

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему: «Інформаційна технологія візуалізації даних для системи підтримки прийняття рішень при управлінні енергетичними мікромережами»

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

Виконавець роботи: студент групи ІТ.м-01 Майковський Валентин Олександрович

**Кваліфікаційну роботу
захищено на засіданні ЕК
з оцінкою**

«__» грудня 2021 р.

Науковий керівник

(підпис)

к.т.н., доцент Парфененко Ю.В.

Голова комісії
(підпис)

Шифрін Д.М.

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

Студент _____
(підпис)

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

ЗАТВЕРДЖУЮ

Зав. кафедри ІТ

_____ В. В. Шендрик
«__» _____ 2021 р.

ЗАВДАННЯ

на кваліфікаційну роботу магістра студентіві

Майковський Валентин Олександрович

(прізвище, ім'я, по батькові)

1 Тема проекту Інформаційна технологія візуалізації даних для системи підтримки прийняття рішень при управлінні енергетичними мікромережами

затверджена наказом по університету від « 29 » _____ 10 _____ 2021 р. № 0787-VI

2 Термін здачі студентом закінченого проекту « 10 » _____ грудня _____ 2021 р.

3 Вхідні дані до проекту перелік вимог на розробку інформаційної технології візуалізації даних для системи підтримки прийняття рішень при управлінні енергетичними мікромережами

4 Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити) аналіз предметної області, постановка задачі та методи дослідження, проектування web – орієнтованої інформаційної системи візуалізації даних для системи підтримки прийняття рішень, практична реалізація інформаційної системи візуалізації даних

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

Мета і задачі, аналіз підходів до візуалізації даних в енергетиці, порівняльна таблиця додатків візуалізації енергетичних даних, аналіз web – додатків для візуалізації даних, порівняльна таблиця web – додатків, функціональні вимоги до web – додатку, засоби реалізації, контекстна діаграма процесу візуалізації, декомпозиція діаграми процесу візуалізації, діаграма варіантів використання web – додатку, діаграма послідовності використання головної адміністративної сторінки, діаграма послідовності використання сторінки сонячних панелей, діаграма послідовності сторінка з архівними даними, архітектура системи моніторингу, реалізація, демонстрація web – додатку, висновки.

6. Консультанти випускної роботи із зазначенням розділів, що їх стосуються:

| Розділ | Консультант | Підпис, дата | |
|--------|-------------|----------------|------------------|
| | | Завдання видав | Завдання прийняв |
| | | | |

Дата видачі завдання _____.

Керівник _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

| № п/п | Назва етапів випускної проекту | Термін виконання етапів проекту | Примітка |
|-------|--|---------------------------------|----------|
| 1 | Аналіз предметної області | 01.10.21 – 06.10.21 | |
| 2 | Планування робіт | 07.10.21 – 07.10.21 | |
| 3 | Розробка контекстної діаграми | 07.10.21 – 07.10.21 | |
| 4 | Декомпозиція на етапи інформаційної технології | 08.10.21 – 08.10.21 | |
| 5 | Ідентифікація мети ІТ – проекту | 08.10.21 – 09.10.21 | |
| 6 | Планування змісту структури робіт | 09.10.21 – 10.10.21 | |
| 7 | Побудова календарного графіку | 12.10.21 – 12.10.21 | |
| 8 | Управління ризиками | 13.10.21 – 13.10.21 | |
| 9 | Розробка макету | 14.10.21 – 15.10.21 | |
| 10 | Написання скриптів для відображення погоди | 15.10.21 – 27.10.21 | |
| 11 | Візуалізація даних | 28.10.21 – 05.12.21 | |
| 12 | Тестування та відладка | 06.12.21 – 09.12.21 | |
| 13 | Запуск web – додатку | 10.12.21 – 10.12.21 | |

Магістрант _____

Майковський В.О.

Керівник роботи _____

к.т.н., доцент Парфененко Ю.В.

РЕФЕРАТ

Тема кваліфікаційної роботи магістра «Інформаційна технологія візуалізації даних для системи підтримки прийняття рішень при управлінні енергетичними мікромережами».

Пояснювальна записка складається зі вступу, 4 розділів, висновків, списку використаних джерел із 31 найменування, додатків. Загальний обсяг роботи – 104 сторінки, у тому числі 70 сторінок основного тексту, 2 сторінки списку використаних джерел, 30 сторінок додатків, 90 рисунків, 6 таблиць.

Кваліфікаційну роботу магістра присвячено розробці інформаційної технології візуалізації даних для системи підтримки прийняття рішень при управлінні енергетичними мікромережами, що дозволить в подальшому допомогти користувачам в вирішенні, чи користуватися мікромережею, чи припинити її використання.

В роботі проведено аналіз предметної області, проаналізовано та обрано технології реалізації, розроблено та проведено тестування інформаційної технології візуалізації даних для системи підтримки прийняття рішень, розроблено web-додаток та завантажено на хостинг.

У роботі виконано авторизацію користувачів, візуальне представлення даних, що надходять з бази даних, створено скрипт звернення до API OpenWeatherMap для отримання погодних показників.

Результатом проведеної роботи є готовий та працездатний web – додаток інформаційної технології візуалізації даних.

Практичне значення роботи полягає у допомозі користувачу прийняти рішення щодо використання мікромережі в подальшому.

Ключові слова: інформаційна технологія, web – додаток, мікромережа.

ЗМІСТ

| | |
|---|----|
| ВСТУП..... | 7 |
| 1.2 Аналіз підходів до візуалізації даних в енергетиці | 12 |
| 1.3 Аналіз сайтів для візуалізації даних..... | 21 |
| 2. ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ..... | 38 |
| 2.1 Мета та задачі | 38 |
| 2.2 Вибір технології реалізації інформаційної технології візуалізації даних | 39 |
| 3 ПРОЕКТУВАННЯ WEB-ОРІЄНТОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ВІЗУАЛІЗАЦІЇ ДАНИХ ДЛЯ СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ..... | 41 |
| 3.1 Структурно-функціональне моделювання візуалізації даних для системи підтримки прийняття рішень..... | 41 |
| 3.2 Моделювання варіантів використання інформаційної системи візуалізації даних..... | 44 |
| 4 ПРАКТИЧНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ВІЗУАЛІЗАЦІЇ ДАНИХ | 49 |
| 4.1 Архітектура інформаційної системи | 49 |
| 4.2 Реалізація web-додатку | 50 |
| 4.3 Приклад використання web-додатку | 62 |
| ВИСНОВОК | 70 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ | 71 |
| Додаток А. Планування робіт..... | 75 |
| Додаток Б. Фрагмент коду API OpenWeatherMap | 83 |
| Додаток В. Код головної сторінки | 84 |

ВСТУП

Альтернативна енергія – це все, що зараз стоїть на думці всіх людей. Адже з кожною новою будівлею зростає попит на електроенергію, тільки з кожним додатковим попитом зростає нестача енергії, а відповідно відбувається перевантаження мережі. І раз за разом збільшується видобуток традиційних джерел енергії, тобто корисних копалин – нафти, газу, вугілля. Та є велика проблема, вони не можуть відновитися. Це перший фактор, який мотивує використання відновлювальних джерел енергії. Ось тому вчені та інженери намагаються спростити зручність надходження і безпечного використання джерел енергії. Зараз єдиний і доступний спосіб знизити зміни клімату – це перехід на екологічні джерела енергії, що можуть відновлюватися, або як прийнято говорити альтернативні: сонце, вітер, вода та ін. Саме заради цього були побудовані сонячні та вітрові електростанції [1]. Адже за допомогою сонячної енергії можна частково або повністю забезпечити мешканців приватного сектору виробленою електроенергією [2].

Тому все частіше інженери намагаються створити мікромережі – системи, які здатні робити окремо від центральної мережі електропостачання. Зазвичай такі мережі включають в себе високий відсоток джерел відновлювальної енергії таких, як фотоелектрична та вітрогенерація. Але у таких мікромережах часто виникають проблеми, які пов'язані з їх внутрішніми обмеженнями джерел живлення [3]. Через це зазвичай неможливо побачити, як ефективно використовується енергія та на скільки її використовують.

Відносно цього постає проблема, як переглядати та контролювати “домашню” енергію звичайному клієнтові. Саме для вирішення цих фундаментальних проблем використовуються системи управління мікромережами, які допоможуть співробітникам компанії прийняти рішення за допомогою візуалізації даних, що в свою чергу обумовлює актуальність роботи.

Об'єкт дослідження – інформаційне забезпечення підтримки прийняття рішень при управлінні енергетичними мікромережами.

Предмет дослідження – інформаційна технологія візуалізації даних при управлінні енергетичними мікромережами.

Мета – розробити інформаційну технологію візуалізації даних для системи підтримки прийняття рішень при управлінні енергетичними мікромережами.

Для досягнення мети потрібно вирішити наступні задачі:

- провести детальний аналіз предметної області;
- розробити інформаційну технологію візуалізації даних для системи підтримки прийняття рішень;
- розробити web – додаток візуалізації даних;
- провести тестове моделювання процесу візуального відображення даних.

Практичне значення: інформаційна технологія буде впроваджена у web-додаток, що буде використовуватись для надання користувачу повноцінного контролю в мікромережі з можливістю перегляду даних у вигляді графічного відображення для подальшого прийняття рішень щодо використання альтернативної електроенергії.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Дослідження актуальності проблеми

Зі збільшенням потреб в електричній енергії, виникає питання в нехватці її для користувачів. Більшість намагається знайти вихід в незалежному користуванні електричною енергією, а тому все частіше дивляться в бік незалежних мікромереж. Адже це дасть змогу самостійно контролювати споживання енергії в критичний момент, наприклад, під час віялового відключення світла. Таким способом є можливість забезпечити енергією важливі пристрої, яким потрібна електрика для постійної роботи.

Саме через такі моменти з'являється потреба у впровадженні мікромереж, що мають змогу працювати автономно, без залежності від центральної енергетичної мережі. У більшості випадків такі мережі мають високий показник природньої відновлювальної енергії. Та якби добре не було, у кожній такої мережі, є помилки чи проблеми, що виникають у момент їх використання та самими внутрішніми обмеженнями джерел живлення.

Для вирішення цих фундаментальних проблем використовуються системи підтримки прийняття рішень при управлінні мікромережами [4-6]. Основна роль такої системи полягає в підвищеній відмовостійкості мережі.

Системи підтримки прийняття рішень (DSS) – це інформаційні системи, які об'єднують моделі та інструменти збирання, обробки та аналізу даних для сприяння прийняттю рішень.

Розглянемо типову архітектуру системи підтримки прийняття рішень для мікромереж (рис 1.1).

Платформа Smart DSS, представлена в роботі [7], реалізована у вигляді клієнт-серверного веб-додатку. Модуль на стороні клієнта дозволяє користувачеві реєструватися і згодом виконувати операції по визначенню та управлінню як моделями, так і екземплярами, а саме такі, як створення, модифікація, клонування, збереження і видалення. Крім того, клієнт пропонує можливості визначення логічних

функцій, оцінки значення, ваги критеріїв прийняття рішень і для розрахунку коефіцієнтів остаточного рішення. Серверна частина працює як інтерфейс для управління на стороні клієнта і адміністрування користувачів, а також інтерфейс до реляційної бази даних SQL.

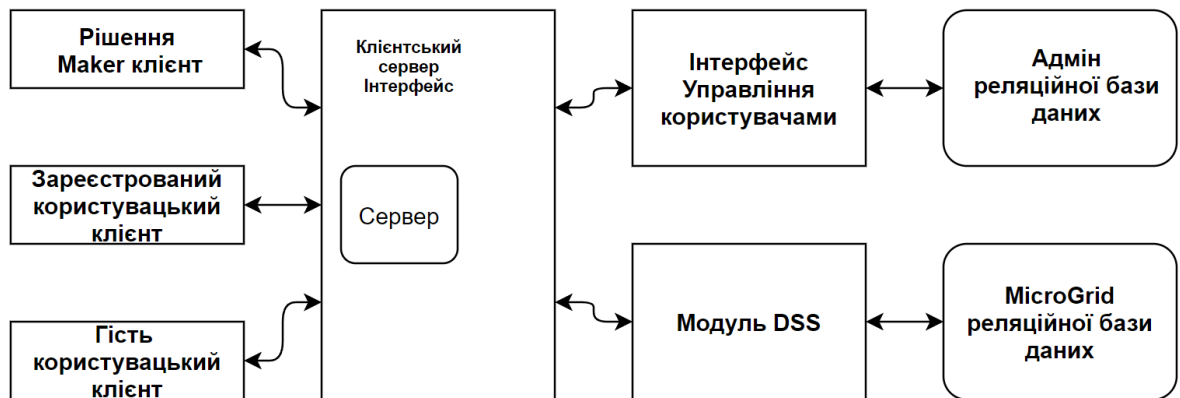


Рисунок 1.1 – Архітектура системи підтримки прийняття рішень (DSS)

Система надає чотири різних типи користувачів:

- Гостьовий користувач, який не потрібний для аутентифікації;
- Зареєстрований / просунутий користувач: після реєстрації просунутий користувач може змінювати моделі і екземпляри;
- Особа, яка приймає рішення, є зареєстрованим користувачем, якому були надано можливості прийняття рішень адміністратором. Особи, які приймають рішення, можуть створювати нові моделі і екземпляри, змінювати і зберігати їх, збереження і видалення існуючих і імпорт даних у клонованих примірниках з іншого примірника тієї ж моделі.
- Адміністратор, який може виконувати ті ж дії, що і Особа, яка приймає рішення, і, крім того, що володіє повним управління всіма зареєстрованими користувачами, можливість призначати і змінювати всі ролі користувачів.

Сервер функціонально розділений на два основних модуля: модуль DSS і інтерфейс у правління користувачами. Перший відповідає за управління модулями та екземплярами DSS, а також операції, виконувані над ними користувачами, відповідно

до їх ролей і привілеїв. Останні використовуються адміністратором для управління різними типами зареєстрованих користувачів і ролей. Кожен з цих двох модулів складаються з інтерфейсу клієнт / сервер, ядра модуля та додаткового інтерфейсу, призначеного для управління інформацією, зібраної або, що зберігається в базі даних SQL.

Користувацький інтерфейс веб-програми на стороні клієнта дозволяє користувачам виконувати операції в відповідно до їхніх привілеїв та ролей, тому інтерфейс візуалізації є важливим компонентом системи підтримки прийняття рішень.

Моніторинг та управління енергоспоживанням – процеси, спрямовані на зниження витрат на електроенергію в галузях промисловості та в приватному домогосподарстві. Такі системи цілодобово контролюють споживання енергії в різних приміщеннях чи локаціях.

Дані по використанню енергетичного потоку дають уявлення про споживання енергії в різних приміщеннях, або джерелах їх надходження, а також дають нам інформацію про раптові зміни в потребах в енергії з додатковою інформацією, такою як конкретний час використання, чи інформацію про конкретний прилад: сонячна батарея чи заправна електростанція.

Платформа управління енергоспоживанням є прикладом використання Інтернету для моніторингу та управління енергоспоживанням. Це рішення дає змогу переглянути те, як люди використовують енергію та накопичувачі енергії і керують ними. Після підключення до даної системи, будь який прилад перетворюється на інтелектуальний простір з багатим набором функцій та можливостей для моніторингу та контролю енергоспоживання [8].

До переваг системи моніторингу можна зробити помітку на динамічності, оскільки об'єкти за якими спостерігають, постійно перебувають в зміні характеристик чи розвитку. Під час такого спостереження є можливість передбачити, збій об'єкта чи ймовірність небезпеки, в момент його роботи [9 - 10].

Розробка такої системи є необхідною для полегшення спостереження використання енергії з сонячних батарей чи зарядної станції. А особливо при

використанні спеціальних трекерів для спостереганні за сонцем [11], оскільки стаціонарно фіксовані накопичувачі, дають змогу виробити енергію тільки в певний проміжок часу. То такі трекери збільшують кількість згенерованої електроенергії приблизно на 35-50% [12], так як рухають батареї за сонцем.

Тому для покращення розуміння витраченої енергії, потрібно розробити web – додаток, який буде обробляти дані, які використовуються для управління мікромережею, та в короткий час візуалізувати данні у вигляді графіків та інших діаграм.

1.2 Аналіз підходів до візуалізації даних в енергетиці

У результаті попереднього дослідження актуальності проблеми було визначено, що до аналізу підходів до інформаційної технології візуалізації даних для системи підтримки прийняття рішень при управлінні енергетичними мікромережами, потрібно обрати системи, які включають візуалізацію даних з використанням діаграм, графіків, таблиць, тощо. Тому було вирішено проаналізувати системи моніторингу мереж, які працюють в режимі реального часу, та мають можливість візуалізувати дані.

Оскільки в відкритому доступі таких систем не має, тому аналіз відбувався за допомогою огляду літературних джерел, в який представлений опис програмних додатків. Метою даного розділу є порівняння систем для візуалізації даних, а саме яким чином можна візуалізувати дані: діаграми, графіки, схематичні дані, таблиці тощо. Якими додатками та які дані можна візуалізувати.

Першим додатком було розглянуто Microgrid Digital Control Solution from Schneider Electric [13]. Це стаціонарна програма, яка дозволяє переглядати дані, що надходять в реальному часі з функцією симуляції, де можна побачити звідки і куди надходять данні, також переглянути погодні умови. Додатково можна побачити максимальну напругу та відсоток споживання енергії (рис 1.2).

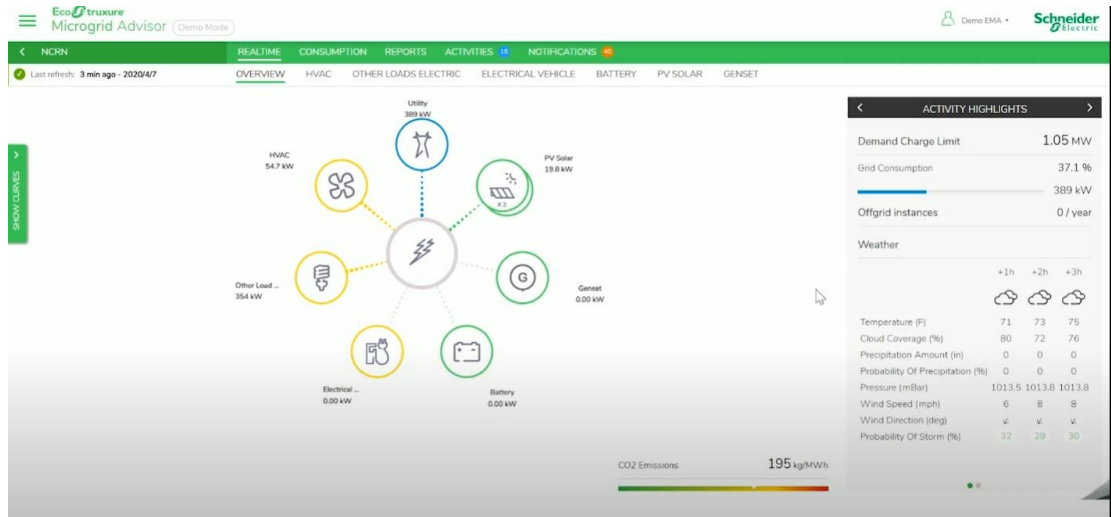


Рисунок 1.2 – Головна сторінка MDCS from Schneider Electric

В інтерфейсі даної програми є можливість переглянути у вигляді графіків скільки вироблено енергії (рис 1.3).



Рисунок 1.3 – Відображення у вигляді графіків

Особливістю даної програми є можливість переглянути окремі компоненти мережі та відсоток їх енергії (рис 1.4).

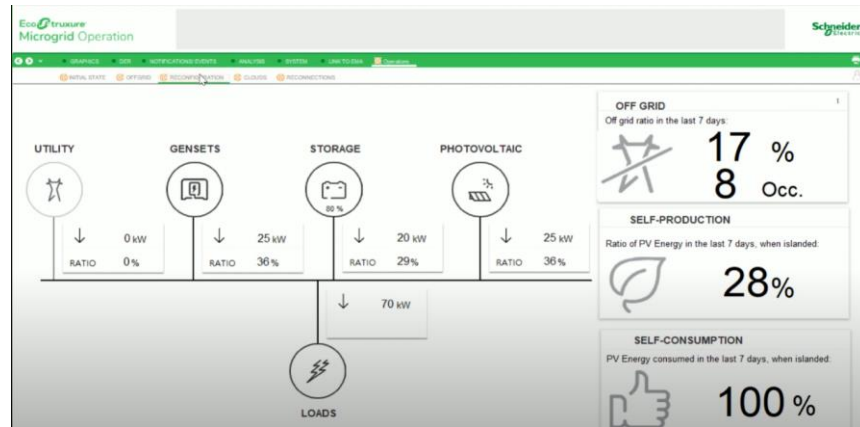


Рисунок 1.4 – Детальні показники енергоефективності

Зробити заключення після огляду одного додатку не можливо, тому на огляді було також розглянуто додаток «Grafana» [14]. Панель моніторингу Grafana збирає інформацію і відображає її в зручній формі для користувача, який встановив онлайн-з'єднання через веб-браузер. Інтерфейс моніторингу можна побачити на рис. 1.5.

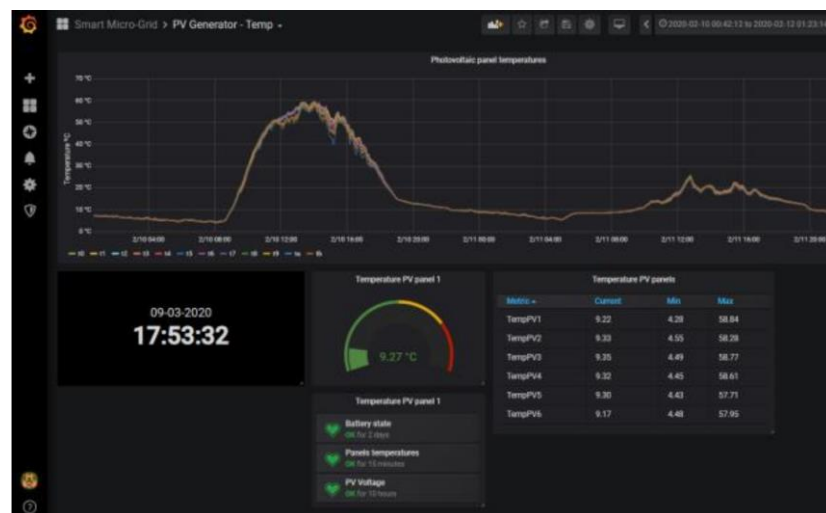


Рисунок 1.5 – Внутрішній вигляд Grafana

Він складається з графіка часових рядів, щоб оператор міг спостерігати за еволюцією вимірюваних температур. Користувач може збільшити масштаб графіка, а також вибрати часовий інтервал для візуалізації. Крім того, датчик ілюструє миттєве значення однієї температури, щоб з першого погляду дізнатися значення на датчику. Іншим елементом, який був визнаний актуальним, є таблиця, в якій представлені

показники, а саме поточна, мінімальна і максимальна температура для кожного датчика.

Розробка інформаційних панелей за допомогою Grafana – просте завдання, головним чином, завдяки широкому спектру вбудованих функцій і доступних доповнень. Отже, інтуїтивно зрозуміла і приваблива графіка може бути розгорнута без тривалих термінів розробки або навичок програмування.

Третій додаток є модифікацією програмного забезпечення Matlab, точніше його бібліотекою. Simscape Power Systems – це стандартна бібліотека SIMULINK. Вона складається з бази даних електронних компонентів, а також інструментів моделювання та аналізу для електроенергетичних систем. Бібліотека також призначена для розробки і тестування систем управління в інтеграції з електроенергетичними системами; це робить її придатною для моделювання мікромереж [15]. Представлена бібліотека дозволяє створити симуляцію мікромережі та виявити її недоліки. Приклад готової системи для тестування зображено на рис.1.6.

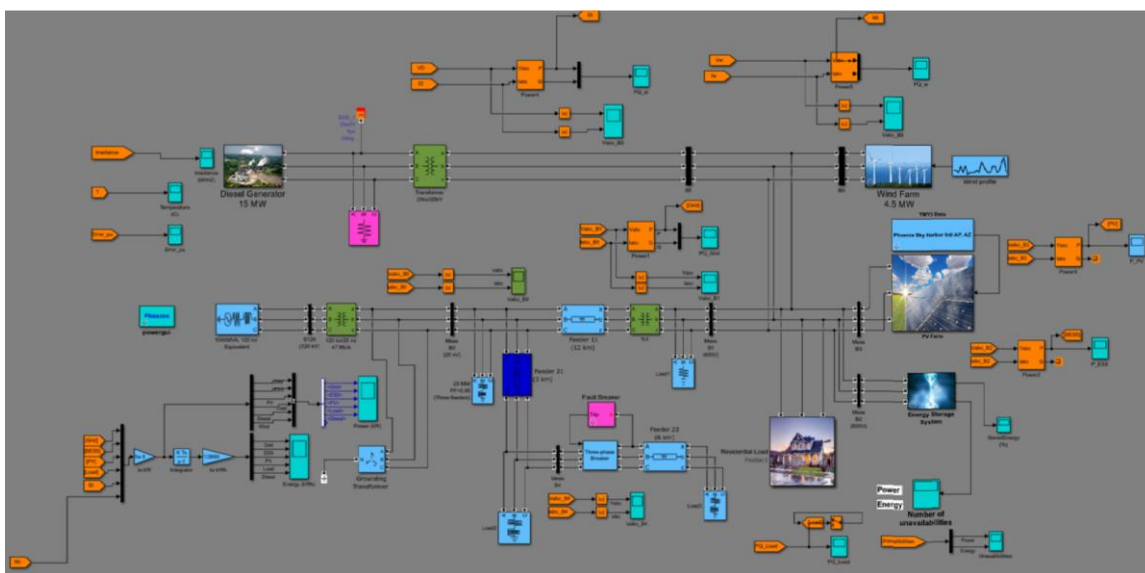


Рисунок 1.6 – Імітація мікросистеми в SIMULINK

Дана бібліотека дозволяє провести симуляцію, в результаті якої програма виводить візуально показники у вигляді графіків в залежності від заданих параметрів (рис. 1.7 – 1.8).

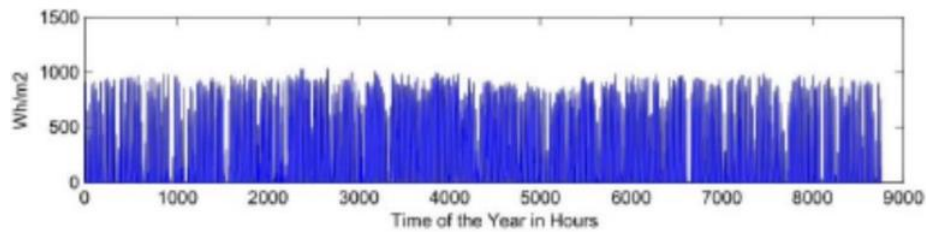


Рисунок 1.7 – Стовпчаста діаграма

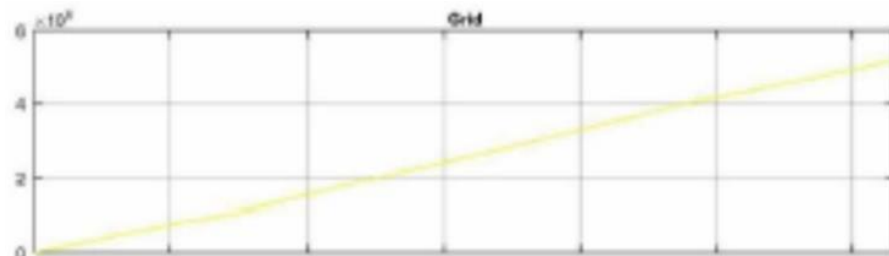


Рисунок 1.8 – Лінійна діаграма

Розглянемо повнофункціональний додаток для симуляції мікросистем в реальному часі ETAP Microgrid Management [16]. Приклад схеми симуляції можна поглянути на рис.1.9.

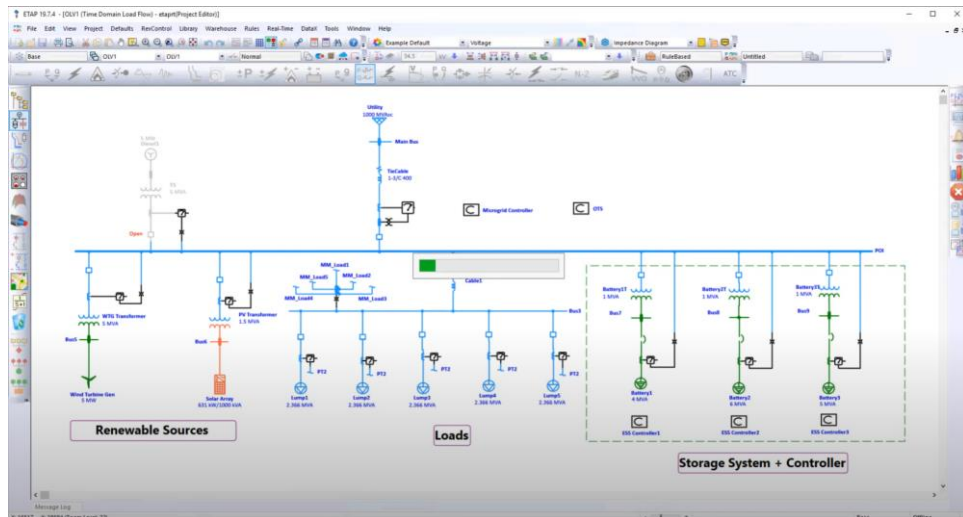


Рисунок 1.9 – Симуляція ETAP Microgrid Management

Після включення симуляції можна переглянути детальні показники окремих пристроїв в реальному часі, так нібито зараз підключені фізичні пристрої. Якщо детально поглянути на рис. 1.10, можна побачити, які саме пристрої підключені до

системи та можливість відстежувати в реальному часі їх показники. Поглянувши праворуч, можна побачити, що програма має змогу виводити інформацію у вигляді графіків.

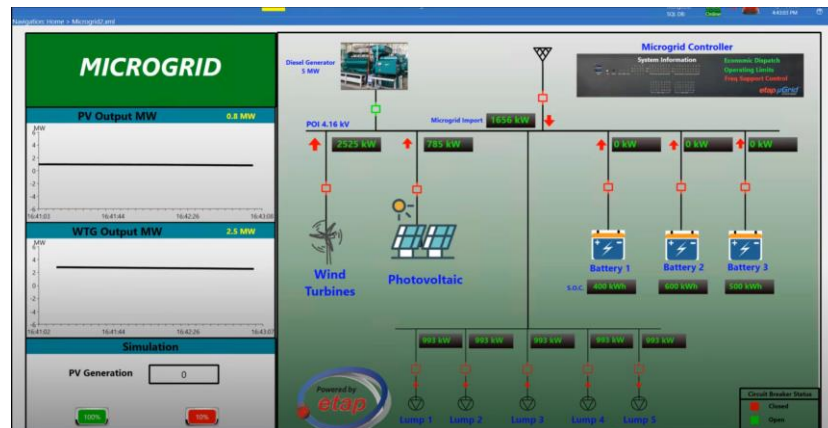


Рисунок 1.10 – Детальні показники

Є змога переглянути показники окремо підключеного пристрою (рис. 1.11).

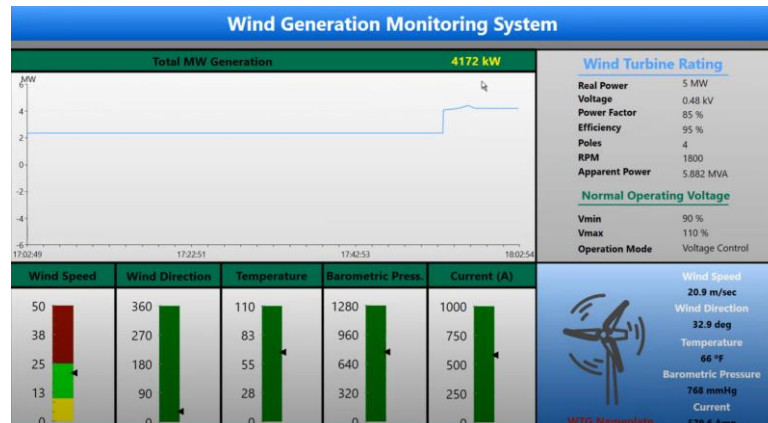


Рисунок 1.11 – Показники пристрою

Детальніше поглянувши на рисунок, можна побачити, що в даному шаблоні є змога переглянути конкретну енергетичну спроможність у вигляді графіку, а також окремі показники, такі, як швидкість, температура, напруга, вологість та ін. у вигляді стовпчикової діаграми і звичайної таблиці. Є можливість переглянути окремі показники у вигляді кругової діаграми (рис 1.12).

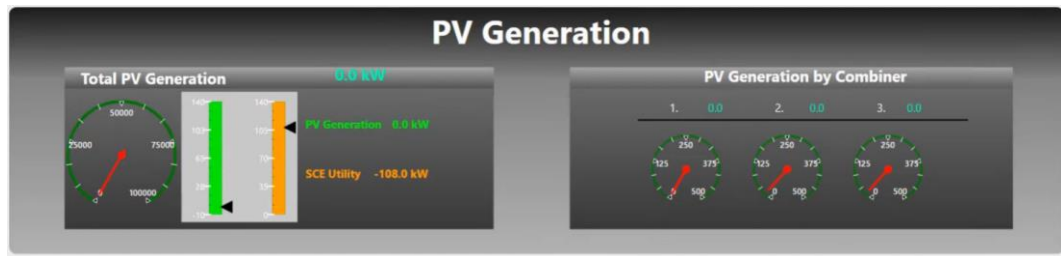


Рисунок 1.12 – Кругові діаграми

Цікавою для огляду є система, яка на даний момент в розробці, але привертає до себе увагу – це Monash Microgrid [17].

Дана система дозволяє візуалізувати мікросистему у вигляді 3D моделі. А точніше робить візуалізацію реальних будівель на Google карті та зображує систему підключення енергетичної потужності та її підключення до будівель (рис 1.13).

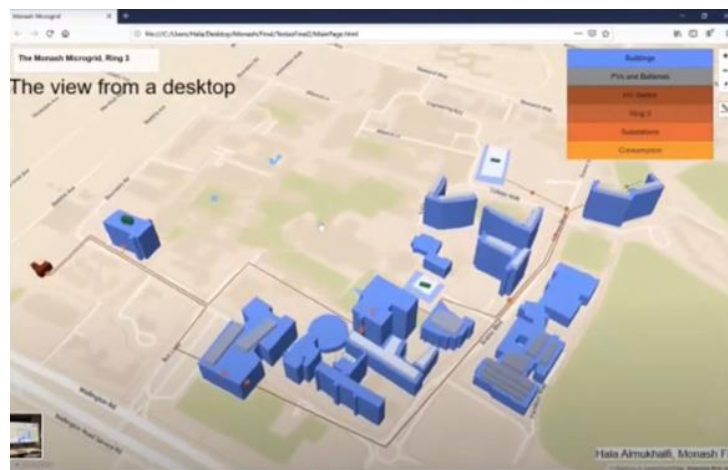


Рисунок 1.13 – Monash Microgrid

Також хотілося б взяти до уваги найстарішу, але досі популярну систему для візуалізації – це LabVIEW Visualization [18]. Графічний інтерфейс розроблений з використанням програмного забезпечення LabVIEW для візуалізації параметрів системи в режимі реального часу. Наприклад, можна візуалізувати величини постійного струму інверторів, такі як постійна напруга, струм і потужність сонячної панелі. В даній програмі є можливість зробити декілька режимів візуалізації даних в реальному часі (рис. 1.14 – 1.16).

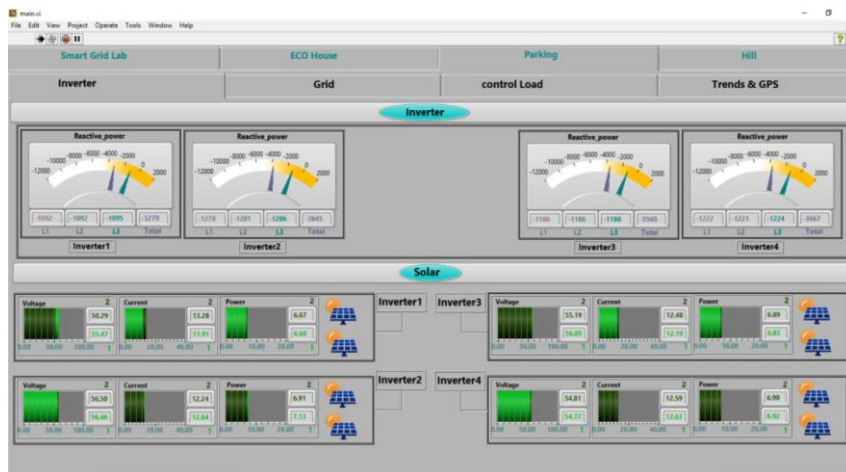


Рисунок 1.14 – 1 з 3 режимів візуалізації

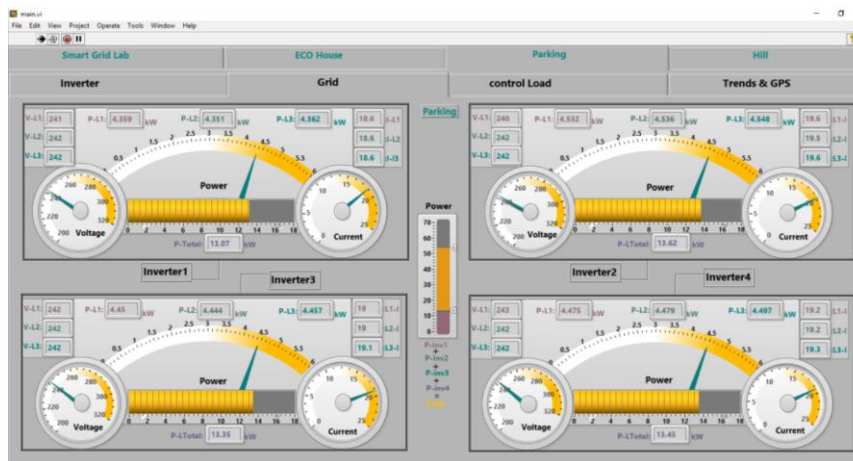


Рисунок 1.15 – 2 з 3 режимів візуалізації

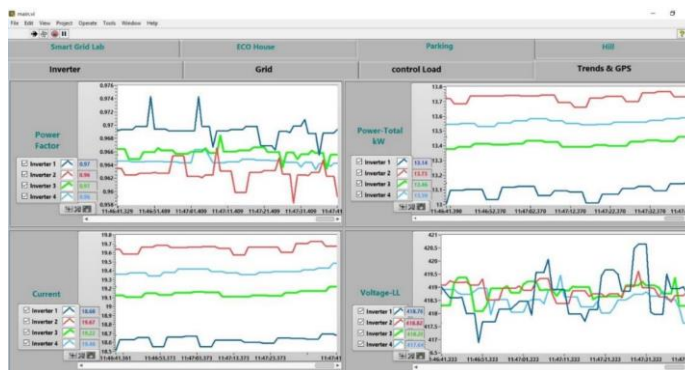


Рисунок 1.16 – 3 з 3 режимів візуалізації

В заключенні проведеного аналізу програмного забезпечення було створено табл. 1.1, яка дозволяє поглянути на переваги та недоліки програм за наступними характеристиками:

- можливість візуалізації в реальному часі;
- декілька типів візуалізації даних;
- можливість симуляції;
- можливість контролю за приладами.

*N/A – в таблиці позначено, ті характеристики до яких не знайдено детальної інформації

Таблиця 1.1 – Таблиця порівняння за характеристиками енергетичних додатків

| Характеристика | MDCS from Schneider Electric | Grafana | SIMULINK | LabVIEW Visualization | Monash Microgrid |
|--|---------------------------------------|---------|----------|--------------------------|---------------------|
| можливість візуалізації в реальному часі | + | + | + | + | N/A |
| декілька типів візуалізації даних | + | + | + | + | + |
| можливість симуляції | + | - | + | + | N/A |
| можливість контролю за приладами | + | - | - | + | N/A |

В заключенні над даним розділом можна сказати, що існує багато спеціалізованих програм, як локальних з встановленням на персональний комп'ютер, так і з можливістю віддаленого підключення через web – додаток. В залежності від задач кожний знайде для себе потрібний додаток, але найбільш підходящий додаток, який може підійти, як приклад для розробки інформаційної технології, це Grafana.

Оскільки така система дає змогу віддаленого підключення, тому що не потребує встановлення на локальний пристрій, наприклад комп'ютер.

Підбивши підсумки можна додати наступні вимоги для розробки інформаційної технології візуалізації:

- можливість візуалізації в реальному часі;
- декілька типів візуалізації даних.

1.3 Аналіз сайтів для візуалізації даних

Також було вирішено проаналізувати сайти, які дозволяють візуалізувати графічно дані з можливістю підключення баз даних або CSV файлів.

Провівши детальний пошук, прийнято рішення обрати три найоптимальніше підходящі ресурси:

- infogram.com;
- chartblocks.com;
- datadeck.com.

Після вибору даних ресурсів зроблено роботу над детальним аналізом, за основу взято графічну побудову та детальність інформації на графіках і можливість їх налаштування. Також підібрано критерії, за якими вони порівнювалися, які можна поглянути в табл. 1.2.

Першим ресурсом для порівняння був обраний ресурс [infogram](http://infogram.com). Зареєструвавшись на цьому сайті, в очі одразу потрапляє приємний градієнтний синьо-фіолетовий фон, а також вибір графіка, який треба побудувати (рис. 1.17), це можуть бути, як одиночні графіки так і цілі інформаційні панелі [19].



Рисунок 1.17 – Головна сторінка infogram

Для детального огляду було обрано інформаційну панель, одразу після вибору з'являється приємного виду панель з уже заготовленою інформацією (рис. 1.18).

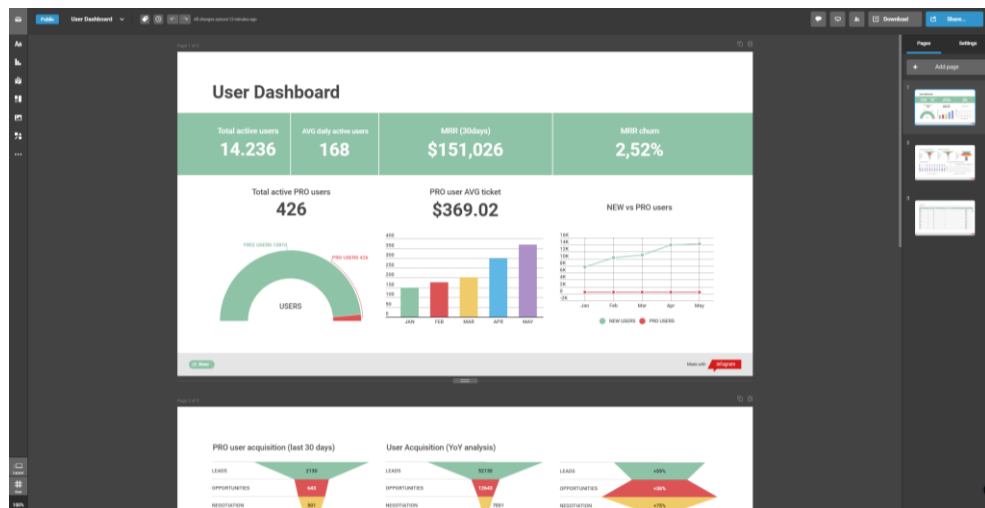


Рисунок 1.18 – Обрана інформаційна панель

Також, що дуже приємно, на сайті є темний фон. На сторінці окрім самих графіків є дві основні панелі меню, в яких можна налаштувати або додати додаткові об'єкти. Візьмемо на детальний огляд панель зліва, в ній можна додати текст, мапу, або додатковий графік з зображенням (рис. 1.19 – 1.20).



Рисунок 1.19 – Ліве меню infogram

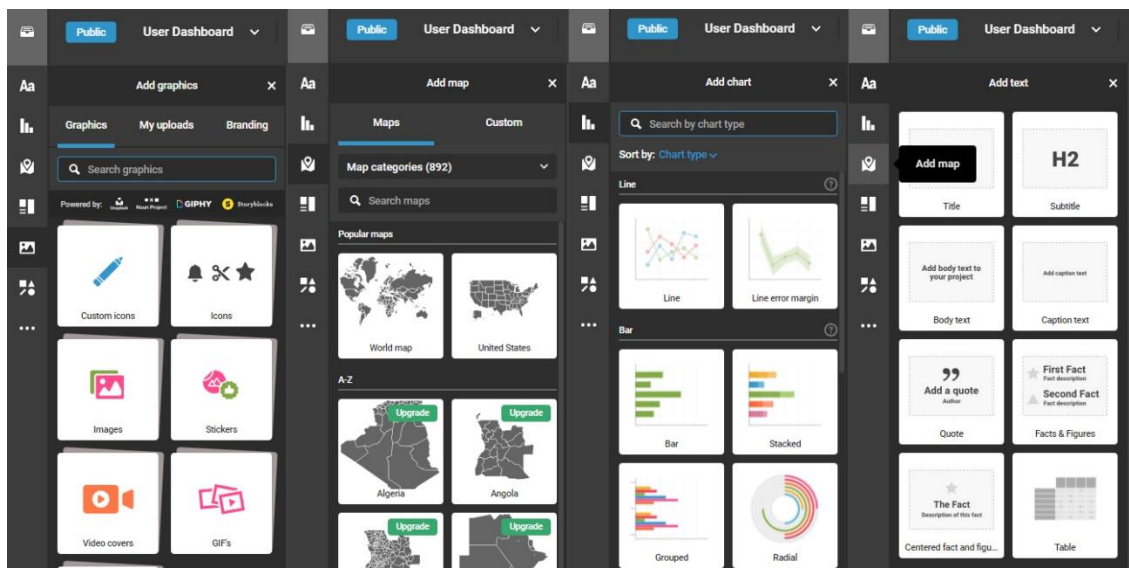


Рисунок 1.20 – Вибір блоку в лівому меню infogram

Далі, обравши один графік та натиснувши на нього, з'явилося додаткові налаштування в правому меню (рис 1.21).

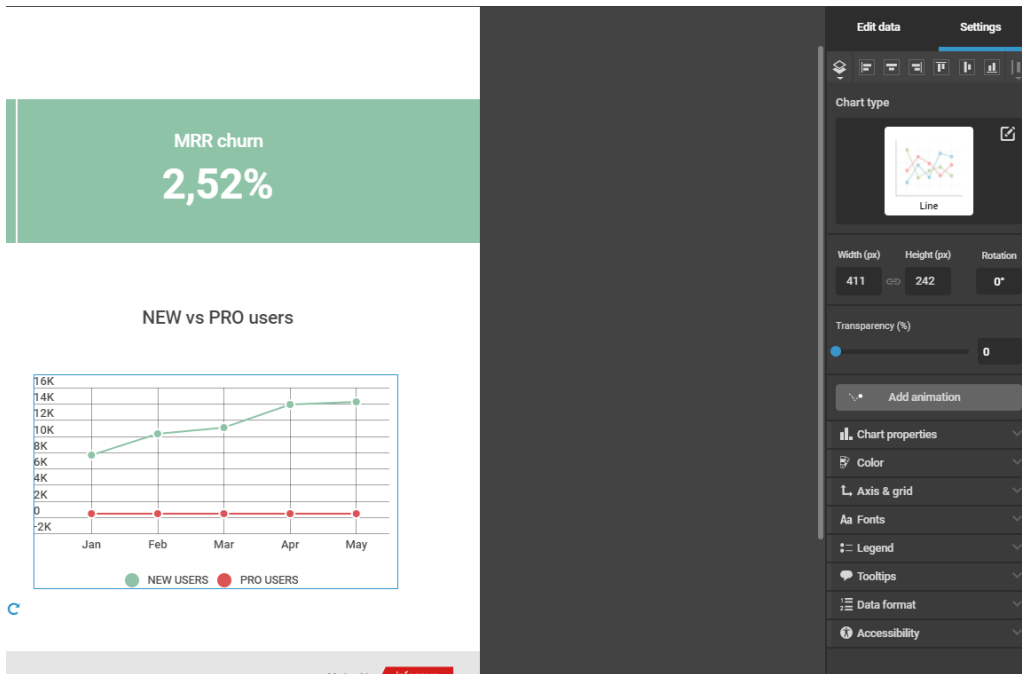


Рисунок 1.21 – Праве меню infogram

В цьому меню є два пункти. В першому маємо можливість налаштувати детальність відображення графіка, наприклад: додати анімацію, налаштувати текст, чи змінити колір (рис. 1.22).

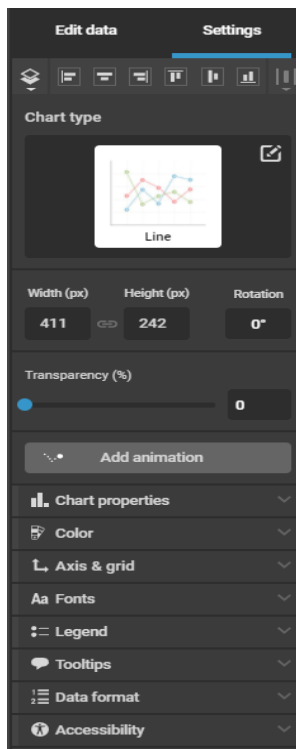
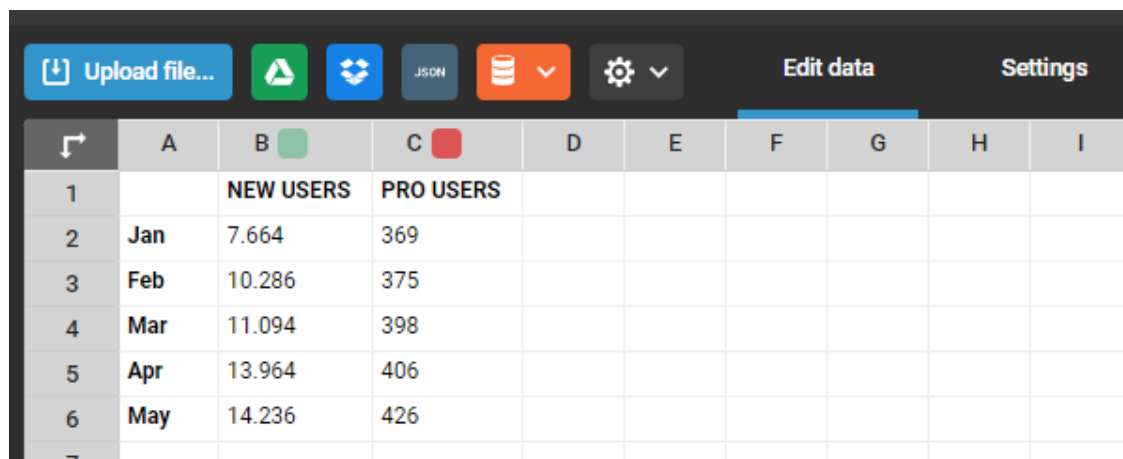


Рисунок 1.22 – Детальне дослідження графіків

В другому пункті є змога налаштувати вивід даних за допомогою самостійного вводу, документи, json файлу чи бази даних яку підтримує даний сайт (рис. 1.23 – 1.24).



| | A | B | C | D | E | F | G | H | I |
|---|-----|-----------|-----------|---|---|---|---|---|---|
| 1 | | NEW USERS | PRO USERS | | | | | | |
| 2 | Jan | 7.664 | 369 | | | | | | |
| 3 | Feb | 10.286 | 375 | | | | | | |
| 4 | Mar | 11.094 | 398 | | | | | | |
| 5 | Apr | 13.964 | 406 | | | | | | |
| 6 | May | 14.236 | 426 | | | | | | |
| 7 | | | | | | | | | |

Рисунок 1.23 – Дані для графіка

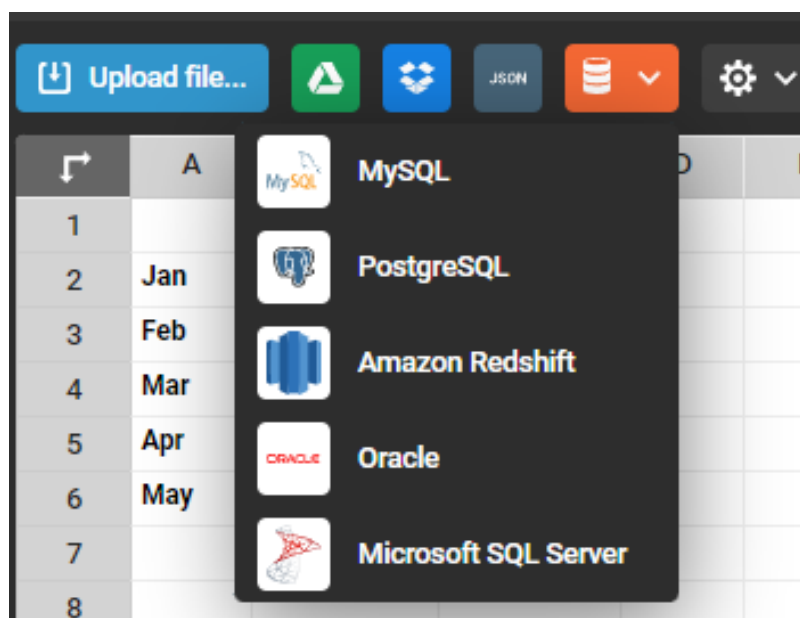


Рисунок 1.24 – Доступні бази даних

В заключенні огляду цього сайту можна сказати, в цілком дуже приємне меню, а особливо темна тема, яка дає змогу працювати в вечір або в ночі. Також, що сподобалося на графіках можна розгледіти всю інформацію, яка представлена в базі. Детально підписані вісі графіка, та є можливість переглянути збіг інформації в кожній точці координат (рис 1.25).

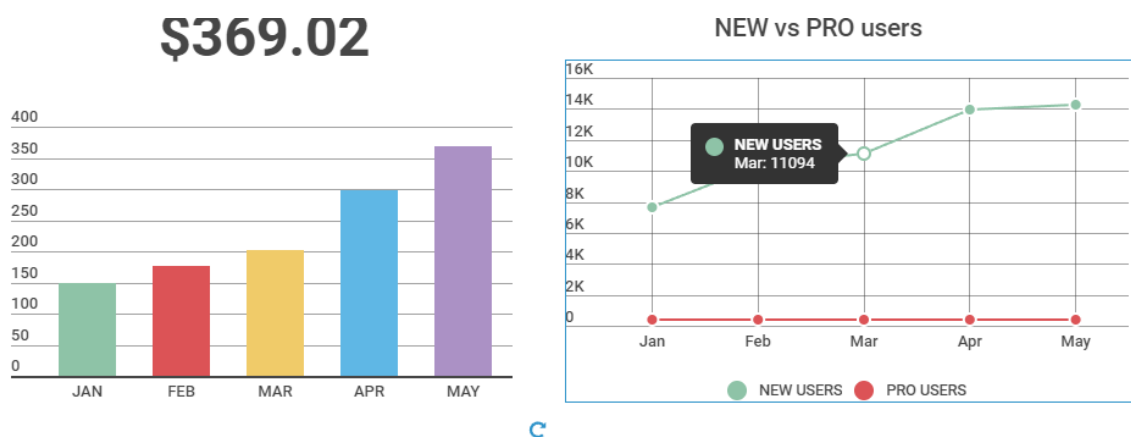


Рисунок 1.25 – Відображення графіків

До речі є можливість зберегти графіки до себе на комп'ютер в п'яти доступних форматах: png, jpeg, pdf, gif, а також mp4 (рис. 1.26).

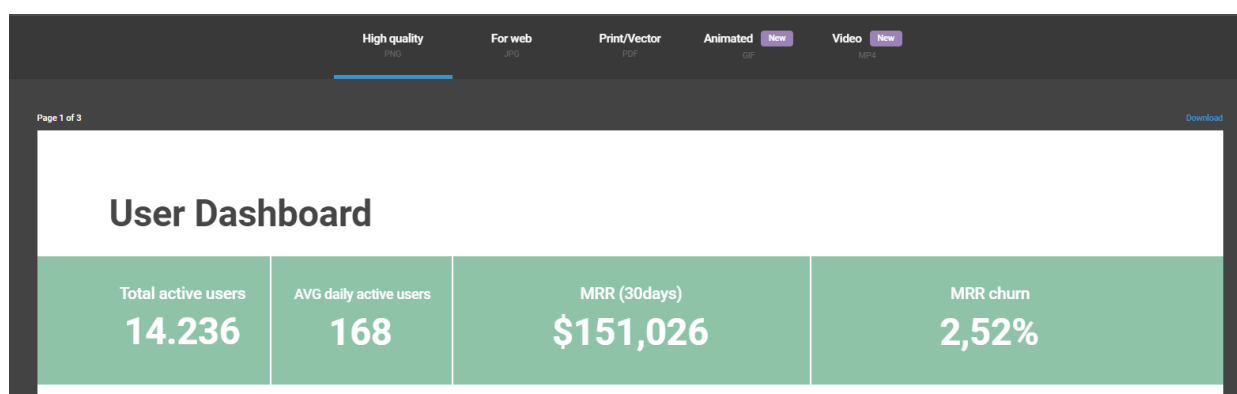


Рисунок 1.26 – Меню збереження графіків

На другому порталі, а саме Chartblocks [20] після авторизації, потрапляємо в адміністративну панель, де є змога розпочати роботу з графіками, або спочатку додати інформацію, у форматі csv або excel, чи створити базу самостійно (рис. 1.27 – 1.28).

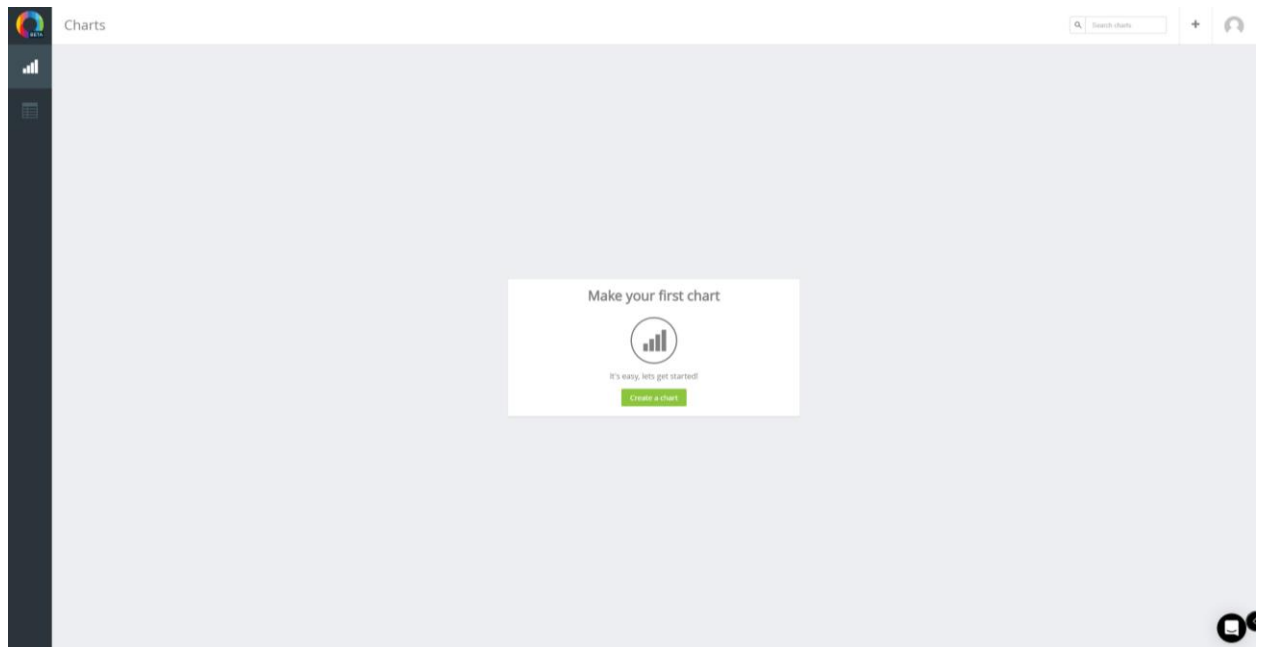


Рисунок 1.27 – Меню створення графіків

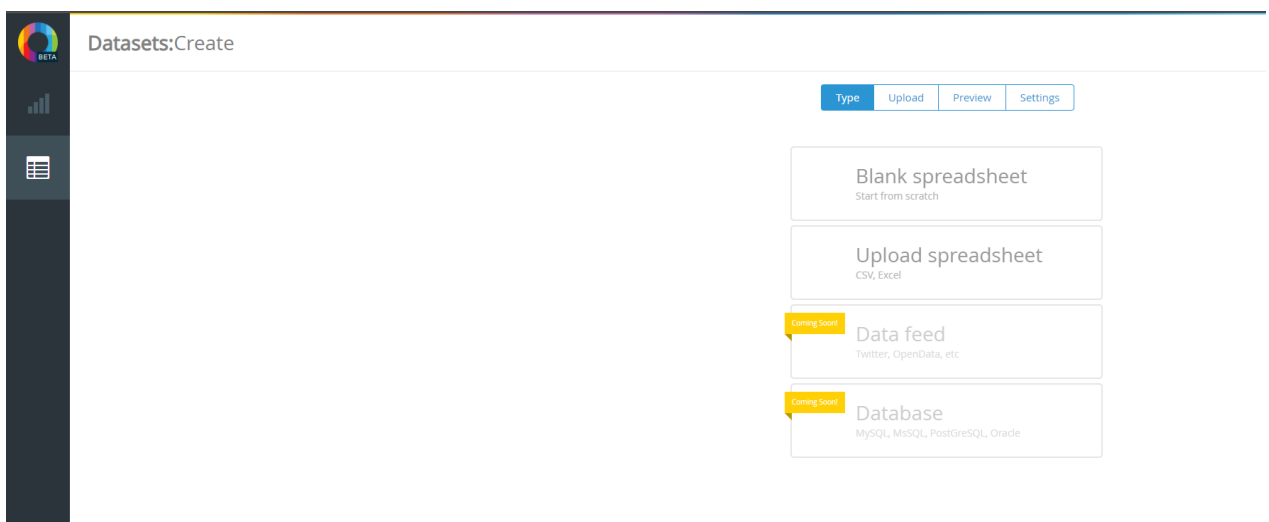


Рисунок 1.28 – Меню створення бази даних

Недоліком є те, що немає можливості одразу переглянути існуючі типи графіків, як на попередньому ресурсі. Спочатку треба вибрати базу даних (рис. 1.29), тому було вирішено створити імпровізовану базу даних і запозичити дані з минулого сайту (рис. 1.30).

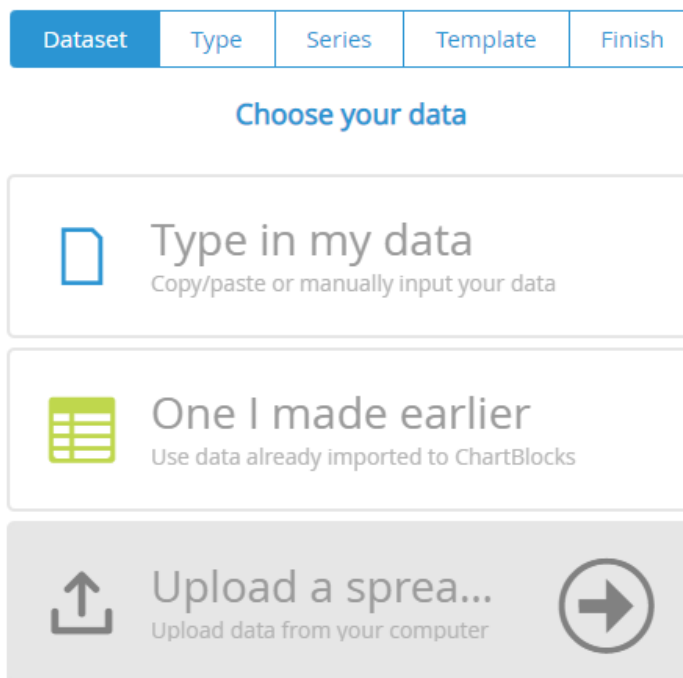


Рисунок 1.29 – Вибір бази даних

| | A | B | C |
|---|-----|-----------|-----------|
| 1 | | NEW USERS | PRO USERS |
| 2 | Jan | 7.664 | 369 |
| 3 | Feb | 10.286 | 375 |
| 4 | Mar | 11.094 | 398 |
| 5 | Apr | 13.964 | 406 |
| 6 | May | 14.236 | 426 |
| 7 | | | |

Рисунок 1.30 – Створення бази даних

Тільки після вибору бази є можливість обрати тип графіку (рис. 1.31), який потрібен, для порівняння було вирішено обрати лінійний графік.

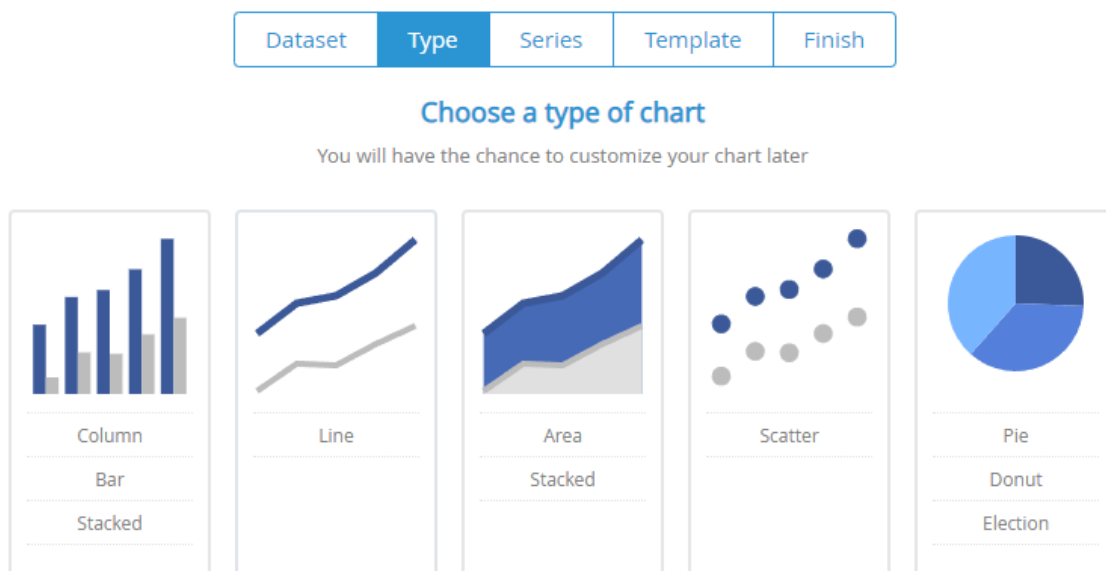


Рисунок 1.31 – Вибір типу графіку

Обравши графік, наступним кроком пропонують обрати колонки даних, які треба відобразити. Тут є можливість обрати, що саме треба відобразити на осі x та y (рис. 1.32). Встановлення даних відбувається в меню справа.

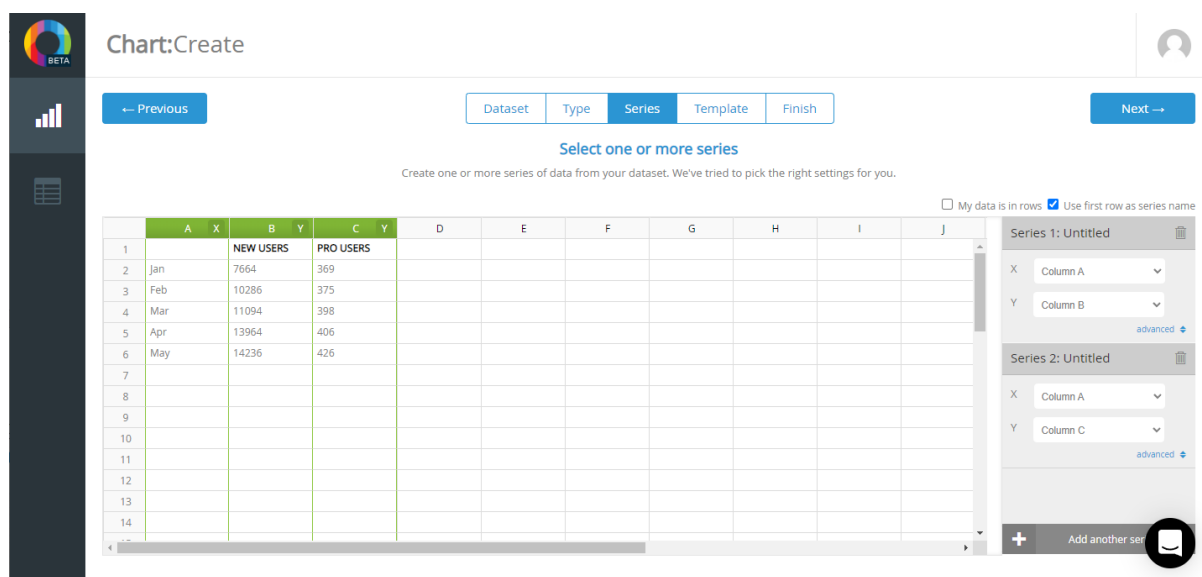


Рисунок 1.32 – Вибір параметрів

Встановивши параметри і зробивши підтвердження, з'являється сторінка, де пропонують обрати необхідний шаблон (рис. 1.33).



Рисунок 1.33 – Вибір шаблону відображення

Підтвердивши шаблон, з'являється результат роботи. Немає можливості налаштувати графік, але детальніше оглянувши сторінку, виявлено зелену кнопку “Customize” (рис. 1.34).

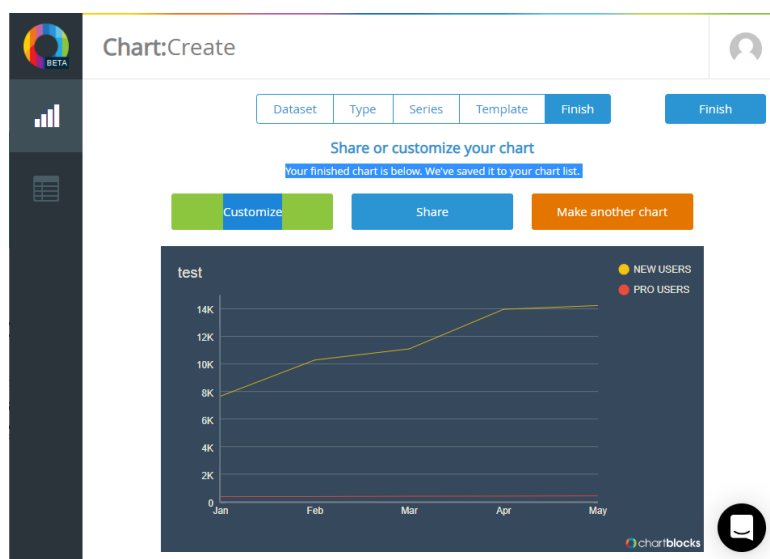


Рисунок 1.34 – Готовий результат

Натиснувши на кнопку, відкрилася сторінка з налаштуваннями, де можна було змінити тип графіку, відображення даних, змінити назву легенд, або взагалі поміняти місцями чи налаштувати координатну сітку (рис. 1.35).



Рисунок 1.35 – Сторінка налаштування

Отже в результаті отримуємо готовий графік, але щоб до нього дістатися потрібно пройти декілька пунктів. Також не має темної теми, але є можливість зберегти данні в двох форматах png та pdf.

Підійшовши до Datadeck [21] та провівши на ньому авторизацію, відкрилася адміністративна панель, з можливістю вибору побудови графіку чи підключення бази даних (рис. 1.36).

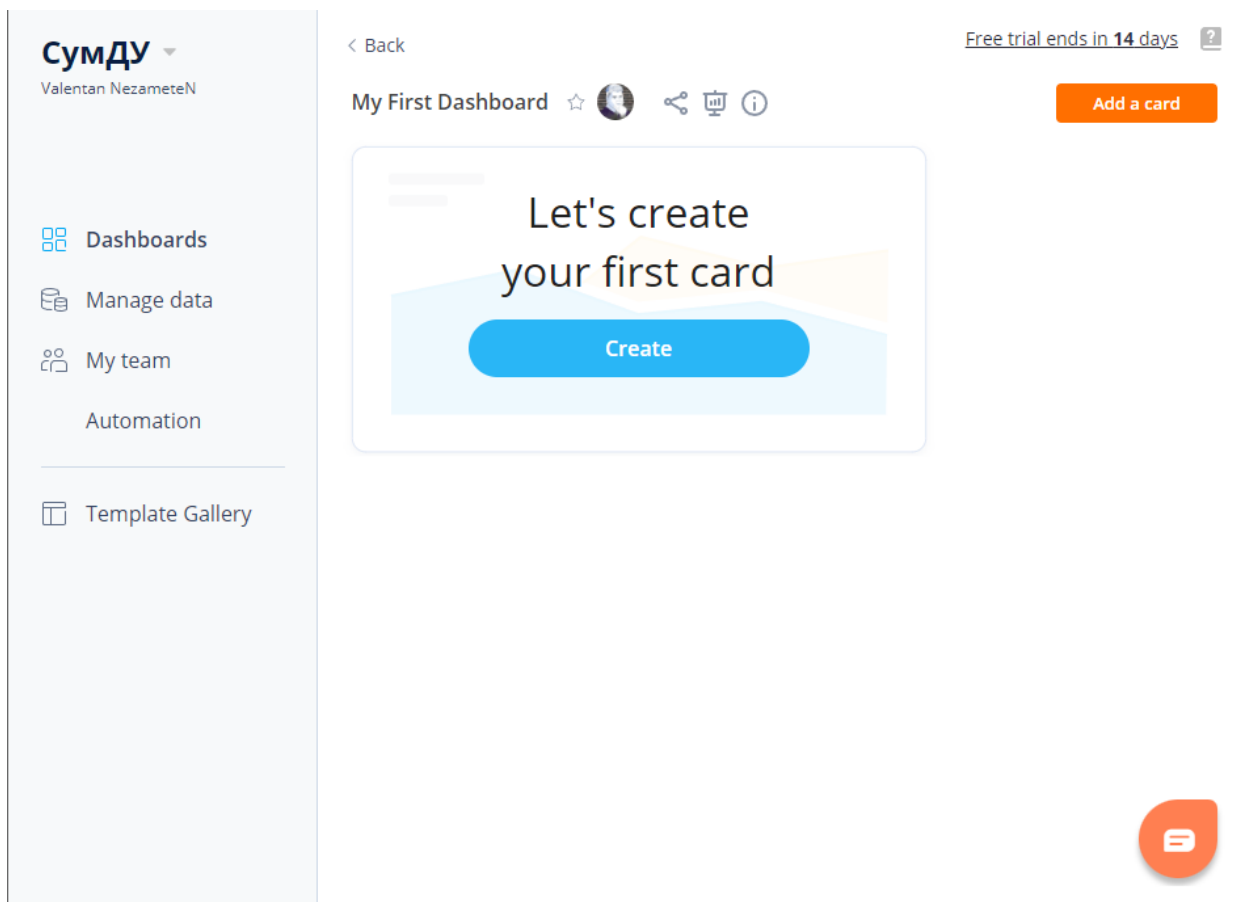


Рисунок 1.36 – Адміністративна панель

Натиснувши кнопку створити, з'явилося вікно, в якому пропонують обрати одну із доступних баз даних (рис. 1.37).

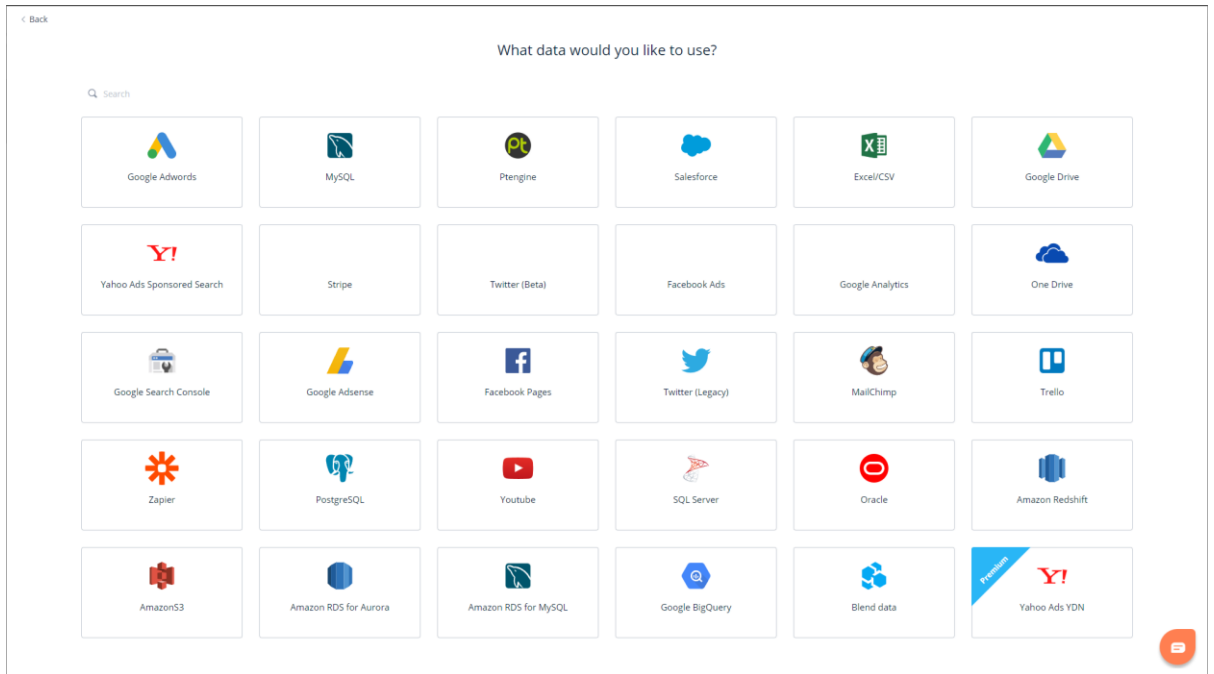


Рисунок 1.37 – Вибір бази

Для імпровізованої бази даних було обрано ексел файл, де дані запозичені з першого ресурсу. Після завантаження файлу відобразилося вікно для підтвердження точних даних, що знаходилися в файлі (рис. 1.38).

| | A | B | C |
|---|------------|-------------|-------------|
| 1 | T Column_1 | # NEW USERS | # PRO USERS |
| 2 | Jan | 7.664 | 369 |
| 3 | Feb | 10.286 | 375 |
| 4 | Mar | 11.094 | 398 |
| 5 | Apr | 13.964 | 406 |
| 6 | May | 14.236 | 426 |

Рисунок 1.38 – Дані, що знаходяться в файлі

Після всіх маніпуляцій відбулася переадресація на пусту панель, де є можливість налаштувати графік (рис. 1.39).

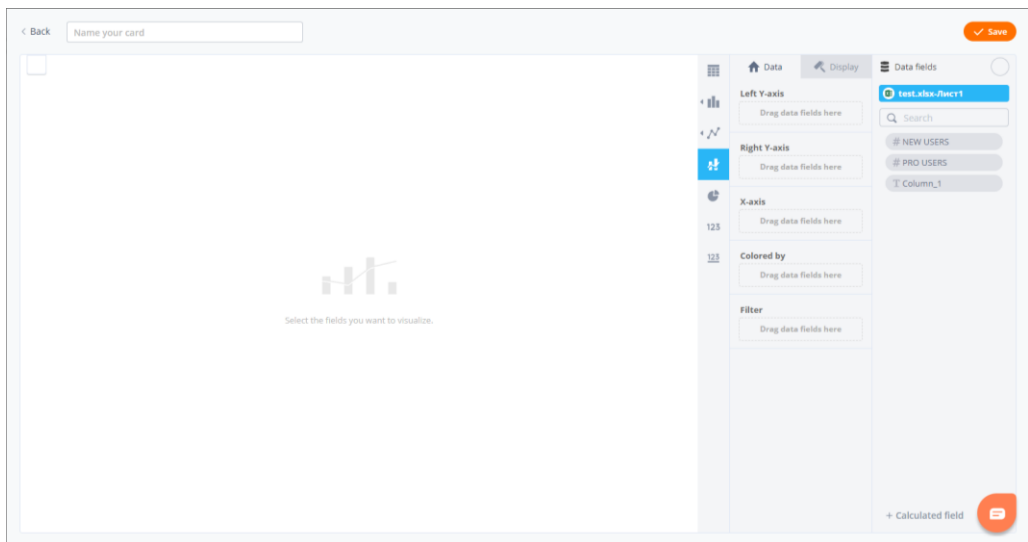


Рисунок 1.39 – Панель налаштування

Налаштувавши відображення даних, а саме перемістивши дані в правому меню з однієї колонки в іншу (рис. 1.40), отримано графік, який будується автоматично, як тільки дані додаються до нього (рис. 1.41).

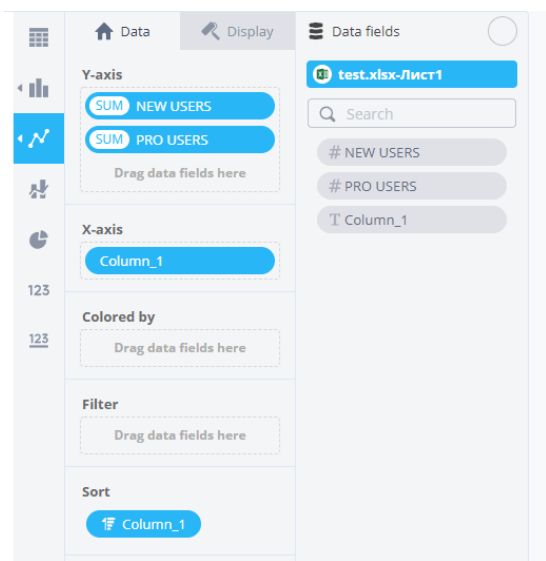


Рисунок 1.40 – Налаштування даних

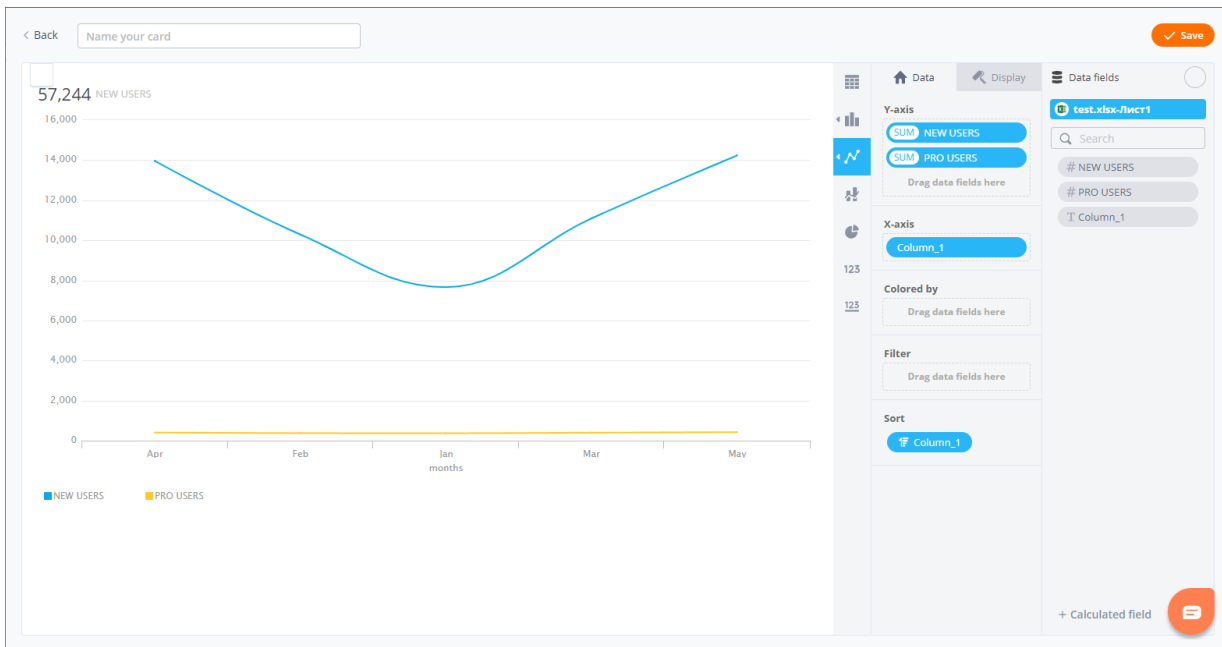


Рисунок 1.41 – Готовий графік

Також є можливість зробити кольорові налаштування в сусідній вкладці (рис. 1.42).

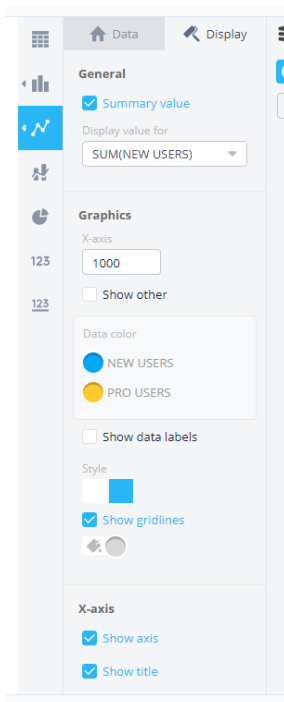


Рисунок 1.42 – Вкладка налаштувань

На даному ресурсі присутня функція збереження лише в форматі CSV (рис. 1.43), але є змога поділитися за допомогою адресного посилання.

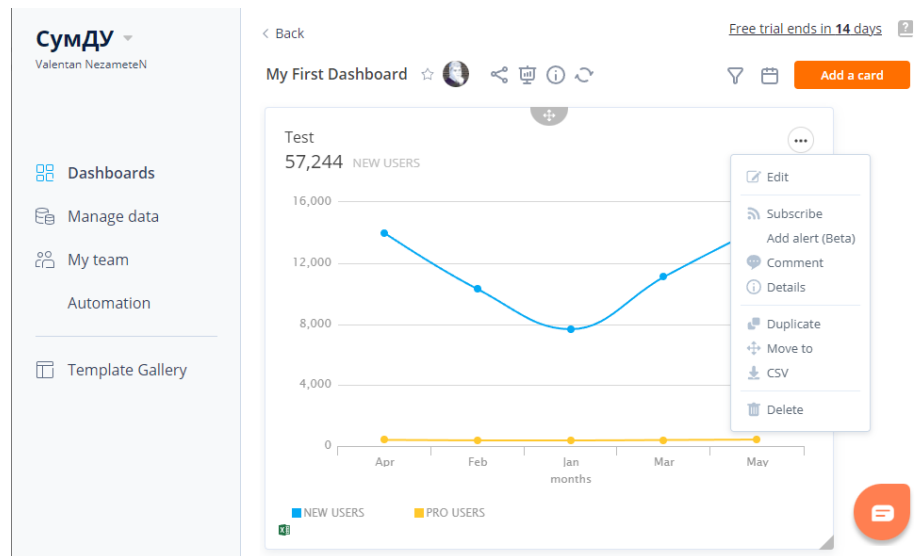


Рисунок 1.43 – Збереження графіка

В заключення можна додати, що даний ресурс схожий на перший, але без варіантів збереження, темної теми. Сподобалося, що є велика кількість баз даних, які підтримує даний сайт. Також є одна цікава особливість в тому, що тут є підтримка YouTube каналів, це дасть перевагу для аналітиків каналів.

Зробивши детальний аналіз аналогів, провести порівняння можна за такими критеріями:

- інтерфейс, на скільки зручно користуватися ресурсом;
- можливість зберігання даних;
- швидкодія на обробку запитів;
- детальність інформації на графіку;
- дизайн, сучасний дизайн грає важливу роль для утримання користувача;
- різноманітність візуалізації.

За представленими критеріями було створено табл. 1.2, яка дає змогу поглянути на порівняння ресурсів та їх переваги в представлених аспектах.

Таблиця 1.2 – Порівняльна характеристика web-додатків візуалізації

| Характеристика (Аспект) | infogram.com | chartblocks.com | datadeck.com |
|------------------------------|--------------|-----------------|--------------|
| Інтерфейс | + | - | + |
| Зберігання даних | + | + | + |
| Швидкодія | + | + | + |
| Детальність інформації | + | - | + |
| Дизайн | + | - | - |
| Різноманітність візуалізації | + | - | - |

За допомогою представлених характеристик зроблено висновок, що кожний web – додаток має свої переваги, але є критерій, що присутній на всіх ресурсах – це зберігання даних, кожен з них має свої формати зберігання. Всіма сайтами в більшій мірі зручно користуватися. Всі сайти мають швидку обробку даних. Але один сайт програє майже за всіма аспектами. А ось краще всього користуватися сайтом Infogram, зручний інтерфейс, велика кількість форматів збереження. Шкода, що мала доступність підтримуваних баз даних. Також велика різноманітність графіків та сторінок шаблонів. Добре, що існує велика кількість таких ресурсів, але краще створити свою систему, оскільки такий додаток дасть змогу повного налаштування відображення в наступних аспектах: змога налаштування візуалізації відображення, підключити та налаштувати свою базу даних, створити розмежування на користувачів та інше.

За даним висновком можна додати такі умови для розробки web – додатку:

- Зручність управління сайтом та його навігацією;
- Детальна інформативність на графіку;
- Збереження даних в форматі png.

2. ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ

2.1 Мета та задачі

Метою кваліфікаційної роботи магістра є розробка інформаційної технології з реалізацією web – інтерфейсу візуалізації даних для контролю за показниками системи підтримки прийняття рішень при управлінні мікромережами з альтернативними джерелами енергії. Даний проект нестиме цінність для кола користувачів, які мають в себе енергетичні мікромережі, а також мають бажання переглядати та контролювати їх використання. Це дасть змогу обробляти дані, що надійшли, та в короткий час візуалізувати їх у вигляді графіків та інших діаграм, і прийняти рішення, щодо подальшого їхнього використання.

Створений web – додаток повинен мати такі функціональні вимоги:

- інформаційна система має реалізовуватись у вигляді сайту;
- авторизація користувачів, з розмежуванням прав доступу;
- візуалізацію даних у вигляді графіків, діаграм та таблиць;
- можливість збереження інформації;
- навігація – відвідувач сайту повинен вільно орієнтуватися в структурі і навігації сайту;

На основі поставлених критеріїв за час роботи треба реалізувати такі вимоги:

- створити інформаційну систему у вигляді web – додатку;
- візуалізувати дані, що надходять з бази даних, у вигляді графіків, таблиць та діаграм;
- реалізувати збереження даних в форматі .png;
- реалізувати швидку роботу оновлення даних в системі;
- графічний модуль, що дасть змогу будувати різні типи графіків та їх налаштування, а саме вивід даних:

- побудова графіків з оновленням даних кожних 5 хв. з відображенням даних: по осі X час (день, місяць, чи тиждень), по осі Y показник використання електрики;
 - відображення таблиць з можливістю додавання та редагування даних з інформацією про: користувачів, енергетичні данні та інше;
 - відображення погодних даних як окремого віджита.
- модуль, що дасть змогу зробити авторизацію на сайті, з розмежуванням прав доступу:
 - користувач – має не повний доступ до системи, а саме має змогу переглянути данні та додати їх до бази;
 - адміністратор – має повний доступ до системи та її налаштувань, такі як: у користувача, а також змогу додавати та видаляти користувачів, видаляти інформацію з бази та інше;
- навігаційний модуль – за допомогою якого буде змога переміщатися сайтом.

2.2 Вибір технології реалізації інформаційної технології візуалізації даних

Метою кваліфікаційної роботи магістра є розробка інформаційної технології візуалізації даних для системи підтримки прийняття рішень при управлінні енергетичними мікромережами. Розробка програмного забезпечення – складний процес. Це вимагає виконання багатьох завдань, включаючи кодування, проектування та тестування. Програмні платформи полегшують життя розробникам, дозволяючи їм контролювати весь процес розробки програмного забезпечення або більшу його частину з однієї платформи. Саме через це розробники звертаються до фреймворків.

Вибір такої технології є важливою частиною в розробці, як і мова на якій буде розроблятися web – додаток. Розробка інформаційної системи було розділена на декілька етапів. Першим етапом відбудеться робота над front – end інтерфейсом за допомогою мови верстки HTML 5 та мови стилів CSS3. А для полегшення розробки буде використано фреймворк з поєднанням цих мов – Bootstrap 5 та додатково

Bootstrap 4, оскільки є елементи, які ще не перенесено до п'ятого [22,23]. Для анімації та різних переходів буде використовуватися JavaScript.

Другим етап відбудуватиметься розробка back – end. Для розробки цієї частини є багато фреймворків та API. Було вирішено, що система буде створена на чистому PHP, без використання різних фреймворків та додаткових API. Також додатково для обробки деяких моментів, таких-як авторизація користувачів використовуватиметься JavaScript.

Основними функціями проєкту буде відображення погодних умов та енергетичних показників. Так як енергетичні показники будуть надходити з приладів, з цим не виникне проблеми, а для відображення погоди треба використати API. Тому було проаналізовано найпопулярніші додатки: OpenWeatherMap, Weatherbit, AccuWeather [24-26]. Дані API було проаналізовано за наступними критеріями: дані прогнозу, історичні дані, погода в реальному часі, ціна, кількість безкоштовних запитів.

Всі порівняльні характеристики наведено в табл. 2.1.

Таблиця 2.1 – Таблиця порівняння API

| Характеристика | OpenWeatherMap | Weatherbit | AccuWeather |
|--------------------------------|----------------|---------------------------|---------------------------|
| Дані прогнозу | + | + | + |
| Історичні данні | + | - | - |
| Погода в реальному часі | + | + | + |
| Ціна | Безкоштовно | Безкоштовно з обмеженнями | Безкоштовна пробна версія |
| Кількість безкоштовних запитів | Необмежено | 150 в день | 50 в день |

Отже після проведеного аналізу для відображення погодних даних було обрано API OpenWeatherMap, яка має всі можливості, які потрібні для проєкту.

3 ПРОЕКТУВАННЯ WEB-ОРІЄНТОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ВІЗУАЛІЗАЦІЇ ДАНИХ ДЛЯ СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ

3.1 Структурно-функціональне моделювання візуалізації даних для системи підтримки прийняття рішень

Існує широкий спектр методів побудови діаграм процесів, таких як блок-схеми і діаграми потоків даних, але не всі з них є визначеними стандартами. Одним із методів, який належить до визначених стандартів є опис виконання процесів у нотації IDEF0 [24].

IDEF0 – це метод моделювання, заснований на комбінованій графіці і тексті, які представлені організованим і систематичним чином для отримання розуміння, підтримки аналізу, забезпечення логіки потенційних змін, визначення вимог або підтримки діяльності з проектування та інтеграції на рівні систем. Модель IDEF0 складається з ієрархічної серії діаграм, які поступово відображають все більш докладні рівні опису функцій і їх інтерфейсів в контексті систем. Існує три типи діаграм: графічна, текстова і глосарій. Графічні діаграми дозволяють визначити функції і функціональні взаємозв'язки за допомогою синтаксису і семантики полів і стрілок. Діаграми тексту і глосарію надають додаткову інформацію на підтримку графічних діаграм [27].

Функціональне моделювання web – сайту зображено у діаграмі IDEF0 (рис. 3.1).

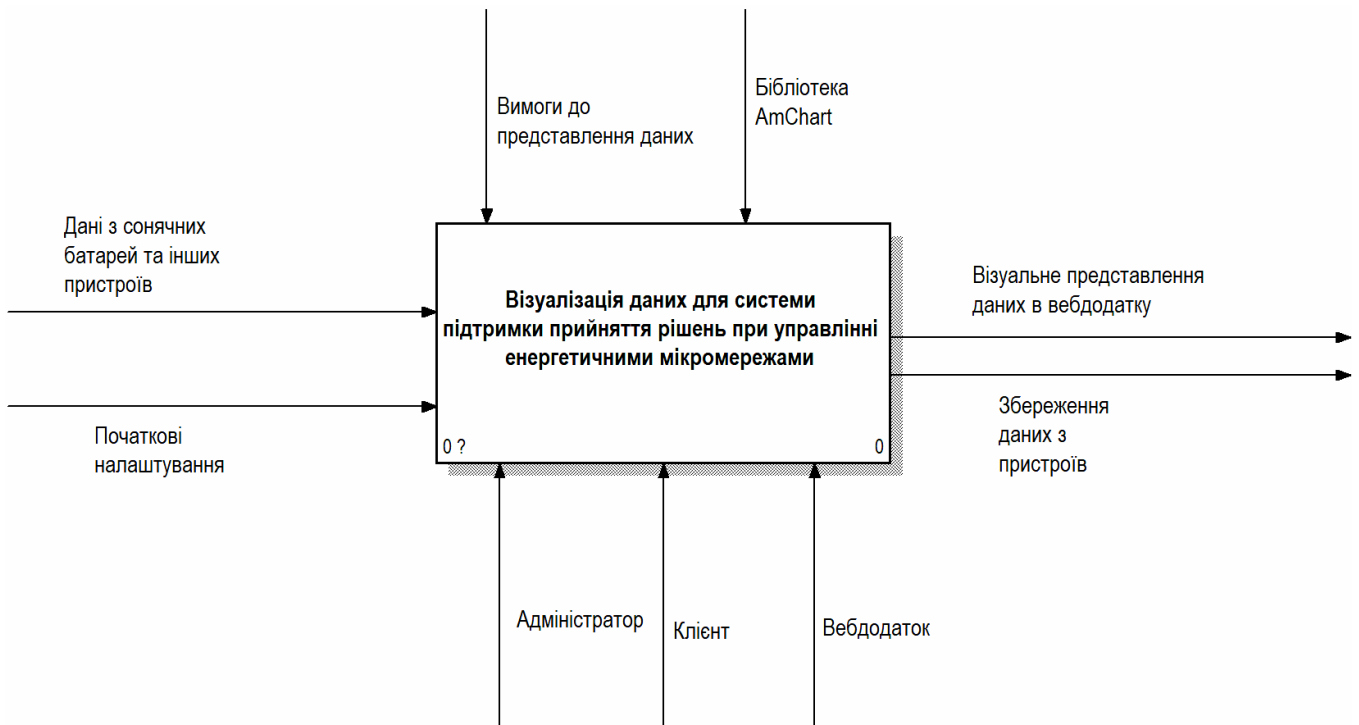


Рисунок 3.1 – Контекстна діаграма інформаційної технології візуалізації

Функціональна модель представленої інформаційної технології у нотації IDEF0 показана у вигляді діаграми декомпозиції (рис. 3.2).

У функціональній моделі визначено 5 етапів:

- авторизація;
- відображення поточних даних;
- запит до архівних даних;
- відображення архівних даних;
- редагування інформації;
- збереження даних.

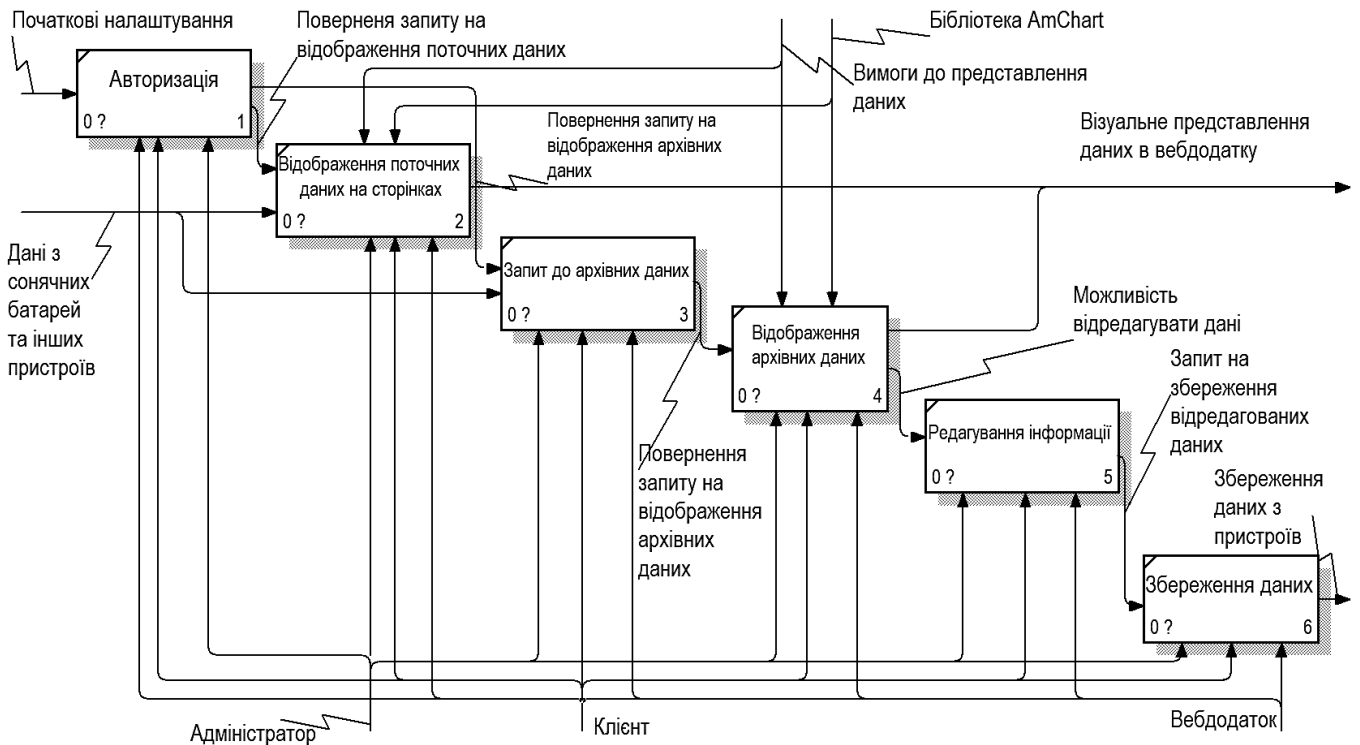


Рисунок 3.2 – Функціональна модель інформаційної технології візуалізації

Етап 1. Аутентифікація. На цьому етапі відбувається аутентифікація користувача. Після введення даних, відбувається порівняння з даними, що знаходяться в базі, і якщо данні збігаються, то клієнт потрапляє на головну сторінку адміністративної панелі.

Етап 2. Візуалізація оперативних даних. У web – інтерфейсі користувач може бачити поточну температуру, чи швидкість вітру або загальний стан окремих приладів, які потрібні користувачу та інше.

Етап 3. Запит до архівних даних. На цій сторінці інтерфейсу користувач може бачити всі архівні данні у вигляді таблиць, що знаходяться в базі даних. Та має можливість відправити запит на отримання окремого результату, наприклад: отримати координати де знаходиться прилад, чи всі данні про клієнта.

Етап 4. Відображення архівних даних. Тут клієнт може бачити данні на які він відправив запит до бази даних.

Етап 5. Редагування інформації. На даному етапі клієнт може відредагувати потрібну йому інформацію, наприклад: додати інформацію про нового клієнта.

Етап 6. Збереження даних. Даний етап дає можливість зберегти поточні данні, що відображуються на графіку, собі на пристрій в форматі png.

3.2 Моделювання варіантів використання інформаційної системи візуалізації даних

UML (Unified Modeling Language) – скорочення від уніфікованої мови моделювання, являє собою стандартизовану мову моделювання, що складається з інтегрованого набору діаграм, розроблених для допомоги розробникам систем і програмного забезпечення у визначенні, візуалізації, побудові та документуванні артефактів програмних систем.

UML являє собою набір кращих інженерних практик, які довели свою ефективність при моделюванні великих і складних систем. Вона є дуже важливою частиною розробки об'єктно-орієнтованого програмного забезпечення та процесу розробки програмного забезпечення. UML використовує в основному графічні позначення для вираження дизайну програмних проєктів [28].

Використання UML допомагає проєктним командам спілкуватися, вивчати потенційні проєкти та перевіряти архітектурний дизайн програмного забезпечення.

Актор – це поведінковий класифікатор, який визначає роль, яку відіграє зовнішній об'єкт, що взаємодіє з суб'єктом, користувач – людина розробленої системи, будь-яка інша система або апаратне забезпечення, що використовують послуги суб'єкта [29].

Актори:

Клієнт – людина, яка використовує web – додаток, де має змогу переглянути їй доступні данні, такі як, погода, дані з сонячної батареї та інше. Також має змогу переглянути архівні данні та додати або відредагувати данні.

Адміністратор – людина, що має повний доступ до сайту та бази даних. Може додавати та видаляти клієнтів, редагувати інформацію про всі прилади та інше.

Взаємодія акторів з web – додатком зображено на Рисунку 3.3.

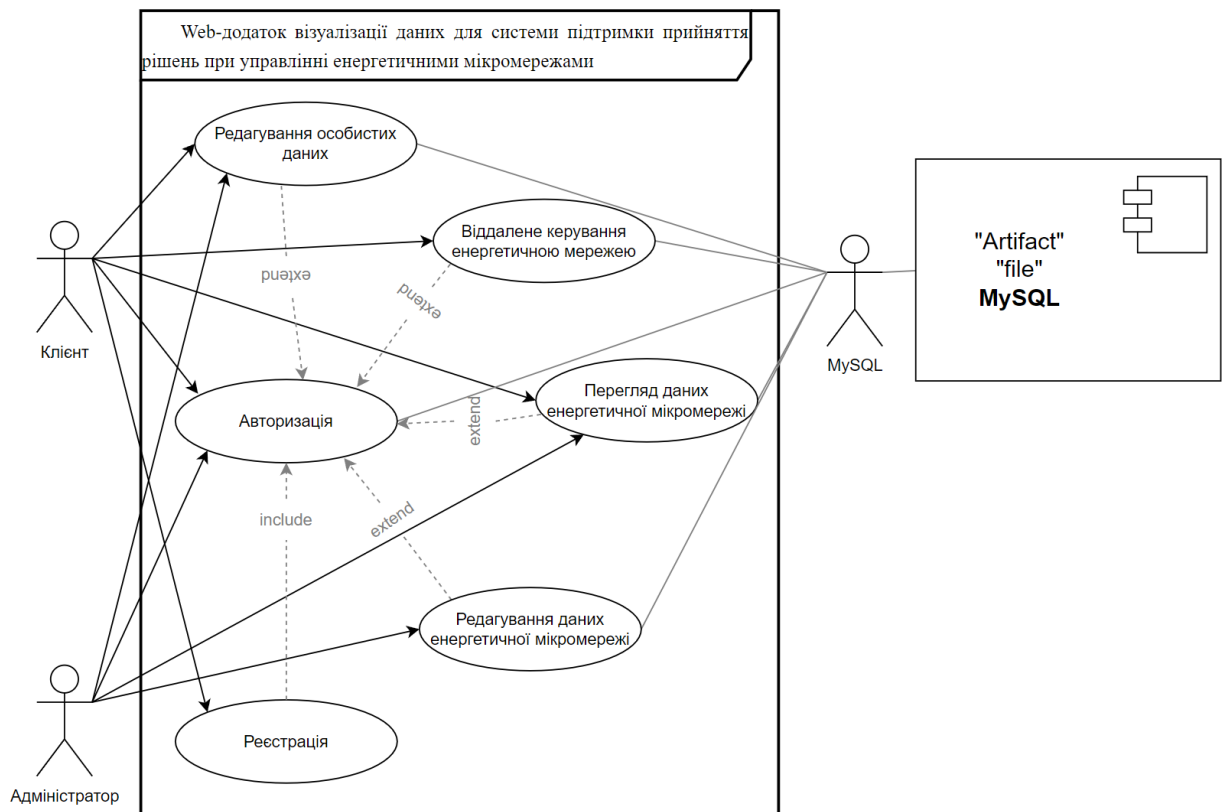


Рисунок 3.3 – Діаграма варіантів використання

Варіанти використання:

- ВВ Редагування особистих даних – дає змогу самостійно змінити чи додати особисті данні про клієнта або адміністратора в залежності від доступу;
- ВВ Авторизація – надає змогу авторизуватися в системі для подальшої роботи;
- ВВ Реєстрація – надає можливість клієнтові зареєструватися в системі;
- ВВ Віддалене керування енергетичною мережею – дає змогу включити або виключити окремий прилад в енергетичній мережі;
- ВВ Перегляд даних енергетичної мережі – користувач має змогу переглянути всі данні в енергетичній мікромережі;
- ВВ Редагування даних енергетичної мережі – в залежності від статусу доступу користувач має змогу відредагувати дані в мікромережі.

Діаграма варіантів використання UML є основною формою вимог до системи програмного забезпечення для нової недостатньо розробленої програми. Варіанти використання визначають очікувану поведінку, а не точний метод її реалізації.

Варіанти використання, одного разу зазначені, можуть бути позначені як текстовим, так і візуальним поданням, тобто схемою варіантів використання. Ключова концепція моделювання варіантів використання полягає в тому, що воно допомагає нам проектувати систему з точки зору кінцевого користувача. Це ефективний метод для передачі поведінки системи в термінах користувача шляхом вказівки всього зовнішнього видимого поведінки системи [30].

Для чіткого розуміння деяких моментів варіантів використання було створено три Use Case діаграми (рис. 3.4 – 3.6).

Use Case 1. Використання головної адміністративної сторінки:

Передумова: Відкрита сторінка авторизації акторами клієнт або адміністратор.

Процес: Після авторизації відбувається перехід на головну сторінку, де відображуються завчасно налаштовані данні

Післяумова: В додатку відкрилася головна сторінка з відображенням даних.

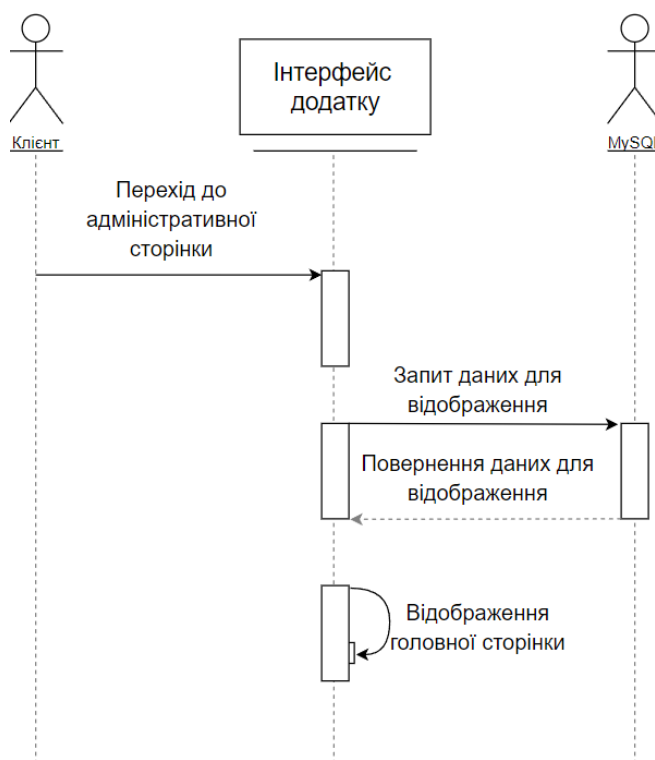


Рисунок 3.4 – Use Case 1 Використання головної адміністративної сторінки

Use Case 2. Використання сторінки сонячних панелей:

Передумова: Відкрита будь-яка сторінка акторами клієнт або адміністратор.

Процес: Натиснувши спеціальну кнопку в меню, відкриється сторінка з сонячними панелями.

Післяумова: В додатку відкрилася сторінка з інформацією про сонячні панелі або вітряки.



Рисунок 3.5 – Use Case 2 Використання сторінки сонячних панелей

Use Case 3. Сторінка з архівними даними:

Передумова: Відкрита адміністративна головна сторінка акторами клієнт або адміністратор.

Процес: Натиснувши спеціальну кнопку в меню, відкриється сторінка з даними бази.

Післяумова: В додатку відкрилася сторінка з даними в залежності від доступу актора.

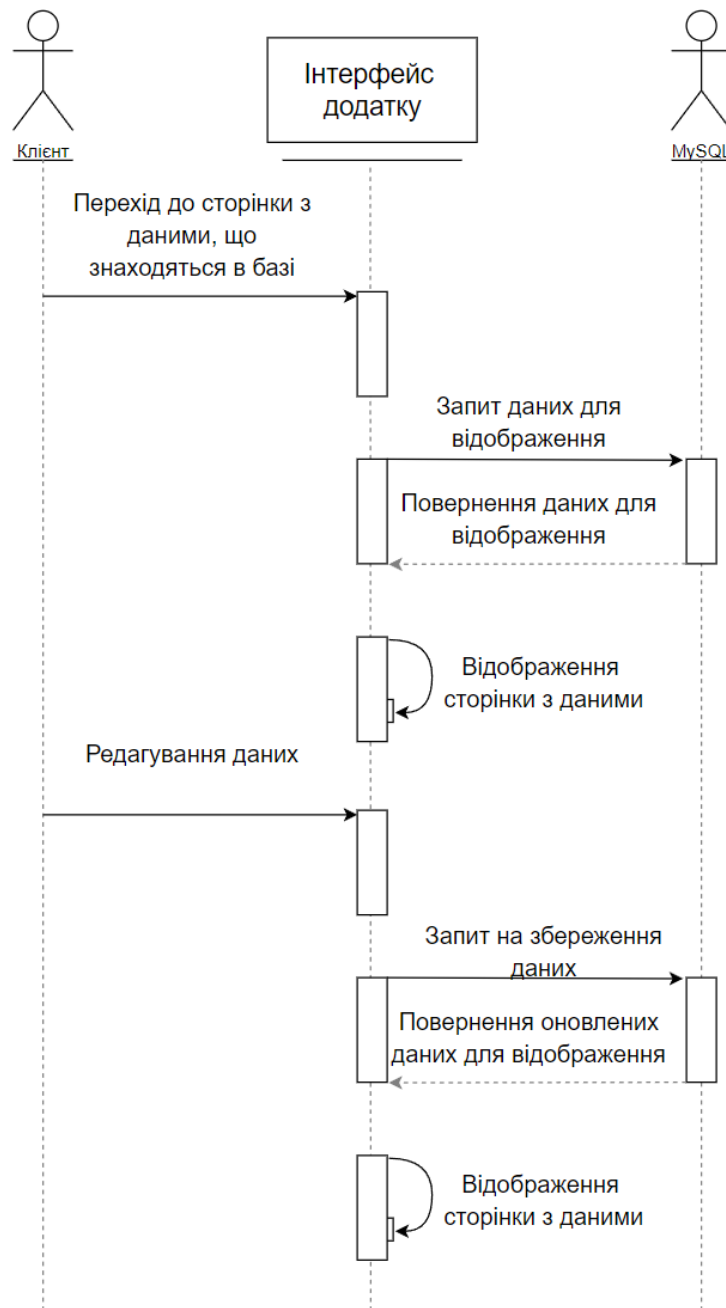


Рисунок 3.6 – Use Case 3 Сторінка з архівними даними

4 ПРАКТИЧНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ВІЗУАЛІЗАЦІЇ ДАНИХ

4.1 Архітектура інформаційної системи

На початку роботи було розроблено архітектуру web – додатку (рис. 4.1).

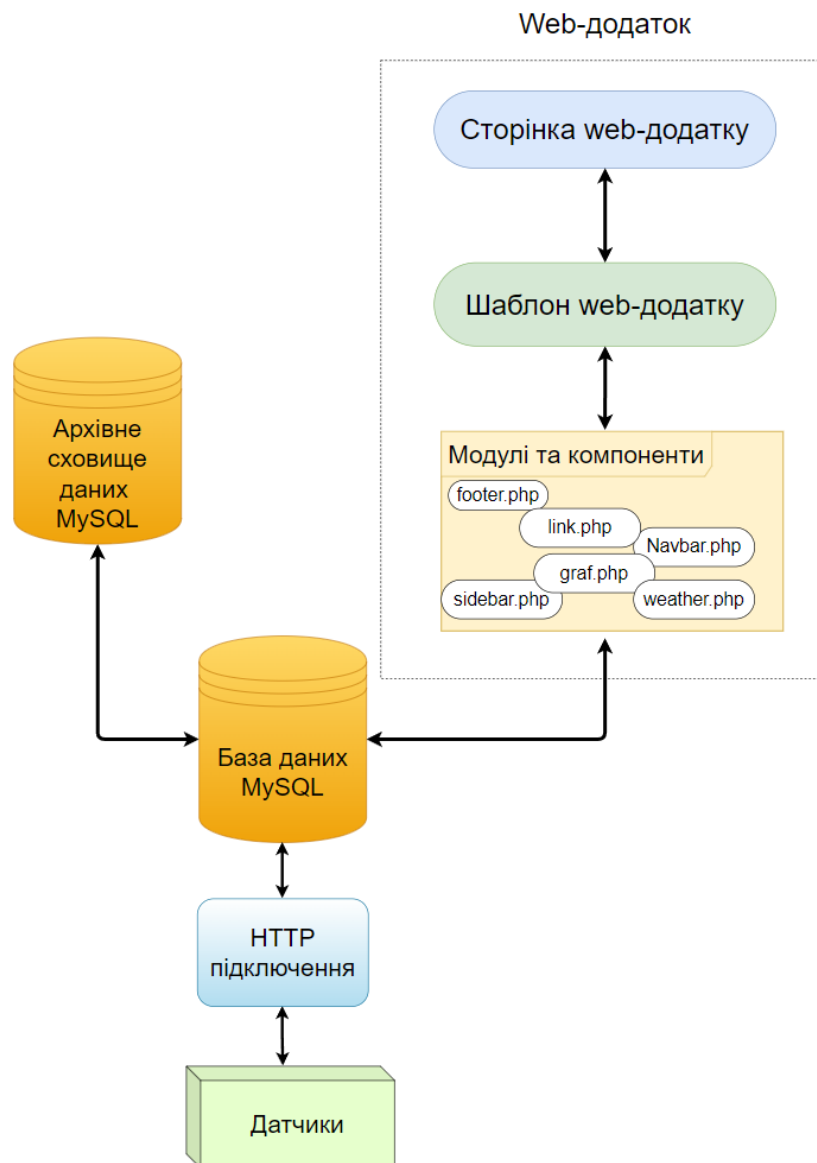


Рисунок 4.1 – Архітектура web-додатку візуалізації даних.

Даний web-додаток розроблений за допомогою компонентів та модулів, які наведено в табл. 4.1.

Таблиця 4.1 – Призначення модулів та компонентів

| № | Компоненти та модулі | Призначення |
|---|----------------------|---|
| 1 | link.php | Знаходяться всі зовнішні посилання, що використовуються в web – додатку |
| 2 | grafscript.php | В модулі знаходяться налаштування для графіків |
| 3 | navbar.php | Модуль навігаційного меню |
| 4 | weather.php | Налаштування API погоди, які використовується в web – додатку |
| 5 | footer.php | Модуль “підвалу” де знаходиться контактна інформація |
| 6 | include_database.php | Модуль підключення до бази даних |
| 7 | config.php | Компонент для налаштування додатку |

Дані, надходять до додатку з бази даних. Інформація потрапляє з API OpenWeatherMap та приладів, за допомогою протоколу HTTP. Web – додаток створено за допомогою модулів, які з’єднані між собою та утворюють повноцінні шаблони сторінок, та з’єднуються в повноцінний web – додаток інформаційної системи візуалізації даних.

4.2 Реалізація web-додатку

На етапі реалізації web – додатку було використано наступні модулі на стороні клієнта:

- HTML 5;
- CSS 3;

- Bootstrap 4;
- Bootstrap 5;
- Plotly.js;
- AmChart.js;

Для частини серверної реалізації було обрано базу збереження даних MySQL.

Розробка інформаційної системи відбувалася за принципом модульної розробки [31]. Першим модулем, який було зроблено на початку – модуль підключення до бази даних та файл конфігурації, для швидкої зміни параметрів підключення. Далі розпочалася робота над створенням модуля навігації, який винесено в окремий файл для оперативної зміни шляхів сторінок, під час чого зміниться і на всіх подальших сторінках (рис. 4.2).



 Головна  Про нас  Кабінет

Рисунок 4.2 – Модуль навігації

З представленого рисунку можна побачити, що зліва знаходиться логотип СумДУ та меню:

- Головна – перехід до головної сторінки;
- Про нас – сторінка з додатковою інформацією про компанію;
- Кабінет – пункт меню для переходу в адміністративну панель.

Для завершення шаблону було розроблено нижню частину, де знаходяться контакти компанії та інформація про самого розробника, детально можна поглянути на рис. 4.3.

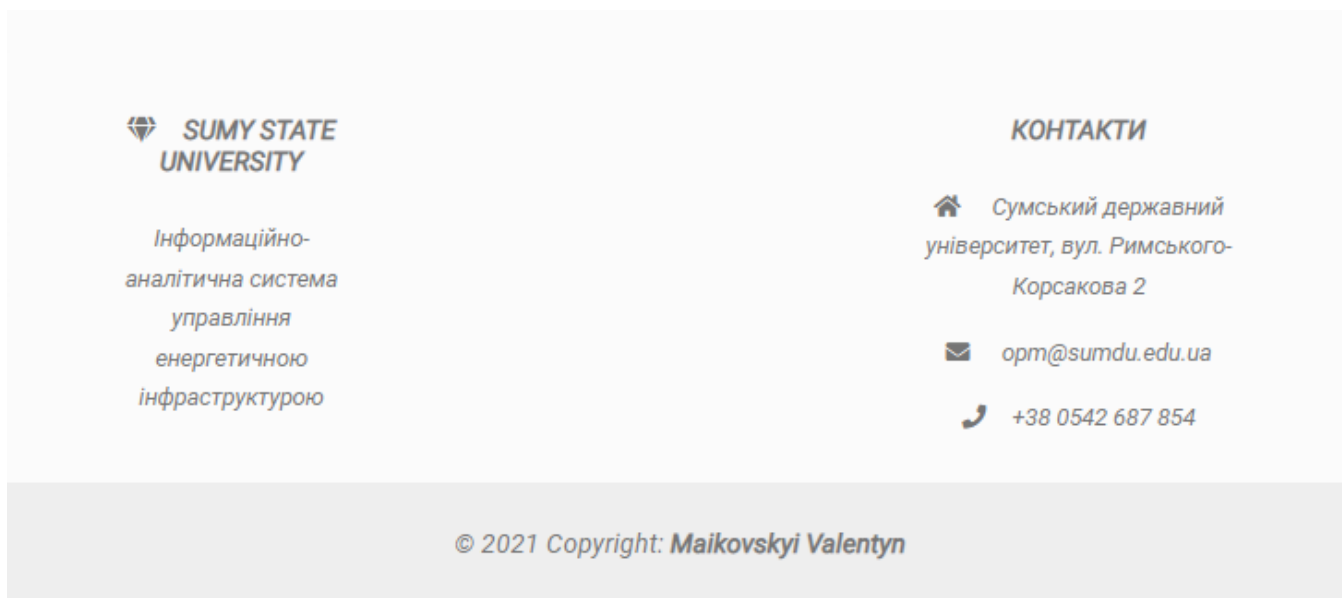


Рисунок 4.3 – Модуль підвалу

Після розробки ключових модулів додатково створено модуль контейнера, де знаходиться потрібна інформація. Для головної сторінки в модуль контейнера було розташовано слайдер з картинками для приваблення клієнтів, що зображено на рис. 4.4, який переключається автоматично, а також має змогу ручного перемикання.



Рисунок 4.4 – Модуль слайдера

Розробивши фінальний модуль, було перейдено до етапу з'єднання в єдине ціле за допомогою функції `require_once __DIR__` в мові програмування PHP [32], чка дозволяє підключати модулі, з перевіркою чи підключено даний файл, і чи не потрібно підключати його знову, тим самим дає змогу одразу виявити помилки в

роботі будь – якого модуля. Підключивши всі модулі, було отримано головну сторінку, яка буде з’являтися під час запуску сайту.

Наступним кроком було створення сторінки “Про нас”, де знаходиться інформація про систему та компанію в цілому.

Після розроблення представлених сторінок було розпочато роботу над модулем авторизації та самої сторінки авторизації з використання головних модулів: меню та підвал.

В модулі авторизації була розроблена можливість для клієнтів зареєструватися в системі, де потрібно ввести певні дані, після чого підтвердити їх введення, натиснувши на певну кнопку, фрагмент використаного коду на додавання в базу даних під час реєстрації користувача можна переглянути на рис. 4.5.

```
$res = $conn->query("select id from customer where (login = '{$login}' OR email = '{$login}')");
if ($res->num_rows === 0) {
    $query = $conn->query("insert into customer set name = '{$name}',
        surname = '{$surname}', login='{$login}', password = '{$password}',
        email = '{$login}', phone = '{$phone}'");
} else {
    $msg_e = "Користувач вже зареєстрований";
}
```

Рисунок 4.5 – Фрагмент коду додавання користувача

Після реєстрації сторінка оновиться та з’явиться форма авторизації, де потрібно ввести логін у вигляді Email і пароль.

Завершивши сторінку авторизації, перейдено до розробки адміністративної панелі. Першим кроком створено нове навігаційне меню спеціально для панелі (рис 4.6).

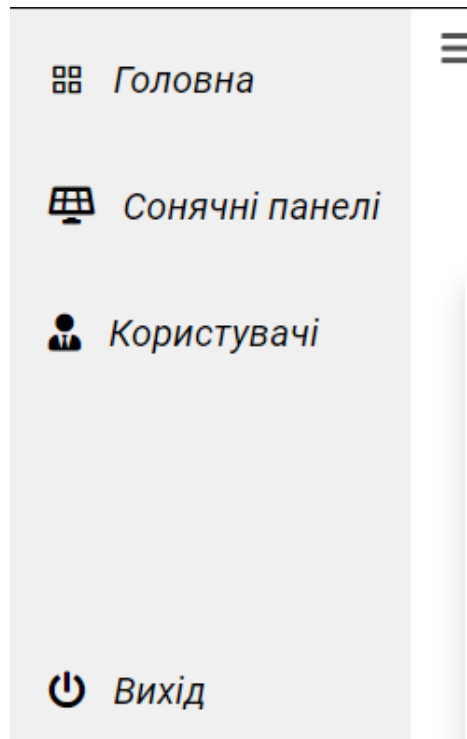


Рисунок 4.6 – Меню адміністративної сторінки

В представленому меню знаходиться декілька пунктів:

- Головна – головна сторінка адміністративної панелі;
- Сонячні панелі – знаходиться графічна інформація про сонячні панелі та можливості їх контролю;
- Користувачі – знаходяться данні про користувачів, що зареєстровані в системі.

Наступним кроком створено основний контейнер панелі, де було створено інформаційні панелі, які можуть знадобитися користувачу.

Після цього було розроблено основну погодну панель, де знаходиться загальна інформація в цілому (рис 4.7).

Сумы

пасмурно

П'ятниця

Грудень 10, 2021

1:03 AM

Температура**4.57**

Швидкість вітру

3.36 Km/h

Хмарність

100 %

Вологість

95 %

Тиск

1020 hPa

Час сходу

08:32:36

Час заходу

16:34:51

Рисунок 4.7 – Погодна інформаційна панель

На представленій панелі можна роздивитися інформацію про місто та погодні умови в ньому. Всі дані надходять з API OpenWeatherMap через звернення погодного модулю. Код реалізації модуля представлено в додатку В.

Додатково створено панель температури, де відображується поточна температура в пульсуючому крузі, та температура, яка відображує, як вона відчувається організмом людини.

На основі OpenWeatherMap та бібліотеки для побудови графіків та інших графічних показників AmChart [33], було розроблено дві панелі, на першій панелі відображується компас з позначками та текстово – числова інформація про кут вітру (рис 4.8). Де також реалізовано можливість, за потреби збереження даного компаса у форматі png, jpg, svg, pdf.

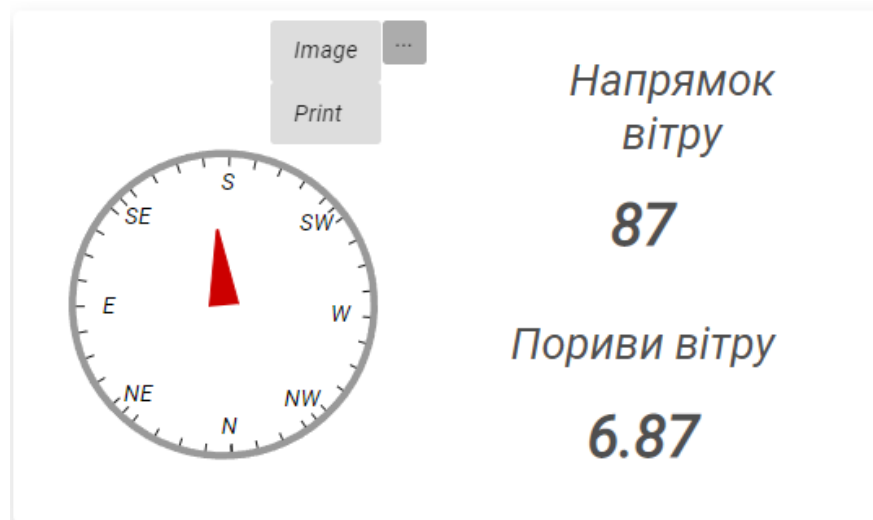


Рисунок 4.8 – Інформаційна панель про стан вітру

Фрагмент коду реалізації наведено на рис. 4.9, де детально наведено налаштування полотна та меню збереження.

```
// налаштування теми
am4core.useTheme(am4themes_animated);

// Створення компасу
var chart = am4core.create("chartdiv", am4charts.GaugeChart);
chart.exporting.menu = new am4core.ExportMenu();
chart.hiddenState.properties.opacity = 0;

chart.startAngle = 0;
chart.endAngle = 350;

var axis = chart.xAxes.push(new am4charts.ValueAxis());
axis.min = 0;
axis.max = 350;
axis.renderer.line.strokeWidth = 5;
axis.renderer.line.strokeOpacity = 1;
axis.renderer.line.stroke = am4core.color("#999");
axis.renderer.inside = true;
axis.renderer.axisFills.template.disabled = true;
axis.renderer.grid.template.disabled = true;
axis.renderer.ticks.template.disabled = false
axis.renderer.ticks.template.length = 8;
axis.renderer.ticks.template.strokeOpacity = 1;
axis.renderer.labels.template.radius = 30;
axis.renderer.labels.template.disabled = true;
axis.renderer.ticks.template.disabled = true;

function createLabel(label, deg) {
    var range = axis.axisRanges.create();
    range.value = deg;
    range.grid.disabled = true;
    range.label.text = label;
}
}
```

Рисунок 4.9 – Налаштування компасу

Другу панель розроблено також з використанням бібліотеки AmChart для графічного відображення атмосферного тиску.

Також створено панель карти, за допомогою Google Maps де відображується місце встановлення приладу та календар для поточного перегляду дати.

Після створення інформаційних панелей було створено таблиці з архівними даними. На рис. 4.10 представлено таблицю з архівними показниками погоди.

Архів погоди

| Країна | Місто | Широта | Довгота | Часовий пояс | Дата | Час | Швидкість вітру | Напрямок вітру | Хмарність | Температура | Час сходу | Час заходу |
|---------|-------|---------|---------|--------------|-----------|----------|-----------------|----------------|-----------|-------------|-----------|------------|
| Україна | Суми | 50.9216 | 34.8003 | 2 | 6.12.2021 | 16:05:21 | 5 | 120 | 100 | 0.3 | 08:28:07 | 16:35:53 |
| Україна | Суми | 50.9216 | 34.8003 | 2 | 6.12.2021 | 18:50:41 | 6 | 100 | 30 | 0.4 | 08:28:07 | 16:35:53 |
| Україна | Суми | 50.9216 | 34.8003 | 2 | 6.12.2021 | 18:50:41 | 6 | 100 | 30 | 0.4 | 08:28:07 | 16:35:53 |
| Україна | Суми | 50.9216 | 34.8003 | 2 | 6.12.2021 | 19:13:54 | 5.52 | 136 | 0 | 1.1 | 08:28:07 | 16:35:53 |
| Україна | Суми | 50.9216 | 34.8003 | 2 | 6.12.2021 | 19:14:10 | 5.52 | 136 | 0 | 1.1 | 08:28:07 | 16:35:53 |
| Україна | Суми | 50.9216 | 34.8003 | 2 | 6.12.2021 | 19:14:26 | 5.52 | 136 | 0 | 1.1 | 08:28:07 | 16:35:53 |
| Україна | Суми | 50.9216 | 34.8003 | 2 | 6.12.2021 | 19:14:42 | 5.52 | 136 | 0 | 1.1 | 08:28:07 | 16:35:53 |
| Україна | Суми | 50.9216 | 34.8003 | 2 | 6.12.2021 | 19:14:58 | 5.52 | 136 | 0 | 1.1 | 08:28:07 | 16:35:53 |

Сторінок: 1 2 3 4 5 ... Вперед Останній 1 з 22

Рисунок 4.10 – Архів погоди

Результат запиту на звернення до бази для отримання архівних показників погоди представлено на рис. 4.11.

```
$result = mysqli_query($conn, "SELECT country.name, city.name, city.latitude, city.longitude, city.time_zone, datetimeinfo.day, datetimeinfo.month, datetimeinfo.year, actualweather.wind_speed, actualweather.wind_deg, actualweather.clouds, actualweather.temperature, actualweather.sunrise, actualweather.sunset FROM actualweather INNER JOIN datetimeinfo ON actualweather.date_time_id = datetimeinfo.id INNER JOIN city ON actualweather.city_id = city.id INNER JOIN country ON city.country_id = country.id ORDER BY {$sort_sql}");
```

Рисунок 4.11 – Реалізація запиту

Для створення повноцінних графіків було використано бібліотеку Plotly.js, яка дає змогу чіткого та гнучкого налаштування. Таким чином було створено графіки які відображують стан панелей з показниками фаз сонячних панелей (рис. 4.12).

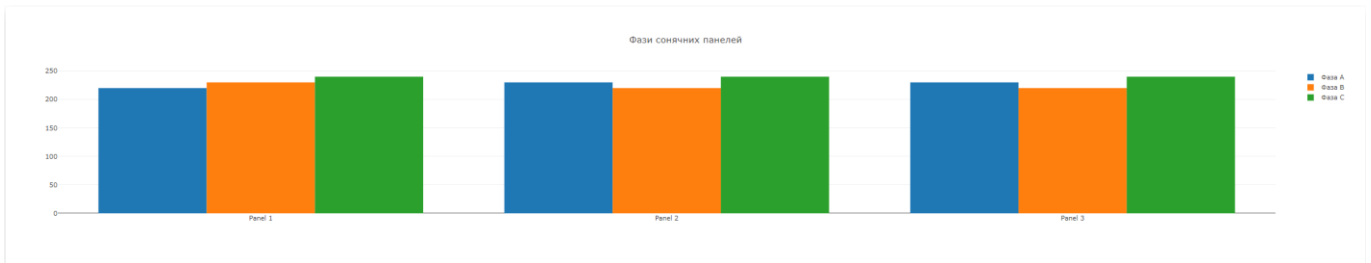


Рисунок 4.12 – Графік фаз сонячних панелей

Код реалізації раніше наведеного графіку представлено на рис. 4.13.

```
<script type="text/javascript">
  var trace3 = {
    x: ['<?php echo $namepanel ?>', '<?php echo $namepanel1 ?>', '<?php echo $namepanel2 ?>'],
    y: ['<?php echo $UApanel ?>', '<?php echo $UApanel1 ?>', '<?php echo $UApanel2 ?>'],
    name: 'Фаза А',
    type: 'bar'
  };
  var trace4 = {
    x: ['<?php echo $namepanel ?>', '<?php echo $namepanel1 ?>', '<?php echo $namepanel2 ?>'],
    y: ['<?php echo $UBpanel ?>', '<?php echo $UBpanel1 ?>', '<?php echo $UBpanel2 ?>'],
    name: 'Фаза В ',
    type: 'bar'
  };
  var trace5 = {
    x: ['<?php echo $namepanel ?>', '<?php echo $namepanel1 ?>', '<?php echo $namepanel2 ?>'],
    y: ['<?php echo $UCpanel ?>', '<?php echo $UCpanel1 ?>', '<?php echo $UCpanel2 ?>'],
    name: 'Фаза С ',
    type: 'bar'
  };
  var data = [trace3, trace4, trace5];
  var layout = {
    title: 'Фази сонячних панелей',
    barmode: 'group'
  };
  Plotly.newPlot('myDiv1', data, layout);
</script>
```

Рисунок 4.13 – Код графіку фаз сонячних панелей

Для чіткого відображення даних запиту до бази даних використано функцію ROUND() для округлення результату, а так, як ні бібліотека AmChart, ні Plotly.js [34] не підтримують тип даних float в стовпчикових діаграмах, було використано функцію CAST(), для приведення типу в доступний для них формат. Результат запиту представлено на рис. 4.14.

```
$resultbat = mysqli_query($conn, "SELECT location.name, CAST(ROUND(locationdata.UA,0) as int),
CAST(ROUND(locationdata.UB,0) as int), CAST(ROUND(locationdata.UC,0) as int),
CAST(ROUND(locationdata.UAB,0) as int), CAST(ROUND(locationdata.UAC,0) as int),
CAST(ROUND(locationdata.UBC,0) as int) FROM locationdata INNER JOIN location ON
locationdata.location_id = location.id");
```

Рисунок 4.14 – реалізація запиту

Всі дані, що відображуються в таблицях та графіках, повністю отримуються з бази даних у вигляді масивів, а вже потім заносяться до потрібних змінних.

Після створення додаткового графіку та таблиці на головну адміністративну панель, роботу над головною сторінкою закінчено в результаті чого отримано головну сторінку у вигляді Landing Page.

Наступним кроком було створення сторінки сонячних панелей, де можна переглянути інформацію про кожну панель окрема і змінити стан роботи чи стан заряду акумуляторів підключених до панелей. Код реалізації зміни стану сонячної батареї та стану акумулятору наведено на рис. 4.15 разом з запитом на отримання результату з бази даних.

```
if (isset($_POST['stan'])) {
    $id = $_POST['stan'];

    $res = mysqli_query($conn, "SELECT convector FROM accumulator WHERE id = '$id'");
    $row = mysqli_fetch_array($res);
    $stan = $row[0];

    if ($stan == 0) {
        mysqli_query($conn, "UPDATE accumulator SET convector = 1 WHERE id = '$id'");
    } else if ($stan == 1) {
        mysqli_query($conn, "UPDATE accumulator SET convector = 0 WHERE id = '$id'");
    }
}

if (isset($_POST['charge'])) {
    $id = $_POST['charge'];

    $res1 = mysqli_query($conn, "SELECT charge, discharge FROM accumulator WHERE id = '$id'");
    $row1 = mysqli_fetch_array($res1);
    $charge = $row1[0];
    $discharge = $row1[1];

    if ($charge == 1 && $discharge == 0) {
        mysqli_query($conn, "UPDATE accumulator SET charge = 0, discharge = 1 WHERE id = '$id'");
    } else if ($charge == 0 && $discharge == 1) {
        mysqli_query($conn, "UPDATE accumulator SET charge = 1, discharge = 0 WHERE id = '$id'");
    }
}
```

Рисунок 4.15 – Код реалізації зміни стану

В кінці було розроблено сторінку користувачів, де в залежності від прав доступу є змога додати, видалити або змінити інформацію про користувача. Реалізовано дану сторінку у вигляді двох таблиць. Сторінку користувачів представлено на рис. 4.16.

| # | Ім'я | Прізвище | Логін | Пароль | Пошта | Телефон |
|---|----------|------------|----------------|----------------------------------|----------------|-----------------|
| 1 | Aleks | Markovets | Cas | e8e5ada83b2402b9757781980d6bfc7e | fas@gmail.com | 380663357890 |
| 2 | Toto | Kotr | kot | e8e5ada83b2402b9757781980d6bfc7e | kot@gmail.com | 380663357890 |
| 3 | Valentan | | maik@gmail.com | d733c2305e7f62063ec4ea2b44a13161 | maik@gmail.com | 380666257545 |
| 4 | Валентин | Майковский | mai@gmail.com | abec21533a50d9068cfe0f515caf6e7a | mai@gmail.com | 380656512454545 |

| # | Ім'я | Прізвище | Логін | Пароль | Пошта | Телефон | Роль | + |
|----|----------|------------|----------|--|------------------|-------------|-------|---|
| 14 | Валентин | Майковский | Valentan | 242e3d3aff1912371c023304203a9641 | Maikov@gmail.com | +3806698457 | Читач | |
| 21 | Олег | Марков | Omar | 512e3d3aff18964171c089455424203a9641 | Omar@gmail.com | +3806689124 | Читач | |
| 22 | Вікторія | Любовець | Vilub | 9e1e3d3aff18564112371c087414778963203a9785 | ghkghkhghkghg@ | +3809738521 | Читач | |

Рисунок 4.16 – Сторінка користувачів

Фрагмент коду на розмежування ролей зображено на рис. 4.17. Отримання ролей для подальшого їх виділення для прав доступу представлено на рис. 4.18.

```
$customers = json_encode((new CustomerRepository($conn))->getAllNoFilter());
$roles = json_encode((new RoleRepository($conn))->getAll());
$conn->close();

$customer_edit = $_SESSION['type'] === "admin";
$user_edit = $_SESSION['type'] === "customer";
```

Рисунок 4.17 – Фрагмент коду розмежування прав доступу

```
<?php

if (!isset($_SESSION)) {
    session_start();
}

include "Role.php";

class RoleRepository
{
    protected $db;

    public function __construct(MYSQLI $db)
    {
        $this->db = $db;
    }

    public function getAll()
    {
        $sql = "SELECT * FROM userroles";
        $q = $this->db->query($sql);
        $rows = $q->fetch_all(MYSQLI_ASSOC);

        $result = array();
        foreach ($rows as $row) {
            $result[] = $this->read($row);
        }
        return $result;
    }

    public function read($row)
    {
        $result = new Role();
        $result->id = $row['id'];
        $result->role_title = $row['role_title'];
        return $result;
    }

    public function getById($id)
    {
        $sql = "SELECT * FROM userroles WHERE id = '{$id}'";
        $q = $this->db->query($sql);
        $rows = $q->fetch_all(MYSQLI_ASSOC);
        return $this->read($rows[0]);
    }
}
```

Рисунок 4.18 – Код на отримання ролей

4.3 Приклад використання web-додатку системи

Ввівши адресу сайту в адресному рядку, користувач потрапляє на головну сторінку сайту (рис. 4.19), де він має змогу визначитися, чи він бажає переглянути інформацію про компанію, чи перейти до адміністративної сторінки (рис. 4.20).

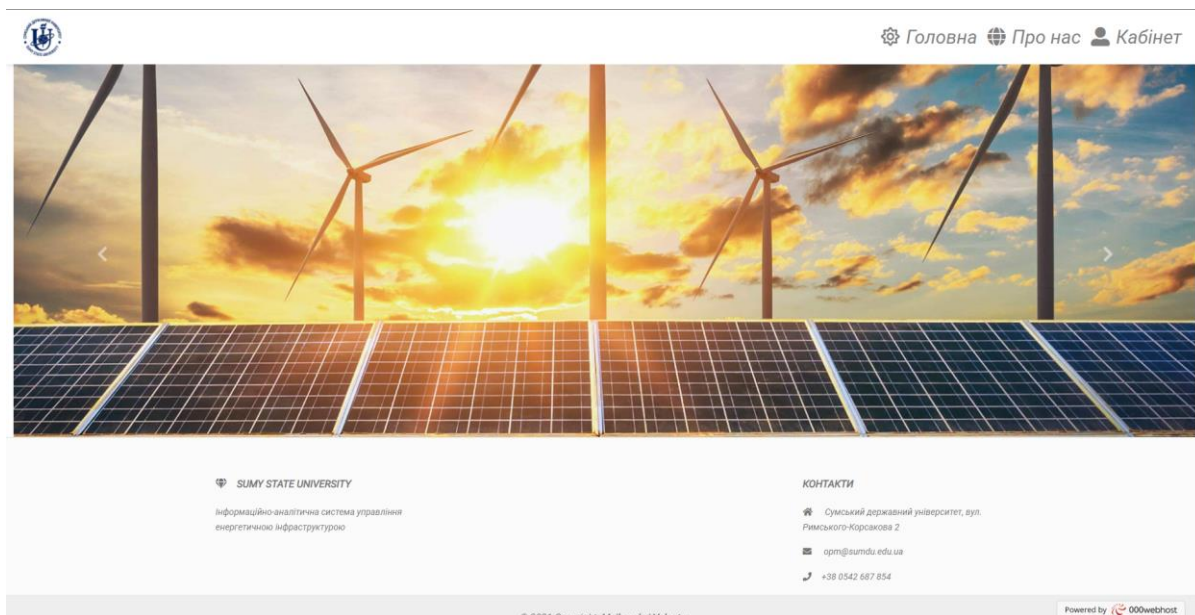


Рисунок 4.19 – Головна сторінка сайту



Рисунок 4.20 – Інформаційна сторінка

Обравши перехід до кабінету, користувач повинен ввести свої дані та підтвердити вхід, натиснувши на кнопку (рис. 4.21).

Рисунок 4.21 – Сторінка авторизації

Якщо клієнт не зареєстрований в системі, йому пропонується зареєструватися в ній, натиснувши на стрілку на формою авторизації, і перед ним з'явиться форма реєстрації (рис. 4.22). Після введення і підтвердження даних система його знову поверне на форму авторизації.

Рисунок 4.22 – Сторінка реєстрації

Підтвердивши всі дані, користувач потрапляє на головну адміністративну сторінку (рис. 4.23), де одразу має змогу переглянути поточні показники погоди (рис. 4.24).

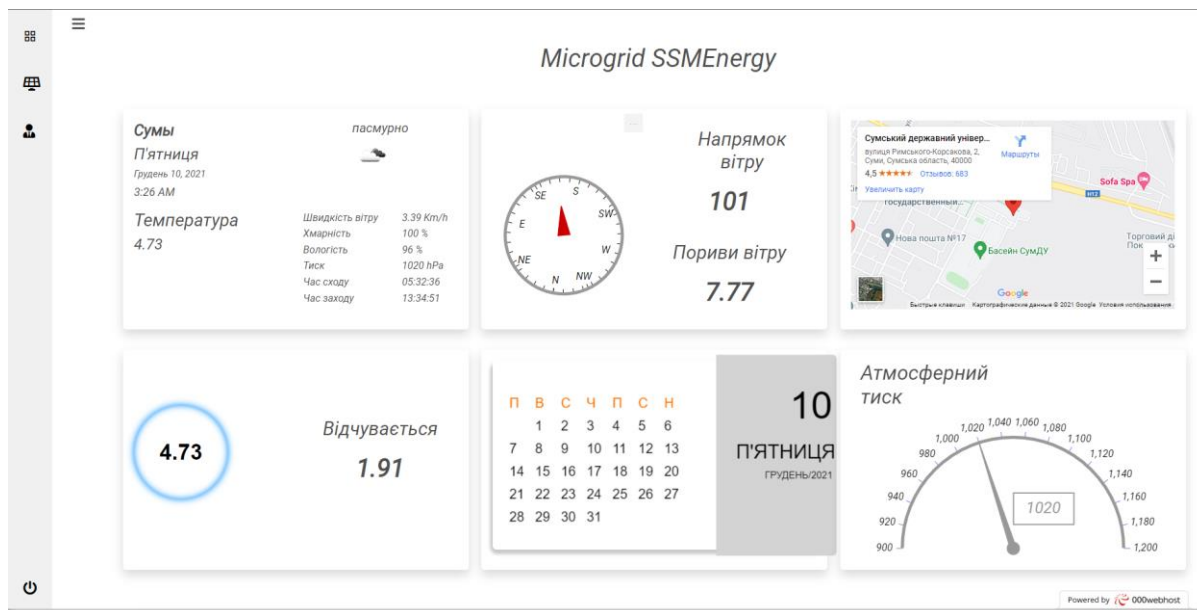


Рисунок 4.23 – Головна адміністративна сторінка

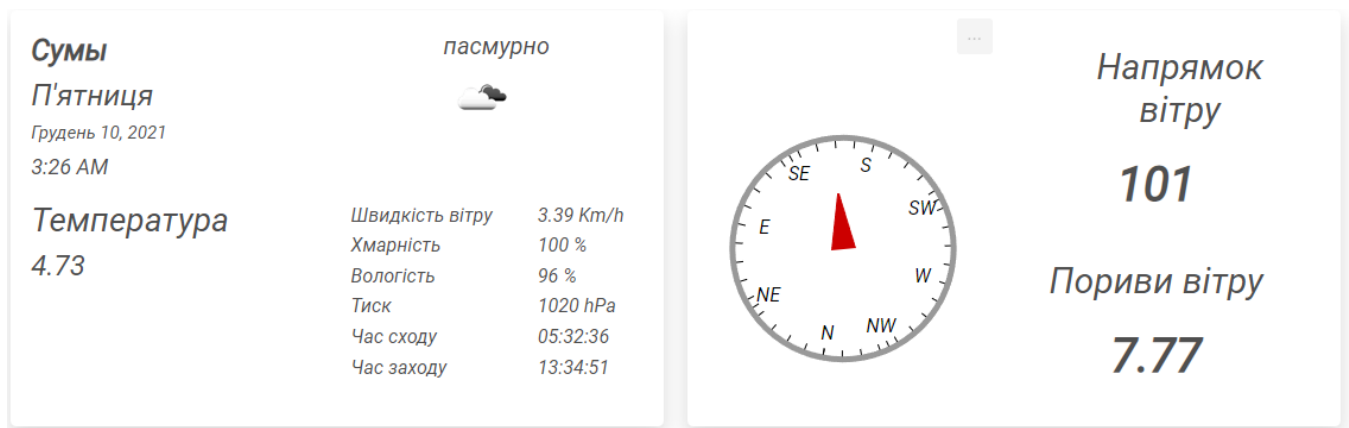


Рисунок 4.24 – Показники погодних умов

Опустившись нижче від погодних показників, користувач має змогу переглянути архівну інформацію щодо показників сонячних батарей (рис. 4.25).

Архівні показники

| # | Генерація (кВт) | Споживання (кВт) | Продаж (кВт) | Локація | Дата | Час | Фаза А (Вт) | Фаза В (Вт) | Фаза С (Вт) | Міжфазна АВ (Вт) | Міжфазна АС (Вт) | Міжфазна ВС (Вт) | Поточна потужність (Вт) | Балансе навантаження | Некретичне навантаження | Лчильник електроенергі |
|---|-----------------|------------------|--------------|---------|-----------|----------|-------------|-------------|-------------|------------------|------------------|------------------|-------------------------|----------------------|-------------------------|------------------------|
| 1 | 100 | 50 | 3 | 3 | 6.12.2021 | 19:18:29 | 230 | 220 | 240 | 380 | 370 | 350 | 220 | 1 | 1 | 1 |
| 2 | 80 | 60 | 4 | 3 | 6.12.2021 | 19:18:25 | 230 | 220 | 240 | 340 | 350 | 310 | 218 | 1 | 0 | 1 |
| 3 | 60 | 50 | 3 | 1 | 6.12.2021 | 19:17:39 | 220 | 230 | 240 | 350 | 340 | 380 | 222 | 0 | 1 | 0 |
| 4 | 40 | 40 | 2 | 2 | 6.12.2021 | 19:16:50 | 230 | 220 | 240 | 330 | 330 | 390 | 215 | 1 | 0 | 0 |
| 5 | 110 | 30 | 2 | 3 | 6.12.2021 | 19:17:06 | 230 | 220 | 240 | 380 | 390 | 330 | 222 | 0 | 0 | 1 |
| 6 | 150 | 11 | 5 | 1 | 6.12.2021 | 19:16:18 | 230 | 220 | 240 | 350 | 315 | 310 | 230 | 0 | 1 | 0 |
| 7 | 200 | 34 | 3 | 2 | 6.12.2021 | 19:14:10 | 230 | 220 | 240 | 360 | 310 | 310 | 219 | 1 | 0 | 1 |
| 8 | 110 | 38 | 16 | 3 | 6.12.2021 | 19:17:23 | 230 | 220 | 240 | 380 | 370 | 350 | 220 | 1 | 1 | 1 |

Сторінки: 1 2 3 4 Вперед Останній 1 з 4

Рисунок 4.25 – Таблиця з архівними показниками

Нижче від представленої таблиці є змога переглянути напругу фаз сонячних батарей у вигляді графіку, який представлено на Рисунку 4.26. Правіше від графіку є невелике меню, де можна обрати режим відображення одна чи декілька сонячних батарей (рис. 4.27).

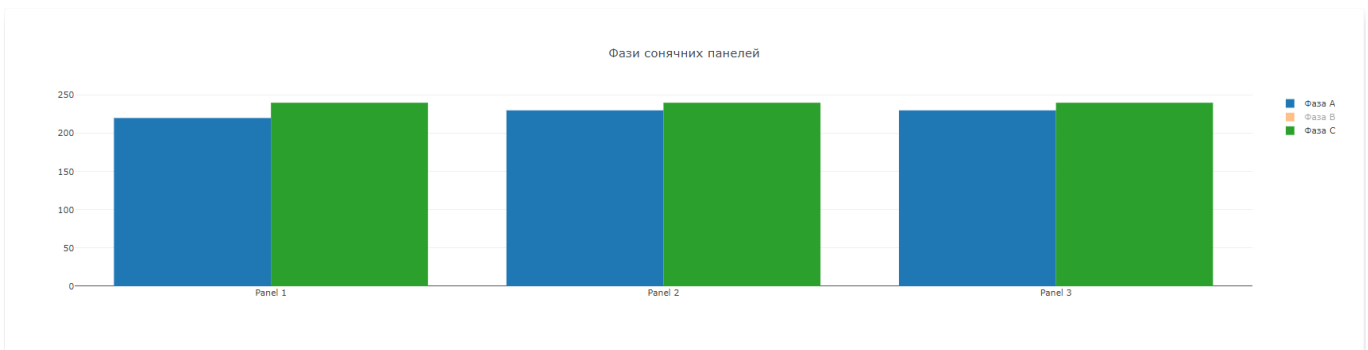


Рисунок 4.26 – Графік фаз

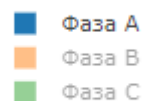


Рисунок 4.27 – Меню відображення

Також можна переглянути інформацію про напругу сонячних панелей відносно їх потужності, у вигляді точкового графіка (рис. 4.28), з додатковим меню, як у попередньо представленого.

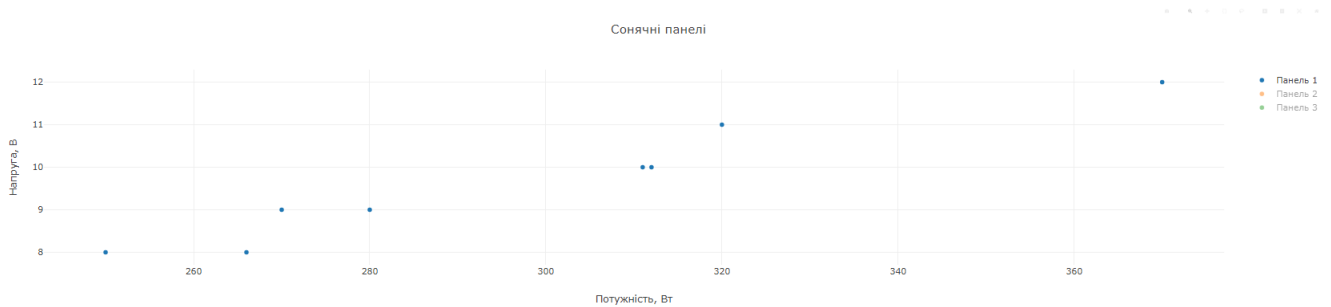


Рисунок 4.28 – Точковий графік

В кінці сторінки користувач має змогу переглянути архівні дані погоди, які представлено на рис. 4.29.

Архів погоди

| Країна | Місто | Широта | Довгота | Часовий пояс | Дата | Час | Швидкість вітру | Напрямок вітру | Хмарність | Температура | Час сходу | Час заходу |
|---------|-------|---------|---------|--------------|-----------|----------|-----------------|----------------|-----------|-------------|-----------|------------|
| Україна | Суми | 50.9216 | 34.8003 | 2 | 6.12.2021 | 16:05:21 | 5 | 120 | 100 | 0.3 | 08:28:07 | 16:35:53 |
| Україна | Суми | 50.9216 | 34.8003 | 2 | 6.12.2021 | 18:50:41 | 6 | 100 | 30 | 0.4 | 08:28:07 | 16:35:53 |
| Україна | Суми | 50.9216 | 34.8003 | 2 | 6.12.2021 | 18:50:41 | 6 | 100 | 30 | 0.4 | 08:28:07 | 16:35:53 |
| Україна | Суми | 50.9216 | 34.8003 | 2 | 6.12.2021 | 19:13:54 | 5.52 | 136 | 0 | 1.1 | 08:28:07 | 16:35:53 |
| Україна | Суми | 50.9216 | 34.8003 | 2 | 6.12.2021 | 19:14:10 | 5.52 | 136 | 0 | 1.1 | 08:28:07 | 16:35:53 |
| Україна | Суми | 50.9216 | 34.8003 | 2 | 6.12.2021 | 19:14:26 | 5.52 | 136 | 0 | 1.1 | 08:28:07 | 16:35:53 |
| Україна | Суми | 50.9216 | 34.8003 | 2 | 6.12.2021 | 19:14:42 | 5.52 | 136 | 0 | 1.1 | 08:28:07 | 16:35:53 |
| Україна | Суми | 50.9216 | 34.8003 | 2 | 6.12.2021 | 19:14:58 | 5.52 | 136 | 0 | 1.1 | 08:28:07 | 16:35:53 |

Сторінки: 1 2 3 4 5 ... Вперед Останній 1 з 22

Рисунок 4.29 – Архіви погоди

Зліва адміністративної панелі є меню, що розгортається по натисканню на нього, де є переходи на сторінки з сонячними панелями (рис. 4.30) та користувачами.

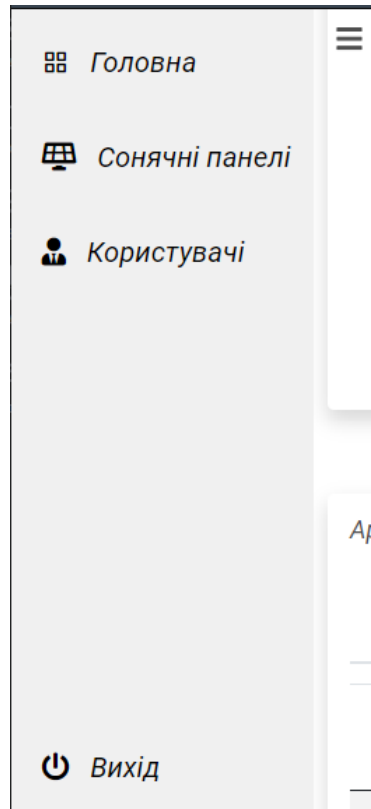


Рисунок 4.30 – Меню

Натиснувши на сонячні панелі, відбувається перехід до цієї сторінки, де є змога переглянути всі панелі разом, але з персональним відображенням даних (рис. 4.31).



Рисунок 4.31 – Сторінка сонячних панелей

Також на цій сторінці є змога частково керувати режимами панелей. Натиснувши на першу кнопку, користувач має змогу змінити режим роботи сонячної батареї, тобто чи працює вона, чи в режимі вимкнення. Натиснувши на другу кнопку, користувач має змогу змінити режим акумуляторів в режим зарядки або віддачі заряду (рис. 4.32).

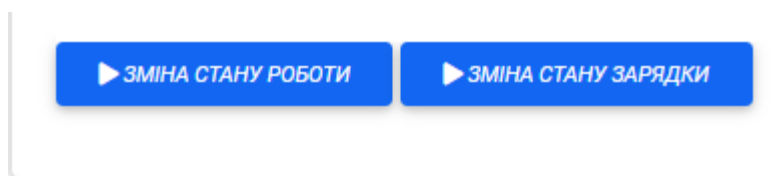


Рисунок 4.32 – Кнопки керування

Натиснувши на сторінку, користувачі користувач в залежності від прав має змогу додати, видалити чи відредагувати інформацію про користувача (рис. 4.33 – 4.35).



Рисунок 4.33 – Кнопка додавання



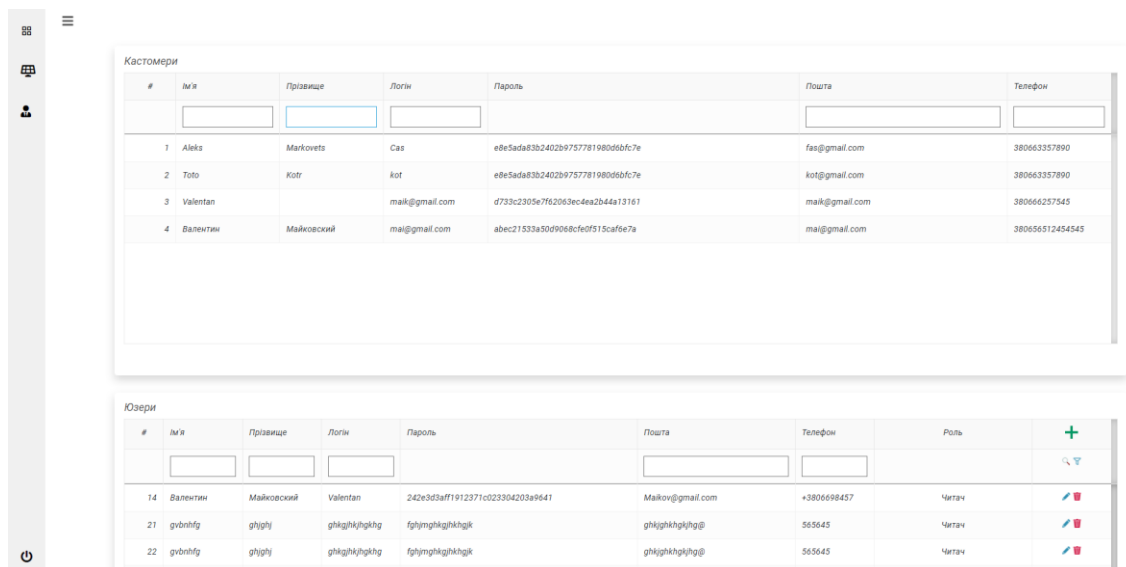
Рисунок 4.34 – Кнопка видалення



Рисунок 4.35 – Кнопка редагування

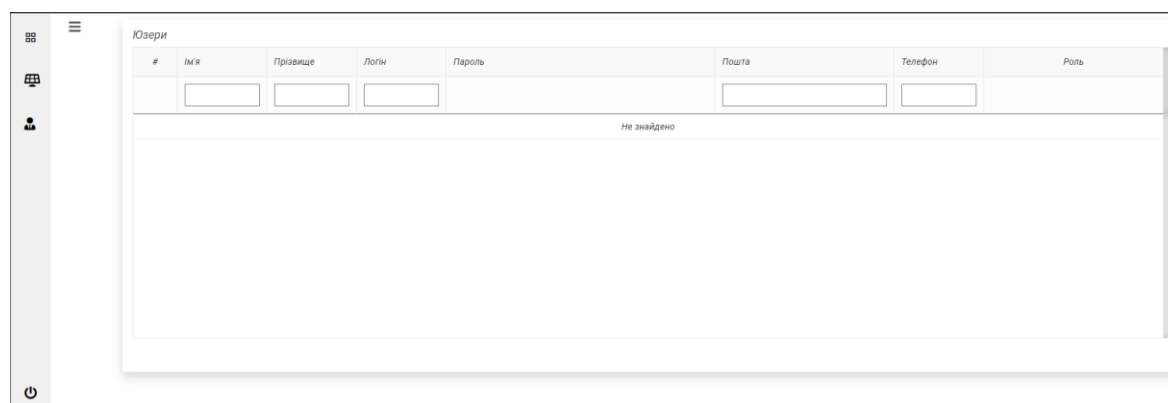
Для редагування можна натиснути на кнопку або просто натиснути на поле, яке треба змінити.

В залежності від прав доступу сторінка буде по різному відображатися користувачу. Сторінку з правами клієнта представлено на Рисунку 4.36. А з правами доступу користувач на Рисунку 4.37.



| # | Ім'я | Прізвище | Логін | Пароль | Пошта | Телефон |
|---|----------|-------------|----------------|----------------------------------|----------------|----------------|
| 1 | Aleks | Markovets | Cas | e8e5ada83b2402b9757781980d6bfc7e | fes@gmail.com | 380663357890 |
| 2 | Toto | Kotr | kot | e8e5ada83b2402b9757781980d6bfc7e | kot@gmail.com | 380663357890 |
| 3 | Valentan | | maik@gmail.com | d733c2305e7fd2063ec4ea2b44a13161 | maik@gmail.com | 380662257545 |
| 4 | Валентин | Майковський | mai@gmail.com | abec21533a50d908cfe1f515caf6e7a | mai@gmail.com | 38065512454545 |

Рисунок 4.36 – Сторінка з правами клієнта



| # | Ім'я | Прізвище | Логін | Пароль | Пошта | Телефон | Роль |
|-------------|------|----------|-------|--------|-------|---------|------|
| Не знайдено | | | | | | | |

Рисунок 4.37 – Сторінка з правами користувач

Після завершення роботи в панелі користувач може закрити вкладку в браузері або натиснути на кнопку вихід в лівому нижньому куті (рис. 4.38).



Рисунок 4.38 – Кнопка виходу з адміністративної панелі

ВИСНОВОК

За час роботи над кваліфікаційною роботою магістра було проведено аналіз предметної області, на підставі чого було проведено аналіз підходів до візуалізації даних систем підтримки прийняття рішень в енергетиці. Для початку проаналізовано системи СППР з можливістю відстеження даних у реальному часі та можливістю побудови графіків для візуального відображення даних. Також для додаткового порівняння було проаналізовано сайти, які дозволяють візуалізувати графічно данні, з можливістю підключення до них баз даних.

На підставі проведеного аналізу предметної області сформовано мету та задачі роботи. Також проведено планування робіт, під час яких було визначено ідентифікацію мети проєкту. Також зроблено планування змісту структури, де визначено етапи розробки проєкту та його виконавців, після чого побудовано календарний графік на основі Діаграми Ганта.

На основі попереднього етапу було перейдено до етапу проектування web – додатку візуалізації. Було проведено функціональне моделювання, розроблено функціональну модель інформаційної технології візуалізації. Розроблено UML діаграми варіантів використання та послідовності для web-додатку

Заключним етапом магістерської роботи реалізовано інформаційну технологію візуалізації даних для системи підтримки прийняття рішень при управлінні енергетичними мікромережами у вигляді web-додатку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Альтернативні джерела енергії [Електронний ресурс] // ecodevelop. – 2019. – Режим доступу до ресурсу: <https://ecodevelop.ua/alternativni-dzherela-energiyi/> (дата звернення: 20.10.2021).
2. Енергія сонця [Електронний ресурс] // Держенергоефективності. – 2017. – Режим доступу до ресурсу: <https://sae.gov.ua/uk/ae/sunenergy> (дата звернення: 20.10.2021).
3. Manson S. Microgrid Systems: Design, Control Functions, Modeling, and Field Experience / S. Manson, K. Ravikumar, S. Raghupathula., 2017. – 9 с. – (Schweitzer Engineering Laboratories, Inc) (дата звернення: 20.10.2021).
4. Shendryk, S., Shendryk, V., Tymchuk, S., Parfenenko, Y. Information Technology of Decision-Making Support on the Energy Management of Hybrid Power Grid. Communications in Computer and Information Science. – 2021. –1486 CCIS. – P. 72-83. DOI: 10.1007/978-3-030-88304-1_6.
5. Vira Shendryk, Olha Boiko, Yuliia Parfenenko, Sergii Shendryk, Sergii Tymchuk. Decision Making for Energy Management in Smart Grid. In: Research Anthology on Clean Energy Management and Solutions. – 2021. –P. 1742-1776.
6. Shendryk S., Shendryk V., Parfenenko Y., Drozdenko O. and Tymchuk S. Decision Support System for Efficient Energy Management of MicroGrid with Renewable Energy Sources. The 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS'2021). IEEE, 2021. . – P. 225-230.
7. Bartolozzi M. A Smart Decision Support System for Smart City / M. Bartolozzi, Bellini., 2015. – 9 с. – (University of Florence).
8. Системы мониторинга ИТ [Електронний ресурс] // ALP. – 2019. – Режим доступу до ресурсу: https://alp-itsm.ru/interesting/sistemyi_monitoringa_it/ (дата звернення: 20.10.2021).
9. IoT Power Consumption Control and Monitoring [Електронний ресурс] // Digiteum. – 2016. – Режим доступу до ресурсу:

<https://www.digiteum.com/portfolio/electricity-consumption-monitoring-remote-control/>
(дата звернення: 20.10.2021).

10. Мониторинг инфраструктуры и сервисов [Электронный ресурс] // Pronet. – 2018. – Режим доступа до ресурсу: <https://pronet.ua/ru/monitoring-infrastrukturyi-i-servisov/> (дата звернення: 20.10.2021).

11. Системы мониторинга и управления it-инфраструктурой [Электронный ресурс] // OSC. – 2017. – Режим доступа до ресурсу: <http://osc.com.ua/ru/activity/64> (дата звернення: 20.10.2021).

12. Гібридна сонячна електростанція з використанням системи автоматичного спостереження за сонцем [Электронный ресурс] // solar tech. – 2020. – Режим доступа до ресурсу: <https://solar-tech.com.ua/ua/our-objects/sistema-avtomaticheskogo-slezheniya-za-solncem-16kwatt.html> (дата звернення: 20.10.2021).

13. Microgrid Digital Control Solution from Schneider Electric | Schneider Electric [Электронный ресурс] // Schneider Electric. – 2020. – Режим доступа до ресурсу: <https://www.youtube.com/watch?v=YkN09Ui3eX8>. (дата звернення: 22.10.2021).

14. Pérez I. G. Monitoring interfaces for photovoltaic systems and dc microgrids: brief survey and application case / I. G. Pérez, A. J. Calderón Godoy // XLII JORNADAS DE AUTOMÁTICA : LIBRO DE ACTAS / I. G. Pérez, A. J. Calderón Godoy. – Castellón, 2021. – (Universitat Jaume I). – С. 183–189 (дата звернення: 25.10.2021).

15. Mahmoud L. Investigating the use of MATLAB/SIMULIK and LabVIEW in Microgrid Modelling and Simulation / Laban Mahmoud., 2017. – 119 с. – (Department of Mechanical and Aerospace Engineering) (дата звернення: 27.10.2021).

16. ETAP Microgrid Management [Электронный ресурс]. – 2021. – Режим доступа до ресурсу: <https://www.youtube.com/watch?v=MhAGsa5yLrM> (дата звернення: 28.10.2021).

17. Monash Microgrid: data visualisation and platform, load and generation management, operating systems [Электронный ресурс] // MONASH ENERGY INSTITUTE. – 2020. – Режим доступа до ресурсу: <https://www.monash.edu/energy-institute/news-events/past-events/events/2020/monash-microgrid-data-visualisation-and->

platform,-load-and-generation-management,-operating-systems (дата звернення: 28.10.2021).

18. A Real-Time Monitoring Platform for Distributed Energy Resources in a Microgrid—Pilot Study in Oman / N.Hosseinzadeh, A. Maashri, N. Tarhuni, A. Elhaffar., 2021. – 21 с. – (MDPI) (дата звернення: 29.10.2021).

19. Infogram [Електронний ресурс] // infogram.com. – 2021. – Режим доступу до ресурсу: <https://infogram.com> (дата звернення: 29.10.2021).

20. Chartblocks [Електронний ресурс] // chartblocks – Режим доступу до ресурсу: <https://app.chartblocks.com/> (дата звернення: 29.10.2021).

21. Datadeck [Електронний ресурс] // datadeck – Режим доступу до ресурсу: <https://dashv2.datadeck.com/> (дата звернення: 29.10.2021).

22. Bootstrap 5 [Електронний ресурс] – Режим доступу до ресурсу: <https://getbootstrap.com> (дата звернення: 30.10.2021)

23. Bootstrap 4 [Електронний ресурс] – Режим доступу до ресурсу: <https://getbootstrap.com> (дата звернення: 30.10.2021)

24. OpenWeatherMap [Електронний ресурс] – Режим доступу до ресурсу: <https://openweathermap.org/> (дата звернення: 2.11.2021)

25. Weatherbit [Електронний ресурс] – Режим доступу до ресурсу: <https://www.weatherbit.io/> (дата звернення: 2.11.2021)

26. AccuWeather [Електронний ресурс] – Режим доступу до ресурсу: <https://www.accuweather.com/> (дата звернення: 2.11.2021)

27. IDEF0 - Part 1 [Електронний ресурс] // SyQue. – 2018. – Режим доступу до ресурсу: http://syque.com/quality_tools/tools/Tools19.htm(дата звернення: 3.11.2021).

28. IDEF0 standard with ConceptDraw DIAGRAM [Електронний ресурс] // conceptdraw. – 2017. – Режим доступу до ресурсу: <https://www.conceptdraw.com/How-To-Guide/idef0>(дата звернення: 3.11.2021).

29. What is Unified Modeling Language (UML)? [Електронний ресурс] // VisualParadigm. – 2019. – Режим доступу до ресурсу: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/>(дата звернення: 3.11.2021).

30. UML Actor [Электронный ресурс] // Uml Diagrams. – 2016. – Режим доступа до ресурсу: <https://www.uml-diagrams.org/use-case-actor.html>(дата звернення: 3.11.2021).
31. What is Use Case Diagram? [Электронный ресурс] // VisualParadigm. – 2016. – Режим доступа до ресурсу: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>(дата звернення: 10.11.2021).
32. Achour M. Руководство по PHP [Электронный ресурс] / М. Achour, F. Betz, А. Dovgal // PHP. – 2021. – Режим доступа до ресурсу: <https://www.php.net/manual/ru/index.php#index> (дата звернення: 5.11.2021).
33. AmCharts [Электронный ресурс] – Режим доступа до ресурсу: <https://www.amcharts.com/> (дата звернення: 25.11.2021).
34. Plotly JavaScript [Электронный ресурс] // Plotly. – 2021. – Режим доступа до ресурсу: <https://plotly.com/javascript/> (дата звернення: 27.11.2021).

Додаток А. Планування робіт

А.1 Ідентифікація мети ІТ-проекту

Для деталізації кваліфікаційної роботи магістра було використано технологію SMART. Це підхід який дозволить зробити постанову цілей, що треба зробити. Така система дозволить на етапі постанови мети, що дозволить узагальнити всю наявність ресурсів та надати конкретні завдання для підвищення ймовірності досягнення бажаних результатів.

SMART-метод з'являється з розшифровки термінів:

- конкретна (Specific);
- вимірювана (Measurable);
- досяжна (Achievable);
- реалістична (Relevant);
- обмежена у часі (Time-framed).

Результат конкретизації кваліфікаційної роботи магістра з використанням методології SMART відображений у табл. А.1.

Таблиця А.1 – Конкретизація мети проекту методологією SMART

| | |
|-------------------------|--|
| Specific (конкретна) | Розробити інформаційну технологію візуалізації даних для системи підтримки прийняття рішень при управлінні енергетичними мікромережами |
| Measurable (вимірювана) | Результатом, має бути підсистема візуалізації, отримати доступ до якої, можна ввівши адресу. Пов'язані з ним модулі мають бути вірно налаштовані та під'єднані |

Продовження таблиці А.1 – Конкретизація мети проекту методологією SMART

| | |
|-------------------------------|---|
| Achievable (досяжна) | Проект містить потребу в спеціалісті із знанням HTML, CSS, PHP та програмного забезпечення (ПЗ), яке використовується для розроблення даного додатку. |
| Relevant (реалістична) | Досягнення мети реальне, оскільки розробка за допомогою сучасних технологій, не вимагає складних дій. |
| Time framed (обмежена у часі) | Обмеженість в часі викликана бажанням замовника, для отримання програмного забезпечення, як найшвидше. |

А.2 Планування змісту структури робіт ІТ - проекту

Поділ роботи на більш дрібні завдання – це поширений метод підвищення продуктивності, який використовується для того, щоб зробити роботу більш керованою і доступною. Для проектів структура розбивки робіт (WBS) є інструментом, що використовує цю техніку, і є одним з найбільш важливих документів з управління проектами. Тому для роботи над кваліфікаційною роботою було створено WBS, яка дає наглядно оцінити всі під-задачі даної роботи(рис А.1).

Основа планування управління проектом включає в себе не тільки визначення того, що являє собою робота, але і встановлення загальних підходів до того, як буде виконуватися робота. Додатковим механізмом до WBS є організаційна структурна структура (OBS), яка забезпечить ресурси, необхідні для виконання робіт за проектом (рис А.2).

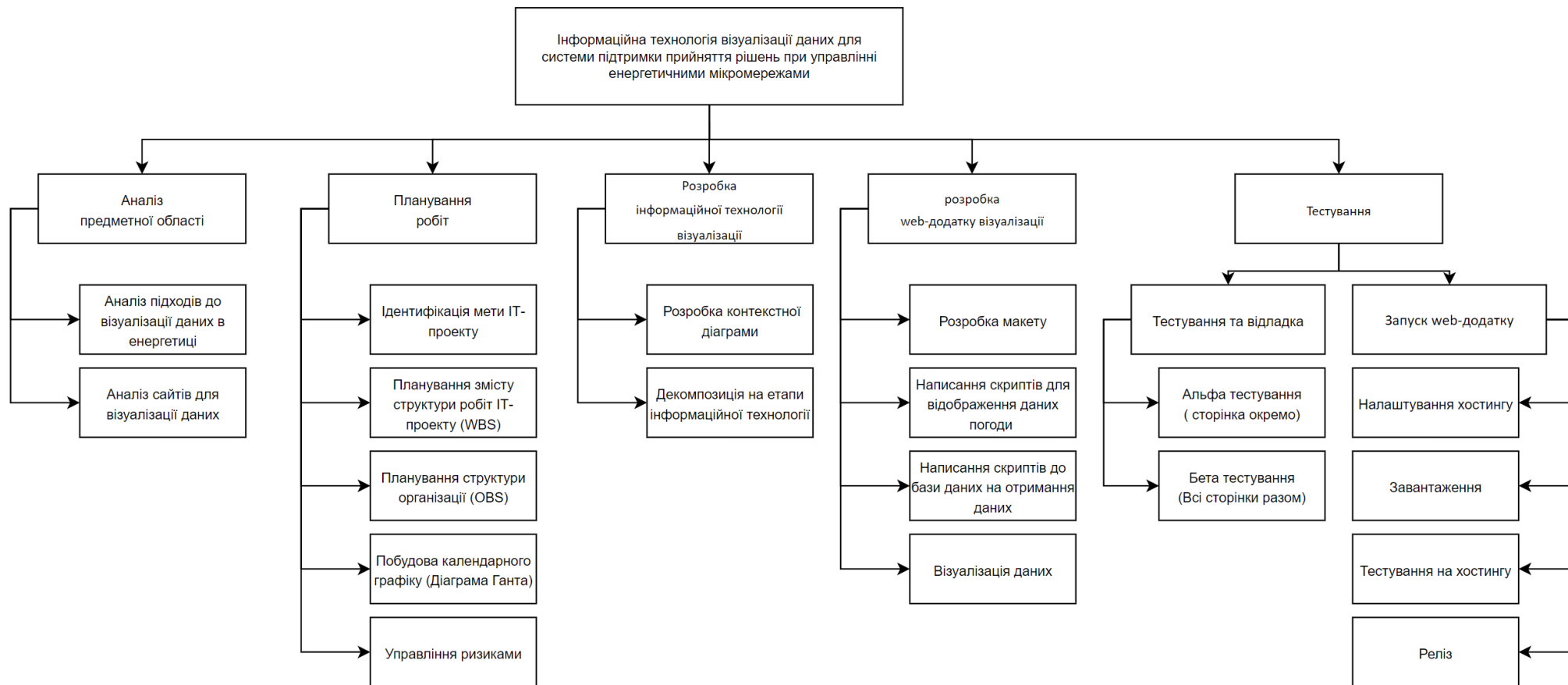


Рисунок А.1 – WBS – структура проекту

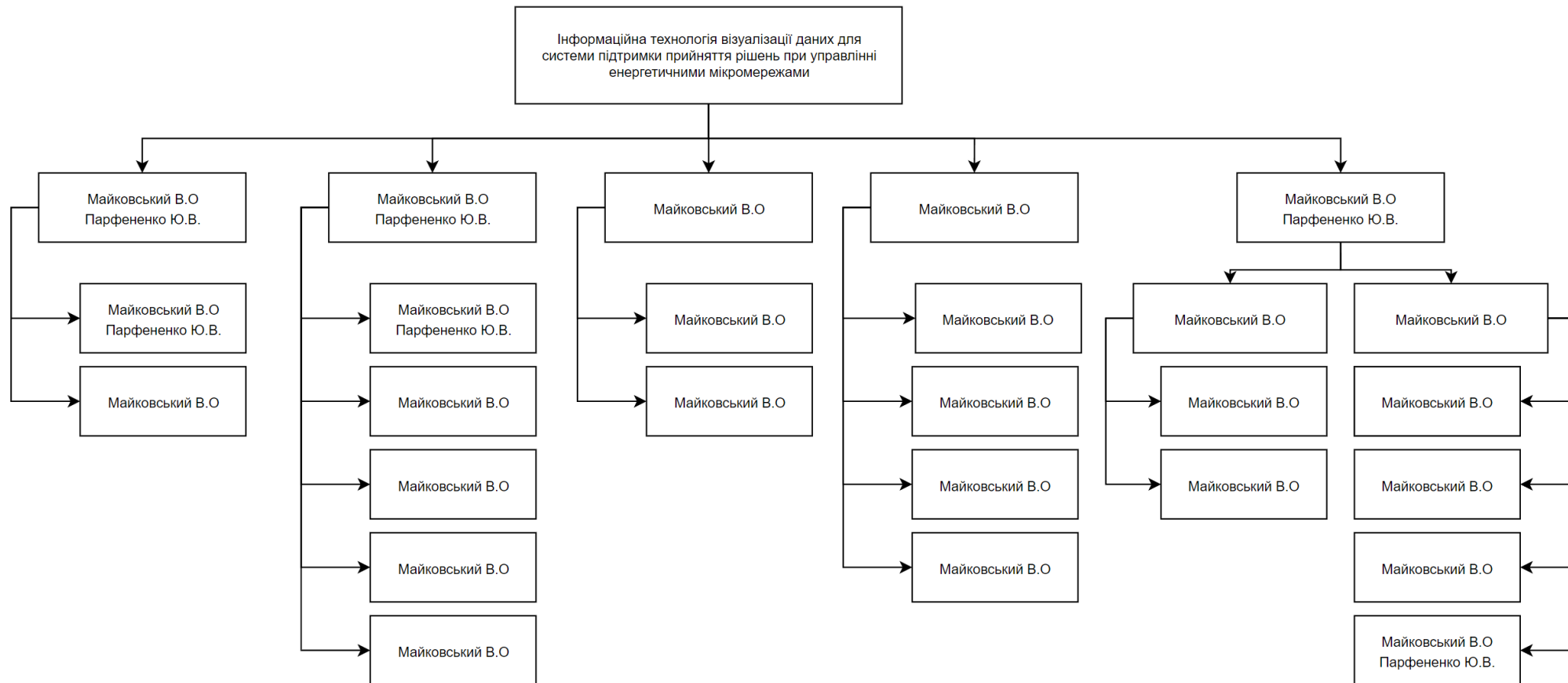


Рисунок А.2 – OBS - структура проекту

А.3 Побудова календарного графіку виконання ІТ-проекту

Для конкретного відстеження часу на роботу використовуються діаграми Ганта для відображення графіків часу. Діаграма Ганта – це інструмент управління проектом, який ілюструє план проекту. У ньому показані заплановані завдання і день або період, коли члени команди повинні працювати і виконувати їх. Він може надавати вичерпну інформацію на додаток до завдань і термінів виконання, таким як хід виконання окремих завдань, загальний хід проекту та учасники, відповідальні за виконання завдань. Діаграми Ганта демонструють взаємозв'язок завдань у часі та їх послідовність, залежності, критичний шлях і віхи. Це життєво важливий план проекту, який допомагає слідувати послідовності завдань за заданим графіком, щоб переконатися, що завершиться проект до заданої дати.

Зазвичай він включає в себе два розділи: в лівій частині представлений список завдань, а в правій - тимчасова шкала зі стовпцями розкладу, які візуалізують роботу. Діаграма Ганта також може включати залежності між завданнями, що наведено на рис. А.3.

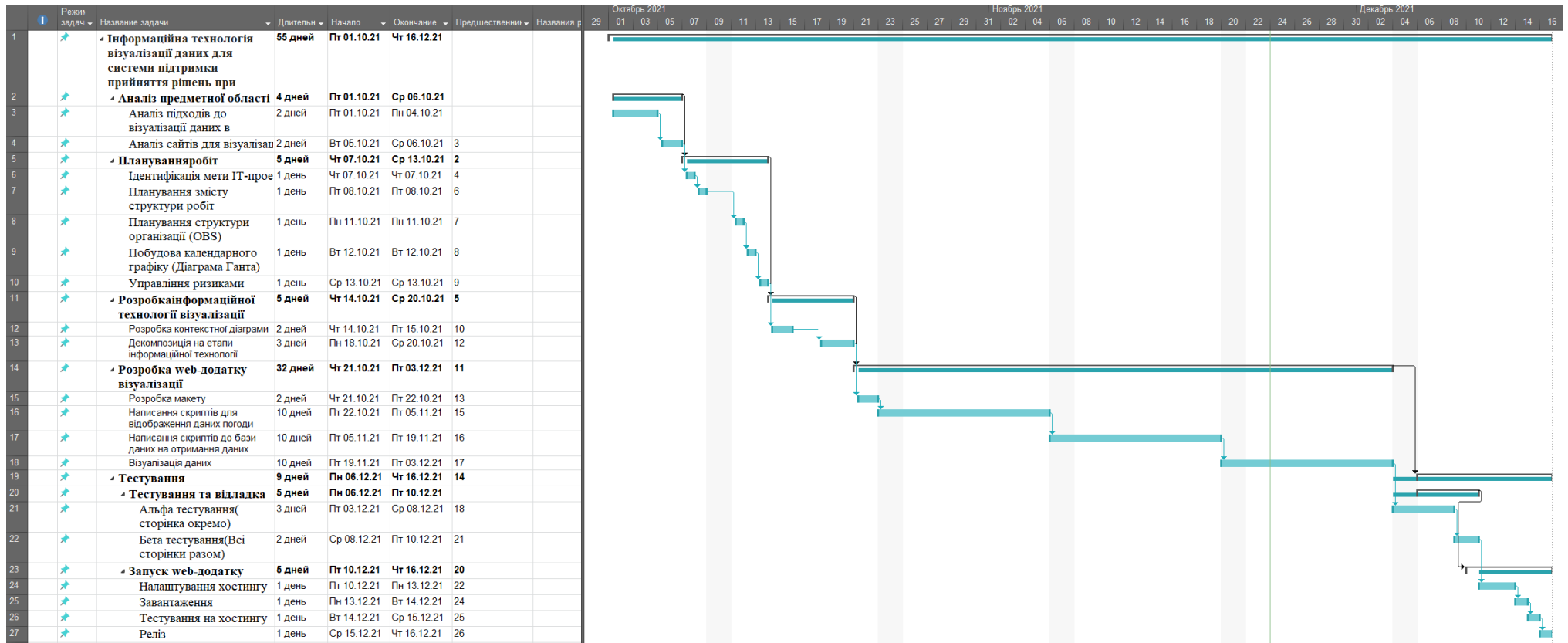


Рисунок А.3 - Діаграма Ганта

А.4 Планування ризиків проекту

Велика кількість проектів у ІТ галузі не завжди відповідає цілям, які поставили перед собою, наприклад: бюджет або терміни виконання. В більшій частині випадків такі проблеми пов'язані з нестачею, чи відсутністю якісного планування системою ризиків.

Протягом роботи над кваліфікаційною роботою магістра, а саме: “Інформаційна технологія візуалізації даних для системи підтримки прийняття рішень при управлінні енергетичними мікромережами”, було сформовано і проаналізовано можливі моменти ризиків, які мають можливість впливу на якість кінцевого продукту (табл. А.2).

Для оцінки ризиків за основу взята емпірична шкала оцінки ризику:

- 5 балів - критичний ризик (0,81 - 1);
- 4 бали - максимальний ризик (0,61 - 0,8);
- 3 бали - високий ризик (0,41 - 0,6);
- 2 бали - нормальний ризик (0,31 - 0,4); 1 бал - малий ризик (0 - 0,3).

Оцінки ризиків проекту наведено в табл. А.2.

Таблиця А.2 – Оцінка ризиків

| № | Ризик | Вірогідність | Вплив | Значення ризику | Оцінка | Опис ризику | Реакція на ризик | Боротьба з ризиком (час/гроші/час & гроші) |
|---|-------------------------|--------------|-------|-----------------|---------|---|--|--|
| 1 | Невизначений функціонал | 0,5 | 0,7 | 0,35 | Середня | Замовник не чітко визначає функціональні і можливості проекту | Чітко визначати функціональні вимоги проекту | час & гроші |

Продовження таблиці А.2 – Оцінка ризиків

| № | Ризик | Вірогідність | Вплив | Значення ризику | Оцінка | Опис ризику | Реакція на ризик | Боротьба з ризиком (час/гроші/час & гроші) |
|---|--------------------------------|--------------|-------|-----------------|-------------|---|--|--|
| 2 | Неповні вимоги | 0,4 | 0,4 | 0,16 | Низька | Замовник не може описати більшу частину загальних вимог під час початку роботи проекту, а тому, протягом всієї роботи вимоги змінюються | Замовнику треба описувати більше вимог віз загального числа під час старту роботи над проектом, щоб уникнути великої кількості зміни вимог | час & гроші |
| 3 | Хостинг | 0,4 | 0,7 | 0,28 | Середня | Проблеми з хостингом | Підібрати хостинг, який відповідає технологічним потребам | час & гроші |
| 4 | Низька кваліфікація розробника | 0,8 | 0,8 | 0,64 | Висока | Високі вимоги кваліфікації | Знайти курси чи статті, які допоможуть зробити поставлену задачу або консультація зі спеціалістом | час |
| 5 | Нереальний бюджет | 0,9 | 1 | 0,9 | Дуже висока | Передбачуваний бюджет залежить від заданого часу, зусиль і ресурсів. Тому вартість може перевищити бюджет | Призупинення роботи до відновлення фінансування | час |

Додаток Б. Фрагмент коду API OpenWeatherMap

```

<?
require_once __DIR__ . "../dbscript/include_database.php";

$lon = 34.800289;
$lat = 50.9216;
$lang = "ru";

$dt = 1637941221;
$url = "https://api.openweathermap.org/data/2.5/onecall?lat=" . $lat . "&lon="
. $lon . "&units=metric&lang=" . $lang . "&appid=" . $apiKey;

$url2 = "https://api.openweathermap.org/data/2.5/weather?lat=" . $lat . "&lon="
. $lon . "&units=metric&lang=" . $lang . "&appid=" . $apiKey;

$content = file_get_contents($url);
$weather = json_decode($content);

$content2 = file_get_contents($url2);
$weather2 = json_decode($content2);

$temp_now = $weather->current->temp; //Коректна погода
$feels_now = $weather->current->feels_like; //Відчувається погода
$clouds = $weather->current->clouds; //Хмарність
$pres = $weather->current->pressure;

$hum = $weather->current->humidity; //Вологість
$cityname = $weather->timezon; //Місто
$cityname2 = $weather2->name; //Місто 2 формат
$sunrise = $weather->current->sunrise; //Ранок
$sunset = $weather->current->sunset; //Захід
$wind_speed = $weather->current->wind_speed; // Швидкість вітру
$wind_deg = $weather->current->wind_deg; //Кут вітру
$wind_gust = $weather->current->wind_gust; //Пориви вітру

$cloud = $weather->current->weather[0]->description; //Хмарність
$icon = $weather->current->weather[0]->icon . ".png";

?>

```

Додаток В. Код головної сторінки

```

<!DOCTYPE html>
<html lang="en">

<head>
  <?php require_once __DIR__ . "../include/link.php"; ?>
  <title>SSMEnergy</title>
  <!-- Style -->
  <script src="../JS/datatable.js"></script>
  <style>
    #chartdiv {
      width: 100%;
      height: 400px;
    }

    #chartdiv2 {
      width: 100%;
      height: 250px;
    }

    @media (max-width: 768px) {
      #chartdiv {
        height: 300px;
      }

      #chartdiv2 {
        height: 250px;
      }
    }

    @media (max-width: 576px) {
      #chartdiv {
        height: 200px;
      }

      #chartdiv2 {
        height: 250px;
      }
    }

    [aria-labelledby="id-71-title"],
    [aria-labelledby="id-241-title"],
    [aria-labelledby="id-361-title"] {
      display: none;
    }
  </style>

  <!-- Style -->
  <!-- Resources Grafic -->
  <script src="https://cdn.amcharts.com/lib/4/core.js"></script>
  <script src="https://cdn.amcharts.com/lib/4/charts.js"></script>
  <script src="https://cdn.amcharts.com/lib/4/themes/animated.js"></script>
  <script src="https://cdn.plot.ly/plotly-2.7.0.js"></script>
  <script src="https://cdn.plot.ly/plotly-2.6.3.min.js"></script>
  <!-- Resources Grafic -->

  <script src="../JS/draganddrop.js"></script>
  <script src="../JS/weather.js"></script>
  <script src="../JS/calendar.js"></script>

```

```

        <script src="../../../JS/datatable.js"></script>

</head>

<body id="body-pd" onload="startTime()">

    <div class="row ">
        <div class="col-md-1 left ">
            <?php require_once("../include/sidebar_admin.php"); ?>
        </div> <!-- col-md-1 left -->
        <div class="col-md-11 right pads">
            <div class="row">
                <div class="col-lg-12 col-md-6 col-sm-12 pad">
                    <h1 class="text-center">Microgrid SSMEnergy</h1>
                </div>
            </div>
            <div class="card-block pos">
                <div class="row " id="sortable">
                    <!-- DRAG AND DROP -->
                    <div class="row " id="sortable">
                        <div class="col-lg-4 col-md-6 col-sm-12
pad">
                            <div class="card-sub1 shadow">
                                <div class="card-block ">
                                    <div class="widget-
container">
                                        <div class="row">
                                            <div
class="col-6">
                                                <h1
class="city" id="city"><?= $cityname2 ?></h1>
                                                <h1
id="day">Day</h1>
                                                <h3
id="date">Month, Day Year</h3>
                                                <h3
id="time">Time</h3>
                                                <p
class="geo"></p>
                                            </div> <!--
col -->
                                            <div
class="col-6 mx-auto">
                                                <div
class="row">
                                                    <div class="col-12">
                                                        <h1 id="weather-status"><?php echo "$cloud"; ?></h1>
                                                    </div><!-- row -->
                                                    </div><!-- col -->
                                                </div>
                                            </div>
                                        </div>
                                    </div>
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>

    <div class="col-12 mx-auto" style="width: 100px; height:100px;">
        <?php echo "<img src='http://openweathermap.org/img/w/" . $icon . "' / >"; ?>
    </div>

```

```

        </div><!-- row -->
-->
class="col-6">
    <h3>Температура</h3>
    <h4><?php echo "$temp_now"; ?></h4>
col -->
class="col-6">
class="row">
    <div class="col-7">Швидкість вітру</div>
    <div class="col-5"><?= $wind_speed ?> Km/h</div>
<!-- row -->
class="row">
    <div class="col-7">Хмарність</div>
    <div class="col-5"><?= $clouds ?> %</div>
<!-- row -->
class="row">
    <div class="col-7">Вологість</div>
    <div class="col-5"><?= $hum; ?> %</div>
</div><!-- row -->
class="row">
    <div class="col-7">Тиск</div>
    <div class="col-5"><?= $pres; ?> hPa</div>
</div><!-- row -->
class="row">
    <div class="col-7">Час сходу</div>
    <div class="col-5"><?= date("H:i:s", $sunrise); ?></div>
</div><!-- row -->
class="row">
    <div class="col-7">Час заходу</div>
    <div class="col-5"><?= date("H:i:s", $sunset); ?></div>

```

```

        </div><!-- row -->
col -->
container -->
-->
">
md-12">
        <script>
            am4core.ready(function() {
Themes begin
                am4core.useTheme(am4themes_animated);
Themes end
create chart
chart = am4core.create("chartdiv", am4charts.GaugeChart);
                chart.exporting.menu = new am4core.ExportMenu();
                chart.hiddenState.properties.opacity = 0; // this creates initial fade-in
                chart.startAngle = 0;
                chart.endAngle = 350;
                var axis
                = chart.xAxes.push(new am4charts.ValueAxis());
                axis.min
                = 0;
                axis.max
                = 350;
                axis.renderer.line.strokeWidth = 5;
                axis.renderer.line.strokeOpacity = 1;
                axis.renderer.line.stroke = am4core.color("#999");
                axis.renderer.inside = true;
                axis.renderer.axisFills.template.disabled = true;
                axis.renderer.grid.template.disabled = true;

```

```
axis.renderer.ticks.template.disabled = false
axis.renderer.ticks.template.length = 8;
axis.renderer.ticks.template.strokeOpacity = 1;

axis.renderer.labels.template.radius = 30;
axis.renderer.labels.template.disabled = true;
axis.renderer.ticks.template.disabled = true;

function
createLabel(label, deg) {
    var range = axis.axisRanges.create();
    range.value = deg;
    range.grid.disabled = true;
    range.label.text = label;
}

createLabel("N", 0);
createLabel("", 10);
createLabel("", 20);
createLabel("", 30);
createLabel("", 40);
createLabel("NE", 45);
createLabel("", 50);
createLabel("", 60);
createLabel("", 70);
createLabel("", 80);
createLabel("E", 90);
createLabel("", 100);
createLabel("", 110);
createLabel("", 120);
createLabel("", 130);
createLabel("SE", 135);
createLabel("", 140);
createLabel("", 150);
```

```
createLabel("", 160);
createLabel("", 170);
createLabel("S", 180);
createLabel("", 190);
createLabel("", 200);
createLabel("", 210);
createLabel("", 220);
createLabel("SW", 225);
createLabel("", 230);
createLabel("", 240);
createLabel("", 250);
createLabel("", 260);
createLabel("W", 270);
createLabel("", 280);
createLabel("", 290);
createLabel("", 300);
createLabel("", 310);
createLabel("NW", 315);
createLabel("", 320);
createLabel("", 330);
createLabel("", 340);
createLabel("", 350);

// hands
var
northHand = chart.hands.push(new am4charts.ClockHand());
    northHand.radius = am4core.percent(50);
    northHand.startWidth = 20;
    northHand.endWidth = 1;
    northHand.rotationDirection = "clockWise";
    northHand.pin.disabled = true;
    northHand.zIndex = 0;
    northHand.fill = am4core.color("#c00");
```



```

    northHand.stroke = am4core.color("#c00");
    northHand.value = 0;

    rotateCompass();

rotateCompass() {
    var angle = parseInt('<?=$wind_deg ?>');

    chart.animate({
        property: "startAngle",
        to: angle
    },
    1000, am4core.ease.cubicOut);

    chart.animate({
        property: "endAngle",
        to: angle + 360
    },
    1000, am4core.ease.cubicOut);

    northHand.animate({
        property: "value",
        to: -90 - angle
    },
    1000, am4core.ease.cubicOut);
    }

    });
</script>
<div
id="chartdiv"></div>
md-12 -->
md-12 ">
status"> Напрямок вітру</h2>
echo "$wind_deg"; ?></h2>
gust">Пориви вітру</h2>
echo "$wind_gust"; ?></h2>
md12 -->
-->
</div> <!-- col-lg-8 col-
<div class="col-lg-5 col-
    <h2 id="wind-
    <h2 id="wind"><?php
    <h2 id="wind-
    <h2 id="gust"><?php
</div> <!-- col-lg-4 col-
    </div> <!-- row-->
</div> <!-- card-syb -->
</div> <!-- col-lg-4 col-md-6 col-sm-12 pad
-->

```



```
<table>
  <tr class="orangeTr">
    <th>Π</th>
    <th>B</th>
    <th>C</th>
    <th>ϣ</th>
    <th>Π</th>
    <th>C</th>
    <th>H</th>
  </tr>
  <tr class="whiteTr">
    <th></th>
    <th>1</th>
    <th>2</th>
    <th>3</th>
    <th>4</th>
    <th>5</th>
    <th>6</th>
  </tr>
  <tr class="whiteTr">
    <th>7</th>
    <th>8</th>
    <th>9</th>
    <th>10</th>
    <th>11</th>
    <th>12</th>
    <th>13</th>
  </tr>
  <tr class="whiteTr">
    <th>14</th>
    <th>15</th>
```

```

        <th>16</th>
        <th>17</th>
        <th>18</th>
        <th>19</th>
        <th>20</th>
    </tr>
    <tr class="whiteTr">
        <th>21</th>
        <th>22</th>
        <th>23</th>
        <th>24</th>
        <th>25</th>
        <th>26</th>
        <th>27</th>
    </tr>
    <tr class="whiteTr">
        <th>28</th>
        <th>29</th>
        <th>30</th>
        <th>31</th>
        <th></th>
        <th></th>
        <th></th>
    </tr>
</table>
</div> <!-- month-calendar -->
<div class="date-calendar ">
<div class="datecont-calendar">
    <div class="row">
        <div id="date-calendar"></div>
    </div>

```

```

    <div class="row">
        <div id="day-calendar"></div>
    </div>

    <div class="row">
        <div id="month-calendar"></div>
    </div>
</div> <!-- datecont-calendar -->

</div> <!-- date-calendar-->
</div> <!-- contenrfront-calendar -->
front-calendar -->
calendar -->
calendar -->
</div> <!-- card-block -->
</div> <!-- card-syb -->
</div><!-- col-lg-4 col-md-6 col-sm-12 pad
-->
<div class="col-lg-4 col-md-6 col-sm-12
pad">
    <div class="card-sub1 shadow">
        <div class="card-block">
            <h2 class="col-lg-4 col-
md-6 col-sm-12">Атмосферний тиск</h2>
            <script>

                am4core.ready(function() {

begin
                am4core.useTheme(am4themes_animated);
                chart
                am4core.create("chartdiv2", am4charts.GaugeChart);
                chart.hiddenState.properties.opacity = 0;

                chart.xAxes.push(new am4charts.ValueAxis());
                900;
                1200;
                axis.strictMinMax = true;
                axis.renderer.inside = true;
                axis.renderer.radius = am4core.percent(97);
                // Themes
                // Themes end
                // create
                var chart =
                // var axis =
                // axis.min =
                // axis.max =
                //
                //
                //

```

```

axis.renderer.line.strokeOpacity = 1; //
axis.renderer.line.strokeWidth = 5; //
axis.renderer.line.stroke = chart.colors.getIndex(1); //
axis.renderer.ticks.template.disabled = false //
axis.renderer.ticks.template.stroke = chart.colors.getIndex(0); //
axis.renderer.labels.template.radius = 35; //
axis.renderer.ticks.template.strokeOpacity = 1; //
axis.renderer.grid.template.disabled = true; //
axis.renderer.ticks.template.length = 10; //
axis.hiddenState.properties.opacity = 1; //
axis.hiddenState.properties.visible = true; //
axis.setStateOnChildren = true; //
axis.renderer.hiddenState.properties.endAngle = 180; //

var axis2 =
chart.xAxes.push(new am4charts.ValueAxis());
axis2.min =
900;
axis2.max =
1200;

axis2.strictMinMax = true;

axis2.renderer.line.strokeOpacity = 1;
axis2.renderer.line.strokeWidth = 5;
axis2.renderer.line.stroke = am4core.color("#999");
axis2.renderer.ticks.template.stroke = chart.colors.getIndex(3);

axis2.renderer.ticks.template.disabled = false
axis2.renderer.ticks.template.strokeOpacity = 1;
axis2.renderer.grid.template.disabled = true;
axis2.renderer.ticks.template.length = 10;
axis2.hiddenState.properties.opacity = 1;
axis2.hiddenState.properties.visible = true;
axis2.setStateOnChildren = true;
axis2.renderer.hiddenState.properties.endAngle = 180;

```

```

chart.hands.push(new am4charts.ClockHand());
axis.renderer.line.stroke;
= axis.renderer.line.stroke;
axis;
hand.pin.radius = 14;
hand.startWidth = 10;

chart.hands.push(new am4charts.ClockHand());
axis2.renderer.line.stroke;
axis2.renderer.line.stroke;
axis2;

    hand2.pin.radius = 10;
    hand2.startWidth = 10;

    setInterval(function() {
ЗАПИСАТЬ ЗНАЧЕНИЯ !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
hand.showValue(parseInt("? echo "pres"; ?"), 1000, am4core.ease.cubicOut);
label.text = Math.round(hand.value).toString();

    hand2.showValue(parseInt('<?=$pres ?>'), 1000, am4core.ease.cubicOut);
    label2.text = Math.round(hand2.value).toString();

= new am4core.ListTemplate(new am4core.Label());
    labelList.template.isMeasured = false;
    labelList.template.background.strokeWidth = 2;
    labelList.template.fontSize = 25;
= am4core.color("#999");
    labelList.template.padding(10, 20, 10, 20);
    labelList.template.y = am4core.percent(50);
    labelList.template.horizontalCenter = "middle";

labelList.create();
label.parent = chart.chartContainer;
am4core.percent(40);

```

```

// var hand =
// hand.fill =
// hand.stroke
// hand.axis =
//
//
//
var hand2 =
hand2.fill =
hand2.stroke =
hand2.axis =
// СЮДА
//
//
}, 2000);
var labelList
labelList.fill
// var label =
//
// label.x =

```



```

<div class="card-block pos">
  <div class="row ">
    <div class="col pad">
      <div class="card-sub-table shadow">
        <div class="card-block ">
          <div class="mx-auto">
            <h5 class="mt-0">
              Архів погоди
            </h5>
          </div>
        </div>
      <?php

$sort_list = array(
  'country.name',
  'country.name DESC',
  'city.name',
  'city.name DESC',
=> 'datetimeinfo.day',
=> 'datetimeinfo.day DESC',
=> 'datetimeinfo.month',
=> 'datetimeinfo.month DESC',
=> 'datetimeinfo.year',
=> 'datetimeinfo.year DESC',

);

$sort = @$_GET['sort'];
if (array_key_exists($sort,
  $sort_list)) {
  $sort_sql =
  $sort_list[$sort];
  reset($sort_list);

  $a, $b)

class="active" href="?sort=' . $b . '">' . $title . ' <i>▲</i></a>';
class="active" href="?sort=' . $a . '">' . $title . ' <i>▼</i></a>';
href="?sort=' . $a . '">' . $title . '</a>';

} else {
  $sort_sql =
}

function sort_link_th($title,
{
  $sort = @$_GET['sort'];

  if ($sort == $a) {
    return '<a
  } elseif ($sort == $b) {
    return '<a
  } else {
    return '<a

}
}

```

```

?>
<div class="scroll-table len">
  <table class="table">
    <thead>
      <tr>
        <th><?php echo sort_link_th('Країна', 'country_name_asc', 'country_name_desc');
?></th>
        <th><?php echo sort_link_th('Місто', 'city_name_asc', 'city_name_desc');
?></th>
        <th>Широта</th>
        <th>Довгота</th>
        <th>Часовий пояс</th>
        <th><?php echo sort_link_th('День', 'datetimeinfo_day_asc',
'datetimeinfo_day_desc'); ?></th>
        <th><?php echo sort_link_th('Місяць', 'datetimeinfo_month_asc',
'datetimeinfo_month_desc'); ?></th>
        <th><?php echo sort_link_th('Рік', 'datetimeinfo_year_asc',
'datetimeinfo_year_desc'); ?></th>
        <th>Швидкість вітру</th>
        <th>Напрямок вітру</th>
        <th>Хмарність</th>
        <th>Температура</th>
        <th>Час
сходу</th>
        <th>Час
заходу</th>
      </tr>
    </thead>
  </table>
  <div class="scroll-table-
body">
    <table
class="table">
      <tbody>
        <?php
$result
= mysqli_query($conn, "SELECT country.name, city.name,
city.latitude, city.longitude, city.time_zone, datetimeinfo.day,
datetimeinfo.month, datetimeinfo.year, actualweather.wind_speed,
actualweather.wind_deg, actualweather.clouds, actualweather.temperature,
actualweather.sunrise, actualweather.sunset FROM actualweather
INNER
JOIN datetimeinfo ON actualweather.date_time_id = datetimeinfo.id

```

```

INNER
JOIN city ON actualweather.city_id = city.id INNER JOIN country ON
    city.country_id = country.id ORDER BY {$sort_sql}");

if
(mysqli_num_rows($result) > 0) {
    $row = mysqli_fetch_array($result);
do
{
    echo "<tr>
    <td>" . $row[0] . "</td>
    <td>" . $row[1] . "</td>
    <td>" . $row[2] . "</td>
    <td>" . $row[3] . "</td>
    <td>" . $row[4] . "</td>
    <td>" . $row[5] . "</td>
    <td>" . $row[6] . "</td>
    <td>" . $row[7] . "</td>
    <td>" . $row[8] . " m/c</td>
    <td>" . $row[9] . " °</td>
    <td>" . $row[10] . " %</td>
    <td>" . $row[11] . " °C</td>
    <td>" . date("H:i:s", $row[12]) . " </td>
    <td>" . date("H:i:s", $row[13]) . " </td>
    </tr>";
}
while ($row = mysqli_fetch_array($result));
}
?>
</tbody>
</table>
</div>
</div>
</div>
</div>
</div><!-- card-block -->
</div> <!-- col-md-11 right pads -->
</div> <!-- row -->

```

```

<script type="text/javascript">
    var trace3 = {
        x: ['<?php echo $namepanel ?>', '<?php echo $namepanel1 ?>',
'<?php echo $namepanel2 ?>'],
        y: ['<?php echo $UApanel ?>', '<?php echo $UApanel1 ?>',
'<?php echo $UApanel2 ?>'],
        name: 'Фаза A',
        type: 'bar'
    };

    var trace4 = {
        x: ['<?php echo $namepanel ?>', '<?php echo $namepanel1 ?>',
'<?php echo $namepanel2 ?>'],
        y: ['<?php echo $UBpanel ?>', '<?php echo $UBpanel1 ?>',
'<?php echo $UBpanel2 ?>'],
        name: 'Фаза B ',
        type: 'bar'
    };

    var trace5 = {
        x: ['<?php echo $namepanel ?>', '<?php echo $namepanel1 ?>',
'<?php echo $namepanel2 ?>'],
        y: ['<?php echo $UCpanel ?>', '<?php echo $UCpanel1 ?>',
'<?php echo $UCpanel2 ?>'],
        name: 'Фаза C ',
        type: 'bar'
    };

    var data = [trace3, trace4, trace5];

    var layout = {
        title: 'Фази сонячних панелей',
        barmode: 'group'
    };

    Plotly.newPlot('myDiv1', data, layout);
</script>

<script type="text/javascript">
    var trace1 = {
        x: [<?php echo $power ?>],
        y: [<?php echo $voltage ?>],
        mode: 'markers',
        name: 'Панель 1'
    };

    var trace2 = {
        x: [<?php echo $power1 ?>],
        y: [<?php echo $voltage1 ?>],
        mode: 'markers',
        name: 'Панель 2'
    };

    var trace3 = {
        x: [<?php echo $power2 ?>],
        y: [<?php echo $voltage2 ?>],
        mode: 'markers',
        name: 'Панель 3'
    };

```

```
var data = [trace1, trace2, trace3];

var layout = {
  title: 'Сонячні панелі',
  xaxis: {
    title: 'Потужність, Вт',
  },
  yaxis: {
    title: 'Напруга, В',
  }
};

var config = {
  toImageButtonOptions: {
    format: 'png', // one of png, svg, jpeg, webp
    filename: 'custom_image',
    responsive: true,
    height: 500,
    scale: 1 // Multiply title/legend/axis/canvas sizes by
this factor
  }
};

Plotly.newPlot('myDiv', data, layout, {
  displaylogo: false
}, config);
</script>

</body>

</html>
```