

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Розробка онлайн-платформи для супроводу середовища «розумної»
квартири

Виконав: студент IV курсу, групи СН-42

спеціальності 122 Комп'ютерні науки

(шифр і назва спеціальності)

(підпис)

Старуцак В.В.

(прізвище та ініціали)

Керівник

(підпис)

Гащин Н.Б.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Шимчук Г.В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Боднарчук І.О.

(прізвище та ініціали)

Рецензент

(підпис)

Голотенко О.С.

(прізвище та ініціали)

Тернопіль
2024

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.
(підпис) (прізвище та ініціали)

«__» червня 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

Студенту Старушак Віталій Володимирович
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка онлайн-платформи для супроводу середовища «розумної» квартири

Керівник роботи Гашин Надія Богданіна, к.т.н., доцент кафедри ММ
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «29» квітня 2024 року № 4/7-470

2. Термін подання студентом завершеної роботи 24 червня 2024р.

3. Вихідні дані до роботи Інтернет джерела про технології розробки сайту

4. Зміст роботи (перелік питань, які потрібно розробити)

Розділ 1. Аналіз предметної області та постановка завдання на розробку онлайн-платформи для відображення моніторингу метеоінформації для «розумної» квартири. 1.1 Постановка задачі. 1.2 Аналіз варіантів вирішення поставленої задачі. 1.3 Висновок до першого розділу. Розділ 2. Проектування онлайн-платформи для відображення моніторингу метеоінформації для «розумної» квартири. 2.1 Моделювання архітектури онлайн-платформи. 2.2 Проектування структури онлайн-платформи. 2.3 Проектування поведінки онлайн-платформи. 2.4 Проектування інтерфейсу онлайн-платформи. 2.5 Висновок до другого розділу. Розділ 3. Розгортання та налаштування онлайн-платформи для відображення моніторингу метеоінформації для «розумної» квартири. 3.1 Перелік етапів розгортання онлайн-платформи. 3.2 Розміщення онлайн-платформи на хостинг. 3.3 Структура бази даних. 3.4 Тестування онлайн-платформи для відображення моніторингу метеоінформації для підприємства та дому. 3.5 Висновки до третього розділу. Розділ 4. Безпека життєдіяльності, основи охорони праці. 4.1 Захист стійкості комп'ютерної техніки від ураження в надзвичайних ситуаціях. 4.2 Надзвичайні ситуації в Україні та їх класифікація. 4.3 Загальні вимоги безпеки з охорони праці для користувачів ПК. 4.4 Висновок до четвертого розділу

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

1. Титульний лист. 2. Актуальність роботи. 3. Мета і завдання роботи. 4. Практичне значення одержаних результатів. 5. Порівняльний аналіз існуючих рішень. 6. Життєвий цикл онлайн-платформи. 7. Архітектура онлайн-платформи для відображення моніторингу метеоінформації. 8. Діаграма компонентів онлайн-платформи. 9. Функціональна схема роботи онлайн-платформи. 10. Головна сторінка онлайн-платформи. 11. Робота адаптивного дизайну онлайн-платформи. 12. Інтерфейс головних сторінок онлайн-платформи. 13. Етапи розгортання онлайн-платформи для відображення моніторингу метеоінформації для квартири. 14. Діаграма потоків даних. 15. Тестування на кросбраузерність. 16. Висновки

АНОТАЦІЯ

Розробка онлайн-платформи для супроводу середовища «розумної» квартири // Кваліфікаційна робота освітнього рівня «Бакалавр» // Старуцак Віталій Володимирович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедри комп'ютерних наук, група СН-42 // Тернопіль, 2024 // С. – 67, рис. – 36, табл. – 6, слайди – 16, додат. – 23 , бібліогр. – 28.

Ключові слова: онлайн-платформа, метеоінформація, моніторинг, компонент, шаблон.

Кваліфікаційна робота присвячена розробці розробці онлайн-платформи для відображення моніторингу метеоінформації для «розумної» квартири.

Перший розділ досліджує предметну область і визначає завдання для розробки онлайн-платформи для відображення моніторингу метеоінформації для «розумної» квартири. Аналіз предметної області дозволив визначити ключові аспекти, пов'язані з вимогами до платформи та її варіантами використання. Вимоги включають не тільки точність відображення метеоданих, але й зручність інтерфейсу для кінцевих користувачів.

Другий розділ присвячено проектуванню онлайн-платформи для відображення моніторингу метеоінформації для «розумної» квартири. Моделювання архітектури включало розгляд різних можливостей структурування системи з урахуванням потреб високої доступності та швидкодії.

Третій розділ описує процес розгортання та налаштування онлайн-платформи для відображення моніторингу метеоінформації для «розумної» квартири. У розділі наведено перелік етапів розгортання платформи, включаючи підготовку середовища, встановлення необхідного програмного забезпечення, налаштування серверів і забезпечення безперебійної роботи системи.

ANNOTATION

Development of an Online Platform to Support the Environment of a «Smart» Apartment // Qualification work of the educational, level «Bachelor» // Starushchak Vitaliy Volodymyrovych // Ternopil Ivan Pulyu National Technical University, Computer and Information Systems and Software Engineering Faculty, Computer Sciences Department, group SN-42 // Ternopil, 2024 // P. – 67, fig. – 36, tabl. – 6, slides – 16, annexes. – 23, references – 41.

Keywords: online platform, weather information, monitoring, component, template.

The qualification work is devoted to the development of an online platform for displaying meteorological information monitoring for a «smart» apartment.

The first chapter explores the subject area and defines the task for the development of an online platform for displaying meteorological information monitoring for a «smart» apartment. The analysis of the subject area made it possible to determine the key aspects related to the requirements for the platform and its use cases. The requirements include not only the accuracy of the display of meteorological data, but also the user-friendliness of the interface.

The second chapter is devoted to the design of an online platform for displaying meteorological information monitoring for a "smart" apartment. Architecture modeling included consideration of various system structuring possibilities taking into account the needs of high availability and speed.

The third section describes the process of deployment and configuration of an online platform for displaying meteorological information monitoring for a "smart" apartment. The section lists the steps involved in deploying the platform, including preparing the environment, installing the necessary software, configuring the servers, and ensuring the system is up and running.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

AJAX (Asynchronous Javascript And Xml) – підхід до побудови користувацьких інтерфейсів веб-застосунків.

HTML (HyperText Markup Language) – стандартизована мова гіпертекстової розмітки документів у Всесвітній мережі.

XHTML (Extensible Hypertext Markup Language) – розширена мова гіпертекстової розмітки.

DOM (Document Object Model) – програмний інтерфейс, що дозволяє програмам і скриптам отримати доступ до вмісту HTML, XHTML і XML.

CSS (Cascading Style Sheets) – список інструкцій для браузера, - як і де відображати елементи веб-сторінки, написаний особливим чином.

DNS (Domain Name System) – комп'ютерна розподілена система для отримання інформації про домен.

FTP (File Transfer Protocol) – протокол передачі файлів.

HTTP (Hyper Text Transfer Protocol) – протокол прикладного рівня передачі даних (у вигляді гіпертекстових документів).

TCP/IP (Transmission Control Protocol/Internet Protocol) – протокол керування передачею.

PHP (Hypertext Preprocessor) – скриптова мова загального призначення, інтенсивно застосовується для розробки веб-додатків.

ASCII (American Standard Code for Information Interchange) – система кодів.

JSON (JavaScript Object Notation) – це текстовий формат обміну даними між комп'ютерами.

API (Application Programming Interface) – набір визначень взаємодії різнотипного програмного забезпечення.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ НА РОЗРОБКУ ОНЛАЙН-ПЛАТФОРМИ ДЛЯ ВІДОБРАЖЕННЯ МОНІТОРИНГУ МЕТЕОІНФОРМАЦІЇ ДЛЯ «РОЗУМНОЇ» КВАРТИРИ	10
1.1 Постановка задачі	10
1.1.1 Аналіз предметної області	10
1.1.2 Вимоги до онлайн-платформи для відображення моніторингу метеоінформації для «розумної квартири»	11
1.1.3 Варіанти використання.....	13
1.2 Аналіз варіантів вирішення поставленої задачі.....	15
1.2.1 Оцінка методів розв’язання поставленої задачі.....	15
1.2.2 Вибір оптимального алгоритму вирішення задачі	19
1.2.3 Вибір середовища розробки та обґрунтування його використання ...	20
1.2.4 Життєвий цикл онлайн-платформи для відображення моніторингу метеоінформації для «розумної» квартири	23
1.3 Висновок до першого розділу.....	26
РОЗДІЛ 2. ПРОЕКТУВАННЯ ОНЛАЙН-ПЛАТФОРМИ ДЛЯ ВІДОБРАЖЕННЯ МОНІТОРИНГУ МЕТЕОІНФОРМАЦІЇ ДЛЯ «РОЗУМНОЇ» КВАРТИРИ	27
2.1 Моделювання архітектури онлайн-платформи.....	27
2.2 Проектування структури онлайн-платформи	28
2.3 Проектування поведінки онлайн-платформи.....	29
2.4 Проектування інтерфейсу онлайн-платформи.....	31
2.5 Висновок до другого розділу	38
РОЗДІЛ 3. РОЗГОРТАННЯ ТА НАЛАШТУВАННЯ ОНЛАЙН-ПЛАТФОРМИ ДЛЯ ВІДОБРАЖЕННЯ МОНІТОРИНГУ МЕТЕОІНФОРМАЦІЇ ДЛЯ «РОЗУМНОЇ» КВАРТИРИ	40
3.1 Перелік етапів розгортання онлайн-платформи	40
3.2 Розміщення онлайн-платформи на хостинг	41
3.3 Структура бази даних	43

3.4 Тестування онлайн-платформи для відображення моніторингу метеоінформації для підприємства та дому	47
3.4.1 Тестування користувацького інтерфейсу	47
3.4.2 Тестування функціоналу онлайн-платформи.....	50
3.5 Висновки до третього розділу	55
РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	56
4.1 Захист стійкості комп'ютерної техніки від ураження в надзвичайних ситуаціях	56
4.2 Надзвичайні ситуації в Україні та їх класифікація.....	56
4.2 Загальні вимоги безпеки з охорони праці для користувачів ПК.....	59
4.4 Висновок до четвертого розділу.....	61
ВИСНОВКИ.....	62
ПЕРЕЛІК ДЖЕРЕЛ	65
ДОДАТКИ	

ВСТУП

Актуальність теми. У сучасному світі, де інформаційні технології стрімко розвиваються, зростає потреба у зручних та ефективних засобах моніторингу різноманітної інформації. Однією з таких важливих сфер є моніторинг метеоінформації, яка є ключовою для багатьох підприємств та домогосподарств. Відстеження погодних умов дозволяє не лише планувати діяльність, а й забезпечувати безпеку та комфорт. Тому створення онлайн платформи для відображення метеоінформації є актуальним і важливим завданням.

Метою даної роботи є розробка онлайн-платформи для підтримки та управління середовищем «розумної» квартири. Платформа повинна забезпечувати інтеграцію різних пристроїв і систем, що входять до складу «розумної» квартири, дозволяючи користувачам зручно керувати ними через інтуїтивно зрозумілий інтерфейс. Це включає моніторинг та контроль систем безпеки, освітлення, опалення, вентиляції та кондиціонування повітря, а також інших елементів домашньої автоматизації.

Постановка задачі включає кілька етапів, починаючи з аналізу предметної області. На цьому етапі важливо дослідити потреби користувачів, визначити ключові елементи системи та зрозуміти, як найкраще представити метеоінформацію. Вимоги до онлайн-платформи повинні включати функціональні та нефункціональні аспекти, що забезпечують ефективне використання ресурсу як підприємствами, так і домашніми користувачами.

Далі слід розглянути варіанти використання, які демонструють можливі сценарії взаємодії користувачів з онлайн-платформою. Це дозволить чітко визначити функціонал, необхідний для задоволення різних потреб користувачів. Аналіз варіантів вирішення поставленої задачі передбачає оцінку різних методів та підходів, вибір оптимального алгоритму вирішення задачі та обґрунтування вибору середовища розробки.

Проектування онлайн-платформи охоплює моделювання архітектури, структури та поведінки платформи, а також проектування її інтерфейсу. Це

забезпечить узгодженість усіх компонентів системи та створення зручного і привабливого для користувачів інтерфейсу.

Розгортання та налаштування онлайн-платформи включає етапи розгортання, розміщення на хостингу та структуру бази даних. Це забезпечить надійну роботу ресурсу та доступ до нього з будь-якого пристрою. Завершальним етапом є тестування онлайн платформи, яке включає перевірку користувацького інтерфейсу та функціоналу. Це дозволить виявити та усунути можливі недоліки, забезпечуючи високу якість кінцевого продукту.

Таким чином, розробка онлайн-платформи для відображення метеоінформації для підприємства та дому є комплексним завданням, яке включає аналіз, проектування, розгортання та тестування. Це забезпечить користувачів зручним інструментом для моніторингу погодних умов, що сприятиме підвищенню їхньої ефективності та комфорту.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ НА РОЗРОБКУ ОНЛАЙН-ПЛАТФОРМИ ДЛЯ ВІДОБРАЖЕННЯ МОНІТОРИНГУ МЕТЕОІНФОРМАЦІЇ ДЛЯ «РОЗУМНОЇ» КВАРТИРИ

1.1 Постановка задачі

1.1.1 Аналіз предметної області

Онлайн-платформа для моніторингу метеоінформації на підприємстві та вдома є автоматизованою інформаційною системою. Основні елементи, що забезпечують повноцінне функціонування онлайн-платформи, включають:

- дані, що передаються з хмарного середовища та відображаються на сайті;
- датчики, додані користувачами для збору метеоінформації;
- графічне відображення даних.

Ці структурні елементи повинні бути правильно інтегровані та представлені в зручному для користувача форматі. Інтерфейс має бути інтуїтивно зрозумілим.

Основне призначення цієї онлайн-платформи – спростити процес моніторингу метеоінформації на підприємстві чи вдома за допомогою зручного відображення даних, які надходять від датчиків.

Введення в експлуатацію онлайн-платформи для моніторингу метеоінформації дозволить відстежувати різні метеорологічні показники, що дає змогу контролювати погодні умови як на підприємстві, так і вдома.

Моніторинг метеоінформації широко застосовується в багатьох сферах. Автоматизація цього процесу за допомогою онлайн-платформи потребує ретельної обробки даних, що надходять від датчиків.

1.1.2 Вимоги до онлайн-платформи для відображення моніторингу метеоінформації для «розумної квартири»

Веб-сайт для моніторингу метеоінформації для підприємств і дому є частиною інформаційної системи. Інформаційна система – це взаємопов'язаний набір інструментів, методів і персоналу, що використовуються для зберігання, обробки та подання даних, з метою вирішення конкретних завдань користувача.

Робота будь-якої інформаційної системи складається з наступних процесів, які умовно можна представити у вигляді блок-схеми:

- введення даних з зовнішніх або внутрішніх джерел;
- обробка вхідної інформації та подання її у зручному вигляді;
- виведення оброблених даних або передача їх до іншої системи;
- аналіз отриманих результатів;
- зворотний зв'язок для корекції вхідних даних на основі аналізу.

Метою даної роботи є створення функціональної онлайн-платформи, головним призначенням якого є відображення моніторингу метеоінформації. Сайт повинен мати зручний, інтуїтивно зрозумілий інтерфейс для комфортного використання.

Предметом дослідження є технологія проектування та реалізації інформаційної системи на прикладі онлайн-платформи для моніторингу метеоінформації для підприємств і дому. Основний функціонал онлайн-платформи включає:

- реєстрація нових користувачів та авторизація існуючих;
- додавання і видалення дачачів інформації;
- розміщення дачачів та їх видалення;
- перегляд метеоінформації у вигляді графіків;
- налаштування профілю та акаунту;
- авторизація хмарного сервісу.

Система передбачає два основні типи користувачів: незареєстровані та зареєстровані. Незареєстровані користувачі можуть зареєструватися для

отримання доступу до повного функціоналу сайту. Для зареєстрованих користувачів передбачено:

- додавання і видалення давачів;
- розміщення і видалення давачів;
- перегляд метеоінформації у вигляді графіків;
- налаштування профілю та акаунту;
- авторизація хмарного сервісу.

Головна сторінка сайту повинна бути інтуїтивно зрозумілою та привабливою. Вона забезпечує доступ до всіх розділів сайту, включаючи розташування давачів, графіки та налаштування профілю.

Інтерфейс онлайн-платформи повинен бути простим у використанні, задовольняти всі потреби користувача та мати сучасний вигляд. Сайт має бути функціональним і правильно відобразитися на всіх пристроях. Дизайн онлайн-платформи повинен бути стильним, кожна сторінка має мати єдину стилістику, забезпечуючи легку навігацію без ризику "загубитися в лабіринті" інформації.

Веб-сторінки створюються з використанням мови HTML і формуються як текст у форматі ASCII зі спеціальними розмічувальними кодами (тегами). Веб-ресурс повинен бути простим у використанні, мати приємний та зрозумілий інтерфейс, забезпечувати швидкий пошук інформації.

Виходячи з призначення та мети розробки сайту, визначимо наступні вимоги до його технічної реалізації:

- приємний та зручний інтерфейс;
- форма для авторизації для обмеження доступу;
- можливість відображення даних, отриманих від давачів;
- можливість додавання/видалення давачів;
- інформаційне табло з оновленням даних у реальному часі;
- налаштування відображення даних.

Завданням проекту є створення функціонального онлайн-платформи для моніторингу метеоінформації для підприємств і дому з зручним та зрозумілим інтерфейсом. Вимоги до технічної реалізації системи включають:

- гнучкість налаштувань: зручний інтерфейс з зрозумілими модулями;

- масштабованість: сайт має бути динамічним і змінюваним за необхідністю;
- безпека: різні рівні доступу до даних, ієрархія користувачів.

Інтерфейс сайту повинен бути максимально простим і водночас забезпечувати весь необхідний функціонал для зручної роботи користувачів і відповідати їхнім естетичним вимогам.

1.1.3 Варіанти використання

Діаграма варіантів використання відображає взаємодію між акторами та випадками використання в системі. Вона являє собою граф, що складається з акторів і випадків використання (обмежених межами системи у вигляді прямокутника), а також включає асоціації між ними, відносини між випадками використання та узагальнення між акторами. Такі діаграми демонструють елементи моделі варіантів використання.

Основна ідея цієї діаграми полягає в тому, що проектована система представлена у вигляді сукупності сутностей або акторів, які взаємодіють із системою через варіанти використання. Варіанти використання описують послуги, які система надає актору, визначаючи набір дій, що виконує система під час взаємодії з актором. При цьому діаграма не уточнює, як саме реалізована взаємодія акторів із системою.

Цілі діаграми варіантів використання включають:

- Визначення загальних меж і контексту моделюваної предметної області на початкових етапах проектування системи.
- Формулювання загальних вимог до функціональної поведінки проектованої системи.
- Розробка вихідної концептуальної моделі системи для її подальшої деталізації у вигляді логічних і фізичних моделей.
- Підготовка вихідної документації для взаємодії розробників системи з її замовниками та користувачами.

Відповідно до вимог до функціоналу сайту моніторингу метеоінформації для підприємства та дому, передбачено два типи користувачів: зареєстровані та незареєстровані.

Зареєстровані користувачі мають доступ до повного функціоналу сайту, включаючи можливість додавати датчики в систему, редагувати їх параметри, змінювати дані свого профілю та налаштовувати відображення даних на своїй сторінці.

Незареєстровані користувачі можуть зареєструватися, щоб отримати можливість додавати свої датчики та відображати інформацію з них у своєму профілі.

На основі наведених вимог створюється діаграма варіантів використання, яка показана на рисунку 1.1.



Рисунок 1.1 – Діаграма варіантів використання

Таким чином, весь необхідний функціонал онлайн-платформи представлений у зручній для користувача візуальній формі. Виконання всіх варіантів використання, показаних на цій діаграмі, дозволить забезпечити реалізацію всіх необхідних функцій відповідно до вимог.

1.2 Аналіз варіантів вирішення поставленої задачі

1.2.1 Оцінка методів розв'язання поставленої задачі

Щоб правильно спроектувати та розробити онлайн-платформу для моніторингу метеоумов у «розумній» квартирі, необхідно провести детальний аналіз доступних у глобальній мережі аналогічних сайтів.

На сьогодні в Інтернеті існує кілька значущих аналогів для моніторингу метеоумов, які спроектовані за різними архітектурними підходами та реалізовані з використанням різноманітних технологій, методів та спеціалізованого програмного забезпечення.

Ось деякі приклади наявних в Інтернеті сайтів-аналогів даного напрямку. Веб-сайт ThingSpeak наведений на рисунку 1.2.

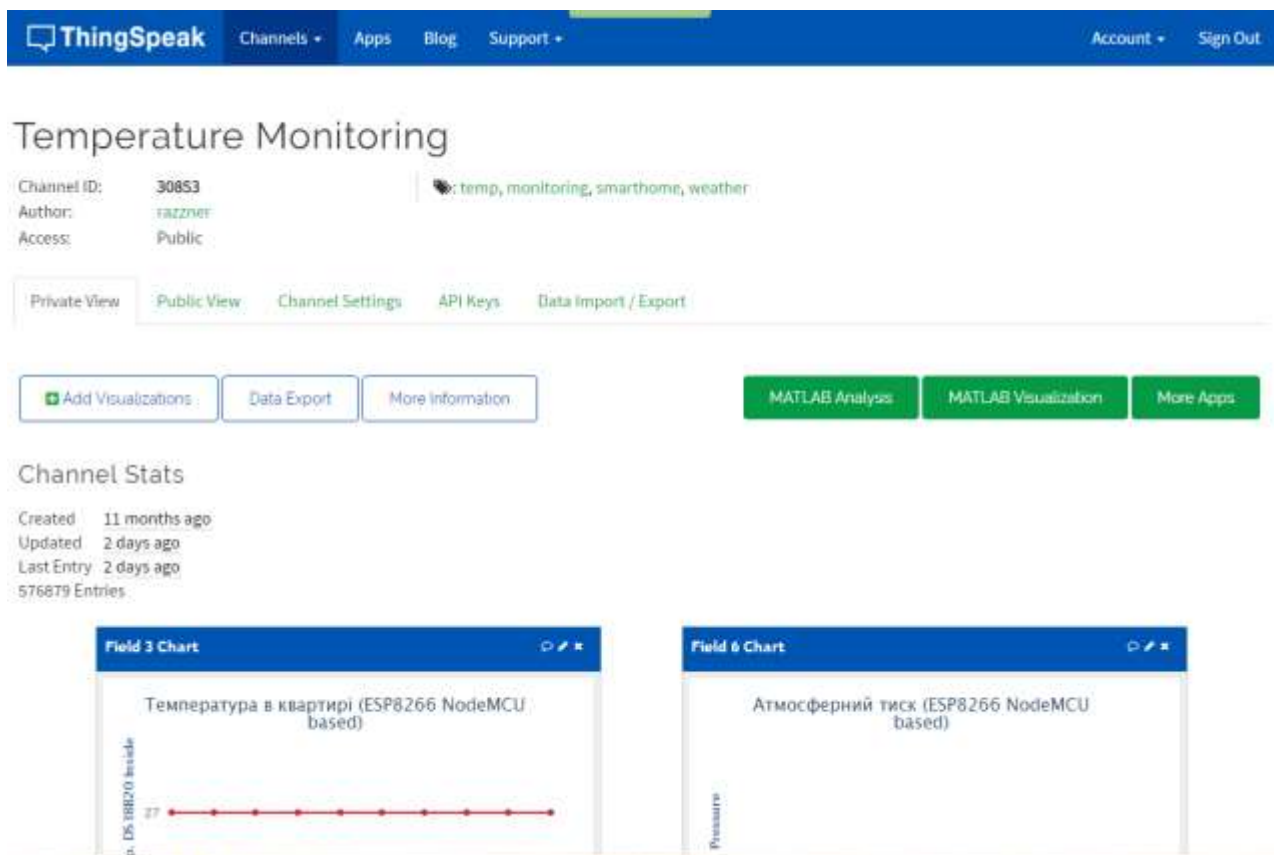


Рисунок 1.2 – Головна сторінка ThingSpeak

Використовується як хмарне сховище для пристроїв, підключених до IoT (Internet of Things), дозволяючи підключати до сервісу до 8 пристроїв з різними типами даних. Обмін даними між датчиками і хмарним сховищем здійснюється через HTTP-запити та JSON. Захист даних забезпечується за допомогою API-ключів для читання і запису даних. Сервіс також надає можливість візуалізувати дані з давачів у вигляді графіків, для кожного з яких можна задати певні параметри відображення, як показано на рисунку 1.3.

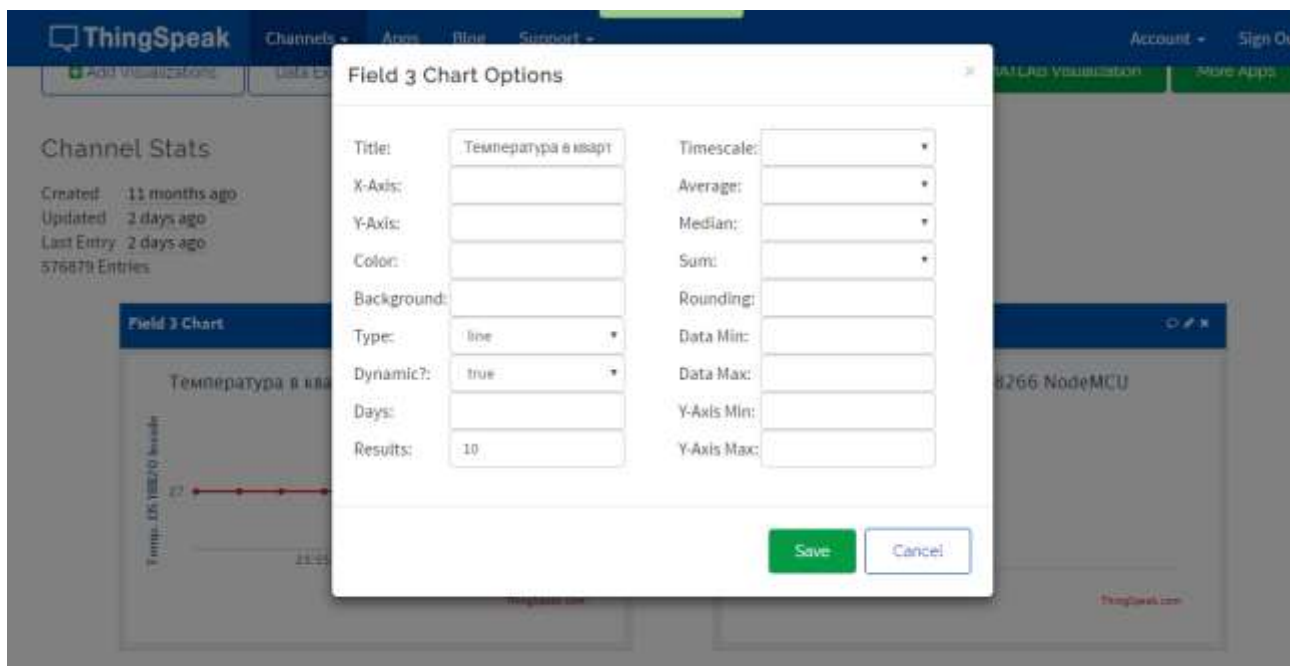


Рисунок 1.3 – Налаштування графіку на ThingSpeak

Негативні аспекти цього сервісу включають відсутність прямого доступу до бази даних, можливість налаштування відображення лише останніх показників від конкретного датчика, а також обмежену кількість давачів, які можна додати до свого акаунту. Веб-сторінка Exosite показана на рисунку 1.4.



Рисунок 1.4 – Головна сторінка – Exosite

Цей сервіс, подібно до попереднього, дозволяє додавати датчики до системи та зберігати дані з них. Однак його функціонал розширено завдяки можливості більш детального налаштування експорту даних з датчиків, як показано на рисунку 1.5. Крім того, він не має обмежень щодо кількості датчиків, які можна додати до системи.



Рисунок 1.5 – Налаштування експорту даних Exosite

Серед недоліків цього хмарного сервісу зберігання даних можна виділити високий поріг входження для нових користувачів і відсутність прямого доступу до бази даних. Головна сторінка онлайн-платформи Narodmon представлена на рисунку 1.6.

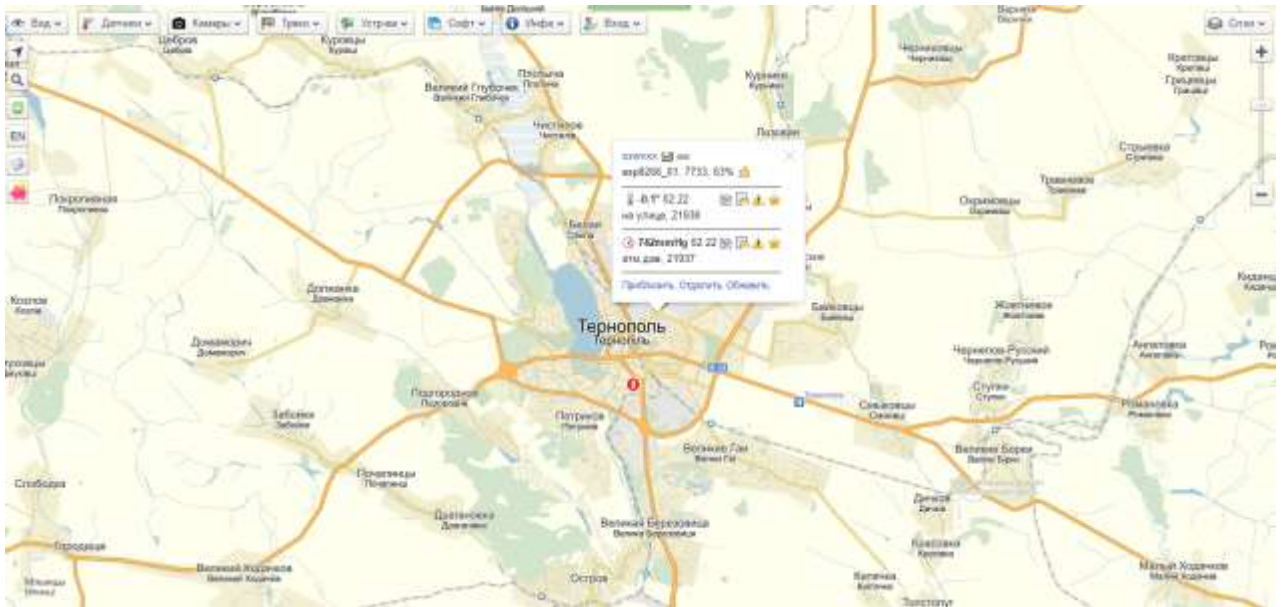


Рисунок 1.6 – Головна сторінка Narodmon

Ця онлайн-платформа є геоінформаційним сервісом, який відображає на мапі світу показники датчиків учасників проєкту з будь-якої точки планети. До таких даних можуть належати температура, вологість, атмосферний тиск, швидкість і напрямок вітру, енергоспоживання тощо. Передача даних на онлайн-платформу здійснюється за допомогою протоколів TCP/UDP або HTTP GET/POST-запитів. Для передачі даних з датчиків на сервіс необхідно вибрати зручний спосіб і передати дані разом з датою і MAC-адресою датчика, як показано на рисунку 1.7.

```

Если Вы не умеете доукомплектовать свой IP адрес или у Вас проблемы с DNS, то используйте любой IP из списка narodmon.ru до которого у Вас
возможно использовать для массовой загрузки показаний на сервер "видные частоты" путем передачи в одном пакете данных показаний датчика в формате
показаний как датчиков в одном пакете данных для индивидуальной адресации на индивидуальное соединение с сервером и импорт показаний датчика.

Два датчика реалтайм:      Формат пакета данных:
#E2-28-90-E9-F8-E3        #MAC[+NAME][+LAT][+LONG][+ELE]n
#T1#-23.41                #mac1#val1#(time)#name1]n
#N1#20                    --
#P1#708.18                #mac2#val2#(time)#name2]n
##                         ##

Загрузка историч показаний: С названием и GPS:
#E2-28-90-E9-F8-E3        #E2-28-90-E9-F8-E3#Метео#49.5623#25.6021#123.5
#T1#-23.41#1457915431     #T1#-23.41#Улица
#T1#-23.52#1457914831     #T2#21.77#Дом
#T1#-23.63#1457914231     #P1#708.18#Барометр
##                         ##

n - спецзнак конца строки - код 10(N), \n - спецзнак переноса строки - код 13(O),
## - признак окончания пакета данных(после него разрыв соединения), а [ ] заключает необязательный параметр.

= Пример передачи показаний TCP на PHP

$fp = @fsockopen("tcp://narodmon.ru", 8283, $errno, $errstr);
if(!$fp) exit("ERROR!".$errno."");
fwrite($fp, "#E2-28-90-E9-F8-E3\n#T1#-23.41\n#N1#20\n#P1#708.18\n##");
fclose($fp);

= Пример передачи показаний TCP на Bash

#!/bin/bash
# by Tulinov Andrey voytnar @ yandex.ru

SERVER="narodmon.ru"
PORT="8283"

# MAC адрес устройства. Заменить на свой!
DEVICE_MAC="0123456789012"

# идентификатор устройства, для простоты добавляется 01 (02) к MAC устройства
SENSOR_ID_1=$DEVICE_MAC"01"
SENSOR_ID_2=$DEVICE_MAC"02"

```

Рисунок 1.7 – Вибір способу передачі даних від давача даних на Narodmon

Онлайн-платформа для моніторингу погодних умов квартир передбачає можливість реєстрації, після якої пропонується додати давачів даних в систему та розпочати моніторинг метеоумов. Інформація відображається у вигляді блоків з даними від давачів та графіків, параметри відображення яких налаштовуються користувачем. Також на платформі присутній міні-блог, щоб користувачі могли ділитися новинами про нововведення в системі або про нових доданих давачів. Онлайн-платформа реалізована з використанням новітніх технологій і має зрозумілий інтерфейс. Розробка сервісу моніторингу є доцільною, зважаючи на той факт, що кількість пристроїв та давачів, що працюють в «інтернеті речей», зростає з кожним роком.

1.2.2 Вибір оптимального алгоритму вирішення задачі

Алгоритм - це набір інструкцій, який описує послідовність дій, необхідних для досягнення певного результату в розв'язанні задачі за обмежену кількість дій. Це система правил виконання дискретного процесу, яка забезпечує досягнення поставленої мети за обмежений час. Для візуалізації таких алгоритмів часто

використовують блок-схеми, де різні елементи, такі як операції, потік або дані, позначаються спеціальними символами (геометричні образи).

Після визначення вимог до функціоналу онлайн-платформи було розроблено наступний алгоритм для вирішення поставленої задачі:

- Аналіз предметної області.
- Огляд існуючих аналогів.
- Визначення вимог до платформи.
- Визначення основного функціоналу платформи.
- Концептуальне проектування.
- Аналіз доступних засобів для реалізації.
- Вибір інструментів для реалізації.
- Реалізація платформи з використанням обраних інструментів.
- Тестування онлайн-платформи.
- Введення сайту в експлуатацію.
- Забезпечення технічної підтримки протягом всього життєвого циклу.

Виконавши всі ці кроки, можливе успішне введення платформи в експлуатацію, що підвищує його вірогідність успішного функціонування. Онлайн-платформа буде здатна надати повноцінне відображення моніторингу погодних умов для квартир та підприємств.

1.2.3 Вибір середовища розробки та обґрунтування його використання

Для розробки онлайн-платформи з відображенням моніторингу метеоінформації для підприємства та дому був обраний фреймворк Bootstrap як основний інструмент. У процесі реалізації сайту використовувались наступні компоненти та засоби розробки:

- PHP - скриптова мова програмування для створення HTML-сторінок на сервері, що підтримується більшістю хостинг-провайдерів.
- HTML - стандартна мова розмітки веб-сторінок, яка обробляється браузером та відтворює сторінку для користувача.

- CSS - каскадні таблиці стилів для візуального оформлення сторінок і компонентів.

- JavaScript - мова програмування для реалізації інтерактивності веб-сторінок.

Ці технології і компоненти були інтегровані з використанням фреймворку Bootstrap 3, що спростило процес розробки та забезпечило швидкий розвиток веб-застосунку.

JavaScript є реалізацією стандарту мови програмування ECMA Script компанії Netscape і базується на принципах прототипного програмування. Основне застосування мови полягає у написанні сценаріїв для веб-сторінок, а також у впровадженні сценаріїв керування об'єктами, вбудованими в інші програми. JavaScript був розроблений у компанії Netscape і на сьогоднішній день підтримується більшістю браузерів. Текст програм включається безпосередньо в HTML-документ і інтерпретується вбудованим у браузер рушієм JavaScript. Він застосовується для часткової автоматизації обробки і маніпуляції даними, що використовуються на сторінці.

Каскадні таблиці стилів CSS є спеціальною мовою, яка використовується для оформлення сторінок, написаних мовами розмітки даних. CSS часто використовується для візуального оформлення сторінок, написаних у форматах HTML та XHTML, а також може застосовуватися до інших видів XML-документів.

Онлайн-платформа для відображення моніторингу метеоінформації для квартир використовує сучасні технології для оптимізації інтерфейсу та обміну даними. В основі її функціоналу лежать AJAX та формат обміну даними JSON. AJAX (Asynchronous JavaScript And XML) дозволяє веб-сторінці взаємодіяти з сервером без перезавантаження, що дозволяє динамічно оновлювати вміст сторінки у фоновому режимі, відправляючи запити та отримуючи відповіді без перерви у роботі користувача.

JSON (JavaScript Object Notation) використовується для структурованої передачі даних між сервером і клієнтом через мережу. Цей текстовий формат

дозволяє легко зрозуміти структуру даних і є популярним в засобах веб-розробки для обміну даними.

Дана онлайн-платформа була реалізована з використанням фреймворку Twitter Bootstrap 3. Цей CSS-фреймворк надає готові компоненти та стилі для розробки сучасних веб-застосунків і сайтів, включаючи типографію, форми, кнопки, таблиці, сітки, навігаційні панелі та інші елементи інтерфейсу. Bootstrap також містить розширення JavaScript для покращення інтерактивності та функціональності веб-застосунків.

Для роботи з онлайн-платформою, клієнту потрібен інтерфейс користувача. Інтерфейс – це набір засобів і правил, що забезпечують взаємодію між пристроями обчислювальної системи та програмами.

Інтерфейс користувача включає в себе засоби для обробки та відображення інформації, максимально пристосовані для зручності користувача. У графічних системах інтерфейс користувача реалізується багатовіконним режимом, змінами кольору, розміру та видимості вікон, такими як прозорість, напівпрозорість та невидимість. Також важливими аспектами є розташування вікон, сортування елементів вікон, гнучкі налаштування окремих елементів (файли, папки, ярлики, шрифти тощо) та можливість налаштування для багатьох користувачів.

Під час роботи з онлайн-платформою використовувався редактор коду Sublime Text 2, який є текстовим редактором, написаним на C++. Цей редактор:

- працює на платформах Linux, OS X і Windows;
- має високу швидкість роботи;
- має приємний інтерфейс з анімаціями;
- гнучко налаштовується;
- підтримує безліч плагінів;
- використовує функцію fuzzy-пошуку.

На рисунку 1.8 показано шаблон онлайн-платформи для відображення моніторингу метеоінформації у квартирах по всьому.

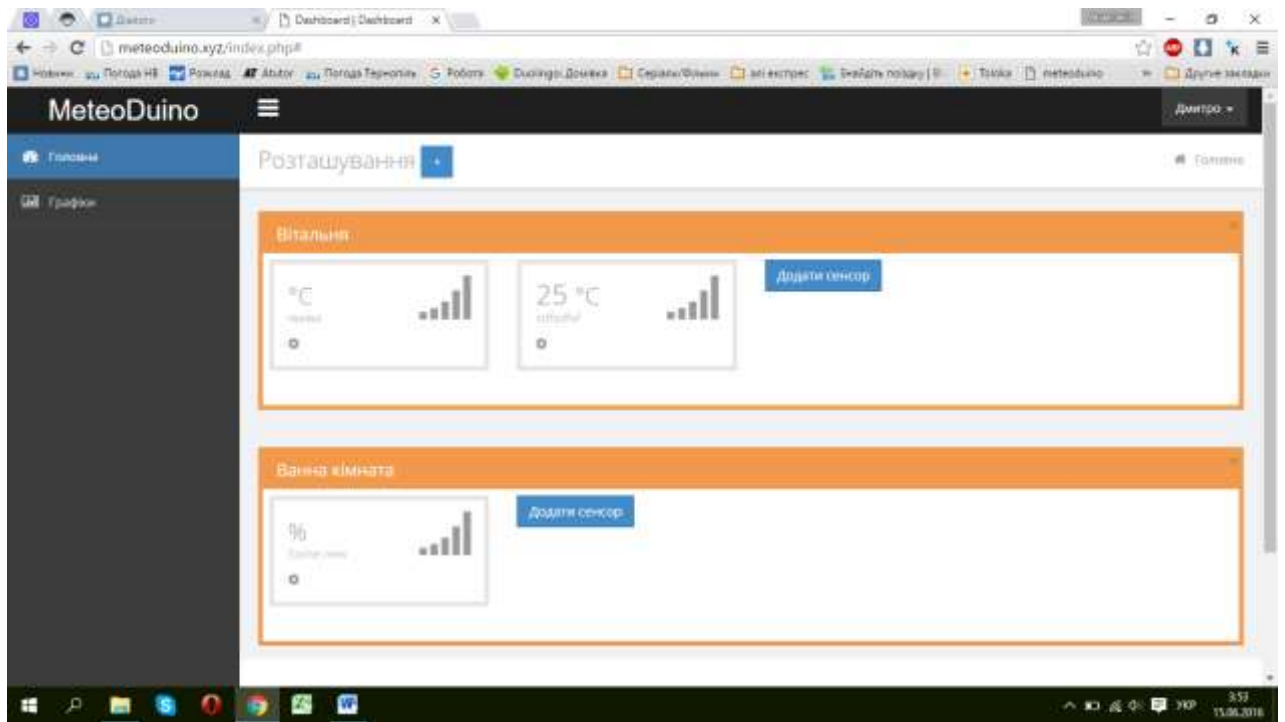


Рисунок 1.8 – Шаблон веб сайту для відображення моніторингу метеоінформації для підприємстві та дому

В основному створення шаблону відбувалося з використанням HTML-розмітки. Це дозволяє створювати шаблони із зовсім іншим виглядом. Для цього спочатку треба створити звичайний текстовий документ у будь-якому текстовому редакторі. Розробка сучасних онлайн-платформ передбачає створення набору компонентів і модулів, які необхідні для реалізації функціоналу цих платформ, орієнтованих на потреби їхніх користувачів. Без цих компонентів і модулів платформи не зможуть забезпечити всю необхідну інформацію та сервіси відвідувачам. Вибір відповідних компонентів і модулів є критичним етапом процесу створення онлайн-платформи і визначає, наскільки ефективною вона буде в подальшому.

1.2.4 Життєвий цикл онлайн-платформи для відображення моніторингу метеоінформації для «розумної» квартири

Життєвий цикл програмного забезпечення – це послідовність етапів, що виконуються у певному порядку протягом певного часу, починаючи з розробки програмного забезпечення і закінчуючи його припиненням використання.

Модель життєвого циклу – це структура, що складається з процесів, робіт і задач, включаючи розробку, експлуатацію і підтримку програмного продукту, що охоплює всі аспекти від визначення вимог до його припинення. Сьогодні найпоширеніші дві моделі: каскадна та спіральна.

Однією з перших є каскадна модель, де кожен етап виконується один раз і передається наступному етапу тільки після верифікації результатів.

Модель розробки ітераціями відображає існуючий спіральний цикл створення системи, де неповне завершення робіт на кожному етапі дозволяє переходити до наступного етапу без очікування повного завершення поточного.

Основні етапи життєвого циклу включають:

- Визначення завдання – встановлення вимог до функціоналу, цілей розробки, визначення сценаріїв використання, труднощів і методів реалізації.
- Проектування системи – розробка інтерфейсу, взаємодії між модулями системи.
- Реалізація системи – розробка системи за допомогою обраної технології.
- Верифікація – перевірка правильності виконання системи.
- Тестування системи – тестування на відповідність вимогам.
- Супровід та підтримка – підтримка системи до її виведення з експлуатації.

Ці етапи можуть змінюватись в залежності від конкретного завдання. Життєвий цикл онлайн-платформи для відображення моніторингу метеоінформації на квартирі буде виглядати так, як зображено на рисунку 1.9.

Виконання перерахованих етапів життєвого циклу онлайн-платформи для відображення моніторингу метеоінформації на квартирі дозволить створити високоякісний програмний продукт і забезпечить його підтримку протягом всього періоду експлуатації.

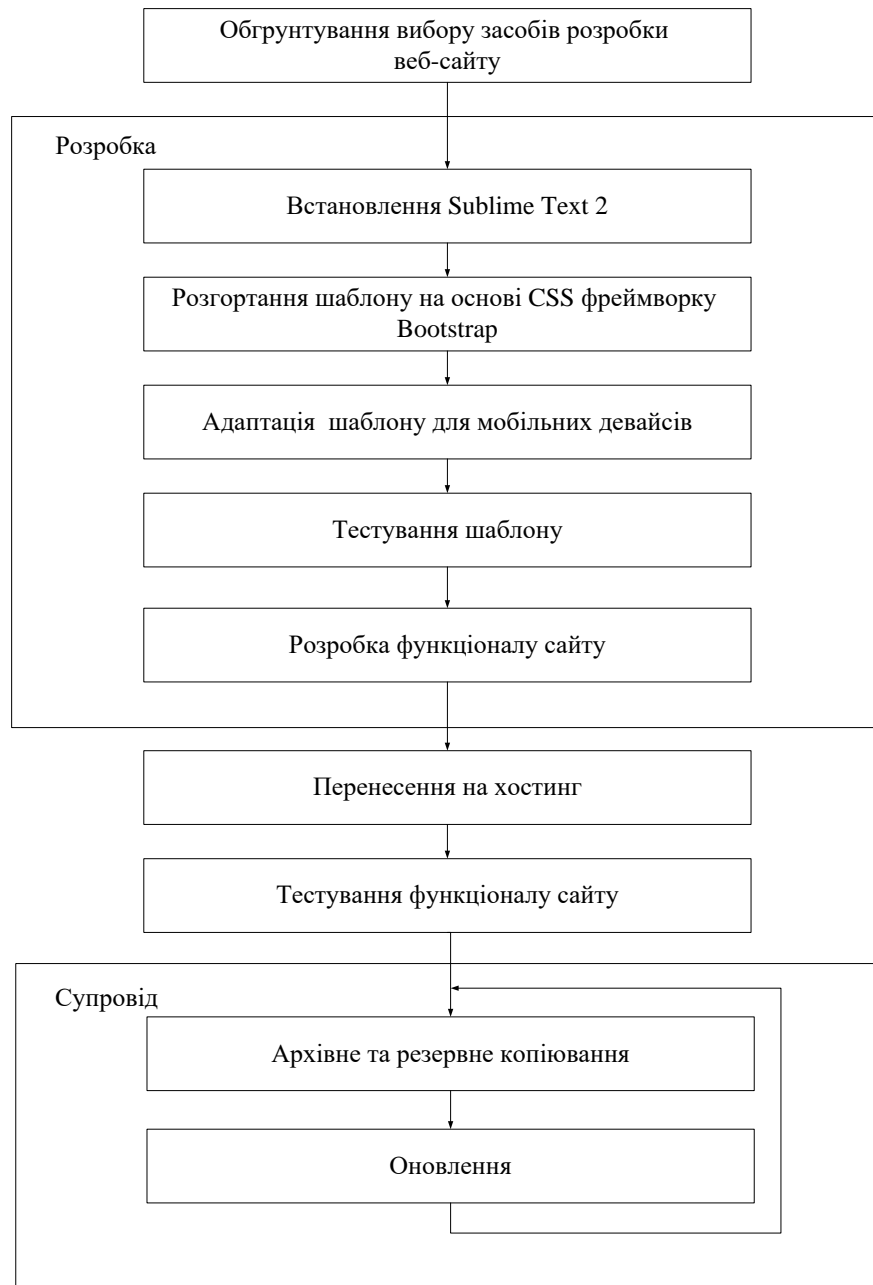


Рисунок 1.9 – Модель життєвого циклу онлайн-платформи для відображення моніторингу метеоінформації для підприємства та дому

Обрані етапи та їх кількість можуть бути змінені за необхідності для редагування структури сайту та забезпечення його коректної роботи з метою збільшення тривалості експлуатації веб-платформи.

1.3 Висновок до першого розділу

В розділі 1 досліджено предметну область і поставлено завдання на розробку онлайн-платформи для відображення моніторингу метеоінформації для «розумної» квартири. Аналізуючи предметну область, було визначено ключові аспекти, пов'язані з вимогами до платформи та її варіантами використання. Вимоги включають не лише точність відображення метеоданих, а й зручність інтерфейсу для кінцевих користувачів.

Проведено аналіз різних варіантів вирішення поставленої задачі, включаючи оцінку методів і вибір оптимального алгоритму. Оптимальний алгоритм вибрано з урахуванням його ефективності та можливості інтеграції з існуючими системами "розумної" квартири. Також обрано середовище розробки, що найкращим чином підходить для реалізації цієї платформи, зокрема з урахуванням масштабованості та безпеки.

Життєвий цикл онлайн-платформи обґрунтовано з урахуванням специфіки предметної області. Він включає етапи від проектування до впровадження, з фокусом на забезпечення надійності, швидкості і зручності користування для кінцевих користувачів. Загальний висновок полягає в тому, що ретельний аналіз і вибір оптимальних рішень на початковому етапі сприятимуть успішному розвитку та ефективній експлуатації платформи для "розумних" квартир.

РОЗДІЛ 2. ПРОЕКТУВАННЯ ОНЛАЙН-ПЛАТФОРМИ ДЛЯ ВІДОБРАЖЕННЯ МОНІТОРИНГУ МЕТЕОІНФОРМАЦІЇ ДЛЯ «РОЗУМНОЇ» КВАРТИРИ

2.1 Моделювання архітектури онлайн-платформи

На закінченні етапів дослідження предметної області, визначення вимог до онлайн-платформи, аналізу варіантів використання, перегляду існуючих рішень та вибору технічних засобів для створення онлайн-платформи для показу моніторингу метеоінформації для квартир, було виконано проектування архітектури, яку представлено на рисунку 2.1.

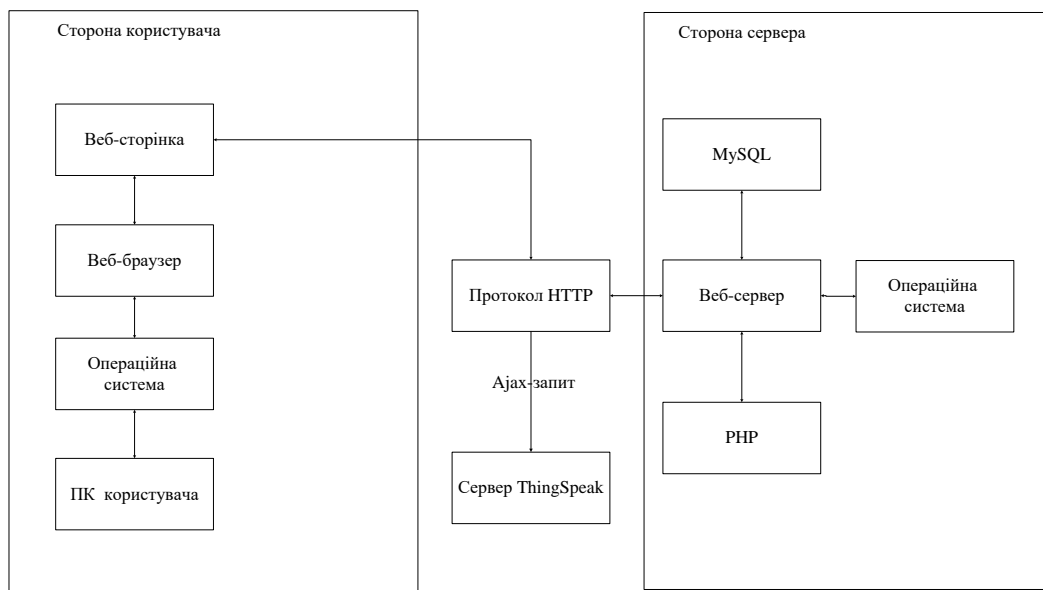


Рисунок 2.1 – Архітектура онлайн-платформи для відображення моніторингу метеоінформації для підприємства та дому

Робота онлайн-платформи для відображення моніторингу метеоінформації для квартир відбувається на двох основних рівнях: зі сторони користувача і зі сторони сервера. На стороні сервера розташований сам веб-сайт з усіма необхідними компонентами і модулями, а на стороні користувача – персональний комп'ютер з операційною системою і веб-браузером, через який відбувається завантаження веб-сторінок сайту.

Серверна частина включає веб-сервер з власною операційною системою, клієнт MySQL і PHP-інтерпретатор. Всі дані між сервером і користувачем передаються через протокол HTTP. Зовнішнє джерело, яким є сервер ThingSpeak, використовується для отримання метеоданих через Ajax-запити, щоб забезпечити коректне відображення інформації на веб-сайті.

Ajax – це технологія, яка дозволяє веб-сайту динамічно оновлювати інформацію без перезавантаження сторінки. Завдяки цьому підходу час на веб-сайті відображається динамічно, а оновлення годинника не вимагає повного перезавантаження веб-сторінки.

2.2 Проектування структури онлайн-платформи

Під час взаємодії з онлайн-платформою, що відображає моніторинг метеоінформації для квартир, користувач бачить лише користувацький інтерфейс – набір веб-сторінок, які змінюються відповідно до його дій. Однак справжня робота платформи базується на взаємодії багатьох компонентів, веб-сервера та різних інтерпретаторів, які постійно обробляють і відправляють програмний код та інформацію на сторону користувача. Веб-браузер на користувацькому ПК відображає цю інформацію у зручному для сприйняття вигляді.

Кожен сайт має унікальну структуру, спроектовану для досягнення своїх цілей та функціональності. Це можуть бути як готові рішення, так і індивідуально розроблені компоненти для виконання специфічних завдань.

Для відображення структури онлайн-платформи, що відображає моніторинг метеоінформації для квартир, розробленої на основі фреймворку Bootstrap, доцільним є представлення всіх компонентів, що взаємодіють з платформою, у вигляді діаграми компонентів. Така діаграма представлена на рисунку 2.2.

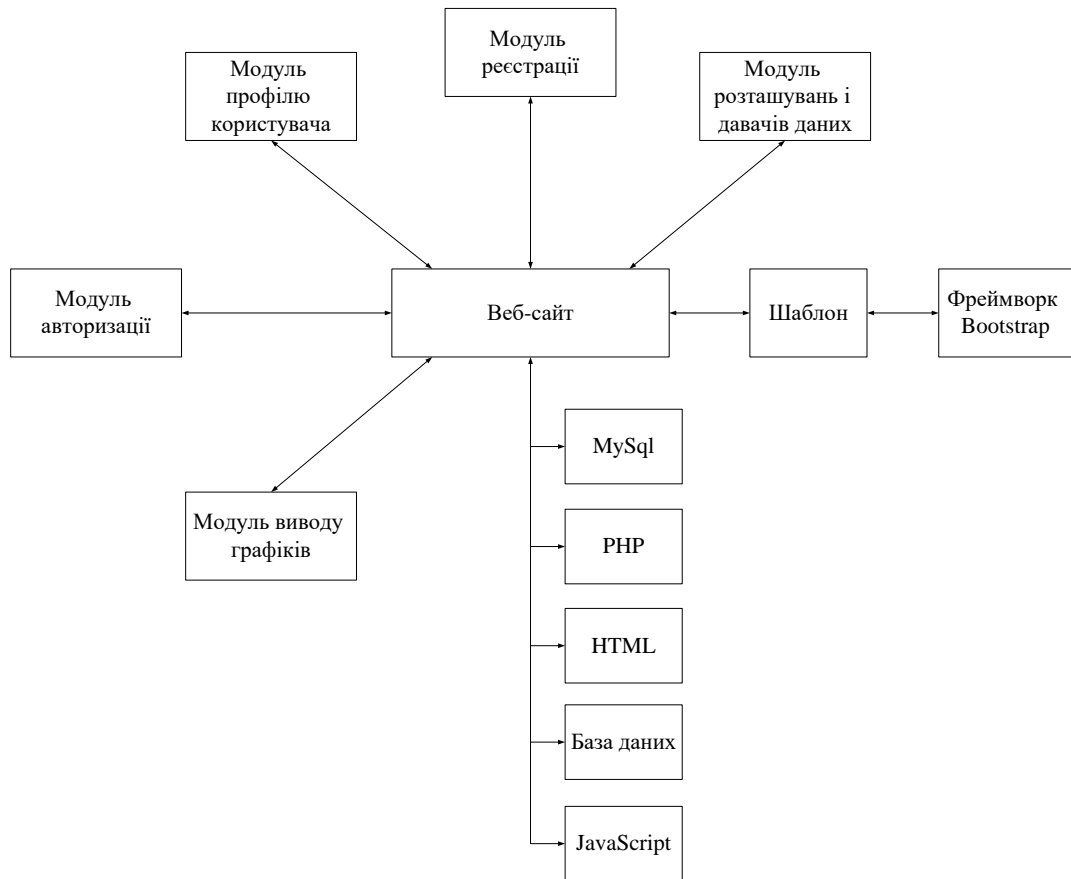


Рисунок 2.2 – Діаграма компонентів онлайн-платформи

На цій діаграмі показано, які компоненти працюють і взаємодіють безпосередньо з онлайн-платформою.

2.3 Проектування поведінки онлайн-платформи

Найбільш вигідним способом для відображення функціонування онлайн-платформи, яка показує моніторинг метеоінформації для квартир та дому, є створення функціональної схеми, яка демонструє взаємодію окремих компонентів. Функціональну схему платформи представлено на рисунку 2.3.

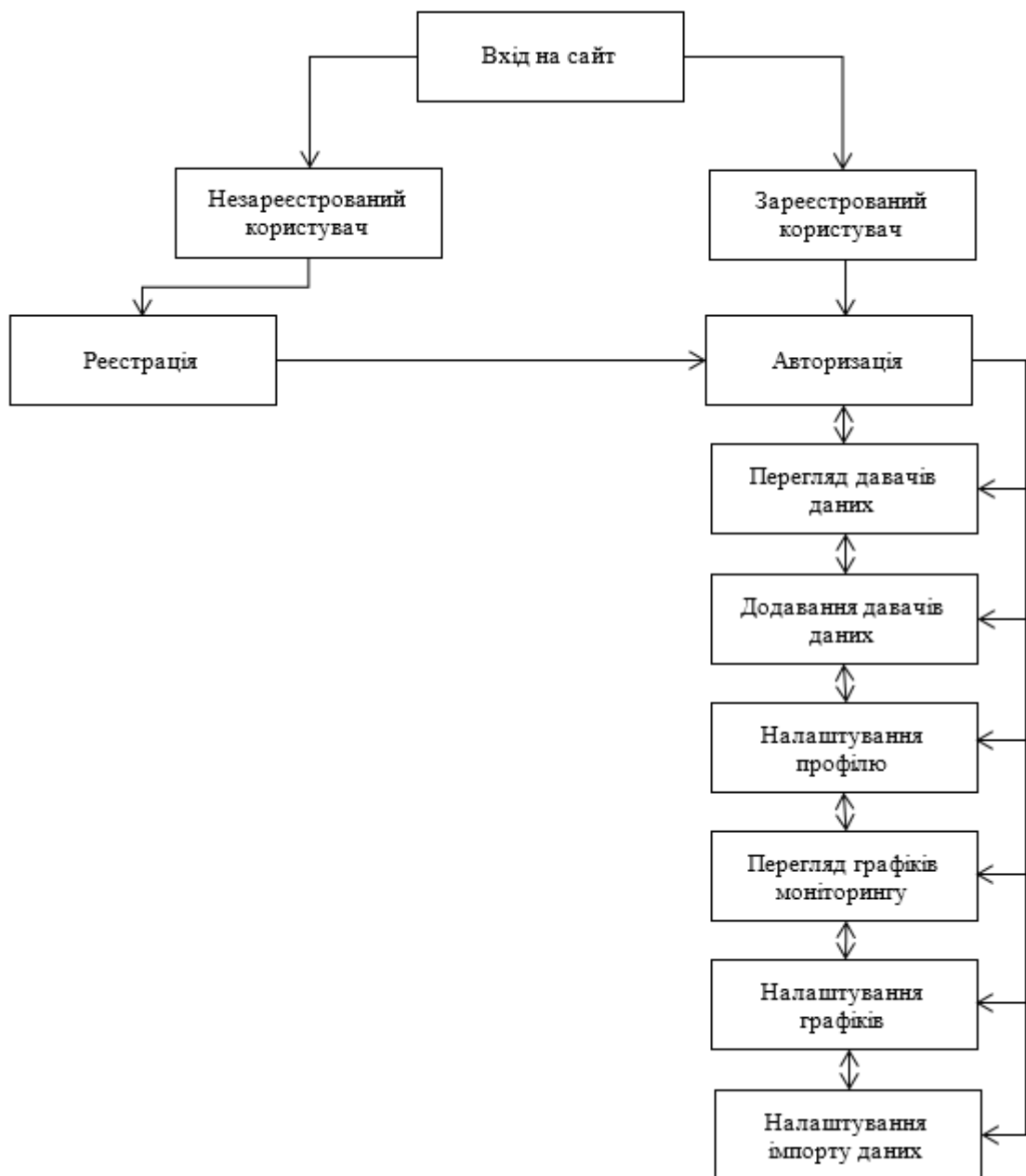


Рисунок 2.3 – Функціональна схема роботи онлайн-платформи для відображення моніторингу метеоінформації для підприємства та дому

Будь-який програмний продукт характеризується своїми можливостями та відповіддю на дії користувача. Ця функціональна схема показує, яким чином працює онлайн-платформа для відображення моніторингу метеоінформації для квартир та дому. Вона відображає всі необхідні операції сайту.

Після входу на платформу незареєстрованому користувачу пропонується зареєструватись, а зареєстрованому користувачу увійти до системи. Після введення в систему користувач має можливість користуватись усіма функціями

онлайн-платформи для відображення моніторингу метеоінформації для квартир та дому.

2.4 Проектування інтерфейсу онлайн-платформи

Інтерфейс є першим, на що звертає увагу користувач. Від нього залежить, чи буде у користувача бажання повертатися на сайт і використовувати його регулярно. Це залежить від багатьох чинників, таких як:

- колірна палітра онлайн-платформи;
- стиль шрифту;
- дизайн онлайн-платформи;
- розміщення елементів на сторінці;
- функціональні можливості онлайн-платформи.

Після визначення функціоналу онлайн-платформи для відображення моніторингу метеоінформації для підприємства та дому були розроблені макети його сторінок. Макет головної сторінки показано на рисунку 2.4.

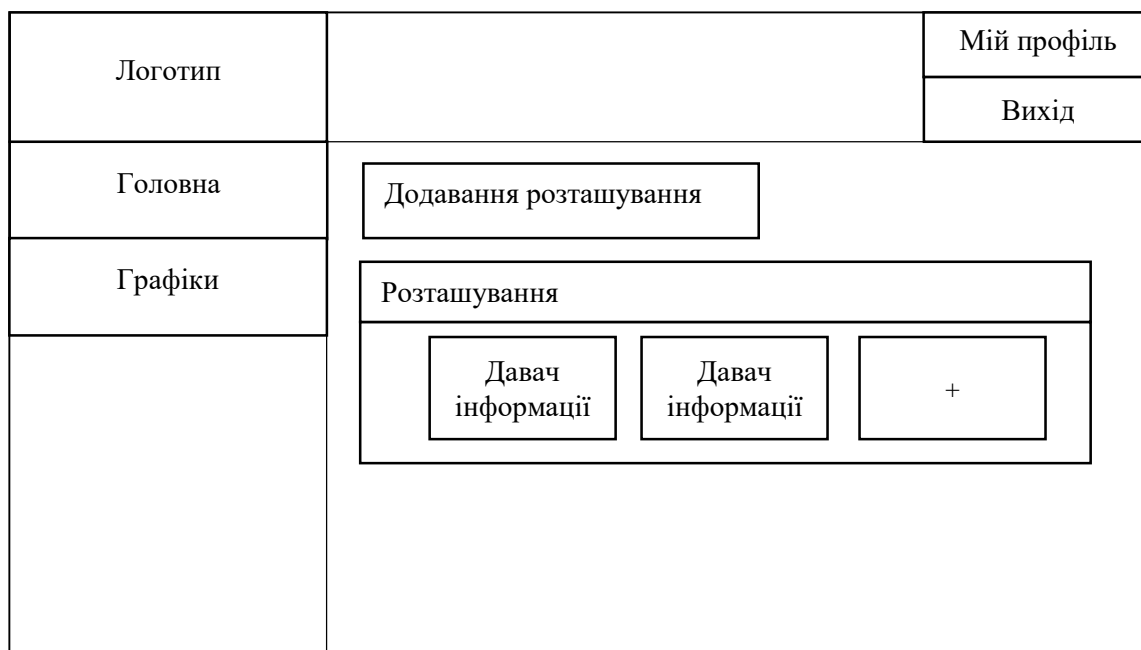


Рисунок 2.4 – Макет головної сторінки онлайн-платформи

На головній сторінці розташовано наступні елементи:

- логотип сайту;

- посилання на сторінку "Головна";
- посилання на сторінку "Графіки";
- посилання на сторінку "Мій профіль";
- кнопка для завершення сесії користувача;
- кнопка для додавання розташування даних інформації;
- кнопка для додавання даних інформації.

Після проектування макету головної сторінки онлайн-платформи, було розроблено макет сторінки відображення графіків, який показано на рисунку 2.5.

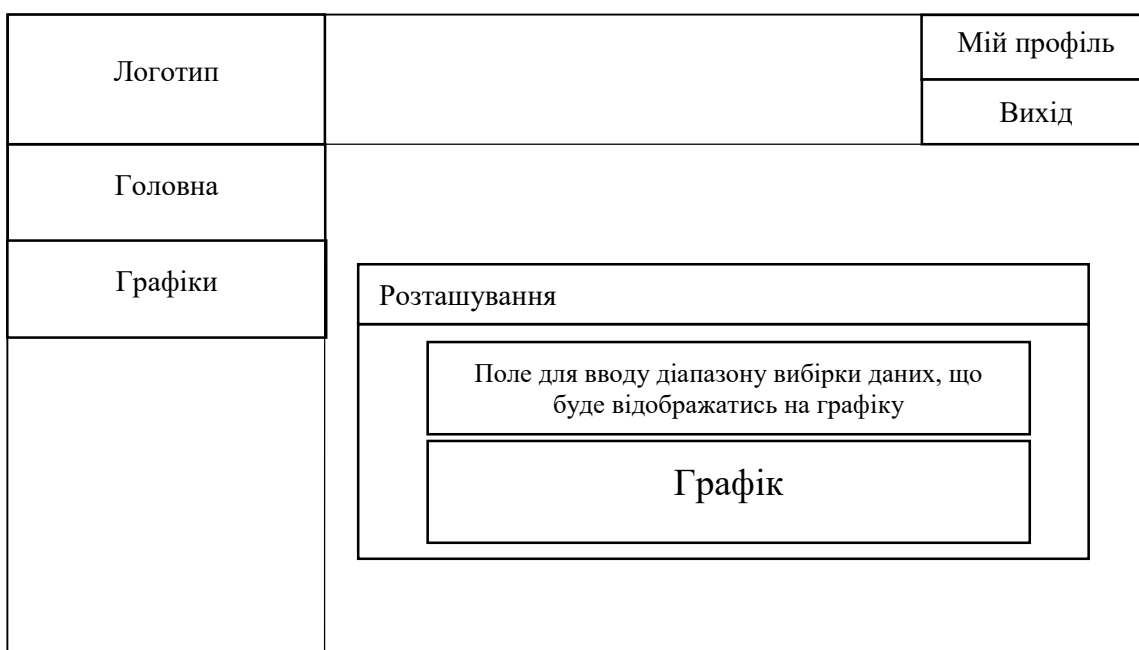


Рисунок 2.5 – Макет сторінки відображення графіків

На сторінці "Графіки" розташовані наступні компоненти:

- логотип сайту;
- посилання на сторінку "Головна";
- посилання на сторінку "Графіки";
- посилання на сторінку "Мій профіль";
- кнопка для завершення сесії користувача;
- поле для введення діапазону вибірки даних, які будуть відображені на графіку;
- сам графік.

Далі було проведено проектування макету сторінки "Мій профіль", призначенням якої є налаштування особистого профілю користувача. Цей розділ поділений на дві сторінки: "Профіль користувача" та "Налаштування".

Макет сторінки "Профіль користувача" показано на малюнку 2.6.

Логотип			Мій профіль
			Вихід
Головна	Аватар	Профіль користувача	Налаштування
Графіки	Завантажити аватар	Налаштування аккаунту	
	Інформація про користувача	Email	Текстове поле
		Пароль	Текстове поле
		Підтвердження паролю	Текстове поле
		Налаштування профілю	
		Ім'я	Текстове поле
		Прізвище	Текстове поле
		Стать	Список
		Про себе	Велике текстове поле
		Регіональні параметри	
		Країна	Текстове поле
		Місто	Текстове поле
		Зберегти	

Рисунок 2.6 – Макет сторінки «Профіль користувача»

Сторінка "Профіль користувача" включає наступні компоненти:

- логотип сайту;
- посилання на сторінку "Головна";
- посилання на сторінку "Графіки";
- посилання на сторінку "Мій профіль";

- кнопка для завершення сесії користувача;
- аватар користувача;
- кнопка для завантаження аватара користувача;
- інформація про користувача;
- налаштування облікового запису (email, пароль, підтвердження паролю);
- налаштування профілю (ім'я, прізвище, стать, про себе);
- регіональні параметри (країна, місто);
- кнопка для збереження даних про користувача;
- поле для списку;
- текстові поля для введення даних користувача.

Після проектування сторінки "Профіль користувача", можна створити макет сторінки "Налаштування", на якій будуть розміщені налаштування хмарного сервісу та експорту даних, зображені на рисунку 2.7.

Логотип			Мій профіль
			Вихід
Головна	Аватар	Профіль користувача	Налаштування
Графіки	Завантажити аватар	Авторизація хмарного сервісу	Список
	Інформація про користувача	Хмарний сервіс	Текстове поле
		Channel ID	
		Налаштування агрегації даних	
		Інтервал укрупнення даних	Список
		Розташування	Список
		Налаштування експорту даних	
		Часова зона	Список
		Експорт даних в БД	Прапорець
		Зберегти	

Рисунок 2.7 – Макет сторінки «Налаштування»

На сторінці "Профіль користувача" розташовані такі елементи:

- логотип сайту;
- посилання на сторінку "Головна";
- посилання на сторінку "Графіки";
- посилання на сторінку "Мій профіль";
- кнопка для завершення сесії користувача;
- аватар користувача;
- кнопка для завантаження аватара користувача;
- інформація про користувача;
- авторизація у хмарному сервісі (хмарний сервіс, Channel ID);
- налаштування агрегації даних (інтервал укрупнення даних, розташування);
- налаштування експорту даних (часова зона, експорт даних в базу даних);
- текстові поля для введення даних користувача.

Після розробки макету сторінки «Профіль користувача», був спроектований макет сторінки «Авторизація», призначений для входу користувача в систему, зображений на рисунку 2.8.

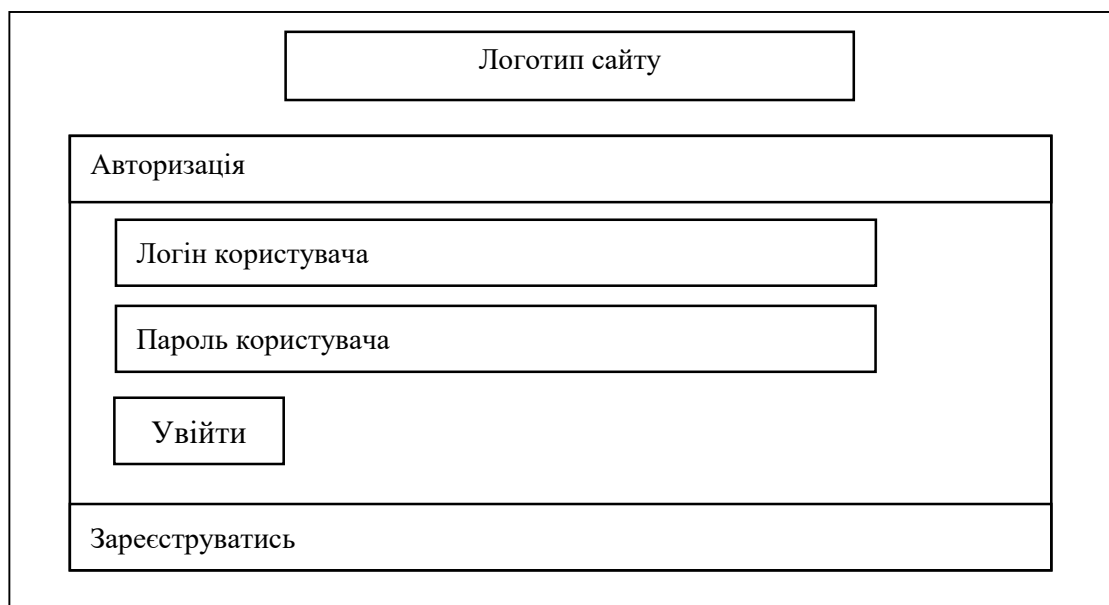


Рисунок 2.8 – Макет сторінки авторизації

Відповідно до проектування сторінки "Авторизація", наступним етапом є розробка макету сторінки "Реєстрація", призначеної для реєстрації нових користувачів у системі, яка представлена на рисунку 2.9.

Рисунок 2.9 – Макет сторінки реєстрації

Перехід між функціями онлайн-платформи відбуватиметься через вибір користувачем відповідних пунктів меню. Доступ до функціоналу платформи буде залежати від того, чи зареєстрований користувач.

Після проектування всіх необхідних макетів сторінок можна буде визначити розташування ключових елементів шаблону. Це означає проведення його декомпозиції.

Декомпозиція шаблону дозволяє структурувати його, визначити основні позиції, де під час подальшої розробки можна розмістити компоненти та модулі для забезпечення всього необхідного функціоналу і створення зручного та зрозумілого інтерфейсу, який максимально відповідає спроектованим макетам сторінок онлайн-платформи для відображення моніторингу метеоінформації для підприємства та дому.

Головну сторінку онлайн-платформи для відображення моніторингу метеоінформації для підприємства та дому зображено на рисунку 2.10.

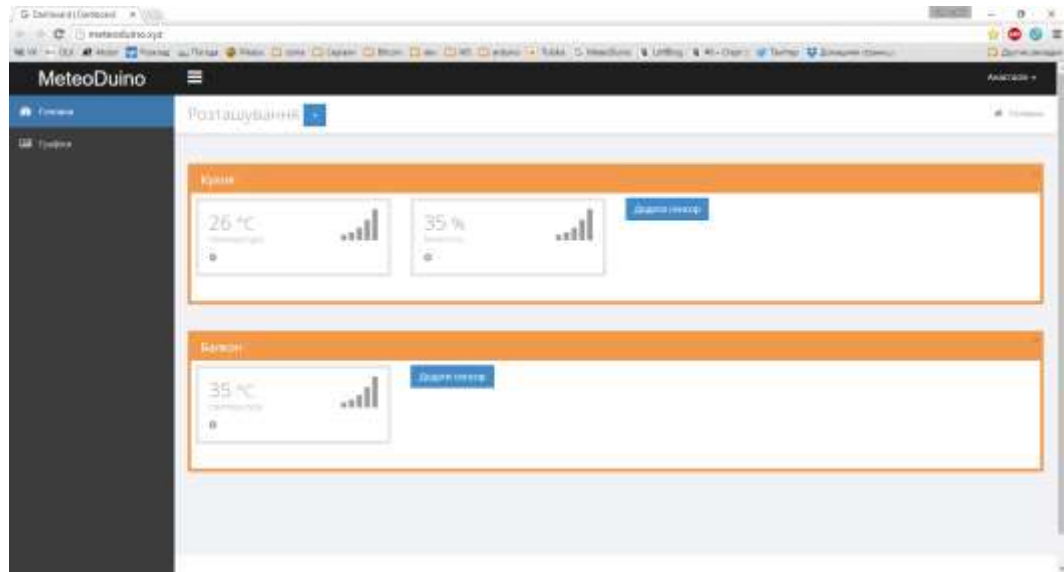


Рисунок 2.10 – Головна сторінка онлайн-платформи для відображення моніторингу метеоінформації для підприємства та дому

Оскільки онлайн-платформа повинна забезпечувати відображення моніторингу метеоінформації для підприємства та дому, вона має бути доступною в будь-який час і в будь-якому місці з доступом до глобальної мережі Інтернет. Користувачі повинні мати можливість користуватися платформою не лише через персональні комп'ютери і ноутбуки, а й через портативні пристрої, такі як смартфони та планшети, які мають різні розміри та співвідношення екранів.

Для забезпечення доступності платформи на різних пристроях прийнято рішення реалізувати адаптивний дизайн. Адаптивний веб-дизайн забезпечує коректне відображення сайту на різних пристроях, які підключені до Інтернету, динамічно адаптуючи контент під різні розміри вікон браузера. Головною метою адаптивного дизайну є універсальність відображення вмісту платформи для різних типів пристроїв. За технологією адаптивного веб-дизайну не потрібно створювати окремі версії платформи для кожного типу пристрою. Один і той же сайт може ефективно працювати на смартфонах, планшетах, ноутбуках і телевізорах з доступом в Інтернет, охоплюючи всі можливі пристрої. Роботу адаптивного дизайну можна побачити на рисунку 2.11.

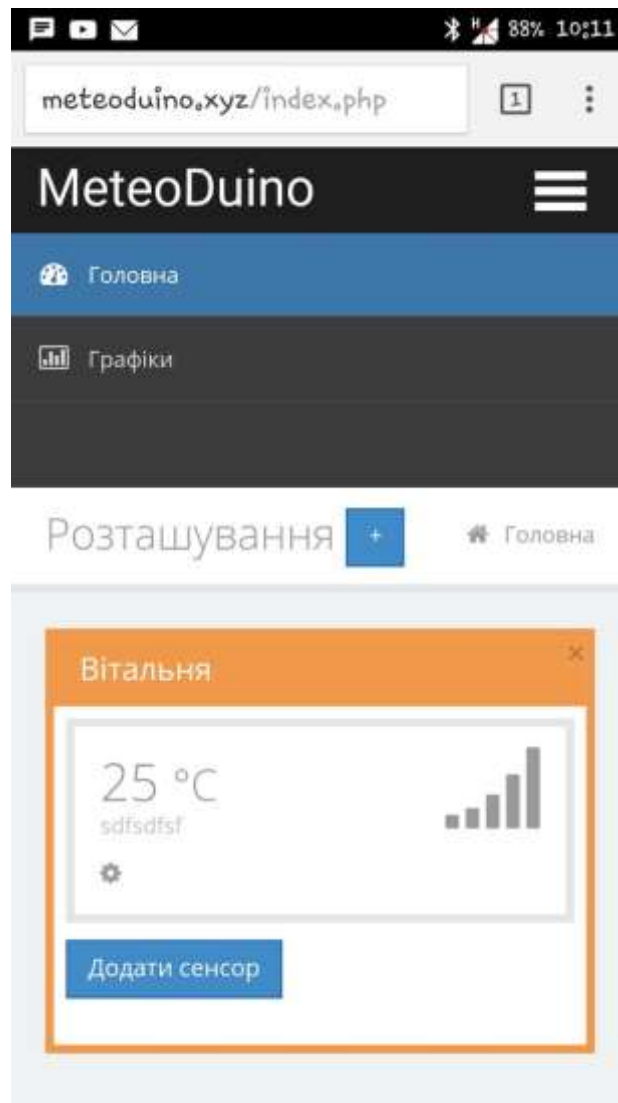


Рисунок 2.11 – Робота адаптивного дизайну онлайн-платформи для відображення моніторингу метеоінформації для підприємства та дому

Адаптивний дизайн сайту дозволяє зберігати практично весь контент в повному обсязі для максимальної кількості користувачів на будь-яких пристроях, відмінно від мобільних версій сайтів, які часто пропонують обмежену інформацію. Цей підхід дозволяє автоматично та організовано адаптувати контент і структуру сайту для різноманітних пристроїв, включаючи мобільні телефони і широкоформатні монітори.

2.5 Висновок до другого розділу

Другий розділ присвячено проектуванню онлайн-платформи для відображення моніторингу метеоінформації для «розумної» квартири.

Моделювання архітектури включало розгляд різних можливостей структурування системи з урахуванням потреб високої доступності та швидкодії. Проектування структури детально визначило складові частини системи та їх взаємозв'язки для забезпечення ефективної інтеграції та сумісності з іншими технологіями "розумних" квартир.

Проектування поведінки платформи фокусувалося на розробці логічного ядра системи, яке забезпечує надійність та точність збору, обробки та відображення метеоданих. Цей аспект був критично важливим для забезпечення коректності функціонування платформи в усіх умовах експлуатації.

Проектування інтерфейсу врахувало не лише естетичні аспекти, але й зручність використання для кінцевих користувачів. Інтерфейс платформи був розроблений таким чином, щоб надавати інтуїтивно зрозумілу інформацію і забезпечувати зручний доступ до всіх функцій системи.

У цілому, другий розділ визначив ключові аспекти проектування онлайн-платформи для моніторингу метеоінформації для «розумних» квартир, забезпечивши фундаментальні засади для успішного подальшого розвитку та реалізації проекту.

РОЗДІЛ 3. РОЗГОРТАННЯ ТА НАЛАШТУВАННЯ ОНЛАЙН-ПЛАТФОРМИ ДЛЯ ВІДОБРАЖЕННЯ МОНІТОРИНГУ МЕТЕОІНФОРМАЦІЇ ДЛЯ «РОЗУМНОЇ» КВАРТИРИ

3.1 Перелік етапів розгортання онлайн-платформи

Перед початком користування онлайн-платформою необхідно виконати етап розгортання. Цей процес складається з послідовних етапів, які забезпечують готовність платформи до повноцінного використання. Розгортання є важливою частиною життєвого циклу онлайн-платформи і включає встановлення та налаштування необхідних компонентів для ефективного користування та оптимальної роботи платформи.

Етапи розгортання онлайн-платформи для відображення моніторингу метеоінформації для підприємства та дому можна побачити на рисунку 3.1.

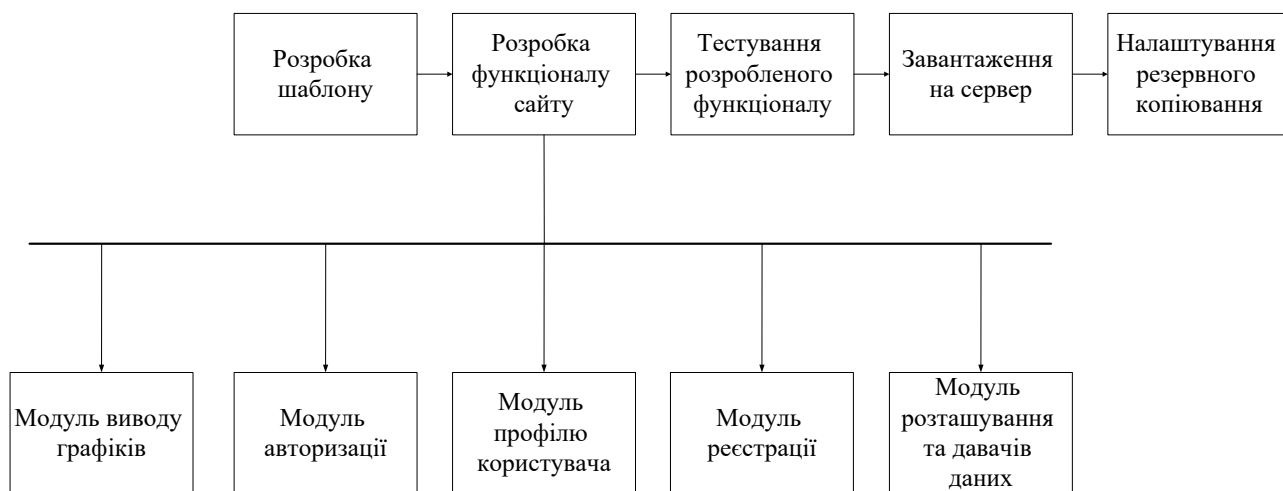


Рисунок 3.1 – Етапи розгортання онлайн-платформи для відображення моніторингу метеоінформації для підприємства та дому

Наведена схема показує основні етапи розгортання онлайн-платформи:

- Розробка шаблону.
- Розробка функціоналу сайту.
- Тестування розробленого функціоналу.
- Завантаження на сервер.

- Налаштування резервного копіювання.

Після виконання усіх цих етапів розгортання онлайн-платформи та проведення кінцевих налаштувань він готовий до експлуатації.

3.2 Розміщення онлайн-платформи на хостинг

Розміщення на хостингу є ключовою частиною запуску онлайн-платформи. Хостинг забезпечує надання простору на сервері, підключення до мережі та інших ресурсів для зберігання фізичних даних, які доступні через Інтернет. Послуги хостингу можуть включати різноманітні апаратні та програмні рішення.

Хостинг може бути безкоштовним або платним. Безкоштовні хостинги зазвичай заробляють на розміщенні реклами на своїх сайтах або наданні платних додаткових послуг разом з безкоштовним пакетом.

Для розміщення онлайн-платформи для відображення моніторингу метеоінформації для квартири був обраний безкоштовний хостинг, який повністю відповідає потребам платформи. Спочатку потрібно зареєструвати домен для платформи. Доменна система імен (DNS) – це розподілена система, яка перетворює імена хостів (комп'ютерів або інших мережевих пристроїв) в IP-адреси. Кожен комп'ютер в Інтернеті має свою унікальну IP-адресу, але використання доменних імен значно спрощує запам'ятовування адрес.

Процес реєстрації домену передбачає кілька кроків, які забезпечують прив'язку унікального доменного імені до конкретного користувача чи організації. Ось основні етапи процесу реєстрації домену:

- Вибір доменного імені: Користувач обирає унікальне ім'я для свого сайту або онлайн-платформи. Доменне ім'я повинно відповідати певним правилам (наприклад, відсутність спеціальних символів, довжина тощо) та бути унікальним на своєму рівні домену (наприклад, .com, .net, .org).

- Вибір реєстратора: Користувач обирає компанію-реєстратора, яка буде проводити реєстрацію домену. Реєстратори мають доступ до офіційних баз

даних доменних імен і мають повноваження реєструвати домени для своїх клієнтів.

– Перевірка доступності домену: Реєстратор проводить перевірку наявності обраного користувачем доменного імені в обраному рівні домену. Якщо ім'я доступне, користувач може переходити до наступного етапу. Якщо ім'я зайняте, реєстратор пропонує інші варіанти або зміни імені.

– Реєстрація домену: Після підтвердження доступності домену, користувач вводить необхідні контактні дані і здійснює оплату за реєстрацію на обраному терміні (зазвичай від одного до десяти років).

– Дані WHOIS: Після успішної реєстрації, дані про домен та контактні дані власника додаватимуться до публічної бази даних WHOIS, яка містить інформацію про всі зареєстровані доменні імена.

– Активація домену: Після завершення реєстрації і публікації даних у базі WHOIS домен стає активним і готовим до використання для веб-сайту або іншої цифрової послуги.

Процес реєстрації домену може тривати від кількох хвилин до декількох днів в залежності від реєстратора та обраних налаштувань.

Після реєстрації домена та хостингу здійснюється завантаження файлів на веб-сервер, для цього використовується програма Filezilla яка зображена на рисунку 3.2. FileZilla – це безкоштовний, багатоплатформний клієнт FTP з відкритим кодом.

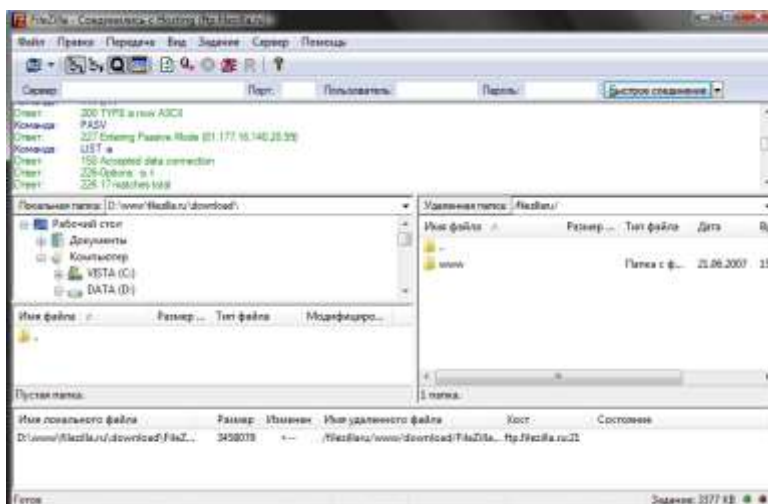
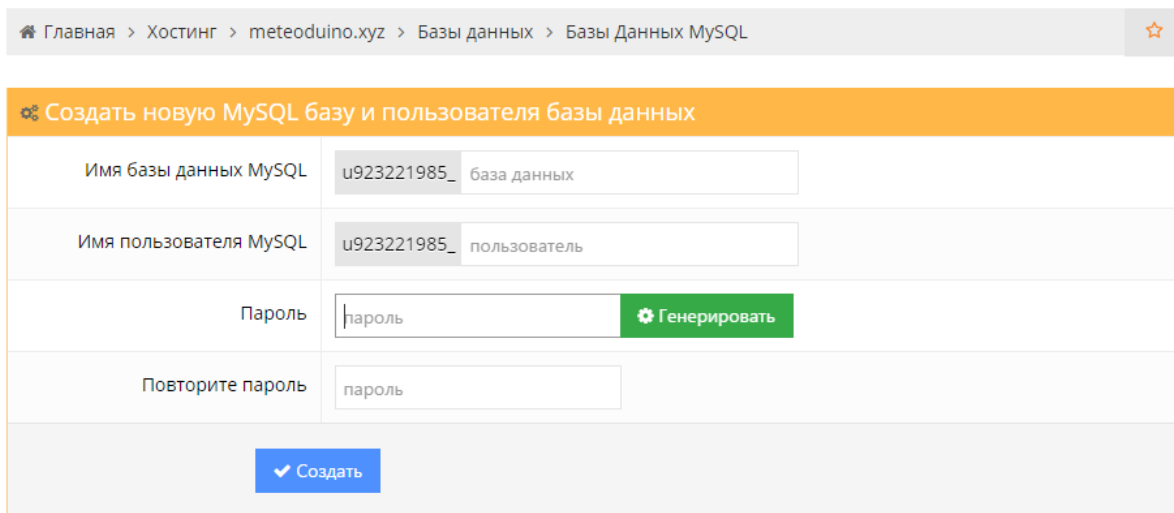


Рисунок 3.2 – Зовнішній вигляд програми FileZilla

Після завантаження файлів на веб-сервр на хостингу створюється база даних показана на рисунку 3.3.



The screenshot shows a web browser window with a breadcrumb trail: Главная > Хостинг > meteoduino.xyz > Базы данных > Базы Данных MySQL. Below the breadcrumb is a title bar: Создать новую MySQL базу и пользователя базы данных. The form contains four input fields: 'Имя базы данных MySQL' with the value 'u923221985_ база данных', 'Имя пользователя MySQL' with the value 'u923221985_ пользователь', 'Пароль' with the value 'пароль' and a 'Генерировать' button, and 'Повторите пароль' with the value 'пароль'. At the bottom of the form is a blue 'Создать' button.

Рисунок 3.3 – Створення бази даних

База даних є важливою складовою даної онлайн-платформи.

Після виконання перерахованих вище етапів можна переходити до тестування онлайн-платформи для відображення моніторингу метеоінформації для підприємства та дому.

3.3 Структура бази даних

Для максимально точного аналізу предметної області, потрібно чітко розмежувати досліджувані об'єкти між собою. У зв'язку з тим, що на сайті передбачено розподілення прав доступу, то дані можуть взаємодіяти, як показано на рисунку 3.4.



Рисунок 3.4 – Діаграма потоків даних

Розроблена база складається із п'яти основних таблиць – «Users», «Cloud_settings», «Sensors», «Locations» та «Sensor_data».

Таблиця «Locations» містить розташування датчиків інформації. Структура приведена нижче у таблиці 3.1.

Таблиця 1.1 – Розташування датчиків інформації

Ім'я поля	Тип даних і розмір	Опис
Id	int(11)	Ідентифікатор розташування
User_id	int(11)	Ідентифікатор користувача
Location_name	varchar(120)	Назва розташування

Таблиця «Cloud_settings» призначення для зберігання налаштувань хмарного сервісу. Структура приведена нижче у таблиці 3.2.

Таблиця 3.2 – Зберігання налаштувань хмарного сервісу

Ім'я поля	Тип даних і розмір	Опис
User_id	int(11)	Ідентифікатор користувача
Cloud_servise	varchar(120)	Назва хмарного сервісу
Channel_id	int(1)	Ідентифікатор каналу
Timezone	varchar(55)	Часова зона
ApiKey	varchar(120)	Ключ апи
Export_timescale	int(1)	Агригація даних
Export_enable	int(1)	Увімкнення експорту даних бази даних

Таблиця «Sensors» містить дані про давачі інформації які додані в систему. Структура приведена нижче у таблиці 3.3.

Таблиця 3.3 – Дані про давачі інформації

Ім'я поля	Тип даних і розмір	Опис
Sensor_id	int(11)	Ідентифікатор давача даних
User_id	int(11)	Ідентифікатор користувача
Sensor_name	varchar(55)	Назва давача даних
Location	varchar(120)	Розташування
Measurement	varchar(55)	Одиниці вимірювання
Cloud_servise	varchar(55)	Хмарний сервіс
Cloud_sensor_id	int(1)	Ідентифікатор давача даних на хмарному сервісі
Enable_chart	tinyint(1)	Увімкнути графіки

Таблиця «Users» містить інформацію про користувачів. Структура вказана у таблиці 3.4.

Таблиця 3.4 – Інформація про користувачів

Ім'я поля	Тип даних і розмір	Опис
Id	int(11)	Ідентифікатор користувача
Firstname	varchar(55)	Ім'я
Lastname	varchar(120)	Прізвище
City	varchar(120)	Місто
Country	varchar(120)	Країна
Birthday	date	Дата народження
Sex	varchar(25)	Стать
Password	varchar(255)	Пароль
Email	varchar(255)	Емейл
Avatar	varchar(120)	Фото користувача
About	text	Інформація про користувача
Status	int(4)	Підтвердження реєстрації
Activecode	int(11)	Код активації для підтвердження реєстрації
Timestamp	tinyint(11)	Відмітка часу

Таблиця «Sensor_data» містить інформацію від датчиків даних. Структура вказана у таблиці 3.5.

Таблиця 3.5 – Інформація від датчиків даних

Ім'я поля	Тип даних і розмір	Опис
1	2	3
Id	int(11)	Ідентифікатор датчика даних
User_id	int(11)	Ідентифікатор користувача
Channel_id	varchar(55)	Ідентифікатор каналу
Created_at	datetime	Дата та час отримання даних
Field1	float	Дані від датчика даних 1
Field2	float	Дані від датчика даних 2

Продовження таблиці 3.5

1	2	3
Field3	float	Дані від давача даних 3
Field4	float	Дані від давача даних 4
Field5	float	Дані від давача даних 5
Field6	float	Дані від давача даних 6
Field7	float	Дані від давача даних 7
Field8	float	Дані від давача даних 8

Розроблена база даних онлайн-платформи для моніторингу метеоінформації для підприємства та дому цілком відповідає його потребам.

3.4 Тестування онлайн-платформи для відображення моніторингу метеоінформації для підприємства та дому

3.4.1 Тестування користувацького інтерфейсу

Для забезпечення ефективної експлуатації сайту необхідно провести тестування користувацького інтерфейсу з метою уникнення непередбачених проблем. Користувачі використовують різноманітні операційні системи, браузерери та пристрої для доступу до веб-сайтів, що впливає на безліч аспектів його роботи.

Для виконання тестування користувацького інтерфейсу онлайн-платформи, яка відображає моніторинг метеоінформації для підприємства та дому, потрібно:

- перевірити сумісність онлайн-платформи з різними веб-браузерами;
- протестувати роботу адаптивного дизайну сайту.

Починаючи з перевірки відображення онлайн-платформи у браузері Google Chrome, зображеного на рисунку 3.5.

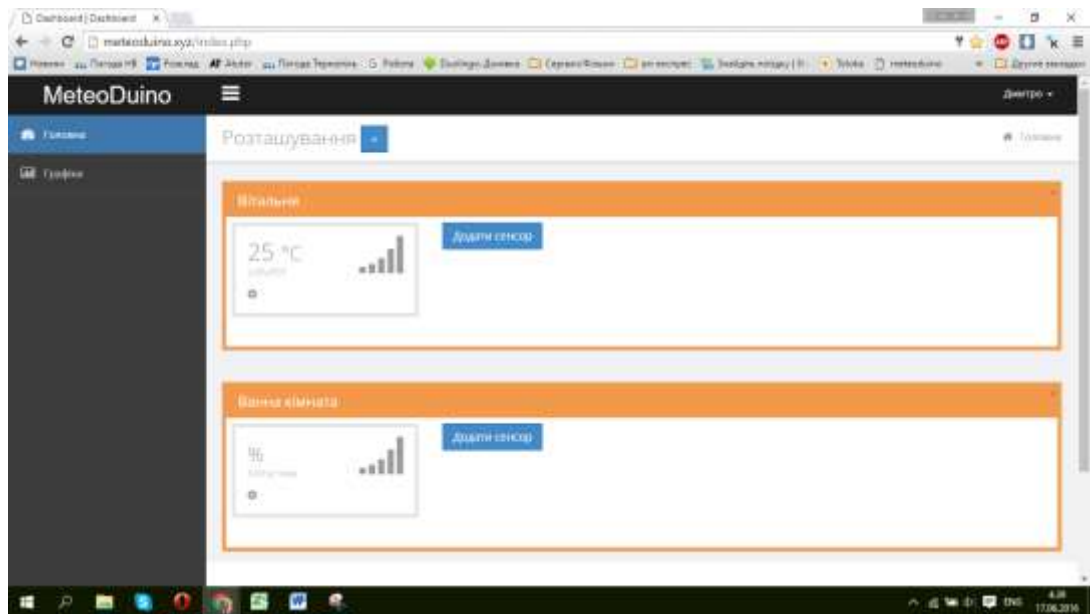


Рисунок 3.5 – Відображення онлайн-платформи у браузері Google Chrome

Тестування відображення онлайн-платформи у браузері Opera, відображено на рисунку 3.6.

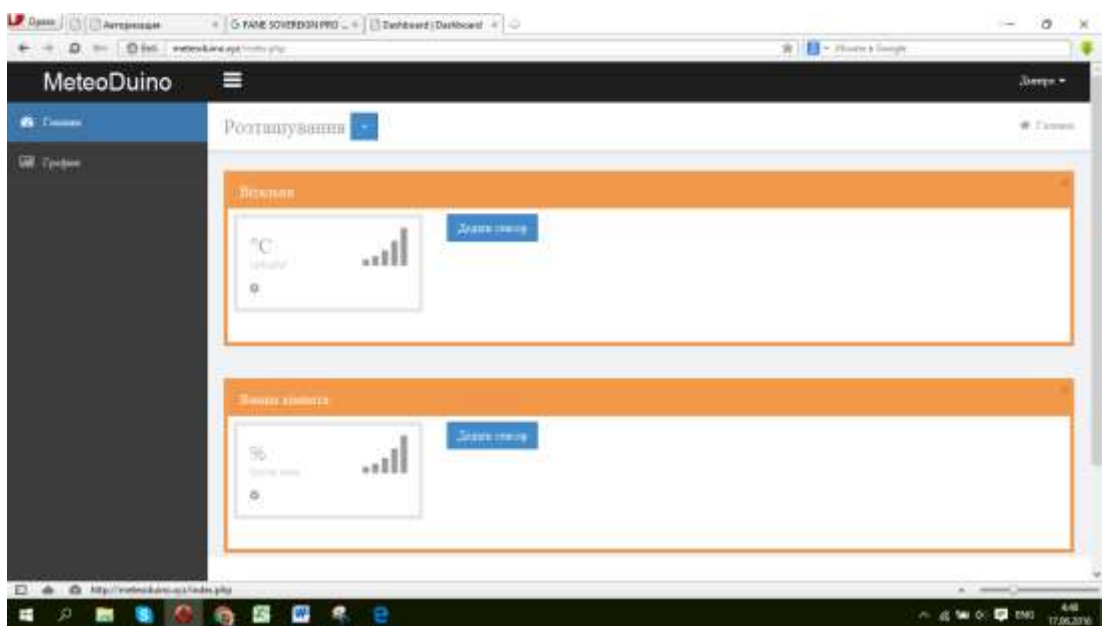


Рисунок 3.6 – Відображення онлайн-платформи у браузері Opera

На рисунку видно, що сайт завантажується у повному обсязі та із всіма структурними елементами.

Перевірку правильності завантаження онлайн-платформи у браузері Microsoft Edge, продемонстровано на рисунку 3.7.

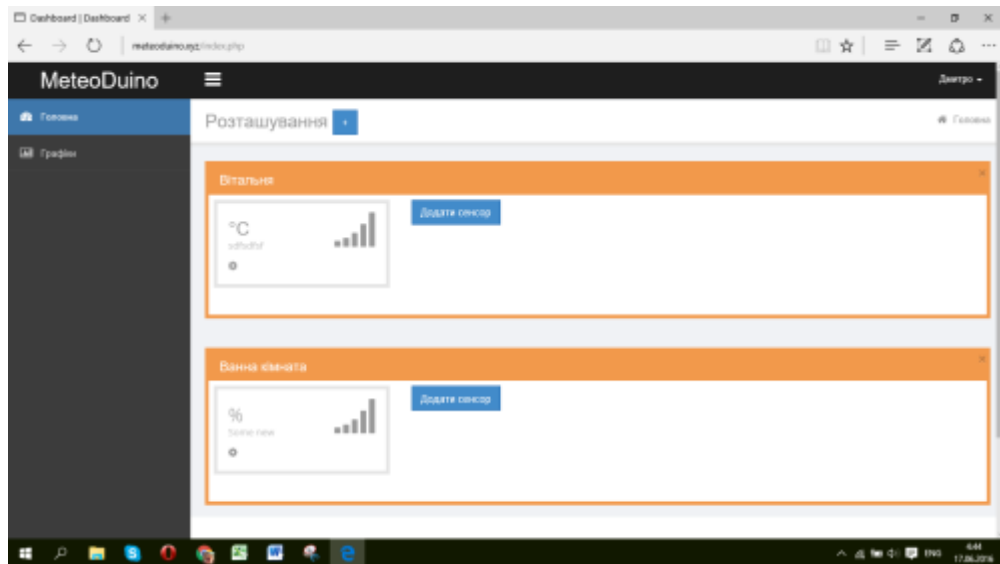


Рисунок 3.7 – Відображення онлайн-платформи у браузері Microsoft Edge

Під час тестування сумісності з різними браузерами було виявлено, що веб-сайт повністю завантажується із всіма структурними елементами в усіх трьох обраних для тестування веб-браузерах.

Після цього було проведено тестування адаптивного дизайну, результати якого представлені на рисунку 3.8.

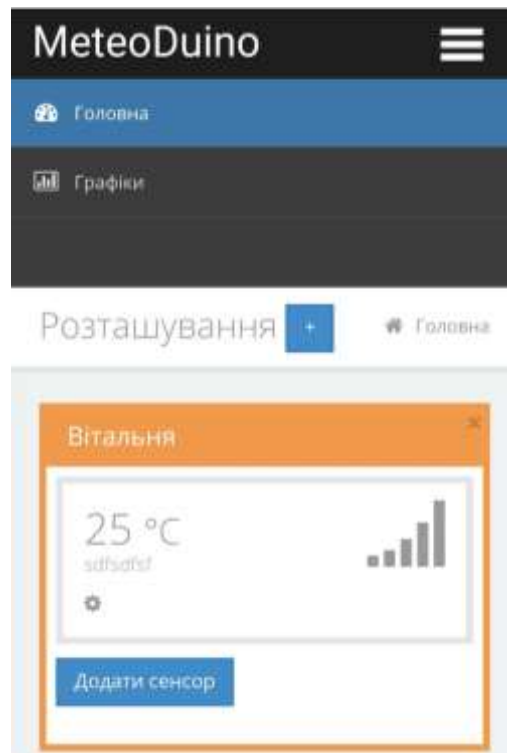


Рисунок 3.8 – Робота адаптивного дизайну онлайн-платформи

Перевірка показала, що адаптивний дизайн функціонує вірно.

3.4.2 Тестування функціоналу онлайн-платформи

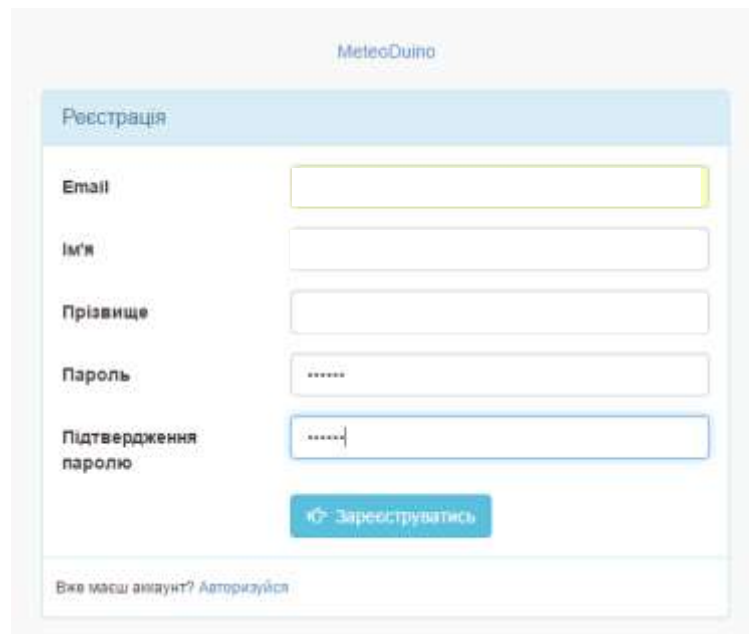
Після проведення перевірки користувацького інтерфейсу онлайн-платформи для відображення моніторингу метеоінформації для підприємства та дому необхідно також пройти тестування її функціоналу. Оскільки сайт є складною системою, тестування повинно охоплювати різні аспекти, а виявлені недоліки мають бути виправлені. Тестування онлайн-платформи включає перевірку його відповідності вимогам та характеристикам, ефективність у різних умовах та з різними навантаженнями, а також оцінку його зручності використання. Залежно від мети тестування перевіряються різні аспекти функціоналу сайту.

Тестування функціоналу онлайн-платформи для відображення моніторингу метеоінформації для підприємства та дому включало наступний тест-план:

- Перевірка реєстрації.
- Перевірка авторизації.
- Навігація між розділами сайту.
- Додавання місць розташування сенсорів.
- Додавання сенсорів збирання даних.
- Перевірка відображення графіків.
- Налаштування особистого профілю.

Після складання цього тест-плану ми приступили до тестування функціоналу сайту.

Для реєстрації на сайті користувачу необхідно натискнути відповідну кнопку, ввести свої реєстраційні дані та натиснути "Зареєструватись", як показано на рисунку 3.9.



The screenshot shows the registration page for MeteoDuino. At the top, the site name 'MeteoDuino' is displayed. Below it is a light blue header with the word 'Рєєстрація' (Registration). The form contains five input fields: 'Email', 'Ім'я' (Name), 'Прізвище' (Surname), 'Пароль' (Password), and 'Підтвердження паролю' (Confirm password). The password fields contain six dots. A blue button with a plus icon and the text 'Зареєструватись' (Register) is positioned below the fields. At the bottom of the form, there is a link: 'Вже маєш акаунт? Авторизуйся' (Already have an account? Log in).

Рисунок 3.9 – Форма реєстрації користувачів

Після завершення реєстрації з'являється повідомлення про успішне її завершення, яке показано на рисунку 3.10.

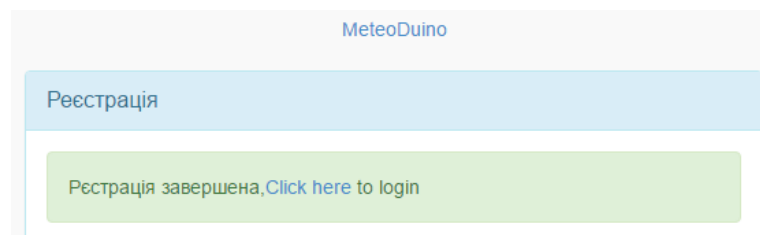


Рисунок 3.10 – Повідомлення про успішне завершення реєстрації

Після завершення реєстрації користувач може авторизуватись на сайті.

Для авторизації на сайті в формі входу потрібно ввести логін та пароль і натиснути кнопку «Увійти», як це відображено на рисунку 3.11.

Рисунок 3.11 – Форма входу на сайт

Для перевірки роботи переходу між функціоналом онлайн-платформи здійснено перехід з головної сторінки сайту до розділу «Мій профіль» та до решти веб-сторінок. Сторінку «Мій профіль» відображено на рисунку 3.12.

Рисунок 3.12 – Сторінка «Мій профіль»

Перехід між всіма розділами сайту відбувається без затримок і коректно: пункти меню відображаються правильно і перенаправляють користувача у відповідні розділи.

Функція додавання розташування давачів інформації розташована на головній сторінці. Для того щоб додати розташування, користувач повинен

натиснути кнопку з плюсом, ввести назву розташування і натиснути кнопку "Зберегти", як показано на рисунку 3.13

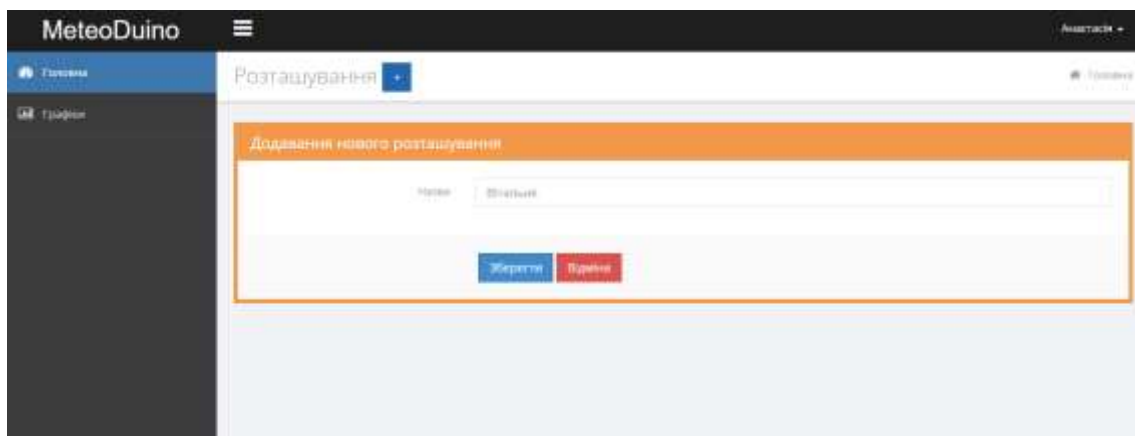


Рисунок 3.13 – Додавання розміщення датчиків даних

Після додавання розміщення виконується додавання датчиків інформації натиснувши на кнопку «Додати сенсор» наведено на рисунку 3.14.

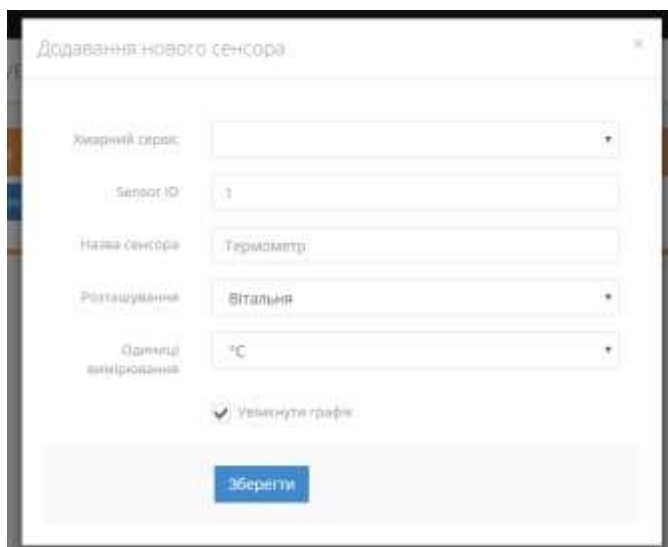


Рисунок 3.14 – Додавання датчиків інформації

Щоб переглянути графіки, потрібно перейти на сторінку «Графіки» обрати необхідний графік та вибрати період часу за який він буде відображатись наведено на рисунку 3.15.

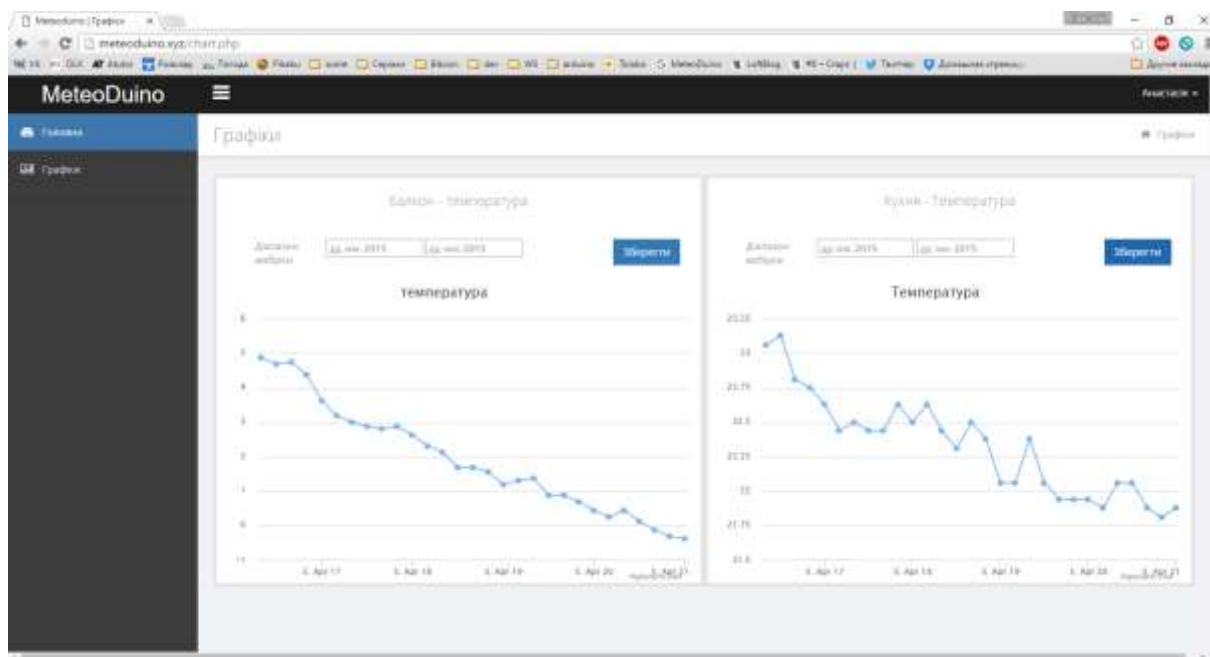


Рисунок 3.15 – Сторінка «Графіки»

Для налаштування особистого профілю потрібно натиснути на кнопку «Мій профіль». На даній сторінці можна редагувати особисту інформацію, змінювати аватар, змінити пароль, налаштувати регіональні параметри, налаштувати експорт даних та здійснити налаштування авторизації хмарного сервісу наведено на рисунку 3.16.

Рисунок 3.16 – Налаштування особистого профілю на сторінці «Мій профіль»

Для виходу з облікового запису потрібно натиснути кнопку «Вийти».

Після виконання тестування користувацького інтерфейсу та перевірки функціоналу онлайн-платформи для відображення моніторингу метеоінформації для підприємства та дому можна вважати тестування завершеним. В процесі тестування не виявлено жодних помилок і підтверджено, що всі необхідні функції платформи працюють належним чином.

3. 5 Висновки до третього розділу

Третій розділ описує процес розгортання та налаштування онлайн-платформи для відображення моніторингу метеоінформації для "розумної" квартири. В розділі наведено перелік етапів розгортання платформи, включаючи підготовку середовища, встановлення необхідного програмного забезпечення, налаштування серверів і забезпечення безперебійної роботи системи.

У розділі детально описано процес розміщення онлайн-платформи на хостинг, що включає вибір хостинг-провайдера, налаштування доменного імені, а також забезпечення відповідних умов для надійного функціонування платформи в Інтернеті.

Даний розділ присвячено структурі бази даних, яка є фундаментом для зберігання та управління метеоінформацією. В цьому розділі розглянуто основні таблиці, зв'язки між ними та методи забезпечення цілісності та безпеки даних.

Описано тестування онлайн-платформи. Тестування користувацького інтерфейсу включає перевірку зручності використання, коректності відображення елементів та загальної естетики. Тестування функціоналу зосереджено на перевірці роботи основних компонентів платформи, включаючи обробку даних, взаємодію з користувачем та відповідність функціоналу заявленим вимогам.

Загалом, третій розділ детально розкриває процес розгортання, налаштування та тестування онлайн-платформи, що забезпечує її готовність до ефективного використання для відображення метеоінформації в "розумних" квартирах.

РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Захист стійкості комп'ютерної техніки від ураження в надзвичайних ситуаціях

Відповідно до Закону України, прийнятого в лютому 1993 року, і Положення про Цивільну оборону, Женевської конвенції від 12 серпня 1949 року і Додаткових Протоколів від 8 червня 1977 року громадяни України мають право на захист свого життя і здоров'я від наслідків катастроф, стихійних лих і можуть вимагати від уряду України, інших органів державної виконавчої влади гарантій щодо забезпечення його реалізації.

Держава як гарант цього права утворює систему органів управління, сил і заходів, способів і дій, об'єднаних і визначених як Цивільна оборона України.

До комплексу дій держави і суб'єктів щодо попередження надзвичайних ситуацій належать відслідковування передумов і вплив можливих подій на їх розвиток, а в разі виникнення надзвичайної ситуації збір, опрацювання, аналіз і передача у встановленому порядку інформації, підготування проектів і прийняття відповідних невідкладних рішень, термінові дії з локалізації і ліквідації надзвичайної ситуації; ослаблення й ліквідація наслідків надзвичайної ситуації; організація компенсації втрат населенню.

Для вирішення цих завдань створена Постійна урядова комісія з питань техногенно-екологічної безпеки і надзвичайних ситуацій.

Для оперативного реагування на надзвичайні ситуації або їх загрозу відповідно до наказу Міністерства надзвичайних ситуацій України діє оперативна мобільна група (ОМГ).

4.2 Надзвичайні ситуації в Україні та їх класифікація

Характерною рисою розвитку земної цивілізації є збільшення небезпеки її загибелі. Сьогодні на нашій планеті посилюється глобальна системна криза, яка проявляється у погіршенні екології, зміні кліматичних умов, збільшенні кількості

і масштабів природних і техногенних катастроф, терористичних актів та інших соціальних і політичних небезпек.

Тільки за останні 20 років від стихійних лих, промислових аварій і катастроф постраждало у всьому світі понад 1 млрд. людей, в т.ч. 5 млн. загинуло, а матеріальний збиток становить трильйони доларів.

За останні роки щороку в Україні стається в середньому 350 надзвичайних ситуацій.

Від надзвичайних ситуацій (НС) щорічно в Україні гине більше 70 тис. осіб, населення і держава зазнають значних матеріальних збитків. Так, наприклад, у 2008 році внаслідок НС техногенного та природного характеру державі було завдано збитків на суму понад 4,7 млрд. грн, що у 5,7 раз перевищує показники 2007 року і майже в 11 разів втрати від НС 2006 -го. При цьому понад 4,6 млрд. грн. складають збитки від НС природного характеру.

Надзвичайні ситуації класифікують за характером походження, ступенем поширення, розміром людських втрат і матеріальних збитків.

Згідно зі змінами, які вносяться в Державний класифікатор надзвичайних ситуацій ДК 019-2001, затверджений наказом Держстандарту України від 19.11.01 р. № 552, уточнені види надзвичайних ситуацій та їх зміст.

Види надзвичайних ситуацій (залежно від характеру походження, що можуть зумовити виникнення НС на території України):

- техногенного характеру;
- природного характеру;
- соціального характеру;
- воєнного характеру.

Надзвичайна ситуація техногенного характеру – це порушення нормальних умов життя і діяльності людей на окремій території чи об'єкті на ній або на водному об'єкті внаслідок транспортної аварії (катастрофи) пожежі, вибуху, аварії з викиданням (загрозою викидання) небезпечних хімічних, радіоактивних та біологічно небезпечних речовин, раптового руйнування споруд; аварії в електроенергетичних системах, системах життєзабезпечення, системах

телекомунікацій, на очисних спорудах, у системах нафтогазового промислового комплексу, гідродинамічних аварій тощо.

Надзвичайна ситуація природного характеру – це порушення нормальних умов життя і діяльності людей на окремій території чи об'єкті на ній або на водному об'єкті, пов'язане з небезпечним геофізичним, геологічним чи гідрологічним явищем, деградацією ґрунтів чи надр, пожежею у природних екологічних системах, зміною стану повітряного басейну, інфекційною захворюваністю та отруєнням людей, інфекційним захворюванням свійських тварин, масовою загибеллю диких тварин, ураженням сільськогосподарських рослин хворобами та шкідниками тощо.

Надзвичайна ситуація соціального характеру – це порушення нормальних умов життя і діяльності людей на окремій території чи об'єкті на ній або на водному об'єкті спричинене протиправними діями терористичного і антиконституційного спрямування, або пов'язане із зникненням (викраденням) зброї та небезпечних речовин, нещасними випадками з людьми тощо.

Надзвичайна ситуація воєнного характеру – це порушення нормальних умов життя і діяльності людей на окремій території чи об'єкті на ній або на водному об'єкті спричинене застосуванням звичайної зброї або зброї масового ураження, під час якого виникають вторинні чинники ураження населення, що визначаються окремими нормативними документами.

Згідно з Порядком класифікації НС техногенного та природного характеру за їх рівнями, затвердженим постановою Кабінету Міністрів України від 24.03.04 № 368 залежно від обсягів заподіяних надзвичайною ситуацією наслідків, кількості постраждалих і загиблих, обсягів технічних і матеріальних ресурсів, необхідних для ліквідації її наслідків, визначають такі рівні надзвичайних ситуацій:

- державний;
- регіональний;
- місцевий;
- об'єктовий.

4.2 Загальні вимоги безпеки з охорони праці для користувачів ПК

Перелік загальних вимог безпеки з охорони праці для користувачів ПК доволі великий. Перед початком роботи з ПК, потрібно пройти навчання з питань безпеки, включаючи інструктаж з охорони праці та пожежної безпеки.

Кожен користувач ПК повинен дотримуватися ряду правил для забезпечення безпеки та легальності використання. Перш за все, необхідно уникати доступу сторонніх осіб до ПК, що запобігає порушенням приватності та безпеки даних. Крім того, важливо використовувати лише ліцензійне програмне забезпечення, щоб уникнути порушень авторських прав та забезпечити надійну роботу системи. Не менш важливо дотримуватися вимог з охорони праці, що включає в себе правильну організацію робочого місця, уникнення перевантажень та забезпечення заходів безпеки при взаємодії з ПК.

Перед початком роботи за ПК для забезпечення безпеки охорони праці слід дотримуватися таких вимог:

- Протерти клавіатуру й екран монітора.
- Відстань від екрану до очей має бути 60-80 сантиметрів.
- Забезпечити, щоб пряме світло не впливало на екран ПК.
- Регулювати освітленість приміщення за допомогою сонцезахисних пристроїв.
- Провести налаштування положення робочого стільця, кут нахилу спинки, та екрану монітора.
- Підготувати робоче місце, переконавшись, що воно чисте та безпечне.
- Переконавшись, що всі пристрої (монітор, клавіатура та інші) правильно підключені до системного блоку.
- Переконавшись, що в елементах ПК відсутні струмопровідні елементи.
- Перевірити стан ізоляції кабелів живлення та їх надійність.
- Переконавшись у належному функціонуванні елементів керування.
- Перевірити надійність заземлення електричного устаткування.
- При виявленні будь-яких несправностей не вмикати ПК і повідомити відповідальну особу [40].

Під час користування ПК важливо дотримуватися основних вимог безпеки:

- Увімкнути ПК вимикачами на корпусах у послідовності: периферійне обладнання, монітор, системний блок.
- Відрегулювати яскравості монітору.
- При роботі з текстовою інформацією, використовувати чорні знаки на світлому фоні.
- Для нейтралізації статичної електрики підвищуйте вологість повітря за допомогою зволожувачів.
- Для зменшення навантаження роботи на ПК, потрібно розподіляти рівномірно, враховуючи її складність.
- Тривалість безперервної роботи за ПК не повинна перевищувати дві години, після чого слід зробити перерву на 15 хвилин.
- Забороняється захарашувати робоче місце сторонніми предметами.
- Забороняється самостійно розбирати та ремонтувати елементи ПК.
- Забороняється класти будь-які предмети на ПК та його елементи.
- Забороняється вимикати апаратуру під час роботи.
- Забороняється переключати з'єднувальні шнури при включеному живленні.
- У разі виявлення запаху горілого, потрібно вимкнути пристрій та звернутися до відповідальної особи.

Після закінчення роботи з ПК потрібно зберегти інформацію, при необхідності файли створити резервну копію даних. Потім слід вимкнути ПК та, за потреби відключити периферійні пристрої від електромережі. Після цього потрібно прибрати робоче місце.

Під час роботи на ПК можуть виникнути небезпечні ситуації, такі як коротке замикання, перевантаження блоку живлення, перегрівання, пожежа або поломка обладнання. У разі таких ситуацій необхідно негайно відключити ПК від електромережі та повідомити відповідальну особу. Забороняється допускати сторонніх осіб у небезпечну зону. Якщо сталася аварія, важливо зберегти стан робочого місця та обладнання та повідомити відповідальній особі для отримання подальших інструкцій та запобігти подібним ситуація в майбутньому. У разі

пожежі потрібно повідомити відповідальну особу, викликати рятувальну службу та, якщо можливо, гасити пожежу за допомогою вогнегасників, але лише після відключення обладнання від електромережі. У разі наявності потерпілих потрібно надати долікарську допомогу та викликати швидку медичну допомогу [41].

4.4 Висновок до четвертого розділу

В даному розділу кваліфікаційної роботи описано захист стійкості комп'ютерної техніки від ураження в надзвичайних ситуаціях. Описано надзвичайні ситуації в Україні та їх класифікація.

Проаналізовано вимоги безпеки охорони праці для користувачів перед початком роботи, під час виконання роботи та після завершення виконання роботи за ПК. Дотримання цих вимог дозволяє запобігти виникненню різноманітних небезпечних ситуацій.

ВИСНОВКИ

В кваліфікаційній роботі освітнього рівня “Бакалавр” проведено розробку онлайн-платформи для відображення моніторингу метеоінформації для «розумної» квартири.

Перший розділ досліджує предметну область і визначає завдання для розробки онлайн-платформи для відображення моніторингу метеоінформації для «розумної» квартири. Аналіз предметної області дозволив визначити ключові аспекти, пов'язані з вимогами до платформи та її варіантами використання. Вимоги включають не тільки точність відображення метеоданих, але й зручність інтерфейсу для кінцевих користувачів.

Проведено аналіз різних варіантів вирішення поставленої задачі, включаючи оцінку методів і вибір оптимального алгоритму. Оптимальний алгоритм вибрано з урахуванням його ефективності та можливості інтеграції з існуючими системами "розумної" квартири. Також було обрано середовище розробки, яке найкраще підходить для реалізації цієї платформи, зокрема з урахуванням масштабованості та безпеки.

Життєвий цикл онлайн-платформи обґрунтовано з урахуванням специфіки предметної області. Він включає етапи від проектування до впровадження, з фокусом на забезпечення надійності, швидкості та зручності користування для кінцевих користувачів. Загальний висновок полягає в тому, що ретельний аналіз і вибір оптимальних рішень на початковому етапі сприятимуть успішному розвитку та ефективній експлуатації платформи для "розумних" квартир.

Другий розділ присвячено проектуванню онлайн-платформи для відображення моніторингу метеоінформації для «розумної» квартири. Моделювання архітектури включало розгляд різних можливостей структурування системи з урахуванням потреб високої доступності та швидкодії. Проектування структури детально визначило складові частини системи та їх взаємозв'язки для забезпечення ефективної інтеграції та сумісності з іншими технологіями "розумних" квартир.

Проектування поведінки платформи зосереджувалося на розробці логічного ядра системи, яке забезпечує надійність та точність збору, обробки та відображення метеоданих. Цей аспект був критично важливим для забезпечення коректного функціонування платформи в усіх умовах експлуатації.

Проектування інтерфейсу врахувало не лише естетичні аспекти, але й зручність використання для кінцевих користувачів. Інтерфейс платформи був розроблений таким чином, щоб надавати інтуїтивно зрозумілу інформацію і забезпечувати зручний доступ до всіх функцій системи.

У цілому, другий розділ визначив ключові аспекти проектування онлайн-платформи для моніторингу метеоінформації для «розумних» квартир, забезпечивши фундаментальні засади для успішного подальшого розвитку та реалізації проекту.

Третій розділ описує процес розгортання та налаштування онлайн-платформи для відображення моніторингу метеоінформації для "розумної" квартири. У розділі наведено перелік етапів розгортання платформи, включаючи підготовку середовища, встановлення необхідного програмного забезпечення, налаштування серверів і забезпечення безперебійної роботи системи.

Детально описано процес розміщення онлайн-платформи на хостинг, який включає вибір хостинг-провайдера, налаштування доменного імені, а також забезпечення відповідних умов для надійного функціонування платформи в Інтернеті.

Даний розділ присвячено структурі бази даних, яка є фундаментом для зберігання та управління метеоінформацією. В цьому розділі розглянуто основні таблиці, зв'язки між ними та методи забезпечення цілісності та безпеки даних.

Описано тестування онлайн-платформи. Тестування користувачького інтерфейсу включає перевірку зручності використання, коректності відображення елементів та загальної естетики. Тестування функціоналу зосереджено на перевірці роботи основних компонентів платформи, включаючи обробку даних, взаємодію з користувачем та відповідність функціоналу заявленим вимогам.

Загалом, третій розділ детально розкриває процес розгортання, налаштування та тестування онлайн-платформи, що забезпечує її готовність до ефективного використання для відображення метеоінформації в "розумних" квартирах.

ПЕРЕЛІК ДЖЕРЕЛ

1. Система стандартів з інформації, бібліотечної та видавничої справи. Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання : ДСТУ ГОСТ 7.1:2006. – [Чинний від 01.01.2007]. – Режим доступу : URL : http://www.mmt.zmaro.edu.ua/mmt_ukr/bibl_standart.html. – Дата доступу : 11.04.2016. – Загол. з екрану.
2. Методичні вказівки до виконання дипломної роботи для студентів усіх форм навчання напряму 6.050101 : метод. посіб. для студ. ВНЗ / Боднарчук І.О., Дуда О.М., Маєвський О.В. [та ін.] ; за ред. Є.Гриценка. – Тернопіль : Видавництво ТНТУ ім. І.Пулюя, 2014. – 119 с.
3. Гетьман Оксана Олександрівна. Економіка підприємства : навч. посіб. для вузів / О.О. Гетьман, В. М. Шаповал. – Вид 2-ге. стер. – К.: Центр учбової літератури, 2014. - 488 с. – ISBN 978-6-11-010005-2
4. Голінько Василь Іванович. Основи охорони праці : підручник [Електронний ресурс] / М-во освіти і науки України; Нац. гірн. ун-т. – 2-ге вид. – Д.: НГУ, 2014. – 271 с. – Режим доступу : URL : <http://ir.nmu.org.ua/bitstream/handle/123456789/108579/CD498.pdf?sequence=1&isAllowed=y>. – Дата доступу : 21.04.2016 – Загол. з екрану.
5. Вікіпедія [Електронний ресурс] : Інформаційна система. – Режим доступу : URL : https://uk.wikipedia.org/wiki/Інформаційна_система. – Дата доступу: 18.04.2016 – Загол. з екрану.
6. Вікіпедія [Електронний ресурс] : Життєвий цикл. – Режим доступу : URL : https://uk.wikipedia.org/wiki/Життєвий_цикл. – Дата доступу : 20.04.2016 – Загол. з екрану.
7. Вікіпедія [Електронний ресурс] : Тестування програмного забезпечення. – Режим доступу : URL : https://uk.wikipedia.org/wiki/Тестування_програмного_забезпечення. – Дата доступу : 20.04.2016 – Загол. з екрану.
8. Вікіпедія [Електронний ресурс] : jQuery. – Режим доступу : URL : <https://uk.wikipedia.org/wiki/JQuery> – Дата доступу: 15.04.2016. – Загол. з екрану.

9. Вікіпедія [Електронний ресурс] : CSS. – Режим доступу : URL : <https://uk.wikipedia.org/wiki/CSS> – Дата доступу: 15.04.2016. – Загол. з екрану.
10. Вікіпедія [Електронний ресурс] : HTML. – Режим доступу : URL : <https://uk.wikipedia.org/wiki/HTML> – Дата доступу: 15.04.2016. – Загол. з екрану.
11. Вікіпедія [Електронний ресурс] : PHP. – Режим доступу : URL : <https://uk.wikipedia.org/wiki/PHP> – Дата доступу: 18.04.2016. – Загол. з екрану.
12. Веб-технології та веб-дизайн [Електронний ресурс] – Режим доступу: <http://victoria.lviv.ua/html/wp/index.html> – Дата доступу: 30.05.2016. – Загол. с екрана.
13. Моделі потоків даних (DFD-моделі): призначення, місце застосування в системному аналізі, правила побудови, приклади [Електронний ресурс] – Режим доступу: <http://victoria.lviv.ua/html/wp/index.html> – Дата доступу: 15.05.2016.
14. Навчальні матеріали онлайн [Електронний ресурс] : Дія електричного струму на організм людини. Режим доступу : URL : http://pidruchniki.com/14990528/bzhd/bezpeka_pratsi_pid_chas_roboti_obchislyuval_ponu_keruyuchoyu_tehnikoyu. – Дата доступу : 21.04.2016 – Загол. з екрану.
15. V. Kozlovskiy, Y. Balanyuk, H. Martyniuk, O. Nazarevych, L. Scherbak and G. Shymchuk, «Information Technology for Estimating City Gas Consumption During the Year,» 2022 International Conference on Smart Information Systems and Technologies (SIST), Nur-Sultan, Kazakhstan, 2022, pp. 1-4, doi: 10.1109/SIST54437.2022.9945786.
16. Approach to gas consumption process forecasting on the basis of a mathematical model in the form of a random cyclic process / Serhii Lupenko, Iaroslav Lytvynenko, Oleg Nazarevych, Grigorii Shymchuk, Volodymyr Hotovych // ICAAEIT 2021, 15-17 December 2021. – Tern. : TNTU, Zhytomyr «Publishing house „Book-Druk“» LLC, 2021. – P. 213–219. – (Mathematical modeling in power engineering and information technologies).
17. Lytvynenko, S. Lupenko, O. Nazarevych, G. Shymchuk and V. Hotovych, «Mathematical model of gas consumption process in the form of cyclic random process,» 2021 IEEE 16th International Conference on Computer Sciences and

Information Technologies (CSIT), LVIV, Ukraine, 2021, pp. 232-235, doi: 10.1109/CSIT52700.2021.9648621.

18. Additive mathematical model of gas consumption process / Iaroslav Lytvynenko, Serhii Lupenko, Oleh Nazarevych, Hryhorii Shymchuk, Volodymyr Hotovych // Scientific Journal of TNTU. – Tern. : TNTU, 2021. – Vol 104. – No 4. – P. 87–97.

19. O. Nazarevych, Y. Leshchyshyn, S. Lupenko, V. Hotovych, G. Shymchuk and N. Shablii, «Method of Gas Consumption Change-point Detection Based on Seasonally Multicomponent Model,» 2020 10th International Conference on Advanced Computer Information Technologies (ACIT), Deggendorf, Germany, 2020, pp. 152-155, doi: 10.1109/ACIT49673.2020.9208924.

20. Y. Leshchyshyn, L. Scherbak, O. Nazarevych, V. Gotovych, P. Tymkiv and G. Shymchuk, «Multicomponent Model of the Heart Rate Variability Change-point,» 2019 IEEE XVth International Conference on the Perspective Technologies and Methods in MEMS Design (MEMSTECH), Polyana, Ukraine, 2019, pp. 110-113, doi: 10.1109/MEMSTECH.2019.8817379.

21. Панченко С. Електробезпека / С. Панченко, О. Акімов, М. Бабаєв, В. Блиндюк, В. Панченко, О. Супрун, Д. Сушко. – УкрДУЗТ, 2018. – 295 с. – ISBN 978-617-654-085-4.

22. Запорожець О. Основи охорони праці / О. Запорожець, О. Протоєрейський, Г. Франчук, І. Боровик. – Цент учбової літератури, 2021. – 264 с. – ISBN 9786176734239.

23. Удар електричним струмом: перша допомога, наслідки після ураження [Електронний ресурс]. Режим доступу до ресурсу: <https://www.uzhnu.edu.ua/uk/news/strum.htm> (дата звернення: 13.06.2024).

24. Інструкція з охорони праці при роботі на персональному комп'ютері [Електронний ресурс]. Режим доступу до ресурсу: <https://pro-op.com.ua/article/485-nstruktsya-z-ohoroni-prats-pri-robot-na-personalnomu-kompyuter> (дата звернення: 14.06.2024).

25. Катренко А. Охорона праці в галузі комп'ютерингу / А. Катренко, Л. Катренко. – Манголія 2006, 2024. – 544 с. – ISBN 978-617-574-049-1.

ДОДАТКИ

Лістинг файлу charts.js

```

$(document).ready(function() {
    $(".loadingimg").hide();
    // Split timestamp into [ Y, M, D, h, m, s ]
    var max_date = $('[type="date"]').prop('max').split(/[-
]//),
        min_date = $('[type="date"]').prop('min').split(/[-
]//),
    // Apply each element to the Date function
    max_date_js = new Date(max_date[0], max_date[1]-1,
max_date[2]),
    min_date_js = new Date(min_date[0], min_date[1]-1,
min_date[2]);
    $('[type="date"]').prop('max', function(){
        return max_date_js.toJSON().split('T')[0];
    });
    $('[type="date"]').prop('min', function(){
        return min_date_js.toJSON().split('T')[0];
    });
    $(".date-form").submit(function (e) {
        e.preventDefault();
        var form_data = $(this).serialize(),
            form_url = $(this).attr("action"),
            form_method
=
$(this).attr("method").toUpperCase(),
            user_id = $(this).data("id"),
            measurement = $(this).data("measurement"),
            cloud_sensor_id
=
"field"+$(this).data("cloud_sensor_id"),
            sensor_name
=
$(this).parent().find(".sensor_name").html(),
            sensor_name_translit
=
$(this).parent().find(".flot-chart").find("div").attr("id");
        console.log(form_data);
        $(".loadingimg").show();
        $.ajax({
            url: form_url,
            type: form_method,
            data:
form_data+"&user_id="+user_id+"&field="+cloud_sensor_id,
            cache: false,
            success: function (returnhtml) {
                $(".loadingimg").hide();
                var chart,
                    options = {
                chart: {
                    renderTo: sensor_name_translit,
                    defaultSeriesType: 'line',
                    marginRight: 0,
                    marginBottom: 30
                },
                title: {
                    text: sensor_name,

```

```

        x: -20 //center
    },
    xAxis: {
        type: 'datetime',
        labels: {
            formatter: function () {
                return Highcharts.dateFormat('%e. %b
%H', this.value);
            },
            dateTimeLabelFormats: {
                minute: '%H:%M',
                hour: '%H:%M',
                day: '%e. %b',
                week: '%e. %b',
                month: '%b \'%y',
                year: '%Y'
            }
        }
    },
    yAxis: {
        title: null,
        plotLines: [{
            value: 0,
            width: 1,
            color: '#808080'
        }]
    },
    tooltip: {
        formatter: function() {
            return
Highcharts.dateFormat('%e. %b %Y %H : %M', this.x) + ': было <b>'+
this.y + " "+ measurement+' </b>';
        }
    },
    legend: {
        enabled: false
    },
    series: [{
        name: sensor_name
    }]
}
// Load data asynchronously using jQuery. On
success, add the data
// to the options and initiate the chart.
// This data is obtained by exporting a GA
custom report to TSV.
// http://api.jquery.com/jquery.get/
console.log(returnhtml);
var resultb = jQuery.parseJSON( returnhtml ),
    traffic = [];
jQuery.each(resultb, function(key,value) {
    date
    =
Date.parse(resultb[key].created_at);
    traffic.push([
        date,
parseFloat(eval("resultb["+key+"]."cloud_sensor_id"))

```

```
        ]);
    });
    options.series[0].data = traffic;
    chart = new Highcharts.Chart(options);
    },
    error: function(xhr, str){
    alert('Виникла помилка: ' + xhr.responseCode);
    $(".loadingimg").hide();
    }
    });
});
});
```


Лістинг файлу location.js

```
$(document).ready(function (e) {
    $('#location_setting').hide();
    $('#loadingimg').hide();
    $('#loadingimg_sadd').hide();
    $('#loadingimg_sedit').hide();
    $('#add_location').click(function() {
        $('#location_setting').show();
    });
    $('#location_cancel').click(function() {
        $('#location_setting').fadeOut("slow", function() {
            $(this).hide();
        });
    });
});

function loadXMLDoc (dname) {
    var xmlDoc;
    try {
        var xmlhttp = new XMLHttpRequest();
        xmlhttp.open('GET', dname, false);
        xmlhttp.setRequestHeader('Content-Type', 'text/xml');
        xmlhttp.send('');
        xmlDoc = xmlhttp.responseXML;
    } catch (e) {
        try {
            xmlDoc = new ActiveXObject("Microsoft.XMLDOM");
        } catch (e) {
            console.error(e.message);
        }
    }
    return xmlDoc;
}

var locations = $('#div#panel-heading'),
    location_names = [];
var user_id = $('#user_id').val();
//Locations array
$.each( locations, function( i, el ) {
```

```

        location_names.push($(locations[ i ]).html());
    });
    console.log(location_names);
$.ajax({
url: "sensor_add.php",
type: "POST",
data: {"locations": location_names,"user_id": user_id },
success: function (returnhtml)
{
    var sensors = $.parseHTML(returnhtml);
    console.log(sensors);
    $.each( sensors, function( i, val ) {
        var loc_page = $(locations[i]).html(), //розташування
користувача на сторінці
        loc_sensors =
$(sensors[i]).find("#sensor_location").html(); //розташування
користувача в шаблоні сенсора
        if (typeof $(sensors[i]).find("#channel_id").html()
!= "undefined") {
            var channel_id =
$(sensors[i]).find("#channel_id").html(); //ідентифікатор каналу
thingspeak
        }
        var sensor_id =
$(sensors[i]).find("#sensor_id").html(), // номер field сенсора в
аккаунті thingspeak
        sensor_checked = [];
        sensor_checked.push($(sensors[i]).find("#sensor_id").html()), //
збереження всіх field сенсорів з аккаунту користувача на сайті
        xml =
String("https://api.thingspeak.com/channels/"+channel_id+"/feeds/l
ast.xml"); //останні покази за вказаним аккаунтом
        if (typeof $(sensors[i]).find("#sensor_id").html()
!= "undefined") {
            console.log(sensor_checked[0]);
            $.get( xml, function( data ) {
                var xmlDoc = data,

```

```

        xml = $( xmlDoc ), // парсинг xml з останніми
показами (не працює в опері і ie)
        field_id =
parseInt($(xml[0]).find("field"+sensor_checked[0]).html()); //
значенням сенсорів доданих на сайт з xml запиту від thingspeak
        // console.log(field_id);

$(sensors[i]).find("#sensor_value").html(field_id); //додавання
значень до відповідних сенсорів
    });
}
    $("."+loc_sensors).prepend(sensors[i]); // додавання
відповідних сенсорів у відповідні розташування
    $(sensors[i]).find("#sensor_edit").on("click", this,
function(){
        var onclick_id =
$(this).parent().parent().find("#sensor_id").html(); //отримання
sensor_id

        $.ajax({
            url: "sensor_edit.php",
            type: "POST",
            data: {"onclick_id": onclick_id},
            success: function (returnhtml) {
                // console.log(onclick_id);
                // console.log(returnhtml);
                var result = JSON.parse(returnhtml);
                /* {"sensor_id":"15",
                "user_id":"3",
                "sensor_name":"sdfsdf",

"location": "\u041a\u0443\u0445\u044d\u043d\u044f",
                "measurement": "C",
                "cloud_service": "ThingSpeak",
                "cloud_sensor_id": "2",
                "enable_chart": "1"} */
            }
        });
    });
    $(".cloud_sensor_id").val(result.cloud_sensor_id);

```

```

$(".sensor_name").val(result.sensor_name);
        $(".user_id").val(result.user_id);
        $(".sensor_id").val(result.sensor_id);
        console.log(result.sensor_id);
        $(".cloud_service
option').each(function() {
            if($(".html() ==
result.cloud_service) {

$(".this).attr("selected","selected");
                }
            });
            $(".location option').each(function() {
                if($(".html() == result.location)
{

$(".this).attr("selected","selected");
                }
            });
            $(".measurement option').each(function()
{
                if($(".html() ==
result.measurement) {

$(".this).attr("selected","selected");
                }
            });
            if(result.enable_chart == 1)
{$(".enable_chart").attr('checked', true);}
            else if(result.enable_chart == 0) {
$(".enable_chart").attr('checked', false);}
                //
$(".cloud_sensor_id").val(result.cloud_sensor_id);
                //
$(".cloud_sensor_id").val(result.cloud_sensor_id);
            },

```

```

        error: function(xhr, str){
            console.log('Виникла помилка: ' +
xhr.responseCode);
        }
    });
    //
$("#query_report").append(returnhtml).show().fadeOut(3500);
},
error: function(xhr, str){
    console.log('Виникла помилка: ' + xhr.responseCode);
}
});
//Видалення сенсора
$(".sensor_delete").click(function() {
    $("#loadingimg").show();
    var sensor_id_del =
$(this).parent().parent().parent().parent().find(".sensor_id").val
();
    $.ajax({
        url: "sensor_edit_main.php",
        type: "POST",
        data: {"delete": sensor_id_del},
        success: function (returnhtml) {
            $("#loadingimg").hide();
            // $("#sensor_edit_result"
).append(returnhtml).show();
            location.reload();
        },
        error: function(xhr, str){
            alert('Виникла помилка: ' + xhr.responseCode);
            $("#loadingimg").hide();
        }
    });
});
//Видалення розташування

```

```

        $(".delete_location").click(function() {
            var                location_id                =
$(this).parent().find("#location_id").html();
            // console.log(location_id);
            $.ajax({
                url: "location_proc.php",
                type: "POST",
                data: {"location_id": location_id},
                success:                function                (returnhtml)                {
$( "#query_report" ).append(returnhtml).show().fadeOut(2500);
                    // alert(returnhtml);
                    // location.reload();
                },
                error:                function(xhr, str){
                    alert('Виникла помилка: ' + xhr.responseCode);
                    $("#loadingimg").hide();
                }
            });
        });
//Відправка форми редагування сенсора
$("#sensor_edit").submit(function (e) {
    e.preventDefault();
    var form_data = $(this).serialize();
    var form_url = $(this).attr("action");
    var                form_method                =
$(this).attr("method").toUpperCase();
    $("#loadingimg_sedit").show();
    $.ajax({
        url: form_url,
        type: form_method,
        data: form_data,
        cache: false,
        success:                function                (returnhtml)                {
$("#loadingimg_sedit").hide();
$("#sensor_edit_result").append(returnhtml).show().fadeOut(3500);
                location.reload();
            },

```

```

        error: function(xhr, str){
            alert('Виникла помилка: ' + xhr.responseCode);
            $("#loadingimg_sedit").hide();
        }
    });
});
//Відправка форми додавання сенсора
$("#sensor").submit(function (e) {
    e.preventDefault();
    var form_data = $(this).serialize();
    var form_url = $(this).attr("action");
    var form_method = $(this).attr("method").toUpperCase();
    $("#loadingimg_sadd").show();
    $.ajax({
        url: form_url,
        type: form_method,
        data: form_data,
        cache: false,
        success: function (returnhtml) {
            $("#loadingimg_sadd").hide();

            $("#sensor_result").append(returnhtml).show();
            // location.reload();
        },
        error: function(xhr, str){
            $("#loadingimg_sadd").hide();
            alert('Виникла помилка: ' + xhr.responseCode);
        }
    });
});
//Відправка форми додавання розташування
$("#location_form").submit(function (e) {
    e.preventDefault();
    var form_data = $(this).serialize();
    var form_url = $(this).attr("action");

```

```
var form_method =
$(this).attr("method").toUpperCase();
$("#loadingimg").show();
$.ajax({
    url: form_url,
    type: form_method,
    data: form_data,
    cache: false,
    success: function (returnhtml) {
$("#loadingimg").hide();
$("#query_report").append(returnhtml).show().fadeOut(3500);
        location.reload();
    },
    error: function(xhr, str){
alert('Виникла помилка: ' + xhr.responseCode);
$("#loadingimg").hide();
    }
});
});
});
```


Лістинг файлу index.php

```
<?php
    /** Important do not change ***/
    session_start();
    if(!isset($_SESSION['db_login'])){
        header('Location: auth/register.php');
        exit();
    }
    /** Important END ***/
require_once('auth/db_conn.php');
require_once('auth/translit.php');
$id = $_SESSION['db_login']['id'];
?>
<!DOCTYPE html>
<html>
<head>
    <title>Dashboard | Dashboard</title>
    <meta charset='utf-8'>
    <meta
        http-equiv='X-UA-Compatible'
content='IE=edge,chrome=1'>
    <meta
        name='viewport'
        content='width=device-width,
initial-scale=1.0'>
    <link
        rel='shortcut
        icon'
href='images/icons/favicon.ico'>
    <link
        rel='apple-touch-icon'
href='images/icons/favicon.png'>
    <link
        rel='apple-touch-icon'
        sizes='72x72'
href='images/icons/favicon-72x72.png'>
    <link
        rel='apple-touch-icon'
        sizes='114x114'
href='images/icons/favicon-114x114.png'>
    <!--Loading bootstrap css-->
    <link
        type='text/css'
        rel='stylesheet'
href='http://fonts.googleapis.com/css?family=Open+Sans:400italic,4
00,300,700'>
    <link
        type='text/css'
        rel='stylesheet'
href='http://fonts.googleapis.com/css?family=Oswald:400,700,300'>
```

```

        <link                type='text/css'                rel='stylesheet'
href='styles/jquery-ui-1.10.4.custom.min.css'>
        <link type='text/css' rel='stylesheet' href='styles/font-
awesome.min.css'>
        <link                type='text/css'                rel='stylesheet'
href='styles/bootstrap.min.css'>
        <link                type='text/css'                rel='stylesheet'
href='styles/all.css'>
        <link                type='text/css'                rel='stylesheet'
href='styles/main.css'>
        <link                type='text/css'                rel='stylesheet'
href='styles/style-responsive.css'>
        <link                type='text/css'                rel='stylesheet'
href='styles/pace.css'>
        <!--                <link type='text/css' rel='stylesheet'
href='styles/jquery.news-ticker.css'> -->
    </head>
    <body>
<div>
<!--BEGIN                BACK                TO                TOP-->
<a id='totop' href='#'><i class='fa fa-angle-up'></i></a>                <!--
-END                BACK                TO                TOP-->
<?php                include('template/topbar_menu.tpl');                ?>
<div                id='wrapper'>
<?php                include('template/sidebar_menu.tpl');                ?>
<!--BEGIN                PAGE                WRAPPER-->
<div                id='page-wrapper'>
<!--BEGIN                TITLE                &                BREADCRUMB                PAGE-->
<div                id='title-breadcrumb-option-demo'                class='page-title-
breadcrumb'>
<div                class='page-header                pull-left'>
<div                class='page-title'>
Розташування
button                type='button'                id='add_location'                class='btn                btn-
primary'>+</button>
</div>
</div>

```

```

        <ol class='breadcrumb page-breadcrumb pull-right'>
<li>
        <i class='fa fa-home'></i>&nbsp;
<a href="http://<?php echo
$_SERVER[HTTP_HOST].$_SERVER[REQUEST_URI]; ?>">Головна</a></li>
</ol>
<div class='clearfix'>
</div>
</div>
<!--END TITLE & BREADCRUMB PAGE-->
<!--BEGIN CONTENT-->
<div class='page-content'>
<div class='col-lg-12' id='query_report'>
</div>
<!-- Create location start -->
<div class='row mbl' id='location_setting'>
<div class='col-lg-12'>
<div class='panel panel-yellow'> <div
class='panel-heading'>
Додавання нового розташування</div>
        <div class='panel-body pan'><form action='location_proc.php'
method="POST" id='location_form' class='form-horizontal'>
<div class='form-body pal'> <div class='form-group'>
<label for='inputName' class='col-md-3 control-label'>
        Назва
        </label> <div
class='col-md-9'> <div
class='input-icon right'>
<input id='inputName' name='location_name' type='text'
placeholder='Вітальня' class='form-control'>
<input name='id' value='<?php echo $id; ?>' type='hidden'>
</div> </div>
</div>
</div>
<div class='form-actions pal'>
<div class='form-group mbl'>
<div class='col-md-offset-3 col-md-6'>

```

```

<label>
<img      id='loadingimg'      style='margin:      0      auto;'
src='http://dev.cloudcell.co.uk/bin/loading.gif' />
</label>
<button
type='submit'      class='btn      btn-primary'>
Зберігти</button>
<button
type='button'      id='location_cancel'      class='btn      btn-danger'>
Відміна</button>
</form>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<!--      Create      location      end      -->
<?php
    $result = $db->query("SELECT location_name, id FROM locations
WHERE user_id = $id");
    $myrow = mysqli_fetch_array ($result);
    if (!empty($myrow['location_name'])) {
    do {
        $location_translit = rus2translit($myrow['location_name']);
        printf      ("<div      class='row      mbl'>
<div      class='col-lg-12'>
<div class='panel panel-yellow'>
            <button type='button' data-dismiss='alert' aria-hidden='true'
class='close      delete_location'>x</button>
<div      class='panel-heading'      id='panel-heading'>%s</div>
<span      id='location_id'      style='display:      none'>%s</span>
<div id='sum_box'      class='row      mbl      %s'      style='padding:      7px;'>
<div class='col-sm-6 col-md-1'>
            <button
type='button'      id='%s'      data-toggle='modal'      data-target='#modal-
config'      class='btn      btn-primary'>Додати      сенсор</button>
</div>
</div>
</div>
</div>
</div>

```

```

        </div>",          $myrow['location_name'],          $myrow['id'],
$location_translit, $location_translit);
    }
    while ($myrow = mysqli_fetch_array ($result));
    };
    ?>
<div                                                    class='row'>
</div>
</div>
</div>
<!--END                                                    CONTENT-->
<!--BEGIN                                                    FOOTER-->
<div                                                    id='footer'>
<div class='copyright'>                                </div>
<!--END                                                    FOOTER-->
</div>
<!--END PAGE WRAPPER-->
    </div>
</div>
<!--BEGIN            MODAL            CONFIG            PORTLET-->
<div            id="modal-config"            class="modal            fade">
<div            class="modal-dialog">
<div            class="modal-content">
<div            class="modal-header">
<button    type="button"    data-dismiss="modal"    aria-hidden="true"
class="close">
&times;</button>
<h4            class="modal-title">
Додавання            НОВОГО            сенсора</h4>
</div>
<div            class="modal-body">
<div            id="sensor_result"></div>
<form action="sensor.php" id="sensor" method="POST" class="form-
horizontal">
<input    name='id'    id="user_id"    value='<?php    echo    $id;    ?>'
type='hidden'>
<div            class="form-body            pal">

```

```

<div class="form-group">
<label class="col-md-3 control-label">
Хмарний </label>
<div class="col-md-9">
<div class="input-icon right">
<select class="form-control" name="cloud_service">
<?php
$location_select = $db->query(
"SELECT cloud_service, channel_id FROM cloud_settings WHERE user_id
= $id");
$select_row = mysqli_fetch_array($location_select);
do {
printf("<option>%s</option>",
$select_row['cloud_service']);
}
while ($select_row = mysqli_fetch_array($location_select));
?>
</select>
</div>
</div>
</div>
<div class="form-group">
<label class="col-md-3 control-label">
Sensor ID</label>
<div class="col-md-9">
<div class="input-icon right">
<input name="cloud_sensor_id" type="text" placeholder="1"
class="form-control" data-toggle="tooltip" data-placement="right"
data-original-title="Homep field в вашому аккаунті ThingSpeak"
/></div>
</div>
</div>
<div class="form-group">
<label class="col-md-3 control-label">
Назва сенсора</label>
<div class="col-md-9">
<div class="input-icon right">
<input name="sensor_name" type="text" placeholder="" class="form-
control"></div>
</div>
</div>
<div class="form-group">

```



```

</div>
</div>
</div>
<div class="form-actions" style="text-align: right;">
<div class="form-group" style="text-align: right;">
<div class="col-md-offset-3 col-md-6">
<label> <img alt="loading.gif" style="margin: 0 auto; width: 100px; height: 20px; vertical-align: middle;"/>
id='loadingimg_sadd' style='margin: 0 auto;'
src='http://dev.cloudcell.co.uk/bin/loading.gif' />
</label> <button
type="submit" class="btn btn-primary" style="float: right;">
Зберегти</button>
</div>
</div>
</div>
</form>
</div>
</div>
</div>
</div>
</div>
</div>
<!--END MODAL CONFIG PORTLET-->
<!--BEGIN MODAL CONFIG PORTLET-->
<div id="modal-edit" class="modal fade" style="display: none;">
<div class="modal-dialog">
<div class="modal-content">
<div class="modal-header">
<button type="button" data-dismiss="modal" aria-hidden="true"
class="close">
&times;</button>
<h4 class="modal-title">
Редагування даних сенсора</h4>
</div>
<div class="modal-body">
<div id="sensor_edit_result">
<form action="sensor_edit_main.php" id="sensor_edit"
method="POST" class="form-horizontal">
<input name='user_id' class="user_id" type='hidden'>

```



```

        <input name='sensor_id' class="sensor_id" type='hidden'>
        <div class="form-body pal">
        <div class="form-group">
            <label                class="col-md-3                control-label">
Хмарний                                                                    cepvic</label>
        <div                                                                    class="col-md-9">
        <div                class="input-icon                right">
        <select class="form-control cloud_service" name="cloud_service">
        <?php
        $location_select                =                $db->query(
        "SELECT cloud_service, channel_id FROM cloud_settings WHERE user_id
        = $id");
            $select_row                =                mysqli_fetch_array($location_select);
            do {                printf("<option>%s</option>",
        $select_row['cloud_service']);
            }
        while ($select_row                =                mysqli_fetch_array($location_select));
        ?>
        </select>                                                                    </div>
        </div>
        </div>
        <div                                                                    class="form-group">
        <label                class="col-md-3                control-label">
        Sensor ID</label>                                                                    <div
        class="col-md-9">                                                                    <div
        class="input-icon                right">
        <input                name="cloud_sensor_id" type="text" class="form-control
        cloud_sensor_id" data-toggle="tooltip" data-placement="right" data-
        original-title="Номер field в вашому аккаунті ThingSpeak" /></div>
        </div>
        </div>
        <div                                                                    class="form-group">
        <label                class="col-md-3                control-label">
        Назва сенсора</label>                                                                    <div
        class="col-md-9">                                                                    <div
        class="input-icon                right">
        <input name="sensor_name" type="text" placeholder="" class="form-

```

```

control sensor_name"></div>
</div>
</div>
<div class="form-group">
<label class="col-md-3 control-label">
Розташування</label> <div
class="col-md-9"> <div
class="input-icon right">
<select class="form-control location" name="location">
<?php $location_select
= $db->query(
"SELECT location_name FROM locations WHERE user_id = $id");
$select_row = mysqli_fetch_array($location_select);
do {
printf("<option>%s</option>", $select_row['location_name']);
}
while ($select_row = mysqli_fetch_array($location_select));
?></select> </div>
</div>
</div>
<div class="form-group">
<label class="col-md-3 control-label">
Одиниці вимірювання</label>
<div class="col-md-9">
<div class="input-icon right">
<select class="form-control measurement" name="measurement">
<option>&degC</option>
<option>%</option>
<option>F</option>
<option>мм.рт.ст.</option>
<option>немає</option>
</select> </div>
</div>
</div>
<div class="form-group mbn">
<div class="col-md-offset-3 col-md-6">
<div class="checkbox">

```



```
<i data-toggle='modal' data-target='#modal-config' class='fa fa-
cog' style='cursor: pointer;'></i>
</div>
</div>
</div>
</div>
<!-- END SENSOR TEMPLATE -->
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.2/jquery.mi
n.js"></script>
    <script src='script/location.js'></script>
    <script src='script/jquery-migrate-
1.2.1.min.js'></script>
    <script src='script/jquery-ui.js'></script>
    <script src='script/bootstrap.min.js'></script>
    <script src='script/bootstrap-hover-
dropdown.js'></script>
    <script src='script/html5shiv.js'></script>
    <script src='script/respond.min.js'></script>
    <script src='script/jquery.metisMenu.js'></script>
    <script src='script/jquery.slimscroll.js'></script>
    <script src='script/jquery.cookie.js'></script>
    <script src='script/icheck.min.js'></script>
    <script src='script/custom.min.js'></script>
    <script src='script/jquery.menu.js'></script>
    <script src='script/pace.min.js'></script>
    <script src='script/holder.js'></script>
    <script src='script/responsive-tabs.js'></script>
    <!--CORE JAVASCRIPT-->
    <script src="script/main.js"></script>
</body>
</html>
```