

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ТЕХНОЛОГІЙ ТА ДИЗАЙНУ  
ФАКУЛЬТЕТ МЕХАТРОНИКИ ТА КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ  
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ТЕХНОЛОГІЙ

## КВАЛІФІКАЦІЙНА РОБОТА

на тему:

Розроблення програмного забезпечення для прогнозування використання енергоресурсів компанії

Рівень вищої освіти	<u>другий (магістерський)</u>
Спеціальність 122	<u>Комп'ютерні науки</u>
Освітня програма	<u>Комп'ютерні науки</u>

Виконав: студент групи МгІТ-1-21

Олексій ТКАЛЕНКО

Науковий керівник: к.т.н., доц. Тетяна ДЕМКІВСЬКА

Рецензент: д.т.н., проф. Віктор ЧУПРИНКА

Київ – 2023

# КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ТЕХНОЛОГІЙ ТА ДИЗАЙНУ

Факультет мехатроніки та комп'ютерних технологій

Кафедра комп'ютерні науки

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки

Освітня програма Комп'ютерні науки

## **ЗАТВЕРДЖУЮ**

Завідувач кафедри КН

\_\_\_\_\_ Володимир ЩЕРБАНЬ

« \_\_\_ » \_\_\_\_\_ 2023 року

## **З А В Д А Н Н Я**

### **НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТА**

#### **Ткаленку Олексію Сергійовичу**

1. Тема роботи Розроблення програмного забезпечення для прогнозування використання енергоресурсів компанії

Науковий керівник роботи Демківська Тетяна Іванівна, к.т.н., доц.,  
затверджені наказом закладу вищої освіти від “12” 09. 2023 року № 210-уч

2. Вихідні дані до кваліфікаційної роботи:

Розробка кафедри комп'ютерних наук

3. Зміст дипломної роботи (перелік питань, які потрібно розробити)

Вступ, РОЗДІЛ 1. Аналіз потреб та вимог до програмного забезпечення прогнозування використання енергоресурсів компанії; РОЗДІЛ 2. Етапи розробки програмного забезпечення для прогнозування використання енергоресурсів компанії; РОЗДІЛ 3. Описання розробленого програмного забезпечення прогнозування використання енергоресурсів компанії; Додатки – програмні коди модулів системи.

4. Дата видачі завдання 08.2023р..

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної магістерської роботи	Терміни виконання етапів	Примітка про виконання
1	Вступ	20.09.2023	
2	Розділ 1. Основні принципи роботи системи керування вмістом WordPress	03.10.2023	
3	Розділ 2. Розробка музичного блогу на WordPress	10.10.2023	
4	Розділ 3. Програмна реалізація додатку для відображення глибини прослуховування	19.10.2023	
5	Висновки	24.10.2023	
6	Оформлення кваліфікаційної роботи (чистовий варіант)	06.11.2023	
7	Подача кваліфікаційної роботи (проекту) науковому керівнику для відгуку (за 14 днів дозахисту)	08.11.2023	
8	Подача кваліфікаційної роботи (проекту) для рецензування	08.11.2023	
9	Перевірка кваліфікаційної роботи на наявність ознак плагіату		
10	Подання кваліфікаційної роботи на затвердження завідувачу кафедри	10.11.2023	

З завданням ознайомлений:

Студент

\_\_\_\_\_

Олексій ТКАЛЕНКО

Науковий керівник

\_\_\_\_\_

Тетяна ДЕМКІВСЬКА

## АНОТАЦІЯ

**Ткаленко О.С.** Розроблення програмного забезпечення для прогнозування використання енергоресурсів компанії.

Кваліфікаційна робота за спеціальністю 122 – «Комп'ютерні науки та технології». – Київський національний університет технологій та дизайну, Київ, 2022 рік.

В кваліфікаційній роботі було розроблено додаток, призначений для прогнозування використання енергоресурсів компанії. Для прогнозування використовувався метод поліноміальної регресії, який було вбудовано в онлайнове програмне забезпечення, що написано мовою програмування PHP з використанням СУБД MySQL. Це програмне забезпечення дозволить підприємствам впроваджувати ефективні стратегії енергозбереження та раціонального використання ресурсів.

Робота розкриває актуальну проблематику в галузі управління енергетичними ресурсами та пропонує практичні рішення для її вирішення.

**Ключові слова:** програмне забезпечення, прогнозування, енергоресурси, енергозбереження, ресурсна ефективність.

## ANNOTATION

**Tkalenko O.S.** Development of software for forecasting energy resource usage of a company.

Qualification thesis in the field of 122 - "Computer Science and Technology". - Kyiv National University of Technologies and Design, Kyiv, 2023.

The qualification work includes the development of an application designed for forecasting the use of energy resources by a company. The polynomial regression method was employed and integrated into the online software developed For forecasting with using the PHP programming language and the MySQL database management system. This software will enable enterprises to implement effective energy-saving strategies and rational resource utilization.

The work addresses the current issues in the field of energy resource management and provides practical solutions for their resolution.

**Keywords:** software, forecasting, energy resources, energy conservation, resource efficiency.

## ЗМІСТ

<b>ВСТУП .....</b>	<b>7</b>
<b>РОЗДІЛ 1. АНАЛІЗ ПОТРЕБ ТА ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПРОГНОЗУВАННЯ ВИКОРИСТАННЯ ЕНЕРГОРЕСУРСІВ КОМПАНІЇ .....</b>	<b>10</b>
1.1 Аналіз потреб та вимог компаній при прогнозуванні використання енергоресурсів .....	10
1.2 Вимоги до прогнозування енергоспоживання підприємства .....	12
1.3 Огляд існуючих підходів до використання програмного забезпечення при прогнозуванні використання енергоресурсів компанії .....	15
1.4 Аналіз існуючого програмного забезпечення аналізу та оптимізації споживання енергоресурсів .....	19
Висновки до розділу 1 .....	24
<b>РОЗДІЛ 2. ЕТАПИ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ПРОГНОЗУВАННЯ ВИКОРИСТАННЯ ЕНЕРГОРЕСУРСІВ КОМПАНІЇ....</b>	<b>25</b>
2.1 Вибір вхідних даних та методів прогнозування .....	25
2.2 Збір та підготовка даних .....	27
2.3 Обґрунтування архітектури програмного забезпечення прогнозування використання енергоресурсів компанії .....	32
Висновки до розділу 2 .....	42
<b>РОЗДІЛ 3. ОПИСАННЯ РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПРОГНОЗУВАННЯ ВИКОРИСТАННЯ ЕНЕРГОРЕСУРСІВ КОМПАНІЇ .....</b>	<b>44</b>
3.1 Описання компонентів розробленого програмного забезпечення .....	44
3.2 Приклади застосування додатку .....	57
Висновки до розділу 3 .....	66
<b>ВИСНОВКИ.....</b>	<b>67</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....</b>	<b>69</b>
<b>ДОДАТКИ .....</b>	<b>76</b>

## ВСТУП

**Актуальність роботи.** Однією з найактуальніших проблем сучасного світу є раціональне та ефективне використання енергоресурсів в промисловості, комерційних та господарських структурах. Зростання витрат енергії, вплив на довкілля та нестабільність цін на енергоресурси роблять необхідним розробку та впровадження програмного забезпечення, яке б допомогло компаніям ефективно управляти та прогнозувати використання енергоресурсів.

Системи обліку та прогнозування витрат енергоресурсів стали надзвичайно важливими в сучасному світі, особливо для підприємств та організацій, які великі споживачі енергії. Однією з основних проблем, які стоять перед такими організаціями, є необхідність ефективно використовувати ресурси та раціонально керувати енергоспоживанням.

Для досягнення цих цілей необхідно мати систему, яка б дозволила збирати, аналізувати та візуалізувати дані про витрати енергоресурсів. Така система допомагає підприємствам контролювати та оптимізувати своє енергоспоживання, що може призвести до значних економічних вигід та сприяти збереженню енергоресурсів та захисту довкілля.

Розробка програмного забезпечення для прогнозування використання енергоресурсів компанії є актуальною та необхідною з кількох ключових причин:

- ефективне управління енергоспоживанням: підприємства та організації щодня споживають великі обсяги електроенергії, газу, води та інших ресурсів. Завдяки програмному забезпеченню для прогнозування витрат, вони можуть ефективно контролювати своє енергоспоживання та уникати надмірних витрат;

- раціональне використання ресурсів: прогнозування дозволяє підприємствам раціонально використовувати енергоресурси, зменшуючи їхні витрати та сприяючи збереженню цих ресурсів для майбутніх потреб;

- економічні вигоди: зменшення витрат на енергоресурси призводить до значних економічних вигід для підприємств. Гроші, які раніше витрачалися на

платежі за енергію та інші ресурси, можуть бути використані для інших потреб або інвестовані в розвиток компанії;

– екологічна стійкість: ефективне управління енергоспоживанням також сприяє зменшенню викидів CO<sub>2</sub> та інших шкідливих речовин в атмосферу. Це сприяє екологічній стійкості та допомагає компаніям виконувати свої екологічні обов'язки;

– підтримка прийняття рішень: прогнози витрат енергоресурсів надають компаніям цінну інформацію для прийняття стратегічних та оперативних рішень. Вони можуть адаптувати свої плани, забезпечуючи ефективне управління підприємством.

Саме прогнозування витрат енергоресурсів є найважливішою частиною цієї системи. Прогнози дозволяють підприємствам адаптувати свої стратегії та приймати рішення на основі передбачень щодо майбутнього споживання енергії. Це допомагає уникнути надмірних витрат та забезпечує ефективне використання ресурсів.

Тому розробка програмного забезпечення для прогнозування використання енергоресурсів має велике практичне значення для підприємств та організацій, оскільки вона допомагає підвищити ефективність управління енергоспоживанням та забезпечує економічну та екологічну стійкість.

**Метою кваліфікаційної роботи** є розробка програмного забезпечення, призначеного для прогнозування використання енергоресурсів компаній.

Основні завдання дослідження включають:

- аналіз поточних вимог та потреб компаній щодо енергоресурсів;
- вивчення існуючих методів та програмного забезпечення для прогнозування споживання енергії;
- розробка програмного модуля для прогнозування використання енергоресурсів;
- проведення експериментів та аналіз отриманих результатів;
- визначення практичної значущості розробленого програмного рішення та його можливої апробації в реальних умовах компаній.



У дослідженні використовуються наступні основні терміни та поняття:

– енергоресурси - електроенергія, природний газ, тепло, вода та інші види енергії та ресурсів, які використовуються в промисловості та господарстві;

– прогнозування - процес передбачення майбутніх витрат енергоресурсів на основі наявних даних та аналізу;

– програмне забезпечення - набір програмних модулів, призначених для автоматизації процесу прогнозування використання енергоресурсів;

– ефективне використання - забезпечення оптимального споживання енергії з мінімізацією витрат та негативного впливу на навколишнє середовище.

**Об'єктом дослідження** є діяльність компаній та організацій у сфері споживання енергоресурсів.

**Предметом дослідження** є розробка та впровадження програмного забезпечення для прогнозування використання енергоресурсів компаній.

Дана кваліфікаційна робота включає такі елементи наукової новизни:

– розробка програмного рішення для прогнозування використання енергоресурсів, яке враховує індивідуальні потреби компаній.

– аналіз і порівняння існуючих методів прогнозування споживання енергії та їх відповідність сучасним потребам компаній.

**Практична значущість роботи:** результати даної кваліфікаційної роботи можуть бути використані компаніями та організаціями для оптимізації споживання енергоресурсів, зменшення витрат та покращення енергоефективності. Програмне забезпечення, розроблене в рамках дослідження, може бути впроваджене в практичну діяльність підприємств для покращення управління енергоресурсами.

**Апробація результатів роботи:** результати дослідження можуть бути апробовані та впроваджені в реальних умовах різних компаній та організацій, що споживають енергоресурси.

# **РОЗДІЛ 1. АНАЛІЗ ПОТРЕБ ТА ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПРОГНОЗУВАННЯ ВИКОРИСТАННЯ ЕНЕРГОРЕСУРСІВ КОМПАНІЇ**

## **1.1 Аналіз потреб та вимог компаній при прогнозуванні використання енергоресурсів**

Споживання енергоресурсів є необхідною частиною функціонування будь-якої компанії. Перш за все, необхідно зрозуміти, які саме види енергоресурсів споживаються та в якій кількості. При цьому важливо враховувати різноманітні аспекти:

– види енергоресурсів: основні види енергоресурсів, такі як електроенергія, природний газ, тепло, вода, можуть споживатися в компанії. Важливо визначити, які саме ресурси є важливими для діяльності компанії.

– споживання за різними періодами: споживання енергоресурсів може коливатися в різні періоди. Наприклад, в зимовий період може бути збільшене споживання тепла, а влітку – електроенергії для кондиціонування повітря. Важливо вивчити ці сезонні коливання та їх вплив на діяльність компанії.

– процеси та обладнання: різні процеси та обладнання в компанії можуть споживати енергію по-різному. Наприклад, виробництво може вимагати більше електроенергії порівняно з адміністративними приміщеннями. Важливо встановити, які конкретні процеси і обладнання є найбільш енергоємними.

Для ефективного прогнозування та управління споживанням енергоресурсів необхідно визначити ключові показники ефективності, які допоможуть в оцінці результатів і вирішенні наступних завдань:

– споживання енергоресурсів на одиницю продукції: цей показник визначає, скільки енергії витрачається на виробництво одиниці продукції чи послуги. Велике споживання енергії на одиницю продукції може свідчити про неефективність виробничих процесів.

– витрати на енергоресурси: моніторинг та аналіз витрат на енергоресурси є важливим етапом для зменшення витрат та оптимізації бюджету компанії.

Аналіз потреб та вимог компаній при прогнозуванні використання енергоресурсів є критичним етапом розробки програмного забезпечення для оптимізації цього процесу. Ось більш детальний аналіз:

– історичні дані: першим етапом є збір та аналіз історичних даних щодо використання енергоресурсів. Це може включати в себе дані про споживання електроенергії, газу, води та інших ресурсів. Аналіз цих даних допомагає виявити сезонність, тенденції та аномалії в споживанні.

– специфікація об'єктів: кожна компанія може мати свої унікальні характеристики та об'єкти, які використовують енергоресурси. Це можуть бути виробничі площадки, офісні будівлі, склади тощо. Важливо визначити, які саме ресурси вимірюються на кожному об'єкті та які параметри впливають на їх споживання.

– оцінка поточних витрат: після аналізу історичних даних та характеристик об'єктів, компанії можуть розрахувати поточні витрати на енергоресурси. Це дозволяє їм мати уявлення про їх поточну ситуацію та визначити можливі області для оптимізації.

– прогнозування споживання: наступним етапом є розробка моделей прогнозування, які враховують різні фактори, що впливають на споживання енергоресурсів. Це може включати в себе фактори, такі як погода, виробничий графік, кількість працівників тощо.

– планування та оптимізація: на основі прогнозів компанії можуть розробити плани для оптимізації використання енергоресурсів. Це може включати в себе регулювання споживання в залежності від прогнозів, вибір більш ефективних систем та устаткування, а також розробку стратегій зменшення витрат.

– моніторинг та контроль: після впровадження програмного забезпечення компанії можуть здійснювати постійний моніторинг та контроль за витратами енергоресурсів, порівнюючи їх з прогнозами та планами. Це дозволяє своєчасно виявляти аномалії та реагувати на них.

– аналіз ефективності: компанії можуть проводити аналіз ефективності впроваджених заходів з оптимізації для оцінки їх впливу на зниження витрат та покращення сталості споживання енергоресурсів.

Загальна мета програмного забезпечення полягає в тому, щоб допомогти компаніям ефективно управляти та оптимізувати витрати енергоресурсів, забезпечуючи тим самим економічну вигоду та сприяючи збереженню природних ресурсів.

## **1.2 Вимоги до прогнозування енергоспоживання підприємства**

В наукових роботах розглядається велика кількість методів прогнозування енерговитрат.

Методи прогнозування енергоспоживання можна класифікувати за різними критеріями. Ось декілька основних способів класифікації методів прогнозування енергоспоживання:

### **1. За типом даних:**

– кількісні методи: використовуються для прогнозування числових значень, таких як кількість спожитої енергії.

– якісні методи: використовуються для прогнозування категорійних даних, наприклад, класифікації споживачів за типами.

### **2. За характером часового ряду:**

– статичні методи: використовуються, коли дані вважаються сталими в часі і не враховують зміни.

– динамічні методи: враховують залежність між даними в різні моменти часу та зміни в часі.

### **3. За методами аналізу:**

– класичні методи: включають у себе методи, які використовують статистичні та математичні моделі для прогнозування, такі як метод середнього, експоненційного згладжування, регресійний аналіз.

– машинне навчання: використовуються алгоритми машинного навчання для прогнозування на основі аналізу великої кількості даних, такі як нейронні мережі, дерева рішень, методи кластеризації, тощо.

### **4. За тривалістю прогнозування:**

– короткострокове прогнозування: прогнози на найближчий період (години, дні, тижні).

– середньострокове прогнозування: прогнози на середні терміни (місяці, квартали).

– довгострокове прогнозування: прогнози на великий період (роки, десятиліття).

#### 5. За джерелом даних:

– історичні дані: використовуються попередні дані про споживання енергії для прогнозування майбутніх значень.

– сенсорні дані: використовуються дані, зібрані з сенсорів та моніторингових систем.

#### 6. За обсягом даних:

– одиночні методи: використовуються для прогнозування для окремого споживача або об'єкта.

– масштабні методи: використовуються для прогнозування на рівні міста, регіону, країни тощо.

#### 7. За специфікою завдання:

– економічні методи: використовуються для прогнозування споживання енергії для економічних аналізів і планування бюджету.

– екологічні методи: використовуються для оцінки впливу споживання енергії на навколишнє середовище.

Класифікація методів прогнозування може бути поділена на фактографічні, гібридні та експертні методи, що досить широко використовується у сфері прогнозування (рис. 1.1).

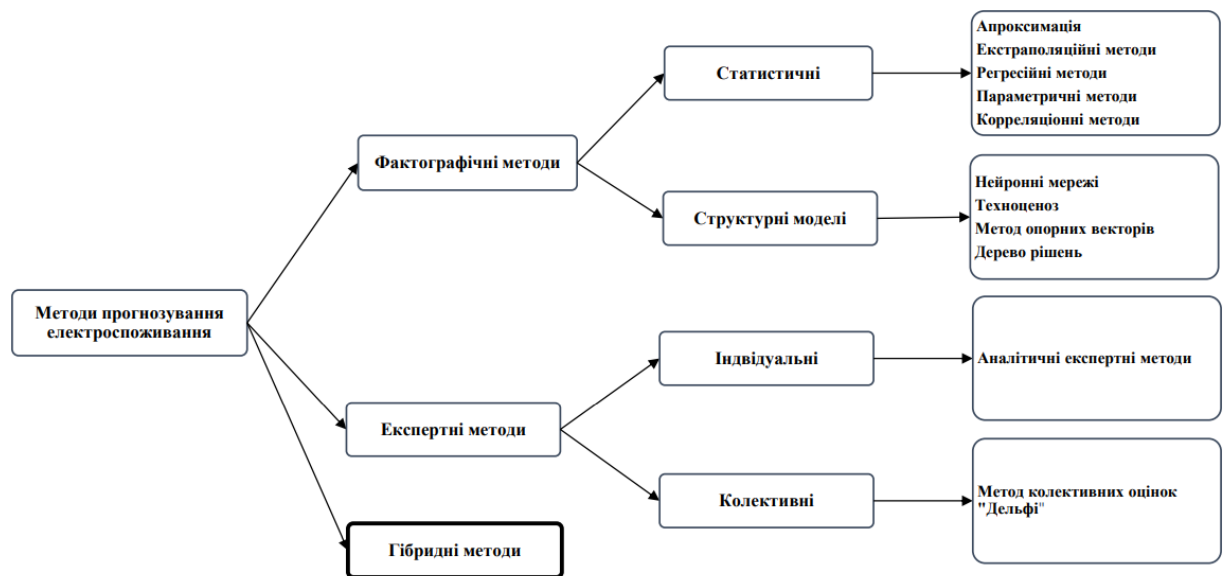


Рис. 1.1. Класифікація методів прогнозування енергоспоживання

Розглянемо кожну з цих груп більш детально:

1. Фактографічні методи (квалітативні методи): ці методи базуються на аналізі історичних даних та фактів. Вони використовуються, коли існують досить повні та достовірні дані для прогнозування. До цієї групи відносяться такі методи:

- метод середнього значення: простий метод, де прогноз робиться на основі середнього значення історичних даних. Найпростіший варіант – це метод середньої арифметичної;

- метод експоненціального згладжування: використовується для моделювання трендів та сезонних коливань в даних. Включає в себе методи згладжування Ельфа та Гольдвінтера-Вінтерса;

- метод індексу базового року: використовується для порівняння змін в показниках відносно базового року;

- метод аналогії: передбачає прогноз на основі аналогії з історичними даними.

2. Гібридні методи: гібридні методи поєднують в собі різні методи прогнозування для покращення точності та надійності прогнозів. Ці методи

використовують як фактографічні, так і кількісні або якісні підходи. До гібридних методів можна віднести такі:

– машинне навчання та аналіз даних: використовує методи навчання з учителем (наприклад, лінійна регресія, дерева рішень, нейронні мережі) для аналізу та прогнозування;

– методи часових рядів: використовуються для аналізу та прогнозування даних, які мають часову залежність. Включають методи ARIMA (авторегресія зі змінною ковзаючою середньою), Exponential Smoothing (експоненціальне згладжування) тощо;

– гібриди різних методів: можуть використовувати комбінації методів для досягнення більш точних результатів.

3. Експертні методи: ці методи використовують експертні знання та думки для прогнозування. Вони можуть бути корисними там, де історичних даних недостатньо або коли ситуація дуже специфічна. До експертних методів відносяться:

– експертні оцінки: експерти надають свої прогнози на основі свого досвіду та знань;

– метод Делфі: включає в себе ітеративний процес зібрання прогнозів від групи експертів з подальшим аналізом і агрегацією результатів.

– сценарний аналіз: експерти розробляють різні сценарії розвитку подій та відповідні прогнози для кожного сценарію.

Обираючи метод прогнозування, важливо враховувати особливості конкретної задачі та доступні дані. Також часто використовують комбінацію різних методів для отримання більш точних та надійних прогнозів.

### **1.3 Огляд існуючих підходів до використання програмного забезпечення при прогнозуванні використання енергоресурсів компанії**

Для ефективного прогнозування використання енергоресурсів існують різні методи та підходи, які можуть бути використані. Декілька з найпоширеніших методів включають:

– часові ряди: моделі на основі часових рядів використовують історичні дані споживання енергії для прогнозування майбутніх значень. Вони можуть

бути ефективними для виявлення сезонних та циклічних змін у споживанні енергії.

– методи машинного навчання: використання алгоритмів машинного навчання, таких як нейронні мережі, дерева рішень і глибоке навчання, дозволяє створити складні моделі, які можуть адаптуватися до різноманітних факторів, що впливають на споживання енергії (рис. 1.2).

– аналітика даних та статистичні методи: Використання аналітики даних та статистичних методів дозволяє виявляти зв'язки та залежності між різними факторами та споживанням енергії.

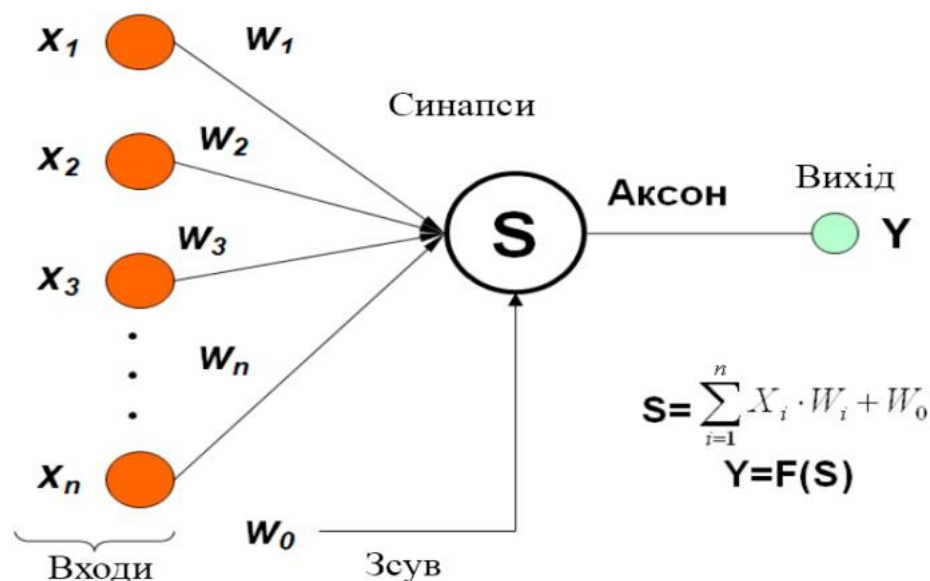


Рис. 1.2. Принцип використання нейронних мереж в прогнозуванні

Аналіз існуючих методів прогнозування використання енергоресурсів є ключовим етапом в розробці програмного рішення для прогнозування енергоефективності компанії. Цей аналіз допомагає визначити, які методи вже існують у літературі та практиці, які їх переваги та недоліки та які з них можна використовувати для досягнення поставлених завдань.

Аналіз існуючих методів прогнозування:

– методи часових рядів передбачають аналіз історичних даних про споживання енергії в різні моменти часу. Методи ARIMA (авторегресія з інтегрованими ковзними середніми) та експоненціальне згладжування є популярними серед методів часових рядів для прогнозування споживання



енергоресурсів. Перевагою цих методів є їх здатність враховувати сезонність та тенденції у даних. Однак вони можуть бути обмеженими в тому випадку, коли вплив зовнішніх факторів на споживання не враховується.

– методи машинного навчання: регресія, випадковий ліс, градієнтний бустінг та нейронні мережі, стали дуже популярними для завдань прогнозування. Вони можуть використовувати різноманітні фактори для прогнозування споживання енергії та дозволяють створювати складні моделі з високою точністю. Однак вони можуть вимагати великої кількості даних та обчислювальних ресурсів.

– аналіз дисперсії використовується для вивчення впливу різних факторів на споживання енергії та визначення їх значущості. Він може допомогти в ідентифікації ключових факторів, які впливають на енергоефективність.

– експертні системи можуть використовувати знання експертів у галузі енергетики для прогнозування споживання енергії. Вони можуть бути корисними в ситуаціях, де доступ до великої кількості даних обмежений, але наявні експерти, які можуть надати додаткові знання.

Переваги та недоліки існуючих методів:

– методи часових рядів мають високу точність для даних з вираженою сезонністю, але можуть бути обмеженими для даних з багатомірною залежністю.

– методи машинного навчання можуть створювати дуже точні моделі, але вимагають велику кількість даних та обчислювальних ресурсів.

– аналіз дисперсії допомагає в ідентифікації факторів, що впливають на споживання енергії, але не завжди може прогнозувати це споживання.

– експертні системи використовують знання експертів, але можуть бути обмеженими в обсязі знань та не завжди враховувати складні залежності.

Аналіз існуючих методів прогнозування використання енергоресурсів дозволяє вибрати найбільш підходящі методи для розробки програмного рішення та враховувати їх переваги та недоліки при розробці:

– часові ряди: цей метод добре підходить для прогнозування сезонних змін та трендів. Однак він може бути менш ефективним, коли є значні випадкові фактори, що впливають на споживання енергії.

– методи машинного навчання: можуть враховувати більше факторів і виявляти складні залежності, але для їх застосування потрібно багато даних і обчислювальних ресурсів.

– аналітика даних та статистичні методи: для їх використання важливо мати глибокі знання в області статистики та аналізу даних.

Визначення ключових показників ефективності (KPIs) є критичним етапом в аналізі потреб та вимог замовника для розробки програмного забезпечення для прогнозування використання енергоресурсів компанії. Ці KPIs допомагають виміряти та оцінити різні аспекти споживання енергоресурсів і служать як ключові критерії для визначення ефективності використання цих ресурсів.

Важливо враховувати, що KPIs повинні бути специфічними для конкретної компанії та її бізнес-потреб. Нижче наведені деякі з ключових показників ефективності, які можуть бути використані для аналізу та вимірювання споживання енергоресурсів. Розкриємо більш детально ці показники.

**Енергоефективність:** цей показник визначає, як ефективно компанія використовує енергію для виробництва продукції або надання послуг. Вимірюється в одиницях енергії на одиницю продукції або об'єм послуги. Високий рівень енергоефективності свідчить про раціональне використання ресурсів.

**Витрати на енергоресурси:** цей KPI вказує на загальну вартість споживаних енергоресурсів, включаючи електроенергію, газ, воду тощо. Вимірюється в грошовому еквіваленті та може бути корисним для бюджетування та оптимізації витрат.

**Кошторисна точка ефективності (Break-Even Point):** цей показник вказує на обсяг продукції або послуг, при якому компанія не зазнає збитків, і всі витрати (включаючи витрати на енергоресурси) компенсуються прибутками.

Цей показник може бути важливим для планування виробничих операцій та фінансового аналізу.

Загальні витрати на енергоресурси: цей показник визначає загальну вартість всіх видів споживаних енергоресурсів та може допомогти виявити основні джерела витрат.

Кількість енергії, що заощаджена завдяки енергоефективності: цей показник вказує на обсяг енергії, який було заощаджено завдяки впровадженню енергоефективних заходів та технологій.

Споживання енергії на обсяг споживачів: цей KPI дозволяє визначити, скільки енергії в середньому споживає кожен споживач (наприклад, на одного працівника чи на одиницю обладнання). Ця інформація може бути корисною для ідентифікації точок збитків та резервів ефективності.

Економічний ефект від застосування прогнозування енергоресурсів: Цей KPI дозволяє визначити економічний вигравш, який компанія отримує завдяки використанню програмного забезпечення для прогнозування споживання енергоресурсів.

Визначення цих ключових показників ефективності допомагає не лише виміряти ефективність використання енергоресурсів, але і встановити базові бар'єри для подальших досліджень та розробки програмного рішення для прогнозування використання енергоресурсів компанії. Правильно вибрані KPIs допоможуть визначити стратегію та переваги використання програмного забезпечення в конкретному бізнесі.

#### **1.4 Аналіз існуючого програмного забезпечення аналізу та оптимізації споживання енергоресурсів**

На ринку існує різне програмне забезпечення, призначене для прогнозування використання енергоресурсів. Деякі з цих програм мають широкий функціонал та можуть працювати для різних галузей, в той час як інші можуть бути спеціалізованими для конкретних потреб:

– програми на основі часових рядів: Деякі програми використовують методи аналізу часових рядів для прогнозування споживання енергоресурсів.

Вони можуть бути корисними для компаній, які мають доступ до великої кількості історичних даних;

- програми на основі машинного навчання: Деякі програми використовують алгоритми машинного навчання для створення точних та адаптивних моделей прогнозування. Вони можуть враховувати більше факторів, що впливають на споживання енергії;

- спеціалізовані програми для конкретних галузей: Деякі програми призначені для конкретних галузей, таких як виробництво, транспорт або будівництво. Вони можуть мати спеціалізовані функції та моделі, які враховують особливості галузі.

Аналіз існуючого програмного забезпечення для прогнозування використання енергоресурсів компанії є важливим етапом в розробці кваліфікаційної роботи. Вибір відповідного програмного рішення або системи може значно вплинути на успішність вашого дослідження та ефективність прогнозування.

Розглянемо деякі аспекти такого аналізу:

- функціональність програмного забезпечення: оцінка функціональності існуючого програмного забезпечення.

- відкритий вихідний код та ліцензія: відкритість вихідних кодів або платність, ліцензійні умови та можливості доступу до вихідного коду, оскільки це може вплинути на використання та модифікацію програми.

- сумісність та інтеграція: взаємодія з іншими системами та програмами, які вже використовуються в компанії та можливості інтеграції з іншими інформаційними системами.

- масштабованість та продуктивність: здатність ефективно працювати з великими обсягами даних та чи можливо розширити його функціональність для потреб компанії.

- доступність підтримки та документації: наявність активної спільноти користувачів, яка надає підтримку та допомогу. Доступність документації та навчальних матеріалів також є важливими факторами.

– безпека даних: заходи безпеки, які надає програмне забезпечення для захисту конфіденційності та цілісності даних про споживання енергоресурсів.

– аналіз точності та ефективності: тестування програмного забезпечення на історичних даних, щоб оцінити його точність та ефективність у конкретному випадку.

– специфічні функції та параметри: підтримка специфічних функцій або параметрів, які важливі для компанії.

– вартість та ліцензійні витрати: вартість впровадження та підтримки програмного забезпечення. Порівняйте це з вашим бюджетом та ресурсами.

Після проведення аналізу існуючого програмного забезпечення, зможете визначити, чи підходить воно для потреб компанії або чи потрібно розробити власне програмне рішення. Оцінка переваг та недоліків існуючих програмних засобів допоможе вам вибрати оптимальний шлях для реалізації вашого дослідження та досягнення мети кваліфікаційної роботи.

Наведемо опис трьох програмних продуктів, які використовуються в галузі енергетики для аналізу та оптимізації споживання енергоресурсів:

1. EnergyPlus – це безкоштовне програмне забезпечення для моделювання та аналізу енергоефективності будівель та систем опалення, вентиляції та кондиціонування повітря. Воно розроблено та підтримується американським Департаментом енергетики (DOE) і використовується для розрахунків енерговитрат та теплового балансу будівель.

Функції: моделювання теплового та електричного споживання енергії в будівлях, врахування впливу клімату, конструкцій та систем опалення та кондиціонування повітря на енергоефективність, проведення аналізу різних сценаріїв для вдосконалення дизайну будівель та систем.

EnergyPlus використовується (рис. 1.3) архітекторами, інженерами та дослідниками для оптимізації проектів будівель, зменшення енерговитрат і викидів CO<sub>2</sub>.

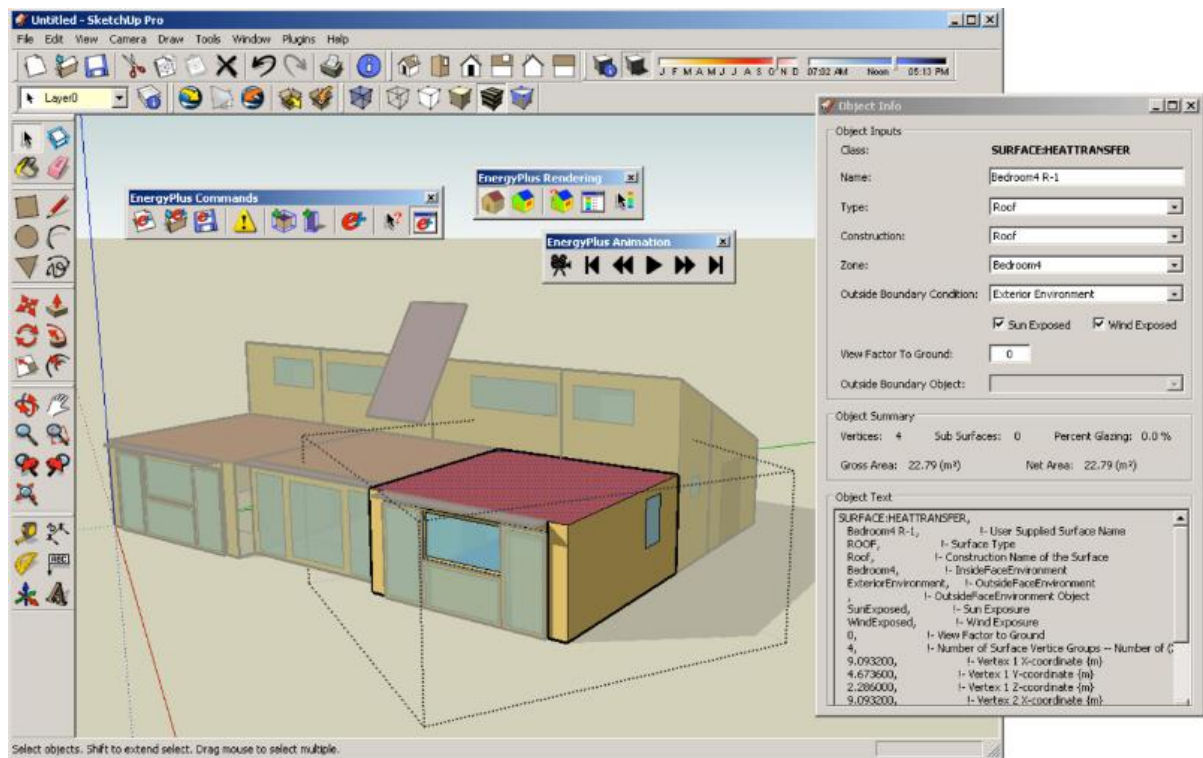


Рис. 1.3. Приклад вікна системи EnergyPlus

2. RETScreen – це безкоштовний програмний інструмент для аналізу енергетичної ефективності та економічної доцільності проектів у сфері відновлюваної енергетики та енергоефективності. Він розроблений Канадським центром енергоефективності та використовується для оцінки проектів з відновлюваної енергетики.

Функції: аналіз відновлюваних джерел енергії, таких як сонячна, вітрова, гідро- та біомасова енергія, розрахунок витрат та викидів парникових газів, прогнозування економічної доцільності та оцінка впливу на навколишнє середовище.

RETScreen використовується (рис. 1.4) приймачами рішень, проектними менеджерами та аналітиками для оцінки та планування проектів з використання відновлюваних джерел енергії.

3. EnergyCAP – це програмне забезпечення для управління енергетичними ресурсами та обліку витрат енергії. Воно допомагає організаціям ефективно відстежувати, аналізувати та контролювати свої витрати на енергоресурси.

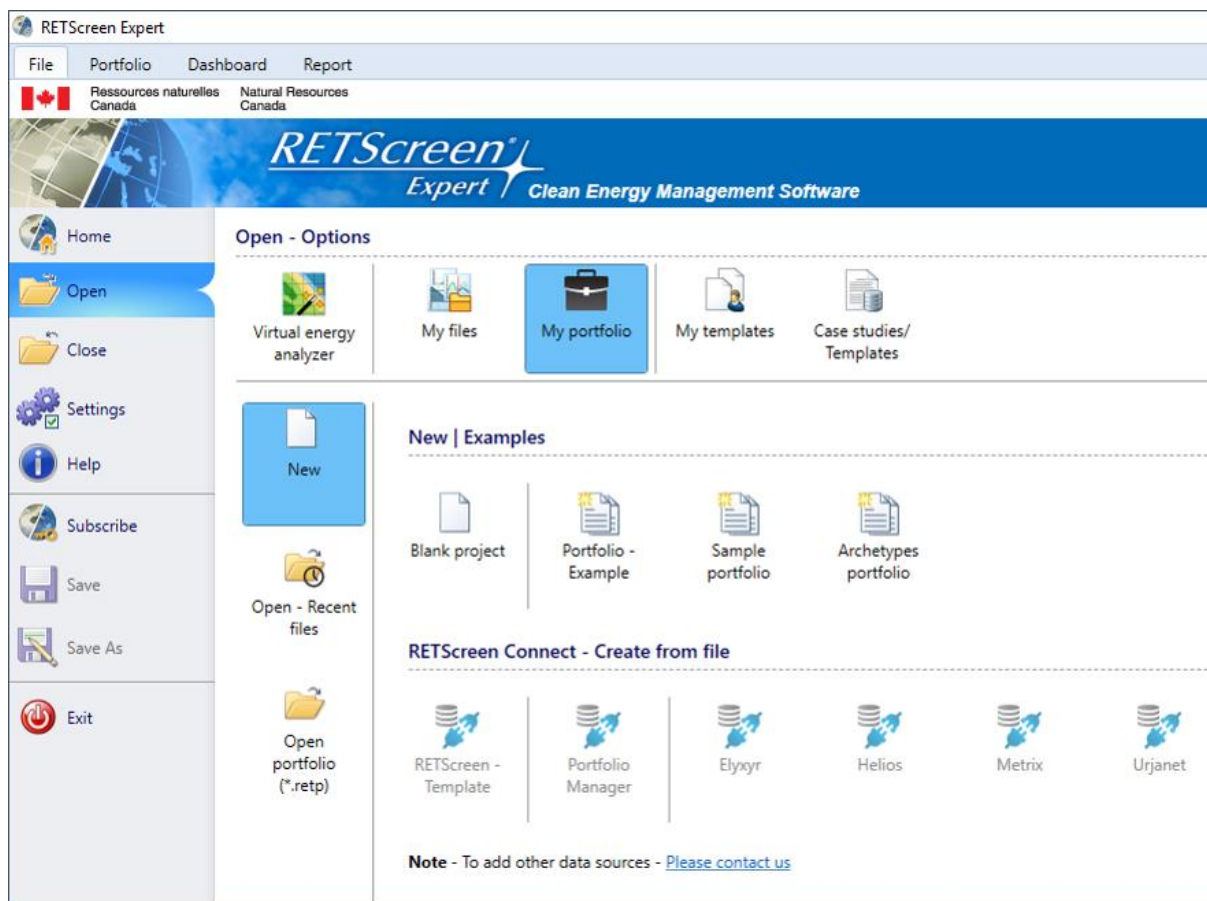


Рис. 1.4. Приклад вікна системи RETScreen

Функції: збір, обробка та аналіз даних про споживання енергії з різних джерел, виявлення та вирішення проблем з ефективністю витрат енергії, звітність та аналітика для приймальників рішень та регуляторів.

EnergyCAP (рис. 1.5) використовується комерційними компаніями, установами, урядовими органами та іншими організаціями для оптимізації управління енергетичними ресурсами та зменшення витрат.

Ці програмні продукти мають різні застосування та функціональність, і вони можуть бути корисними для організацій, які прагнуть знизити свої витрати на енергоресурси та зробити їхнє споживання більш ефективним.

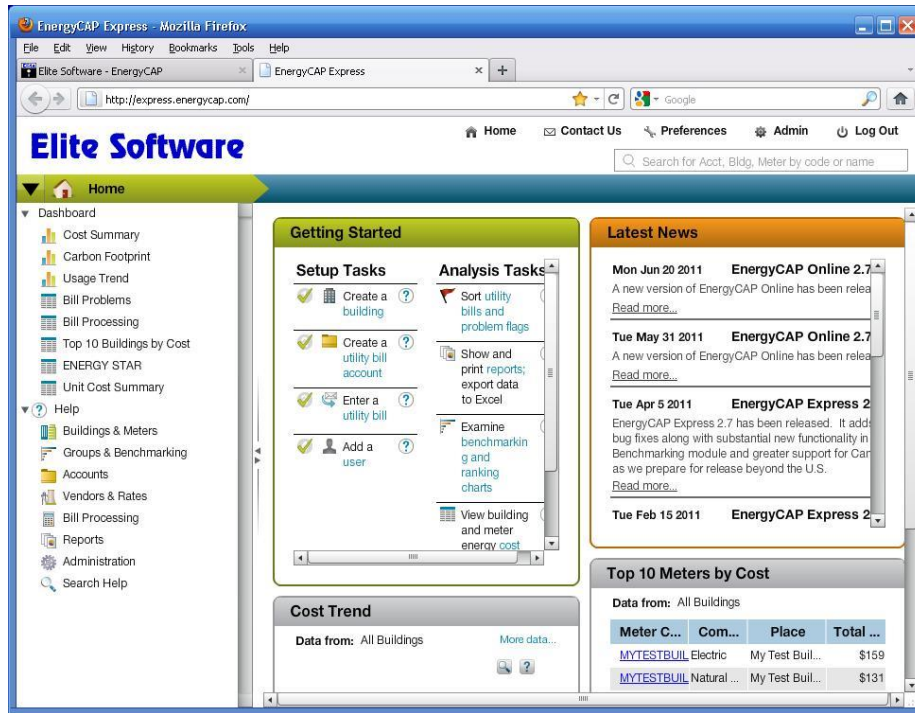


Рис. 1.5. Приклад вікна системи EnergyCAP

## Висновки до розділу 1

Проведено аналіз потреб та вимог компаній щодо прогнозування використання енергоресурсів. Даний аналіз дозволив визначити ключові аспекти та вимоги, які ставлять перед програмним забезпеченням для прогнозування енергоспоживання та підтвердити гіпотезу, що розробка програмного забезпечення для прогнозування використання енергоресурсів є актуальною задачею, яка може вирішити потреби компаній у точних та надійних прогнозах.

Виявлено, що компанії потребують точних та надійних прогнозів для планування витрат та оптимізації енергоспоживання. Вимоги включають в себе можливість аналізу різних типів енергоресурсів, зручний інтерфейс для користувачів, можливість враховувати різні фактори та здатність генерувати точні та надійні прогнози. Також виявлено, що існуючі рішення мають свої обмеження та недоліки, і для задоволення потреб компаній може бути необхідно розробити нове програмне забезпечення.



## **РОЗДІЛ 2. ЕТАПИ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ПРОГНОЗУВАННЯ ВИКОРИСТАННЯ ЕНЕРГОРЕСУРСІВ КОМПАНІЇ**

### **2.1 Вибір вхідних даних та методів прогнозування**

Перед тим як обрати методи прогнозування, необхідно визначити, які типи даних будуть використовуватися в процесі прогнозування використання енергоресурсів. Основні типи даних, які можуть бути використані, включають в себе:

– часові ряди: історичні дані про споживання енергії зазвичай представлені у вигляді часових рядів. Вони містять дані про споживання енергії в різний час, і ці дані можуть бути використані для аналізу та прогнозування майбутнього споживання.

– економічні дані: Дані про економічний стан компанії, галузі або ринку також можуть бути важливими факторами для прогнозування. Наприклад, економічний зріст чи спад може впливати на обсяг споживання енергії.

– кліматичні дані: Кліматичні умови, такі як температура, вологість, опади і т. д., можуть мати великий вплив на споживання енергії. Тому дані про клімат можуть бути включені в моделі для покращення точності прогнозування.

Після визначення типів даних важливо обрати відповідні методи аналізу та моделювання для прогнозування споживання енергоресурсів. Деякі з популярних методів включають:

– часові ряди: для аналізу часових рядів можна використовувати методи, такі як експоненціальне згладжування, авторегресія (AR), ковзне середнє (MA), або авторегресія з ковзним середнім (ARMA). Для прогнозування можна також використовувати методи ARIMA (авторегресія з інтегрованими ковзними середніми) або моделі Гольдфельда-Квандта.

– методи машинного навчання: методи машинного навчання, такі як регресія, випадковий ліс, градієнтний бустінг та нейронні мережі, дозволяють створювати складні моделі, які можуть адаптуватися до різних типів даних та залежностей.

– методи статистичного аналізу: методи, такі як аналіз часових рядів, кореляційний аналіз та аналіз дисперсії, можуть бути використані для виявлення зв'язків та залежностей між різними змінними.

Для прогнозування використання ресурсів з коливаннями від дня до дня, вам можуть допомогти різні методи аналізу часових рядів і статистики.

Згладжування часових рядів: метод згладжування може включати в себе застосування ковзного середнього (Moving Average) або методу згладжування Хольта-Вінтерса. Вони допоможуть вам створити гладкий тренд з часового ряду і зменшити вплив випадкових коливань.

Авторегресійні моделі (AR): цей метод використовує інформацію з попередніх точок даних для прогнозування майбутніх значень. Модель AR враховує кореляцію між значеннями в різні моменти часу.

Комбіновані моделі (ARIMA): ARIMA (Autoregressive Integrated Moving Average) – об'єднує авторегресійні та згладжування часових рядів разом з інтегруванням (перетворенням ряду в стаціонарний).

Експоненційне згладжування (Exponential Smoothing): цей метод підходить для даних з трендом і сезонністю. він використовує згладжування експоненційним способом для оцінки значень.

Нейронні мережі (Neural Networks): глибоке навчання і нейронні мережі можуть бути використані для складних часових рядів, де є багато неочікуваних змін.

Методи регресії: якщо є фактори, які можуть впливати на коливання використання ресурсів (наприклад, день тижня, свята тощо), то можна використовувати методи регресії для моделювання цих впливів.

При виборі методу важливо враховувати особливості вашого часового ряду, доступні дані та вимоги точності прогнозу. Рекомендується також регулярно перевіряти та оновлювати модель, оскільки вона може змінюватися з часом через зміни у споживанні ресурсів.

Методи регресії – це клас статистичних методів, які використовуються для аналізу та моделювання залежності між однією або більше незалежними (пояснюючими) змінними і однією залежною (відгуком) змінною. У контексті

прогнозування використання ресурсів компанії методи регресії можуть бути корисними для врахування впливу різних факторів на споживання енергоресурсів.

Лінійна регресія: найпоширеніший тип регресії, де залежність між змінними моделюється як лінійна функція. У простій лінійній регресії є одна незалежна змінна, яка впливає на залежну змінну. У множинній лінійній регресії може бути більше однієї незалежної змінної.

Логістична регресія: використовується, коли залежна змінна є категоріальною (бінарною або дискретною). Вона дозволяє прогнозувати ймовірність належності до певного класу.

Поліноміальна регресія: створюються поліноми вищих ступенів для незалежних змінних. Це дозволяє моделювати нелінійні залежності між змінними.

Регресія з деревами рішень: використовується для створення дерева рішень, яке може бути використано для прогнозування залежної змінної на основі визначення критеріїв прийняття рішення на кожному вузлі дерева.

Регресія на основі машинного навчання: методи машинного навчання, такі як Support Vector Machines (SVM), Random Forests та Gradient Boosting, можуть бути використані для регресійного аналізу. Вони дозволяють моделювати складні нелінійні залежності та враховувати багато факторів.

Регресія з часовими рядами: для прогнозування використання ресурсів з коливаннями від дня до дня може бути використана регресія з часовими рядами, де змінним додаються попередні значення залежної змінної в якості незалежних змінних.

Регресія з глибоким навчанням: глибокі нейронні мережі можуть бути використані для моделювання складних залежностей в даних. Із завданням великої кількості даних цей метод може бути дуже потужним для прогнозування.

## **2.2 Збір та підготовка даних**

Перед тим як розпочати розробку програмного рішення, необхідно зібрати та підготувати дані про споживання енергоресурсів.

Цей етап включає в себе:

- збір даних: завантаження історичних даних про споживання енергії, включаючи дані про час, обсяг та, можливо, інші фактори, що впливають на споживання.

- очищення даних: видалення аномалій, відсутніх значень та інших неправильностей в даних, які можуть впливати на точність прогнозування.

- візуалізація даних: використання графіків та візуалізації для вивчення паттернів та залежностей у даних.

Після очищення та аналізу даних, необхідно підготувати їх для навчання та тестування моделі прогнозування. Це включає в себе:

- розділення даних: розділення історичних даних на навчальний та тестовий набори. Навчальний набір використовується для навчання моделі, тестовий – для перевірки її ефективності;

- масштабування даних: масштабування даних до визначеного діапазону значень, що допомагає моделі швидше навчатися та покращує її стійкість.

Існують різні нормативи і стандарти, які встановлюють граничні витрати електроенергії на один квадратний метр ( $W/m^2$  або  $kWh/m^2$ ) для різних типів будівель та застосувань. Ці нормативи можуть відрізнятися в залежності від регіону, законодавства і галузевих вимог.

Наприклад, Leadership in Energy and Environmental Design (LEED) – це система сертифікації будівель, яка встановлює стандарти енергоефективності для зелених будівель. Для отримання LEED-сертифікації, будівля повинна відповідати певним вимогам щодо енергоефективності, включаючи граничні значення витрат енергії на квадратний метр.

Leadership in Energy and Environmental Design (LEED) – це світово визнана система сертифікації для оцінки сталості та екологічності будівель і споруд. LEED розроблена та управляється Грін Білдінг Консіл (Green Building Council, USGBC) в Сполучених Штатах. Ця система спрямована на стимулювання створення здорових, ефективних і сталої експлуатації будівель та споруд.

Основні принципи та складові системи LEED включають:

1) категорії рейтингу: LEED розділена на категорії, які відображають різні аспекти будівництва: LEED для нових будівель і реконструкції (LEED BD+C): оцінюється будівлі та споруди під час будівництва або реконструкції, LEED для операції та обслуговування будівель (LEED O+M): оцінюється робота та обслуговування існуючих будівель, LEED для інтер'єрів будівель (LEED ID+C): оцінюється дизайн та внутрішній облаштування приміщень, LEED для житлових будівель (LEED Homes): оцінюється житлові будівлі, такі як дачі, квартири і домівки;

2) рейтингові бали: проектам присвоюються бали за виконання певних критеріїв в межах кожної категорії. Щоб отримати сертифікат LEED, будівлі повинні набрати певну кількість балів, яка відповідає рівню сертифікації (Certified, Silver, Gold або Platinum).

3) критерії оцінки: кожна категорія LEED має свої критерії оцінки, які включають аспекти: енергоефективність: спрямована на зменшення споживання енергії та використання відновлювальних джерел енергії, якість повітря в будівлі: забезпечення чистого та здорового повітря для мешканців та користувачів будівлі;

4) матеріали та ресурси: зменшення викидів та використання стійких до обробки матеріалів;

5) якість води та водоспоживання: ефективне використання води та забезпечення якості питної води;

6) дизайн та інновації: заохочення інновацій у будівництві та дизайні;

7) сертифікаційні рівні: залежно від кількості набраних балів, будівлі можуть отримати один із чотирьох сертифікаційних рівнів: Certified, Silver, Gold або Platinum. Кожен рівень відображає ступінь сталості та екологічності будівлі.

Головна мета LEED – це стимулювати розробників, будівельників і власників будівель до створення сталого та екологічно відповідального середовища. LEED сприяє зменшенню споживання ресурсів, зниженню викидів, покращенню якості повітря та здоров'я користувачів будівель.

Поширення стандарту LEED свідчить про зростання свідомості щодо екологічних проблем та сталого розвитку, що є ключовими завданнями у сучасному світі будівництва та архітектури.

Для розрахунку оптимальних енергетичних витрат за стандартом LEED (Leadership in Energy and Environmental Design), необхідно дотримуватися ряду критеріїв та виконувати деякі заходи. LEED має специфічні критерії та рекомендації для досягнення оптимальної енергоефективності в будівлях та спорудах:

- дизайн та впровадження систем енергозбереження: для досягнення оптимальних енергетичних витрат, необхідно включити в дизайн будівлі системи енергозбереження, такі як ізоляція, ефективні вікна та двері, добре утеплені стіни та дах, а також сучасні системи опалення, вентиляції і кондиціонування повітря. Важливо також правильно налаштувати ці системи для максимальної ефективності.

- використання відновлювальних джерел енергії: залучення відновлювальних джерел енергії, таких як сонячні панелі, вітрові турбіни або гідроелектростанції, може допомогти зменшити споживання енергії з традиційних джерел. LEED надає бали за використання відновлювальних джерел енергії.

- ефективне освітлення: Використовуйте ефективні системи освітлення, такі як LED лампи та датчики руху, для зменшення витрати енергії на освітлення в приміщеннях.

- зниження споживання води: Водопостачання і водовідведення також впливають на енергетичні витрати будівлі. Зменшення витрати води, встановлення ефективних систем фільтрації та використання відновлюваних джерел води можуть зменшити енергетичні витрати.

- оптимізація конструкцій та дизайну: Планування і дизайн будівлі також можуть впливати на її енергоефективність. Оптимізуйте розташування вікон, дверей і інших структур, щоб максимізувати природне освітлення та тепло.

– використання систем управління будівлею: впровадження сучасних систем управління будівлею (Building Management Systems, BMS) дозволяє ефективно керувати споживанням енергії в реальному часі.

– перевірка та аналіз даних: Після побудови будівлі важливо здійснювати регулярний моніторинг та аналіз її енергетичної ефективності, виявляти потенційні проблеми та вчасно їх усувати.

Загальний підхід до досягнення оптимальних енергетичних витрат за LEED полягає в інтеграції енергоефективних рішень у всі аспекти будівництва, починаючи від проектування і закінчуючи експлуатацією та обслуговуванням. Багато з цих заходів можуть бути підтримані і впроваджені за допомогою спеціальних програмних засобів для аналізу та оптимізації енергоспоживання.

Існують програмні засоби, які можуть допомогти в оцінці будівель за системою LEED (Leadership in Energy and Environmental Design) та сприяти отриманню сертифікатів LEED:

– LEED Online: Це офіційний онлайн-інструмент для реєстрації та подання проектів для оцінки LEED. Він також надає інструменти для введення даних про проект, визначення балів за критеріями LEED і отримання сертифіката LEED.

– AutoCASE for Sites: Це програмне забезпечення для розрахунку економічних та екологічних вигод, які можна отримати при виконанні проектів з оцінки LEED. Воно допомагає приймати рішення, що стосуються сталості та ефективності проектів.

– Green Building Studio: Це програмне забезпечення, розроблене компанією Autodesk, яке використовується для моделювання та аналізу енергоефективності будівель. Воно допомагає визначити, як можна покращити енергоефективність будівлі з урахуванням стандартів LEED.

– Green Globes: Це альтернативна система сертифікації, яка базується на принципах сталого будівництва, схожих на LEED. Існує програмне забезпечення для оцінки будівель за критеріями Green Globes.

– EnergyPlus: Це програмне забезпечення для моделювання енергоспоживання будівель та аналізу їх ефективності. Воно може

використовуватися для розрахунків, пов'язаних з енергетичними аспектами LEED.

Ці програмні засоби можуть допомогти інженерам, архітекторам та іншим фахівцям у галузі будівництва та дизайну визначити, як вони можуть досягнути цілей сталості та енергоефективності у своїх проєктах та отримати сертифікати LEED.

Для промислових будівель і об'єктів існує система сертифікації та стандарти, схожі на LEED (Leadership in Energy and Environmental Design), але призначені специфічно для цієї галузі. Один з найвідоміших стандартів для промислових будівель і об'єктів – це ISO 14001.

ISO 14001 – це міжнародний стандарт для системи управління довкіллям. Він не обмежується тільки промисловими будівлями, але також застосовується до всіх видів організацій та об'єктів, які прагнуть зменшити свій негативний вплив на навколишнє середовище. ISO 14001 встановлює вимоги до системи управління довкіллям, що допомагає контролювати та покращувати середовищне вплив діяльності організації.

Поза ISO 14001 існують інші стандарти та програми, призначені для промислових будівель і об'єктів. Наприклад, Leadership in Energy and Environmental Design (LEED) for Industrial Facilities – це розширення стандарту LEED, призначене спеціально для промислових об'єктів.

Також існують різні національні стандарти та ініціативи, спрямовані на покращення енергоефективності та сталості в промисловому будівництві.

### **2.3 Обґрунтування архітектури програмного забезпечення прогнозування використання енергоресурсів компанії**

Обґрунтування архітектури програмного забезпечення для прогнозування використання енергоресурсів компанії є критичним етапом у розробці системи. Ураховуючи різні підходи до моделювання програмного забезпечення, необхідно обґрунтувати обрану архітектуру, щоб вона відповідала потребам користувачів та була ефективною у вирішенні завдань.

– багатофункціональність: основним вимогам до програмного забезпечення є здатність прогнозувати використання різних енергоресурсів



(електроенергія, газ, вода), а також враховувати різні фактори, які впливають на їх споживання. Ось чому важливо обрати архітектуру, яка підтримує багатофункціональність та можливість розширення.

– модульність і розширюваність: обрана архітектура повинна бути модульною, щоб розділити різні функціональні блоки програми на окремі модулі. Це дозволить легко розширювати та модифікувати систему в майбутньому, коли з'являться нові вимоги або функції.

– взаємодія з даними: проведення прогнозу вимагає роботи з великою кількістю даних, таких як вимірювання лічильників, історичні дані, параметри об'єктів тощо. Отже, важливо обрати архітектуру, яка дозволяє ефективно зберігати, обробляти і аналізувати дані.

– зручний інтерфейс користувача: передбачається, що програмне забезпечення буде використовуватися персоналом компанії, тому важливо мати зручний інтерфейс користувача для введення та візуалізації даних, а також отримання прогнозів.

– алгоритми прогнозування: оскільки прогнозування є ключовою функцією системи, важливо мати можливість використовувати різні алгоритми прогнозування, включаючи статистичні методи, машинне навчання, експертні системи тощо.

– безпека і конфіденційність: оскільки дані щодо витрат енергоресурсів можуть бути конфіденційними, обрана архітектура повинна забезпечувати високий рівень безпеки та конфіденційності даних.

– швидкодія і масштабованість: система повинна працювати ефективно, навіть при обробці великої кількості даних. Також важливо, щоб система була масштабованою і могла обробляти зростаючий обсяг даних та користувачів.

З урахуванням вимог та потреб користувачів можна обрати архітектуру програмного забезпечення, яка базується на модульному підході, з використанням сучасних технологій для забезпечення швидкодії та безпеки. Програмне забезпечення має розширену функціональність для прогнозування використання різних енергоресурсів та підтримує можливість розширення функціоналу у майбутньому.

Перед розробкою програмного модуля важливо ретельно спланувати його архітектуру. Архітектура повинна відображати вибрані методи прогнозування та інші важливі аспекти, такі як обробка даних та інтерфейс користувача. Зазвичай програмне рішення для прогнозування споживання енергоресурсів складається з таких компонентів:

- модуль збору даних: відповідає за збір та очищення даних про споживання енергії.

- модуль аналізу та моделювання: включає в себе алгоритми для аналізу та моделювання споживання енергоресурсів на основі зібраних даних.

- модуль інтерфейсу користувача: дозволяє користувачам взаємодіяти з програмою, вводити дані та отримувати результати прогнозування.

Після визначення архітектури програмного рішення можна розпочати розробку та імплементацію алгоритмів прогнозування. Вибір методів та алгоритмів залежить від вибраної архітектури та типів даних. Наприклад, для часових рядів можуть бути використані методи ARIMA або нейронні мережі, тоді як для даних з економічними факторами можуть бути використані регресійні моделі.

Розробка програмного модуля включає в себе створення програмного коду для кожного компонента архітектури, імплементацію алгоритмів прогнозування та створення інтерфейсу користувача для взаємодії з програмою.

Зважаючи на описану систему прогнозування використання енергоресурсів компанії можна використовувати UML-моделі:

- діаграма варіантів використання (Use Case Diagram) передбачають Ролі "Користувач системи" і "Адміністратор системи" та варіанти використання "Робота з списком об'єктів", "Аналіз показань лічильників", "Генерація графіків", "Додавання нових об'єктів", "Редагування інформації про об'єкти" тощо;

- діаграма класів (Class Diagram): класи включають "Користувач", "Адміністратор", "Об'єкт", "Лічильник", "Прогноз", "Показання". Взаємозв'язки між класами відображають зв'язки.

– діаграму послідовності (Sequence Diagram): можна використовувати для моделювання послідовності взаємодій між об'єктами та класами під час виконання конкретних операцій: вхід до системи, внесення даних про об'єкти, прогнозування витрат ресурсів;

– діаграму активності (Activity Diagram): можна використовувати для моделювання бізнес-процесів, таких як "Прогнозування витрат електроенергії" або "Аналіз показань лічильників". Вони показують послідовність дій та рішень.

– діаграма компонентів (Component Diagram): допомагає відобразити компоненти системи та їх взаємозв'язки. Можна визначити, які компоненти відповідають за прогнозування, аналіз, візуалізацію тощо.

– діаграма розгортання (Deployment Diagram): показує, як програмне забезпечення розгортатиметься на апаратних серверах або областях інфраструктури. Вона важлива для розуміння інфраструктури, на якій працює система.

Розробка програмної системи для прогнозування та оптимізації використання енергоресурсів в компаніях може включати різні компоненти та функціональність для досягнення ефективності та контролю. Ось можливий опис такої системи:

Основні функції та компоненти системи:

1. збір та моніторинг даних:

– збір історичних даних про споживання енергоресурсів (електроенергії, газу, тепла, води) з різних джерел: лічильники, сенсори, IoT-пристрої, бази даних тощо;

– моніторинг в режимі реального часу для отримання актуальних даних.

2. Аналіз та обробка даних:

– сегментація даних за різними параметрами: часом, локацією, типом ресурсу тощо;

– аналіз історичних даних для виявлення патернів та закономірностей в споживанні;

– застосування методів статистичного аналізу та машинного навчання для прогнозування майбутнього споживання.

### 3. Прогнозування споживання:

– розробка моделей прогнозування на основі історичних даних, включаючи моделі часових рядів, регресії, нейронні мережі тощо;

– автоматичне оновлення моделей з плином часу та змінами в даних.

### 4. Оптимізація використання енергоресурсів:

– розробка алгоритмів оптимізації, які враховують бюджет, вартість енергоресурсів, екологічні показники тощо;

– планування енергоспоживання для максимізації ефективності та мінімізації витрат.

### 5. Візуалізація та звітність:

– графічний інтерфейс для відображення результатів аналізу та прогнозів;

– генерація звітів та аналітичних даних для прийняття рішень.

Після аналізу основного функціоналу було побудовано діаграму використання системи (рис. 2.1). На діаграмі `<extend>` та `<include>` - це спеціальні відносини між використовуваними випадками (use cases), які визначають специфічні взаємозв'язки між цими випадками.

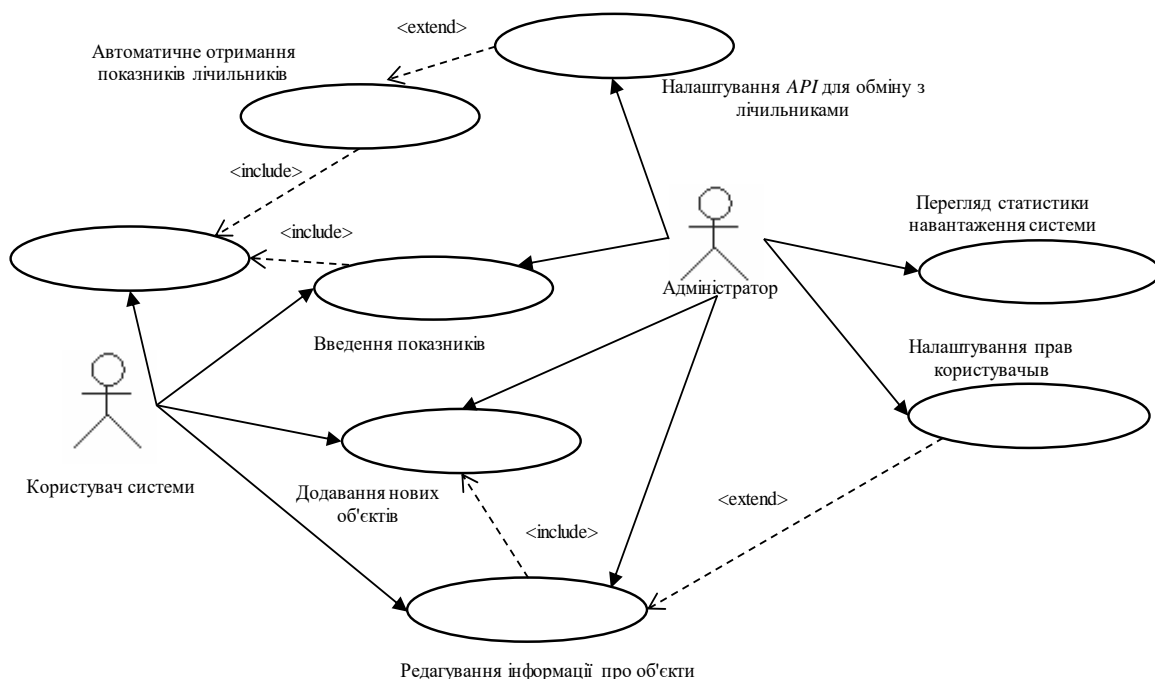


Рис. 2.1. Діаграма варіантів використання

Розширення (<extend>) використовується, коли один випадок (use case) може розширювати функціональність іншого випадку в певних умовах або сценаріях. Випадок, який розширює інший випадок, зазвичай виконується у певних додаткових умовах, і він може розширити основний випадок, надаючи додатковий функціонал.

Включення (<include>) використовується, коли один випадок (use case) включає функціональність іншого випадку. Випадок, який включає інший випадок, виконується завжди, але може включати в себе один або декілька інших випадків, які надають додаткову функціональність.

Діаграми послідовності дій для різних режимів використання системи допомагають спростити процес кодування. Наведемо основні діаграми: реєстрація користувача (рис. 2.2), введення даних об'єктів (рис. 2.3), побудова графіків історії показників (рис. 2.4).



Рис. 2.2. Діаграми послідовності дій “Реєстрація користувача”

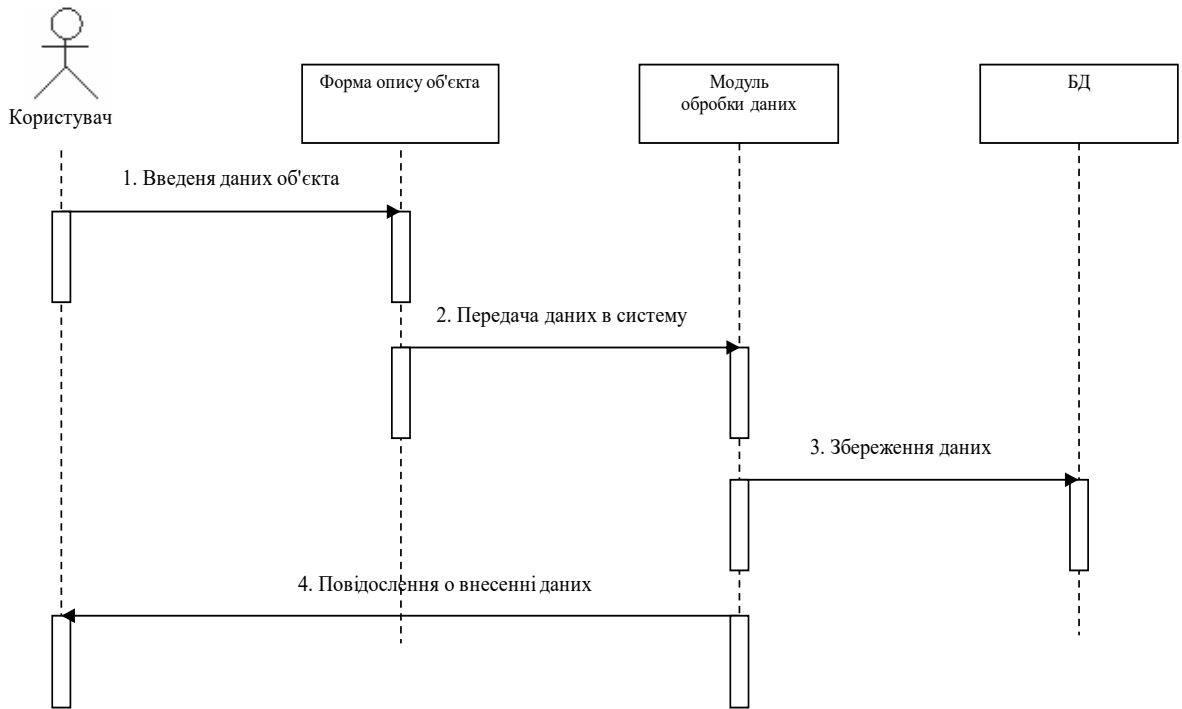


Рис. 2.3. Діаграми послідовності дій “Введення даних об'єктів”

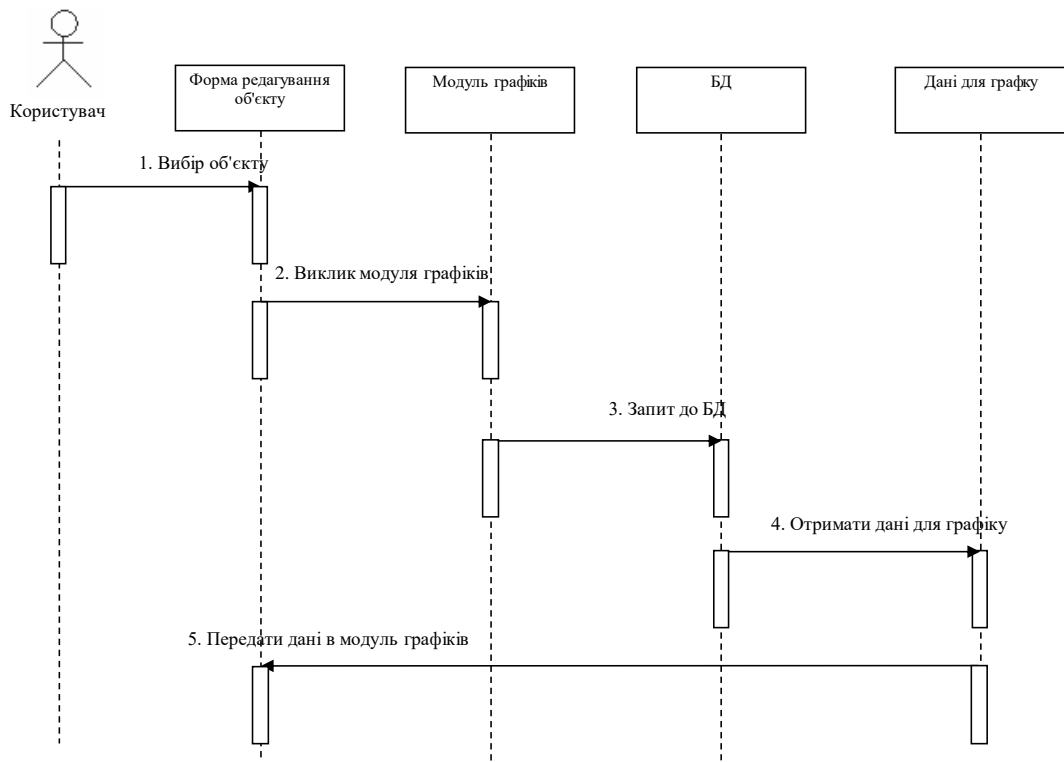


Рис. 2.4. Діаграми послідовності дій “Побудова графіків історії показників”

Діаграму активності для прогнозування витрат енергоресурсів зображено на рис. 2.5.

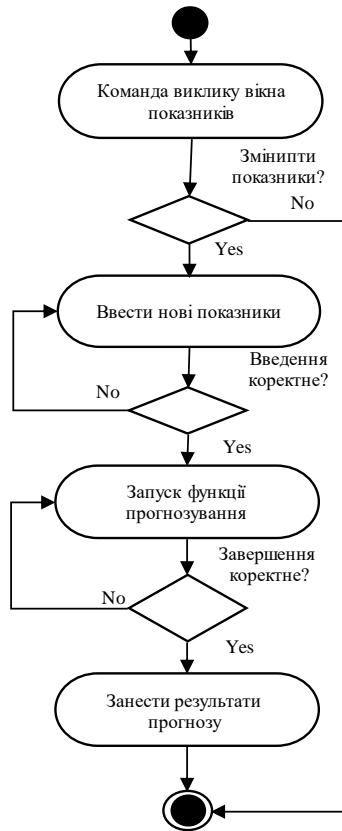


Рис. 2.5. Діаграма активності “Прогнозування витрат енергоресурсів”

Для реалізації описаного функціоналу було розроблено наступну діаграму класів (рис. 2.6).

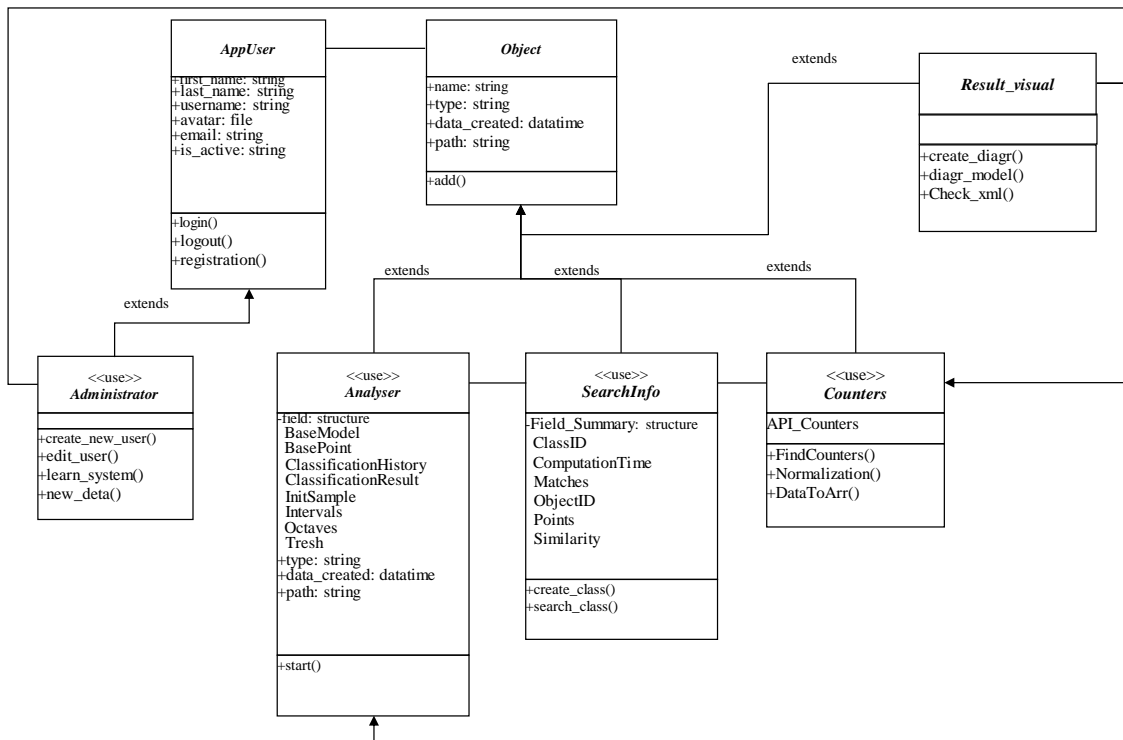


Рис. 2.6. Діаграма класів

Розробка модуля прогнозування передбачає реалізацію алгоритма, в якому необхідно провести збір даних, підготувати їх, обрати ступень полінома, побудувати модель та за необхідності відкоригувати її. Після цього модуль може проводити прогнозування.

Для оцінки точності прогнозу модель перевіряється за допомогою оцінки середньоквадратичної помилки (MSE), яку було прийнято на рівні 0,1.

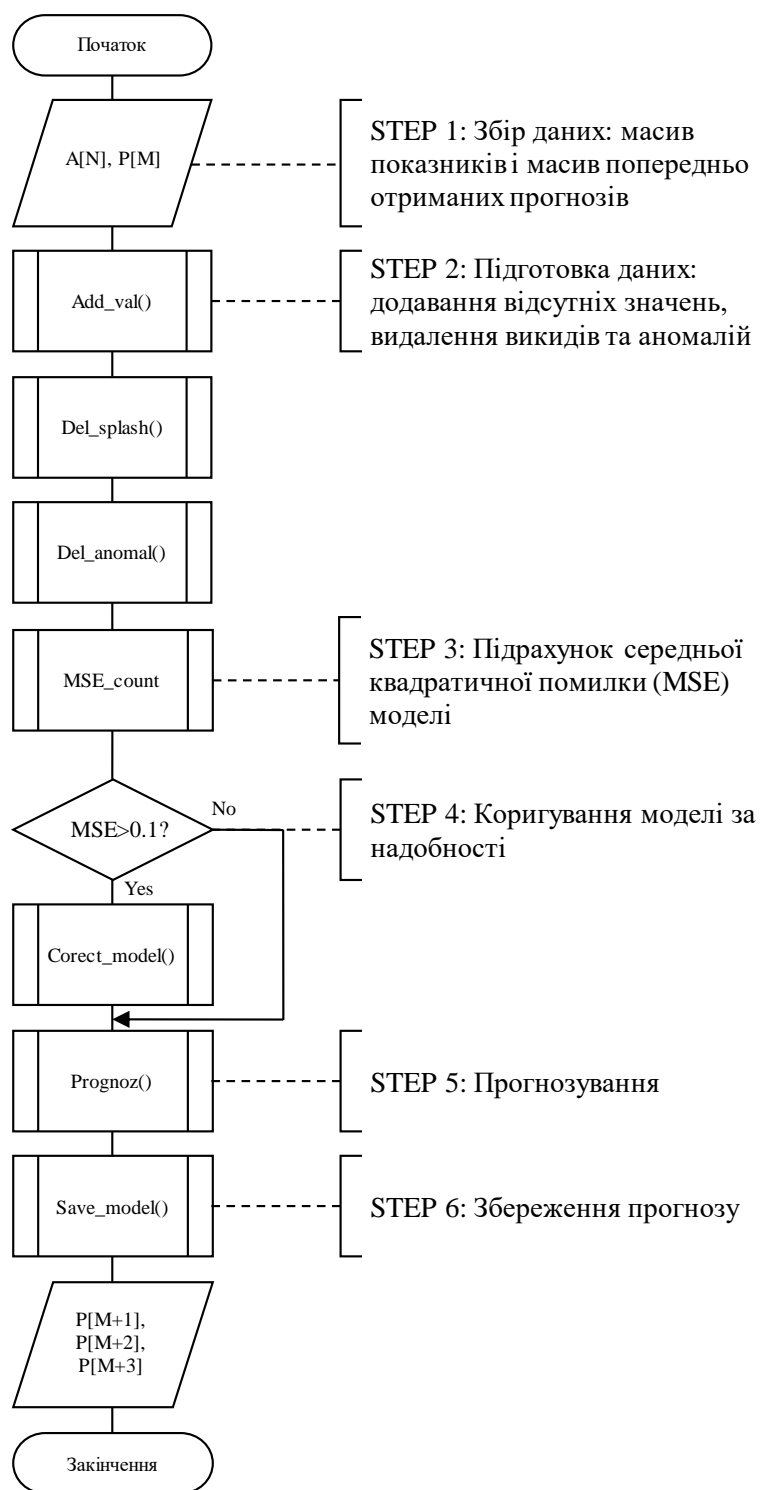


Рис. 2.7. Схема алгоритму прогнозування витрат енергоресурсів



Алгоритм передбачає 6 чітко визначених кроків – від збору даних до збереження прогнозу.

Для реалізації описаних механізмів було розроблено наступну діаграму компонентів (рис. 2.8)

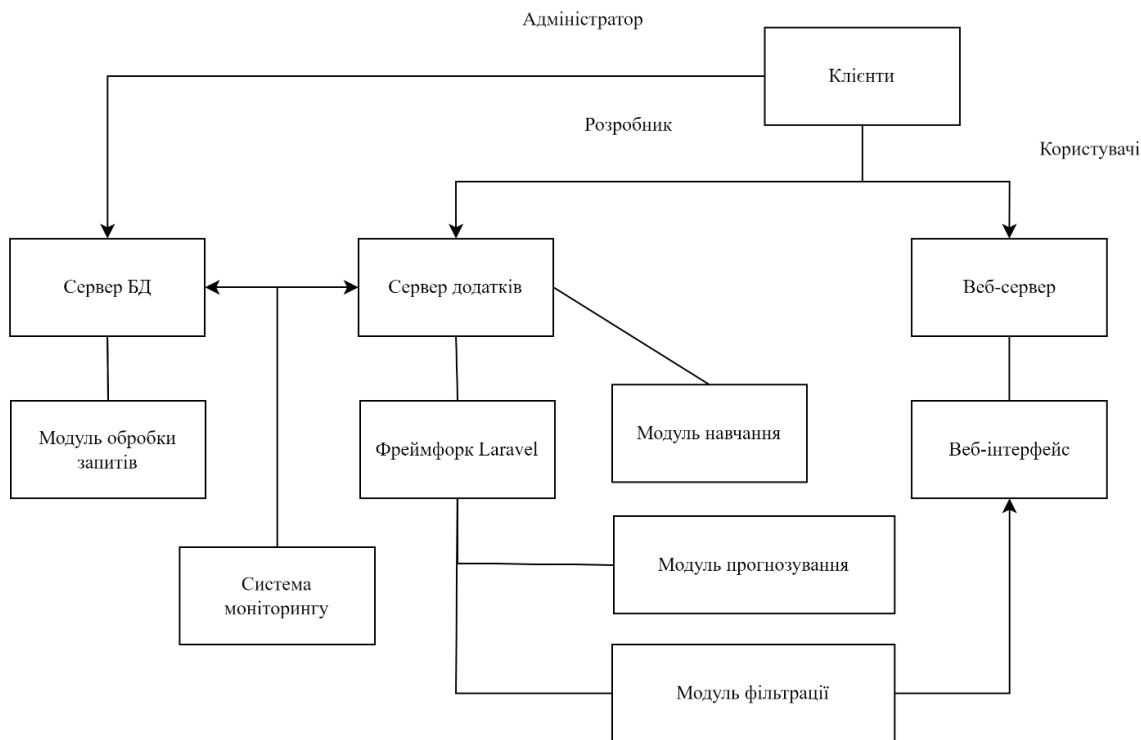


Рис. 2.8. Діаграма компонентів системи

Розробка програмного забезпечення для прогнозування та оптимізації використання енергоресурсів у компаніях є надзвичайно важливою у сучасному світі з огляду на наступні об'єктивні необхідності:

– зменшення витрат: один із основних стимулів для компаній – це зменшення витрат на енергоресурси. Ефективне прогнозування та оптимізація споживання енергії дозволяють підприємствам знижувати свої енергетичні рахунки та покращувати фінансову ефективність.

– співвідношення з довкіллям: споживання енергоресурсів, особливо великими компаніями, може суттєво впливати на довкілля через викиди в атмосферу та інші негативні екологічні наслідки. Оптимізація споживання допомагає зменшити цей вплив.

– підвищення конкурентоспроможності: компанії, які використовують ресурси ефективніше, можуть пропонувати конкурентноспроможніші ціни для своїх товарів і послуг. Це допомагає підвищити конкурентоспроможність на ринку.

– законодавство та нормативи: у багатьох країнах діють закони та регуляції, які обмежують викиди та вимагають від компаній звітувати про свої енергетичні витрати. Розробка програмного забезпечення допомагає компаніям виконувати ці нормативи та звітність.

– постійні зміни витрат та обсягів: енергоресурси мають тенденцію до змін у вартості та обсягах. Прогнозування їхнього споживання та витрат допомагає підприємствам адаптуватися до змін на ринку та планувати відповідно.

– технологічний прогрес: розвиток технологій у сфері даних, машинного навчання та аналізу даних надає можливість створювати потужні програмні рішення для енергетичної оптимізації.

– зростання свідомості громадськості та інвесторів: громадськість та інвестори стають більш уважними до питань сталості та екології. Компанії, які ведуть активну політику з енергоефективності, можуть залучити більше інвестицій та клієнтів.

– забезпечення надійності енергосистем: прогнозування споживання допомагає забезпечити надійність подачі енергії та уникнути перебоїв в постачанні.

## **Висновки до розділу 2**

Розробка програмного забезпечення для прогнозування та оптимізації використання енергоресурсів стала необхідною для більшості компаній у зусиллях знизити витрати, зменшити вплив на довкілля та підвищити ефективність їхньої діяльності.

В розділі надано детальний огляд ключових етапів розробки програмного продукту для прогнозування використання енергоресурсів компанії.

На першому етапі розробки було ретельно обрано методи прогнозування, які найкраще відповідають потребам компаній та специфікаціям проекту.

Обґрунтування розробки та архітектури ПЗ дозволило провести вибір основних компонентів та технологій для розробки системи. А побудовані діаграми використання, послідовності дій, активності і розгортання, що значно спростило розробку програмного продукту.

Розроблений алгоритм прогнозування, що використовує метод пономінальної регресії, дозволяє вести поточний прогноз і перераховувати модель у разі перевищення рівня середньоквадратичної помилки.

При реалізації даного алгоритму можна використовувати вбудовані в мови програмування бібліотеки і при виборі мови програмування враховувати наявність даних бібліотек.

## **РОЗДІЛ 3. ОПИСАННЯ РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПРОГНОЗУВАННЯ ВИКОРИСТАННЯ ЕНЕРГОРЕСУРСІВ КОМПАНІЇ**

**Мета розроблюваного програмного забезпечення** полягає в створенні інструменту, який дозволить компаніям оптимізувати та прогнозувати використання енергоресурсів з метою досягнення ефективного споживання енергії та зменшення витрат.

**Розроблюване програмне забезпечення має отримати таку функціональність:**

- збирати та аналізувати дані про споживання енергоресурсів компанією;
- прогнозувати майбутнє споживання енергії на основі історичних даних та інших факторів;
- забезпечувати моніторинг та контроль за витратами енергоресурсів в реальному часі;
- видавати рекомендації щодо оптимізації використання енергії та зменшення витрат;
- аналізувати результати та надавати користувачам звіти та статистику про споживання енергоресурсів.

### **3.1 Описання компонентів розробленого програмного забезпечення**

Програмне забезпечення побудовано на базі фреймворку Laravel, який є популярним фреймворком для розробки веб-додатків на мові програмування PHP. Він надає зручний та ефективний спосіб створення як простих, так і складних веб-додатків, і включає в себе низку функцій та засобів для полегшення процесу розробки.

Основні принципи Laravel включають:

- MVC архітектура: Laravel використовує архітектурний шаблон Model-View-Controller (MVC), що дозволяє розділити логіку додатку на три основні компоненти:
- Моделі (Models) відповідають за взаємодію з базою даних і представляють дані додатку.

– Перегляди (Views) відображають дані користувачам і представляють інтерфейс користувача.

– Контролери (Controllers) обробляють запити користувачів і взаємодіють як з моделями, так і з переглядами.

– Еклектичність: Laravel надає велику кількість готових рішень і бібліотек для різних завдань, що дозволяє розробникам швидко втілювати функціональність. Наприклад, автентифікація користувачів, маршрутизація, валідація даних, робота з базою даних і багато іншого.

– Маршрутизація: Laravel має потужну систему маршрутизації, яка дозволяє визначити, які дії повинні бути виконані при обробці кожного запиту. Маршрутизація зазвичай визначається в файлі `routes/web.php` або `routes/api.php`.

– ORM (Eloquent ORM): Laravel має вбудований ORM під назвою Eloquent, який дозволяє взаємодіяти з базою даних за допомогою PHP-об'єктів. Це спрощує роботу з базою даних і дозволяє виразно виражати логіку.

– Міграції і сідери: Laravel дозволяє створювати міграції для опису структури бази даних та сідери для наповнення її тестовими даними. Це допомагає зберігати базу даних в однаковому стані на різних середовищах розробки.

– Автентифікація і авторизація: Laravel надає готові засоби для реалізації систем автентифікації та авторизації користувачів. можете легко визначити, хто має доступ до певних ресурсів вашого додатку.

– Шаблонізація і компоненти: Laravel використовує шаблонний двигун Blade для відображення сторінок, і він підтримує вбудовані компоненти і директиви, що спрощують створення і управління переглядами.

– Тестування: Laravel включає в себе фреймворк для тестування, що дозволяє створювати автоматизовані тести для перевірки функціональності вашого додатку.

– Зручна робота з фронтендом: Laravel може легко інтегруватися з фронтенд-фреймворками, такими як Vue.js, React або Angular, щоб створювати сучасні односторінкові додатки.

Архітектура Model-View-Controller (MVC) – це популярний підхід до організації структури додатків, який використовується для розробки програмного забезпечення з графічним інтерфейсом користувача. В архітектурі MVC програма розділяється на три основні компоненти: Модель (Model), Перегляд (View) і Контролер (Controller). Кожен з цих компонентів має свою функціональність і взаємодіє з іншими компонентами для створення повноцінного додатку.

#### 1. Модель (Model):

- Модель відповідає за зберігання даних і логіку, яка обробляє ці дані.
- Вона не залежить від Перегляду і Контролера, що робить її незалежною від користувацького інтерфейсу.
- Модель може включати базу даних, обробників даних, бізнес-логіку та інші компоненти, які не прив'язані до інтерфейсу користувача.

#### 2. Перегляд (View):

- Перегляд відповідає за відображення даних користувачу та обробку подій від користувача.
- Він отримує дані з Моделі та відображає їх у вигляді, зрозумілому користувачу, часто включаючи графічний інтерфейс.
- Перегляд також може надсилати події та дії користувача Контролеру для обробки.

#### 3. Контролер (Controller):

- Контролер відповідає за обробку подій та дій користувача і взаємодіє з Моделлю та Переглядом.
- Він приймає запити від користувача, визначає, яку дію потрібно виконати, і спрямовує цю дію в Модель для оновлення даних або в Перегляд для відображення результатів.
- Контролер є посередником між Моделлю і Переглядом і забезпечує взаємодію між ними.

Робочий процес додатку на основі архітектури MVC зазвичай виглядає так:

– Користувач взаємодіє з Переглядом, натискаючи кнопки, вводячи дані тощо.

– Перегляд відправляє запит Контролеру з відомостями про подію або дію користувача.

– Контролер обробляє запит, виконує необхідні дії і може взаємодіяти з Моделлю для оновлення даних.

– Модель виконує логіку додатку і зберігає актуальні дані.

– Після обробки запиту Контролер може оновити Перегляд, відправивши йому нові дані для відображення.

Один з головних принципів архітектури MVC – це розділення обов’язків між компонентами, що дозволяє полегшити розробку, підтримку і розширення програмного забезпечення.

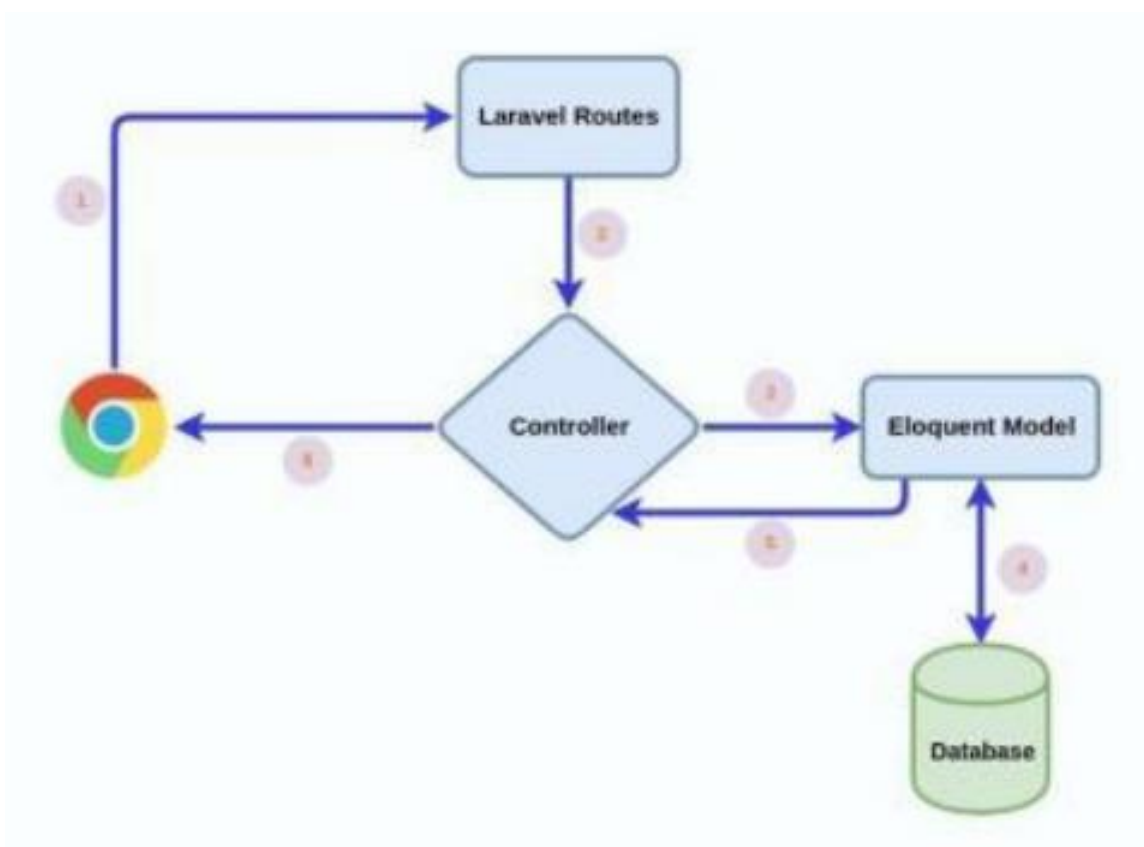


Рис. 3.1. Принцип роботи додатку за архітектурою MVC

Структура папок та файлів в Laravel-проекті може варіюватися в залежності від версії фреймворку та потреб вашого конкретного проекту.

Проте, в основі проекту Laravel зазвичай можна знайти наступні основні папки та файли:

- Папка "app": Ця папка містить основний код вашого додатку, включаючи моделі (в директорії "app/Models"), контролери (в директорії "app/Http/Controllers"), middleware та інші класи.

- Папка "bootstrap": Тут розміщуються файли, які ініціалізують фреймворк та виконують деякі налаштування перед обробкою запитів.

- Папка "config": У цій папці розміщені файли конфігурації для вашого додатку, такі як налаштування бази даних, автентифікації, маршрутизації та інші.

- Папка "database": Тут зберігаються файли для міграцій (в директорії "database/migrations") та сідерів (в директорії "database/seeders"), які допомагають вам управляти базою даних та даними.

- Папка "public": У цій папці розміщуються публічні ресурси вашого додатку, такі як стилі CSS, JavaScript-файли, зображення та інші ресурси, до яких можуть отримати доступ користувачі.

- Папка "resources": Тут зазвичай знаходяться ресурси, які необхідні для розробки фронтенду, такі як шаблони Blade (в директорії "resources/views"), файлів розробки JavaScript та CSS.

- Папка "routes": В цій папці визначаються маршрути вашого додатку. Маршрути визначаються в файлах "web.php" для веб-додатків та "api.php" для API-додатків.

- Папка "storage": Ця папка використовується для зберігання завантажених файлів, логів та інших тимчасових даних, які не повинні бути доступні публічно.

- Папка "tests": Тут розміщуються тести для вашого додатку.

- Папка "vendor": Ця папка містить залежності та бібліотеки, встановлені через Composer.

- Файл "composer.json": Цей файл містить інформацію про залежності та налаштування вашого проекту, і він використовується Composer для управління пакетами.



– Файл ".env": В цьому файлі зберігаються конфігураційні змінні, такі як налаштування бази даних та секретні ключі.

– Файл "routes/web.php" та "routes/api.php": Ці файли містять описи маршрутів для веб-додатків та API-додатків відповідно.

– Файл "app/Http/Kernel.php": У цьому файлі визначається HTTP-ядро вашого додатку, включаючи середній обробки, які застосовуються до кожного запиту (рис. 3.2).



Folder Name	Modified	Permissions
..		
.git	05.11.2023	rwxrwxr-x
app	03.11.2023	rwxrwxr-x
bootstrap	03.11.2023	rwxrwxr-x
config	03.11.2023	rwxrwxr-x
database	03.11.2023	rwxrwxr-x
public	03.11.2023	rwxrwxr-x
resources	03.11.2023	rwxrwxr-x
routes	03.11.2023	rwxrwxr-x
storage	03.11.2023	rwxrwxr-x
tests	03.11.2023	rwxrwxr-x
vendor	03.11.2023	rwxrwxr-x

Рис. 3.2. Основні папки проекту

У файлі manifest.js міститься описання розроблених модулів TypeScript. Він містить інформацію про різні модулі або компоненти, які складаються з файлів програмного коду та їх залежностей (рис .3.3).

У цьому файлі кожен модуль (або компонент) ідентифікується за ключем, який відображає ім'я вихідного файлу (наприклад, "\_ApplicationLogo-b48a7856.js"). Для кожного модуля також вказано, який файл коду він включає (параметр "file") та які інші модулі він імпортує (параметр "imports").

Цей файл може використовуватися для побудови і оптимізації проекту програмного забезпечення, зокрема, для створення бандлів (збірок) JavaScript або для керування залежностями між модулями.

..			
<input type="checkbox"/>	ApplicationLogo-b48a7856.js	3.12 KB	05.11.2023 rw-rw-r--
<input type="checkbox"/>	AuthenticatedLayout-87c7ec58.js	5.77 KB	05.11.2023 rw-rw-r--
<input type="checkbox"/>	ConfirmPassword-fa05cce9.js	1.22 KB	05.11.2023 rw-rw-r--
<input type="checkbox"/>	Create-4dc1ecac.js	6.53 KB	05.11.2023 rw-rw-r--
<input type="checkbox"/>	Dashboard-1424bbea.js	196.41 KB	05.11.2023 rw-rw-r--
<input type="checkbox"/>	Dashboard-baac8f85.css	22.76 KB	03.11.2023 rw-rw-r--
<input type="checkbox"/>	DeleteUserForm-faec534c.js	2.92 KB	05.11.2023 rw-rw-r--
<input type="checkbox"/>	Edit-92705c3c.js	1.20 KB	05.11.2023 rw-rw-r--
<input type="checkbox"/>	ForgotPassword-4ba707ff.js	1.18 KB	05.11.2023 rw-rw-r--
<input type="checkbox"/>	GuestLayout-7603d770.js	0.5 KB	05.11.2023 rw-rw-r--
<input type="checkbox"/>	InputError-a97ef805.js	0.2 KB	05.11.2023 rw-rw-r--

Рис. 3.3. Папка з згенерованими Type-script кодами

Для збереження даних було обрано СУБД MySQL.

MySQL (або "My Structured Query Language") – це відкрита система керування базами даних (СКБД), яка використовує мову запитів SQL для зберігання, управління та отримання даних. Вона є однією з найпопулярніших СКБД у світі і використовується в багатьох веб-додатках, додатках для зберігання даних та інших програмних системах. Ось деякі з переваг MySQL:

- Відкрита і безкоштовна: MySQL є відкритою програмою з вільними ліцензіями, такими як GPL (GNU General Public License). Це означає, що можете використовувати MySQL безкоштовно, а також можете отримати доступ до вихідного коду, що робить його гнучким і контрольованим.

- Висока продуктивність: MySQL володіє високою продуктивністю та швидкодією завдяки оптимізаціям, які робляться в основному для транзакцій та обробки запитів.

- Масштабованість: MySQL може легко масштабуватися від невеликих веб-сайтів до великих підприємств. можете використовувати реплікацію, кластеризацію та інші методи масштабування для забезпечення високої доступності та відмовостійкості.

– Широкий спектр функцій: MySQL підтримує багато різних типів даних, індексацію, збережені процедури, тригери та інші функції, що роблять його потужним для різноманітних завдань.

– Підтримка транзакцій: MySQL підтримує транзакції, що робить його відмінним вибором для систем, які вимагають консистентності та надійності даних, таких як системи управління банківськими операціями.

– Зручна інтеграція: MySQL легко інтегрується з багатьма мовами програмування та платформами розробки, включаючи PHP, Python, Java, Ruby та інші.

– Активна спільнота і підтримка: MySQL має велику та активну спільноту користувачів, а також підтримку від Oracle, яка надає оновлення та заходи безпеки.

– Багатофункціональність: MySQL підтримується на різних операційних системах, включаючи Linux, Windows, macOS, а також може бути встановлений на хмарних платформах.

– Можливості резервного копіювання і відновлення: MySQL надає засоби для створення резервних копій та відновлення даних, що допомагає забезпечити безпеку даних.

– Безпека: MySQL має ряд заходів безпеки, таких як аутентифікація та авторизація користувачів, що роблять його надійним для зберігання конфіденційної інформації.

Для створення бази даних для обліку витрат світла, газу, води на різних об'єктах та надання доступу користувачам, потрібно спочатку створити таблиці для зберігання даних.

Запит на створення БД наведено в Додатку А. Опишемо структуру таблиць, їх типи і призначення полів:

1. Таблиця "Users" - ця таблиця зберігає інформацію про користувачів системи.

– UserID (INT): Унікальний ідентифікатор користувача.

– Username (VARCHAR(255)): Логін користувача.

– Password (VARCHAR(255)): Пароль користувача.

2. Таблиця "Objects" - ця таблиця містить інформацію про об'єкти, які відстежуються в системі, та їх зв'язок з користувачами.

– ObjectID (INT): Унікальний ідентифікатор об'єкта.

– UserID (INT): Зовнішній ключ, що посилається на UserID в таблиці "Users" та встановлює зв'язок між користувачами і об'єктами.

– ObjectName (VARCHAR(255)): Назва об'єкта.

– Description (TEXT): Опис об'єкта.

– TotalAreaSqM (DECIMAL(10, 2)): Загальна площа об'єкта в квадратних метрах.

– HeatedAreaSqM (DECIMAL(10, 2)): Площа об'єкта, яка підігрівається в квадратних метрах.

– TotalVolumeCuM (DECIMAL(10, 2)): Загальний об'єм об'єкта в кубічних метрах.

– HeatedVolumeCuM (DECIMAL(10, 2)): Об'єм об'єкта, який підігрівається в кубічних метрах.

– Personnel (INT): Кількість персоналу.

– Visitors (INT): Кількість відвідувачів.

– PermanentResidents (INT): Кількість постійних мешканців.

– WorkHoursPerDay (INT): Робочі години на день.

– Floors (INT): Кількість поверхів.

– YearBuilt (INT): Рік побудови об'єкта.

– MapLink (VARCHAR(255)): Посилання на карту.

В даній таблиці також є зовнішній ключ "UserID," який посилається на "UserID" в таблиці "Users" для встановлення зв'язку між користувачами та об'єктами.

3. Таблиця "Contracts" - ця таблиця зберігає інформацію про договори, пов'язані з об'єктами.

– ContractID (INT): Унікальний ідентифікатор договору.

– ObjectID (INT): Зовнішній ключ, що посилається на ObjectID в таблиці "Objects" та встановлює зв'язок між об'єктами та договорами.

– Supplier (VARCHAR(255)): Постачальник (наприклад, постачальник тепла або газу).

– ContractNumber (VARCHAR(255)): Номер договору з постачальником.

– BillNumber (VARCHAR(255)): Номер рахунку з постачальником.

– HeatingRate (DECIMAL(10, 2)): Тариф на опалення.

– HotWaterRate (DECIMAL(10, 2)): Тариф на гарячу воду.

– ElectricitySupplier (VARCHAR(255)): Постачальник електроенергії.

– ElectricityContractNumber (VARCHAR(255)): Номер договору з постачальником електроенергії.

– ElectricityBillNumber (VARCHAR(255)): Номер рахунку за електроенергію.

– ElectricityPowerKW (DECIMAL(10, 2)): Потужність електроенергії (кВт).

– WaterSupplier (VARCHAR(255)): Постачальник води.

– WaterContractNumber (VARCHAR(255)): Номер договору з постачальником води.

– WaterBillNumber (VARCHAR(255)): Номер рахунку за воду.

– WaterInputMM (INT): Розмір вводу води (мм).

4. Таблиця "Meters" - ця таблиця зберігає інформацію про лічильники енергоресурсів та їх зв'язок з об'єктами та типами лічильників.

– MeterID (INT): Унікальний ідентифікатор лічильника.

– ObjectID (INT): Зовнішній ключ, що посилається на ObjectID в таблиці "Objects" для встановлення зв'язку між об'єктами та лічильниками.

– MeterTypeID (INT): Зовнішній ключ, що посилається на MeterTypeID в таблиці "MeterTypes" для встановлення зв'язку між лічильниками та типами лічильників.

– SerialNumber (VARCHAR(255)): Серійний номер лічильника.

– Manufacturer (VARCHAR(255)): Виробник лічильника.

5. Таблиця "MeterReadings" - ця таблиця зберігає інформацію про показання лічильників енергоресурсів.

– ReadingID (INT): Унікальний ідентифікатор показання.

– MeterID (INT): Зовнішній ключ, що посилається на MeterID в таблиці "Meters" для встановлення зв'язку між показаннями та лічильниками.

– ReadingDate (DATE): Дата показання лічильника.

– Consumption (DECIMAL(10, 2)): Обсяг спожитих ресурсів.

У цій структурі таблиць:

– Objects використовується для зберігання інформації про об'єкти.

– MeterTypes містить інформацію про типи лічильників (електрика, газ, вода тощо).

– Meters відображає дані про конкретні лічильники, включаючи приналежність до об'єкта та типу лічильника.

– MeterReadings містить показники лічильників із вказанням дати та кількості спожитих ресурсів. Кожен запис вказує на певний лічильник за його MeterID.

Ця структура дозволить ефективно зберігати та відстежувати дані про різні типи лічильників та їх приналежність до об'єктів.

Опис роботи з MySQL передбачає створення, налаштування, адміністрування та взаємодію з базою даних MySQL (рис. 3.4).

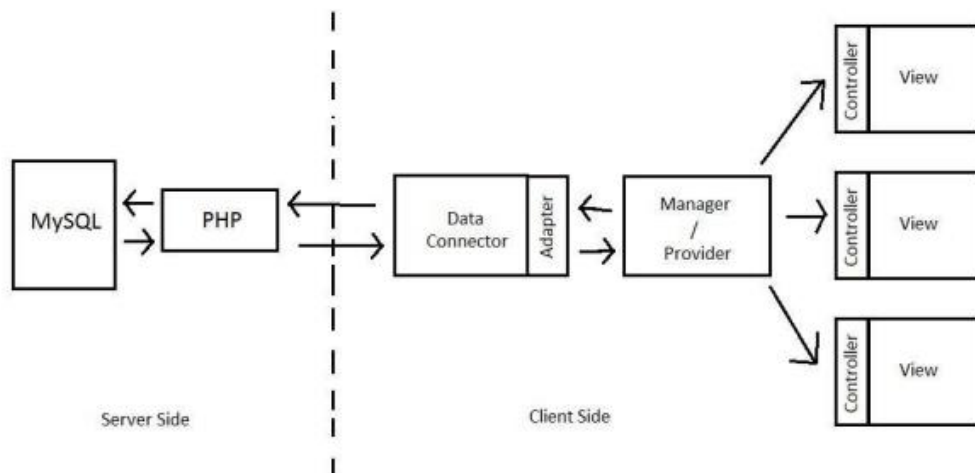


Рис. 3.4. Робота СУБД MySQL з компонентами Laravel

Для роботи з СУБД також використовується патерн репозиторій (Repository Pattern), який є структурним патерном проектування, який використовується для розділення логіки доступу до даних від інших частин

програми. Його основна ідея полягає в тому, щоб створити окремий шар абстракції для взаємодії з базою даних або іншими джерелами даних.

Основні компоненти цього патерну включають (рис. 3.5):

– Інтерфейс репозиторію: Визначає методи для доступу до даних, такі як додавання, оновлення, видалення та отримання даних. Цей інтерфейс повинен бути абстрагованим від конкретного джерела даних.

– Конкретний репозиторій: Реалізує інтерфейс репозиторію і надає конкретну реалізацію методів доступу до даних для певного джерела даних, наприклад, бази даних.

– Моделі даних: Представляють структуру даних, які зберігаються в репозиторії. В вашому випадку, це може бути інформація про споживання енергоресурсів за різні періоди часу.

– Клієнтський код: Використовує репозиторій для доступу до даних, не знаючи, як саме ці дані зберігаються.

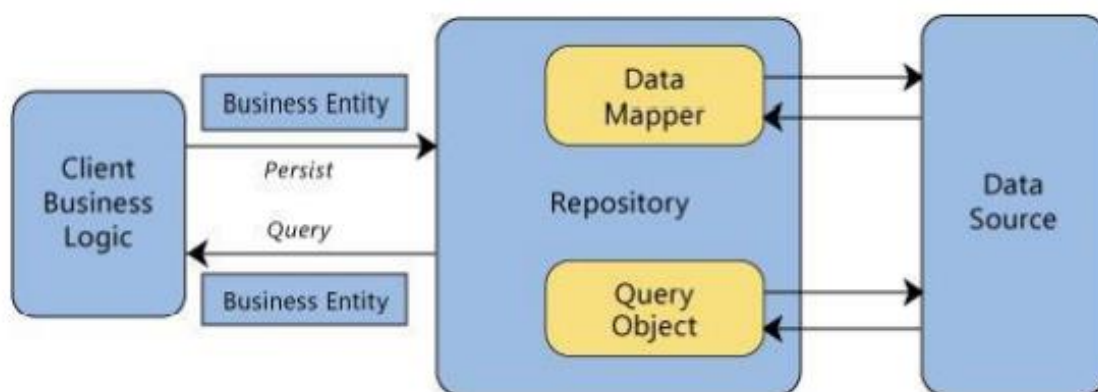


Рис. 3.5. Патерн «Репозиторій» для доступу до БД

При розробці програмного забезпечення використані елементи бібліотеки React Router. React Router – це бібліотека для навігації і маршрутизації в додатках React. Вона дозволяє вам визначити, які компоненти мають відображатися на сторінках вашого додатка на основі URL. Основною частиною React Router є компоненти, які визначають шляхи (routes), і менеджер історії для керування URL.

Основні концепції React Router:

– **BrowserRouter**: Це компонент, який використовується для обгортання вашого додатка і встановлює контекст для React Router. Він використовує HTML5 API для маршрутизації.

– **Route**: Компонент Route визначає співставлення між шляхом URL і компонентом, який має бути відображений, коли URL відповідає цьому шляху.

Приклад:

```
<Route path="/about" component={About} />
```

У цьому прикладі, коли URL має шлях "/about", компонент About буде відображений.

– **Link**: Компонент Link створює гіперпосилання для навігації між сторінками вашого додатка. Він дозволяє вам переходити до інших URL без перезавантаження сторінки. Приклад:

```
<Link to="/about">About</Link>
```

– **Switch**: Компонент Switch використовується для забезпечення вибору першого збігу шляху URL. Він вказує React Router, що потрібно відобразити тільки перший компонент, який відповідає поточному URL.

```
<Switch>  
<Route path="/about" component={About} />  
<Route path="/contact" component={Contact} />  
<Route path="/" component={Home} />  
</Switch>
```

– **Redirect**: Компонент Redirect використовується для перенаправлення користувача на іншу сторінку.

```
<Redirect from="/old-url" to="/new-url" />
```



Загальна схема роботи React Router виглядає так (рис. 3.6):

– Визначення шляхів (Routes): визначаєте шляхи URL і відповідні компоненти Route, які повинні бути відображені при збігу URL.

– Навігація: використовується компонент Link для створення гіперпосилань для навігації між сторінками.

– Відображення компонентів: коли користувач переходить за URL або використовує гіперпосилання, React Router визначає, який компонент Route потрібно відобразити на сторінці.

– Зміна URL: при навігації React Router змінює URL без перезавантаження сторінки.

– Виконання дій: у компонентів можна визначити логіку, яка повинна виконуватися при переході на конкретну сторінку.

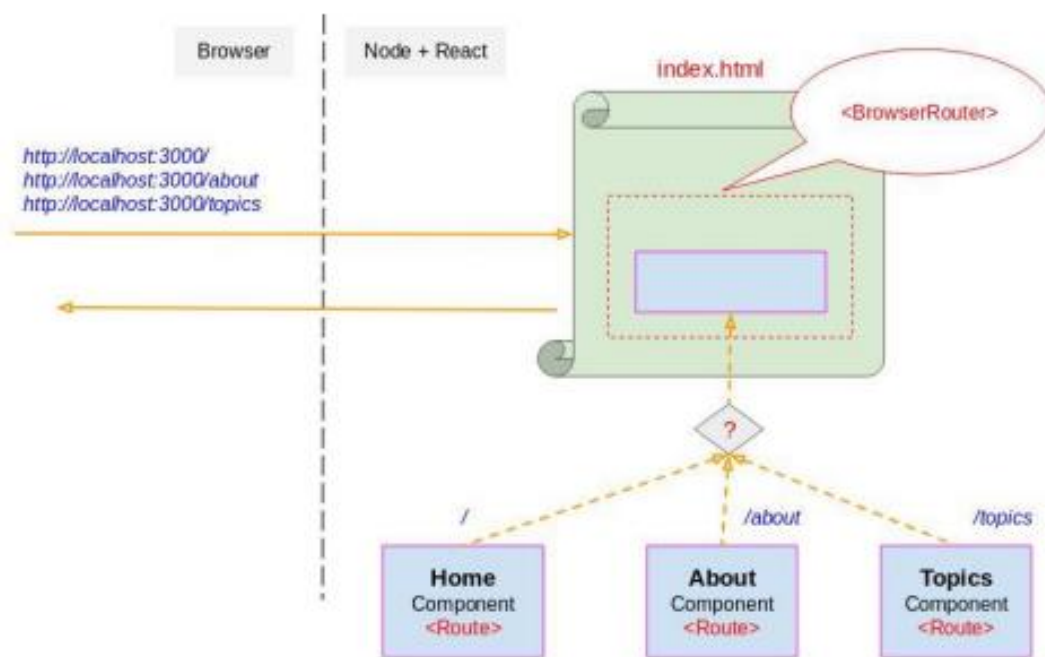


Рис. 3.6. Загальна схема роботи React Router

### 3.2 Приклади застосування додатку

Сторінка авторизації в систему – це важлива частина будь-якої веб-додатку, яка дозволяє користувачам увійти до своїх облікових записів, надаючи

доступ до функціоналу та конфіденційних даних. Ось опис типової сторінки авторизації:

- Форма входу – головний елемент сторінки – це форма входу, де користувачі вводять свої облікові дані для авторизації. Форма містить два основних поля: "Логін" і "Пароль";

- кнопка "Увійти" – після заповнення полів користувач натискає кнопку "Увійти", щоб надіслати свої дані на сервер для перевірки;

- Посилання на реєстрацію – на сторінці авторизації є посилання, яке веде на сторінку реєстрації для нових користувачів;

- посилання на відновлення паролю – сторінка авторизації містить посилання на сторінку відновлення паролю, якщо користувач забув свій пароль.

- повідомлення про помилки – якщо користувач ввів неправильні облікові дані або сталася інша помилка, сторінка виводить повідомлення про це користувачеві;

- захист від перебору паролів – для запобігання атакам перебору паролів сторінка авторизації має захист, такий як обмеження на кількість спроб вводу пароля і обмеження швидкості введення;

- безпека передачі даних – дані, які вводить користувач (логін та пароль), повинні передаватися через захищене з'єднання (https) для забезпечення конфіденційності та захисту від перехоплення;

- локалізація та дизайн – Сторінка авторизації має зручний та зрозумілий інтерфейс для користувачів.

- логування та аудит – для безпеки та аналізу діяльності користувачів ведуться логи подій.

Сторінка авторизації є однією з основних точок входу в систему і повинна бути добре розробленою та забезпечувати необхідний рівень безпеки та зручності для користувачів (рис. 3.7).

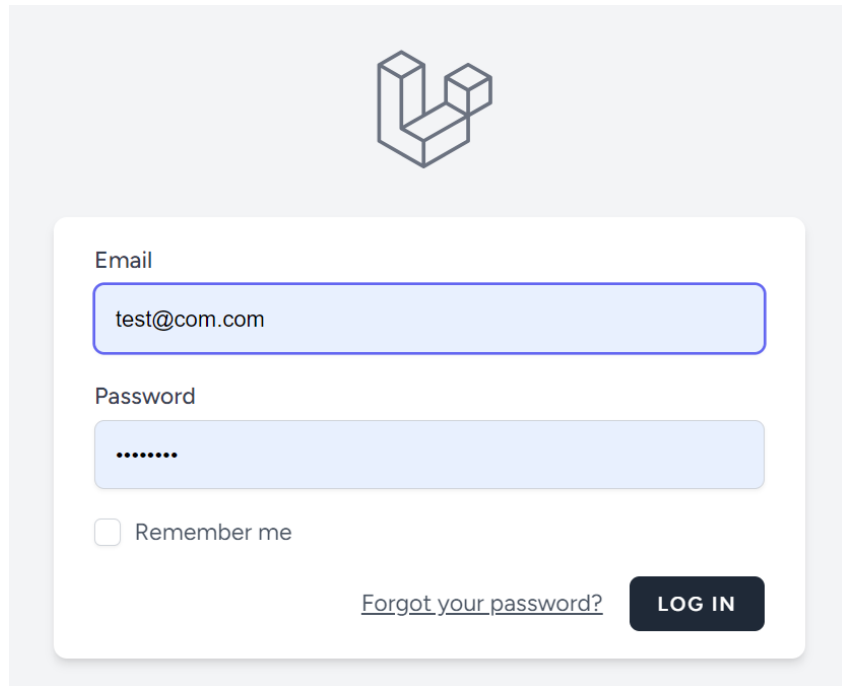


Рис. 3.7. Сторінка авторизації

Кожен користувач має доступ тільки до тих об'єктів, які ввів сам, або йому їх призначив адміністратор. Форму списку об'єктів користувача виглядає наступним чином (рис. 3.8)

Сторінка введення даних про будівлю компанії – це інтерфейс, який дозволяє користувачам внести інформацію про конкретну будівлю або об'єкт, що використовується компанією. Така сторінка може включати наступні елементи та функціонал:

- Форма введення даних:
- Основний елемент сторінки – це форма, яка містить різні поля для введення даних про будівлю. Ці поля можуть включати:
  - Загальна площа (м<sup>2</sup>).
  - Опалювальна площа (м<sup>2</sup>).
  - Загальний об'єм (м<sup>3</sup>).
  - Опалювальний об'єм (м<sup>3</sup>).
  - Кількість персоналу.
  - Кількість відвідувачів.
  - Кількість постійно присутніх.
  - Режим роботи (години на день).

- Кількість поверхів.
- Рік побудови.
- Посилання на карту (Google Maps або інше).
- Договірне навантаження тощо.

Форма була реалізована наступним чином (рис. 3.8).

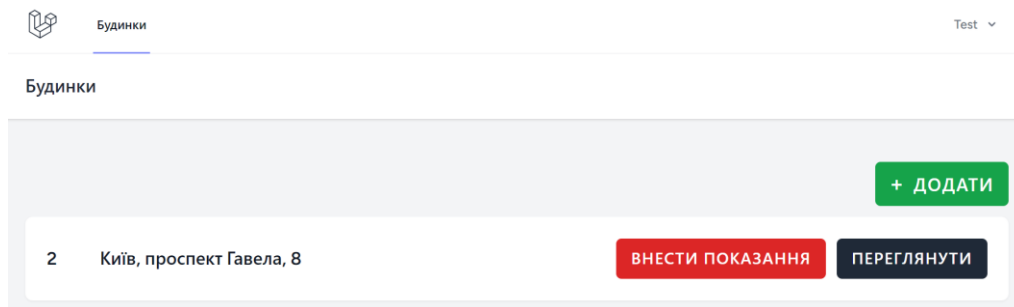


Рис. 3.8. Форма списку об'єктів користувача

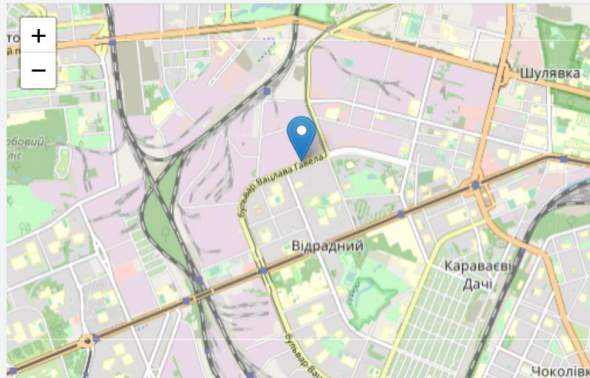
Форма повинна проводити валідацію введених даних для запобігання помилкам та некоректним значенням. Після введення та валідації даних користувач може натиснути кнопку "Зберегти" або подібну для збереження введеної інформації у базі даних. Після збереження даних користувачеві може відобразитися повідомлення про успішне збереження. Користувач має можливість редагувати введену інформацію в майбутньому, наприклад, для оновлення даних про будівлю.

Сторінка введення даних про будівлю компанії є важливою частиною системи управління інфраструктурою та ресурсами компанії, і вона має допомагати збирати та оновлювати необхідну інформацію для подальшого аналізу та прийняття рішень (рис. 3.9).

Для внесення даних лічильників використовується спеціальна форма (рис. 3.10). Також в системі передбачено налаштування API для отримання даних з лічильників за відкритим протоколом.

## Інформація

Київ, проспект Гавела, 8

Загальна S, м <sup>2</sup>	10000	<b>Адреса</b> Київ, проспект Гавела, 8 
Опалювальна S, м <sup>2</sup>	8000	
Загальний V, м <sup>3</sup>	30000	
Опалювальний V, м <sup>3</sup>	24000	
Персонал	100	
Відвідувачі	1000	
Постійно присутні	500	

Договірне навантаження					
Постачальник	ТеплоЕнерго	Постачальник	КиївЕнерго	Постачальник	Київ Водоканал
Договір	23111/234	Договір	23333/231	Договір	23333/231
Рахунок	31134555662	Рахунок	31134555662	Рахунок	31134559999
Опалення, ГКал/г	200	Електро ввід, кВт	10	Ввод води, мм	100
ГВП, ГКал/г	300				

Рис. 3.9. Форма введення даних про будівлю

<b>Внести показання</b>	Відмінити
Київ, проспект Гавела, 8	
Дата внесення	<input type="text" value="11/07/2023"/>
Газ	<input type="text" value="0"/>
Світло	<input type="text" value="0"/>
Вода	<input type="text" value="0"/>
<b>ДОДАТИ ПОКАЗНИКИ</b>	

Рис. 3.9. Форма введення даних лічильників

Для формування графіків для візуалізації історії показань лічильників з виведенням дня і значення витрат на день можна використовувати різні інструменти і технології, включаючи мови програмування, бібліотеки для графічного візуалізації та фреймворки.

Загальний підхід до цього завдання:

– збір та організація даних: спочатку потрібно мати доступ до історичних даних показань лічильників. Ці дані повинні бути організовані у вигляді таблиці з двома колонками: "Дата" і "Значення витрат";

– вибір інструменту для графіків: популярними бібліотеками для візуалізації даних є Matplotlib (для Python), Chart.js (для JavaScript), Plotly (для Python та JavaScript) та інші;

– завантаження та обробка даних: обробка даних таким чином, щоб мати можливість побудувати графік витрат на день за історичними даними;

– побудова графіка: використання інструменту для графіків для створення графіка історії показань лічильників. На осі X відобразіть дні (часову шкалу), а на осі Y відобразіть значення витрат на день;

– налаштування графіка: підписи до осей, заголовки, легенду і т. д. Доцільно також вказати шкалу дати на осі X, щоб зробити графік більш зрозумілим;

– відображення графіка: виводиться на веб-сторінці для подальшого використання та аналізу;

– взаємодія з користувачем: забезпечено можливість користувачу взаємодіяти з графіком, наприклад, масштабувати, переміщати по часовій шкалі або отримувати додаткову інформацію при наведенні курсору на точки графіку.

Цей загальний підхід може бути реалізований різними способами в залежності від інструменту і мови програмування, які ви використовуєте. Наприклад, для веб-додатків може бути використаний JavaScript та бібліотека Chart.js для відображення графіків на веб-сторінці (рис. 3.10).

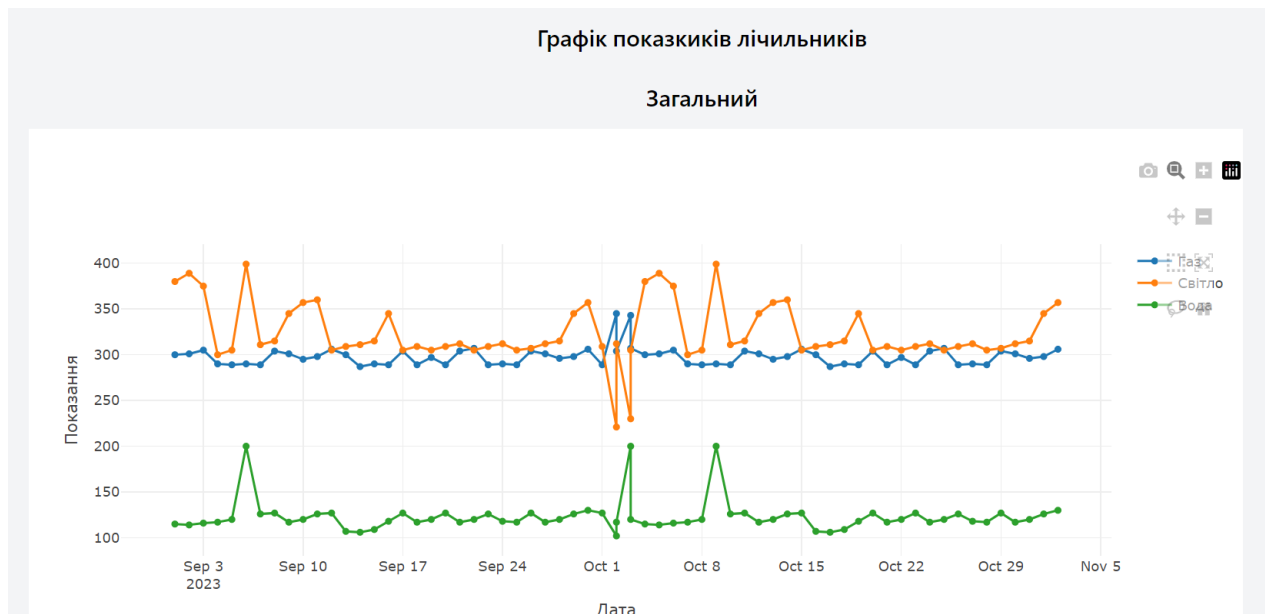


Рис. 3.10. Розроблений інструмент візуалізації показників лічильників

В розробленій системі було використано формулу поліноміальну регресії для простої лінійної моделі виглядає так:

$$y=b_0+b_1\cdot x+b_2\cdot x^2+\dots+b_n\cdot x^n+\varepsilon,$$

де:  $y$  – залежна змінна,  $x$  – незалежна змінна,  $b_0, b_1, b_2, \dots, b_n$  – коефіцієнти регресії, які потрібно оцінити під час моделювання,  $\varepsilon$  – помилка.

Для отримання коефіцієнтів  $b_0, b_1, b_2, \dots, b_n$  використано метод мінімізації помилки для чого використано бібліотеку RHP-ML і побудовано модель для поліному другого ступеня. На рис. 2 показано 3 прогностичні результати та пунктиром лінію оптимальних енерговитрат для даного приміщення.

Розроблене програмне забезпечення включає в себе інтеграцію поліноміальної регресії для прогнозування витрат енергоресурсів наступних трьох днів. Це рішення створено з метою надання користувачам більш деталізованої інформації та можливості зробити більш обґрунтовані рішення щодо ефективного використання енергоресурсів.

На графіках, які відображають історію показань лічильників, враховано не тільки фактичні значення витрат, але також і обраховану лінію оптимальних значень витрат, яка базується на поліноміальній регресії. Ця лінія дозволяє користувачам бачити, як реальні витрати відхиляються від оптимальних значень на основі історичних даних.

Додавання прогностичних значень наступних трьох днів також відображено на графіку, що допомагає користувачам передбачити майбутні витрати енергоресурсів. Ці прогнози обчислюються на основі поліноміальної регресії та дозволяють користувачам робити більш обґрунтовані рішення щодо енергозбереження (рис. 3.11).

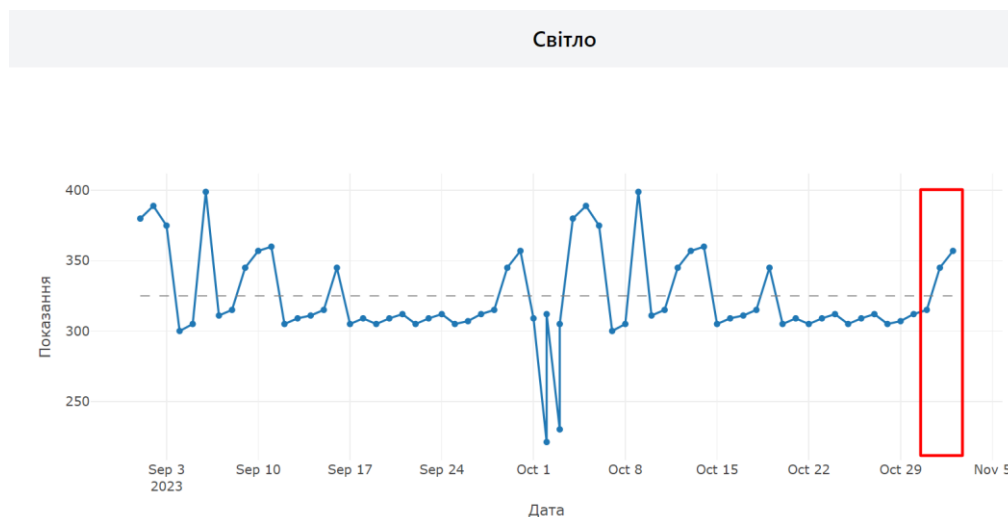


Рис. 3.11. Відображення прогностичних показників

Завдяки цим функціям програмного забезпечення, користувачі мають можливість отримувати більш повну картину щодо свого споживання енергоресурсів, ефективно планувати своє споживання і вчасно реагувати на можливі зміни в споживанні, що дозволяє їм ефективно управляти ресурсами та заощаджувати кошти.

Для обчислення поліноміальної регресії другого ступені в мові програмування PHP можна використовувати бібліотеку PHP-ML, яка містить в собі реалізацію методів машинного навчання, включаючи регресію. Ось приклад, як використовувати цю бібліотеку для обчислення поліноміальної регресії:

Встановлення бібліотеки PHP-ML за допомогою Composer:

```
composer require php-ai/php-ml
```

Нижче наведено код для обчислення поліноміальної регресії другого ступеня, який було використано в програмі для обрахунку 3 наступних значень.

```
require 'vendor/autoload.php';
```



```

use Phpml\Regression\LeastSquares;
// Вхідні дані
$x = [...];
$y = [...];
// Побудова моделі регресії
$regression = new LeastSquares(2); // Другий аргумент – ступінь полінома
// Навчання моделі
$regression->train($x, $y);
// Прогнозування значення для нового x
$newX = 6;
$predictedY = $regression->predict([$newX]);
echo "Прогнозоване значення для x=$newX: $predictedY[0]";

```

Вхідні дані представлені масивами  $x$  і  $y$ , і використовується метод найменших квадратів (LeastSquares) для побудови поліноміальної регресії другого ступеня. За допомогою навченої моделі можна прогнозувати значення для нових значень  $x$ .

В табл. 3.1 зведено значення коефіцієнтів поліному другого ступеня, які було отримано на момент проведення експерименту.

Таблица 3.1

Розраховані коефіцієнти регресійної моделі другого порядку

Тип ресурсу	$\beta_0$	$\beta_1$	$\beta_2$	$r^2$
Газ	299.6542	-0.6794	0.0236	0.0608
Електроенергія	382.0956	-7.5216	0.1970	0.3892
Вода	128.1537	-0.7974	0.0212	0.0143

Також було проведено експериментальні обчислення для поліномів вищого порядку, але навантаження на систему, яке передбачало перерахунок моделі за великим масивом вхідних даних, змусило повернутись до поліному другого порядку.

### Висновки до розділу 3

Розглянуто основні компоненти розробленого програмного забезпечення, такі як модуль збору даних, модуль прогнозування, модуль візуалізації та модуль звітності

Наведені приклади застосування розробленого додатку в реальних умовах. Додаток може бути використаний компаніями для оптимізації споживання енергоресурсів, планування бюджету на комунальні послуги та ефективного використання ресурсів. Він дозволяє підприємствам раціонально використовувати енергетичні ресурси та підтримувати сталу ефективність в галузі енергозбереження.

Представлено повний огляд розробленого програмного забезпечення та його потенціалу в сфері управління енергетичними ресурсами компаній, що підкреслює важливість такого програмного рішення для сучасних організацій, які прагнуть забезпечити ефективне використання енергії та ресурсів для забезпечення сталого розвитку.

Описане в даному розділі програмне забезпечення призначене для прогнозування використання енергоресурсів компаній, яке використовує метод поліноміальної регресії і написано мовою програмування PHP з використанням системи управління базами даних MySQL.

Головні функції програмного забезпечення включають збір і збереження даних про використання енергоресурсів для різних об'єктів компанії, підготовку даних для аналізу і прогнозування, побудову моделі на основі поліноміальної регресії для прогнозування витрат енергоресурсів, візуалізацію результатів у вигляді графіків і звітів.

Розроблене програмне забезпечення допомагає компаніям прогнозувати витрати на енергоресурси і визначати відхилення від прогнозів, що дозволяє вчасно знаходити причини відхилень і усувати їх.

## ВИСНОВКИ

Кваліфікаційна робота присвячена розробці програмного забезпечення для прогнозування використання енергоресурсів компаній. Головним завданням було створення засобу, який допоможе підприємствам ефективно управляти своїми енергетичними ресурсами та раціонально використовувати їх для забезпечення сталого розвитку.

Проведено аналіз потреб та вимог компаній у сфері прогнозування використання енергоресурсів. Було визначено, що підприємства потребують засобу для збору, аналізу та візуалізації даних про енергоспоживання, а також можливості прогнозування цього споживання на майбутнє. Визначено, що така інформація є важливою для планування бюджету на комунальні послуги, ефективного використання ресурсів та зменшення витрат.

Після детального огляду ключових етапів розробки програмного продукту було побудовано діаграми використання, послідовності дій, активності і розгортання, що спростило розробку програмного продукту.

У процесі розробки було вибрано методи прогнозування, зібрано та підготовлено дані для аналізу, розроблено архітектуру програмного забезпечення та реалізовано його функціональність. Результатом роботи став програмний продукт, який дозволяє користувачам збирати дані про енергоспоживання, аналізувати їх, будувати прогнози та отримувати візуальну інформацію для прийняття рішень.

Створено структуру, яка дозволяє зберігати та управляти інформацією про об'єкти, користувачів та їх договори з постачальниками енергоресурсів, а також вести облік витрат енергоресурсів. БД є ключовою складовою системи та дозволяє зберігати та організовувати інформацію, необхідну для прогнозування та управління енергоресурсами компанії. Вона спроектована таким чином, щоб надавати користувачам можливість ефективно виконувати аналіз та планування щодо використання ресурсів та зменшення споживання енергії.

Наведені приклади застосування розробленого програмного забезпечення, де показано, як підприємства можуть використовувати його для оптимізації споживання енергоресурсів, планування бюджету та аналізу ефективності енергозбереження.

Розроблене програмне забезпечення має великий потенціал для вирішення актуальних завдань управління енергетичними ресурсами компаній. Воно дозволяє підприємствам ефективно використовувати ресурси, планувати бюджет на комунальні послуги та забезпечувати сталий розвиток. Розроблене програмне забезпечення може стати важливим інструментом для підтримки сталої ресурсної ефективності та забезпечення енергозбереження в компаніях у сучасному світі.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ДСТУ ISO 50001:2014 Енергозбереження. Системи енергетичного менеджменту. Вимоги та настанова щодо використання (ISO 50001:2011, IDT) – [Чинний від 16.09.2014]. – (Державний стандарт України).
2. ДСТУ ISO 50006:2016 Системи енергетичного менеджменту. Вимірювання рівня досягнутої енергоефективності з використанням базових рівнів енергоспоживання та показників енергоефективності. Загальні положення і настанова (ISO 50006:2014, IDT) – [Чинний від 01.09.2016]. -(Державний стандарт України).
3. ДСТУ ISO 50002:2016 Енергетичні аудити. Вимоги та настанова щодо їх проведення (ISO 50002:2014, IDT) – [Чинний від 01.09.2016]. - (Державний стандарт України).
4. ДСТУ ISO 50004:2016 Системи енергетичного менеджменту. Настанова щодо впровадження, супровід та поліпшення системи енергетичного менеджменту (ISO 50004:2014, IDT) – [Чинний від 01.09.2016]. - (Державний стандарт України).
5. Principal Component Analysis (PCA) and Partial Least Squares (PLS) Technical Notes / [Електронний ресурс]. – Режим доступу: <http://documentation.statsoft.com>.
6. Ahn J., Chung D.H., Choa S. Energy cost analysis of an intelligent building network adopting heat trading concept in a district heating model // Energy. – 2018 – Vol. 151. – pp. 11-25.
7. Alobaidi M.H., Chebana F., Meguid M.A. Robust ensemble learning framework for day-ahead forecasting of household based energy consumption // Energy. — 2018 – Vol. 222. – pp. 997-1012.
8. Постанова Кабінету Міністрів України "Загальне положення про порядок нормування питомих витрат паливно-енергетичних ресурсів у суспільному виробництві" від 15.07.1997 р. № 786 (зі змінами та доповненнями, внесеними постановою Кабміну від 08.11.2006 р. № 1571)

[Електронний ресурс]. – Офіційний веб-сайт Верховної Ради України. – Режим доступу : <http://zakon.rada.gov.ua/cgi-bin/laws/main.cgi?nreg=786-97-%EF>.

9. Особливості короткострокового прогнозування електричного навантаження енергосистеми із суттєвою складовою промислового електроспоживання / П. О. Черненко, О. В. Мартинюк, В. О. Мірошник // Праці Інституту електродинаміки Національної академії наук України. – 2016. – Вип. 43. – С. 24-31.
10. Park, D.C. Electric load forecasting using an artificial neural network Текст. / D.C. Park, M.A. El-Sharkawi, R.J. Marks II et al. // Transaction on power systems. – 1991. – Vol. 6 (no. 2). – pp. 442-449.
11. Voronin S., Partanen J. Forecasting electricity price and demand using a hybrid approach based on wavelet transform, ARIMA and neural networks // International Journal of Energy Research. – Vol. 38. – 2014. – pp. 626-637.
12. Lidong Zhou; Bo Wang; Zheng Wang; Fei Wang; Minghui Yang Seasonal classification and RBF adaptive weight based parallel combined method for day-ahead electricity price forecasting // Conference Paper. – 2018. – P. 5.
13. Шараєвський Ф.Ю. Модель і процедури короткострокового прогнозування електроспоживання в оптовому ринку електричної енергії. // Системні дослідження та комплексні проблеми енергетики. – 2007. – С. 36–40.
14. Наказ Державного комітету з енергозбереження "Про затвердження Основних положень з нормування питомих витрат паливно-енергетичних ресурсів у суспільному виробництві" від 22.10.2002р. №112 (із змінами, внесеними згідно з наказу НАЕР від 15.06.2007р. № 92) [Електронний ресурс]. – Офіційний веб-сайт Верховної Ради України. – Режим доступу : <http://zakon.rada.gov.ua/cgi-bin/laws/main.cgi?nreg=z0878-02>.
15. Pooley John. Quick Start Guide to Energy Monitoring & Targeting (M&T) [Електронний ресурс] / John Pooley // Effective Energy Management Guide. – 2005. – Режим доступу: <http://www.oursouthwest.com/SusBus/susbus9/m&tguide.pdf>.

16. Jones Phil. Getting started with Monitoring & Targeting (M&T) / Phil Jones // Fundamental Series. – 2004. – №7. – pp. 29-32.
17. Amara F., Agbossou K., Dub?e Y., Kelouwani S., Cardenas A., Hosseini S.S. A residual load modeling approach for household short-term load forecasting application // Energy Build. – 2019. – Vol. 187 – pp. 132-143.
18. Amber K.P., Ahmad M.W. Intelligent techniques for forecasting electricity consumption of buildings // Energy. – 2018. – Vol. 157 – pp. 886-893.
19. Ascione F., Bianco N., Stasio C., Mauro G.M., Vanoli G.P. Simulation-based model predictive control by the multi-objective optimization of building energy performance and thermal comfort // Energy Build. – 2016. – Vol. 111 – pp. 131-144.
20. Mazur V. L. (2010), Metallurgy of Ukraine: camp, competitiveness, prospects, // Metallurgical and mining industry. – 2018. – Vol. 2 – pp. 12-16.
21. New methods of analysis and forecasting of hourly series in financial markets, available at: // [http://www.nauka.com/13\\_EISN\\_2013/Economics/4\\_136384](http://www.nauka.com/13_EISN_2013/Economics/4_136384) (last accessed: 11.10.2023).
22. Eckmann, J., Kamphorst, S. Recurrence Plots of Dynamical Systems // Europhysics Letters. – 1987. – Vol. 4 (9) – pp. 973 – 977.
23. Butko M. Model of thematic interpretation of view images // Innovative technologies and scientific solutions for industries. – 2021. – Vol. 2 (16) – pp. 5–11. DOI: <https://doi.org/10.30837/ITSSI.2021.16.005>
24. Zou Y., Romano MC., Thiel M., Marwan M., Kurths J. (2011), Inferring direct coupling by means of recurrences // International Journal of Bifurcation and Chaos, – 2011. – Vol. 21, No. 21 (4). – pp. 1099 – 1111.
25. Thiel M., Romano MC., Kurths J., Rolfs M., Klieg R. Generating surrogates from recurrences. Philosophical Transactions of the Royal Society // Mathematical, Physical & Engineering Sciences. – 2008. – No. 366. – pp. 545–557.
26. Piskun O.V. Peculiarities of using recurrent charts and recurrent quantitative analysis for the study of financial time series // Financial space. – 2011. – No. 3(4). – pp. 111-118.

27. Vasyuta K. S., Tansyura O. B., Revin O. V. (2013), Development of radio signal detection methods in radio engineering systems using recurrent analysis // Science and technology of the Air Force of the Armed Forces of Ukraine. – 2013. – No. 2 (11). – pp. 135 – 139.
28. Kiselev V. B. Determining the stability of the process trajectory in the phase space with the help of recurrent analysis // Scientific and technical bulletin of information technologies, mechanics and optics of SPbSU ITMO. – 2007. – No. 6 (40). – pp. 121 – 130.
29. Massaro, A. Electronic in Advanced Research Industry // From Industry 4.0 to Industry 5.0. – Wiley: Hoboken, NJ, USA. – 2021. – No. 11. – pp. 109 – 121 2021; ISBN 9781119716877.
30. Drakoulogkonas P., Apostolou D. On the Selection of Process Mining Tools // Electronics. – 2021. – No. 10. – P. 451
31. Gope A.K., Liao Y.S., Kuo C.J. Quality Prediction and Abnormal Processing Parameter Identification in Polypropylene Fiber Melt Spinning Using Artificial Intelligence Machine Learning and Deep Learning Algorithms // Polymers. – 2022. – No. 14. – pp. 2739-2749.
32. Wong, L.T., Mui K.W., Tsang, T.W. Updating Indoor Air Quality (IAQ) Assessment Screening Levels with Machine Learning Models. // Int. J. Environ. Res. Public Health. – 2022. – No. 19. – pp. 5724 – 5731.
33. Álvarez-Díaz N., Caballero-Gil P. Decision Support System Based on Indoor Location for Personnel Management // Remote Sens. – 2021. – No. 13. – pp. 248–256.
34. Fernandes J., Reis J., Melão N., Teixeira L., Amorim M. The Role of Industry 4.0 and BPMN in the Arise of Condition-Based and Predictive Maintenance: A Case Study in the Automotive Industry. Appl. Sci. – 2021. – No. 11. – pp. 34 – 38.
35. Zareen S., Akram A., Ahmad Khan S. Security Requirements Engineering Framework with BPMN 2.0.2 Extension Model for Development of Information Systems // Appl. Sci. – 2020. – No. 10. – pp. 4981-4990.



36. Massaro A. Information Technology Infrastructures Supporting Industry 5.0 Facilities // *Electronics in Advanced Research Industries*. – Wiley: Hoboken . – 2021. –No. 5. – pp. 51–101.
37. Massaro A., Maritati V., Savino N., Galiano A., Convertini D., De Fonte E., Di Muro M. A Study of a Health Resources Management Platform // *Integrating Neural Networks and DSS Telemedicine for Homecare Assistance. Information*. – 2018. –No. 9. – pp. 176-187.
38. Bastos P., Lopes I., Pires L. Application of data mining in a maintenance system for failure prediction. In *Safety Reliability and Risk Analysis // Beyond the Horizon*. – Taylor & Francis Group: London UK . – 2014. – pp. 933-940. ISBN 978-1-138-00123-7.
39. Berthold M.R., Cebron N., Dill F., Gabriel T.R., Kötter T., Meinl T., Ohl P., Sieb C., Thiel K., Wiswedel B. KNIME: The Konstanz Information Miner. In *Data Analysis Machine Learning and Applications // Springer: Berlin/Heidelberg Germany*. – 2008. – pp. 319–326.
40. Environmental Sensor Telemetry Data. Available online: <https://www.kaggle.com/datasets/garystafford/environmentalsensor-data-132k> (accessed 24 October 2023).
41. Kulinska E., Dendera-Gruszka M. Hazards in the Production Process Characteristics and Methods of Risk Elimination // *Sci. – J. Mil. Univ. Land Forces*. – 2018. –No. 189. – pp. 244–252.
42. Morales-Sánchez V., Rivero-Cruz I., Laguna-Hernández G., Salazar-Chávez G., Mata R. Chemical Composition Potential Toxicity and Quality Control Procedures of the Crude Drug of *Cyrtopodium Macrobullbon* // *Ethnopharmacol.* – 2014. –No. 154. – pp. 790–797.
43. Massaro A. “Energetic” KPI in Logistics: Complex System Theory and Multi-level Modeling. Zenodo 2022. Available online: <https://zenodo.org/record/6137729#.Y11LGXZBxPY> (accessed on 26 October 2023).
44. Magaletti N., Cosoli G., Massaro A. Wearable Smart Sensors for Health Security in Transport: The Case of Study of Diabetic Risk Management

- Thought Advanced Data Analysis Approaches Integrated into Enterprise Process Models // *Inf. Technol. Ind.* – 2021. –No. 10. – pp. 1-8.
45. Escaler X., Mebarki T. Full-Scale Wind Turbine Vibration Signature Analysis// *Machines.* – 2018. –No. 6. – pp. 63-77
46. Massaro A., Starace G. Advanced and Complex Energy Systems Monitoring and Control: A Review on Available Technologies and Their Application Criteria // *Sensors.* – 2020. –No. 22. – pp. 4929-4938.
47. Guerdelli H., Ferrari C., Barhoumi W., Ghazouani H., Berretti S. Macro- and Micro-Expressions Facial Datasets: A Survey // *Sensors.* – 2020. –No. 22. – pp. 1524-1536.
48. Savard C., Iakovleva E.V. A Suggested Improvement for Small Autonomous Energy System Reliability by Reducing Heat and Excess Charges // *Batteries.* – 2019. – No. 5. – pp. 29-39. <https://doi.org/10.3390/batteries5010029>.
49. Tsai C.-W., Chiang K.-C., Hsieh H.-Y., Yang C.-W., Lin J., Chang Y.-C. Feature Extraction of Anomaly Electricity Usage Behavior in Residence Using Autoencoder // *Electronics.* – 2022. – No. 11. – pp. 1450-1461.
50. Kusnandar, Permana I., Chiang W., Wang F., Liou C. Energy Consumption Analysis for Coupling Air Conditioners and Cold Storage Showcase Equipment in a Convenience Store // *Energies.* – 2022. –No. 15. – pp. 4857-4868. <https://doi.org/10.3390/en15134857>.
51. Ali U., Buccella C., Cecati C. Households Electricity Consumption Analysis with Data Mining Techniques // *Proceedings of the IECON 2016–42nd Annual Conference of the IEEE Industrial Electronics Society Florence Italy (23–26 October 2016).* . – 2016. – pp. 3966–3971.
52. Lis A., Sudolska A., Pietryka I., Kozakiewicz A. Cloud Computing and Energy Efficiency: Mapping the Thematic Structure of Research // *Energies.* – 2022. – No. 13. – pp. 4117-4128.
53. Khan M.H., Asar A.U., Ullah N., Albogamy F.R., Rafique M.K. Modeling and Optimization of Smart Building Energy Management System Considering Both Electrical and Thermal Load // *Energies.* – 2022. –No. 15. – pp. 574-589.

- 54.Laayati O., Bouzi M., Chebak A. Smart Energy Management System: Design of a Monitoring and Peak Load Forecasting System for an Experimental Open-Pit Mine // Appl. Syst. Innov. – 2022. –No. 5 – pp. 18-27.
- 55.Devitofrancesco A., Belussi L., Meroni I., Scamoni F. Development of an Indoor Environmental Quality Assessment Tool for the Rating of Offices in Real Working Conditions // Sustainability. – 2019. –No. 11. – pp. 1645-1654.
- 56.Nguyen T.A., Aiello M. Energy Intelligent Buildings Based on User Activity // A Survey. Energy Build. – 2013. –No. 56. – pp. 244-257.
- 57.Ткаленко О.С., Демківська Т. І. Прогнозування використання об'ємів енергоресурсів компанії // Інформаційні технології в науці, виробництві та підприємстві: Збірник наукових праць молодих вчених, аспірантів, магістрів кафедри комп'ютерних наук та технологій. – К.:ТОВ "Фастбінд Україна", 2023. – С. 128-131.

## **ДОДАТКИ**

**Код створення БД**

```
CREATE TABLE Users (
    UserID INT PRIMARY KEY AUTO_INCREMENT,
    Username VARCHAR(255) NOT NULL,
    Password VARCHAR(255) NOT NULL
);
```

Таблиця для об'єктів і її зв'язок з користувачами:

```
CREATE TABLE Objects (
    ObjectID INT PRIMARY KEY AUTO_INCREMENT,
    UserID INT,
    ObjectName VARCHAR(255) NOT NULL,
    Description TEXT,
    TotalAreaSqM DECIMAL(10, 2),
    HeatedAreaSqM DECIMAL(10, 2),
    TotalVolumeCuM DECIMAL(10, 2),
    HeatedVolumeCuM DECIMAL(10, 2),
    Personnel INT,
    Visitors INT,
    PermanentResidents INT,
    WorkHoursPerDay INT,
    Floors INT,
    YearBuilt INT,
    MapLink VARCHAR(255),
    CONSTRAINT FK_Users_Objects FOREIGN KEY (UserID) REFERENCES
Users(UserID)
);
```

Таблиці для договорів і пов'яжіть їх з об'єктами:

```
CREATE TABLE Contracts (
```

```

ContractID INT PRIMARY KEY AUTO_INCREMENT,
ObjectID INT,
Supplier VARCHAR(255) NOT NULL,
ContractNumber VARCHAR(255) NOT NULL,
BillNumber VARCHAR(255) NOT NULL,
HeatingRate DECIMAL(10, 2),
HotWaterRate DECIMAL(10, 2),
ElectricitySupplier VARCHAR(255) NOT NULL,
ElectricityContractNumber VARCHAR(255) NOT NULL,
ElectricityBillNumber VARCHAR(255) NOT NULL,
ElectricityPowerKW DECIMAL(10, 2),
WaterSupplier VARCHAR(255) NOT NULL,
WaterContractNumber VARCHAR(255) NOT NULL,
WaterBillNumber VARCHAR(255) NOT NULL,
WaterInputMM INT,
CONSTRAINT FK_Objects_Contracts FOREIGN KEY (ObjectID)
REFERENCES Objects(ObjectID)
);

```

Тепер можна створювати таблиці для обліку витрат енергоресурсів.

```

CREATE TABLE Objects (
    ObjectID INT PRIMARY KEY AUTO_INCREMENT,
    Name VARCHAR(255) NOT NULL,
    ...
);

CREATE TABLE MeterTypes (
    MeterTypeID INT PRIMARY KEY AUTO_INCREMENT,
    Name VARCHAR(255) NOT NULL
);

```

```

CREATE TABLE Meters (
    MeterID INT PRIMARY KEY AUTO_INCREMENT,
    ObjectID INT,
    MeterTypeID INT,
    SerialNumber VARCHAR(255) NOT NULL,
    Manufacturer VARCHAR(255),
    ...
    CONSTRAINT FK_Objects_Meters FOREIGN KEY (ObjectID)
REFERENCES Objects(ObjectID),
    CONSTRAINT FK_MeterTypes_Meters FOREIGN KEY (MeterTypeID)
REFERENCES MeterTypes(MeterTypeID)
);

CREATE TABLE MeterReadings (
    ReadingID INT PRIMARY KEY AUTO_INCREMENT,
    MeterID INT,
    ReadingDate DATE NOT NULL,
    Consumption DECIMAL(10, 2),
    CONSTRAINT FK_Meters_Readings FOREIGN KEY (MeterID)
REFERENCES Meters(MeterID)
);

```

### **Код програми**

NewPasswordController.php

```

<?php
namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;
use Illuminate\Auth\Events>PasswordReset;
use Illuminate\Http\RedirectResponse;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Hash;

```

```
use Illuminate\Support\Facades\Password;
```

```
use Illuminate\Support\Str;
```

```
use Illuminate\Validation\Rules;
```

```
use Illuminate\Validation\ValidationException;
```

```
use Inertia\Inertia;
```

```
use Inertia\Response;
```

```
class NewPasswordController extends Controller
```

```
{
```

```
    /**
```

```
     * Display the password reset view.
```

```
     */
```

```
    public function create(Request $request): Response
```

```
    {
```

```
        return Inertia::render('Auth/ResetPassword', [
```

```
            'email' => $request->email,
```

```
            'token' => $request->route('token'),
```

```
        ]);
```

```
    }
```

```
    /**
```

```
     * Handle an incoming new password request.
```

```
     *
```

```
     * @throws \Illuminate\Validation\ValidationException
```

```
     */
```

```
    public function store(Request $request): RedirectResponse
```

```
    {
```

```
        $request->validate([
```

```
            'token' => 'required',
```



```

        'email' => 'required|email',
        'password' => ['required', 'confirmed', Rules\Password::defaults()],
    ]);

    // Here we will attempt to reset the user's password. If it is successful we
    // will update the password on an actual user model and persist it to the
    // database. Otherwise we will parse the error and return the response.
    $status = Password::reset(
        $request->only('email', 'password', 'password_confirmation',
'token'),

        function ($user) use ($request) {
            $user->forceFill([
                'password' => Hash::make($request->password),
                'remember_token' => Str::random(60),
            ]->save();
            event(new PasswordReset($user));
        }
    );

    // If the password was successfully reset, we will redirect the user back to
    // the application's home authenticated view. If there is an error we can
    // redirect them back to where they came from with their error message.
    if ($status == Password::PASSWORD_RESET) {
        return redirect()->route('login')->with('status', __($status));
    }
    throw ValidationException::withMessages([
        'email' => [trans($status)],
    ]);
}
}

```

## HomeController.php

```
<?php
namespace App\Http\Controllers;
use App\Http\Controllers\Controller;
use App\Http\Requests\StoreHomeRequest;
use App\Http\Requests\UpdateHomeRequest;
use App\Models\Home;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Inertia\Inertia;

class HomeController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $user_id = Auth::user()->id;
        $homes = Home::whereNull('deleted_at')->where('user_id', $user_id)-
>get(['id', 'address', 'description']);
        return Inertia::render('Dashboard', compact('homes'));
    }

    public function show($id)
    {
        $user_id = Auth::user()->id;
```

```
    $home = Home::whereNull('deleted_at')->where('user_id', $user_id)->where('id', $id)->with('pokaznyky')->first();
```

```
        return Inertia::render('Home/Show', compact('home'));
    }
```

```
public function create()
{
    return Inertia::render('Home/Create');
}
```

```
public function edit(Home $home)
{
    return Inertia::render('Home/Edit', [
        'home' => $home
    ]);
}
```

```
public function store(StoreHomeRequest $request)
{
    $requestData = $request->all();

    // Convert the 'description' array to a JSON-encoded string
    $requestData['description'] = json_encode($requestData['description']);

    // Create the Home model instance with the updated data
    Home::create($requestData);

    return redirect()->route('home')
```

```

        ->with('message', __('Home created successfully.));
    }

public function update(UpdateHomeRequest $request, Home $home)
{
    $home->update($request->all());

    return redirect()->route('home')
        ->with('message', __('Home updated successfully.));
}

/**
 * Remove the specified resource from storage.
 *
 * @param \App\Models\Home $home
 * @return \Illuminate\Http\Response
 */
public function destroy(Home $home)
{
    try {
        $home->delete();
    } catch (\Throwable $th) {
        return redirect()->route('home')
            ->with('error', __($th));
    }

    return redirect()->route('home')
        ->with('message', __('Home deleted successfully'));
}

```

```
}
```

```
ProfileController.php
```

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use App\Http\Requests\ProfileUpdateRequest;
```

```
use Illuminate\Contracts\Auth\MustVerifyEmail;
```

```
use Illuminate\Http\RedirectResponse;
```

```
use Illuminate\Http\Request;
```

```
use Illuminate\Support\Facades\Auth;
```

```
use Illuminate\Support\Facades\Redirect;
```

```
use Inertia\Inertia;
```

```
use Inertia\Response;
```

```
class ProfileController extends Controller
```

```
{
```

```
    /**
```

```
     * Display the user's profile form.
```

```
     */
```

```
    public function edit(Request $request): Response
```

```
    {
```

```
        return Inertia::render('Profile/Edit', [
```

```
            'mustVerifyEmail' => $request->user() instanceof MustVerifyEmail,
```

```
            'status' => session('status'),
```

```
        ]);
```

```
    }
```

```

/**
 * Update the user's profile information.
 */
public function update(ProfileUpdateRequest $request): RedirectResponse
{
    $request->user()->fill($request->validated());

    if ($request->user()->isDirty('email')) {
        $request->user()->email_verified_at = null;
    }

    $request->user()->save();

    return Redirect::route('profile.edit');
}

/**
 * Delete the user's account.
 */
public function destroy(Request $request): RedirectResponse
{
    $request->validate([
        'password' => ['required', 'current_password'],
    ]);

    $user = $request->user();

    Auth::logout();
}

```

```
$user->delete();
```

```
$request->session()->invalidate();
```

```
$request->session()->regenerateToken();
```

```
return Redirect::to('/');
```

```
}
```

```
}
```

## Копії публікації з темою кваліфікаційної роботи

Інформаційні технології в науці, виробництві та підприємстві  
Київський національний університет технологій та дизайну

- покращення продуктивності та ефективності додатків завдяки автоматизованому аналізу даних та прийняттю розумних рішень;
- забезпечення високоякісного взаємодії з користувачем через розпізнавання образів, тексту та звуку;
- підвищення персоналізації та адаптивності додатків для кращого відповіді на потреби користувачів.

Дослідження та аналіз результатів свідчать про значні переваги та можливості, які вони пропонують розробникам у сфері машинного навчання.

## Література

1. Apple's Machine Learning [Електронний ресурс] Режим доступу: <https://developer.apple.com/machine-learning/>
2. Mars Geldard, et al. Practical Artificial Intelligence with Swift. O'Reilly, 2019. - 493

ТКАЛЕНКО О.С., ДЕМКІВСЬКА Т. І.

**ПРОГНОЗУВАННЯ ВИКОРИСТАННЯ ОБ'ЄМІВ  
ЕНЕРГОРЕСУРСІВ КОМПАНІЇ**

TKALENKO O.S., DEMKIVSKA T. I.

**SOFTWARE DEVELOPMENT FOR FORECASTING USE OF THE ENERGY RESOURCES  
COMPANY**

*Annotation: This article discusses the development of software that utilizes a multiplicative time series model to predict energy consumption. The software aims to analyze general energy usage trends within a company and anticipate future consumer behavior. The study involves aligning the input time series using the moving average method, evaluating the seasonal component, and performing analytical leveling of the data. The model parameters are estimated, and the adequacy of the model is assessed. Ultimately, the software provides a forecast of the company's electric power usage.*

*The keywords associated with this article include multiplicative model, time series, trend, seasonal component, application development, and forecasting.*

## Вступ

Прогнозування майбутнього використання енергоресурсів є важливим аспектом для успішної діяльності будь-якого підприємства. Це має прямий вплив на обсяги природних ресурсів, які будуть потрібні для виробництва, а також на фінансові витрати, пов'язані з цими ресурсами. Розробка спеціалізованого програмного забезпечення може значно спростити процес управління ресурсами в компанії.

Використання такого програмного забезпечення дозволяє компанії аналізувати тенденції використання енергоресурсів та здійснювати передбачення на майбутнє щодо споживачів. Це дає можливість компанії



наперед планувати витрати на енергоресурси і, можливо, вносити зміни відповідно до цього прогнозу, що сприяє економії коштів у майбутньому.

У підсумку, розробка програмного забезпечення для передбачення майбутнього використання енергоресурсів є важливою стратегічною ініціативою для компаній, яка допомагає забезпечити оптимальне управління ресурсами та досягнення економічної ефективності.

#### **Постановка завдання**

Мета цього дослідження полягає в створенні програмного забезпечення, яке здатне прогнозувати обсяг використання енергоресурсів на основі мультиплікативної моделі часового ряду. Це програмне забезпечення має забезпечувати можливість аналізу загальних тенденцій використання енергоресурсів компанією та передбачати ймовірну споживчу поведінку у майбутньому.

Основним завданням цього дослідження є розробка додатку, який буде доступний для різних популярних настільних і планшетних комп'ютерів, що працюють на операційній системі Windows. Метою є створення універсального програмного інструменту, який можна використовувати широкою аудиторією.

#### **Основна частина**

В ході дослідження було проведено розробку, тестування та впровадження програмного додатку для Windows. Досліджено технології, засоби розробки й мови програмування для мультиплікативної моделі часового ряду на Windows. У роботі здійснено детальний аналіз процесу розробки, тестування та впровадження додатку на Windows. Основним об'єктом дослідження були технології, засоби розробки й мови програмування для реалізації мультиплікативної моделі часового ряду. Перед початком розробки досліджено різні технології для Windows. Вивчено засоби розробки, фреймворки й мови програмування, що підходять для мультиплікативної моделі часового ряду. Обрано оптимальний набір технологій для реалізації програмного додатку на Windows

Зазвичай виділяють такі моделі часового ряду:

- адитивна модель:  $Y_t = T_t + S_t + E_t$ ;
- мультиплікативна модель:  $Y_t = T_t * S_t * E_t$ ;
- змішана:  $Y_t = T_t * S_t + E_t$ .

Для прогнозування було обрано мультиплікативну модель. Її побудова складається з таких кроків:

1. Вирівнювання вхідного часового ряду методом ковзної середньої. Отримані таким чином вирівняні значення уже не містять сезонної

квартал	ОП	КС	ЦКС	ОСК
1	77			
2	86			
3	79	78,5	78,375	0,625
4	72	78,25	78,125	-6,125
5	76	78	78,375	-0,375
6	85	74,75	76	9
7	66	77,25	77,75	-11,75
8	82	78,25	77,375	4,625
9	80	76,5		
10	78			

компоненти.

Рис.1 Вирівнювання методом ковзної середньої

2. Розрахунок значення сезонної компоненти. (Рис.2) Маємо  $0,903 + 0,789 + 1,295 + 1,007 = 3,995$ . Обчислюємо коригуючий коефіцієнт:  $k = 4/3,995 = 1,001$ .
3. Вилучення сезонної компоненти з рівнів вхідного ряду і отримання вирівняних даних (Рис. 3)

квартали	1	2	3	4		
	0	0	0,625	-6,125		
ОСК	-0,375	9	-11,75	4,625	сума	0
Середній	-0,1875	4,5	-5,8625	-0,28		-0,8
СК	0,3125	0	5,0625	-0,28		
	№	Оп	Сп	Оп-Сп	T	abs(e)
	1	77	-1,01	78,01	82,898	5,398
	2	86	0,43	85,58	81,492	4,508
	3	79	-1,45	80,45	80,588	1,588
	4	72	2,09	69,89	79,684	7,684
	5	76	-1,01	77,01	78,78	2,78
	6	85	0,43	84,58	77,876	7,124
	7	66	-1,45	67,45	76,972	10,972
	8	82	2,09	79,99	76,068	8,932
	9	80	-1,01	81,01	75,164	4,836
	10	78	0,43	77,58	74,26	3,74
Прогноз	11		-1,45		73,358	54,56
Вих ОСК	12		2,09		72,452	8,488
			Прогноз 11 квартал			71,908
			Прогноз на 12 квартал			74,902

Рис.2 Розрахунок значень сезонної компоненти

t	$Y_t$	$S_t$	$T * E = Y_t / S$	T	$T * S$	$E=Y_t / (T*S)$	$E=Y_t - (T*S)$	$E^2$
1	410	0,904	453,54	441,92	399,496	1,026	10,504	110,334
2	400	0,791	505,689	495,16	391,694	1,021	8,336	69,489
3	715	1,296	551,698	548,38	710,7	1,006	4,3	18,490
4	600	1,009	605,4	601,61	607,024	0,998	-7,024	49,337
5	585	0,904	647,124	654,64	591,976	0,988	-6,976	48,661
6	560	0,791	707,965	708,07	560,083	1,000	-0,083	0,007
7	975	1,296	752,315	761,3	966,645	0,998	-11,645	135,606
8	800	1,009	792,664	814,53	821,861	0,973	-21,861	477,903
9	765	0,904	846,239	867,76	784,456	0,975	-18,456	378,497
10	730	0,791	910,24	920,99	728,503	0,988	-8,503	72,301
11	1235	1,296	952,932	974,22	1262,589	0,978	-27,589	761,153
12	1100	1,009	1090,188	1027,45	1036,897	1,061	63,303	4007,270

Рис.3 Розрахунок вирівняних значень T і помилок E

4. Аналітичне вирівнювання рівнів ряду з використанням отриманого рівняння тренду. (Рис. 4) Рівняння тренду має наступний вигляд:  $T = 388,69 + 53,23 * t$ .

5. Розрахунок отриманих по моделі значень. Абсолютні помилки в мультиплікативній моделі визначаються як  $E = Y_t - (T * S)$ . У даній моделі сума квадратів абсолютних помилок становить 6129,037. Загальна сума квадратів відхилень фактичних рівнів ряду від середнього значення дорівнює 735606,3. Таким чином, частка поясненої дисперсії рівнів ряду дорівнює:  $(1 - 6129,037 / 735606,3) * 100 = 99,17\%$ . Модель може вважатися адекватною.

### Висновок

Досліджено метод ковзного середнього для вирівнювання часового ряду та отримано оцінку сезонної компоненти. Аналітичне вирівнювання рівнів ряду допомогло зменшити вплив незначних варіацій та нерегулярностей. Оцінено параметри мультиплікативної моделі, включаючи тренд та сезонні варіації використання енергоресурсів. Перевірено адекватність моделі для відтворення даних про споживання електроенергії.

КОРОГОД Г.О.

## ІНФОРМАЦІЙНІ СИСТЕМИ ВИЗНАЧЕННЯ ДІЙСНОГО ЗНАЧЕННЯ ФІЗИЧНОЇ ВЕЛИЧИНИ ПРИ НЕСТАБІЛЬНІЙ ФУНКЦІЇ ПЕРЕТВОРЕННЯ

KOROHOD H.O.

### INFORMATION SYSTEMS FOR DETERMINING THE ACTUAL VALUE OF A PHYSICAL QUANTITY WITH AN UNSTABLE TRANSFORMATION FUNCTION

*The work demonstrates ways of increasing the accuracy of measurement information in information systems. Methods of redundant measurements are proposed to solve this problem. It is shown that due to the derived equation of redundant measurements gives a linear dependence of the measurement result on the value of the desired physical quantity. In addition, the processing of the results in accordance with the above equation ensures the*