

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»**

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітньо-професійна програма: «Розробка програмного забезпечення»

Група: 4РП-08

Дипломний проект

здобувача освіти денної форми навчання

РП.08.01.000.ДП

***АНТОНОВА
ІЛІІ ВЛАДИСЛАВОВИЧА***

**м. Одеса
2025 р.**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітньо-професійна програма: «Розробка програмного забезпечення»

Група: 4РП-08

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту на тему:

Розробка веб-застосунку контролю енергоспоживання
В ПОБУТОВИХ УМОВАХ

Проектний матеріал складається з пояснювальної записки на 85 сторінках та графічного (презентаційного) матеріалу на 13 аркушах (слайдах)

Дипломник _____ (Антонов І.В.)

Керівник _____ (Жадан А.С.)

Консультанти:

з економічного розділу _____ (Канський М.Ю.)

з розділу охорони праці та техніки безпеки _____ (Чорновол Н.І.)

з нормоконтролю _____ (Петрашова В.І.)

старший консультант _____ (Кривченко Ю.В.)

До захисту допущений

Голова циклової комісії _____ (Кривченко Ю.В.)

Завідувач відділення _____ (Краснокутська К.Г.)

Захист «21» 06 2025 р.

Протокол ЕК № 1

Оцінка ЕК 5/90

Секретар ЕК _____

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Відділення комп'ютерних систем Комісія КТ та ПІ
Спеціальність 121 «Інженерія програмного забезпечення»
Освітньо-професійна програма «Розробка програмного забезпечення»

ЗАТВЕРДЖУЮ:
Заст. дир. з НВР Беркань І.В.
« 12 » 05 2025 р.

ЗАВДАННЯ

на дипломний проект

Здобувачеві освіти Антонову Іллі Владиславовичу
(прізвище, ім'я, по батькові)

1. Тема проекту Розробка веб-застосунку контролю енергоспоживання в побутових умовах.

затверджена наказом по коледжу від « 14 » листопада 2024р. № 246

2. Термін здачі закінченого проекту _____

3. Вихідні данні до проекту _____

1. Передбачити сучасний UI та дружній UX;

2. Обрати найоптимальніший фреймворк для реалізації поставленої задачі;

3. Визначити та застосувати формули для розрахунку електроенергії;

4. Передбачити будівництво графіків для отриманих розрахунків.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити)

1. Аналіз предметної області; 2. Технології та засоби розробки;


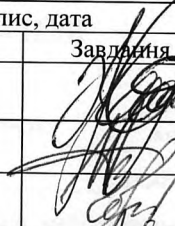
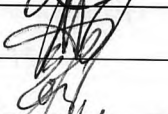
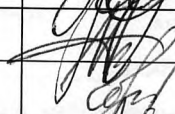
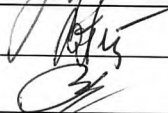
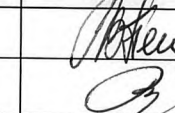
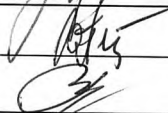
3. Проектування веб-застосунку; 4. Розробка клієнтського веб-застосунку;

5. Тестування створеного веб-застосунку;

6. Економічний розрахунок; 7. Аспекти охорони праці та техніки безпеки.

5. Перелік графічного (презентаційного) матеріалу (з точним зазначенням обов'язкових креслень, кількості слайдів)
Титул; Основні відомості; Вибір технологій; Архітектура застосунку; Структура каталогів;
Лендінг сторінка; Компоненти та маршрути; Додавання пристрою; Розрахунок споживання;
Інформаційні картки; Побудова графіків; Підказки та хелпери; Висновки.

6. Консультанти по проекту, із зазначенням розділів проекту, що їх стосується

| Розділ | Консультант | Підпис, дата | |
|----------------------|----------------|--|---|
| | | Завдання видав | Завдання прийняв |
| Основний розділ | Жадан А.С. |  |  |
| Економічний розділ | Канський М.Ю. |  |  |
| Розділ охорони праці | Чорновол Н.І. |  |  |
| Нормоконтроль | Петрашова В.І. |  |  |
| Старший консультант | Кривченко Ю.В. |  |  |

7. Дата видачі завдання _____


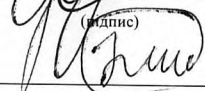
12.05.2025

Керівник

Жадан А.С.

Завдання прийняв до виконання

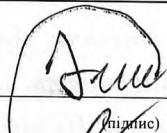
Антонов І. В.


(підпис)

(підпис)

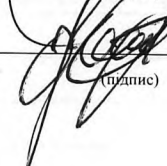
КАЛЕНДАРНИЙ ПЛАН

| № з/р | Назва етапів дипломного проекту | Термін виконання етапів дипломного проекту (роботи) | Відмітка про виконання |
|-------|--|---|------------------------|
| 1 | Формування вступу | 15.05.25 | виконано |
| 2 | Дослідження предметної області | 16.05.25 | виконано |
| 3 | Огляд аналогів | 19.05.25 | виконано |
| 4 | Вибір технічної літератури | 20.05.25 | виконано |
| 5 | Аналіз технологій розробки | 23.05.25 | виконано |
| 6 | Проектування веб-застосунку | 26.05.25 | виконано |
| 7 | Розробка клієнтського веб-застосунку | 29.05.25 | виконано |
| 8 | Тестування створеного веб-застосунку | 30.05.25 | виконано |
| 9 | Оформлення пояснювальної записки | 04.06.25 | виконано |
| 10 | Підготовка графічних матеріалів | 09.06.25 | виконано |
| 11 | Економічний розрахунок | 10.06.25 | виконано |
| 12 | Опис аспектів охорони праці та техніки безпеки | 12.06.25 | виконано |
| 13 | Підведення висновків | 13.06.25 | виконано |
| 14 | Підготовка доповіді для захисту | 16.06.25 | виконано |
| | | | |
| | | | |

Дипломник


(підпис)

Керівник


(підпис)

ЗМІСТ

| | |
|---|----|
| Вступ..... | 7 |
| 1 Основний розділ..... | 8 |
| 1.1 Дослідження предметної області | 8 |
| 1.1.1 Постанова проблематики | 8 |
| 1.1.2 Огляд існуючих рішень | 10 |
| 1.1.3 Технології для реалізації проекту | 12 |
| 1.2 Проектування веб-застосунку..... | 15 |
| 1.2.1 Проектування архітектури веб-застосунку..... | 15 |
| 1.2.3 Проектування лендінг сторінки веб-застосунку. | 20 |
| 1.2.2 Проектування дизайну веб-застосунку. | 22 |
| 1.2.4 Технології тестування клієнтської частини..... | 25 |
| 1.2.5 Маршрутизація..... | 26 |
| 1.2.6 Проектування структури бази даних | 28 |
| 1.2.7 Проектування серверної частини для надання допомоги користувачеві | 31 |
| 1.3 Тестування веб-застосунку | 34 |
| 1.3.1 Ручне тестування | 34 |
| 1.3.2 Функціональний огляд головної сторінки..... | 35 |
| 1.3.3 Елементи форми та функціонал додавання приладу..... | 38 |
| 1.3.4 Розрахунок вартості споживання | 42 |
| 1.3.5 Будування графіків та діаграм | 47 |
| 1.3.6 Генерація даних для графіка | 51 |
| 1.3.8 Функціонал допоміжних підказок (хелперів) | 53 |
| 1.3.9 Допоміжний функціонал для користувача | 56 |
| 2 Економічний розділ | 61 |
| 2.1 Резюме | 61 |
| 2.2 Визначення трудомісткості розробки ПЗ..... | 61 |
| 2.3 Розрахунок ціни програмного продукту..... | 64 |
| 3 Розділ охорони праці та техніки безпеки..... | 66 |

| | | | | | | |
|-----|------|----------|--------|------|--------------------------------|------|
| | | | | | <i>РП 08. 01 000. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 5 |

| | |
|--|----|
| 3.1 Резюме | 66 |
| 3.2 Оцінка небезпечних і шкідливих факторів, що впливають на програміста під час розробки програмного продукту | 66 |
| 3.3 Гігієнічні вимоги до умов виробничого середовища | 67 |
| 3.3.1 Вимоги до приміщення | 67 |
| 3.3.2 Освітлення | 67 |
| 3.3.3 Шум..... | 67 |
| 3.3.4 Мікроклімат..... | 68 |
| 3.3.5 Електробезпека | 68 |
| 3.3.6 Вимоги до організації робочого місця..... | 68 |
| 3.4 Пожежна безпека | 70 |
| Висновки..... | 71 |
| Перелік використаних інформаційних джерел | 72 |
| Додаток А. Фрагмент коду компоненту розрахунків електроспоживання..... | 73 |
| Додаток Б. Слайди мультимедійної презентації | 78 |

ВСТУП

Питання енергозбереження та раціонального використання ресурсів набувають усе більшої актуальності. Зростаюче споживання електроенергії, значні витрати на її виробництво та вплив на навколишнє середовище стимулюють суспільство до пошуку шляхів оптимізації споживання. Одним з ефективних підходів є розвиток культури свідомого споживання електроенергії, що передбачає обізнаність користувачів щодо власного енергоспоживання та мотивацію до його зменшення.

Інформаційні технології, зокрема веб-застосунки, можуть відігравати ключову роль у формуванні такої культури. За допомогою зручних та інтерактивних інтерфейсів користувачі мають змогу в реальному часі відстежувати своє споживання електроенергії, отримувати рекомендації щодо його оптимізації, а також оцінювати ефективність впроваджених змін. Цифрові платформи сприяють підвищенню енергоефективності на побутовому рівні та сприяють сталому розвитку.

Метою даної роботи є розробка веб-застосунку для моніторингу та аналізу споживання електроенергії з акцентом на підвищення обізнаності користувачів та стимулювання свідомої поведінки. Основним завданням є створення інтуїтивно зрозумілого інтерфейсу, який дозволить користувачам переглядати динаміку власного споживання, отримувати поради щодо енергозбереження та аналізувати свій прогрес.

Для досягнення поставлених завдань було використано сучасний стек технологій, зокрема Angular, що забезпечує високу продуктивність клієнтської частини та гнучкість у реалізації складних інтерактивних функцій.

Результатом розробки є веб-застосунок, який може бути корисним для користувачів, що прагнуть контролювати та оптимізувати власне споживання електроенергії. Система дозволяє отримувати достовірну, наочну та актуальну інформацію про витрати електроенергії, а також надає рекомендації щодо енергоощадної поведінки. Вбудований інструмент може стати засобом у формуванні звичок споживання та сприяти зменшенню навантаження.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------------|------|
| | | | | | <i>РП 08. 01 000. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 7 |

1 ОСНОВНИЙ РОЗДІЛ

1.1 Дослідження предметної області

1.1.1 Постанова проблематики

У ХХІ столітті питання енергетичної безпеки та ефективного використання енергоресурсів набувають надзвичайної важливості. Зростання населення, індустріалізація, діджиталізація та глобальні кліматичні виклики змушують держави світу переглядати свої підходи до споживання електроенергії. Україна, як країна з обмеженими власними ресурсами та високою залежністю від енергоносіїв, особливо гостро відчуває потребу в переході до енергоефективної моделі споживання.

За даними Державного агентства з енергоефективності та енергозбереження України, близько 40% електроенергії в країні споживається побутовим сектором. У більшості випадків споживачі не усвідомлюють, які прилади є найбільшими "споживачами", не знають, як правильно оптимізувати використання енергії та не мають доступу до інструментів для моніторингу власного енергоспоживання. Це призводить до перевитрат, збільшення навантаження на енергосистему, нерационального використання електроенергії та, як наслідок, – до зростання витрат як на рівні домогосподарств, так і держави загалом.

Особливо гостро проблема постає в умовах військового стану, коли критична інфраструктура країни знаходиться під загрозою, а стабільність енергосистеми залежить від кожного споживача. Восени та взимку 2022–2023 років Україна вже зіткнулась із необхідністю запровадження графіків відключення електроенергії, що підкреслило важливість раціонального підходу до споживання.

У відповідь на ці виклики Україна поступово впроваджує сучасні технології енергомоніторингу, «розумні» лічильники та автоматизовані системи обліку електроенергії. Такі рішення дають змогу споживачам краще контролювати своє енергоспоживання, виявляти неефективні прилади та змінювати поведінкові звички на користь енергозбереження. Крім того, цифрові платформи, що надають аналітику в реальному часі, дозволяють органам влади ухвалювати більш

| | | | | | | |
|-----|------|----------|--------|------|--------------------------------|------|
| | | | | | <i>РП 08. 01 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 8 |

обґрунтовані рішення щодо енергетичної політики та реагувати на критичні ситуації в енергосистемі.

| Установлена потужність і відпуск електроенергії за типами електростанцій / Setted Capacity of power plants and release of electricity, by type of power plants | | | | | |
|--|--|---------------|--|---------------|---------------------------------|
| Типи генеруючих підприємств | Установлена потужність електростанцій на кінець року, тис. кВт / Power plants capacity by year-end, thsd. kW | | Відпуск електроенергії, млн. кВт-год / Electricity output, mln. kW-h | | Types of generating enterprises |
| | 2020 | 2021 | 2020 | 2021 | |
| Усього | 55138 | 54443 | 137197 | 142279 | Total |
| у тому числі | | | | | including |
| теплові електростанції | 22311 | 21129 | 36300 | 32348 | combined heat and power plants |
| теплоелектроцентраль | 5890 | 5846 | 12837 | 8619 | power plants |
| атомні електростанції | 13835 | 13835 | 71249 | 81016 | nuclear power plants |
| гідроелектростанції ¹ | 6335 | 6365 | 7415 | 10270 | hydro power plant ¹ |
| інші електростанції | 6767 | 7268 | 9396 | 10026 | other power plants |
| з них | | | | | of which |
| вітрові електростанції | 1110 | 1438 | 3271 | 3599 | wind power plants |
| сонячні електростанції | 5194 | 5388 | 5684 | 5884 | solar power plants |
| ¹ З урахуванням гідроакмулюючих електростанцій. / Including pumped hydro power plants. | | | | | Page 2 |
| Установлена потужність і відпуск теплоенергії за типами генеруючих підприємств (установок) / Setted capacity of heat producers and release of heat, by type of power plants | | | | | |
| Типи генеруючих підприємств | Установлена потужність електростанцій на кінець року, Гкал / год / Power plants capacity by year-end, Gcal/h | | Відпуск теплоенергії, тис. Гкал / Output heat, thsnd. Gcal | | Types of generating enterprises |
| | 2020 | 2021 | 2020 | 2021 | |
| Усього | 113384 | 114749 | 88954 | 91317 | Total |
| у тому числі | | | | | including |
| теплові електростанції | 4207 | 4206 | 1125 | 1258 | heat power plants |
| теплоелектроцентраль | 29896 | 29657 | 25517 | 26889 | combined heat and power plants |
| атомні електростанції | 2596 | 2596 | 1387 | 1557 | nuclear power plants |
| теплоцентралі (котельні) | 70485 | 71558 | 52954 | 53441 | heat plants |
| утилізаційні установки | 4358 | 4264 | 7088 | 7322 | utilization units |
| інші установки | 1843 | 2468 | 883 | 850 | other units |

Рисунок 1.1. Установлена потужність і відпуск електроенергії за типами електростанцій

Наразі в Україні не існує широкодоступних цифрових сервісів, які дозволяли б громадянам у зручний спосіб відстежувати своє споживання електроенергії, порівнювати його з середніми показниками, аналізувати структуру витрат та отримувати персоналізовані поради з енергоощадності. Такі сервіси існують у багатьох країнах Європейського Союзу та вже довели свою ефективність. Вони

сприяють не лише економії коштів, а й формуванню культури свідомого енергоспоживання – тобто відповідального підходу до використання ресурсів.

Окрема проблема – відсутність знань та мотивації в населення щодо зміни власних звичок. Багато людей навіть не підозрюють, що, наприклад, залишаючи прилади в режимі очікування або не вимикаючи освітлення в порожніх приміщеннях, вони значно збільшують своє щомісячне споживання електроенергії. Формування відповідального ставлення до енергії можливе лише через обізнаність – а вона, у свою чергу, потребує доступу до зручної та зрозумілої інформації.

Актуальність дослідження полягає в розробці та впровадженні інструментів цифрового контролю та аналізу електроспоживання, які дозволять українським споживачам не лише краще розуміти власні витрати, а й активно впливати на їх зменшення. Створення такого веб-застосунку, побудованого на сучасних технологіях, є логічною відповіддю на виклики часу, а також кроком до енергоефективного та сталого майбутнього.

1.1.2 Огляд існуючих рішень

Більшість енергопостачальних компаній в Україні, таких як YASNO, Львівобленерго, Харківобленерго та інші, пропонують споживачам можливість користування особистими кабінетами на своїх веб-платформах. Через ці кабінети побутові користувачі можуть взаємодіяти з постачальниками електроенергії дистанційно, отримуючи доступ до ключової інформації та основних сервісів. Такі рішення стали стандартом обслуговування в багатьох регіонах.

Серед основних переваг таких сервісів - це можливість онлайн-передачі показників лічильника, перегляду історії нарахувань і оплат, доступу до актуального тарифу та балансу, а також отримання повідомлень про відключення чи аварії. Це значно полегшує контроль за споживанням електроенергії та економить час користувача, зменшуючи потребу в особистому відвідуванні сервісних центрів.

Однак, ці сервіси мають і свої недоліки. Зокрема, відсутній детальний аналіз споживання за приладами, годинами чи днями, а також персоналізовані поради щодо енергоощадності. Інтерфейси багатьох платформ залишаються складними та

| | | | | | | |
|-----|------|----------|--------|------|--------------------------------|------|
| | | | | | <i>РП 08. 01 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 10 |

перевантаженими, що ускладнює їхнє використання, особливо для людей без технічної підготовки. У сервісу, дані не переносяться при зміні постачальника чи адреси де споживач втрачає історію, що негативно впливає на безперервність аналізу. Можна додати, що наступний ресурс як сервіс «Електронний кабінет енергетики» від ДП «Оператор системи передачі» (Укренерго). Даний ресурс орієнтований переважно на фахівців у сфері енергетики, проте має відкриті розділи з актуальною інформацією про стан енергосистеми.

Сервіс надає низку переваг, що роблять його корисним для фахівців у сфері енергетики. Зокрема, він пропонує оперативну інформацію про стан генерації та споживання електроенергії по регіонах, що дозволяє відстежувати енергетичну ситуацію в реальному часі. Також регулярно публікуються графіки аварійних та планових відключень, надаються дані про баланс електроенергії в Україні та поточний стан напруги в окремих містах.

Однак, попри ці функціональні можливості, система має суттєві обмеження. Найбільшим недоліком є складність подачі інформації на інтерфейс і структура даних орієнтовані переважно на професіоналів, а не на пересічних користувачів. Унаслідок цього більшість громадян не зможуть ефективно скористатися доступними даними без додаткових знань.

До того ж сервіс не передбачає персоніфікацію, де вся інформація має загальний характер і не враховує індивідуальні споживчі показники. Важливо також, що відсутня інтеграція з особистими кабінетами або лічильниками користувачів, що обмежує можливість отримання персоналізованої аналітики та рекомендацій.

Також як для прикладу розглянемо мобільні додатки як: «Дія», «YASNO», «ЄлектроКабінет».

Останніми роками енергетичні сервіси активно переходять у мобільний формат, зокрема через додаток «Дія», де планується інтеграція послуг у цій сфері. Хоча на момент підготовки дипломної роботи функціонал додатку ще обмежений, вже зараз користувачі можуть отримати деякі послуги, наприклад, подати заявку на обмін ламп на енергоощадні, ініційований президентом України. Перехід у

| | | | | | | |
|-----|------|----------|--------|------|--------------------------------|------|
| | | | | | <i>РП 08. 01 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 11 |

мобільний формат відкриває нові можливості для зручної взаємодії з системами енергопостачання.

Серед переваг мобільних сервісів варто відзначити їхню доступність і зручність використання на смартфоні, що дозволяє передавати показники лічильника або переглядати рахунки у будь-який момент. Окрім цього, користувачі оперативно отримують сповіщення про зміни, що сприяє кращому контролю та швидкому реагуванню. Завдяки цьому мобільні додатки поступово стають частиною повсякденного управління енергоспоживанням.

Разом з тим, існують і помітні недоліки. Функціональність таких сервісів часто залежить від регіону та доступності інтегрованих постачальників, що обмежує їх використання для певної частини населення. У більшості додатків відсутня глибока аналітика або рекомендації щодо покращення енерговикористання. Більше того, багато з них є лише копією особистого кабінету без суттєвого розширення функціоналу, що знижує їхню цінність у контексті підвищення енергоефективності.

Наявні офіційні сервіси в Україні надають базовий функціонал для взаємодії користувача з енергопостачальниками – передача показників, перегляд нарахунків та сповіщення. Проте жоден з них не орієнтований безпосередньо на формування культури свідомого споживання електроенергії. Вони не надають персоніфікованих аналітичних даних, не дозволяють здійснювати глибокий моніторинг витрат електроенергії та не навчають користувачів основам енергоефективності.

Це створює передумови для розробки нового, інтерактивного веб-застосунку, який буде зосереджений саме на інформуванні, аналізі споживання та мотивації користувачів до зменшення витрат електроенергії через зручні цифрові інструменти.

1.1.3 Технології для реалізації проекту

Для реалізації клієнтської частини веб-застосунку, що сприяє свідомому споживанню електроенергії, було обрано фреймворк Angular – сучасну платформу з відкритим вихідним кодом для побудови односторінкових застосунків (SPA). Angular широко використовується у веб-розробці завдяки своїй продуктивності,

| | | | | | | |
|-----|------|----------|--------|------|--------------------------------|------|
| | | | | | <i>РП 08. 01 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 12 |

модульності, підтримці TypeScript та активній спільноті розробників [1].

Фреймворк Angular дозволяє організовувати розробку за допомогою чіткої архітектури компонентів та сервісів, що значно покращує масштабованість застосунку. Крім того, Angular має потужну систему реактивних форм, двосторонню прив'язку даних, систему маршрутизації та підтримку роботи з HTTP-запитами. У даному проекті використовувалась структура директорій components, pages, services, interfaces та pipes, що свідчить про логічну модульність і дотримання принципів чистої архітектури.



Рисунок 1.2. Основні сервіси технології Angular

В якості мови програмування використовується TypeScript – надмножина JavaScript, яка дозволяє використовувати типізацію. Допоміжний функціонал бібліотеки значно підвищує надійність та безпечність коду, особливо у великих проектах. TypeScript інтегрується з Angular на рівні компіляції, також слід зазначити, що можна використовувати простий JavaScript, забезпечуючи перевірку типів ще до виконання коду.

Для стилізації застосунку використовується SCSS та препроцесор CSS, що дозволяє створювати багаторазові стилі, змінні, міксини та структури з вкладеннями. Виконання даного функціоналу сприяє підтримці чистоти та зручності роботи зі стилями у великих інтерфейсних проектах. SCSS також полегшує підтримку єдиного візуального стилю застосунку завдяки використанню

| | | | | | | |
|-----|------|----------|--------|------|--------------------------------|------|
| | | | | | <i>РП 08. 01 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 13 |

глобальних змінних для кольорів, відступів та шрифтів. Крім того, модульність SCSS дозволяє розділити стилі за компонентами, що покращує читабельність і структуру коду.

Для хостингу та зберігання налаштувань середовища було обрано Firebase, про що свідчать наявність файлів `firebase.json` та `.firebaserc`. Firebase забезпечує швидке розгортання веб-застосунків, інтеграцію з аналітикою, а також підтримку авторизації та бази даних у хмарі. У рамках цього проєкту Firebase виконує роль бекенду як сервісу (BaaS).

У проєкті також використовується `.prettierrc.json` для автоматичного форматування коду згідно з єдиним стилем. Це дозволяє забезпечити консистентність структури коду, підвищити читабельність і спростити командну розробку.

Проєкт зберігається у віддаленому репозиторії GitHub, що забезпечує контроль версій, командну співпрацю та прозору історію змін. Репозиторій містить більше 40 комітів, що свідчить про активну розробку, сам сервіс забезпечує надійне сховище даних, яким переважно користуються більшість компаній світу.

На табл. 1.1 зображено обґрунтування обраних технологій, їх опис та переваги.

Таблиця 1.1. Вибір технологій для розробки веб-застосунку

| Технологія | Призначення | Переваги | Причина вибору у проєкті |
|-------------------|---|--|--|
| Angular | Фреймворк для побудови SPA | Компонентна архітектура, CLI, реактивність, підтримка TypeScript | Створення модульного, масштабованого фронтенду |
| TypeScript | Мова програмування | Статична типізація | Покращення надійності коду |
| SCSS | CSS-препроцесор для стилізації інтерфейсу | Змінні, вкладення, міксини, повторне використання стилів | Організація масштабованої системи стилів |
| Firebase | Хмарна платформа для хостингу та бекенду | Просте розгортання, автентифікація, масштабованість | Швидке налаштування хостингу та хмарної інтеграції |

| Технологія | Призначення | Переваги | Причина вибору у проєкті |
|-----------------|--|---|---|
| Prettier | Інструмент для автоматичного форматування коду | Єдиний стиль, автоматичне вирівнювання, командна узгодженість | Підтримка чистоти та стандартизації коду |
| GitHub | Система контролю версій та віддалене сховище | Історія змін, командна співпраця, CI/CD | Розподілений контроль версій, прозорість розробки |

1.2 Проєктування веб-застосунку

1.2.1 Проєктування архітектури веб-застосунку

Архітектура веб-застосунку відіграє ключову роль у забезпеченні його масштабованості, підтримуваності та зручності в розробці. У випадку використання Angular, проєктування архітектури ґрунтується на принципах модульності, інкапсуляції та поділу обов'язків відповідно до патерну Component-Based Architecture [2].

Ще одним важливим аспектом є використання сервісів (services) для обробки бізнес-логіки та взаємодії з API. Це дозволяє компонентам залишатися максимально "чистими", відповідаючи лише за відображення інтерфейсу користувача. Angular між модулями і сприяє гнучкості та масштабованості архітектури проєкту [3].

Застосування такої архітектури дає змогу легко розширювати функціонал проєкту без суттєвого впливу на вже існуючі модулі. Кожен компонент або сервіс можна протестувати окремо, що значно підвищує стабільність та якість програмного забезпечення. Крім того, чітке розділення відповідальностей спрощує командну розробку: кілька розробників можуть паралельно працювати над різними частинами системи без конфліктів у кодї. Даний підхід є основою для побудови надійного, масштабованого та зручного в підтримці веб-застосунку.

Використання Angular CLI полегшує створення, тестування та розгортання застосунку, забезпечуючи стандартизований підхід до управління проєктом.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------------|------|
| | | | | | РП 08. 01 001. 00 ДП ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 15 |

Застосування реактивного програмування через RxJS дозволяє ефективно обробляти асинхронні операції, такі як запити до API чи обробка подій користувача. Крім того, впровадження системи маршрутизації (routing) в Angular забезпечує плавну навігацію між різними частинами застосунку без необхідності перезавантаження сторінки. Для підвищення продуктивності також використовуються техніки ледачого завантаження (lazy loading) модулів, що зменшує час завантаження застосунку. Усе це разом сприяє створенню високоефективного та користувацьки орієнтованого веб-застосунку.

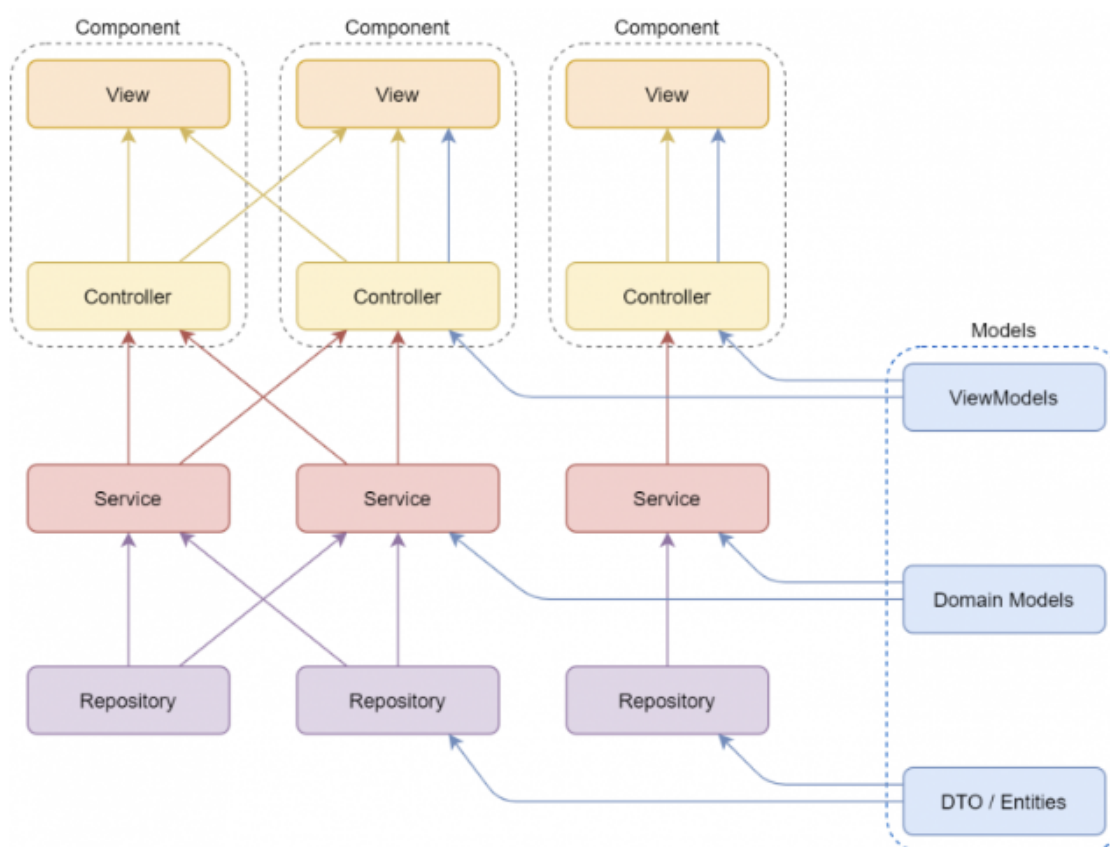


Рисунок 1.3. Архітектура веб-застосунку за допомогою Angular

У Angular застосунок будується з окремих, незалежних компонентів, кожен з яких відповідає за конкретну частину інтерфейсу користувача. Компоненти є основними будівельними блоками, які дозволяють ефективно організувати структуру програми, розділяючи її на логічні частини. Це значно спрощує розробку, тестування та повторне використання коду.

Кожен компонент складається з кількох ключових елементів: HTML-шаблону, що визначає, як виглядає частина інтерфейсу; CSS або SCSS стилів, які

забезпечують візуальне оформлення; TypeScript-класу, де реалізовано логіку компонента; а також метайнформації у вигляді декоратора `@Component`, який описує, як елементи компонента пов'язані між собою. Він забезпечує чітку інкапсуляцію та гнучкість у побудові сучасних веб-застосунків. Даний компонентний підхід сприяє повторному використанню коду та спрощує масштабування проєкту. Завдяки чіткій структурі та ізоляції компонентів, команда розробників може паралельно працювати над різними частинами застосунку без конфліктів.

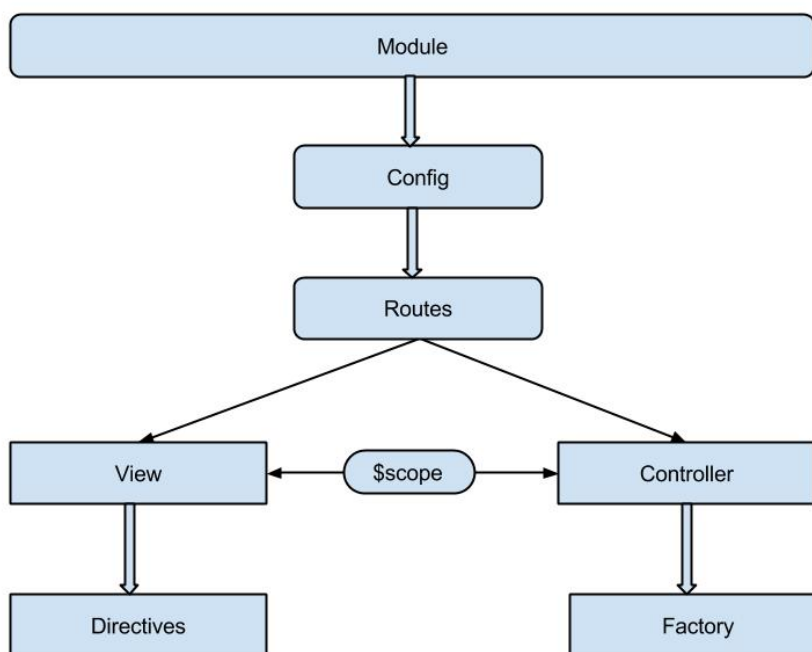


Рисунок 1.4. Структура компонентно-орієнтованого підходу

Цей підхід дозволяє повторно використовувати компоненти, полегшує тестування та спрощує оновлення частин системи без зміни всієї структури.

Структура директорій.

Проєкт організовано згідно з кращими практиками Angular. Основні директорії включають:

1. `components` – загальні UI-компоненти, що використовуються повторно (наприклад, кнопки, картки, діалоги);

2. pages – сторінки застосунку (наприклад, дашборд, результати, профіль користувача);
3. services – сервіси для взаємодії з API, обробки бізнес-логіки та спільних даних;
4. interfaces – інтерфейси TypeScript для типізації даних (DTO, моделі);
5. pipes – власні пайпи (наприклад, конвертація ватів у кіловати);
6. guards – захист маршрутів і логіка доступу до сторінок;
7. styles – глобальні та модульні SCSS-стили;
8. constants та enums – файли з константами та переліками, які полегшують супровід.

Застосунок розділений на Angular-модулі (NgModules), які логічно групують компоненти, сервіси та інші функціональні частини. Це дозволяє розділити застосунок на незалежні частини (наприклад, CoreModule, SharedModule, FeatureModule), що спрощує рефакторинг, lazy loading та пришвидшує час завантаження [4].

В Angular активно використовується інжекція залежностей (Dependency Injection) – сервіси створюються один раз і "впроваджуються" у компоненти. Це дозволяє зменшити кількість повторного коду, забезпечити централізовану логіку (наприклад, доступ до API, авторизація, кешування) та полегшує тестування.

Angular має вбудовану потужну систему маршрутизації, яка дозволяє створювати односторінкові застосунки. В даному проєкті реалізовано маршрути для основних сторінок (/home, /results, /profile тощо) з можливістю lazy loading. Також реалізовано захист маршрутів через Route Guards, які перевіряють наявність пристроїв або статус автентифікації користувача.

Маршрутизація в Angular базується на конфігураційному підході, де кожен маршрут описується у вигляді об'єкта з вказанням шляху, компонента, а також додаткових параметрів, таких як охоронці (guards) або дані для передачі. Це дозволяє гнучко керувати переходами між сторінками, зокрема забезпечувати завантаження лише необхідного контенту та обмежувати доступ до певних частин застосунку залежно від ролі користувача або інших умов.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------------|------|
| | | | | | <i>РП 08. 01 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 18 |

Окрім цього, Angular надає можливість використовувати вкладені маршрути та маршрути з параметрами, що дозволяє створювати динамічні сторінки, наприклад, перегляд деталей конкретного об'єкта за його ідентифікатором. Комбінація цих можливостей забезпечує зручну навігацію, високу продуктивність застосунку та сприяє реалізації складних сценаріїв без необхідності перезавантаження сторінки [5].

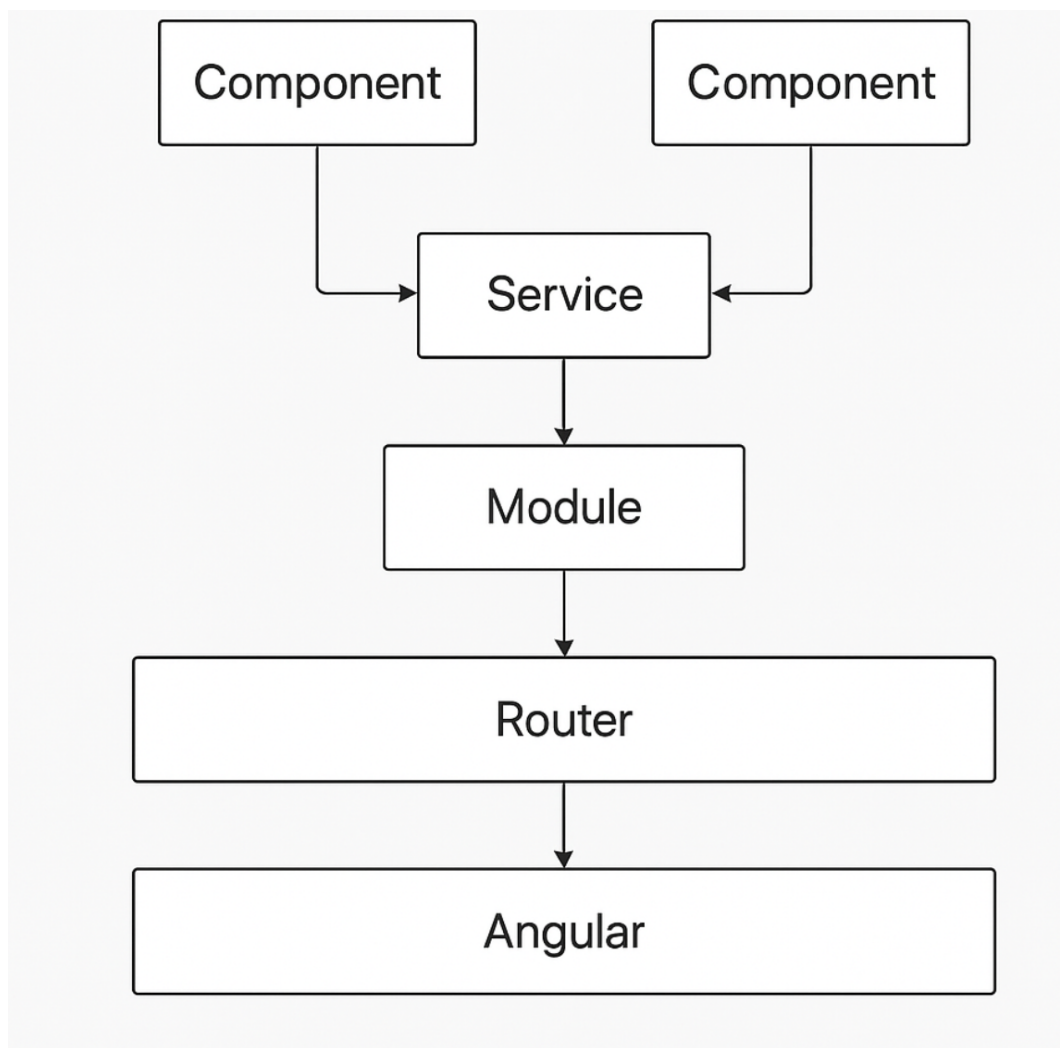


Рисунок 1.5. Блок-схема взаємодії компонентів

Опишемо переваги цієї архітектури:

1. Зрозуміла структура проєкту, що спрощує навчання та супровід;
2. Легкість у масштабуванні – можна додавати нові сторінки, модулі чи функції без впливу на інші частини;
3. Висока гнучкість – можливість повторного використання компонентів та сервісів;

4. Тестованість – окремі компоненти, сервіси та пайпи можна легко протестувати завдяки ізоляції;
5. Оптимізація завантаження – завдяки lazy loading та розділенню модулів.

1.2.3 Проектування лендінг сторінки веб-застосунку.

Одним із ключових елементів веб-застосунку є лендінг-сторінка – перша сторінка, яку бачить користувач при вході на сайт. Саме вона формує перше враження про систему, визначає подальшу зацікавленість користувача у взаємодії із сервісом та має на меті залучення і конверсію відвідувачів у активних користувачів.

Проектування лендінг-сторінки здійснювалось із урахуванням маркетингових і комунікаційних цілей проєкту. Її завдання – чітко, візуально привабливо та інформативно донести суть застосунку, його цінність, основні функціональні можливості та переваги свідомого споживання електроенергії.

Типова структура сторінки в проєкті побудована з урахуванням принципу "F-моделі" перегляду, що відображає природний шлях руху очей користувача по екрану – зверху вниз і зліва направо. Впровадження дозволяє розміщувати ключові елементи інтерфейсу в місцях, де ймовірність їхнього перегляду найбільша, тим самим підвищуючи ефективність взаємодії з користувачем.

На самому початку сторінки зазвичай розміщується головний заголовок, який коротко і чітко пояснює призначення сервісу. Поруч або трохи нижче розташований підзаголовок або слоган, що поглиблює зміст і викликає інтерес. Далі йде заклик до дії (СТА), наприклад, “Спробувати зараз” або “Розпочати моніторинг”, який спрямовує користувача до виконання цільової дії. Ці елементи часто супроводжуються ілюстраціями або анімацією, що допомагають краще сприйняти інформацію та привертають увагу.

Нижче зазвичай розміщуються блоки з описом основних переваг чи функцій сервісу, структуровані у вигляді 2–4 окремих секцій. Додатково можуть бути присутні відгуки користувачів або соціальні докази, які підвищують довіру до продукту. Завершує сторінку футер, де міститься додаткова інформація: політика конфіденційності, контактні дані, посилання на соціальні мережі та інші корисні

| | | | | | | |
|-----|------|----------|--------|------|--------------------------------|------|
| | | | | | РП 08. 01 001. 00 ДП ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 20 |

ресурси. Многівнева структура алгоритму дозволяє ефективно донести ключові меседжі, за допомогою запису до серверу та створити зручний шлях для користувача.

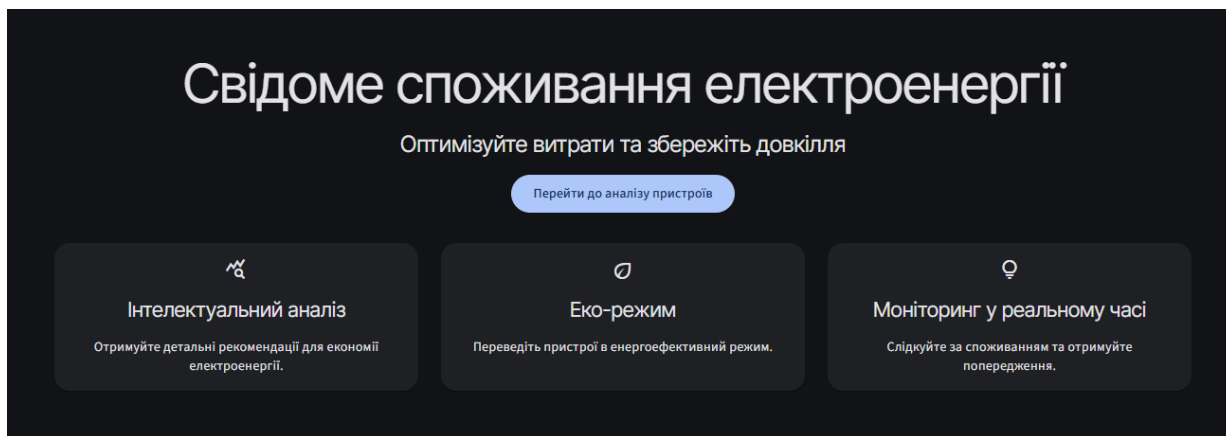


Рисунок 1.6. Огляд лендінг сторінки веб застосунку

Лендінг створено у сучасному мінімалістичному стилі з акцентом на зручність сприйняття інформації, адаптивність під різні пристрої та легку анімацію для привернення уваги користувача. Основою інтерфейсу слугували компоненти Angular Material, що дозволило швидко реалізувати сіткову структуру (layout), забезпечити адаптивність без написання додаткової логіки та впровадити інтерактивні елементи – такі як кнопки, діалоги та тултіпи – без перевантаження власними стилями.

Візуальна частина лендінгу розроблялась з особливою увагою до контрастності кольорів, достатніх відступів між елементами («повітря») та чіткої ієрархії заголовків, що загалом забезпечує приємний і зрозумілий користувацький досвід.

Під час створення дизайну лендінг-сторінки ключову увагу було приділено першому враженню користувача. Відомо, що протягом перших трьох секунд користувач приймає рішення — залишитись на сайті чи закрити його. Саме тому візуальна структура початкового екрану побудована так, щоб одразу дати відповідь на запитання «що це?». Заголовок, підзаголовок і візуальні елементи чітко демонструють призначення сервісу та його цінність.

Важливою умовою стало забезпечення інтуїтивності взаємодії. Усі кнопки, посилання та інформаційні блоки розміщені логічно і зрозуміло — користувач без додаткових пояснень може орієнтуватися в інтерфейсі. Структура побудована так,

| | | | | | | |
|-----|------|----------|--------|------|-------------------------|------|
| | | | | | РП 08. 01 001. 00 ДП ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 21 |

щоб не перевантажувати увагу: основна інформація подається у вигляді компактних секцій з чіткими заголовками та іконками, що полегшує сприйняття.

Особливий акцент зроблено на головній кнопці заклику до дії (СТА), яка повинна бути помітною одразу після відкриття сторінки, без необхідності прокручування. Вона візуально відокремлена, має контрастний колір і супроводжується текстом, що чітко пояснює наступний крок для користувача. Цей алгоритм забезпечує ефективну конверсію відвідувача у активного користувача застосунку.

1.2.2 Проєктування дизайну веб-застосунку.

У сучасній веб-розробці дизайн відіграє не менш важливу роль, ніж функціональність. Від того, наскільки зручно, зрозуміло й привабливо виглядає інтерфейс, залежить ефективність взаємодії користувача із застосунком. Саме тому проєктування дизайну веб-застосунку є невід’ємним етапом загального процесу розробки.

У ході створення інтерфейсу веб-застосунку було дотримано основних принципів UI/UX-дизайну, зокрема:

1. Простота – інтерфейс має бути інтуїтивно зрозумілим без необхідності вивчення інструкцій;
2. Зручність навігації – логічна структура сторінок і зрозуміла маршрутизація;
3. Консистентність – єдиний стиль усіх елементів, повторювані шаблони для однакових дій;
4. Адаптивність – підтримка коректного відображення на різних пристроях (ПК, планшети, смартфони);
5. Доступність – врахування потреб користувачів із різними можливостями.

Використання Angular Material.

Для реалізації візуальної частини проєкту було обрано Angular Material – офіційну бібліотеку UI-компонентів для Angular. Вона містить готові компоненти (кнопки, форми, таблиці, діалоги, іконки тощо), які відповідають рекомендаціям Google щодо Material Design.

Це дозволило:

| | | | | | | |
|-----|------|----------|--------|------|--------------------------------|------|
| | | | | | РП 08. 01 001. 00 ДП ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 22 |

- зменшити час на створення інтерфейсу;
- забезпечити кросбраузерність;
- автоматично отримати адаптивність і доступність елементів;
- підтримувати чистий і сучасний зовнішній вигляд застосунку.

Окрім технічних переваг, Angular Material сприяє уніфікації стилю застосунку, забезпечуючи послідовність зовнішнього вигляду на всіх сторінках. Завдяки гнучкій системі та адаптування інтерфейсу під брендovanі вимоги без значних змін у коді.

Також важливою є підтримка доступності (accessibility) – всі компоненти спроектовані з урахуванням потреб користувачів з обмеженими можливостями, що відповідає сучасним стандартам інклюзивного дизайну. Це робить Angular Material не лише ефективним, а й етичним вибором для побудови інтерфейсів. Інтерфейс застосунку побудовано з урахуванням розподілу на логічні блоки:

Шапка (header) – містить логотип, навігаційне меню, перемикач теми або мови.

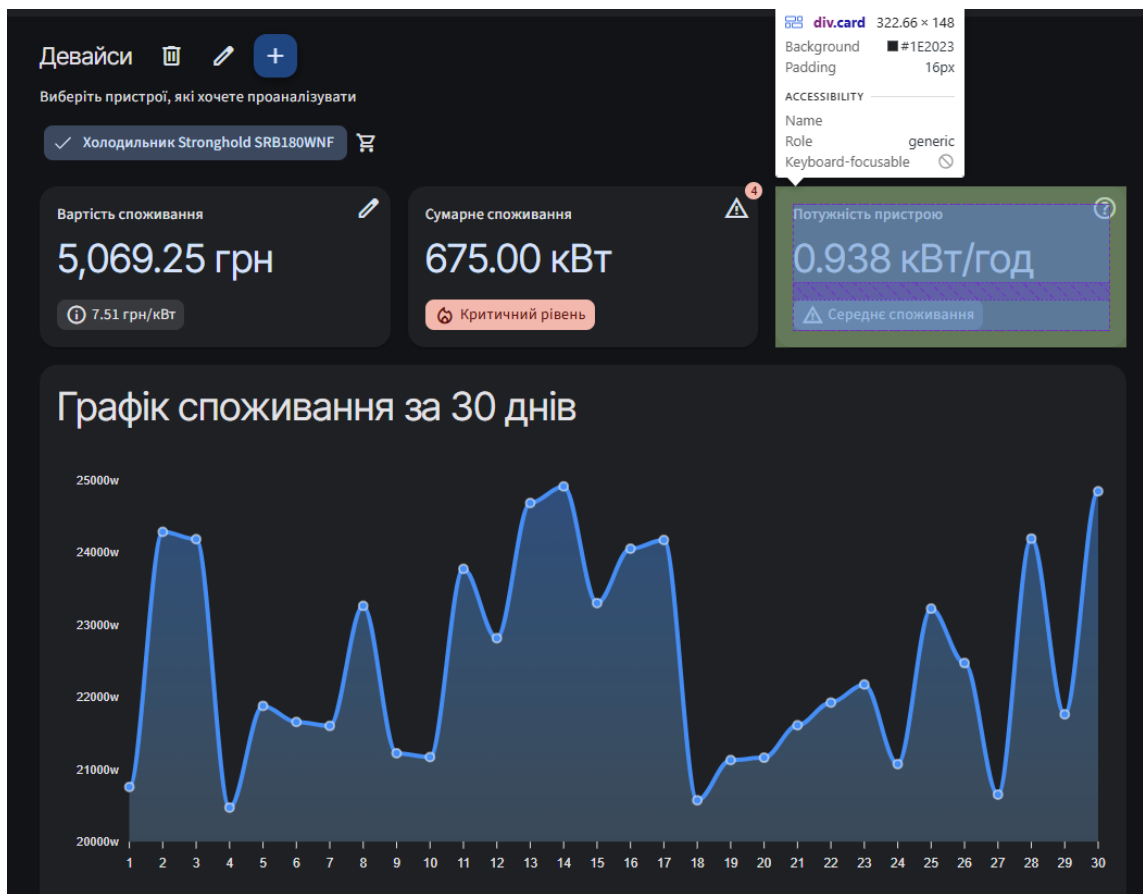


Рисунок 1.7. Структура контенту сайту

Основна частина (content) – динамічний вміст, який змінюється залежно від вибраної сторінки або маршруту. Вона може містити різні типи контенту, зокрема інтерактивні аналітичні панелі, форми введення даних, графіки, таблиці, картки з інформацією тощо. Контент організовано відповідно до принципів UX-дизайну: зручна структура, візуальні акценти на ключових даних, адаптація до різних розмірів екранів. Завдяки Angular компонентам, основний вміст легко оновлюється без перезавантаження сторінки, що забезпечує швидку та плавну взаємодію.

Бокове меню (sidenav) – призначене для швидкого доступу до розділів, налаштувань або додаткових функцій застосунку. Воно є опційним і за потреби може згорнутися, щоб не займати багато простору на екрані. Для мобільних пристроїв меню реалізовано за допомогою функціонального бокового висувного блоку, що відкривається по кліку або свайпу, забезпечуючи зручну навігацію без перевантаження інтерфейсу.

Футер (footer) – контактна інформація або службові посилання.

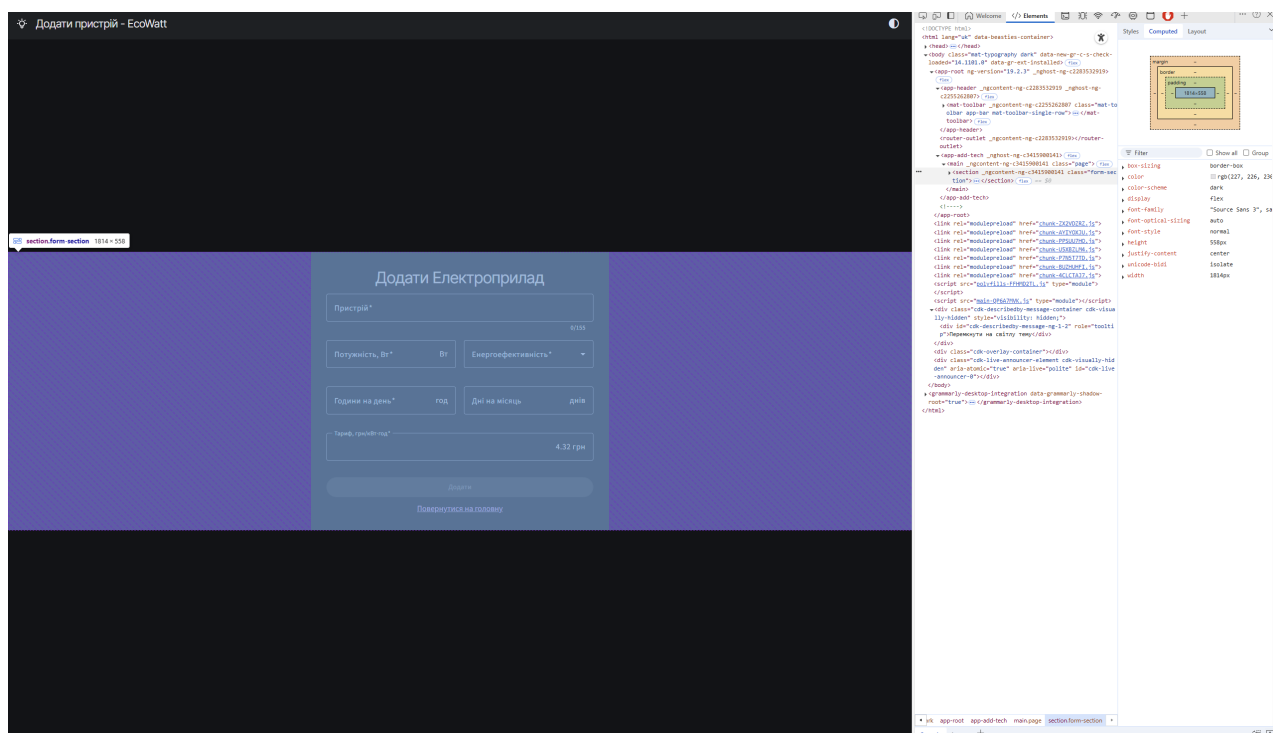


Рисунок 1.8. Приклад основної секції застосунку

У процесі розробки окремих компонентів велика увага приділялась тестуванню зручності використання інтерфейсу з позиції кінцевого користувача. Основна мета полягала в тому, щоб забезпечити не лише функціональність, а й

| | | | | | | |
|-----|------|----------|--------|------|--------------------------------|------|
| | | | | | РП 08. 01 001. 00 ДП ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 24 |

комфортну взаємодію зі сторінками застосунку, незалежно від технічної підготовки користувача.

Одним із перших аспектів, який перевірявся, була контрастність та читабельність тексту. Усі елементи було протестовано на відповідність принципам доступності — колір фону та шрифтів підбрано так, щоб текст залишався легко читабельним навіть за слабкого освітлення. Крім того, тестувалась реакція елементів на взаємодію: наприклад, чи чітко видно ефекти наведення миші (hover), фокусування та натискання.

Окрему увагу було приділено поведінці форм під час помилкового введення. Перевірялась наочність повідомлень про помилки, правильне підсвічування проблемних полів і швидкість відгуку інтерфейсу на зміну даних. Також аналізувався час, за який користувач може знайти та використати основні функції — це дало змогу удосконалити логіку навігації та зробити інтерфейс ще зручнішим.

Візуальний стиль застосунку був обраний з урахуванням тематики свідомого енергоспоживання. Було використано світлу кольорову палітру з акцентами зеленого та синього кольорів, які асоціюються з енергією, технологічністю та екологічністю. Іконки та графічні елементи мають мінімалістичний вигляд, що відповідає сучасним тенденціям у UI-дизайні.

1.2.4 Технології тестування клієнтської частини.

Забезпечення якості програмного забезпечення є одним із найважливіших етапів життєвого циклу розробки веб-застосунку.

Особливу увагу необхідно приділяти тестуванню клієнтської частини, оскільки саме інтерфейс користувача є основною точкою взаємодії з кінцевим користувачем. Для тестування фронтенду на Angular існує ряд ефективних технологій та інструментів, що дозволяють автоматизувати перевірку функціональності, виявляти помилки на ранніх етапах розробки та підвищувати стабільність роботи системи.

Для автоматизації тестування в Angular часто використовують такі інструменти, як Jasmine і Karma, які дозволяють створювати модульні та інтеграційні тести для компонентів і сервісів. Protractor, як інструмент для end-to-

| | | | | | | |
|-----|------|----------|--------|------|--------------------------------|------|
| | | | | | <i>РП 08. 01 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 25 |

end тестування, забезпечує перевірку поведінки застосунку з точки зору користувача, імітуючи реальні сценарії взаємодії. Впровадження безперервної інтеграції (CI) з тестовими пайплайнами допомагає автоматично запускати тести при кожній зміні коду, що знижує ризик регресій. Крім того, використання технік покриття коду (code coverage) дозволяє оцінити, наскільки повно протестовано застосунок, і виявити потенційні слабкі місця. Такий комплексний підхід до тестування гарантує високу якість фронтенду та підвищує довіру користувачів до продукту.

Jasmine – популярний фреймворк для написання юніт-тестів JavaScript/TypeScript коду. Він дозволяє організовувати тестові сценарії у вигляді специфікацій (describe/it блоки), що полегшує читання та підтримку тестів.

Основні особливості Jasmine:

- підтримка незалежного тестування функцій, сервісів та компонентів;
- наявність асерцій для перевірки очікуваної поведінки;
- можливість створення шпигунів (spies) для відстеження викликів функцій;
- мінімальні зовнішні залежності.

Jasmine активно інтегрований в середовище розробки Angular за замовчуванням через Angular CLI.

Karma – тест-раннер для запуску юніт-тестів у реальному браузері.

Поєднання Karma та Jasmine дозволяє створювати автоматизоване середовище тестування клієнтської частини, де кожна зміна в коді оперативно перевіряється на відсутність регресій.

1.2.5 Маршрутизація

Роутинг є невід’ємною частиною побудови сучасних веб-застосунків, особливо односторінкових (SPA, Single Page Applications), де навігація між різними візуальними станами відбувається без повного перезавантаження сторінки. У контексті Angular роутинг являє собою процес визначення відповідності між URL-адресами та компонентами, які повинні бути завантажені у відповідь на запит користувача.

У проєкті було використано Angular Router для організації переходів між

| | | | | | | |
|-----|------|----------|--------|------|--------------------------------|------|
| | | | | | <i>РП 08. 01 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 26 |

основними сторінками. Конфігурація маршрутизації виглядає наступним чином
маємо наступний код:

```
import { Routes } from '@angular/router';
import { AppRoutes } from './enums/app-routes';
import { deviceNotExistGuard } from './guards/devices-not-exist/device-not-exist.guard';

export const routes: Routes = [
  {
    path: AppRoutes.Landing,
    loadComponent: () =>
      import('./pages/landing/landing.component').then(
        (c) => c.LandingComponent
      ),
    title: 'EcoWatt'
  },
  {
    path: AppRoutes.AddGeneratedComponent,
    loadComponent: () =>
      import('./pages/generated-component/add-device.component').then(
        (c) => c.AddDeviceComponent
      ),
    canActivate: [deviceNotExistGuard]
  },
  {
    path: AppRoutes.AddDevice,
    loadComponent: () =>
      import('./pages/add-device/add-device.component').then(
        (c) => c.AddDeviceComponent
      ),
    title: 'Додати пристрій - EcoWatt',
    canActivate: [deviceNotExistGuard]
  },
  {
    path: AppRoutes.ViewResults,
    loadComponent: () =>
      import('./pages/view-results/view-results.component').then(
        (c) => c.ViewResultsComponent
      ),
    title: 'Споживання - EcoWatt'
  },
  {
    path: '**',
    redirectTo: 'landing'
  }
];
```

Landing Page (/landing) – початкова сторінка сайту.

Add Device Page (/add-device) – сторінка додавання нового пристрою, захищена гардом deviceNotExistGuard, що дозволяє обмежити доступ за певними умовами.

View Results Page (/view-results) – сторінка для перегляду результатів споживання електроенергії.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------------|------|
| | | | | | <i>РП 08. 01 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 27 |

Маршрут за замовчуванням ('**') перенаправляє будь-які невідомі шляхи на головну сторінку (landing).

Також розглянемо, чому ми виикористовували, маршрутизацію, за допомогою ротінгу:

По перше, це висока інтеграція з основними механізмами Angular, що забезпечує кращу продуктивність та стабільність. Lazy Loading компонентів дозволяє зменшити початковий об'єм завантаження застосунку. Гнучка система захисту маршрутів через використання гард-сервісів. У проекті також використовиваємо можливість динамічного налаштування мета-даних сторінок (наприклад, заголовків через параметр title). Також простота підтримки і масштабування застосунку за рахунок чіткої конфігурації маршрутів.

На рис. 1.8 зображено навігаційну мапу маршрутизації.



Рисунок 1.9. Навігаційна мапа маршрутизації

1.2.6 Проектування структури бази даних

Для зберігання та обробки даних у веб-застосунку EcoWatt було спроектовано логічно структуровану реляційну базу даних. Основна мета при проектуванні бази даних полягала в забезпеченні надійного, ефективного та масштабованого способу зберігання інформації, пов'язаної з користувачами, пристроями, джерелами енергії, тарифами та щоденними показниками споживання. Уся структура була побудована з урахуванням нормалізації даних і забезпечення зв'язків між таблицями на рівні зовнішніх ключів.

У центрі моделі знаходиться таблиця users, яка містить основну інформацію про користувача: адресу електронної пошти (email) та хешований пароль (password_hash). Кожен користувач має унікальний ідентифікатор (id), що

використовується як зовнішній ключ у всіх пов'язаних таблицях. Ця таблиця є основою для авторизації та аутентифікації в системі, а також для зв'язку з усіма іншими об'єктами, пов'язаними із взаємодією з веб-застосунком.

Завдяки використанню таблиці users як центрального елементу, система забезпечує чітке розмежування даних між користувачами, що дозволяє зберігати індивідуальні налаштування, персоналізовану статистику та прив'язувати кожен пристрій, показник чи джерело енергії до конкретного облікового запису. Зберігання даних паролю у вигляді хеша не лише підвищує рівень безпеки й організованості даних, а й відкриває можливості для подальшої персоналізації функціоналу бази даних це наприклад, впровадження різних ролей користувачів, локалізації даних або фільтрації аналітики за профілем.

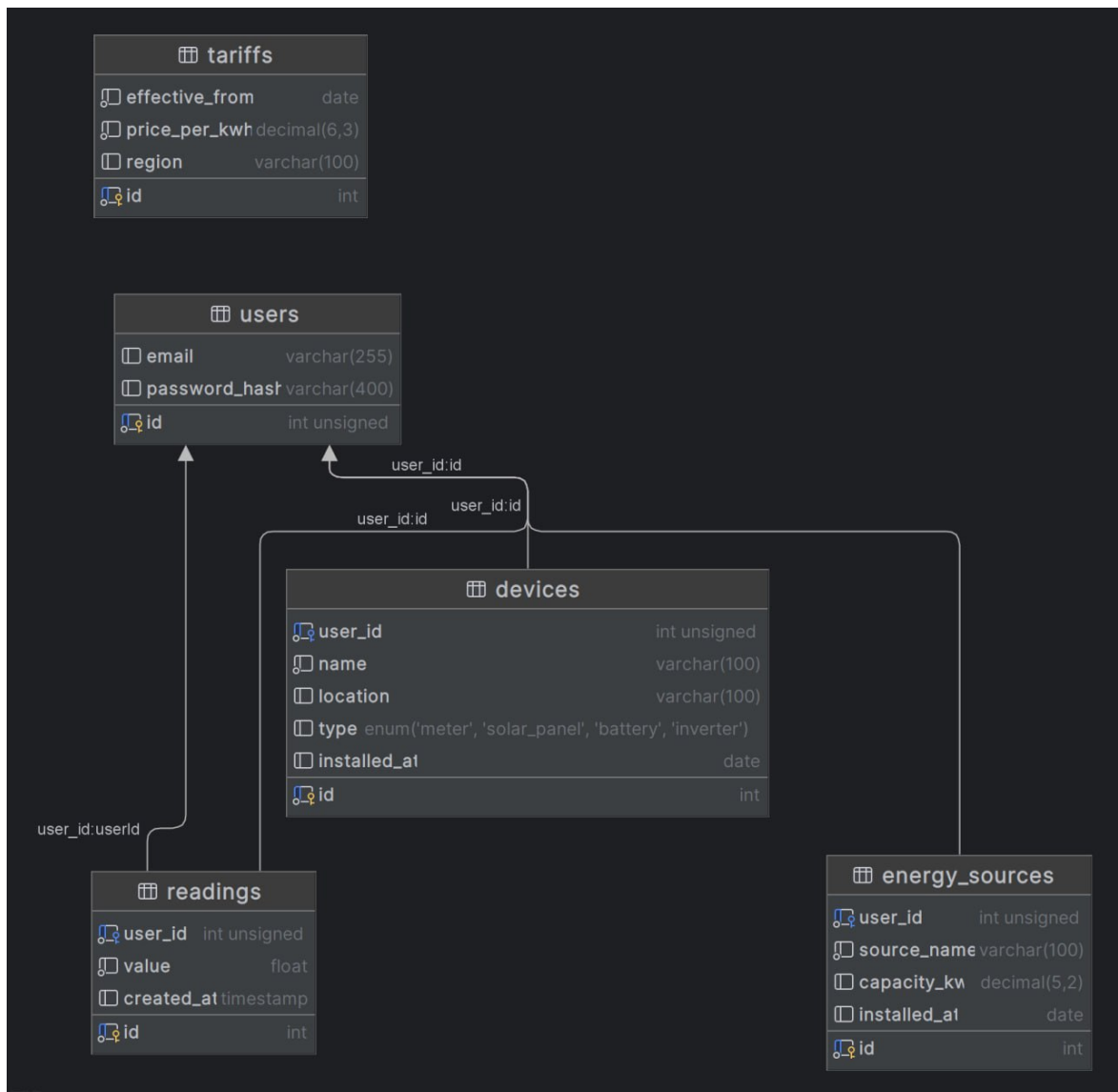


Рисунок 1.10. Структура бази даних

Таблиця `devices` є однією з центральних у структурі, оскільки зберігає дані про всі пристрої, які додає користувач. Вона включає такі поля, як `name` (назва пристрою), `location` (місце розташування), `type` (тип пристрою, обмежений ENUM-значеннями: 'meter', 'solar_panel', 'battery', 'inverter'), а також `installed_at` — дата встановлення. Кожен пристрій прив'язаний до конкретного користувача через поле `user_id`. Використовуючи обробку сховища даних ми можемо формувати для кожного користувача унікальний набір пристроїв, що полегшує подальший аналіз та візуалізацію даних.

Таблиця `readings` виконує роль журналу показників енергоспоживання або генерації. Кожен запис у таблиці містить такі поля: `user_id` (ідентифікатор користувача), `value` (значення споживання або генерації, типу `float`), `created_at` (мітка часу запису). Кожен запис пов'язаний із користувачем і фіксує динаміку енергоспоживання, що критично важливо для побудови графіків та розрахунків у застосунку. У таблиці можливо вести історію змін, аналізувати тренди, а також реагувати на аномалії в режимі реального часу.

Оглянемо таблицю `tariffs`, що зберігає актуальну інформацію про вартість електроенергії в певному регіоні. Вона включає дату набуття чинності тарифу (`effective_from`), вартість у гривнях за кіловат-годину (`price_per_kwh`), регіон (`region`) та унікальний ідентифікатор (`id`). Хоча в даній реалізації ця таблиця не має прямого зовнішнього зв'язку, її можна використовувати для перехресного аналізу або розширити в подальших версіях системи для персоніфікованих тарифів.

Таблиця `energy_sources` відповідає за зберігання даних про джерела енергії, які може додавати користувач, наприклад, сонячні панелі, акумулятори тощо. Вона включає такі поля: `source_name` (назва джерела), `capacity_kw` (потужність у кіловатах), `installed_at` (дата встановлення), а також зовнішній ключ `user_id`, що вказує на власника. Для майбутньої інтеграції з системами прогнозування генерації та аналізу ефективності використання відновлюваних джерел енергії, ми будемо використовувати данні саме отсюди.

Загальна модель бази даних побудована з дотриманням принципів 2NF та 3NF, що дозволяє уникнути дублювання інформації, зменшити обсяг пам'яті та

| | | | | | | |
|-----|------|----------|--------|------|--------------------------------|------|
| | | | | | <i>РП 08. 01 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 30 |

спростити логіку запитів. Зовнішні ключі між таблицями забезпечують цілісність даних, а також дозволяють швидко виконувати приєднання (JOIN) між таблицями при вибірках. Хочу наголосити, що по перше треба покращити ефективність для масштабування системи в майбутньому, коли кількість користувачів і пристроїв зростатиме.

Окремої уваги заслуговує тип даних ENUM у таблиці devices, який обмежує можливі значення типів пристроїв до фіксованого набору. Вона забезпечує консистентність у базі даних і полегшує реалізацію фільтрації та сортування на стороні клієнта. Аналогічно, для числових полів, таких як capacity_kw або value, були обрані типи decimal та float, що забезпечують точність при розрахунках.

Завдяки чіткій структурі, використанню зовнішніх ключів та дотриманню сучасних практик нормалізації, ця модель здатна витримувати високі навантаження та ефективно працювати в умовах реального виробничого середовища.

1.2.7 Проектування серверної частини для надання допомоги користувачеві

Серверна частина веб-застосунку EcoWatt виконує низку критичних функцій, зокрема обробку авторизації, зберігання показників споживання, надання статистики, захист доступу до ресурсів і підтримку персоналізованих запитів. Уся логіка реалізована мовою PHP з використанням шаблону MVC (Model–View–Controller), що забезпечує розділення відповідальностей, підтримуваність та розширюваність архітектури. У цьому підрозділі розглянемо основні елементи серверної частини та пояснимо, як саме вони сприяють підвищенню якості користувацького досвіду.

За обробку автентифікації відповідає клас AuthController, що містить методи login, register і logout. Логіка авторизації передбачає перевірку HTTP-запиту типу POST, зчитування email та пароля, та звернення до моделі UserModel для пошуку відповідного запису. Якщо обліковий запис знайдено та пароль вірний (перевірка здійснюється через password_verify), до якої користувач ідентифікується, і в сесії зберігається його user_id. Далі здійснюється редирект на головну сторінку

| | | | | | | |
|-----|------|----------|--------|------|--------------------------------|------|
| | | | | | РП 08. 01 001. 00 ДП ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 31 |

застосунку. У разі помилки повертається JSON-відповідь із відповідним кодом (401) та текстом помилки.

Код методу register виконує перевірку, чи вже існує користувач із такою адресою, і в разі відсутності, а саме створює нового, хешуючи пароль за допомогою password_hash. Це забезпечує базовий рівень безпеки, адже паролі не зберігаються у відкритому вигляді. Усі помилки обробляються, і користувач отримує JSON-відповідь, з якої одразу зрозуміло, що саме пішло не так, також користувачу допомагає візуальний функціонал.

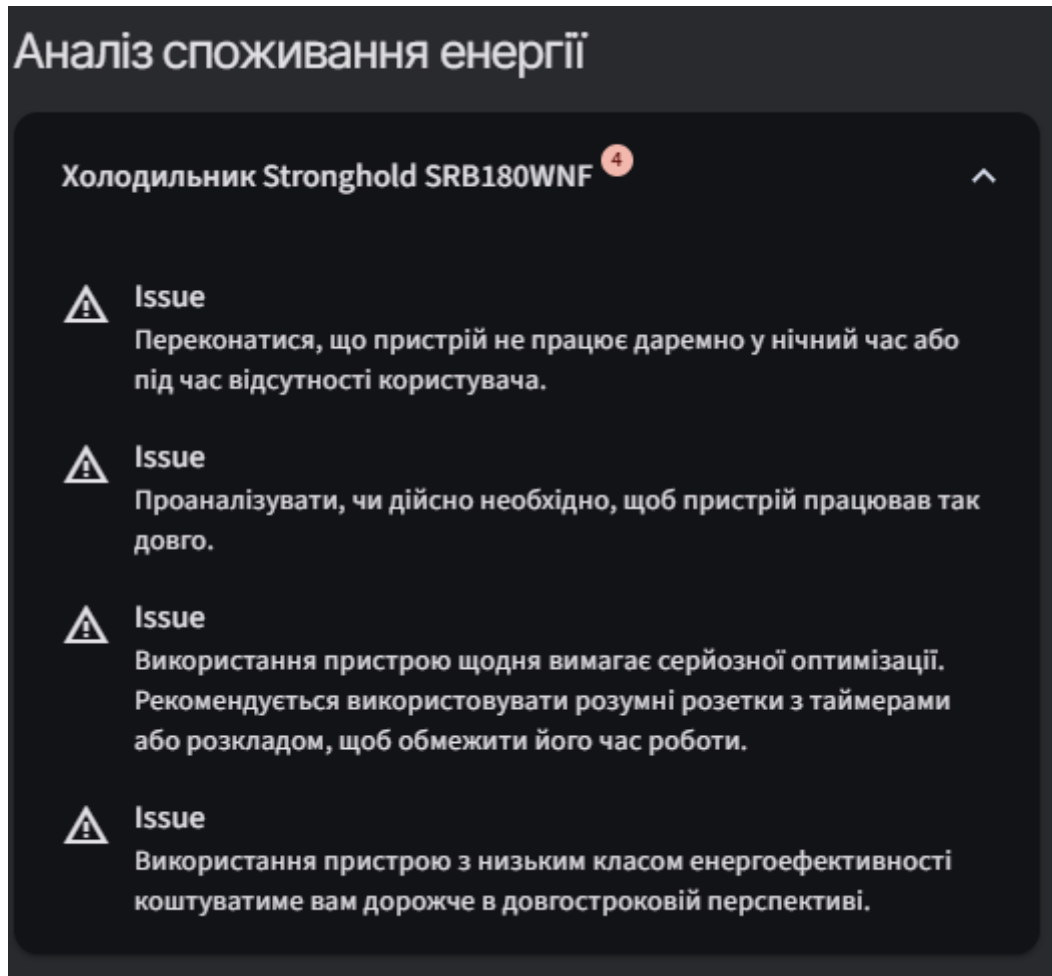


Рисунок 1.11. Функціонал для показу помилок

Один із ключових функціоналів EcoWatt - це прийом і аналіз щоденних показників енергоспоживання. Для цього створено клас ReadingController, який має два методи: addReading та getLatestReadings. Метод addReading забезпечує прийом POST-запиту, перевірку авторизації через сесію, валідацію значення та подальше звернення до моделі ReadingModel::addReading, яка здійснює вставку в таблицю readings.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------------|------|
| | | | | | <i>РП 08. 01 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 32 |

Особливу увагу приділено захисту даних: якщо користувач неавторизований або надсилає запит у неправильному форматі, сервер повертає відповідні коди помилок (403, 400, 405) і зрозумілі повідомлення. Завдяки цьому забезпечується не лише зручність взаємодії, а й стійкість системи до некоректних запитів або зловмисного втручання. Це відповідає сучасним вимогам до якості енергетичних сервісів, які передбачають «надання користувачу достовірної та своєчасної інформації про обсяг спожитої електроенергії та її вартість» [6].

Метод `getLatestReadings` відповідає за отримання останніх десяти показників користувача з можливістю обрахунку динаміки та вартості. Якщо у запиті додано параметр `price`, система автоматично обчислює витрати за кожен день, порівнюючи значення з попереднім. Результат подається у вигляді JSON-масиву, що включає дату, показник, різницю з попереднім значенням та вартість. Обробка допомагає клієнтській частини яка може будувати графіки без додаткової обробки.

Архітектура, коли бізнес-логіка винесена на сервер, а клієнт лише відображає підготовлені дані, значно полегшує підтримку застосунку. Крім того, воно допомагає гнучко змінювати алгоритми розрахунків без необхідності оновлення інтерфейсу у користувача. Такий підхід також узгоджується з напрямками цифровізації, закладеними в Енергетичній стратегії України, згідно з якою «розвиток інтелектуальних систем обліку є ключовим елементом модернізації енергетичної інфраструктури» [7].

Модель `UserModel` реалізує методи `create` та `findByEmail`, що відповідають за створення облікового запису та пошук користувача. В обох випадках використовуються параметризовані SQL-запити з підключенням до бази даних через об'єкт `PDO`, що мінімізує ризик SQL-ін'єкцій. Модель зберігається відповідно до сучасним стандартам безпеки.

Модель `ReadingModel` реалізує вставку нових показників (`addReading`) та отримання останніх значень (`getLatestReadings`) із таблиці `readings`. Дані сортуються за датою, а результат інвертується (`array_reverse`) для зручності виведення у хронологічному порядку. Обидва методи виконуються через підготовлені запити, що гарантує високу швидкодію та надійність.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------------|------|
| | | | | | <i>РП 08. 01 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 33 |

Файл `index.php`, розташований у директорії `client/public`, відповідає за маршрутизацію HTTP-запитів. У ньому відбувається розбір URL і передача запитів до відповідних методів контролерів. Зокрема, URL `/login`, `/register`, `/logout`, `/readings/add` та `/readings/latest` передаються до методів відповідних контролерів. У разі введення невідомого маршруту користувач бачить повідомлення 404 Not Found. Простий роутер ідеально підходить для невеликого застосунку та може бути легко розширений у разі потреби.

Файл `database.php` встановлює з'єднання з базою даних MySQL через об'єкт PDO, налаштовує кодування UTF-8 та режим викидання виключень (`ERRMODE_EXCEPTION`). У разі помилки підключення користувач отримає повідомлення з поясненням проблеми, що спрощує діагностику. Усі моделі використовують це підключення глобально, що дозволяє уникати повторного створення з'єднання в кожному методі окремо.

1.3 Тестування веб-застосунку

1.3.1 Ручне тестування

Ручне тестування є невід'ємною складовою забезпечення якості розробленого веб-застосунку. Його основною метою є виявлення помилок та дефектів, що можуть негативно впливати на працездатність системи, стабільність роботи або користувацький досвід. На відміну від автоматизованого тестування, ручне тестування виконується тестувальником без застосування спеціальних програм для автоматизації процесу перевірки.

У межах даної роботи було виконано ручне тестування створеного веб-застосунку. Основна увага приділялась перевірці правильності переходів між сторінками відповідно до налаштованої маршрутизації: з головної сторінки (Landing Page) на сторінку додавання пристрою (Add Device Page) та сторінку перегляду результатів (View Results Page). Також було протестовано захист доступу до сторінок за допомогою гард-сервісу `deviceNotExistGuard`, який обмежує доступ у разі відсутності необхідних даних.

Додатково перевірено правильність роботи форм на сторінці додавання

| | | | | | | |
|-----|------|----------|--------|------|--------------------------------|------|
| | | | | | <i>РП 08. 01 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 34 |

пристрою, включаючи обробку обов'язкових полів і валідацію введеної інформації. Окремо протестовано динамічне оновлення інтерфейсу без перезавантаження сторінки завдяки двосторонній прив'язці даних Angular. Була проведена перевірка коректності зміни теми застосунку через сервіс theme, де підтверджено правильну роботу механізму переключення стилів оформлення. Також перевірено встановлення заголовків сторінок через сервіс page-title відповідно до маршруту переходу.

Ручне тестування дозволило виявити кілька дрібних недоліків, що стосувалися відображення елементів на різних розширеннях екранів. Ці помилки були оперативно усунені шляхом доопрацювання SCSS-стилів.

Ручне тестування залишається важливою стадією життєвого циклу розробки програмного забезпечення навіть за умов широкого поширення автоматизованих підходів. Воно дає змогу виявити проблеми, що пов'язані з реальним користувацьким досвідом, оцінити зручність інтерфейсу, логічність навігації та читабельність контенту, а також оперативно перевірити специфічні або складні сценарії використання, які складно або недоцільно автоматизувати. Крім того, ручне тестування дозволяє швидко адаптуватися до змін у вимогах без необхідності переписувати автоматизовані тестові скрипти.

Результати проведеного ручного тестування підтвердили коректність реалізованої функціональності веб-застосунку та його готовність до подальшого використання кінцевими користувачами.

1.3.2 Функціональний огляд головної сторінки

Розглянемо головну сторінку веб-застосунку EcoWatt, яка виконує важливу роль у першій взаємодії користувача із системою. Саме на цьому етапі формується перше враження про сервіс та визначається подальша зацікавленість користувача у його функціональних можливостях.

Після завантаження головної сторінки користувач бачить привітальне повідомлення "Свідоме споживання електроенергії", яке супроводжується закликом до дії – "Оптимізуйте витрати та збережіть довкілля". Це рішення позитивно впливає на користувацький досвід, створюючи емоційно привабливий

| | | | | | | |
|-----|------|----------|--------|------|--------------------------------|------|
| | | | | | <i>РП 08. 01 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 35 |

перший контакт із застосунком. Привітання реалізовано у блоці `.head-container`, що структурно налаштований через стилі:

```
.head-container {
  width: 100%;
  text-align: center;
  display: flex;
  flex-direction: column;
  gap: 1rem;
  justify-content: center;
  align-items: center;
}
```

Текст та заклик до дії займають центральне положення і мають привабливий візуальний вигляд незалежно від розміру екрану.

Далі на головній сторінці розміщується кнопка дії – "Перейти до аналізу пристроїв", що веде користувача до сторінки додавання пристроїв (`/add-device`). Цей елемент створений за допомогою компонента `MatAnchor` і маршрутизований через `RouterLink`, як видно з коду:

```
readonly analyzeLink = ["", AppRoutes.AddDevice];
```

Кнопка інтегрована у блок `.action-section`, який стилізований для центрування та забезпечення зручного доступу:

```
.action-section {
  display: flex;
  flex-direction: column;
  align-items: center;
  gap: 0.5rem;
  text-align: center;
}
```

Нижче на сторінці розташовані три тематичні картки-функціональні блоки, які презентують основні переваги застосунку:

Інтелектуальний аналіз – пропонує користувачеві отримувати детальні рекомендації щодо оптимізації енергоспоживання;

Еко-режим – мотивує переводити пристрої у більш енергоефективний режим;

Моніторинг у реальному часі – інформує про можливість відстеження споживання електроенергії та отримання попереджень.

Ці картки реалізовані як окремі компоненти `LandingCardComponent` і організовані у блоці `.card-section`, який використовує `CSS Grid` для адаптивного розташування:

| | | | | | | |
|-----|------|----------|--------|------|--------------------------------|------|
| | | | | | <i>РП 08. 01 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 36 |

```
.card-section {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(18.75rem, 1fr));
  gap: 1.5rem;
  width: 100%;
}
```

Сторінка залишається естетично привабливою на різних пристроях та дозволяє користувачу швидко ознайомитися з можливостями веб-застосунку.

Завдяки використанню сучасних підходів Angular, компонент LandingComponent не лише забезпечує ефективну організацію коду, а й сприяє високій продуктивності та масштабованості проєкту. Інкапсуляція через standalone-компоненти дозволяє легко підтримувати та повторно використовувати окремі частини інтерфейсу, а стратегія OnPush мінімізує зайві перерендери, що особливо важливо для продуктивності на мобільних пристроях.

```
@Component({
  selector: 'app-landing',
  imports: [MatAnchor, LandingCardComponent, RouterLink, MatIcon],
  templateUrl: './landing.component.html',
  styleUrls: ['./landing.component.scss'],
  standalone: true,
  changeDetection: ChangeDetectionStrategy.OnPush
})
export class LandingComponent extends BasePageDirective { ... }
```

Розглянемо наступну таблицю елементів та їх функціональностей:

У контексті реалізації компонент LandingComponent використовує сучасні підходи Angular.

Таблиця 1.2. Елементи головної сторінки та їх функціональність

| Елемент сторінки | Опис | Функціональне призначення |
|--|--|---|
| Привітальне повідомлення | Текст «Свідоме споживання електроенергії» та підзаголовок | Формує перше позитивне враження, орієнтує користувача |
| Кнопка переходу («Перейти до аналізу пристроїв») | Активна кнопка для переходу на сторінку додавання пристрою | Спрямовує користувача до основної дії в застосунку |
| Блок «Інтелектуальний аналіз» | Інформаційна картка з описом аналітичних функцій | Інформує про можливість отримання рекомендацій |
| Блок «Еко-режим» | Інформаційна картка щодо енергоефективного режиму | Мотивує переводити пристрої у еко-режим |

| Елемент сторінки | Опис | Функціональне призначення |
|------------------------------------|---|---|
| Блок «Моніторинг у реальному часі» | Інформаційна картка щодо відстеження споживання | Інформує про можливість моніторингу в реальному часі |
| Тематичне оформлення | Темна тема оформлення сайту із світлими акцентами | Покращує візуальне сприйняття та комфорт роботи з застосунком |

1.3.3 Елементи форми та функціонал додавання приладу

Розглянемо наступну сторінку веб-застосунку – "Додати електроприлад".

Веб-сторінка “Додати електроприлад” надає користувачам можливість ввести основні параметри електроприладу для подальшого аналізу споживання електроенергії. Ця форма є ключовим елементом застосунку, оскільки вона забезпечує збір вихідних даних, на основі яких розраховується витрати енергії.

Користувач може заповнити такі поля:

1. Назва пристрою – текстове поле для введення найменування електроприладу.
2. Потужність (Вт) – числове поле для вказання потужності пристрою в ватах.
3. Енергоефективність – випадаючий список для вибору класу енергоефективності (наприклад, A++, A+, A тощо).
4. Години роботи на день – поле для введення кількості годин, протягом яких пристрій працює щодня.
5. Кількість днів на місяць – поле для зазначення кількості днів активної роботи пристрою протягом місяця.
6. Тариф (грн/кВт·год) – поле для вказання актуального тарифу на електроенергію.

У нижній частині форми розташована кнопка “Додати”, яка активується тільки після правильного заповнення всіх обов'язкових полів. Якщо деякі дані введено некоректно, система миттєво повідомляє користувача про помилки через візуальні підказки та підсвічування відповідних полів.

Всі обов'язкові поля проходять валідацію як на рівні клієнта, так і на рівні логіки Angular, що дає змогу своєчасно виявити помилки ще до надсилання запиту

на сервер. Валідаційні повідомлення адаптовані під мову інтерфейсу та подаються у зрозумілій і лаконічній формі, що полегшує взаємодію для користувача.

Крім того, візуальна підсвітка полів з помилками виконана згідно з принципами доступності – використано кольорові акценти, контрастні рамки та іконки, що сигналізують про проблему. Це дозволяє швидко зорієнтуватися навіть на мобільних пристроях або в умовах обмеженої видимості.

Кнопка “Додати” лишається неактивною до моменту, поки форма не буде заповнена коректно, що стимулює користувача перевірити введену інформацію та усунути всі помилки перед завершенням дії. На рис. 1.9 зображено вигляд форми при правильному заповненні, користувач може бачити поля, та вводити дані о приладі, та вибирати клас потужності який буде считуватись алгоритмом для побудови відповіді.

The screenshot shows a form titled "Додати Електроприлад" (Add Appliance) on a dark background. The form contains several input fields, each with a label and a value:

- Пристрій***: Холодильник Stronghold SRB180WNF
- Потужність, Вт***: 750 Вт
- Енергоефективність***: F (Дуже високе споживання) ▼
- Години на день***: 24 год
- Дні на місяць***: 30 днів
- Тариф, грн/кВт-год***: 7.51 грн

At the bottom of the form, there is a large blue button labeled "Додати" (Add) and a link labeled "Повернутися на головну" (Return to home).

Рисунок 1.9. Сторінка «Додати електроприлад» (правильне заповнення)

Також реалізована розширена валідація даних. Наприклад, максимальна потужність пристрою не може перевищувати 10 000 Вт, тривалість роботи на день обмежена 24 годинами, а кількість днів у місяці – 30. Якщо користувач вводить значення, що виходять за допустимі межі, відповідні поля підсвічуються червоним кольором, а поруч з ними відображається повідомлення про помилку.

Логіка валідації дозволяє запобігти введенню нереалістичних або технічно некоректних даних, що особливо важливо при розрахунках енергоспоживання. Завдяки цьому користувач одразу бачить, які значення є припустимими, і може швидко внести корективи без необхідності перегортати всю форму. Окремий механізм не лише покращує точність розрахунків, а й значно знижує кількість помилок, що могли б призвести до некоректного аналізу або неправильних висновків у підсумковій статистиці.

Крім базових обмежень на значення, система також враховує взаємозв'язки між полями. Наприклад, якщо користувач вказує низький клас енергоефективності пристрою, але вводить високу тривалість роботи, система може попередити про потенційне перевищення споживання або запропонувати змінити налаштування. Це дозволяє валідації працювати не лише як перевірка формату, але і як інструмент для оптимізації поведінки користувача ще на етапі введення даних.

Усі повідомлення про помилки формулюються зрозумілою мовою, без технічних термінів, що робить процес заповнення зручним навіть для користувачів без спеціальної підготовки. Текст підказок може динамічно змінюватися залежно від типу помилки та конкретного параметра, що підвищує гнучкість системи валідації. Це наближає взаємодію з формою до формату «розмови», де система фактично допомагає користувачу зробити правильний вибір.

Важливо також, що валідація працює як на клієнтській, так і на серверній стороні. Це означає, що навіть якщо користувач з якоїсь причини обійде перевірку в інтерфейсі (наприклад, через ручне втручання в код сторінки), сервер все одно зможе виявити порушення і відхилити некоректні дані. Зібраний дворівневий підхід забезпечує надійність усієї системи та мінімізує ризики зберігання неконсистентної інформації в базі даних.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------------|------|
| | | | | | <i>РП 08. 01 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 40 |

Крім технічної точності, система валідації виконує ще й навчальну функцію вона не лише вказує на помилки, а й пояснює, чому введені значення можуть бути нераціональними. Наприклад, при введенні тривалості роботи понад 20 годин на день користувач отримує не сухе повідомлення про помилку, а рекомендацію переглянути режим роботи пристрою або перевірити його налаштування. Цей функціонал формує усвідомлену поведінку, підвищує енергетичну грамотність користувача та створює відчуття «діалогу» між системою та людиною.

На рис. 1.10 наведено приклад роботи системи валідації при некоректному заповненні полів.

Додати Електроприлад

Пристрій*
Холодильник 11/155

Потужність, Вт* 3522200 Вт !
Макс. 10 000

Енергоефективність*
A++ (Дуже ефективний) ▾

Години на день* 232 год !
Макс. 24 год

Дні на місяць 1231 днів !
Макс. 30 днів

Тариф, грн/кВт-год* 7.51 грн

Додати

[Повернутися на головну](#)

Рисунок 1.10. Сторінка «Додати електроприлад» (помилки заповнення форми)

Під час введення даних активно працює система перевірки правильності

| | | | | | | |
|-----|------|----------|--------|------|--------------------------------|------|
| | | | | | РП 08. 01 001. 00 ДП ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 41 |

заповнення полів. У випадку успішного заповнення даних кнопка “Додати” стає активною і дозволяє передати дані для подальшого аналізу. Якщо ж заповнення некоректне, кнопка залишається неактивною, запобігаючи надсиланню некоректної інформації.

Крім цього, реалізована можливість швидкого повернення на головну сторінку через посилання “Повернутися на головну”, що підвищує зручність користування застосунком.

Форма додавання електроприладу забезпечує інтуїтивно зрозумілий процес внесення даних та створює надійну основу для подальшого аналізу споживання електроенергії в системі EcoWatt.

1.3.4 Розрахунок вартості споживання

Однією з головних функцій веб-застосунку EcoWatt є можливість розрахунку вартості споживання електроенергії для кожного доданого користувачем пристрою. Цей функціонал реалізовано через інтеграцію декількох сервісів, що відповідають за збереження даних пристроїв, аналіз споживання та застосування тарифних ставок.

Розрахунок відбувається автоматично на основі введених користувачем параметрів, зокрема потужності пристрою, кількості годин його роботи на день, кількості днів активної роботи за місяць та тарифу на електроенергію в гривнях за кіловат-годину.

На рис. 1.11 зображено результати розрахунку вартості споживання пристрою.

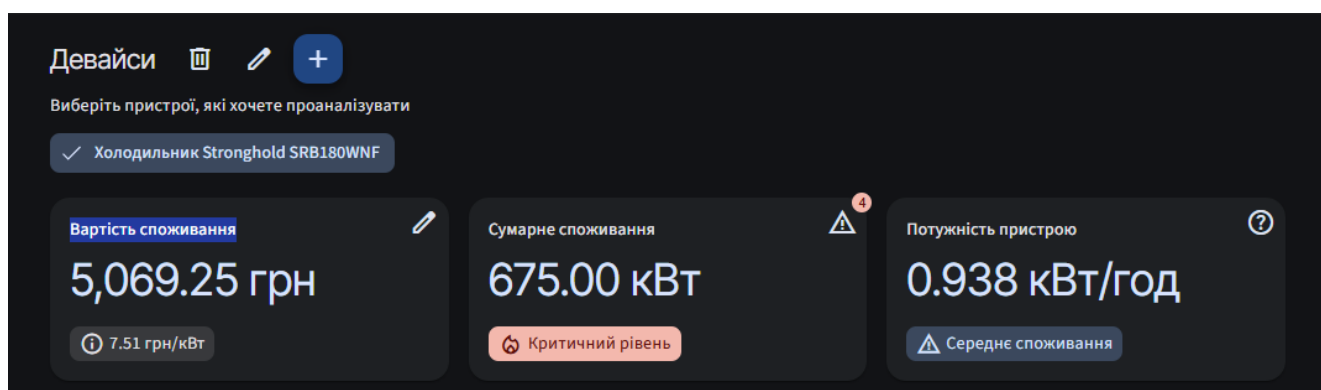


Рисунок 1.11. Результати розрахунку споживання електроенергії

| | | | | | | |
|-----|------|----------|--------|------|-------------------------|------|
| | | | | | РП 08. 01 001. 00 ДП ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 42 |

Для здійснення розрахунку споживання використовуються такі сервіси:

1. DeviceService – відповідає за зберігання пристроїв, обчислення загального споживання енергії пристроєм (calculateTotalWatts) та синхронізацію даних у локальному сховищі браузера.
2. AnalyzeConsumptionService – виконує детальний аналіз параметрів пристрою та генерує попередження у разі виявлення критичних рівнів споживання.
3. TariffService – забезпечує зберігання та використання актуального тарифу на електроенергію, заданого користувачем.

Розрахунок суми витрат базується на формулі:

$$\text{вартість} = \text{спожита_енергія_кВт} * \text{тариф_грн/кВт} \quad (1.1)$$

Де спожита енергія визначається за формулою:

$$\text{споживання (кВт)} = (\text{потужність_у_ватах} * \text{енергоефективність} * \text{години_на_день} * \text{дні_на_місяць}) / 1000 \quad (1.2)$$

При кожному додаванні або оновленні пристрою дані автоматично перераховуються за допомогою методу calculateTotalWatts у DeviceService. Далі ці дані інтегруються з тарифом з TariffService для отримання кінцевої вартості.

Крім базового розрахунку вартості, веб-застосунок проводить автоматичний аналіз споживання через AnalyzeConsumptionService. На основі отриманих даних користувач отримує попередження про надмірне споживання енергії. Наприклад, якщо сумарне споживання пристрою перевищує критичний рівень, на сторінці виводиться відповідний маркер із підказкою.

Система аналізує:

- загальну потужність споживання пристрою;
- кількість годин роботи на день;
- кількість днів роботи на місяць;
- клас енергоефективності пристрою.

Якщо виявлені критичні показники, вони відображаються у вигляді попереджень безпосередньо біля відповідних блоків інформації. Це дозволяє користувачеві миттєво реагувати на аномалії у споживанні та вживати заходів для їх усунення.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------------|------|
| | | | | | РП 08. 01 001. 00 ДП ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 43 |

Користувач не лише отримує підрахунок вартості, але й отримує персоналізовані поради для оптимізації витрат та підвищення енергоефективності.

Однією з основних можливостей веб-застосунку EcoWatt є розрахунок вартості електроспоживання на основі індивідуальних параметрів доданих користувачем пристроїв. Для досягнення цієї цілі застосунок інтегрує кілька сервісів: DeviceService, AnalyzeConsumptionService та TariffService, що працюють у взаємозв'язку.

Розрахунок базується на введених параметрах користувача: потужності пристрою у ватах, годин роботи на день, кількості днів використання на місяць та тарифі на електроенергію.

Усі обчислення виконуються в режимі реального часу, що дозволяє користувачу одразу бачити оновлену інформацію при зміні будь-якого параметра.

Це значно покращує взаємодію з інтерфейсом, оскільки користувач не мусить вручну оновлювати сторінку або підтверджувати зміни – система реагує автоматично. Алгоритм реалізований через спостереження за формами (Reactive Forms), що дозволяє ефективно відслідковувати зміни в даних.

Особливістю реалізації є підтримка різних сценаріїв використання – від розрахунку для одного пристрою до повноцінного моделювання енергоспоживання у квартирі або приватному будинку. Користувач може додавати необмежену кількість пристроїв, комбінувати їх у групи та порівнювати показники між ними. Завдяки цьому веб-застосунок стає універсальним інструментом як для побутового користувача, так і для малого бізнесу, який хоче оптимізувати витрати на електроенергію.

AnalyzeConsumptionService виконує не лише базовий аналіз, а й проводить верифікацію даних з урахуванням класу енергоефективності, який суттєво впливає на підсумкові витрати. Наприклад, пристрій з класом “А+++” при однакових параметрах споживатиме менше енергії, ніж застарілий прилад з класом “С” або “D”. Це дозволяє користувачу зрозуміти, які саме пристрої доцільно замінити, щоб досягти довготривалої економії.

Завдяки інтеграції з TariffService система підтримує кілька типів тарифів –

| | | | | | | |
|-----|------|----------|--------|------|--------------------------------|------|
| | | | | | <i>РП 08. 01 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 44 |

фіксовані, зонні (день/ніч), а також персоналізовані тарифи, які можуть вводитись вручну. Це розширює можливості точного розрахунку вартості навіть для тих користувачів, які користуються спеціальними умовами постачання електроенергії. Використана гнучкість робить застосунок актуальним для різних регіонів і сценаріїв використання.

У підсумку, об'єднання DeviceService, AnalyzeConsumptionService та TariffService дозволяє не просто обчислити витрати на електроенергію, а й сформуванати повноцінну аналітичну картину. Користувач отримує чітке розуміння того, які пристрої споживають найбільше, де виникають перевитрати, і що саме можна покращити. EсоWatt виконує не лише інформаційну, а й освітню функцію, сприяючи формуванню культури енергоефективного мислення.

Алгоритм розрахунку виглядає наступним чином:

1. Отримуються параметри пристрою (watts, energyClass, hours, days).
2. Обчислюється загальне споживання у ватах:

$$totalWatts = watts * energyClass * hours * days \quad (1.3)$$

Ця формула реалізована у методі calculateTotalWatts у класі DeviceService:

```
private calculateTotalWatts(  
  watts: number | null | undefined,  
  energyClass: number | null | undefined,  
  
  hours: number | null | undefined,  
  days: number | null | undefined  
) : number {  
  watts ??= 0;  
  energyClass ??= 0;  
  hours ??= 0;  
  days ??= 0;  
  
  return watts * energyClass * hours * days;  
}
```

Далі отримане значення ват переводиться у кіловат-години шляхом ділення на 1000. Кінцева вартість споживання розраховується множенням спожитої енергії на тариф:

$$вартість_споживання = (totalWatts / 1000) * тариф \quad (1.4)$$

Тариф зберігається окремо в TariffService, що дозволяє легко змінювати тариф без необхідності переписувати дані пристроїв.

```
changeTariff(value: number | null = 0): void {
```

| | | | | | | |
|-----|------|----------|--------|------|--------------------------------|------|
| | | | | | РП 08. 01 001. 00 ДП ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 45 |

```
const validValue = Number(value) ?? 0;
this.persistAndSyncData(validValue);
}
```

Будь-які зміни у тарифі миттєво відображаються у перерахованій вартості споживання для всіх пристроїв.

Крім розрахунку вартості, додаток також проводить аналіз введених даних для виявлення потенційних проблем.

Це реалізовано через сервіс `AnalyzeConsumptionService`, де дані кожного пристрою перевіряються за кількома критеріями:

- надмірна потужність пристрою;
- велика кількість годин роботи на день;
- занадто велика кількість днів використання;
- низький клас енергоефективності.

У разі виявлення відхилень система генерує рекомендації, наприклад:

```
if (totalWatts >= 1000) {
  return 'Переглянути години роботи пристрою, використовувати вночі або в непікові години.';
}
```

або

```
if (hours >= 12) {
  return 'Проаналізувати, чи дійсно необхідно, щоб пристрій працював так довго.';
}
```

Ці повідомлення виводяться користувачеві поряд із відповідними показниками на сторінці перегляду результатів.

Після обчислення даних результати відображаються у вигляді інформаційних карток на сторінці. Кожна картка містить:

- загальну вартість споживання (грн);
- сумарне споживання енергії (кВт);
- середню потужність пристрою (кВт/год);
- індикатори потенційних проблем (наприклад, "Критичний рівень").

Користувач може візуально оцінити ефективність використання енергії та, за необхідності, оптимізувати налаштування своїх пристроїв.

Після завершення обчислень дані виводяться у зручному форматі – за допомогою інформаційних карток, які дозволяють швидко оцінити результати.

Многорівневий формат подачі інформації є наочним і інтуїтивно зрозумілим, що

| | | | | | | |
|-----|------|----------|--------|------|--------------------------------|------|
| | | | | | <i>РП 08. 01 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 46 |

особливо важливо для пересічного користувача без технічної підготовки. Кожна картка оформлена відповідно до єдиного стилю інтерфейсу й містить ключові показники, що відображають стан енергоспоживання конкретного пристрою або групи пристроїв.

Одним із головних показників є загальна вартість електроенергії, спожитої за заданий період. Це дозволяє користувачу миттєво зрозуміти, який із пристроїв спричиняє найбільші витрати. Сумарне споживання енергії, виражене у кіловат-годинах, надає змогу оцінити рівень навантаження на енергосистему. Середня потужність роботи пристрою допомагає побачити, наскільки енергоефективним є конкретний прилад протягом доби.

Окремо варто звернути увагу на індикатори проблем або ризиків, наприклад, позначки “Критичний рівень” або “Високе споживання”, вони дуже важливі при побудові алгоритму відповіді до клієнта.

Вони автоматично з’являються у випадках, коли певні параметри перевищують безпечні або рекомендовані межі. Окремий довод дозволяє не лише аналізувати, а й своєчасно реагувати на можливі проблеми – наприклад, надмірне навантаження на мережу або неекономне використання енергії.

Завдяки візуальному поданню результатів користувач отримує змогу оцінити ефективність використання електроенергії без необхідності розбиратися у складних таблицях або технічних графіках. У разі виявлення надмірного споживання, він може оперативної змінити параметри пристроїв або скорегувати режим їхньої роботи. Система не лише інформує, а й активно допомагає користувачеві приймати обґрунтовані рішення щодо економії енергії.

1.3.5 Будування графіків та діаграм

Одним із важливих функціональних елементів веб-застосунку EcoWatt є можливість візуалізації даних про енергоспоживання у вигляді графіків. Це дозволяє користувачам наочно оцінювати зміни у використанні електроенергії та виявляти аномалії чи потенційні проблеми.

Для реалізації побудови графіків було використано бібліотеку ApexCharts, інтегровану через компонент ng-apexcharts, що забезпечує гнучкі налаштування

| | | | | | | |
|-----|------|----------|--------|------|--------------------------------|------|
| | | | | | РП 08. 01 001. 00 ДП ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 47 |

графічних елементів та високу продуктивність при відображенні великої кількості даних.

Візуалізація даних у вигляді графіків відіграє ключову роль у зручному сприйнятті інформації. Користувачі отримують можливість спостерігати динаміку змін енергоспоживання за день, тиждень або місяць, що допомагає краще розуміти власні звички та ефективність використання пристроїв. Графіки дозволяють не лише бачити сухі цифри, а й інтерпретувати їх у контексті часу, що значно спрощує аналіз.

Інтеграція ApexCharts через компонент ng-apexcharts забезпечила гнучкість у реалізації різних типів графіків – лінійних, стовпчикових, комбінованих тощо. Це дозволяє обрати найбільш відповідний варіант для подання конкретних показників, таких як щоденне навантаження, пікові години споживання або середньодобова динаміка. Крім того, бібліотека підтримує інтерактивність: при наведенні на елементи графіка користувач бачить точні значення, що додає зручності при аналізі.

Окрему увагу приділено адаптивності графіків. Вони коректно відображаються як на великих екранах, так і на мобільних пристроях, змінюючи масштаб і форматування для збереження читабельності. Це дозволяє користувачам переглядати статистику в будь-який зручний момент, незалежно від типу пристрою. Інтерфейс реагує на зміну розмірів екрану без втрати функціональності або візуальної якості.

Також було реалізовано можливість фільтрації даних – користувач може обрати конкретний період часу або окремі пристрій для детального аналізу. Це значно спрощує процес порівняння споживання між різними приладами або часовими відрізками. Наприклад, можна швидко побачити, як змінилося навантаження після оновлення техніки або введення нових налаштувань енергозбереження.

Загалом, функція графічного відображення даних не лише покращує користувацький досвід, а й виконує важливу аналітичну роль. Як зазначено на офіційному порталі YASNO, «аналіз щоденного споживання дозволяє зрозуміти,

| | | | | | | |
|-----|------|----------|--------|------|--------------------------------|------|
| | | | | | <i>РП 08. 01 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 48 |

коли та скільки електроенергії ви використовуєте, і які звички варто змінити для зменшення рахунків» [8].

Крім основних графіків, у застосунку передбачено можливість відображення порівняльної статистики – наприклад, співставлення споживання кількох пристроїв або аналіз змін до й після впровадження енергозберігаючих заходів. Такі графіки дозволяють не лише фіксувати факт перевитрати, а й виявляти конкретні причини зростання витрат. Це особливо корисно для тих користувачів, які прагнуть системно підходити до енергоменеджменту у власному домогосподарстві.

Також реалізована функція збереження та експорту графіків у форматах PNG або PDF, що дає змогу зберігати звіти для подальшого аналізу або обміну з іншими користувачами, наприклад, членами сім'ї чи технічними спеціалістами. Це робить застосунок не лише зручним інструментом для моніторингу, а й потужним засобом для візуального представлення інформації у зовнішніх звітах чи презентаціях.

Можливість експорту графіків значно розширює сферу застосування EcoWatt, оскільки користувачі можуть документувати динаміку споживання електроенергії за певний період і надавати її, наприклад, енергетичним аудиторам або комунальним службам. Експортовані графіки мають чітке оформлення, з підписами осей, легендами та позначками ключових показників, що забезпечує легке сприйняття інформації навіть поза межами інтерфейсу веб-застосунку. Це відповідає сучасним рекомендаціям, які формулюються на рівні регіональних операторів, зокрема ДТЕК Одеса, де зазначено: «визначення енергоефективних рішень починається з точної фіксації та оцінки споживання» [9].

Відповідальним за візуалізацію графіків є компонент EnergyChartComponent, який розроблений як standalone-компонент з високим рівнем оптимізації за рахунок використання ChangeDetectionStrategy.OnPush.

Основні елементи логіки побудови графіка:

1. Прийом даних пристроїв через вхідний параметр selectedDevices:

```
selectedDevices = input.required<IDeviceModel[]>();
```

2. Побудова масиву даних для кожного пристрою за 30 днів за допомогою computed:

```
readonly chartSeries = computed(() => {
```

| | | | | | | |
|-----|------|----------|--------|------|--------------------------------|------|
| | | | | | <i>РП 08. 01 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 49 |

```

const devices = this.selectedDevices();
return devices.map((device) => {
  const data = Array.from({ length: 30 }, (_, i) => device.energyData[i] || 0);
  return {
    name: device.name,
    data
  };
});

```

Для кожного пристрою генерується окрема серія даних, що відображається окремою лінією на графіку.

На рис. 1.12 наведено приклад побудованого графіка споживання електроенергії за 30 днів, тут користувач може бачити піки які потребує прилад, та коли була самий вагоме споживання електроприладу.

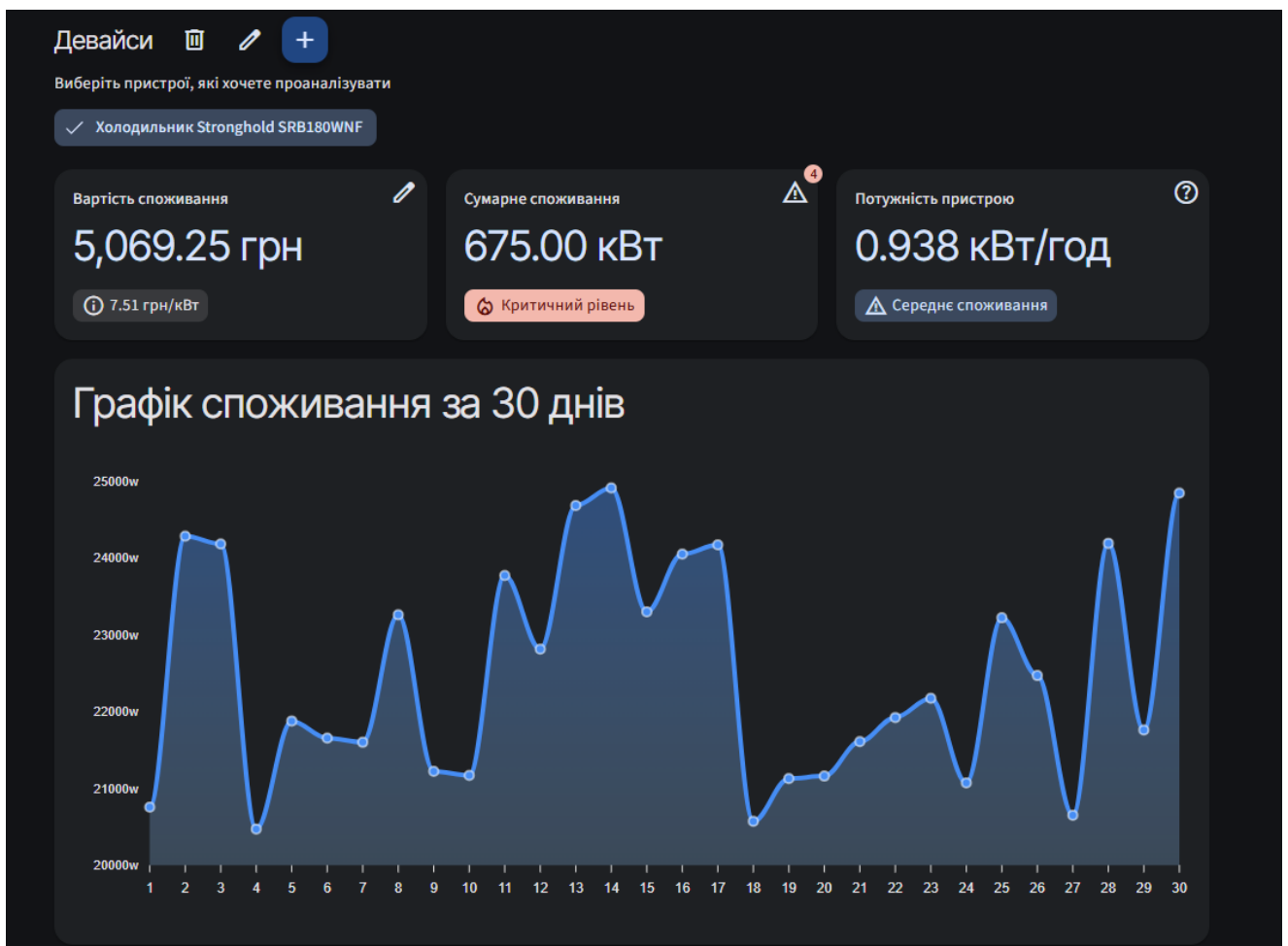


Рисунок 1.12. Графік споживання електроенергії за 30 днів

Налаштування зовнішнього вигляду графіка здійснюється через об'єкт `chartAppearanceOptions`, який містить параметри типу графіка, оформлення осей, підказок, маркерів і сітки:

```
readonly chartAppearanceOptions: IApexAppearanceChartOptions = {
```

```

chart: { type: 'area', height: 400, ... },
xaxis: { categories: Array.from({ length: 30 }, (_, i) => `${i + 1}`), ... },
yaxis: { labels: { formatter: (value: number) => value.toFixed(0) + 'w' }, ... },

stroke: { curve: 'smooth' },
tooltip: { enabled: true, shared: false, intersect: true, ... },
...
};

```

Особливу увагу приділено адаптивності відображення підписів на осі X: залежно від розміру вікна браузера підписи змінюються для зручності читання.

1.3.6 Генерація даних для графіка

Реальні дані споживання генеруються у сервісі ChartDataGeneratorService під час додавання пристрою. Завдяки цьому навіть у тестовому режимі користувач отримує правдоподібний розподіл енергоспоживання на кожен день місяця.

Метод генерації даних виглядає так:

```

generateRandomData(total?: number, length = 0): number[] {
  if (!total) return [];
  const randomNumbers = [];
  let sum = 0;

  for (let i = 0; i < length; i++) {
    const randomValue = this.generateRandomFactor();
    randomNumbers.push(randomValue);
    sum += randomValue;
  }
  const scale = total / sum;
  for (let i = 0; i < length; i++) {
    randomNumbers[i] = Math.round(randomNumbers[i] * scale);
  }
  return randomNumbers;
}

```

Цей підхід забезпечує рівномірний розподіл сумарного споживання між днями з невеликою варіативністю, що робить графік більш реалістичним.

Графіки в EcoWatt розроблено з урахуванням інтерактивності та зручності для користувача. При наведенні курсора на будь-яку точку графіка відображається підказка з детальною інформацією про рівень споживання в обраний день. Він допомагає користувачеві швидко отримати точні значення без потреби аналізувати шкалу вручну. Крім того, при завантаженні графіка застосовується анімація появи (fadeInAnimation), яка створює плавний та естетичний ефект візуалізації даних.

Особливу увагу приділено адаптивності графіків. Завдяки гнучким налаштуванням осей, графік коректно відображається як на великих екранах, так і на мобільних пристроях.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------------|------|
| | | | | | <i>РП 08. 01 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 51 |

У випадках, коли дані ще не завантажені або користувач не додав жодного пристрою, система автоматично виводить placeholder-компонент. Даний функціонал допомагає уникнути порожнього простору на сторінці та повідомляє користувача про необхідність виконання певної дії, та підвищує інформативність інтерфейсу та створює відчуття завершеності навіть у ситуаціях, коли графіки показуються користувачеві. Також користувач може побачити усі проблеми пов'язані з електроприладом.

На рис. 1.13 показано приклад підказки при наведенні на графік.

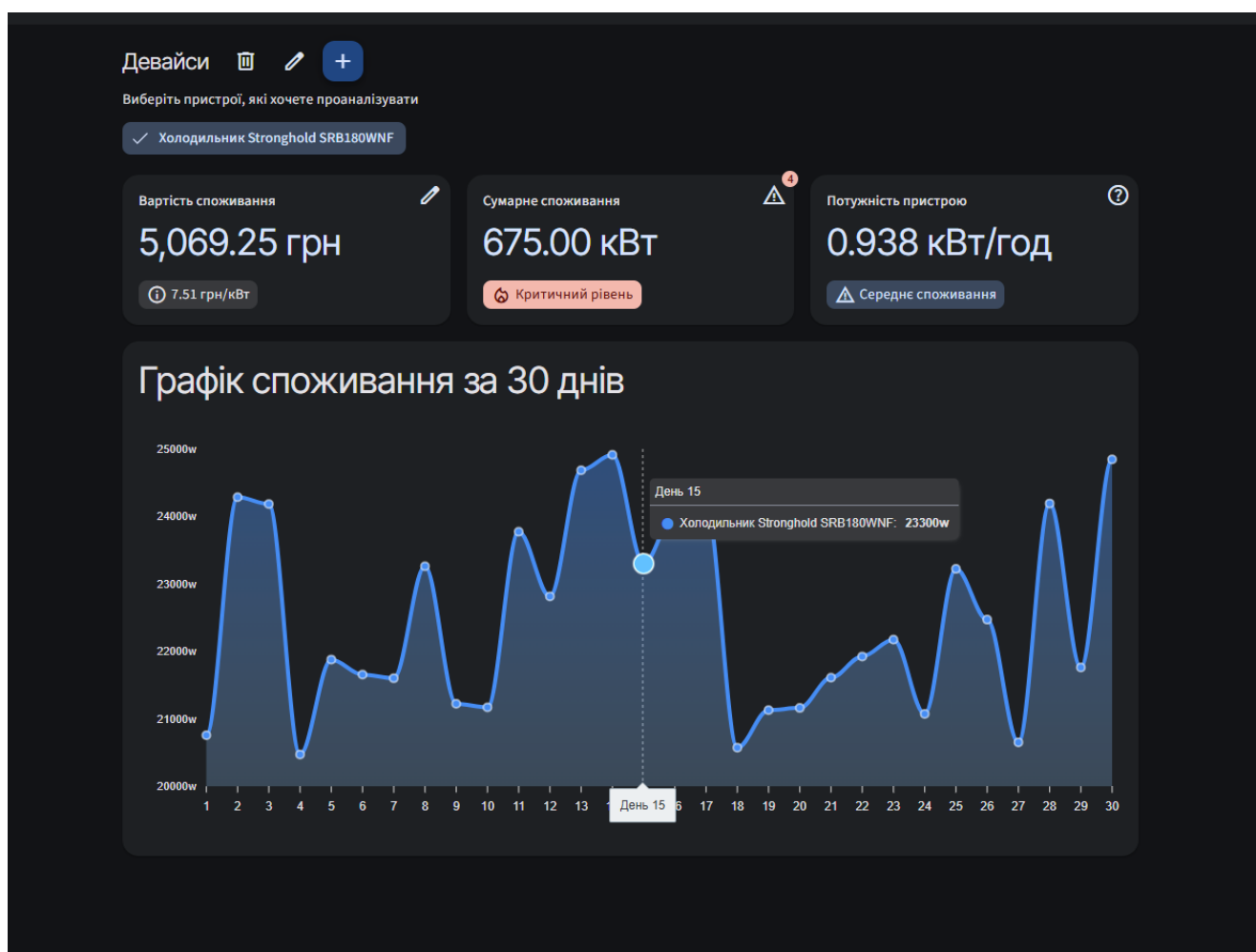


Рисунок 1.13. Підказка на графіку при наведенні на дані

1.3.7 Структура HTML-шаблону компонента

HTML-розмітка компонента `EnergyChartComponent` реалізована з використанням `@defer` та `@if`.

Це дозволяє ефективно керувати станами завантаження:

```
<div class="card">
  <div class="display-small">Графік споживання за 30 днів</div>
```

| | | | | | | |
|-----|------|----------|--------|------|--------------------------------|------|
| | | | | | <i>РП 08. 01 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 52 |

```

@defer (when selectedDevices().length) {

  @if (selectedDevices().length) {
    <apx-chart [series]="chartSeries()" ...></apx-chart>
  } @else {
<app-chart-placeholder />
  }
} @placeholder (minimum 300ms) {

  <app-chart-placeholder />
} @loading (minimum 300ms) {

  <div class="loading">
    <mat-spinner />
  </div>
}
</div>

```

Користувач завжди бачить відповідний стан інтерфейсу: графік, плейсхолдер або індикатор завантаження.

1.3.8 Функціонал допоміжних підказок (хелперів)

Для покращення користувацького досвіду та підвищення рівня зрозумілості інтерфейсу у веб-застосунку EcoWatt реалізовано систему допоміжних підказок (хелперів). Вони забезпечують оперативне надання пояснень щодо різних аспектів споживання електроенергії та особливостей роботи застосунку без необхідності переходити на окремі сторінки або читати довгі інструкції.

Підказки реалізовано у вигляді інтерактивних іконок з символом «?», які розташовані поруч із ключовими елементами інтерфейсу. При наведенні курсора на іконку користувач бачить коротке пояснення через matTooltip, а в окремих випадках може відкрити розширене діалогове вікно з детальною інформацією. Додатково він додає контекст саме в той момент, коли це потрібно, не відволікаючи користувача від основного процесу.

Особливо корисними хелпери є у розділах додавання пристрою або перегляду результатів, де можуть виникати питання щодо розрахунків, енергоефективності або значення окремих показників. Крім того, система підказок підтримує динамічний зміст повідомлення змінюються залежно від ситуації або типу пристрою. Він додає ефект «розумного інтерфейсу», який активно допомагає користувачу та сприяє зменшенню кількості помилок або непорозумінь під час роботи із застосунком. Спадкування від BasePageDirective стандартизує поведінку

| | | | | | | |
|-----|------|----------|--------|------|--------------------------------|------|
| | | | | | <i>РП 08. 01 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 53 |

сторінок, спрощуючи підтримку проекту в довгостроковій перспективі, та може використовуватись у додатковому середовищі.

Для створення підказок використано можливості Angular у поєднанні з Material компонентом і кастомним контентом, який в свою чергу зав'язаний на внутрішніх компонентах середовища фреймворку, та використовує пов'язані бібліотеки.

На рис. 1.14 наведено приклад підказки на картці потужності пристрою.

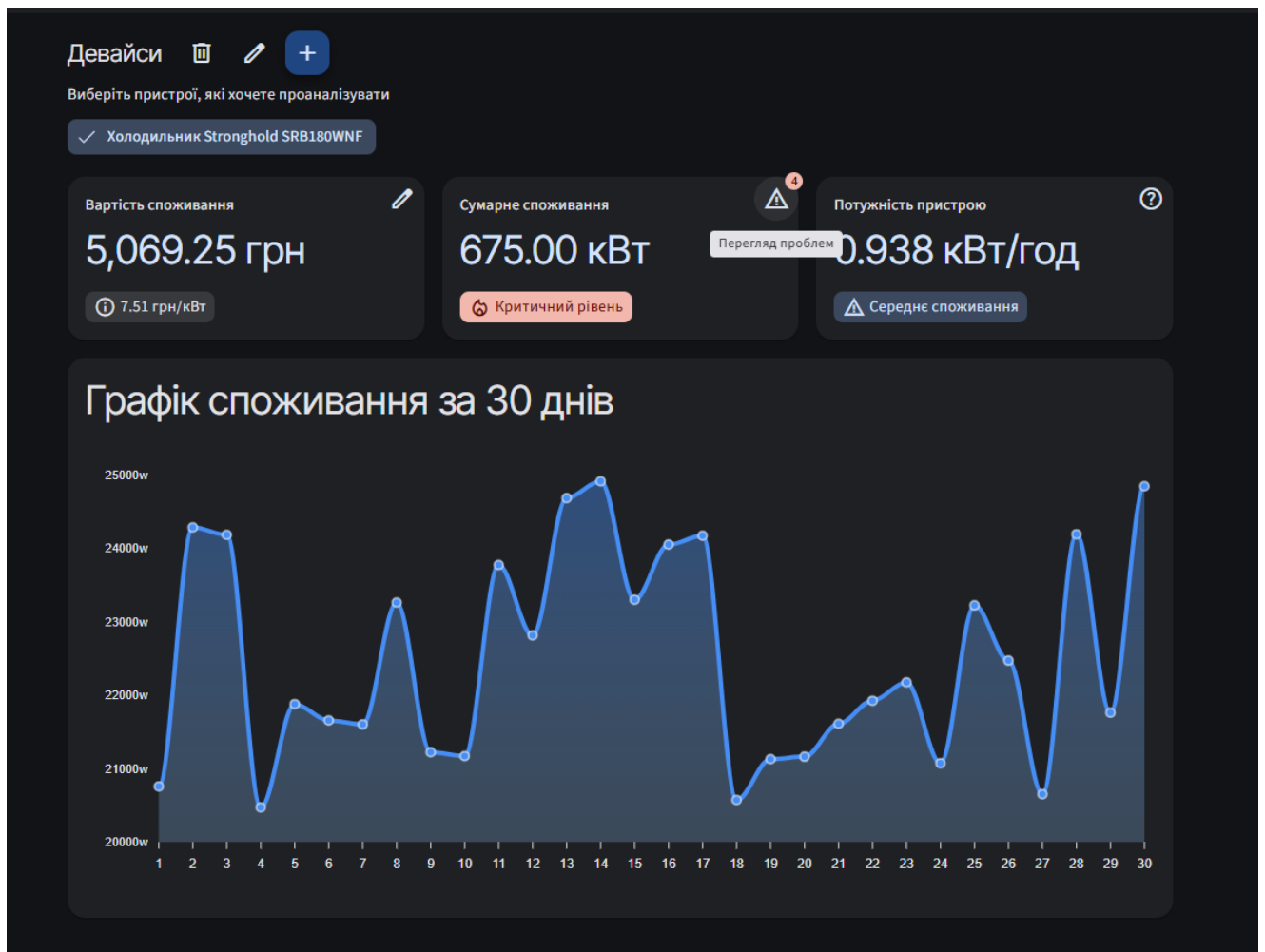


Рисунок 1.14. Пояснення поняття потужності пристрою

Основні елементи реалізації:

Іконка допомоги (символ ?) з matTooltip, яка при наведенні миші відображає пояснення. Додатково реалізовано можливість розширеного діалогу для складніших пояснень (наприклад, для перегляду повного списку проблем енергоспоживання).

Приклад коду реалізації простого хелпера:

| | | | | | | |
|-----|------|----------|--------|------|-------------------------|------|
| | | | | | РП 08. 01 001. 00 ДП ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 54 |

```
<button mat-icon-button [matTooltip]="tooltipText">
  <mat-icon>help_outline</mat-icon>
</button>
```

Де tooltipText може бути передано з компонента:

```
tooltipText = 'Що таке потужність пристрою?';
```

Для складніших випадків використовується відкриття діалогових вікон через MatDialog, наприклад для виведення списку знайдених проблем у споживанні енергії та повідомляє користувача про необхідність виконання установ, подивимось на рисунок (рис. 1.15).

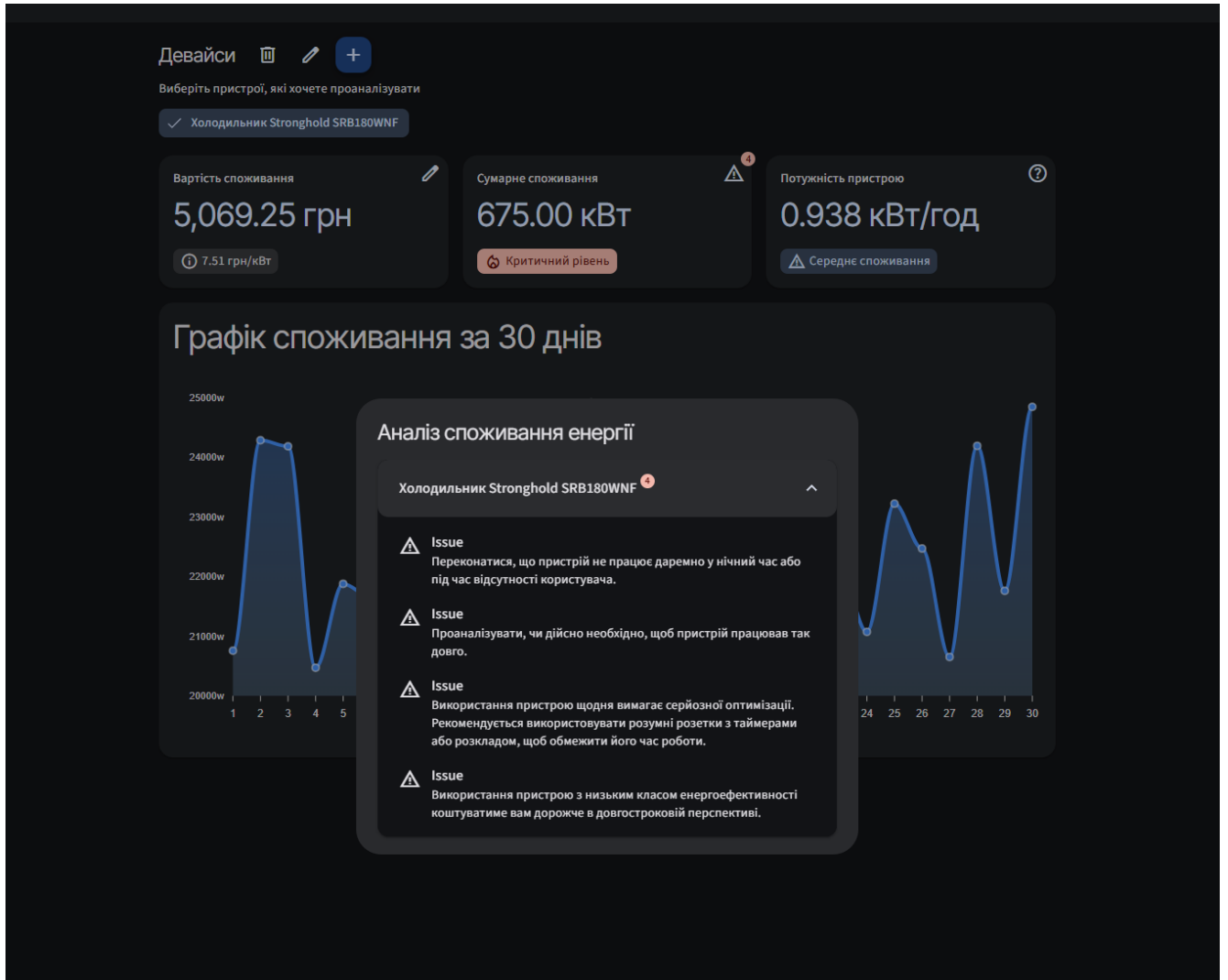


Рисунок 1.15. Відкриття списку попереджень про надмірне споживання

Функціональні можливості хелперів:

1. Надання короткої довідкової інформації про елементи інтерфейсу.
2. Пояснення формул або показників без перенавантаження сторінки текстом.
3. Попередження користувачів про потенційні проблеми або необхідність

оптимізації.

4. Підвищення рівня довіри та залученості користувача завдяки наданню додаткового контексту.

Нижче наведено код типового хелперу:

```
<button mat-icon-button [matTooltip]="'Це вартість електроенергії, що споживається вашим пристроєм'">
```

```
  <mat-icon>info</mat-icon>
</button>
```

або для перегляду критичних проблем:

```
<button mat-icon-button [matTooltip]="'Перегляд проблем енергоспоживання'">
  <mat-icon>warning</mat-icon>
</button>
```

У разі наявності проблем відображається бейдж з кількістю попереджень:

```
<mat-badge [content]="issuesCount" matBadgeOverlap="false">
  <mat-icon>warning</mat-icon>
</mat-badge>
```

При натисканні на хелпер або значок попередження користувач отримує розгорнуту інформацію про проблеми споживання (наприклад: надмірна кількість годин роботи пристрою, низький клас енергоефективності тощо).

1.3.9 Допоміжний функціонал для користувача

Для покращення взаємодії з веб-застосунком EcoWatt було реалізовано низку допоміжних функцій, які значно підвищують зручність, зрозумілість і практичну користь сервісу. Однією з таких функцій є інтерактивний тур по інтерфейсу, який запускається автоматично при першому вході користувача. Підказки з'являються поряд з ключовими елементами сторінки (наприклад, кнопками редагування, додавання пристроїв, перегляду результатів) і пояснюють їх призначення. Даний функціонал дозволяє навіть новачкам швидко зорієнтуватися в інтерфейсі без потреби читати документацію або звертатися по допомогу. Масштаб і кількість підписів автоматично підлаштовуються під розмір вікна, що забезпечує зручність перегляду на будь-якому екрані. Слід наголосити, що візуальне оформлення графіка доповнюється градієнтним заповненням області під лінією, що підсилює сприйняття загальної динаміки. Крім того, тур підтримує локалізацію, тож підказки автоматично відображаються мовою, яку обрав користувач. Це створює більш інклюзивний досвід та сприяє ширшому залученню аудиторії.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------------|------|
| | | | | | <i>РП 08. 01 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 56 |

Такий підхід до дизайну інтерфейсу сприяє інтуїтивному освоєнню функціоналу застосунку, що зменшує бар'єри для нових користувачів. Локалізовані підказки та адаптивність інтерфейсу забезпечують комфортне використання EcoWatt для людей з різними мовними та технічними потребами. Усе це разом підвищує загальну задоволеність користувачів і сприяє їхній активній взаємодії з платформою.

Приклад туру показан на зображенні 1.16.

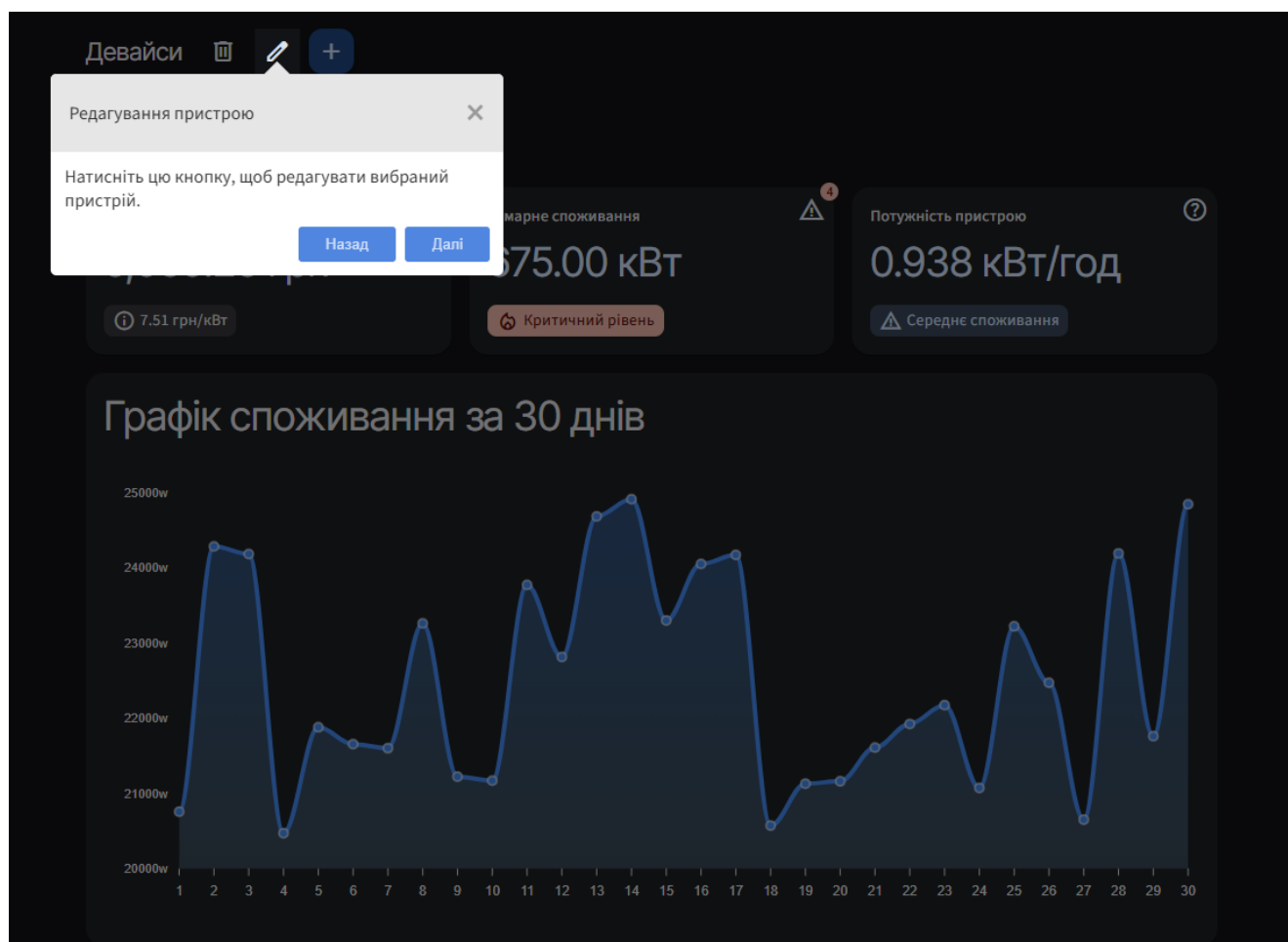


Рисунок 1.16. Приклад активного туру з підказкою «Редагування пристрою»

Кожен крок інтерактивного туру супроводжується кнопками "Назад" та "Далі", що дозволяє користувачу самостійно контролювати швидкість ознайомлення з функціоналом. Всі підказки реалізовані за допомогою кастомізованих компонентів Angular, які забезпечують плавну анімацію, адаптивність до різних екранів і підтримку кількох мов. У майбутньому функціонал може бути доповнений можливістю пропустити окремі блоки або повернутись до туру в будь-який момент із налаштувань профілю.

| | | | | | | |
|-----|------|----------|--------|------|-------------------------|------|
| | | | | | РП 08. 01 001. 00 ДП ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 57 |

Ще одним важливим елементом є сторінка «Ефективне споживання електроенергії», яка виконує роль інтерактивного довідника та мотиваційного блоку. На ній подано поради щодо зменшення споживання, пояснено важливість енергоощадності як для окремого домогосподарства, так і для енергетичної стабільності країни в цілому. Сторінка містить динамічні блоки з візуалізацією потенційної економії в разі впровадження конкретних порад, що сприяє кращому розумінню користувачами результатів своєї поведінки. Вміст адаптується залежно від даних споживання користувача, що дозволяє надавати персоналізовані рекомендації. Такий підхід стимулює зміну звичок і підвищує загальний рівень енергетичної свідомості.

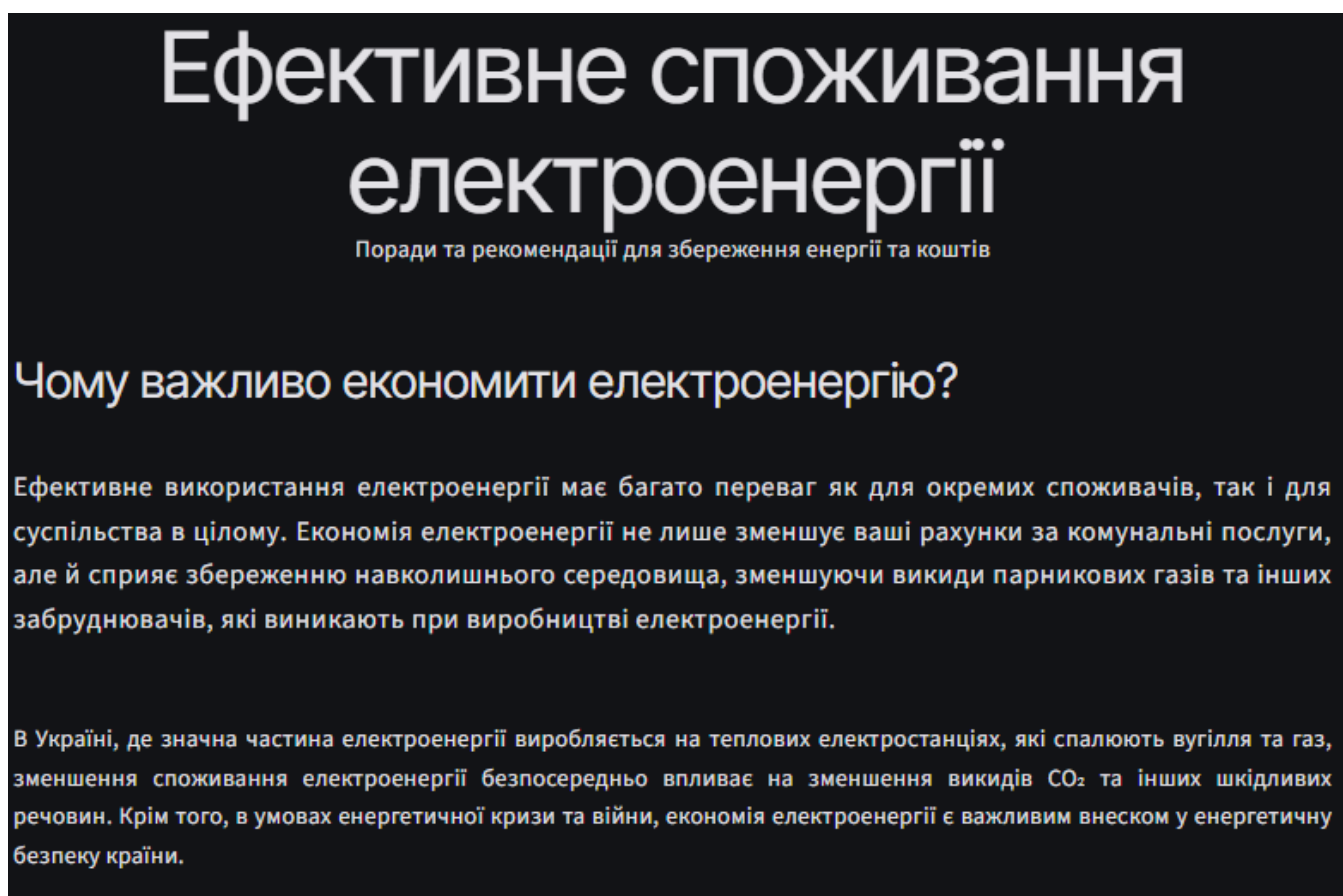


Рисунок 1.17. Приклад сторінки "Ефективне споживання електроенергії"

Контент цієї сторінки підготовлений у формі запитань і відповідей, що наближує його до формату класичного FAQ. Текст доповнено акцентами та заголовками, які допомагають швидко зорієнтуватися у змісті. Особливу увагу приділено візуальній простоті, читабельності та стислості подання, щоб зробити матеріали максимально доступними навіть для користувачів без технічної

| | | | | | | |
|-----|------|----------|--------|------|--------------------------------|------|
| | | | | | <i>РП 08. 01 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 58 |

підготовки. Планується подальше розширення блоку інтерактивними інструментами, наприклад, порівнянням двох пристроїв за ефективністю або індивідуальними порадами. Це дозволить ще більше підвищити зручність використання сторінки та залучити ширшу аудиторію.

Окремо варто зупинитися на інноваційній функції автоматичного підбору енергоефективного обладнання, яка реалізована через інтелектуальний алгоритм EcoWatt. При кожному оновленні або додаванні пристрою система аналізує його основні параметри: потужність, тип, тривалість роботи, клас енергоефективності, частоту використання та співвідносить ці дані з внутрішньою базою типових профілів споживання. Якщо виявляється, що пристрій споживає аномально багато електроенергії або працює із значними перевищеннями, активується логіка генерації рекомендації.

Унікальність алгоритму полягає в тому, що він не просто порівнює споживання з умовними нормами, а враховує контекст: чи типове це навантаження для зазначеного класу пристрою, як воно співвідноситься з іншими пристроями цього користувача, чи була динаміка зростання споживання, чи виявлені "пікові" години роботи. На основі цього формуються висновки, які можуть вказувати на зношеність пристрою, неефективну модель або неправильне налаштування. У таких випадках система автоматично переходить до етапу пошуку альтернативи.

Для підбору товару EcoWatt інтегрується з API популярного онлайн-магазину (наприклад, Rozetka). Алгоритм формує запит із зазначенням типу пристрою, рекомендованої потужності, класу енергоефективності, цінового діапазону та популярності моделей. У відповідь система отримує посилання на конкретну модель, яка найкраще відповідає цим критеріям, і виводить його у вигляді іконки з кошиком поруч із пристроєм у списку.

Даний підхід демонструє новий рівень взаємодії від моніторингу до практичних рекомендацій, що базуються на персоналізованому аналізі. EcoWatt фактично виконує роль асистента з енергоефективності, який не лише показує цифри, а й пропонує шляхи вирішення проблеми. У перспективі функціонал може бути доповнений рейтингами товарів, фільтрами за брендом або наявністю, а

| | | | | | | |
|-----|------|----------|--------|------|--------------------------------|------|
| | | | | | <i>РП 08. 01 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 59 |

також функцією порівняння моделей прямо в інтерфейсі EcoWatt. Багаторівнева інтеграція вигідна як користувачеві, так і розробникам системи, адже поєднує соціальну користь, комерційну доцільність і технологічну новизну. Такий підхід до дизайну інтерфейсу сприяє інтуїтивному освоєнню функціоналу застосунку, що зменшує бар'єри для нових користувачів. Локалізовані підказки та адаптивність інтерфейсу забезпечують комфортне використання EcoWatt для людей з різними мовними та технічними потребами. Усе це разом підвищує загальну задоволеність користувачів і сприяє їхній активній взаємодії з платформою.

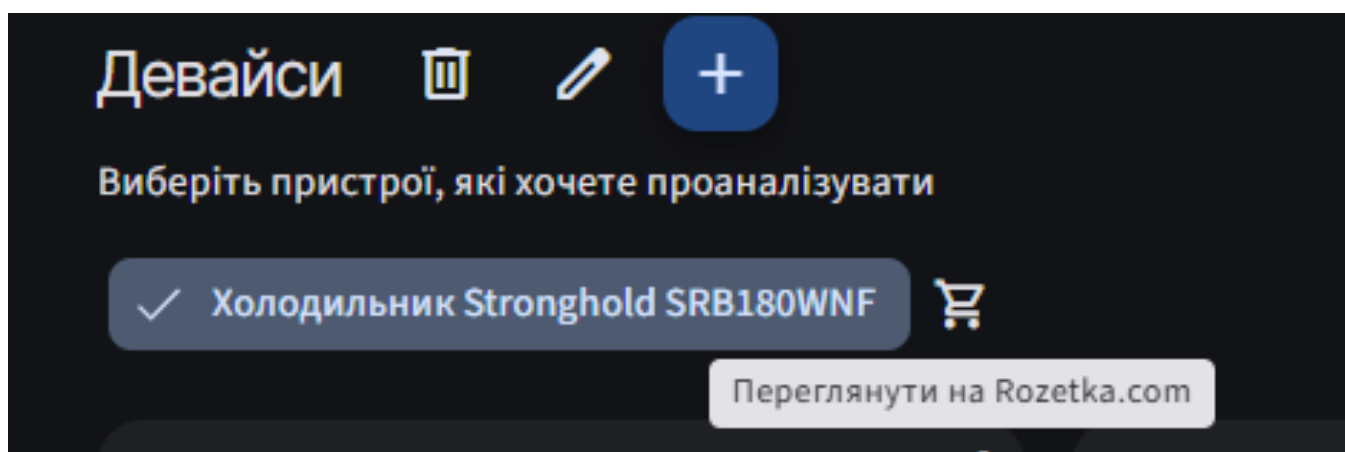


Рисунок 1.18. Алгоритм посилення користувача на сайт поставника

Інтеграція з платформами електронної комерції є прикладом зручного переходу від аналітики до дій. Користувач може одразу побачити причину надмірного споживання та отримати інструмент для її усунення. Надалі можлива реалізація фільтрів, які враховуватимуть бюджет, клас енергоефективності, бренди та інші параметри при виборі рекомендацій для заміни.

Допоміжний функціонал EcoWatt не лише полегшує користування застосунком, а й виконує освітню, рекомендаційну та мотиваційну функцію. Він допомагає зробити енергоспоживання не лише контрольованим, а й свідомим, що відповідає загальній меті системи, це сприяти економії, обізнаності та сталому розвитку.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------------|------|
| | | | | | <i>РП 08. 01 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 60 |

2 ЕКОНОМІЧНИЙ РОЗДІЛ

2.1 Резюме

У дипломному проєкті розроблено веб-застосунок для моніторингу енергоспоживання в побуті, який дає змогу відстежувати показники, створювати графіки та оцінювати витрати. Такий функціонал сприяє підвищенню енергоефективності та економії ресурсів. Якість системи оцінюється за критеріями зручності використання, продуктивності та практичної цінності.

2.2 Визначення трудомісткості розробки ПЗ

Тривалість розробки програмного продукту залежить від його обсягу, складності, кваліфікації розробників і встановлених ринком термінів. Метод структурної аналогії дозволяє оцінити обсяг у тисячах умовних машинних команд на основі подібного програмного забезпечення.

Табл. 2.1 містить аналоги ПЗ з подібними функціями; обраний варіант виділено сірим.

Таблиця 2.1. Каталог аналогів

| Найменування ПП | Обсяг функції ПП – V_0 , умовних. машинних командах |
|---|--|
| 1. ПП автоматизації засобів по каталогу | 680 – 7000 |
| 2. ПП автоматизованих розрахунків | 1300 – 8600 |
| 3. ПП введення інформації | 1060 – 5750 |

Після вибору аналога з обсягом V_0 (умовні машинні команди), трудомісткість визначається за табл. 2.2.

Таблиця 2.2. Обсяг ПП

| Обсяг ПП, тис.умов.машинних команд | Норма часу, люд/год |
|------------------------------------|---------------------|
| 1.00 | 229 |
| 2.00 | 244 |
| 3.00 | 262 |
| 4.00 | 283 |
| 5.00 | 306 |
| 6.00 | 330 |
| 7.00 | 357 |

| Обсяг ПП, тис.умов.машинних команд | Норма часу, люд/год |
|------------------------------------|---------------------|
| 8.00 | 385 |
| 9.00 | 414 |
| 10.00 | 445 |

На основі отриманого значення за довідником визначають укрупнену норму часу, скориговану коефіцієнтом $K_k = 0,7-0,8$ для умов розробки на комп'ютері:

$$T_{ap} = 306 \times 0,8 = 244,8 \text{ (люд/годин)} \quad (2.1)$$

Трудомісткість визначається для кожного етапу окремо, з урахуванням складності, новизни та використання стандартних модулів, за відповідними формулами:

$$T_{T3} = T^a p \times L_1 \times K_H \quad (2.2)$$

$$T_{TP} = T^a p \times L_2 \times K_H \quad (2.3)$$

$$T_{PP} = T^a p \times L_3 \times K_H \times K_T \quad (2.4)$$

Для розрахунку використовуються такі коефіцієнти:

- L_i – частка i -го етапу (табл. 2.3);
- K_H – коефіцієнт новизни (табл. 2.4);
- K_T – коефіцієнт використання типових програм (табл. 2.5).

Наш варіант виділено сірим.

Таблиця 2.3. Питомі коефіцієнти трудомісткості стадії у загальній трудомісткості розробки ПП

| Код стадії | Ступінь новизни | | |
|--------------|-----------------|------|------|
| | А | Б | В |
| ТЗ (L_1) | 0,15 | 0,12 | 0,12 |
| ТП (L_2) | 0,16 | 0,15 | 0,11 |
| РП (L_3) | 0,55 | 0,58 | 0,61 |

Таблиця 2.4. Значення коефіцієнта новизни

| Код ступеня новизни | Ступінь новизни | Значення K_H |
|---------------------|---|----------------|
| А | Принципово новий ПП | 1,75 – 1,2 |
| Б | ПП – розвиток визначеного параметричного ряду | 1,0 – 0,8 |

| Код ступеня новизни | Ступінь новизни | Значення K_n |
|---------------------|-------------------|----------------|
| В | ПП, що має аналог | 0,7 |

Таблиця 2.5. Значення коефіцієнта використання типових програм

| Ступінь охоплення реалізованих функцій розробленого ПП типовими програмами, % | Значення K_T |
|---|----------------|
| 60 і вище | 0,6 |
| 40-60 | 0,7 |
| 20-40 | 0,8 |
| До 20 | 0,9 |

Тепер розраховуємо трудомісткість для всіх етапів і зводимо у табл. 2.6:

Трудомісткість технічного завдання:

$$T_{tz} = Ta * L_1 * K_n = 244,8 * 0,12 * 0,8 = 23,50 \text{ (люд/годин)} \quad (2.2)$$

Трудомісткість розробки технічного проєкту:

$$T_{tp} = Ta * L_2 * K_n = 244,8 * 0,15 * 0,8 = 29,37 \text{ (люд/годин)} \quad (2.3)$$

Трудомісткість розробки робочого проєкту:

$$T_{rp} = Ta * L_3 * K_n * K_m = 244,8 * 0,58 * 0,8 * 0,7 = 79,51 \text{ (люд/годин)} \quad (2.4)$$

Для розрахунків визначили обсяг документації по етапах:

- технічне завдання $N_{tz}=2$ (стор);
- розробка ТП $N_{tp}=53$ (стор);
- розробка робочого проєкту $N_{rp}=9$ (стор);
- пояснювальна записка відповідно $N_{pz}=15$ (стор).

Таблиця 2.6. Розрахунок трудомісткості ПП

| Найменування етапів | Розрахунок, годин | | |
|---------------------|---------------------------------|---------------------------------|-----------------------------------|
| | 1.ТЗ | $T_{РТЗ}=23,50$ | $T_{КК}=0,7*N_{ТЗ}= 0,7*2=1,4$ |
| 2.Розробка ТП | $T_{РТП}=29,37$ | $T_{КК}=0,7*N_{ТП}=0,7*53=37,1$ | $T_{НК}=0,15*N_{ТП}=0,15*53=7,95$ |
| 3.Розробка РП | $T_{РРП}= 79,51$ | $T_{КК}=0,7*N_{РП}=0,7*9=6,3$ | $T_{НК}=0,15*N_{РП}=0,15*9=1,35$ |
| 4.Розробка ПЗ | $T_{ПЗ}=1,5*N_{ПЗ}=1,5*15=22,5$ | $T_{КК}=0,7*N_{ТЗ}=0,7*15=10,5$ | $T_{НК}=0,15*N_{ПЗ}=0,15*15=2,25$ |

| | | | |
|----------------------|-----------|----------|-----------|
| Усього, в т.ч.: | 213,53 | | |
| - на розробку | Тр=154,88 | | |
| - контроль керівника | | Ткк=48,3 | |
| - нормоконтроль | | | Тнк=10,35 |

2.3 Розрахунок ціни програмного продукту

Розраховуємо основну зарплату виконавців, матеріальні та загальні витрати на розробку ПП. Зарплата наведена в табл. 2.7. З 1 квітня 2024 мінімальна місячна зарплата – 8000 грн, погодинна ставка – 46 грн (згідно зі ст. 8 Закону про Держбюджет України).

Таблиця 2.7. Розрахунок основної заробітної плати виконавців

| Найменування робіт | Трудовіткість робіт, години | Погодинна тарифна ставка, грн. | Розрахунок, грн. |
|----------------------|-----------------------------|--------------------------------|------------------|
| 1.Розробка ПП | 154,88 | 55,35 | 8573,48 |
| 2.Контроль керівника | 48,3 | 120,28 | 5810,00 |
| 3.Нормоконтроль | 10,35 | 120,28 | 1245,00 |
| Усього | - | - | Зо= 15628,48 |

Розраховуємо матеріальні витрати на розробку ПП та наведемо їх у табл. 2.8.

Таблиця 2.8. Розрахунок матеріальних витрат на розробку

| Найменування матеріальних витрат | Тип, модель | Кількість | Ціна одиниці, грн. | Вартість, грн. |
|--|-------------|-----------|--------------------|--|
| Папір | Лист А4 | 70 | 4.0 | 280,0 |
| Разом | - | - | - | $V_{M1}=280,0$ |
| Транспортно – заготівельні Витрати (10%) | | | | $V_{тр_з} = 0.1 * V_{M1} = 0,1*280 = 28,00$ |
| Усього | | | | $V_M = V_{M1} + V_{тр_з} = 308,00$ |

За отриманими даними складена калькуляція планової собівартості ПП, наведена в табл. 2.9.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------------|------|
| | | | | | РП 08. 01 002. 00 ДП ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 64 |

Таблиця 2.9. Розрахунок статей витрат планової собівартості

| Стаття витрат | Значення, грн. | Формула розрахунку |
|---|-------------------|---|
| 1. Матеріали | 308,00 | V_M (див. табл. 2.8) |
| 2. Основна заробітна плата | 15628,48 | Z_o (див. табл. 2.7) |
| 3. Додаткова заробітна плата | 1562,84 | $Z_d = 0.1 * Z_o = 15628,48 * 0,1$ |
| 4. Відрахування до єдиного фонду соціального внеску | 3864,37 | $V_{с.с.в.} = 0.22 * (Z_o + Z_d) = 0,22 * (15628,48 + 1562,84)$ |
| 5. Накладні витрати | 6251,39 | $V_{нак.} = 0.4 * Z_o = 0.4 * 15628,48$ |
| 6. Повна собівартість | 27681,08 | $C_{пов} = V_M + Z_o + Z_d + V_{с.с.в.} + V_{нак.} = 308,00 + 15628,48 + 1562,84 + 3864,37 + 6251,39$ |

Розмір прибутку розраховується за формулою:

$$П = (C_n * P) / 100 = (27681,08 * 10) / 100 = 2768,11 \text{ грн.} \quad (2.5)$$

Де p – плановий рівень рентабельності (10-15%).

Оптова ціна розраховується за формулою:

$$C_o = C_n + П = 27681,08 + 2768,11 = 30449,19 \text{ грн.} \quad (2.6)$$

За отриманими даними, ціна реалізації ПП за формулою становить:

$$C_p = C_o + ПДВ = 30449,19 + 30449,19 * 0.2 = 36539,03 \text{ грн.} \quad (2.7)$$

3 РОЗДІЛ ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

3.1 Резюме

Забезпечення безпечних умов праці є не лише чинником комфортного перебування працівників на підприємстві, але перш за все – гарантією їхнього здоров'я та працездатності, що безпосередньо впливає на ефективність і прибутковість підприємства. Високий рівень безпеки праці можливий лише за умови суворого дотримання трудового законодавства, державних стандартів України, а також спеціальних норм і правил, які спрямовані на збереження здоров'я працюючих.

У цьому розділі дипломного проєкту розглядаються умови праці програміста (оператора ПК), які повинні бути створені на підприємстві задля безпечної організації робочого процесу під час розробки веб-застосунку.

3.2 Оцінка небезпечних і шкідливих факторів, що впливають на програміста під час розробки програмного продукту

Під небезпечними факторами розуміють ті, які в певних обставинах можуть спричинити виробничі травми або раптове погіршення стану здоров'я працівника.

Якщо вплив чинника веде до хронічних захворювань чи зниження працездатності, його вважають шкідливим. При тривалому чи сильному впливі шкідливі чинники можуть набути рис небезпечних.

Під час роботи за ПК працівник піддається дії таких небезпечних і шкідливих факторів:

- невідповідність параметрів мікроклімату нормативним вимогам;
- недостатній рівень освітленості робочого місця;
- ризик ураження електричним струмом;
- вплив статичної електрики;
- неправильна організація робочого простору;
- тривале статичне положення тіла, що призводить до перенапруження м'язів;
- підвищене зорове навантаження через тривалу роботу з монітором;

| | | | | | | |
|-----|------|----------|--------|------|--------------------------------|------|
| | | | | | <i>РП 08. 01 003. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 66 |

- шум від роботи комп'ютерного обладнання чи систем охолодження.

3.3 Гігієнічні вимоги до умов виробничого середовища

Відповідно до Правил охорони праці під час використання електронно-обчислювальних машин, робоче місце користувача ПК має бути обладнане таким чином, щоб забезпечити максимальну продуктивність праці. Розглянемо ключові умови.

3.3.1 Вимоги до приміщення

Для приміщень, де встановлені відеотермінали, рекомендується розташовувати вікна на північ або північно-східну сторону. Віконні прорізи повинні бути оснащені регульованими жалюзі або шторами, що дозволяють повністю закривати світло. Приміщення має бути забезпечене як природним, так і штучним освітленням відповідно до ДБН В.2.5-28-2018 «Природне і штучне освітлення». Також у виробничих приміщеннях повинні бути облаштовані кімнати для відпочинку та психологічного розвантаження працівників. На кожне робоче місце користувача ПК повинна припадати площа не менше 6 кв.м, а об'єм приміщення – не менше 20 куб.м. Стіни фарбують матовою фарбою згідно з санітарними нормами.

3.3.2 Освітлення

Освітлення робочої зони користувача має бути змішаним, поєднуючи природне світло з штучним. Для загального освітлення застосовують газорозрядні лампи типу ЛД. Нормативна освітленість робочої поверхні повинна складати 300–500 люкс.

3.3.3 Шум

Для роботи, що потребує високої концентрації уваги, допустимий рівень шуму не повинен перевищувати 50 дБ. Для зниження шумового і вібраційного впливу обладнання розташовують на спеціальних амортизуючих прокладках. Якщо джерелом шуму є стіни, їх облицьовують звукоізоляційними матеріалами. Крім того, для додаткового зменшення шуму використовують акустичні екрани

| | | | | | | |
|-----|------|----------|--------|------|--------------------------------|------|
| | | | | | РП 08. 01 003. 00 ДП ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 67 |

або перегородки, які ефективно поглинають звукові хвилі. Регулярне технічне обслуговування обладнання також сприяє зниженню рівня вібрації та шуму, забезпечуючи комфортні умови праці. Використання шумопоглинаючих навушників для працівників може додатково підвищити комфорт у шумному середовищі.

3.3.4 Мікроклімат

Порушення нормативних параметрів мікроклімату негативно впливає на працездатність, швидкість реакції працівника та рівень помилок. Тому необхідно підтримувати оптимальні умови: температуру повітря в межах 22–25°C, вологість – 40–60%, швидкість руху повітря – 0,1–0,2 м/с. Приміщення повинно мати систему опалення та кондиціонування, що забезпечують рівномірний обігрів, циркуляцію й очищення повітря від пилу й шкідливих речовин.

3.3.5 Електробезпека

Електричний струм, проходячи через організм, може викликати термічні, електролітичні та біологічні ефекти. Для уникнення уражень електричним струмом необхідно:

- суворо дотримуватись правил проведення робіт і технічної експлуатації обладнання;
- виключити доступ працівників до оголених чи неізольованих частин, які працюють під небезпечною напругою;
- застосовувати ізоляційні матеріали для захисту;
- забезпечити надійне заземлення конструкцій (опалювальні батареї, водопровідні труби, кабелі з заземленим екраном), що перебувають у приміщенні з робочими місцями, із використанням діелектричних щитків або захисних сіток, щоб виключити контакт із струмом.

3.3.6 Вимоги до організації робочого місця

Робочі зони повинні розташовуватись так, щоб у поле зору працівника не потрапляли віддзеркалювальні поверхні, вікна та джерела світла. Відеотермінали розміщують під кутом 90–100 градусів до вікон, щоб світло падало збоку. Не

| | | | | | | |
|-----|------|----------|--------|------|--------------------------------|------|
| | | | | | <i>РП 08. 01 003. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 68 |

допускається, щоб працівник сидів обличчям або спиною до вікон при будь-якому типі загального освітлення.

Робочий стіл повинен регулюватися по висоті в межах 680–800 мм, мати ширину, яка забезпечує зону досяжності для рук. Рекомендовані розміри: висота 725 мм, ширина 600–1400 мм, глибина 800–1000 мм. Стілець має бути обладнаний механізмами для регулювання висоти сидіння та спинки, а також кута нахилу останньої. Всі регулювання повинні бути легкими у виконанні, незалежними і надійно фіксованими.

Екран ВДТ розташовується під кутом приблизно $+30^\circ$ до нормальної лінії погляду по вертикалі для зручності зорового спостереження. Клавіатуру розміщують на столі на відстані 100–300 мм від його краю, зверненого до користувача.

Всю організацію робочого місця потрібно будувати відповідно до ергономічних вимог, враховуючи взаємне розташування всіх елементів

На рис. 3.1 представлено робоче місце і робоча поза користувача комп'ютера.

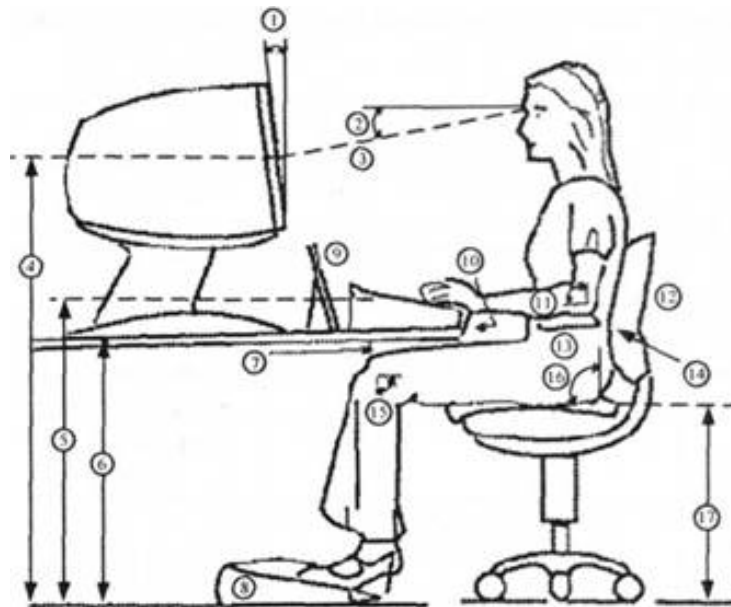


Рисунок 3.1. Робоче місце і робоча поза користувача комп'ютера

Робоче місце і робоча поза користувача комп'ютера включає:

1 – кут екрана; 2 – кут огляду (зору); 3 – відстань огляду; 4 – висота середини екрана; 5 – висота клавіатури; 6 – висота столу; 7 – відстань колін від столу; 8 – підставка для ніг; 9 – підставка для документів; 10 – положення рук; 11 – кут ліктів;

12 – спинка крісла; 13 – підлокітник; 14 – опора для попереку; 15 – кут колін; 16 – кут спинки крісла; 17 – висота сидіння

3.4 Пожежна безпека

Пожежна безпека приміщень з електричними мережами регламентується стандартами ГОСТ 12.1.033-81 та ГОСТ 12.1.004-85. Для роботи за ЕОМ приміщення має відповідати категорії Д за пожежною безпекою, тобто бути виготовленим з негорючих матеріалів у холодному стані.

Пожежна безпека забезпечується комплексом заходів:

- системою запобігання пожежам;
- системою протипожежного захисту;
- організаційно-технічними заходами.

Захист включає автоматичну пожежну сигналізацію, наявність засобів пожежогасіння та організацію евакуації людей.

Для ліквідації невеликих осередків вогню на початковій стадії застосовують первинні засоби пожежогасіння: вогнегасники (порошкові, вуглекислотні), пожежні покривала з негорючих матеріалів, ящики з піском, бочки з водою.

На рис. 3.2 представлено первинні засоби пожежогасіння.



Рисунок 3.2. Первинні засоби пожежогасіння

| | | | | |
|-----|------|----------|--------|------|
| | | | | |
| Зм. | Арк. | № докум. | Підпис | Дата |

РП 08. 01 003. 00 ДП ПЗ

Арк.

70

ВИСНОВКИ

У процесі виконання дипломної роботи було досліджено актуальність проблеми раціонального використання енергетичних ресурсів в Україні та світі. Зважаючи на виклики сучасності, такі як енергетична криза, підвищення тарифів та потреба у збереженні навколишнього середовища, свідоме енергоспоживання набуває особливої ваги.

В рамках роботи було спроектовано та реалізовано веб-застосунок EcoWatt, який дозволяє користувачам:

- розраховувати вартість електроспоживання пристроїв;
- отримувати рекомендації щодо оптимізації використання електроенергії;
- візуалізувати дані споживання у вигляді графіків;
- оперативно виявляти потенційні проблеми у енергоспоживанні.

При розробці системи були використані сучасні технології веб-програмування, такі як Angular, TypeScript, ApexCharts, а також застосовано принципи адаптивної верстки та оптимізації продуктивності інтерфейсу. Особливу увагу було приділено забезпеченню зручності для кінцевого користувача за допомогою інтуїтивного дизайну, інтерактивних підказок та системи миттєвого зворотного зв'язку.

В процесі роботи також було проведено аналіз існуючих рішень на ринку та виявлено їх основні недоліки, що дозволило запропонувати унікальне рішення, орієнтоване на простоту використання, функціональність та адаптивність до різних потреб споживачів.

Поставлені цілі дипломного проектування були досягнуті. Розроблений застосунок EcoWatt має потенціал стати корисним інструментом для широкого кола користувачів, сприяючи формуванню культури свідомого споживання енергоресурсів та збереженню довкілля.

У перспективі можливим є подальший розвиток системи шляхом інтеграції зі смарт-розетками, впровадження персоналізованих порад на основі штучного інтелекту, а також розширення функціоналу для корпоративного використання.

| | | | | | | | | | |
|-----|------|----------|--------|------|-------------------------|--|--|--|------|
| | | | | | | | | | Арк. |
| | | | | | | | | | 71 |
| Зм. | Арк. | № докум. | Підпис | Дата | РП 08. 01 000. 00 ДП ПЗ | | | | |

ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Мороз, О. В. Основи веб-програмування: HTML, CSS, JavaScript, Angular. – Київ: Академвидав, 2021. – 312 с.
2. Коваль, П. М. Сучасна веб-розробка: від HTML до Angular. – Львів: ЛНУ імені Івана Франка, 2020. – 276 с.
3. Krill, M. Angular Development with TypeScript. – Shelter Island: Manning Publications, 2021. – 400 p.
4. Fain, Y., Moiseev, A. Modern Web Development with Angular. – Boston: Addison-Wesley, 2020. – 416 p.
5. Leron, P. Efficient Frontend Development with Angular and TypeScript. – New York: Apress, 2022. – 320 p.
6. Національна комісія, що здійснює державне регулювання у сферах енергетики та комунальних послуг України (НКРЕКП). Стандарти якості електропостачання та надання компенсацій [Електронний ресурс]. – Режим доступу: <https://www.nerc.gov.ua/sferi-diyalnosti/elektroenergiya/yakist-elektropostachannya/standarti-yakosti-elektropostachannya-ta-nadannya-kompensacij> – Назва з екрана. – Дата звернення: 20.05.2025.
7. Національний інститут стратегічних досліджень. Про відповідність Енергетичної стратегії України на період до 2030 року сучасним викликам і загрозам у сфері енергетичної безпеки [Електронний ресурс]. – Режим доступу: <https://niss.gov.ua/doslidzhennya/nacionalna-bezpeka/pro-vidpovidnist-energetichnoi-strategii-ukraini-na-period-do-2030> – Назва з екрана. – Дата звернення: 20.05.2025.
8. Офіційний портал YASNO [Електронний ресурс]. – Режим доступу: <https://yasno.com.ua/> – Назва з екрана. – Дата звернення: 21.05.2025.
9. Офіційний портал DTEK Одеса [Електронний ресурс]. – Режим доступу: <https://www.dtek-oem.com.ua/ua> – Назва з екрана. – Дата звернення: 22.05.2025.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------------|------|
| | | | | | <i>РП 08. 01 000. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 72 |

ДОДАТОК А. Фрагмент коду компоненту розрахунків електроспоживання

```
// src/app/pages/view-results/view-results.component.ts

import {
  AfterViewInit,
  ChangeDetectionStrategy,
  Component,
  computed,
  DestroyRef,
  inject,
  signal,
  viewChildren
} from '@angular/core';
import { takeUntilDestroyed } from '@angular/core/rxjs-interop';
import { EnergyChartComponent } from '../../../components/charts/energy-chart/energy-chart.component';
import { MatIcon } from '@angular/material/icon';
import {
  MatFabButton,
  MatIconButton,
  MatMiniFabButton
} from '@angular/material/button';
import { MatDialog } from '@angular/material/dialog';
import { AppRoutes } from '../../../enums/app-routes';
import { type IDeviceModel } from '../../../interfaces/device-model.interface';
import { MatTooltip } from '@angular/material/tooltip';
import { DeviceService } from '../../../services/device/device.service';
import { TariffService } from '../../../services/tariff/tariff.service';
import { SelectDeviceComponent } from '../../../dialogs/select-device/select-device.component';
import { RemoveDeviceComponent } from '../../../dialogs/remove-device/remove-device.component';
import { filter } from 'rxjs';
import { ManageDeviceComponent } from '../../../dialogs/manage-device/manage-device.component';
import { TotalCostCardComponent } from '../../../components/cards/total-cost-card/total-cost-card.component';
import { TotalConsumptionCardComponent } from '../../../components/cards/total-consumption-card/total-consumption-card.component';
import { DeviceWattageCardComponent } from '../../../components/cards/device-wattage-card/device-wattage-card.component';
import {
  MatChipListbox,
  MatChipOption,
  MatChipSelectionChange
} from '@angular/material/chips';
import { MatToolbar } from '@angular/material/toolbar';
import { BasePageDirective } from '../../../core/base-page/base-page.directive';
import { fadeInAnimation, growUpAnimation } from '../../../animations';
import {
  DELETE_DEVICE_BTN,
  EDIT_DEVICE_BTN,
  ADD_DEVICE_BTN,
  DEVICE_SELECTION,
  TOTAL_COST_CARD,
  TOTAL_CONSUMPTION_CARD,
  DEVICE_WATTAGE_CARD,
  ENERGY_CHART
} from '../../../constants/tour-ids';
```

```

import { ShepherdService } from 'angular-shepherd';

@Component({
  selector: 'app-view-results',
  imports: [
    EnergyChartComponent,
    MatIcon,
    MatTooltip,
    MatFabButton,
    MatIconButton,
    TotalCostCardComponent,
    TotalConsumptionCardComponent,
    DeviceWattageCardComponent,
    MatChipListbox,
    MatChipOption,
    MatMiniFabButton,
    MatToolbar
  ],
  animations: [growUpAnimation, fadeInAnimation],
  templateUrl: './view-results.component.html',
  styleUrls: ['./view-results.component.scss'],
  standalone: true,
  changeDetection: ChangeDetectionStrategy.OnPush
})
export class ViewResultsComponent
  extends BasePageDirective
  implements AfterViewInit
{
  readonly #dialog = inject(MatDialog);
  readonly #deviceService = inject(DeviceService);
  readonly #tariffService = inject(TariffService);
  readonly #destroyRef = inject(DestroyRef);
  readonly #shepherdService = inject(ShepherdService);

  protected readonly AppRoutes = AppRoutes;

  readonly deviceChips = viewChildren('deviceChip', { read: MatChipOption });

  readonly devices = this.#deviceService.devices;
  readonly tariff = this.#tariffService.tariff;

  readonly totalWatts = computed(() => {
    const selectedDevices = this.selectedDevices();

    return selectedDevices.reduce(
      (accumulator, device) =>
        accumulator +
        device.powerWatts *
        device.hoursPerDay *
        device.daysPerMonth *
        device.energyClass,
      0
    );
  });

  readonly #selectedDevices = signal<IDeviceModel[]>([]);
  readonly selectedDevices = this.#selectedDevices.asReadonly();

  constructor() {
    super();
  }

  ngAfterViewInit(): void {
    if (this.devices().length === 1) {

```

```

    this.deviceChips()[0].select();
  }

  this.#shepherdService.modal = true;
  this.#shepherdService.confirmCancel = false;

  const defaultStepOptions = {
    classes: 'shepherd-theme-default',
    scrollTo: true,
    cancelIcon: {
      enabled: true
    }
  };

  this.#shepherdService.defaultStepOptions = defaultStepOptions;

  this.#shepherdService.addSteps([
    {
      id: DEVICE_SELECTION,
      attachTo: {
        element: `#${DEVICE_SELECTION}`,
        on: 'bottom'
      },
      title: 'Вибір пристроїв',
      text: 'Тут ви можете вибрати пристрої, які хочете проаналізувати. Натисніть на чіп, щоб вибрати пристрій.',
      buttons: [
        {
          text: 'Далі',
          action: () => this.#shepherdService.next()
        }
      ]
    },
    {
      id: DELETE_DEVICE_BTN,
      attachTo: {
        element: `#${DELETE_DEVICE_BTN}`,
        on: 'bottom'
      },
      title: 'Видалення пристрою',
      text: 'Натисніть цю кнопку, щоб видалити вибраний пристрій.',
      buttons: [
        {
          text: 'Назад',
          action: () => this.#shepherdService.back()
        },
        {
          text: 'Далі',
          action: () => this.#shepherdService.next()
        }
      ]
    },
    {
      id: EDIT_DEVICE_BTN,
      attachTo: {
        element: `#${EDIT_DEVICE_BTN}`,
        on: 'bottom'
      },
      title: 'Редагування пристрою',
      text: 'Натисніть цю кнопку, щоб редагувати вибраний пристрій.',
      buttons: [
        {
          text: 'Назад',
          action: () => this.#shepherdService.back()
        }
      ]
    }
  ]
);

```

```

    },
    {
      text: 'Далі',
      action: () => this.#shepherdService.next()
    }
  ]
},
{
  id: ADD_DEVICE_BTN,
  attachTo: {
    element: `#${ADD_DEVICE_BTN}`,
    on: 'left'
  },
  title: 'Додавання пристрою',
  text: 'Натисніть цю кнопку, щоб додати новий пристрій.',
  buttons: [
    {
      text: 'Назад',
      action: () => this.#shepherdService.back()
    },
    {
      text: 'Далі',
      action: () => this.#shepherdService.next()
    }
  ]
},
{
  id: TOTAL_COST_CARD,
  attachTo: {
    element: `#${TOTAL_COST_CARD}`,
    on: 'bottom'
  },
  title: 'Загальна вартість',
  text: 'Тут ви можете побачити загальну вартість споживання електроенергії  
вибраними пристроями.',
  buttons: [
    {
      text: 'Назад',
      action: () => this.#shepherdService.back()
    },
    {
      text: 'Далі',
      action: () => this.#shepherdService.next()
    }
  ]
},
{
  id: TOTAL_CONSUMPTION_CARD,
  attachTo: {
    element: `#${TOTAL_CONSUMPTION_CARD}`,
    on: 'bottom'
  },
  title: 'Загальне споживання',
  text: 'Тут ви можете побачити загальне споживання електроенергії вибраними  
пристроями.',
  buttons: [
    {
      text: 'Назад',
      action: () => this.#shepherdService.back()
    },
    {
      text: 'Далі',
      action: () => this.#shepherdService.next()
    }
  ]
}

```

```

    ]
  },
  {
    id: DEVICE_WATTAGE_CARD,
    attachTo: {
      element: `#${DEVICE_WATTAGE_CARD}`,
      on: 'bottom'
    },
    title: 'Потужність пристроїв',
    text: 'Тут ви можете побачити потужність кожного вибраного пристрою.',
    buttons: [
      {
        text: 'Назад',
        action: () => this.#shepherdService.back()
      },
      {
        text: 'Далі',
        action: () => this.#shepherdService.next()
      }
    ]
  },
  {
    id: ENERGY_CHART,
    attachTo: {
      element: `#${ENERGY_CHART}`,
      on: 'top'
    },
    title: 'Графік споживання',
    text: 'Тут ви можете побачити графік споживання електроенергії вибраними
пристроями.',
    buttons: [
      {
        text: 'Назад',
        action: () => this.#shepherdService.back()
      },
      {
        text: 'Завершити',
        action: () => this.#shepherdService.complete()
      }
    ]
  }
}
]);

this.#shepherdService.start();
}

onSelectionChange(change: MatChipSelectionChange): void {
  if (change.selected) {
    this.setSelectedDevice(change.source.value);
  } else {
    this.filterSelectedDevices(change.source.value.id);
  }
}

onEditDevice(): void {
  const selectedDevices = this.selectedDevices();

  if (selectedDevices.length > 1) {
    this.#dialog
      .open(SelectDeviceComponent, {
        maxWidth: '31.25rem',
        width: '100%',
        data: { devices: selectedDevices }
      })
  }
}

```

Розробка веб-застосунку контролю енергоспоживання в побутових умовах

Виконав студент групи 4РП-08: Антонов Ілля Владиславович

Керівник: Жадан А.С.

Основні відомості

Актуальність

Зростаюче споживання електроенергії, енергетична криза та екологічні виклики підкреслюють необхідність свідомого використання енергії.

Мета та завдання

В Україні відсутні зручні сервіси для щоденного моніторингу електроспоживання. EcoWatt – відповідь на цю проблему.

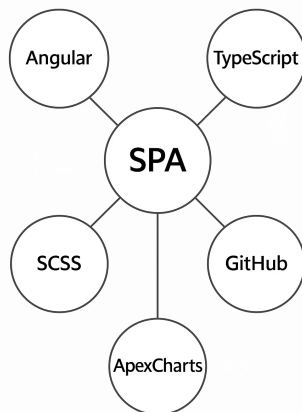
Огляд існуючих рішень

EcoWatt пропонує унікальне рішення для ефективного моніторингу та управління енергоспоживанням.

Вибір технологій

- ▶ Angular – фреймворк для SPA;
- ▶ TypeScript – мова програмування з типізацією;
- ▶ SCSS – для масштабованої стилізації;
- ▶ Firebase – для хостингу й бекенду;
- ▶ GitHub – для контролю версій; ApexCharts – побудова графіків.

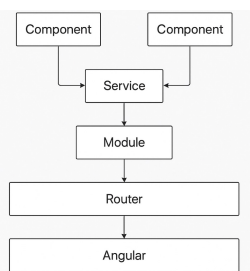
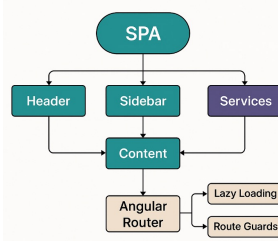
Вибір технологій



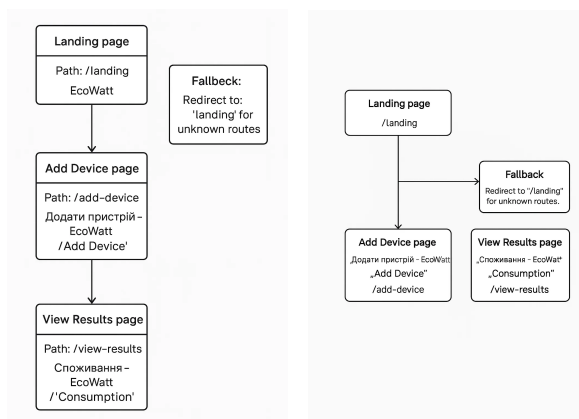
Архітектура застосунку

Проект побудовано на основі **Component-Based Architecture**. Компоненти відокремлені за функціональністю, використовуються сервіси для бізнес-логіки, маршрутизація з Lazy Loading та Route Guards.

Архітектура застосунку



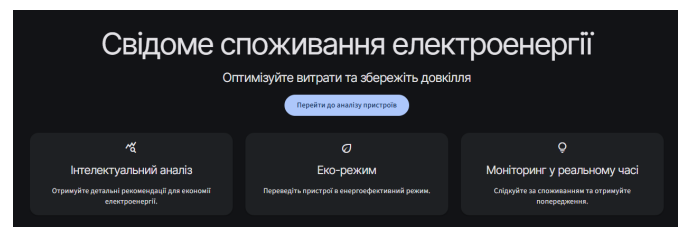
Структура каталогів



- ▶ components, pages, services, interfaces, pipes;
- ▶ Angular-модулі поділені на Core, Shared, Feature;
- ▶ Використовується Dependency Injection для підключення сервісів.

Лендінг сторінка

Лендінг створено за F-моделлю перегляду: заголовок, СТА, блоки переваг, футер. Дизайн адаптивний, реалізований з Angular Material. Використано світлу тему з екологічними кольорами (зелений, синій).

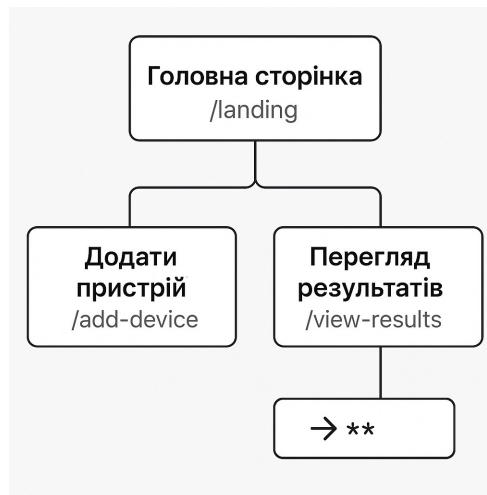


Компоненти та маршрути

Реалізовано три основні сторінки:

- ▶ /landing – презентація системи;
- ▶ /add-device – додавання пристрою;
- ▶ /view-results – перегляд результатів.

Використовується RouterLink, canActivate, заголовки задаються динамічно.



Додавання пристрою

Додати Електроприлад

Пристрій*
Наприклад, Холодильник 0/155

Потужність, Вт* Вт Енергоефективніс... ▾

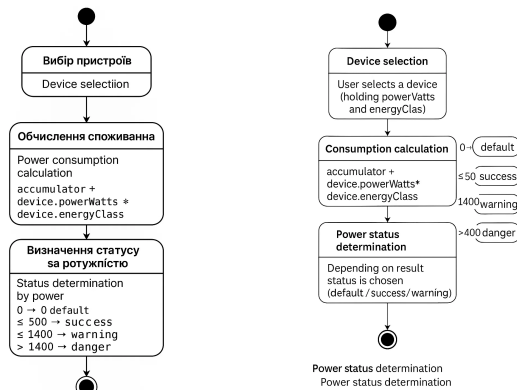
Години на день* год Дні на місяць днів

Форма для вводу параметрів:

- назва, потужність, клас енергоефективності;
- години й дні роботи, тариф.

Поля валідуються одразу, є візуальні підказки. Форма інтуїтивна та доступна.

Розрахунок споживання



Розрахунок відбувається за формулою:

$$\text{споживання} = (\text{ват} \times \text{год} \times \text{днів} \times \text{коэф.}) / 1000 \text{вартість} = \text{споживання} \times \text{тариф}$$

Використано сервіси:

- ▶ DeviceService,
- ▶ TariffService,
- ▶ AnalyzeConsumptionService

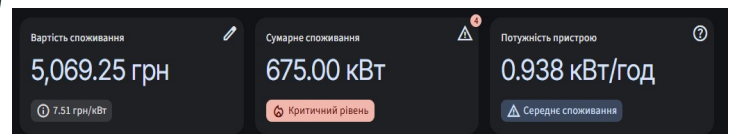
Є перевірка на критичні значення.

Інформаційні картки

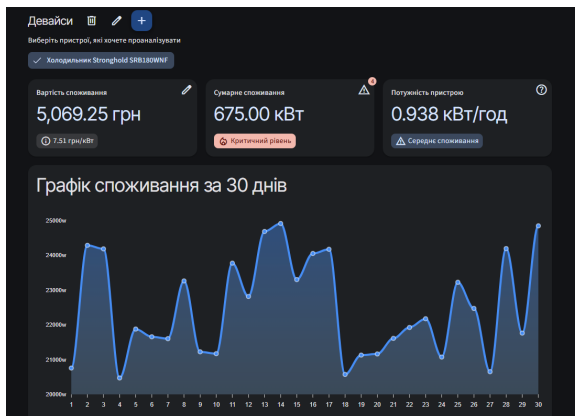
Результати виводяться у вигляді карток:

- ▶ загальна вартість (грн);
- ▶ енергоспоживання (кВт);
- ▶ середня потужність (кВт/год);
- ▶ позначки “Критичний рівень”.

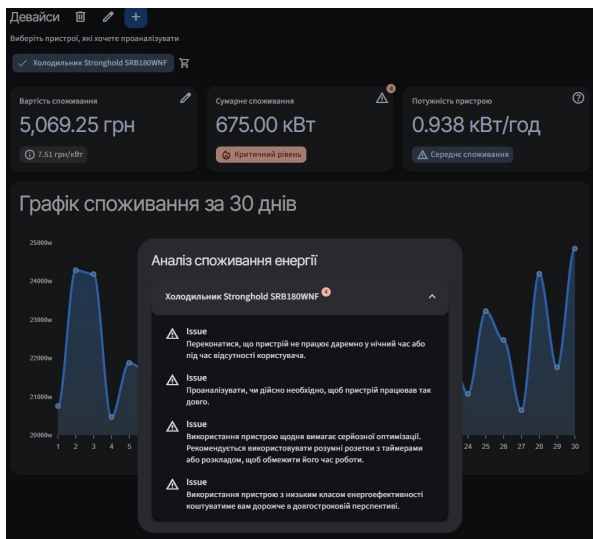
Це надає користувачеві просту й швидку оцінку ефективності.



Побудова графіків



Графіки реалізовано через **ng-apexcharts**.
Відображається динаміка споживання за 30 днів.
Графіки адаптивні, інтерактивні, підтримується фільтрація, експорт у PDF/PNG.



Підказки та хелпери

Для покращення UX додано підказки з поясненням термінів, параметрів і виявлених проблем.
Реалізовано через **matTooltip**, **MatDialog**, кастомні компоненти з діалогами.

Висновки



<https://eco-watt-790f9.web.app/landing>

EcoWatt:

- допомагає контролювати споживання;
- надає рекомендації;
- формує культуру енергозбереження.

РЕЦЕНЗІЯ

на дипломний проект здобувача (здобувачки) освіти

відділення комп'ютерних систем

Антонова Іллі Владиславовича

(прізвище, ім'я та по батькові)

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма «Розробка програмного забезпечення»

Керівник дипломного проекту (роботи) Жадан Артур Сергійович

(прізвище, ім'я та по батькові)

Тема дипломного проекту (роботи) Розробка веб-застосунку контролю енергоспоживання в побутових умовах

Обсяг розрахунково-пояснювальної записки 85 сторінок

Обсяг графічної (презентаційної) частини 13 аркушів (слайдів)

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ (РОБОТИ)

а) заключення про ступінь відповідності виконаного дипломного проекту завданню

Представлений на рецензію дипломний проект відповідає затвердженій темі та виконаний відповідно технічному завданню. Дипломний проект присвячений проблемі контролю електроспоживання та складається з пояснювальної записки, додатку з програмним кодом та мультимедійної презентації, що містить приклади роботи програми.

б) характеристика виконання кожного розділу дипломного проекту

Пояснювальна записка складається з основного розділу (аналізу предметної області, проектування застосунку, реалізації застосунку, тестування застосунку), економічного розділу, розділу охорони праці та додатків. Перелічені розділи поетапно охоплюють розробку, виконані докладно та обґрунтовано. Розділ охорони праці містить загальну інформацію та вимоги до техніки безпеки оператора КТ. Економічний розділ проекту містить розрахунок витрат на НДР та реалізацію проекту.

в) оцінка якості виконання пояснювальної записки та графічної частини дипломного проекту

Графічна частина складається з 13 слайдів мультимедійної презентації, виконаної у програмному продукті MS PowerPoint, які містять ілюстративні схеми, скріншоти роботи програмного застосунку, передбачені технічним завданням. Пояснювальна записка виконана акуратно та у відповідності до норм. Якість виконання графічної частини проекту та пояснювальної записки добра, розробку виконано у повному обсязі.

г) перелік позитивних якостей дипломного проекту Охоплено всі етапи розробки застосунку із підключенням бази даних через ADO.NET

Забезпечено взаємодію з реляційною БД, застосовано ключові концепції проектування таблиць, зв'язків та запитів. GUI містить велику кількість графічних пояснень та графіків для користувачів.

д) основні недоліки дипломного проекту

Використання великої кількості компонентів, сервісів та взаємодія між серверною частиною (PHP) і фронтендом (Angular) може призводити до підвищеної складності підтримки рішення. У розділі тестування відсутній системний підхід, не згадується юніт-тестування або перевірка граничних випадків. Незначні недоліки оформлення пояснювальної записки

Оцінка розрахункової частини Відмінно

Оцінка графічної частини Відмінно

Загальна оцінка Відмінно

Прізвище, ім'я, по батькові рецензента к.т.н. Рудніченко Микола Дмитрович

Місце роботи і посада рецензента Національний університет «Одеська політехніка», доцент кафедри інформаційних технологій

Підпис: _____

« 20 » _____ червня 2025 р.



ВІДГУК

керівника на дипломний проєкт здобувача (здобувачки) освіти
відділення комп'ютерних систем

Антонов Ілля Владиславович

(прізвище, ім'я та по батькові)

Спеціальність: 121 "Інженерія програмного забезпечення"

Освітньо-професійна програма: «Розробка програмного забезпечення»

Тема дипломного проєкту: Розробка веб-застосунку для свідомого
споживання електроенергії

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЄКТУ

а) обсяг і якість виконання проєкту (графічного матеріалу і розрахунково-пояснювальної записки) Дипломний проєкт виконано відповідно технічному завданню.

Пояснювальна записка містить 85 сторінки. У пояснювальній записці виконано опис етапів розробки веб-застосунку для свідомого споживання електроенергії, а також особливостей його програмного забезпечення.

Графічна частина складається з 12 слайдів мультимедійної презентації, які містять схеми, діаграми та скріншоти інтерфейсу, передбачені технічним завданням. Якість виконання пояснювальної записки та графічної частини добра, розробку виконано в повному обсязі.

б) самостійність роботи над проєктом: _____

Протягом всього строку дипломного проєктування та переддипломної практики здобувач освіти Антонов І.В. поступово та послідовно виконував всі етапи розробки. Всі роботи здобувач освіти виконував самостійно, з оглядом на рекомендації керівника.

в) теоретична підготовка випускника (випускниці): Здобувач освіти Антонов І.В. під час роботи над дипломним проєктом вивчив достатню кількість літературних джерел та матеріалів за даною тематикою.

Вважаю, що теоретична підготовка здобувача добра, і він готовий до захисту дипломного проєкту.

г) вміння розв'язувати виробничі та конструкторські питання _____
Під час дипломного проектування здобувач освіти Антонов І.В. мав змогу
самостійно приймати рішення щодо реалізації функціоналу вебзастосунку,
зокрема створення формул для розрахунку споживання електроенергії та її
вартості. Здобувач показав вміння організовано працювати над
поставленим завданням, розробляти логіку обчислень та реалізовувати
інтерфейс за допомогою сучасних вебтехнологій. Сайт адаптовано таким
чином, щоб будь-який користувач міг легко зрозуміти і скористатися ним.

Оцінка розрахункової частини _____ Добре _____

Оцінка графічної частини _____ Відмінно _____

Загальна оцінка _____ Відмінно _____

Прізвище, ім'я, по батькові керівника дипломного проекту _____

_____ Жадан Артур Сергійович _____

Місце роботи і посада керівника дипломного проекту _____

_____ ВСП "Одеський технічний фаховий коледж ОНТУ", викладач _____
_____ специдисциплін комісії комп'ютерних технологій та програмної інженерії _____

Підпис _____

_____ «16» червня 2025 р.

**ДОЗВІЛ
НА РОЗМІЩЕННЯ
ВИПУСКНОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
(ДИПЛОМНОГО ПРОЕКТУ)
В ЕЛЕКТРОННОМУ РЕПОЗИТАРІЇ ВСП «ОТФК ОНТУ»**

Ми, що нижче підписалися,

Антонов Ілля Владиславович,
здобувач освіти гр. 4РП-08, та

Жадан Артур Сергійович,
керівник дипломного проекту,

не заперечуємо щодо розміщення електронного варіанту пояснювальної записки до дипломного проекту фахового молодшого бакалавра на тему:

«Розробка веб-застосунку контролю енергоспоживання в побутових умовах» (автор роботи – Антонов І.В., керівник роботи – Жадан А.С.)

виконаного у ВСП «Одеський технічний фаховий коледж Одеського національного технологічного університету» в 2025 році, у повному обсязі в електронному репозитарії ВСП «ОТФК ОНТУ» для вільного доступу через мережу Інтернет.

Несемо відповідальність за ідентичність електронного та друкованого варіантів випускної кваліфікаційної роботи і даємо згоду на обробку персональних даних.

Виконавець



/ Антонов І.В. /

Керівник



/ Жадан А.С. /

«16» червня 2025 р.

ДОВІДКА

циклової комісії КТ та ПІ
про допуск до захисту дипломного проєкту
здобувача (здобувачки) освіти IV курсу
відділення комп'ютерних систем групи 4РП-08

Антонова Іллі Владиславовича

на тему Розробка веб-застосунку контролю енергоспоживання
в побутових умовах

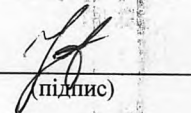
Висновок відповідальної особи за проведення нормоконтролю:
пояснювальна записка до дипломного проєкту виконана з некритичними
порушеннями ДСТУ та оформлена відповідно до вимог Положення про
дипломне проєктування


(підпис)

16.06.2025
(дата)

Петрашова В.І.
(П.І.Б.)

Висновок відповідальної особи за перевірку роботи на наявність академічного
плагіату згідно звіту про перевірку від 12.06.2025 р. значення коефіцієнту
подібності в роботі становить 13,04%, коефіцієнт цитування – 1,49%.


(підпис)

16.06.2025
(дата)

Краснокутська К.Г.
(П.І.Б.)

Попередня експертиза (малий захист) дипломного проєкту

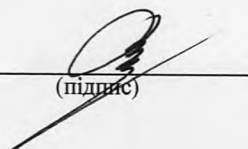
здобувача (здобувачки) освіти

Антонова І.В.
(П.І.Б.)

проведена « 16 » червня 2025 р.

Висновки Пояснювальна записка до дипломного проєкту виконана у повному
обсязі. Випускна кваліфікаційна робота (дипломний проєкт) відповідає
вимогам Положення про дипломне проєктування та рекомендована до
захисту.

Голова ЦК КТ та ПІ


(підпис)

Кривченко Ю.В.
(П.І.Б.)

Звіт подібності

метадані

Назва організації

Odesa Technical Professional College of Odesa National University of Technology

Заголовок

Розробка веб-застосунку контролю енергоспоживання в побутових умовах

Автор

Науковий керівник / Експерт

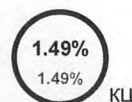
Антонов Ілля Владиславович Жадан Артур Сергійович

підрозділ

Відокремлений структурний підрозділ "Одеський технічний фаховий коледж Одеського національного технологічного університету"

Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.



25

Довжина фрази для коефіцієнта подібності 2

15320

Кількість слів

133044

Кількість символів

Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

| | | |
|------------------------|--|-----|
| Заміна букв | | 15 |
| Інтервали | | 0 |
| Мікропробіли | | 2 |
| Білі знаки | | 531 |
| Парафрази (SmartMarks) | | 95 |

Подібності за списком джерел

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Колір тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

10 найдовших фраз

Копір тексту

| ПОРЯДКОВИЙ НОМЕР | НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ) | КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ) |
|---------------------|---|---|
| 1 | https://card-file.ontu.edu.ua/bitstreams/34a6756b-592f-4b77-a805-183aa03a6a26/download | 62 0.40 % |
| 2 | https://card-file.ontu.edu.ua/server/api/core/bitstreams/995bdcec-4e4d-4321-8070-4d6badcb8e49/content | 40 0.26 % |
| 3 | https://card-file.ontu.edu.ua/bitstreams/1dff552d-7200-49b8-ae1d-ba76a1335685/download | 39 0.25 % |
| 4 | https://card-file.ontu.edu.ua/bitstreams/6cf43324-8f08-4031-ba42-f80b18efbbc8/download | 32 0.21 % |
| 5 | https://card-file.ontu.edu.ua/bitstreams/29489599-0581-4ce6-8890-c3b13d9f2e0e/download | 31 0.20 % |

| | | |
|----|---|-----------|
| 6 | https://card-file.ontu.edu.ua/server/api/core/bitstreams/a141b658-5fa7-4f90-b0bd-7f0ccaed21e5/content | 31 0.20 % |
| 7 | https://card-file.ontu.edu.ua/bitstreams/1dff552d-7200-49b8-ae1d-ba76a1335685/download | 29 0.19 % |
| 8 | https://card-file.ontu.edu.ua/server/api/core/bitstreams/c5cd348b-fc64-4a25-9a5b-6cc8d62db909/content | 27 0.18 % |
| 9 | https://card-file.ontu.edu.ua/bitstreams/1dff552d-7200-49b8-ae1d-ba76a1335685/download | 26 0.17 % |
| 10 | https://card-file.ontu.edu.ua/bitstreams/1dff552d-7200-49b8-ae1d-ba76a1335685/download | 25 0.16 % |

з домашньої бази даних (0.05 %)

| ПОРЯДКОВИЙ НОМЕР | ЗАГОЛОВОК | КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ) |
|---------------------|--|---|
| 1 | Створення web-застосунку цифрового помічника з використанням Open AI 5/28/2025 Odesa Technical Professional College of Odesa National University of Technology (Відокремлений структурний підрозділ "Одеський технічний фаховий коледж Одеського національного технологічного університету") | 7 (1) 0.05 % |

з програми обміну базами даних (0.18 %)

| ПОРЯДКОВИЙ НОМЕР | ЗАГОЛОВОК | КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ) |
|---------------------|--|---|
| 1 | 2021_61220009_Antoniuk_Dmytro_Volodymyrovych_89254 10/25/2024 National University "Lviv Politechnika" (National University Lviv Politechnika) | 17 (3) 0.11 % |
| 2 | bitstream_f050bfe4-889e-4927-ba32-3252933ca7a0 12/8/2024 National Technical University "Kharkiv Polytechnic Institute" students papers (National Technical University "Kharkiv Polytechnic Institute" students papers) | 11 (1) 0.07 % |

з Інтернету (12.81 %)

| ПОРЯДКОВИЙ НОМЕР | ДЖЕРЕЛО URL | КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ) |
|---------------------|---|---|
| 1 | https://card-file.ontu.edu.ua/server/api/core/bitstreams/44c16132-5f53-48e2-b6c0-61e9a2f0fd75/content | 644 (53) 4.20 % |
| 2 | https://card-file.ontu.edu.ua/bitstreams/1dff552d-7200-49b8-ae1d-ba76a1335685/download | 416 (30) 2.72 % |
| 3 | https://card-file.ontu.edu.ua/server/api/core/bitstreams/a141b658-5fa7-4f90-b0bd-7f0ccaed21e5/content | 118 (9) 0.77 % |
| 4 | https://card-file.ontu.edu.ua/server/api/core/bitstreams/c5cd348b-fc64-4a25-9a5b-6cc8d62db909/content | 103 (8) 0.67 % |
| 5 | https://card-file.ontu.edu.ua/bitstreams/53ed22ad-8700-4162-b97a-082a1ad472d6/download | 98 (8) 0.64 % |
| 6 | https://card-file.ontu.edu.ua/bitstreams/34a6756b-592f-4b77-a805-183aa03a6a26/download | 71 (2) 0.46 % |
| 7 | https://card-file.ontu.edu.ua/bitstreams/6cf43324-8f08-4031-ba42-f80b18efbbc8/download | 59 (4) 0.39 % |
| 8 | https://card-file.ontu.edu.ua/bitstreams/035f6436-20b4-4ee6-8e99-bede670e308b/download | 55 (5) 0.36 % |
| 9 | https://card-file.ontu.edu.ua/bitstreams/29489599-0581-4ce6-8890-c3b13d9f2e0e/download | 54 (3) 0.35 % |
| 10 | https://card-file.ontu.edu.ua/server/api/core/bitstreams/995bdcec-4e4d-4321-8070-4d6badcb8e49/content | 52 (2) 0.34 % |
| 11 | https://card-file.ontu.edu.ua/bitstreams/bbed74c8-2ea7-44c5-8d00-0fe3fd9790ee/download | 45 (4) 0.29 % |

