри хостингу відповідали вимогам, зазначеним у ТЗ. На хостинг переноситься програма (сайт), зверстаний шаблон дизайну й структура й наповнення бази даних з подальшою їх доробкою

Додаток Б

Планування робіт

Деталізація мети проекту методом SMART.

Мета проекту: розробити інформаційну систему моніторингу та регулювання показників стану повітря в приміщенні.

Деталізація мети проекту методом SMART наведена в таблиці Б.1.

Таблиця Б.1 – Деталізація мети методом SMART

Specific (конкретна)	Створити інформаційну систему моніторингу та регулювання показників стану повітря в приміщенні
Measurable (вимірювана)	Головним показником успішності виконання даного проекту є оцінка.
Achievable (досяжна)	Ціль є досяжною оскільки технології, за допомогою яких буде розроблятися додаток є у відкритому доступі, а мета була повністю погоджена із замовником.
Relevant (реалістична)	Розробка даного додатку є цілком реалістичною, оскільки розробники мають усі необхідні програмні та технічні засоби, а також достатній рівень кваліфікації.
Time-framed (обмежена у часі)	Ціль має часове обмеження. Всі строки виконання даного додатку вказано в календарному плані.

Планування змісту структури робіт. Для планування змісту структури робіт в графічному вигляді використовують WBS діаграми. На верхньому першому рівні WBS фіксується продукт проекту. Наступний II рівень відповідає діям або основним

заходам для досягнення продукту проекту. Потім триває розбивка цих дій доти, поки не відбувається виконання дій елементарних робіт.

Побудуємо структуру WBS, у якій детально опишемо роботи, які потрібно виконати на кожному етапі створення проекту. Декомповована діаграма WBS зображена на рис. Б.1.

Планування структури організації, для впровадження готового проекту (OBS). Після побудови WBS розробимо організаційну структуру виконавців OBS. Визначається за переліком пакетів робіт нижнього рівня кожної гілки WBS-структури. Відповідальні – це ті співробітники, які безпосередньо організують і відповідають у виконавця за виконання елементарної роботи, зазначеної у WBS. Діаграма OBS зображена на рис. Б.2. Список виконавців, що функціонують в проекті знаходиться в табл. Б.2.

Таблиця Б.2 – Виконавці проекту

Роль	Ім'я	Проектна роль
Розробник	Палажченко Є.В.	Виконує розробку основного функціоналу проекту та інтерфейс користувача
Проектувальник	Палажченко Є.В.	Проектує 3D моделі та елементи тривимірної графіки, розробляє дизайн програми.
Консультант проекту	Парфененко Ю.В. Шендрік В. В. Ващенко С.М.	Формує завдання на розробку проекту.
Менеджер проекту	Палажченко Є.В.	Відповідає за виконання термінів, розподіл ресурсів та завдань між учасниками. Виконує збір та аналіз даних.

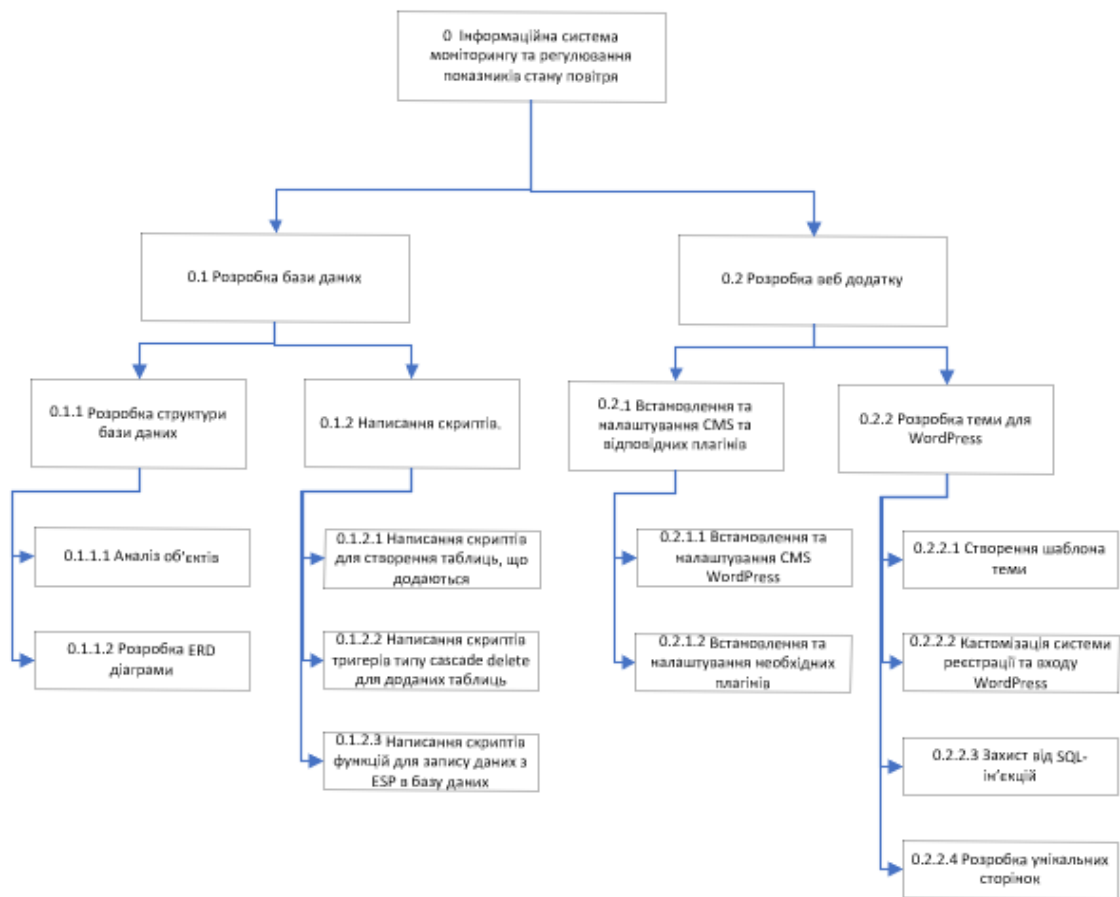


Рисунок Б.1 – WBS. Структура робіт проекту

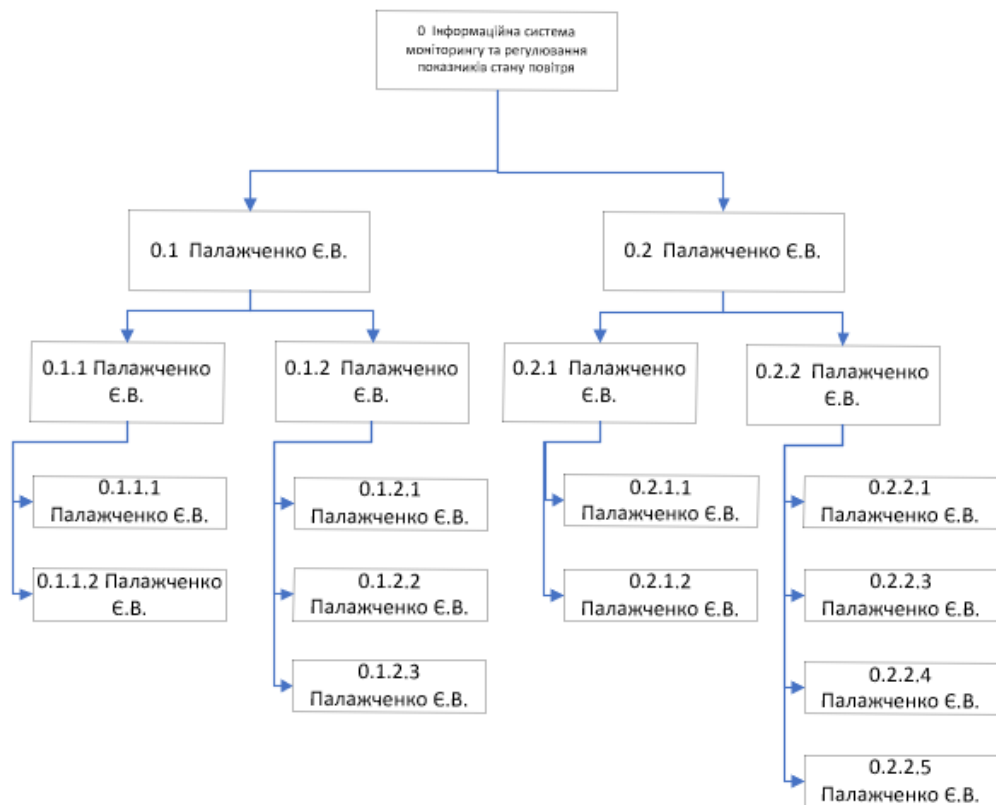


Рисунок Б.2 – Організаційна структура проекту (OBS)

Діаграма Ганта. Для того щоб мати реальне уявлення про тривалість виконання робіт з урахуванням обмеженості у використанні ресурсів, на підставі часткової мережевої моделі, а також, проекту в цілому з урахуванням вихідних та святкових днів, будують календарний графік робіт (Рисунок. Б.3- Рисунок. Б.4). Він є реальним розподілом робіт з пакету за календарними датами, тобто своєрідним розкладом виконання робіт. Графік Ганта є достатньо зручним для користування. Його будують так. На горизонталі фіксують календар у тих одиницях часу, які обрані для проекту (години, дні). Ліворуч на вертикалі розташовують найменування всіх робіт. На полі, що утворилось, поставляють у вигляді прямокутників роботи, довжина яких по горизонталі відповідає їхній тривалості. Між роботами лініями вказують логічні зв'язки.

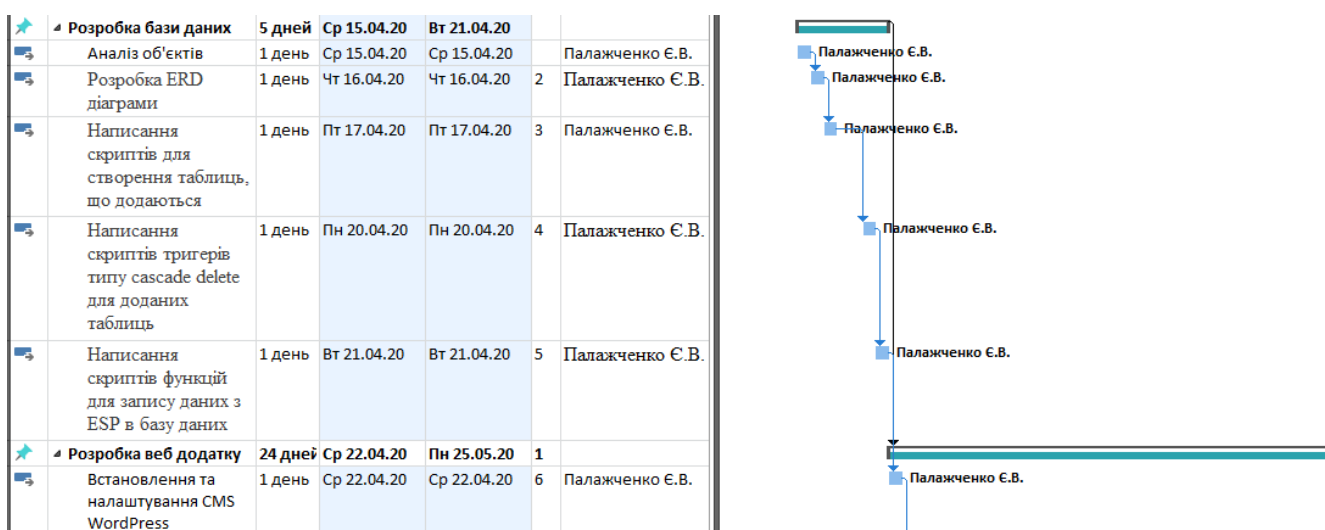


Рисунок Б.3 – Діаграма Ганта

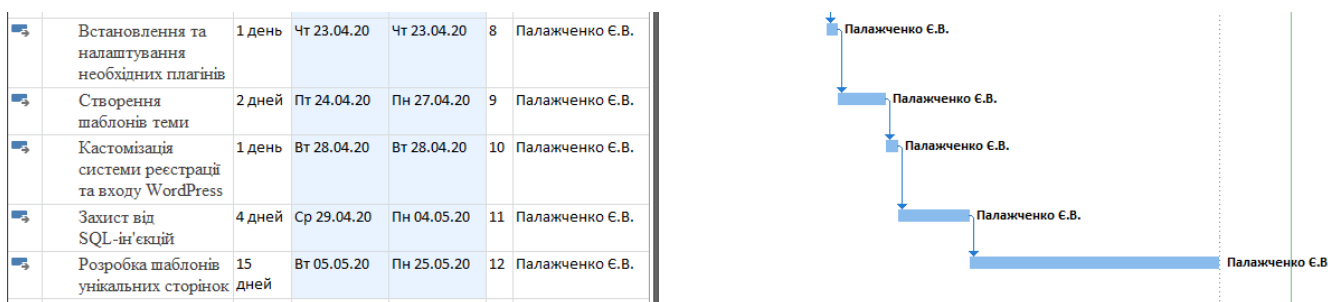


Рисунок Б.4 – Продовження діаграми Ганта

Аналіз ризиків. Виконаємо якісну і кількісну оцінку ризиків роботи. При якісній оцінці визначимо ризики, що потребують швидкого реагування. Така оцінка визначить ступінь важливості ризику і дозволить вибрати спосіб реагування. Кількісна оцінка ризиків буде виконана для більш повної ідентифікації ризиків та ступеня їхнього впливу на виконання проекту. Кількісна і якісна оцінка ризиків можуть використовуватися окремо або разом, залежно від наявного часу і бюджету, необхідності в кількісній або якісній оцінці ризиків. У табл. Б.5 знаходиться класифікація ризиків за показниками ймовірності виникнення ризику та величині втрат.

Далі виконаємо планування реагування на ризики — це розробка методів і технологій зниження негативного впливу ризиків на проект. Визначимо ефективність розробки реагування на проект, визначимо чи будуть наслідки впливу ризику на проект позитивними або негативним. Оцінюємо ризики за показниками, що знаходяться в табл. Б.3. На основі оцінки будемо матрицю ймовірності виникнення ризиків та впливу ризику, що зображена на рис. Б.5.

Таблиця Б.3 – Шкала оцінювання ймовірності виникнення та впливу ризику на виконання проекту

Оцінка	Ймовірність виникнення	Вплив ризику
1	Низька	Низький
2	Середня	Середній
3	Висока	Високий

Ймовірність виникнення	3	RS_2	RS_3,	RS_5, RS_9
	2	RS_1, RS_13	RS_4, RS_6	RS_7, RS_14
	1	RS_12	RS_8, RS_11	RS_10, RS_15
		1	2	3
Вплив ризику				

Рисунок Б.5 – Матриця ймовірності виникнення ризиків та впливу ризику

- зелений колір – прийнятні ризики;
- жовтий колір – виправданні ризики;
- червоний колір – недопустимі ризики.

На підставі отриманого значення індексу ризику класифікують: за рівнем ризику, що знаходиться в табл. Б.4.

Таблиця Б.4 – Шкала оцінювання за рівнем ризику

№	Назва	Межі	Ризики, які входять(номера)
1	Прийнятні	$1 \leq R \leq 2$	1,8,11,12,13
2	Виправдані	$3 \leq R \leq 4$	2,4,6,10,15
3	Недопустимі	$6 \leq R \leq 9$	3,5,7,9,14

Таблиця Б.5 – Оцінка ймовірності виникнення, величини витрат та індексу ризику

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
RS_1	Відкритий	Непорозуміння між розробником та замовником	Низька	Середній	3	Проведення зустрічей для підвищення взаєморозуміння	Попередження	З'ясувати причини непорозуміння та обговорити варіанти вирішення.
RS_2	Відкритий	Вихід на ринок продукту конкурентів до виходу нашого	Низька	Високий	3	Провести попереднє дослідження альтернативних продуктів.	Прийняття	Обрати унікальну стратегію розробки
Rs_3	Відкритий	Перевитрата через затримку постачання чи постачання бракованого товару	Середня	Високий	7	Обрати надійного постачальника	Попередження	Знайти надійного постачальника з гарантією
RS_4	Відкритий	Погано прописане завдання на розробку	Середня	Високий	7	1. Ясно і однозначно обговорити із замовником усі види вимог.	Попередження	Скласти глосарій для запобігання розбіжностей у розумінні слів та термінів.
RS_5	Відкритий	Низька кваліфікація розробників проекту	Середня	Середній	4	Підвищити кваліфікацію персоналу. Переглянути онлайн-ресурси для підвищення рівня знань.	Пом'якшення	Врахувати час на підготовку працівників. Видати літературу, переглянути онлайн-уроки.

Продовження таблиці Б.5

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
RS_6	Відкритий	Неоптимальний розподіл часу	Висока	Високий	9	Провести аналіз актуальності найважливіших процесів та робіт та чітко дотримуватися календарного плану.	Пом'якшення	Знайти способи оптимізації роботи із вже існуючою розстановкою. Обговорити варіанти внесення правок замовником.
RS_7	Закритий	Втрата даних через збої в роботі сервера чи бази даних	Низька	Середній	4	Покривається за рахунок вибору надійного хостинг провайдера	Попередження	

Додаток В

Скрипти створення бази даних

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET AUTOCOMMIT = 0;
START TRANSACTION;
SET time_zone = "+00:00";

--
-- Структура таблиці `advice`
--
CREATE TABLE `advice` (
  `adv_id` bigint(20) NOT NULL,
  `type_id` bigint(20) NOT NULL,
  `exceeded` text NOT NULL,
  `almost_max` text NOT NULL,
  `normal` text NOT NULL,
  `almost_min` text NOT NULL,
  `lowered` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_520_ci;

--
-- Структура таблиці `data`
--
CREATE TABLE `data` (
  `data_id` bigint(20) UNSIGNED NOT NULL,
  `esp_id` bigint(20) UNSIGNED NOT NULL,
  `dt` datetime NOT NULL,
  `value` double NOT NULL DEFAULT 0
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_520_ci;

--
-- Структура таблиці `device`
--
CREATE TABLE `device` (
  `dev_id` bigint(20) UNSIGNED NOT NULL,
  `gr_id` bigint(20) UNSIGNED NOT NULL,
  `name` varchar(60) COLLATE utf8mb4_unicode_520_ci NOT NULL,
  `type` varchar(60) COLLATE utf8mb4_unicode_520_ci NOT NULL,
  `_status` tinyint(1) NOT NULL DEFAULT 0
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_520_ci;

--
-- Структура таблиці `esp`
--
CREATE TABLE `esp` (
  `esp_id` bigint(20) UNSIGNED NOT NULL,
```

```

`mac` varchar(40) COLLATE utf8mb4_unicode_520_ci NOT NULL,
`gr_id` bigint(20) UNSIGNED NOT NULL,
`type_id` bigint(20) UNSIGNED NOT NULL,
`name` varchar(40) COLLATE utf8mb4_unicode_520_ci NOT NULL DEFAULT
't',
`max_` bigint(20) NOT NULL DEFAULT 0,
`min_` bigint(20) NOT NULL DEFAULT 0,
`need` bigint(20) NOT NULL DEFAULT 0
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_520_ci;

--
-- Структура таблиці `group`
--
CREATE TABLE `group` (
  `gr_id` bigint(20) UNSIGNED NOT NULL,
  `us_id` bigint(20) UNSIGNED NOT NULL,
  `name` varchar(80) COLLATE utf8mb4_unicode_520_ci NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_520_ci;

--
-- Структура таблиці `info`
--
CREATE TABLE `info` (
  `inf_id` bigint(20) UNSIGNED NOT NULL,
  `dev_id` bigint(20) UNSIGNED NOT NULL,
  `start_` datetime NOT NULL,
  `end_` datetime DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_520_ci;

--
-- Структура таблиці `type`
--
CREATE TABLE `type` (
  `type_id` bigint(20) UNSIGNED NOT NULL,
  `name` varchar(20) COLLATE utf8mb4_unicode_520_ci NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_520_ci;

--
-- Структура таблиці `units`
--
CREATE TABLE `units` (
  `unit_id` bigint(20) UNSIGNED NOT NULL,
  `type_id` bigint(20) UNSIGNED NOT NULL,
  `unit` varchar(30) COLLATE utf8mb4_unicode_520_ci NOT NULL,
  `mark` varchar(10) COLLATE utf8mb4_unicode_520_ci NOT NULL,
  `formula` varchar(20) COLLATE utf8mb4_unicode_520_ci NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_520_ci;

```

```
--  
-- Індокси таблиці `advice`  
--  
ALTER TABLE `advice`  
  ADD PRIMARY KEY (`adv_id`);  
  
--  
-- Індокси таблиці `data`  
--  
ALTER TABLE `data`  
  ADD PRIMARY KEY (`data_id`),  
  ADD KEY `esp_id` (`esp_id`),  
  ADD KEY `dt` (`dt`),  
  ADD KEY `value_` (`value_`);  
  
--  
-- Індокси таблиці `device`  
--  
ALTER TABLE `device`  
  ADD PRIMARY KEY (`dev_id`),  
  ADD KEY `gr_id` (`gr_id`);  
  
--  
-- Індокси таблиці `esp`  
--  
ALTER TABLE `esp`  
  ADD PRIMARY KEY (`esp_id`),  
  ADD KEY `gr_id` (`gr_id`),  
  ADD KEY `type_id` (`type_id`);  
  
--  
-- Індокси таблиці `group`  
--  
ALTER TABLE `group`  
  ADD PRIMARY KEY (`gr_id`),  
  ADD KEY `us_id` (`us_id`);  
  
--  
-- Індокси таблиці `info`  
--  
ALTER TABLE `info`  
  ADD PRIMARY KEY (`inf_id`),  
  ADD KEY `div_id` (`dev_id`),  
  ADD KEY `start` (`start_`),  
  ADD KEY `end` (`end_`);  
  
--  
-- Індокси таблиці `units`  
--  
ALTER TABLE `units`  
  ADD PRIMARY KEY (`unit_id`),  
  ADD KEY `type_id` (`type_id`);
```

```
--  
-- AUTO_INCREMENT для таблиці `advice`  
--  
ALTER TABLE `advice`  
  MODIFY `adv_id` bigint(20) NOT NULL AUTO_INCREMENT,  
  AUTO_INCREMENT=0;  
  
--  
-- AUTO_INCREMENT для таблиці `data`  
--  
ALTER TABLE `data`  
  MODIFY `data_id` bigint(20) UNSIGNED NOT NULL AUTO_INCREMENT,  
  AUTO_INCREMENT=0;  
  
--  
-- AUTO_INCREMENT для таблиці `device`  
--  
ALTER TABLE `device`  
  MODIFY `dev_id` bigint(20) UNSIGNED NOT NULL AUTO_INCREMENT,  
  AUTO_INCREMENT=0;  
--  
-- AUTO_INCREMENT для таблиці `esp`  
--  
ALTER TABLE `esp`  
  MODIFY `esp_id` bigint(20) UNSIGNED NOT NULL AUTO_INCREMENT,  
  AUTO_INCREMENT=0;  
--  
-- AUTO_INCREMENT для таблиці `group`  
--  
ALTER TABLE `group`  
  MODIFY `gr_id` bigint(20) UNSIGNED NOT NULL AUTO_INCREMENT,  
  AUTO_INCREMENT=0;  
--  
-- AUTO_INCREMENT для таблиці `info`  
--  
ALTER TABLE `info`  
  MODIFY `inf_id` bigint(20) UNSIGNED NOT NULL AUTO_INCREMENT,  
  AUTO_INCREMENT=0;  
--  
-- AUTO_INCREMENT для таблиці `type`  
--  
ALTER TABLE `type`  
  MODIFY `type_id` bigint(20) UNSIGNED NOT NULL AUTO_INCREMENT,  
  AUTO_INCREMENT=0;  
--  
-- AUTO_INCREMENT для таблиці `units`  
--  
ALTER TABLE `units`  
  MODIFY `unit_id` bigint(20) UNSIGNED NOT NULL AUTO_INCREMENT,  
  AUTO_INCREMENT=0;  
COMMIT;
```

Додаток Г

Програмний код основних файлів теми

header.php

```

<?php session_start(); ?>
<!DOCTYPE html>
<html <?php language_attributes(); ?>>
<head>
    <meta charset="<?php bloginfo( 'charset' ); ?>">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="shortcut icon" href="http://smartbuild.ml/wp-
content/themes/smartbuild/img/icon.png" type="image/x-icon">
    <!--[if lt IE 9]>
        <script
src="//html5shiv.googlecode.com/svn/trunk/html5.js"></script>
    <![endif]-->
    <?php wp_head(); ?>
</head>
<body <?php body_class(); ?>>
    <div class="container">
        <header>
            <div class="row bg-light">
                <div class="col-md-12">
                    <nav class="navbar navbar-expand-sm navbar-light">
                        <a class="navbar-brand" href="http://smartbuild.ml"> </a>
                        <button class="navbar-toggler" type="button" data-
toggle="collapse" data-target="#navbarNavDropdown" aria-
controls="navbarNavDropdown" aria-expanded="false" aria-label="Toggle
navigation">
                            <span class="navbar-toggler-icon"></span>
                        </button>
                        <div class="collapse navbar-collapse"
id="navbarNavDropdown">
                            <?php
                                if ( is_user_logged_in() ){
                                    if(curent_user_is_admin()){
                                        $args = array(
                                            'theme_location' => 'admin',
                                            'container'=> false,
                                            'menu_id' => 'bottom-nav',
                                            'items_wrap' => '<ul id="%1$s"
class="navbar-nav %2$s">%3$s</ul>',
                                            'menu_class' => 'top-menu',
                                            'walker' => new
bootstrap_menu(true)
                                        );

```



```

        } else {
            $args = array(
                'theme_location' => 'custom',
                'container'=> false,
                'menu_id' => 'bottom-nav',
                'items_wrap' => '<ul id="%1$s"
class="navbar-nav %2$s">%3$s</ul>',
                'menu_class' => 'top-menu',
                'walker' => new
            bootstrap_menu(true)
        );
        wp_nav_menu($args);
    }
    } else {
        $args = array(
                'theme_location' => 'top',
                'container'=> false,
                'menu_id' => 'top-nav-ul',
                'items_wrap' => '<ul id="%1$s"
class="navbar-nav %2$s">%3$s</ul>',
                'menu_class' => 'top-menu',
                'walker' => new
            bootstrap_menu(true)
        );
        wp_nav_menu($args);
    }
}
?>
</div>
<?php if ( is_user_logged_in() ){ echo "<a class=\"btn
btn btn-outline-primary\" href=\"".wp_logout_url().">Вийти</a>"; }
else {?>
    <button type="button" class="btn btn btn-outline-primary"
data-toggle="modal" data-target="#exampleModal">Увійти</button>
    <?php }?>
</nav>
</div>
</div>
</header>

```

footer.php

```

<footer>
    <div class="row">
        <div class="col-12 text-center py-3">
            Design & Develop by Palazhchenko Yevhen
        </div>
    </div>
</footer>
<?php if(!is_user_logged_in()){ ?>
<!-- Modal -->

```

```

<div class="modal fade" id="exampleModal" tabindex="-1" role="dialog"
aria-labelledby="exampleModalLabel" aria-hidden="true">
  <div class="modal-dialog" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title"
id="exampleModalLabel">Авторизація/Реєстрація</h5>
        <button type="button" class="close" data-
dismiss="modal" aria-label="Close">
          <span aria-hidden="true">&times;</span>
        </button>
      </div>
      <div class="modal-body">
        <nav>
          <div class="nav nav-tabs" id="nav-tab"
role="tablist">
            <a class="nav-item nav-link active"
id="nav-home-tab" data-toggle="tab" href="#nav-home" role="tab" aria-
controls="nav-home" aria-selected="true">Вхід</a>
            <a class="nav-item nav-link" id="nav-
profile-tab" data-toggle="tab" href="#nav-profile" role="tab" aria-
controls="nav-profile" aria-selected="false">Реєстрація</a>
          </div>
        </nav>
        <div class="tab-content" id="nav-tabContent">
          <div class="tab-pane fade show active"
id="nav-home" role="tabpanel" aria-labelledby="nav-home-tab">
            <form class="w-100 mt-4" name="loginform"
id="loginform" action="http://smartbuild.ml/wp-login.php"
method="post">
              <p class="w-100">
                <label for="user_login"
class="w-100">Ім'я користувача або Email адреса<br>
                <input type="text"
name="log" id="user_login" class="input form-control" value=""
size="20" autocapitalize="off"></label>
              </p>
              <p>
                <label for="user_pass"
class="w-100">Пароль<br>
                <input type="password"
name="pwd" id="user_pass" class="input form-control" value=""
size="20"></label>
              </p>
              <p class="forgetmenot"><label
for="rememberme"><input name="rememberme" type="checkbox"
id="rememberme" value="forever"> Пам'ятати мене</label></p>
              <p class="submit">
                <input type="submit" name="wp-
submit" id="wp-submit" class="btn btn-outline-success" value="Увійти">
                <input type="hidden"
name="redirect_to" value="http://smartbuild.ml/account/">
            </form>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```

        <input type="hidden"
name="testcookie" value="1">
    </p>
    </form>
</div>
    <div class="tab-pane fade" id="nav-profile"
role="tabpanel" aria-labelledby="nav-profile-tab">
        <div role="form" class="wpcf7 w-100 mt-4"
id="wpcf7-f14-p9-o1" lang="uk" dir="ltr">
            <div class="screen-reader-
response"></div>
                <form action="/login/#wpcf7-f14-p9-
o1" method="post" class="wpcf7-form" novalidate="novalidate">
                    <div style="display: none;">
                        <input type="hidden"
name="_wpcf7" value="14">
                            <input type="hidden"
name="_wpcf7_version" value="5.1.4">
                                <input type="hidden"
name="_wpcf7_locale" value="uk">
                                    <input type="hidden"
name="_wpcf7_unit_tag" value="wpcf7-f14-p9-o1">
                                        <input type="hidden"
name="_wpcf7_container_post" value="9">
                                            </div>
                                                <p><label class="w-100"> Ваше
ім'я (обов'язково)<br>
                                                    <span class="wpcf7-
form-control-wrap your-name"><input type="text" name="your-name"
value="" size="40" class="wpcf7-form-control wpcf7-text wpcf7-
validates-as-required form-control" aria-required="true" aria-
invalid="false" placeholder="Введіть Ваше ім'я"></span> </label><br>
                                                        <label class="w-100"> Ваш
email (обов'язково)<br>
                                                            <span class="wpcf7-
form-control-wrap your-email"><input type="email" name="your-email"
value="" size="40" class="wpcf7-form-control wpcf7-text wpcf7-email
wpcf7-validates-as-required wpcf7-validates-as-email form-control"
aria-required="true" aria-invalid="false" placeholder="Введіть Ваш
email"></span> </label><br>
                                                                <label class="w-100"> Ваш
номер (обов'язково)<br>
                                                                    <span class="wpcf7-
form-control-wrap tel-248"><input type="tel" name="tel-248" value=""
size="40" class="wpcf7-form-control wpcf7-text wpcf7-tel wpcf7-
validates-as-required wpcf7-validates-as-tel form-control" aria-
required="true" aria-invalid="false" placeholder="Введіть Ваш
номер"></span></label></p>
                                                                    <p><span id="wpcf7-
5dc6f7fa8f3cd" class="wpcf7-form-control-wrap honeypot-796-wrap"
style="display:none !important; visibility:hidden !important;"><label
class="hp-message">Please leave this field empty.</label><input

```

```

class="wpcf7-form-control wpcf7-text" type="text" name="honeypot-796"
value="" size="40" tabindex="-1" autocomplete="nope"></span><br>
        <input type="submit"
value="Відправити" class="wpcf7-form-control wpcf7-submit btn btn-
outline-success"><span class="ajax-loader"></span></p>
        <div class="wpcf7-response-
output wpcf7-display-none"></div>
    </form>
</div>
</div>
</div>
</div>
<div class="modal-footer">
    <a href="http://smartbuild.ml/wp-
login.php?action=lostpassword">Втратили свій пароль?</a>
</div>
</div>
</div>
</div>
<?php } ?>
</div>
<?php wp_footer(); ?>
</body>
</html>

```

page-account.php

```

<?php get_header(); ?>
<section>
    <div class="forh1">
        <h1> <?php if(!curent_user_is_admin()){ the_title();} else {echo
'Користувачі';} ?></h1>
        <div class="l_afterh1"></div>
        <div class="r_afterh1"></div>
        <div class="clearfix"></div>
    </div>
    <div class="row">
        <div class="col-12 my-3">
            <?php if(curent_user_is_admin()) { ?>
                <div class="tab-container">
                    <nav>
                        <div class="nav nav-tabs" id="nav-tab" role="tablist">
                            <a class="nav-item nav-link active" id="nav-home-tab"
data-toggle="tab" href="#nav-home" role="tab" aria-controls="nav-home"
aria-selected="true">Користувачі</a>
                            <a class="nav-item nav-link" id="nav-profile-tab" data-
toggle="tab" href="#nav-profile" role="tab" aria-controls="nav-
profile" aria-selected="false">Вільні модулі</a>
                        </div>
                    </nav>
                    <div class="tab-content" id="nav-tabContent">

```

```

    <div class="tab-pane fade show active" id="nav-home"
role="tabpanel" aria-labelledby="nav-home-tab">
    <?php $users = get_users( array( 'role' => 'subscriber', )
); ?>
    <table class="table">
    <thead class="thead-light">
    <tr>
    <th scope="col">Логін</th>
    <th scope="col">Пошта</th>
    <th scope="col">Публічне ім'я</th>
    <th scope="col">Додати групу</th>
    </tr>
    </thead>
    <tbody>
    <?php foreach( $users as $user ){ ?>
    <tr>
    <th scope="row"><?php echo $user->user_login;
?></th>
    <td><?php echo $user->user_email; ?></td>
    <td><?php echo $user->display_name; ?></td>
    <td>
    <form action="<?php echo esc_url(
admin_url('admin-post.php') ); ?>" method="post">
    <input type="hidden" name="action"
value="post_to_usergroups">
    <button type="submit" name="custom_user_id"
class="btn btn-outline-info" value="<?php echo $user->id;
?>">Переглянути</button>
    </form>
    </td>
    </tr>
    <?php } ?>
    </tbody>
    </table>
    </div>
    <div class="tab-pane fade" id="nav-profile" role="tabpanel"
aria-labelledby="nav-profile-tab">
    <?php } ?>
    <div class="col-12 my-2">
    <button type="button" class="btn btn-outline-success"
data-toggle="modal" data-target="#addgroup">
    Додати групу
    </button>
    <!-- Modal -->
    <div class="modal fade" id="addgroup" tabindex="-1"
role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">
    <div class="modal-dialog" role="document">
    <div class="modal-content">
    <div class="modal-header">
    <h5 class="modal-title"
id="exampleModalLabel">Форма додавання групи</h5>
    <button type="button" class="close" data-
dismiss="modal" aria-label="Close">

```

```

        <span aria-hidden="true">&times;</span>
    </button>
</div>
<form action="<?php echo esc_url(
admin_url('admin-post.php') ); ?>" method="post">
    <div class="modal-body">
        <input type="hidden" name="action"
value="add_admin_group">
        <input type="hidden" name="user_id"
value="<?php echo get_current_user_id(); ?>">
        <input type="hidden" name="redirect"
value="/account">
        <label for="new_gr">Назва групи</label>
        <input class="form-control" type="text"
placeholder="Введіть назву групи" name="name" id="new_gr">
    </div>
    <div class="modal-footer">
        <button type="reset" class="btn btn-outline-
secondary" data-dismiss="modal">Закрити</button>
        <button type="submit" class="btn btn-outline-
success">Додати</button>
    </div>
</form>
</div>
</div>
</div>
</div>
<?php $i=0; foreach ( select_cusom_groups (
get_current_user_id() ) as $group ) { ?>
    <div class="accordion my-3" id="accordion<?php echo
$group['gr_id'] ?>">
        <div class="card">
            <div class="card-header" id="heading<?php echo
$group['gr_id'] ?>">
                <div class="row">
                    <div class="col-6">
                        <h2 class="mb-0">
                            <button class="btn btn-link" type="button"
data-toggle="collapse" data-target="#collapse<?php echo
$group['gr_id']; ?>" aria-expanded="true" aria-
controls="collapse<?php echo $group['gr_id'] ?>">
                                <?php echo $group['name'] ?>
                            </button>
                        </h2>
                    </div>
                    <div class="col-6">
                        <button type="button" class="btn btn-outline-
info" data-toggle="modal" data-target="#edit<?php echo
$group['gr_id']; ?>"><i class="fas fa-pencil-alt"></i></button>
                        <!-- Modal -->
                        <div class="modal fade" id="edit<?php echo
$group['gr_id']; ?>" tabindex="-1" role="dialog" aria-

```

```

labelledby="ModalLabel<?php echo $group['gr_id'];; ?>" aria-
hidden="true">
        <div class="modal-dialog" role="document">
            <div class="modal-content">
                <div class="modal-header">
                    <h5 class="modal-title"
id="ModalLabel<?php echo $group['gr_id'];; ?>">Зміна <?php echo
$group['name'];; ?></h5>
                    <button type="button" class="close"
data-dismiss="modal" aria-label="Close">
                        <span aria-
hidden="true">&times;</span>
                    </button>
                </div>
                <div class="modal-body">
                    <form action="<?php echo esc_url(
admin_url('admin-post.php') ); ?>" method="post">
                        <input type="number" hidden
name="gr_id" value="<?php echo $group['gr_id'];; ?>">
                        <input type="hidden" name="action"
value="edit_group">
                        <input type="hidden" name="redirect"
value="/account">
                        <label for="unit<?php echo
$group['gr_id'];; ?>">Назва групи</label>
                        <input class="form-control"
type="text" placeholder="Введіть нову назву групи" value="<?php echo
$group['name'] ?>" name="name" id="unit<?php echo $group['gr_id'];;
?>">
                    <div class="modal-footer">
                        <button type="reset" class="btn btn-
outline-secondary" data-dismiss="modal">Закрити</button>
                        <button type="submit" class="btn
btn-outline-success">Змінити</button>
                    </div>
                </form>
            </div>
        </div>
    </div>
    <div>
        <form style="float: right; width: fit-content;"
action="<?php echo esc_url( admin_url('admin-post.php') ); ?>"
method="post">
            <input type="hidden" name="action"
value="dell_group">
            <input type="hidden" name="redirect"
value="/account">
            <button type="submit" name="dell_group"
class="btn btn-outline-danger" value="<?php echo $group['gr_id'];;
?>"><i class="fas fa-trash-alt"></i></button>
        </form>
    </div>
</div>

```

```

</div>
<div id="collapse<?php echo $group['gr_id'] ?>"
class="collapse <?php if(++$i==1) echo "show"; ?> " aria-
labelledby="heading<?php echo $group['gr_id'] ?>" data-
parent="#accordion<?php echo $group['gr_id'] ?>">
  <div class="card-body">
    <table class="table">
      <thead class="thead-light">
        <tr>
          <th scope="col">Назва</th>
          <th scope="col">Тип</th>
          <th scope="col">Редагувати</th>
          <th scope="col">Переглянути</th>
        </tr>
      </thead>
      <tbody>
        <?php $i=0;      foreach ( select_cusom_esp(
$group['gr_id'] ) as $esp ) { ?>
          <tr>
            <th scope="row"><?php echo $esp['name'];
?></th>
            <td><?php echo $esp['type']; ?></td>
            <td>
              <button type="button" class="btn btn-
outline-info" data-toggle="modal" data-target="#exampleModalesp<?php
echo $esp['esp_id']; ?>">
                <i class="fas fa-pencil-alt"></i>
              </button>
              <!-- Modal -->
              <div class="modal fade"
id="exampleModalesp<?php echo $esp['esp_id']; ?>" tabindex="-1"
role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">
                <div class="modal-dialog"
role="document">
                  <div class="modal-content">
                    <div class="modal-header">
                      <h5 class="modal-title"
id="exampleModalLabel">Редагування <?php echo $esp['name']; ?> (<?php
echo $esp['type']; ?>)</h5>
                      <button type="button"
class="close" data-dismiss="modal" aria-label="Close">
                        <span aria-
hidden="true">&times;</span>
                      </button>
                    </div>
                    <form action="<?php echo esc_url(
admin_url('admin-post.php') ); ?>" method="post">
                      <div class="modal-body">
                        <input type="hidden"
name="action" value="edit_custom_esp">
                        <input type="hidden"
name="esp_id" value="<?php echo $esp['esp_id']; ?>">

```



```

                                <label for="esp_name">Назва
пристрою</label>
                                <input class="form-control"
type="text" placeholder="Введіть нову назву пристрою" name="name"
value="<?php echo $esp['name']; ?>" id="esp_name">
                                <label for=gr>Група
приладу</label>
                                <select class="form-control"
name="esp_group" id="gr">
                                <?php foreach (
select_cusom_groups( get_current_user_id() ) as $optiongroup ) { ?>
                                <option <?php
if($optiongroup['gr_id']==$group['gr_id']) echo 'selected'; ?>
value="<?php echo $optiongroup['gr_id']; ?>"><?php echo
$optiongroup['name']; ?></option>
                                <?php } ?>
                                </select>
                                </div>
                                <div class="modal-footer">
                                <button type="reset" class="btn
btn-outline-secondary" data-dismiss="modal">Закрити</button>
                                <button type="submit" class="btn
btn-outline-info">Змінити</button>
                                </div>
                                </form>
                                </div>
                                </div>
                                </div>
                                </td>
                                <td>
                                <form action="<?php echo esc_url(
admin_url('admin-post.php') ); ?>" method="post">
                                <input type="hidden" name="action"
value="esp_data">
                                <input type="hidden" name="start"
value="-1">
                                <input type="hidden" name="type_id"
value="<?php echo $esp['type_id']; ?>">
                                <button type="submit" name="esp_id"
class="btn btn-outline-info" value="<?php echo $esp['esp_id']; ?>">
                                <i class="far fa-eye"></i>
                                </button>
                                </form>
                                </td>
                                </tr>
                                <?php } ?>

                                <?php $i=0;    foreach ( select_cusom_devices (
$group['gr_id'] ) as $device ) { ?>
                                <tr>
                                <th scope="row"><?php echo $device['name'];
?></th>

```

```

        <td><?php echo $device['type']; ?></td>
        <td>
            <button type="button" class="btn btn-
outline-info" data-toggle="modal" data-
target="#exampleModaldevice<?php echo $device['dev_id']; ?>"
            <i class="fas fa-pencil-alt"></i>
            </button>
            <!-- Modal -->
            <div class="modal fade"
id="exampleModaldevice<?php echo $device['dev_id']; ?>" tabindex="-1"
role="dialog" aria-labelledby="exampleModaldevice" aria-hidden="true">
                <div class="modal-dialog"
role="document">
                    <div class="modal-content">
                        <div class="modal-header">
                            <h5 class="modal-title"
id="exampleModaldevice">Редагування <?php echo $device['name']; ?>
(<?php echo $device['type']; ?>)</h5>
                            <button type="button"
class="close" data-dismiss="modal" aria-label="Close">
                                <span aria-
hidden="true">&times;</span>
                            </button>
                        </div>
                        <form action="<?php echo esc_url(
admin_url('admin-post.php') ); ?>" method="post">
                            <div class="modal-body">
                                <input type="hidden"
name="action" value="edit_custom_device">
                                <input type="hidden"
name="dev_id" value="<?php echo $device['dev_id']; ?>">
                                <label for="device_name">Назва
пристрою</label>
                                    <input class="form-control"
type="text" placeholder="Введіть нову назву пристрою" name="name"
value="<?php echo $device['name']; ?>" id="device_name">
                                    <label for="type">Тип
приладу</label>
                                        <input class="form-control"
type="text" placeholder="Введіть тип пристрою" name="type"
value="<?php echo $device['type']; ?>" id="type">
                                        <label for="gr_id">Група
приладу</label>
                                            <select class="form-control"
name="gr_id" id="gr_id">
                                                <?php foreach (
select_cusom_groups( get_current_user_id() ) as $optgroup ) { ?>
                                                    <option <?php
if($optgroup['gr_id']==$group['gr_id']) echo 'selected'; ?>
value="<?php echo $optgroup['gr_id']; ?>"><?php echo
$optgroup['name']; ?></option>

```

```

                <?php } ?>
            </select>
        </div>
        <div class="modal-footer">
            <button type="reset" class="btn
btn-outline-secondary" data-dismiss="modal">Закрити</button>
            <button type="submit" class="btn
btn-outline-info">Змінити</button>
        </div>
    </form>
</div>
</div>
</div>
</td>
<td>
    <form action="<?php echo esc_url(
admin_url('admin-post.php') ); ?>" method="post">
        <input type="hidden" name="action"
value="control_info">
        <input type="hidden" name="start"
value="-1">
        <button type="submit" name="dev_id"
class="btn btn-outline-info" value="<?php echo $device['dev_id']; ?>">
            <i class="far fa-eye"></i>
        </button>
    </form>
</td>
</tr>
<?php } ?>
</tbody>
</table>
</div>
</div>
</div>
</div>
<?php } ?>
<?php if(curent_user_is_admin()) { ?></div>
</div>
</div> <?php } ?>
</div>
</div>
</section>
<?php get_footer(); ?>

```

page-data.php

```

<?php get_header(); ?>
<?php
$esp_id = $_SESSION['esp_id'];
$start = $_SESSION['start'];
$end = $_SESSION['end'];

```

```

$unit_arr = $_SESSION['unit_arr'];
foreach ( select_esp_info( $esp_id ) as $esp ){
$min = $esp['min_'];
$max = $esp['max_'];
$need = $esp['need'];
$type= $esp['type'];
$name_esp = $esp['name'];
}
foreach ( get_limits($esp_id, $unit_arr['formula']) as $limit ){

    $min_limit=$limit['min_v'];
    $max_limit=$limit['max_v'];
    $need_limit=$limit['need_v'];
}
foreach ( get_last_data($esp_id) as $last_data){
    $last_value=$last_data['value_'];
    $last_data=$last_data['dt'];
}
?>

<section>
    <div class="forh1">
        <h1>Дані з приладу"<?php echo $name_esp; ?>" (<?php echo $type;
?>) </h1>
        <div class="l_afterh1"></div>
        <div class="r_afterh1"></div>
        <div class="clearfix"></div>
    </div>
    <div class="row">
        <div class="col-12 my-3">
            <?php echo get_alert($unit_arr['type_id'],$min, $max, $need,
$last_value,$last_data ); ?>
        </div>
        <div class="text-center col-12 py-3">
            <script type="text/javascript"
src="https://www.gstatic.com/charts/loader.js"></script>
            <div id="chart_div"></div>
        </div>
        <form class="w-100 col-12" action="<?php echo esc_url(
admin_url('admin-post.php') ); ?>" method="post">
            <div class="row">
                <input type="hidden" name="action" value="esp_data">
                <div class="col-md-4">
                    <label for=start>Початок інтервалу</label>
                    <input class="form-control" id="start" type="date"
name="start" value="<?php echo $start; ?>" max="" min="">
                </div>
                <div class="col-md-4">
                    <label for=end>Кінець інтервалу</label>
                    <input class="form-control" type="date" name="end" id="end"
value="<?php echo $end; ?>" min="" max="">
                </div>
                <div class="col-md-4">

```

```

        <label for=gr>Одиниці вимірювання</label>
        <select class="form-control" name="unit_id" id="gr">
            <?php $i=0; foreach (
select_cusom_units($unit_arr['type_id']) as $unit ) { ?>
                <option <?php if($unit['unit_id']==$unit_arr['unit_id'])
echo 'selected'; ?> value="<?php echo $unit['unit_id']; ?>">
                    <?php echo $unit['unit']; ?> (<?php echo $unit['mark'];
?>)
                </option>
            <?php } ?>
        </select>
    </div>
    <div class="col-md-4">
        <button type="submit" name="esp_id" class="btn btn-outline-
info my-2" value="<?php echo $esp_id; ?>">Повторний запит</button>
    </div>
</div>
</form>
<div class="col-12 py-3">
    <form class="w-100" action="<?php echo esc_url(admin_url('admin-
post.php')); ?>" method="post">
        <input type="hidden" name="action" value="set_limits">
        <div class="row">
            <div class="col-md-4">
                <label for=min_>Мінімум</label>
                <input class="form-control" id="min_" type="number"
name="min_" value="<?php echo $min; ?>">
            </div>
            <div class="col-md-4">
                <label for=max_>Максимум</label>
                <input class="form-control" id="max_" type="number"
name="max_" value="<?php echo $max; ?>">
            </div>
            <div class="col-md-4">
                <label for=need>Необхідний показник</label>
                <input class="form-control" id="need" type="number"
name="need" value="<?php echo $need; ?>">
            </div>
        </div>
        <button type="submit" name="esp_id" class="btn btn-outline-
info my-2" value="<?php echo $esp_id; ?>">Змінити границі</button>
    </form>
</div>
</div>
</section>
<script>
    google.charts.load('current', {
        'packages': ['line', 'corechart']
    });
    google.charts.setOnLoadCallback(drawChart);

    function drawChart() {

```

```

var button = document.getElementById('change-chart');
var chartDiv = document.getElementById('chart_div');

var data = new google.visualization.DataTable();
data.addColumn('string', 'Month');
data.addColumn('number', "Фактичний <?php echo $type; ?>");
data.addColumn('number', "Мінімально допустимий");
data.addColumn('number', "Максимально допустимий");
data.addColumn('number', "Необхідний показник");

data.addRows([
    <?php foreach ( select_cusom_data($esp_id, $start, $end,
$unit_arr['formula']) as $data ){ ?>

        ["<?php echo $data['d']; ?>",
        <?php echo $data['v']; ?>,
        <?php echo $min_limit; ?>,
        <?php echo $max_limit; ?>,
        <?php echo $need_limit; ?>],
        <?php } ?>
]);

var classicOptions = {
    title: 'Графік з <?php echo $start; ?> по <?php echo $end; ?>',
    width: 900,
    height: 500,
    series: {
        0: {
            color: '#c028f8'
        },
        1: {
            color: '#6771e8'
        },
        2: {
            color: '#e2431e'
        },
        3: {
            color: '#2be71b'
        },
    },
    vAxes: {
        0: {
            title: '<?php echo $unit_arr['unit']; ?> (<?php echo
$unit_arr['mark']; ?>)'
        }
    },
    vAxis: {
        viewWindow: {
            max: <?php echo $max_limit+5; ?>,
            min: <?php echo $min_limit-5; ?>
        }
    }
}

```

```
};

function drawClassicChart() {
    var classicChart = new google.visualization.LineChart(chartDiv);
    classicChart.draw(data, classicOptions);
}
drawClassicChart();
}
</script>
<?php get_footer(); ?>
```