

АНОТАЦІЯ

Комп'ютерна система аналізу та візуалізації кліматичних даних // Дипломна робота // Батошний Дмитро Олександрович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно – інформаційних систем та програмної інженерії, група СІд-2 // Тернопіль, 2019 // с. – 99, рис. – 42 , табл. – 8, аркушів А1 – 12, додат. – 1, бібліогр. – 20.

Ключові слова: ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ІНТЕРФЕЙС, MS SQL SERVER, КОМП'ЮТЕРНА СИСТЕМА, JAVASCRIPT, C#, АНАЛІЗ ДАНИХ, ВІЗУАЛІЗАЦІЯ.

Кваліфікаційну роботу магістра присвячено дослідженню методів та засобів для створення системи аналізу та візуалізації кліматичних даних. У дипломній роботі магістра проведено аналіз наукових досліджень та існуючих програм, у результаті якого була доведена актуальність та новизна науково-дослідної роботи. Обґрунтовано доцільність використання системи аналізу та візуалізації кліматичних даних. У кваліфікаційній роботі магістра досліджено програмні засоби для розробки та планування клієнт-серверної архітектури аплікації, методами аналізу та узагальнення обрано MS SQL Server для створення бази даних, для створення бекенд частини обрано C# та JavaScript та фреймворк ReactJS для клієнта аплікації. Сплановано та створено комп'ютерну систему для аналізу та візуалізації даних, методом експерименту та узагальнення доведено наукову новизну дослідження, зокрема, особливістю розробленої аплікації є виконання логічних операцій із аналізу історичних кліматичних показників у клієнтській частині, що пришвидшило процес візуалізації інформації для користувача.

SUMMARY

Computer System of Analysis and Visualization of Climate Data // Thesis // Batozhnyi Dmytro Oleksandrovych // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, group SId-2 // Ternopil, 2019 // p. – 99, pic. – 42, tab. - 8, sheets A1 – 12, annex – 1, bibliography – 20.

Keywords: SOFTWARE, INTERFACE, MS SQL SERVER, COMPUTER SYSTEM, JAVASCRIPT, C#, DATA ANALYSIS, VISUALIZATION.

The Master's qualification work is devoted to the study of methods and tools for creating the system of analysis and visualization of climate data. In the Master's thesis has been carried out the analysis of scientific researches and existing programs, as a result of which has been proved the relevance and novelty of research scientific work. There is substantiated the feasibility of using the system of analysis and visualization of climate data. In the Master's qualification work have been studied software tools for the development and planning of client-server architecture of application, MS SQL Server has been selected as method of analysis and generalization to create the database, C# has been selected to create the backend part and JavaScript and the framework ReactJS for the client application. The computer system for data analysis and visualization has been planned and created, the scientific novelty has been proved by the method of experiment and generalization, in particular, the feature of developed application is the execution of logical operations on analysis of historical climate indicators in the client part, which speeded up the process of visualization of information for the user.

Зміст

ВСТУП	7
РОЗДІЛ 1 КЛІМАТОЛОГІЯ, МЕТЕОРОЛОГІЯ ТА ІСНУЮЧІ СИСТЕМИ ДЛЯ АНАЛІЗУ ТА ВІЗУАЛІЗАЦІЇ ДАНИХ	10
1.1. Кліматологія та метеорологія і їх дослідники	10
1.2. Види кліматичних показників та їх обчислення	14
1.3. Огляд існуючих систем для аналізу та візуалізації кліматичних даних	22
1.4. Висновки до розділу	29
РОЗДІЛ 2 ВИБІР ПРОГРАМНИХ ЗАСОБІВ ТА РОЗРОБКА СЕРВЕРНОЇ ЧАСТИНИ	31
2.1. Мова програмування Javascript та фреймворк React	31
2.2. Мови розмітки та стилів HTML/CSS	35
2.3. Мова програмування C#	37
2.4. Розробка серверної частини та створення бази даних для програми	38
2.5. Висновки до розділу	44
РОЗДІЛ 3 РОЗРОБКА КЛІЄНТСЬКОЇ ЧАСТИНИ СИСТЕМИ АНАЛІЗУ ТА ВІЗУАЛІЗАЦІЇ КЛІМАТИЧНИХ ДАНИХ	46
3.1. Розробка клієнтської частини аплікації	46
3.2. Підключення ApexCharts та створення діаграм для візуалізації даних	58
3.3. Висновки до розділу	66
РОЗДІЛ 4 ОБҐРУНТУВАННЯ ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ.	68
4.1. Розрахунок витрат на розробку та впровадження проектного рішення	68
4.2. Визначення комплексного показника якості	73
4.3. Визначення експлуатаційних витрат	76
4.4. Розрахунок ціни споживання проектного рішення	79

4.5. Визначення показників економічної ефективності	81
4.6. Висновки до розділу	83
РОЗДІЛ 5 ОХОРОНА ПРАЦІ ТА БЕЗПЕКИ В НАДЗВИЧАЙНИХ СИТУАЦІЯХ	84
5.1. Загальна характеристика приміщення і робочого місця	85
5.2. Аналіз потенційно небезпечних і шкідливих виробничих факторів на робочому місці	87
5.3. Безпека в надзвичайних ситуаціях	88
5.4. Висновки до розділу	91
РОЗДІЛ 6 ЕКОЛОГІЯ	92
6.1. Актуальність екологічних проблем	92
6.2. Етапи та техніка збору та обробки екологічної інформації	93
6.3. Висновки до розділу	96
ВИСНОВКИ	97
БІБЛІОГРАФІЯ	98
Додаток А Тези конференцій	100

ВСТУП

Актуальність теми. Проблема аналізу кліматичних даних стала дуже актуальною з початку 2010-их років. Зміни клімату відчутно впливають на різні сфери діяльності людини, як економічні (особливо в аграрному секторі, який згенерував 13% ВВП України у 2019 році), так і для соціальної сфери.

Одним із найзручніших та найлаконічніших методів вивчення та аналізу великих масивів даних є візуалізація за допомогою комп'ютерних систем. Завдяки створеним за допомогою програмного забезпечення динамічним та статичним графікам, діаграмам та картам можна чітко прослідковувати тенденції кліматологічних та метеорологічних явищ, визначати особливості та закономірності процесів для подальшого прогнозування кліматичних змін, що може бути використано для коригування планів та очікуваних результатів.

У XXI столітті створюється все більше програмного забезпечення, що підвищує швидкість та якість процесу аналізу та візуалізації даних, проте, незважаючи на велику кількість створених програм, існує дуже мало спеціальних веб-аплікацій для роботи із кліматичними даними. Серед них варто відзначити програми “GIOVANNI” та “Earth Observatory”, які створені NASA, KNMI Climate Explorer – розроблену та підтримувану Королівським метеорологічним інститутом Нідерландів та ESA CCI.

Незважаючи на сильні сторони перерахованих веб-застосунків існує чимало факторів, які ускладнюють їх експлуатацію користувачами (застарілі технології, незручний користувацький інтерфейс і т.д.). Тому актуальною науково-технічною задачею є дослідження створення клієнт-серверного програмного забезпечення для аналізу та візуалізації кліматичних даних із використанням сучасних мов програмування та засобів візуалізації історичних кліматичних даних, що дасть змогу створити аплікацію, яка б відповідала високим сучасним вимогам архітектури комп'ютерної системи та UI/UX дизайну для зручності користувачів.

Мета дослідження. Метою дослідження є аналіз методів та засобів для створення ефективної комп'ютерної системи для аналізу та візуалізації кліматичних даних.

Основні завдання дослідження:

- провести аналіз наукових публікацій та існуючого програмного забезпечення для аналізу та візуалізації кліматичних та метеорологічних даних;
- розглянути сучасні засоби створення клієнт-серверних веб-аплікацій;
- дослідити особливості опрацювання та аналізу кліматичних показників;
- проаналізувати способи візуалізації великих масивів даних;
- розробити та апробувати комп'ютерну систему для аналізу та візуалізації кліматичних даних з урахуванням сучасних вимог до розробки програмного забезпечення та UI/UX дизайну веб-застосунків.

Об'єкт дослідження – процеси розробки програми для аналізу та візуалізації кліматичних даних.

Предмет дослідження – моделі, методи та засоби представлення та розробки клієнт-серверної веб-аплікації для аналізу та візуалізації кліматичних даних.

Методи дослідження. Для вирішення поставлених задач використано наступні методи:

- аналіз та узагальнення – при проведенні аналізу існуючих інструментів для опрацювання великих масивів даних та їх візуалізації, також при огляді доступних програмних засобів для створення веб-застосунку;
- формалізації – при обґрунтуванні клієнт-серверної архітектури для розробки комп'ютерної системи для аналізу та візуалізації даних;
- проектування та програмування – при використанні обраних мов програмування, фреймворків та бібліотек для створення програмного забезпечення;

- експеримент та вимірювання – для розробки веб-аплікації для аналізу та візуалізації на основі реальних історичних кліматичних показників.

Наукова новизна одержаних результатів полягає у вирішенні науково-практичної задачі аналізу та візуалізації кліматичних показників, при цьому отримано наступні результати:

- детально обґрунтовано засоби аналізу та візуалізації вибраних кліматичних та метеорологічних даних, що дає змогу визначати та прогнозувати тенденції клімату відповідно до географічної прив'язки та часових показників;

- показано доцільність використання сучасних засобів розробки програмного забезпечення та інтерактивних методів візуалізації даних при створенні клієнт-серверних веб-аплікацій.

Практичне значення одержаних результатів. Практична цінність отриманих під час дослідження результатів полягає у тому, що спроектована та розроблена система дає змогу опрацьовувати великі масиви історичних кліматичних даних та візуалізувати результати у вигляді зручних для користування та розуміння графіків та діаграм, що в свою чергу дає змогу коригувати розроблені програми економічного та соціального розвитку підприємств та організацій та використовувати отримані результати при створенні нових програм.

Публікації. Результати дослідження апробовано на VI науково-технічній конференції Тернопільського національного технічного університету імені Івана Пулюя “Інформаційні моделі, системи та технології” (28 листопада 2019 року) у вигляді тез конференцій.

Структура роботи. Робота складається з пояснювальної записки та графічної частини. Пояснювальна записка складається із вступу, шести розділів, висновків, бібліографії та додатку. Обсяг роботи: пояснювальна записка – 100 аркушів формату А4, графічна частина – 10 аркушів формату А1.

РОЗДІЛ 1

КЛІМАТОЛОГІЯ, МЕТЕОРОЛОГІЯ ТА ІСНУЮЧІ СИСТЕМИ ДЛЯ АНАЛІЗУ ТА ВІЗУАЛІЗАЦІЇ ДАНИХ

1.1. Кліматологія та метеорологія і їх дослідники

Наука, що вивчає та пояснює фізичні процеси та явища, які відбуваються в атмосфері при її взаємодії з поверхнею ґрунту, води, рослинністю і т.д. (“підстилаюча поверхня”), називається метеорологія. Вивчення процесів та явищ, що спостерігаються в атмосфері, має велике практичне значення і дозволяє вивчити закономірності їх розвитку, що дає можливість розробляти методи прогнозування атмосферних процесів, а в окремих випадках дозволяє змінити їх розвиток [1].

Складність предмету досліджень метеорології та потреби різних галузей економіки привели до необхідності виникнення різних галузевих дисциплін. На основі поділу предмету дослідження в метеорології виділяють такі підгалузі:

- синоптична метеорологія (прогнозування погоди);
- динамічна метеорологія (теоретичні основи науки);
- актинометрія (вивчення режимів надходження потоків сонячної радіації);
- кліматологія.

Якісно та кількісно фізичний стан атмосфери та процеси, що відбуваються в ній, виражаються в показниках, що називаються метеорологічними елементами [2]. Найбільш важливими для господарської діяльності людини є наступні елементи:

- температура поверхні;
- вологість поверхні;
- атмосферний тиск;
- температура повітря;

- вологість повітря;
- хмарність;
- опади;
- вітер.

Ці елементи перебувають у постійній залежності між собою, для них характерна комплексна взаємодія, яка проявляється у складних та перемінних поєднаннях.

Стан атмосфери над конкретною територією за визначений період часу, що визначається фізичними процесами її взаємодії з підстилаючою поверхнею називається погодою.

Кліматом називається закономірна послідовність атмосферних процесів, що створюються в конкретній місцевості в результаті взаємодії сонячної радіації, атмосферної циркуляції та фізичних явищ, що відбуваються на підстилаючій поверхні, і зумовлює в цій місцевості характерний для неї режим погоди.

Клімат характеризується багаторічним режимом погоди. Під багаторічним режимом погоди розуміється не тільки переважаючі, але і загалом можливі в даній географічній місцевості погодні умови.

Основними кліматоутворюючими факторами є:

- сонячна радіація;
- циркуляція атмосфери;
- характер підстилаючої поверхні.

В результаті їх спільної взаємодії відбувається формування клімату в різних частинах земної кулі.

Область науки, що вивчає умови формування клімату та кліматичний режим країн, називається кліматологією. Кліматологія вивчає взаємозв'язки між кліматоутворюючими факторами та їх взаємодію із підстилаючою поверхнею. Кліматологія займається вивченням закономірностей в розподіленні на поверхні планети різноманітних метеорологічних явищ та типів клімату. Також вона досліджує вплив людини на клімат.

Кліматологія вивчає та описує кліматичні умови великих територій. Проте на невеликих ділянках під впливом місцевих факторів можуть створюються окремі особливості клімату. Сукупність місцевих особливостей клімату називається мікрокліматом даної географічної локації.

Галузь кліматології, що займається вивченням формуванням клімату приземного шару повітря й описом мікрокліматичних особливостей визначених географічних ділянок, називається мікрокліматологія.

Дослідники виділяють такі основні задачі кліматології як науки:

- вивчення генезису формування клімату (кліматоутворення) в результаті кліматоутворюючих процесів і під впливом географічних факторів;
- опис кліматів різних географічних областей, їх класифікація та вивчення поширення;
- дослідження кліматів історичного та геологічного минулого (палеокліматологія);
- прогноз зміни клімату;
- встановлення закономірностей формування мікроклімату та його класифікації;
- створення моделей зміни клімату в майбутньому.

Сучасна наукова метеорологія бере початок з XVII століття, коли були закладені основи фізики, частиною якої спочатку була і метеорологія. Тоді ж були винайдені перші метеорологічні прилади та з'явилася можливість інструментальних спостережень.

В цей період М.В. Ломоносов виділив метеорологію як самостійну науку із своїми задачами та методами, він створив першу теорію атмосферної електрики, дослідив можливість наукового передбачення погоди. В роботі “Про шари земні” Ломоносов одним із перших висловив теорію про зміни клімату нашої планети в процесі її розвитку.

В 1749 році були опубліковані результати дванадцятирічних спостережень за кліматом в Європі, що дало поштовх для активізації подальших метеорологічних досліджень.

В XIX столітті німецькі учені А. Гумбольдт та Г. Дове у своїх дослідженнях заклали основи кліматології. А. Гумбольдт в науковій праці “Космос” дав нове визначення клімату, в якому враховувалися різноманітні метеорологічні чинники.

В 1826 році були створені перші синоптичні карти. Авторство цього методу дослідження клімату належить німецькому ученому Г.В. Брандесу. Із середини XIX століття по ініціативі французького дослідника Л.Леверье синоптичний метод дослідження атмосферних процесів увійшов у широкий вжиток.

До цього періоду також відноситься організація перших метеорологічних науково-дослідних інститутів. В.Феррел (США) та Г.Гельмгольц (Німеччина) закладають основи динамічної метеорології. В кінці XIX століття активізувалося вивчення радіаційних та електричних процесів в атмосфері.

В XX столітті розвиток метеорології прискорилося, в 1920 році Л.Річардсон створив перший математичний прогноз погоди. У 1953 р. вперше для нескладних метеорологічних обчислень було використано ЕОМ.

Наступний етап розвитку метеорології припадає на 1961-1967 роки. У цей період у зв'язку із збільшенням кількості опрацьовуваних кліматичних показників для аналізу даних та прогнозування з'явилися перші комп'ютерні системи.

На сучасному етапі розвитку метеорології та кліматології характерне опрацювання великої маси даних, зібраних наземними та космічними станціями спостереження. Серед сучасних дослідників варто виділити роботи П. Крутцена, Дж. Тейсеран-де-Бора, М. Міланковича та М.І. Кульбіді, які займалися дослідженням глобальних змін клімату та їх впливу на розвиток біосфери.

1.2. Види кліматичних показників та їх обчислення

Кліматологічна обробка спостережень дає можливість отримати ряд кліматологічних показників, які характеризують багаторічний режим метеорологічних величин і атмосферних явищ (погоду) для тієї чи іншої місцевості. Основні види кліматологічних показників такі:

а) кліматичні показники окремих метеорологічних елементів (величин) і явищ (одномірні статистичні характеристики);

б) комплексні кліматичні показники (одномірні та багатомірні статистичні характеристики);

в) показники часової структури метеорологічних елементів і їх комплексів (одномірні та багатомірні статистичні характеристики).

До кліматичних показників окремих метеорологічних величин і явищ відносяться:

1. Середні величини. В кліматології в якості середнього використовують середнє арифметичне значення метеорологічних величин.

При великій кількості членів метеорологічного ряду вигідно велику сукупність представити у вигляді варіаційного ряду – класи і частоти варіацій.

У метеорології використовуються такі статистичні показники розподілу як середнє, мода та медіана. Вони показують загальні характеристики розподілу даних за певною змінною, дозволяють виявити одне значення (або кілька значень – якщо мода в розподілі не одна), що описує весь розподіл. Середнє, мода та медіана – це окремі значення, що представляють весь набір даних, типові для всіх значень у групі.

Є певні загальні правила для використання середнього, зокрема:

1. Середнє – це “центр тяжіння” розподілу та кожне значення дає внесок у визначення середнього значення, коли поширення значень є симетричними довкола центральної точки.

2. Середнє значення більш стабільне, ніж медіана чи мода. Тому, коли потрібно знайти найбільш стабільну міру центральної тенденції, використовують середнє.

Найвідомішою мірою центральної тенденції є середнє, або ж просте середнє, або ж арифметичне середнє (arithmetic mean) – просто середнє значення ряду даних [3]. Для його обчислення досить скласти разом всі значення в розподілі та поділити на кількість спостережень, як видно з формули:

$$M = \frac{\sum x_i}{n}, \quad (1.1)$$

де M – середнє арифметичне значення,

n – кількість показників.

Медіану можна визначити як точку на ряді розподілу (впорядкований набір значень змінної для різних спостережень, наприклад, від найменшого до найбільшого значення), до цієї точки розташовано половина всіх значень, і після цієї точки теж половина значень. Тобто, медіана – це значення, що ділить впорядкований ряд навпіл. Якщо кількість значень непарна, то береться одне зі значень – те, що стоїть у розподілі рівно по центру. Коли значень парна кількість, то беруть два центральні значення і знаходять їхнє середнє.

Медіаною функції розподілу F називається таке число \tilde{x} , що дорівнює:

$$F(\tilde{x}) = 1/2, \quad (1.2)$$

Тобто ймовірність того, що випадкова величина матиме значення більше або менше за медіану однакова і дорівнює $1/2$.

Медіана використовується для вирішення наступних завдань:

1. Коли екстремальні значення впливають на середнє – медіана є найкращою мірою центральної тенденції.

2. Коли потрібно знайти точну середню точку, точку на “півдорозі” від найменшого значення до найбільшого.

3. Медіану використовують, коли потрібно, щоб певні значення впливали на центральну тенденцію, але все, що про них відомо – що вони “нижче” або “вище” медіани.

Третя міра центральної тенденції мода – значення, що найчастіше зустрічається в розподілі. Як правило, вона представляє найбільш типове значення. На моду ніколи не впливають екстремальні значення в розподілі, а впливають – екстремальні частоти значень, наскільки часто те чи інше значення змінної зустрічається в розподілі.

Конкретне модальне значення розраховується за формулою:

$$M_0 = x_0 + h \frac{f_{m_0} - f_{m_0-1}}{(f_{m_0} - f_{m_0-1}) + (f_{m_0} - f_{m_0+1})}, \quad (1.3)$$

де x_0 та h – нижня межа та ширина модального інтервалу,

f_{m_0} , f_{m_0-1} , f_{m_0+1} – частоти (частки) відповідно модального, передмодального та післямодального інтервалів.

Мода використовується для визначення міри центральної тенденції. На моду не впливають екстремальні значення, її можна визначити для відкритих інтервалів/категорій. Для виявлення моди аналізованих кліматичних даних потрібно побудувати графік розподілу чи стовпчасту діаграму.

Можна представити розподіл даних (особливо одновимірний розподіл) різного типу графіками. Як правило, використовуються два виміри (дві осі) у графіках для розподілів. На осі X розташовують значення змінних, а на осі Y – частоту цих змінних, у абсолютних чи відносних значеннях.

Нормальний розподіл описує багато природних явищ, то він став де-факто стандартом для вирішення кліматичних та метеорологічних задач.

Розподіл, у якому всі три міри центральної тенденції збігаються, тобто середнє дорівнює медіані та дорівнює моді, називається нормальним. Його також називають розподілом Гауса або “дзвоноподібним” (рис 1.1)

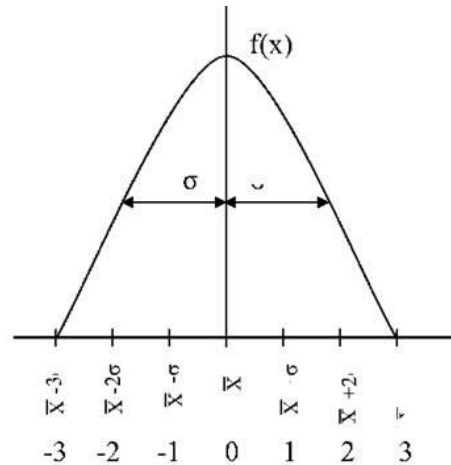


Рис. 1.1. Крива нормального розподілу (крива Гауса)

У нормальному розподілі більша частина значень даних має тенденцію до групування, “кластеризації” довкола середнього значення. Чим далі значення від середнього, тим менша ймовірність його появи. Звичайно, розподіли в реальному житті абсолютно точно не відповідають нормальному. Центральна гранична теорема теорії ймовірності свідчить про те, що сукупність незалежних, приблизно однаково розподілених величин, має нормальний розподіл.

Значення у розподілі також ділять по осі X за стандартними відхиленнями (standard deviations). Стандартне відхилення використовується як індикатор того, наскільки “розкиданими” відносно середнього є значення певної змінної в наборі даних, і вимірюється в тих же одиницях виміру, що й змінна, розподіл.

2. Крайні значення (максимальні і мінімальні), які характеризують діапазони в яких знаходяться значення метеорологічні величини, що відмічаються за певний період часу в тій чи іншій місцевості. Екстремальні значення метеорологічних величин вибирають з багаторічних спостережень. Визначають абсолютний максимум, мінімум (екстремуми), середні максимуми і мінімуми. Середні максимуми і мінімуми обчислюються як середні багаторічні значення щоденних (добовий максимум чи мінімум),

щомісячних (місячний максимум чи мінімум), щорічних (річний максимум чи мінімум).

3. Повторюваність різних значень метеорологічних величин, тобто частоти певного інтервалу в ряді статистичного розподілу виражаються в процентах або долях одиниці. Повторюваність, яку отримують з довгого ряду спостережень, в кліматології називають ймовірністю.

4. Показники мінливості значень метеорологічної величини. Сюди відноситься середнє квадратне, середнє квадратне відхилення, коефіцієнт мінливості (варіації). Середнє квадратичне – сума квадратних відхилень окремих значень даної величини від середньої поділки на число 2 варіантів. Інколи його називають середній квадрат або ж сігма, тобто середнє квадратне відхилення є числом іменованим і виражається в тих одиницях, що і середнє арифметичне. Необхідно пам'ятати, що воно має два знаки: плюс і мінус. Коефіцієнт мінливості є відношенням середнього квадратичного до середньоарифметичного відхилень. Середнє арифметичне, середній квадрат і середнє квадратне (абсолютне) відхилення називають параметрами сукупності.

5. Показники асиметрії і крутизни кривої розподілу беруться у тих випадках, коли розподіл ознаки відрізняється від нормального. Мірою асиметрії розподілу є коефіцієнт A , який дорівнює відношенню середнього куба відхилення величини ознаки від середньої арифметичної до куба середнього квадратичного відхилення. Якщо $(A) < 0,25$ – асиметрія мала, якщо ж $0.25 < (A) < 0.50$ – велика. Для характеристики крутості розподілу використовуються коефіцієнтом ексцесу (ексцес K). Коефіцієнт ексцесу коливання від -2 до 2 . Якщо K близький до 2 , то це говорить про те, що крива розподілу вдавнена і може перетворитися в двохвершинну криву; якщо $K = -2$, то крива розподілу складається з двох окремих кривих. Для метеорологічних явищ відзначаються такі кліматичні показники, як середнє і найбільше число днів з даним явищем, середня тривалість явища (в годинах).

Для аналізу кліматологічних та метеорологічних даних виділяють такі види шкал:

1. Номінальна. Шкала, значення якої не можуть бути впорядковані. Єдина операція, яку можна застосовувати до даних з номінальною шкалою – це перелік, підрахунок частоти появи того чи іншого значення в масиві.

$$A = \{x_1, \dots, x_N\}, \quad (1.4)$$

$$\delta_{ij} = \begin{cases} = 1, & \text{якщо } x_i = x_j \\ = 0, & \text{якщо } x_i \neq x_j \end{cases} \quad (1.5)$$

2. Порядкова. Змінна, значення якої за означенням впорядковані (наприклад, “слабо”, “помірно”, “сильно”). Значення такої змінної можна порівнювати на тотожність, на “більше-менше”, але при цьому не можна сказати наскільки більше чи менше. Наприклад:

$$(x > y) \text{ і } (y > z), \quad (1.6)$$

$$R_i = \sum_{j=1}^N C(x_i - x_j) \quad (1.7)$$

3. Метричні шкали.

- Шкали відношень. Шкала вимірювань кількісної властивості, змінні мають всі властивості, що й інтервальні, але також мають “природню” нульову точку “відліку” – наприклад, початок часового відліку, температура за Кельвіном.

- Інтервальна. Значення змінної впорядковані як в порядковій шкалі, але при цьому відмінності між значеннями є змістовними, наприклад, температура за Цельсієм, часові інтервали тощо. Допустимі операції – додавання та віднімання, але не множення чи ділення.

$$\widehat{x}_{ij} = \frac{x_{ij} - A}{B}, \quad (1.8)$$

де x_{ij} – j -а координата i -го вектора,

A и B – деякі задалегідь призначені числа, які називають характерними масштабами.

Серед комплексних кліматичних показників виділяються:

1. Метеорологічні комплекси однозначного ефекту, які характеризуються тим, що окремі елементи підбираються в таких градаціях і сполученнях (попередньо встановлених), щоб комплекс характеризував кількісно однозначний ефект. Наприклад, сполучення значень температури, вологості та швидкості вітру. Такий комплекс застосовується для оцінки погодної комфортності для людини. Або ж комплекс висоти хмар – видимість, який визначає умови посадки літаків.

2. Механічні комплекси – різноманітні сполучення значень різноманітних метеорологічних величин. Прикладом такого механічного комплексу є метод комплексної кліматології, розроблений Е.Е.Федоровим і Л.А.Чубуковим. Суть цього методу є в тому, що клімат будь-якої території може проявитися тільки через місцеві погоди, а погоди кожного строку спостереження розглядаються як комплекс всього набору метеорологічних елементів. Визначення градацій значень елементів погоди дає можливість виділити типи погоди й обчислити повторюваність кожного типу (градації складають для кожної метеорологічної величини і для кожного явища). Таким чином клімат виражається через місцеві погоди, що складає суть кліматичного аналізу, який називається комплексною кліматологією. Користуючись результатами спостережень складають формулу місцевої погоди кожного дня (кожна метеорологічна величина і явище мають свій шифр), потім складають каталог погод за певний період і знаходять повторюваність погод різного типу. Каталог погод дає можливість слідкувати за характером зміни погод з дня в

день. Метод комплексної кліматології з успіхом застосовується в будівельній, медичній та інших прикладних кліматологіях.

3. Погодні комплекси або комплекси метеорологічних величин і явищ, які фіксуються в період протікання тих чи інших природних процесів. Наприклад, такі комплекси метеорологічних елементів дозволяють говорити про тип погоди, який властивий таким явищам, як посухи, суховії, приморозки при різних синоптичних процесах. Такі методи природнього комплексування, які базуються на спільному використанні синоптичного та кліматологічного аналізу атмосферних процесів називаються синоптико-кліматологічними і використовуються в кліматологічних дослідженнях для вивчення структури клімату, причин його формування, в синоптичних дослідженнях.

Показники часової структури метеорологічних рядів ділять на дві групи:

1. Показники періодичних змін метеорологічних величин і явищ в часі. Сюди відноситься добовий і річний хід метеорологічних величин і явищ, амплітуда (добова і річна), дати наступання значень метеорологічних елементів. Добовий і річний хід метеоелементів виражається середніми добовими та місячними значеннями. Такі показники приводяться в довідниках по клімату, метеорологічних щомісячниках і обчислюються на основі багаторічного ряду спостережень. Добова та річна амплітуда температур визначається як різниця між найвищими та найнижчими значеннями (екстремумами) метеорологічних елементів. Терміни наступання певних значень метеорологічних елементів визначають як дати стійкого переходу значень елемента через визначені межі (наприклад, перехід середньодобової температури через 0, 5, 10 і т.д.).

2. Показники неоднорідних змін в часі метеорологічних елементів, які не приводяться в довідниках. Вони визначаються по аналогії з відповідними характеристиками випадкових функцій, статистичними оцінками яких вони являються. При тому зміна метеорологічних елементів в часі розглядається як

деякий випадковий процес, який підпорядковується законам теорії імовірностей.

Результати щоденних спостережень на метеостанціях фіксуються в спеціальну книжку спостерігача, яка є основним джерелом інформації прогнозу і клімату території, де розташована станція. На основі цього первинного матеріалу, який міститься в книзі спостерігача складаються місячні таблиці, в яких крім результатів приземних термінових спостережень, приводяться середні та сумарні величини, експериментальні значення за добу, декаду, місяць, число днів з атмосферними явищами за місяць. Тобто місячні таблиці являють собою перший етап обробки результатів метеорологічних спостережень за місяць. Таблиці дуже ретельно перевіряються на метеостанції і в обласному гідрометцентрі. Після перевірки місячні таблиці є готові для друку. До 1960 року в СРСР вони друкувалися у вигляді метеорологічних щорічників.

Тепер обласні та державні гідрометцентри видають метеорологічні щомісячники. За рік видається 13 випусків. Щомісячники від 1 до 12 місяця містять дані місячних таблиць, а в 13-му номері публікуються річні дані метеорологічних спостережень. Метеорологічні щомісячники складаються з двох частин.

1.3. Огляд існуючих систем для аналізу та візуалізації кліматичних даних

В процесі дослідження було розглянуто ряд комп'ютерних систем для аналізу та візуалізації кліматичних даних.

Джованні – це аббревіатура для GES-DISC (Центру даних та інформаційних служб Goddard Space Science) [4]. Основні можливості програми виражаються у аббревіатурі GrADS - Grid Analysis and Display System. Інтерактивна онлайн-візуалізація GES-DISC та система аналізу “GIOVANNI”

дозволяє досліджувати супутникові дані, використовуючи складний аналіз та візуалізацію.

Серед них варто виділити систему “GIOVANNI” – це веб-додаток, розроблений GES DISC, який забезпечує простий та інтуїтивний спосіб візуалізації, аналізу та доступу до величезної кількості даних дистанційного зондування Землі без необхідності завантаження даних, зображений на рис. 1.2.

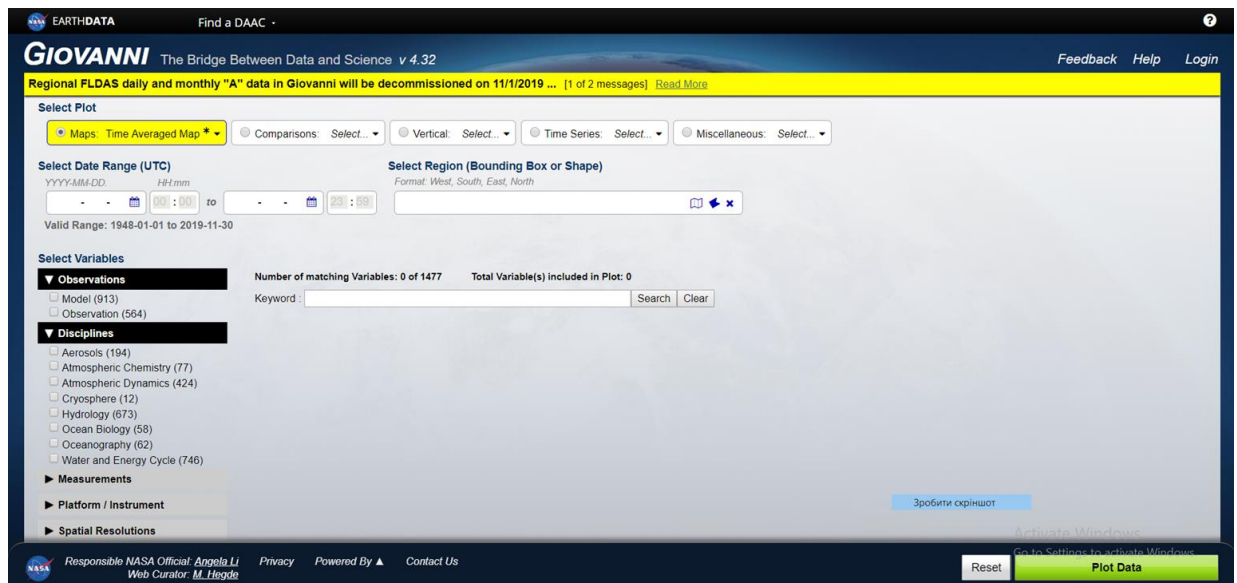


Рис. 1.2. Головна сторінка веб-сайту “GIOVANNI”

Система дозволяє отримати доступ до даних з декількох віддалених сайтів, підтримує кілька форматів даних, включаючи ієрархічний формат даних (HDF), HDF-EOS, мережеву загальну форму даних (netCDF), GRIdded Binary (GRIB), а також двійкові та багато типів сюжетів, включаючи область, час, анімація Ховмоллера та зображення.

Зокрема, ця аплікація дозволяє користувачам швидко візуалізувати, взаємодіяти та аналізувати дані про колір океану з різних місій, а також набори даних з багатьох інших архівів даних NASA.

У застосунку є можливість візуалізації радіометрії супутникового дистанційного зондування, пігментування фітопланктону, температури поверхні моря та солоності морської поверхні, як зображено на рис. 1.3.

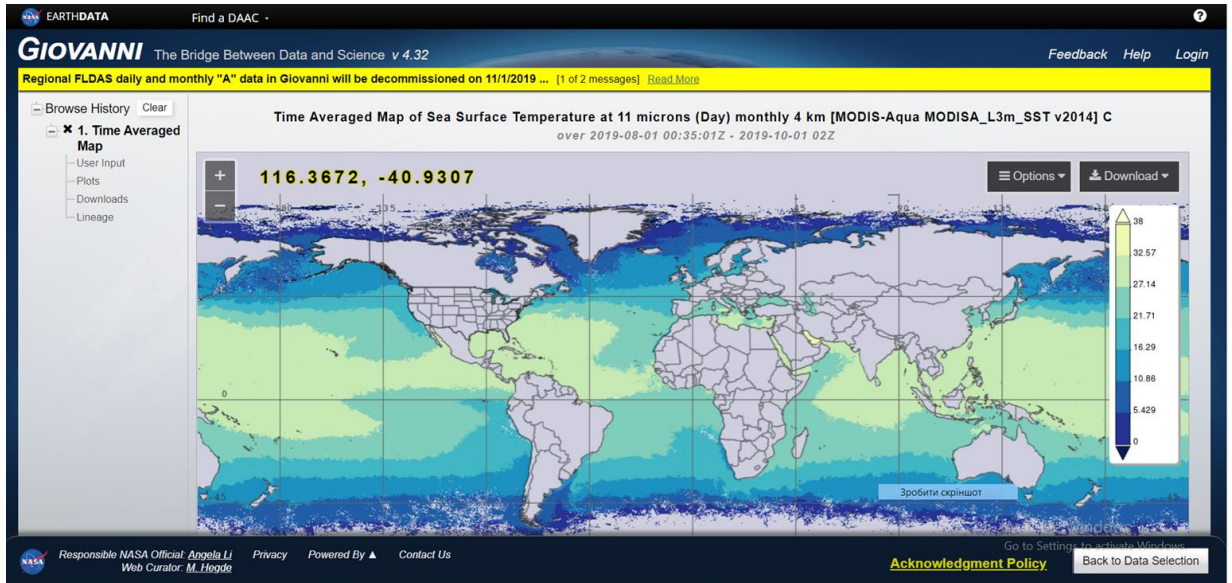


Рис. 1.3. Візуалізація температури океану в програмі “GIOVANNI”

Також доступні анімовані візуалізації аналізованих даних та числовий вивід інформації.

Перевагою інтерактивного веб-застосунку для візуалізації та аналізу даних Центру даних та інформаційних послуг для наук про Землю імені Р. Годдарда є отримання даних з супутників NASA безпосередньо в Інтернеті, без труднощів із традиційними методами збору та аналізу даних.

Вчені використовували цей інструмент в праці 2005 року, що досліджували реакцію затоки Чесапек на сильні опади в середній частині Атлантики у 2005 році, у 2007 році використовували у дослідженнях фітопланктону в Червоному морі та концентрації газів над Індією. У 2008 році із використанням інструменту змодельовано хмарний покрив над Гренландією.

Веб-сайт Giovanni пропонує різні ресурси для освітян та користувачів, включаючи онлайн-посібник користувача та сторінки, що описують доступні інтерфейси даних та типи сюжетів.

Далі було опрацьовано KNMI Climate Explorer. Це веб-додаток для статистичного аналізу кліматичних даних, розроблений Королівським метеорологічним інститутом Нідерландів (KNMI). Він був створений наприкінці 1999 року як проста веб-сторінка для аналізу кліматичних

показників, на даний момент опрацьовано 10 ТБ кліматичних даних та створено десятки інструментів аналізу. Зараз він є частиною Регіонального кліматичного центру ВМО при KNMI разом з ECA&D.

Після безкоштовної реєстрації, науковці можуть досліджувати та завантажувати масиви кліматичних наборів даних, генерувати досліджувані дані, завантажувати власні дані та проводити статистичний аналіз, досліджувати та порівнювати набори даних.

Після початку функціонування дана система аналізу та візуалізації даних швидко здобула популярність як інструмент для досліджень зміни клімату. Форма для створення інфографіки для демонстрації кліматичних змін зображена на рис. 1.4.

The screenshot displays the KNMI Climate Explorer interface. At the top, there is a navigation bar with 'Climate Explorer', 'European Climate Assessment & Data', and 'KNMI'. Below this is a search bar and a navigation menu with options like 'Help', 'News', 'About', 'Contact', 'World weather', 'Effects of El Niño', 'Seasonal forecasts', and 'Climate Change Atlas'. The main content area is titled 'KNMI Climate Change Atlas' and contains several configuration sections:

- Select a region:** Type (IPCC WG1, IPBES, countries, place, box), IPCC WG1 (World).
- Select a season:** Season (First month: Jan, length: 12 months).
- Select a dataset and variable:** Dataset (GCM: CMIP5 (IPCC AR5 Atlas subset)), Variable (near-surface temperature), Output (absolute, relative changes are shown, map, time series).
- Map options:** Scenario (Historical + RCP4.5), Measure (Difference of two periods), Reference period (1986 - 2005), Future period (2081 - 2100), Mean/percentiles (mean).

Additional features include a 'Make map' button, a 'Further information' section with links to reports and scenarios, and a 'Funding' section listing KNMI, SPECS, and the Red Cross / Red Crescent Climate Centre. A 'Зробити скріншот' (Take screenshot) button is also present.

Рис. 1.4. Форма для створення інфографіки у KNMI Climate Explorer

Аплікацію використовує палеокліматичне дослідницьке співтовариство для вивчення та завантаження доступних аналізованих кліматичних показників за обраний період, як показано на рис. 1.5.

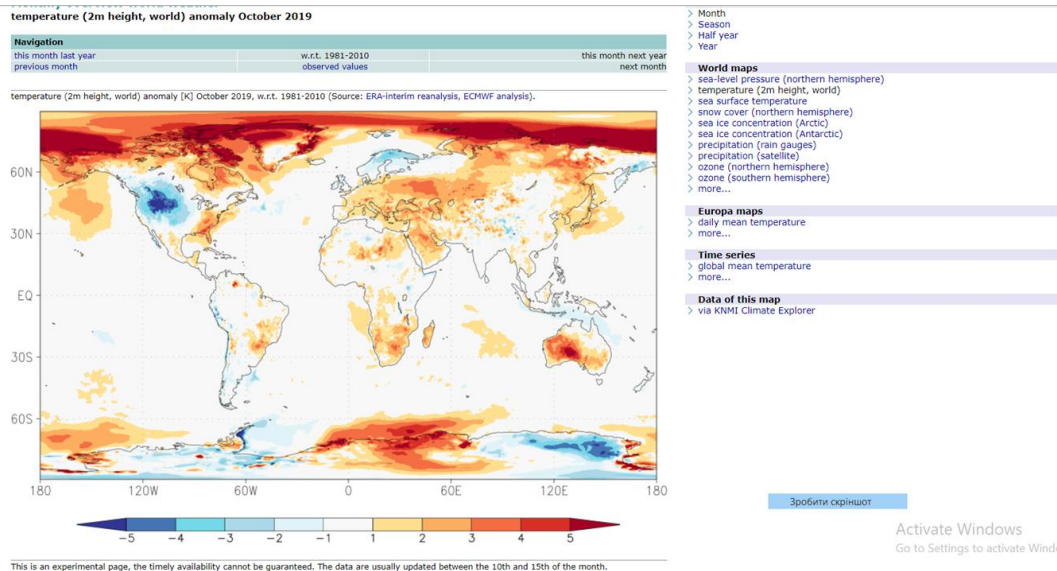


Рис. 1.5. Аналіз кліматичних показників за обраний період в програмі

Застосунок дозволяє здійснювати завантаження інфорграфік високої роздільної здатності для візуалізації кліматичних явищ та створення просторових кліматичних реконструкцій. Зокрема, даний інструмент використаний для дослідження та аналізу даних для створення дендрокліматичного атласу гір Високого Атласу в Марокко [4].

На веб-сайті доступний документ у форматі PDF, що включає покрокову інструкцію для користування інструментом.

NASA Earth Observatory – онлайн-ресурс NASA, який був створений в 1999 році і є важливим джерелом супутникових знімків та іншої наукової інформації про клімат та навколишнє середовище. Фінансування відбувається за рахунок коштів державного бюджету Сполучених Штатів Америки. Інструмент є частиною Наукового бюро проекту “EOS”, розташованого в Центрі космічних польотів імені Годдарда. Тут можна відслідкувати 16 важливих кліматичних показників, таких як опади, температура, рівень вуглекислого газу тощо, які доступні для вибору та навігації на головній сторінці, зображення якої подано на рис. 1.6.

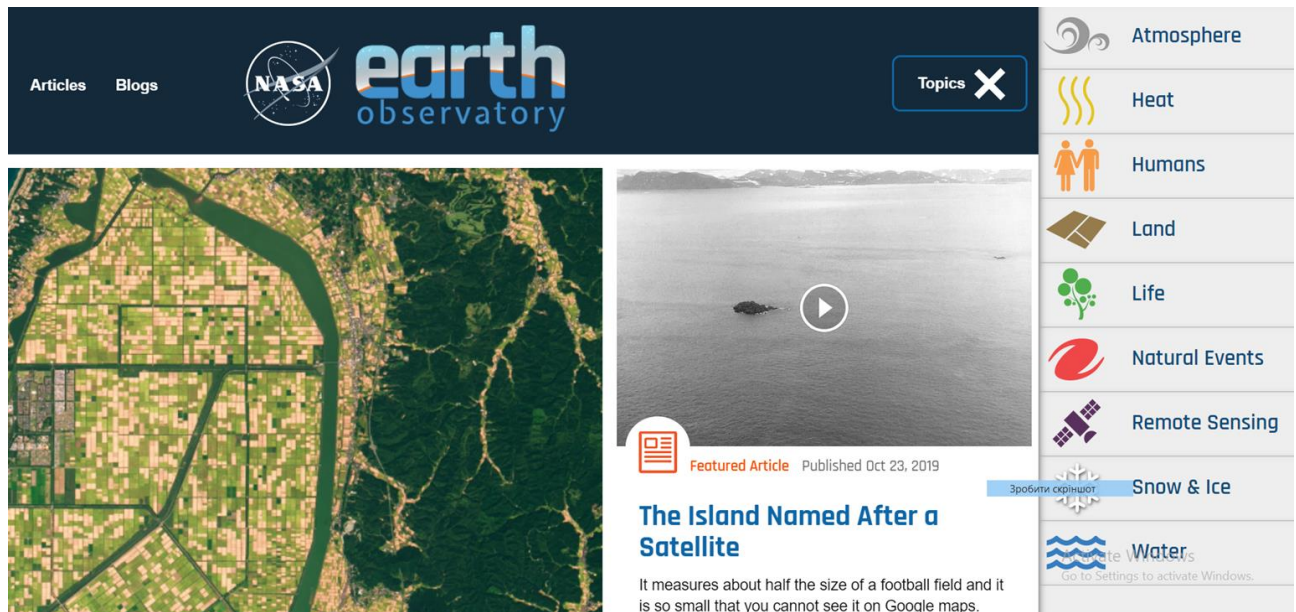


Рис. 1.6. Головна сторінка NASA Earth Observatory

Хоча основна цінність даної системи для аналізу кліматичних та метеорологічних даних полягає у безкоштовному доступі до супутникових знімків земної поверхні та можливості її аналізу, тут також доступні традиційні графіки та діаграми, наприклад, візуалізація температурних аномалій, як зображено на рис. 1.7.

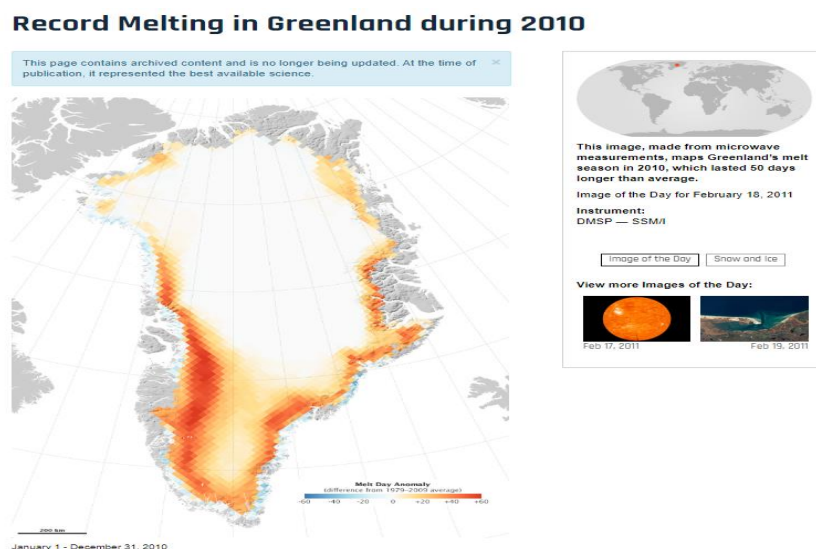


Рис. 1.7. Візуалізація температурних аномалій у Гренландії у 2010 р.

На головній сторінці веб-сайту доступна інструкція для користувача та відеоуроки по використанню інструмента в наукових та навчальних цілях.

Також варто відзначити регулярні оновлення програми останніми даними, що високо оцінюється дослідниками та науковцями.

ESA CCI (European Space Agency Climate Change Initiative) – програма, ініційована та створена ESA для глобального відслідковування змін на основі зібраних за допомогою штучних супутників метеорологічних та кліматичних даних.

CCI Dashboard – забезпечує високий рівень вибору даних у архіві відкритого порталу даних CCI, включаючи ключову інформацію для кожного типу показника, швидке посилання на їх завантаження через FTP, графічне зображення та посилання на відповідну документацію. Зображення інструмента подано на рис. 1.8.

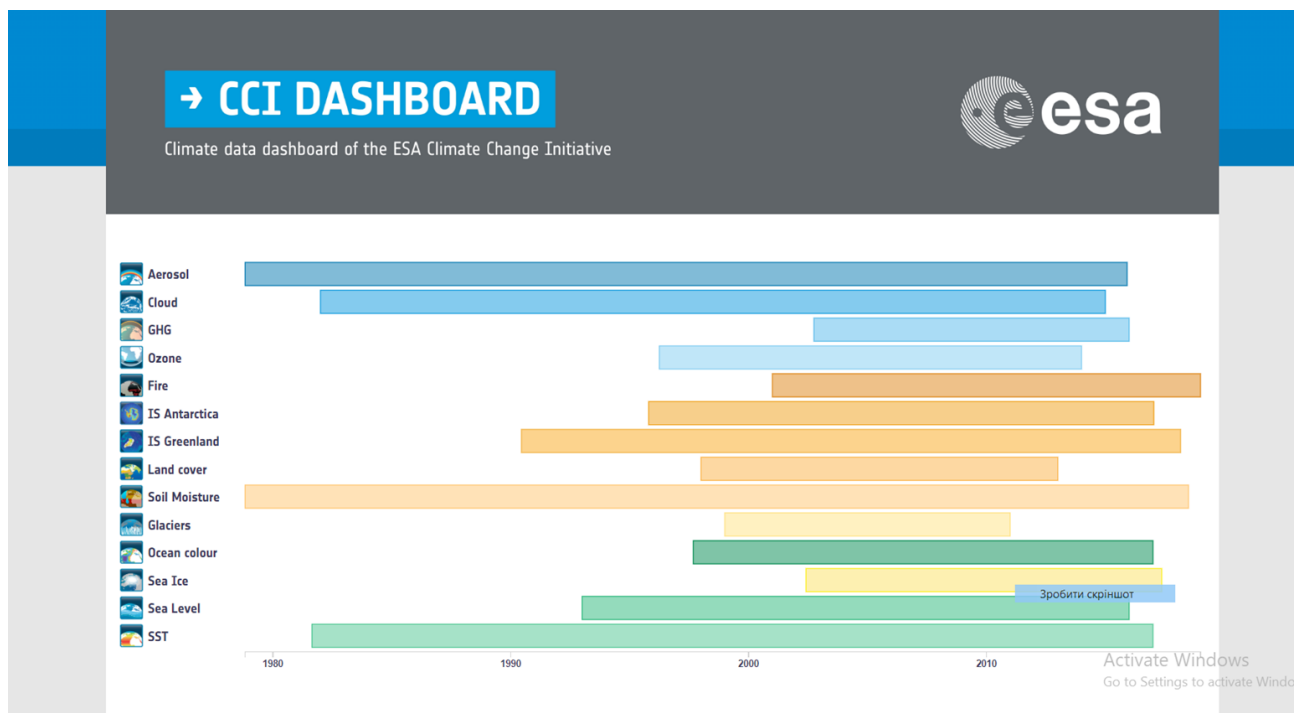


Рис. 1.8. Можливості CCI Dashboard

У 2010 році розпочався перший етап розвитку програми, було запущено 10 пілотних проектів по аналізу кліматичних та метеорологічних даних, обраних членами Консультативної ради з питань кліматичної науки (CSAB) та Консультативного комітету з питань науки про Землю ESA (ESAC). Перші звіти про стан ESA були доступні з початку 2011 року. На даному етапі за

допомогою веб-застосунку досліджується викиди газів в атмосферу, арктичний льодовий щит, хмарність, опади, стан світового океану та озонового шару планети Земля. Для аналізу та візуалізації даних у програмі доступний ряд інструментів.

CCI Data Search використовується для пошуку та завантаження даних з архівів відкритого порталу даних CCI. Інструмент забезпечує пошук тексту та графічних варіантів, що дозволяє користувачам налаштовувати свій пошуковий запит, що продемонстровано на рис. 1.9.

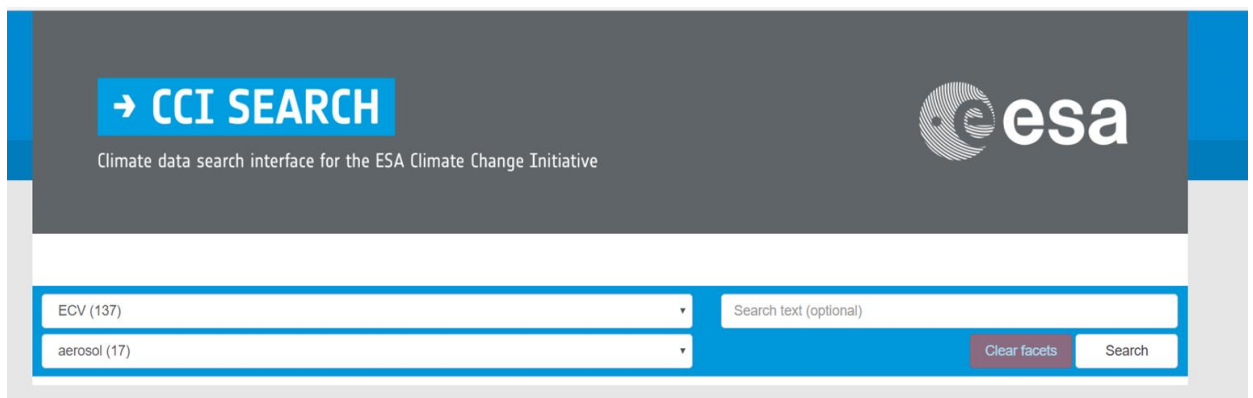


Рис. 1.9. Вигляд інструменту CCI Data Search

Для підтримки користувачів у даній комп'ютерній системі для аналізу кліматичних та метеорологічних даних використовується служба CCI Open Data Portal Helpdesk, що займається консультацією та вирішенням технічних питань.

1.4. Висновки до розділу

У даному розділі розглянуто основні аспекти кліматологічних та метеорологічних досліджень та розглянуто доступні системи для аналізу та візуалізації даних.

Проаналізувавши доступні веб-застосунки для роботи з кліматичними показниками, встановлено, що розглянуті програми працюють, проте мають

ряд недоліків:

- використання застарілих та рідкісних технологій для розробки веб-клієнтів (jQuery v.2.2, HighCharts v.5.5 і т.д), що суттєво ускладнює підтримку, модернізацію та додавання нового функціоналу;
- користувацький інтерфейс, що не відповідає сучасним вимогам UI/UX (User Experience/ User Interface) дизайну, що негативно впливає на доступність та зручність користування програмами;
- повна відсутність або обмежений функціонал для аналізу та візуалізації локальних кліматичних даних;
- незважаючи на наявність інструкцій для користування даними інструментами існує високий поріг входження для використання програми, що серйозно обмежує кількість користувачів;
- обмежені можливості для візуалізації мікрокліматичних та локальних метеорологічних показників.

Також варто відзначити, що незважаючи на те, що досліджуваною проблематикою займаються серйозні науково-дослідні організації (NASA, ESA тощо), існує серйозний дефіцит комп'ютерних систем для аналізу та візуалізації кліматичних даних.

Отже, здійснене у даному розділі дослідження підтверджує актуальність завдань, визначених даною магістерською роботою та високу економічну доцільність створення альтернативної аплікації.

РОЗДІЛ 2

ВИБІР ПРОГРАМНИХ ЗАСОБІВ ТА РОЗРОБКА СЕРВЕРНОЇ ЧАСТИНИ

З метою зручності розробки, підтримки та масштабування аплікації для аналізу та візуалізації було обрано програмні засоби, розглянуті далі.

2.1. Мова програмування Javascript та фреймворк React

JavaScript – об’єктна та прототипно-побудована мова для розробки програмного забезпечення, що в більшості випадків застосовується у веб-розробці [5].

JavaScript відрізняється від інших мов для розробки клієнтської та серверної архітектури, він належить до сторони клієнта, тобто виконується на стороні користувача, а конкретніше ядром браузера операційної системи.

JavaScript зазвичай застосовують як вбудовану мову для доступу до розроблених веб-застосунків. Особливо активно застосовується у браузерах як мова для додавання інтерактивності у роботі веб-сторінок.

Головні архітектурні ознаки: динамічна типізація, об’єкт-прототипна парадигма у програмуванні автоматичне управління пам’яттю, , функціональність або об’єкт-орієнтованість.

Використовуючи синтаксис схожий із Сі , JavaScript має відмінні характеристики:

- доступ до типу об’єкта під час виклику;
- автоматична типізація;
- вбудована можливість збору сміття;
- неназвані функції.

Проте у JavaScript відсутні деякі необхідні можливості:

- створення модулів;
- можливість управління потоками;

- єдині стандарти для інтерфейсів;
- оптимізована система керування пакетами.

Специфікація (документальний опис синтаксису, базових понять та впроваджених алгоритмів) мови для програмування JavaScript називається «ECMA Script» [6].

Ядро JavaScript містить цілий ряд впроваджень, які дають такі можливості, як:

- 1) можливість зберігання інформації у змінних;
- 2) керування мультимедійними та інтерактивними можливостями (робота з відео, зображеннями, анімаціями);
- 3) формувати контент, що буде автоматично оновлюватися ;
- 4) викликати та використовувати певну частину розробленого коду згідно із сценарієм та необхідністю, які існують на сторінці аплікації.

Структура сучасного JavaScript – це так складові:

- основна частина – ECMA Script стандарт,
- документа модель об'єкта – DOM,
- браузерна об'єктна модель – BOM.

JS використовують, головним чином, для програмування на front-end стороні. Якщо розглядати більш детально, то можна з'ясувати, що JS застосовується для керуванням вікном веб-аплікації (браузера), зміни тегів документа при роботі з DOM'ом, опрацювання різноманітних event'ів на браузерній сторінці.

В JavaScript існуючі типи даних поділяються на два види: на примітивні та об'єктні. Нараховується 5 загальновідомих примітивних видів даних: undefined number, boolean, string, null та один відділений тип із назвою object.

JavaScript фреймворки – це набори інструментів та засобів для створення динамічних веб-аплікацій, застосунків для смартфонів або додатків для десктопних пристроїв на мові JavaScript. У загальноприйнятій практиці ці фреймворки використовується для розробки та створення Single-Page

Applications, тобто все, що відбувається на сайті, відбувається на одній сторінці без прямого переходу з неї.

ReactJS – це фреймворк на мові програмування JavaScript, код якого став доступний у 2013 році. Ця бібліотека застосовується для створення та планування різнорозмірних веб-додатків, де дані будуть змінюватися на постійній основі [7].

Переваги ReactJS:

1. Легкий у вивченні. React помітно простіше засвоюється враховуючи простоту та зручність доступного синтаксису. Розробники можуть використати свій досвід створення HTML. Відсутня необхідність у детальному вивченні нової мови програмування TypeScript, як у випадку з фреймворком Angular.
2. Використовуючи з ES6 та ES7 ReactJS демонструє хорошу швидкодію при високих навантаженнях.
3. Хороший рівень адаптивності та гнучкості та максимальна чутливість до потреб користувача.
4. Віртуальна система DOM, що забезпечує можливість працювати з документами формату HTML, XHTML або XML в дереві проекту, оптимально підходить для створення аплікації для аналізу та візуалізації.
5. Достатньо малий об'єм, оскільки дані, що виконуються на стороні клієнтської частини, можуть також бути використовуватися на стороні сервера у той самий момент.
6. Зв'язування даних від більших до дрібніших. Це формує потік інформації, при якому ноди нижчого порядку елементи не будуть впливати на батьківські модулі та компоненти.
7. Міграція між розробницькими версіями, достатньо зручна та проста. Також розробники фреймворку створили інструмент codemods що може автоматизувати значну частину міграцій.

8. Це JavaScript-бібліотека з доступним кодом, що отримує постійні оновлення та покращення відповідно до відгуків девелоперів, що її використовують.

Недоліки ReactJS:

1. Недостатня кількість офіційної документації. Активна та динамічна розробка ReactJS спричинила брак супровідної документації, яка на даний момент не структурована оскільки велика кількість девелоперів вносить в репозиторії персональні зміни без необхідного аналізу та підходу.

2. Необхідний достньо-високий рівень розуміння ReactJS для побудови структури веб-аплікації та процесу інтегрування користувацького інтерфейсу в шаблон MVC.

JSX – це препроцесор, який додає синтаксис XML до JavaScript. Можна використовувати React без JSX, але JSX робить React більш елегантним.

JSX – синтаксис, схожий на XML/HTML, який використовується в React, розширює ECMAScript, так що XML/HTML-подібний текст може співіснувати з кодом JavaScript та фреймворку React. Синтаксис призначений для використання препроцесора (наприклад, Babel), щоб перетворити HTML-подібний текст, знайдений в файлах JavaScript, в стандартні об'єкти JavaScript, які будуть аналізувати ядро JavaScript.

В основному, використовуючи JSX, можна писати стислі структури HTML/XML (наприклад, DOM-подібні деревовидні структури) в тому ж файлі, що і код JavaScript, а потім Babel перетворює ці вирази в код JavaScript. На відміну від минулого, замість того, щоб поміщати JavaScript в HTML, JSX дозволяє поміщати HTML в JavaScript. Можна вбудувати будь-який вираз JavaScript в JSX, обернувши його фігурними дужками.

JSX – це вираз. В результаті компіляції синтаксис JSX трансформується у регулярні об'єкти JavaScript. Це демонструє, що можна застосовувати його для операторів if або for, вказати що він змінна та приймати JSX як аргументи та отримувати з функцій.

Відповідно до типових налаштувань, ReactDOM виключає усі значення, написані в JSX, перед їх рендерингом. Цим гарантується, що не буде можливим виконати те, що явно не написано в коді. Перед рендерингом код приводиться до звичайного рядка. Це допомагає запобігти Cross Site Scripting атаки.

2.2. Мови розмітки та стилів HTML/CSS

HTML (або гіпертекстова мова розмітки) – це синтаксис тегів, який застосовується для розробки, формування та візуалізації веб-застосунків та розташованого в них контенту. Наприклад, інформація може розділятися як таблиця, меню, заголовок та інші частини, в залежності від необхідності її донесення до користувача.

HTML, з допомогою використання таблиць стилізації та програмними функціями це головні засоби побудови веб-додатків.

Він дозволяє:

- створювати інтерактивні форми;
- створювати формалізований та структурний компонент методом позначення особливостей контенту або тексту (заголовки, меню, таблиці)
- отримувати інформацію із Internet через різноманітні гіперпосилання;
- вбудовувати зображення, звук, відео та інші об'єкти у текст.

CSS (каскадна таблиця стилізації) – спеціальна мова, якою стилізуються веб-аплікації, розроблені мовами розмітки веб-сторінок.

Найчастіше CSS використовується для документів, котрі розмічені мовою HTML, XHTML та XML. Специфікації CSS створено Консорціумом Всесвітньої мережі, котрий їх розвиває.

На сучасному етапі HTML в без стилів містить доволі малий вибір інструментів, що не завжди дає можливість вирішувати завдання та

відповідати теперішнім вимогам до дизайну веб-застосунків. Для вирішення цих проблем використовується CSS, що може вирішити головні завдання, що мають стосунок до стильового оформлення сторінки.

Таке розділення має на меті покращити сприйняття та зручність користування контентом, прибрати повтори, забезпечити управління та спостереження за відображенням коду в різних станах, більше структурувати та спростити контент тощо. CSS дозволяє пристосувати сайт для різноманітних умов (монітори, планшети, смартфони і т.д.).

Переваги використання CSS:

- відображати різні дизайни для одного сайту;
- кешування стилів та мінімізація часу завантаження застосунку;
- легкість оновлення дизайну;
- дозволяє створювати досконалу верстку дизайну;
- покращення зручності користування;
- збільшення структурування та впорядкованості сторінок

Сучасні браузеры дозволяють швидко опрацьовувати стилі що покращує UI-дизайн веб-аплікацій.

CSS-фреймворки – це потужні інструменти, які можуть допомогти вдосконалити процеси розробки та проектування.

Bootstrap, який спочатку називався Twitter Blueprint, був створений Марком Отто і Джейкобом Торнтоном і його випуск датовано 19 серпня 2011 року. Це зовнішній інтерфейс з доступним вихідним кодом, що складається з HTML, CSS і JavaScript. Він має модульну структуру та використовує препроцесор SassCSS. Також варто виділити, що він містить не лише CSS, але й JavaScript-фреймворк. У Bootstrap написано готові стилі та скрипти, для них розробнику достатньо тільки створити потрібні стильні класи та атрибути HTML-елемента.

Фреймворк Bootstrap підтримує останні стабільні версії основних браузерів і платформ. Що стосується Windows, то підтримуються Internet Explorer 9-11 і Microsoft Edge, але не IE8.

2.3. Мова програмування C#

Мова для розробки (програмування) C# (“C Sharp”) – структурована, широкоживана, статично типізована, об’єктно-орієнтована мова написання веб-частини програми, синтаксис якої запроваджено девелоперами із Microsoft. C# належить до родини мов розробки та програмування C, процес написання коду та синтаксис доволі схожий C, C++, Java та іншими мовами.

C# поєднав в своєму розробницькому середовищі найкращі сторони засобів-попередників. Зокрема C, C++, Object Pascal, врахувавши реальний досвід використання та підготовки програм. Відомі проблемовмісні моделі, що до цього були застосовані у мовах для програмування, а саме, множинне унаслідування класів (яке активно використовується у мові C), були спеціально виключені.

Вивчивши характеристики C#, можна зробити висновки, що ця мова подібна на Java-синтаксис, це зображено в синтаксі та головних поняттях та основах цих мов розробки.

Як і в бекенд мову Java, у досліджуваній C# розробники додали такі концепції:

- віртуальна машина – динамічний фреймворк або інша основа, наприклад .Net, запускає та виконує розроблену програму ;

- байт-код – підготовлений код компілюється в тимчасову мову MSIL («Microsoft Intermediate Language»), а лише згодом трансформується в машинну мову в повній залежності від платформ, на яких буде виконуватися програма;

- керований код – враховуючи те, що розроблені програми, виконуються лише у віртуальному середовищі CLR або «Common Language Runtime», існує можливість для керування та контролювання за виконання програм та доступні механізми для їх зупинки та вимкнення, а також регулювати завантаженість та використання пам’яті розробленим програмним

забезпеченням, за потребою – масштабувати чи видаляти обрані ділянки пам'яті, які використовуються програмами.

Мова C # побудована з використанням компонентної архітектури та забезпечує захист для написаного коду.

До основних особливостей мови можна віднести такі:

- підхід орієнтований на розділення на компоненти;
- делегати
- інкапсуляція даних, що характерно для ООП;
- індексатори для доступу до компонентів;
- оператори;

Також мова C# унаслідувала від C ідею блоків програм.

Мову C# вважають об'єктно-орієнтованою мовою

Мова програмування C# є “рідною” для розробки застосунків в середовищі фреймворка. NET, оскільки тісно та ефективно інтегрована з ним.

2.4. Розробка серверної частини та створення бази даних для програми

Метою даної роботи є створення системи для аналізу та візуалізації даних. З кінця 2010-х років чітко прослідковується тенденція на ріст популярності підходу в розробці клієнт-серверних застосунків, коли основна логіка системи аналізу зосереджується на стороні front-end, тоді як back-end відповідає за збереження інформації та видачу фактично структурованих, але не аналізованих даних.

При створенні бекенд частини веб-сайту першим етапом є інсталяція необхідного програмного забезпечення та середовищ для розробки:

Visual Studio 2017 (з вибраним при установці .NET Core cross platforms development). Сервер ми будемо писати на .net core 2.0, ось чому нам потрібна версія середовища розробки ≥ 2017 , оскільки в попередніх версія максимальна підтримувана версія – 1.1 [8].

Після цього встановлюємо SDK (Software Development Kit) .NET, як зображено на рис. 2.1.

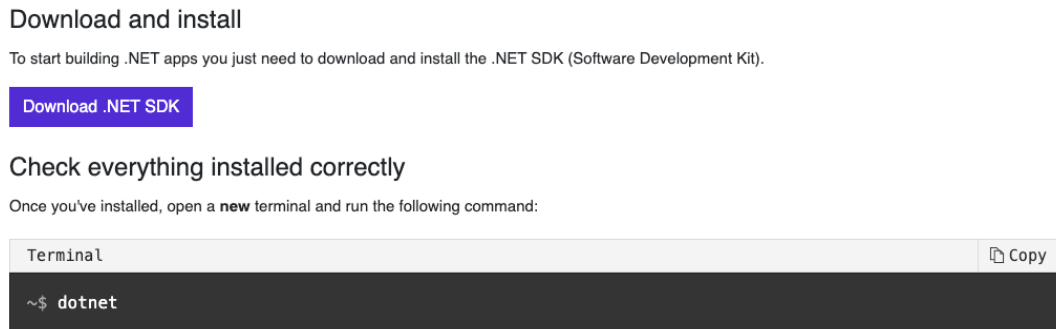


Рис. 2.1. Встановлення SDK .NET

Фреймворк Microsoft .NET Framework складається із великої кількості інструментів та технологій, для розробки бекенд-частини програмного забезпечення. В основі платформи Microsoft .NET Framework лежить мова програмування C# [8].

Далі встановлюємо MS SQL Server для розгортання бази даних (рис. 2.2).



Рис. 2.2. Встановлення SQL Server

Microsoft SQL Server – це система аплікація для управління базами даних, яка розроблена та дистрибутиця корпорацією Microsoft. Для створення запитів використовується – Transact-SQL. Її можна використовувати для

різноманітних, (малих, середніх та великих), тому повністю задовольняє вимоги до бази даних дня нашої аплікації.

Відкриваємо Visual Studio 2017 та обираємо File далі New пістя цього Project. Після обрання проекту відкриється діалогове вікно New Project. Обираємо .NET Core у Visual C# меню в лівій панелі.

Після цього обираємо «ASP.NET Core Web Application» з доступних типів проектів, як показано на рис. 2.3.

Наступний етап – вибір шаблону, по якому було створино проект. З випадаючого меню обираємо ASP.NET Core версії 2.0 та обираємо React.js як основний шаблон. Далі додаємо моделі даних на основі відкритих безкоштовних кліматичних даних та моделі для логування користувачів. Після цього у Visual Studio відкриємо Package Manager Console та введемо команду: enable-migrations.

Після виконання цієї команди в проекті буде створена папка Migrations, в якій розміщений файл Configuration.cs. Цей файл містить оголошення однойменного класу Configuration, який встановлює налаштування конфігурації міграції.

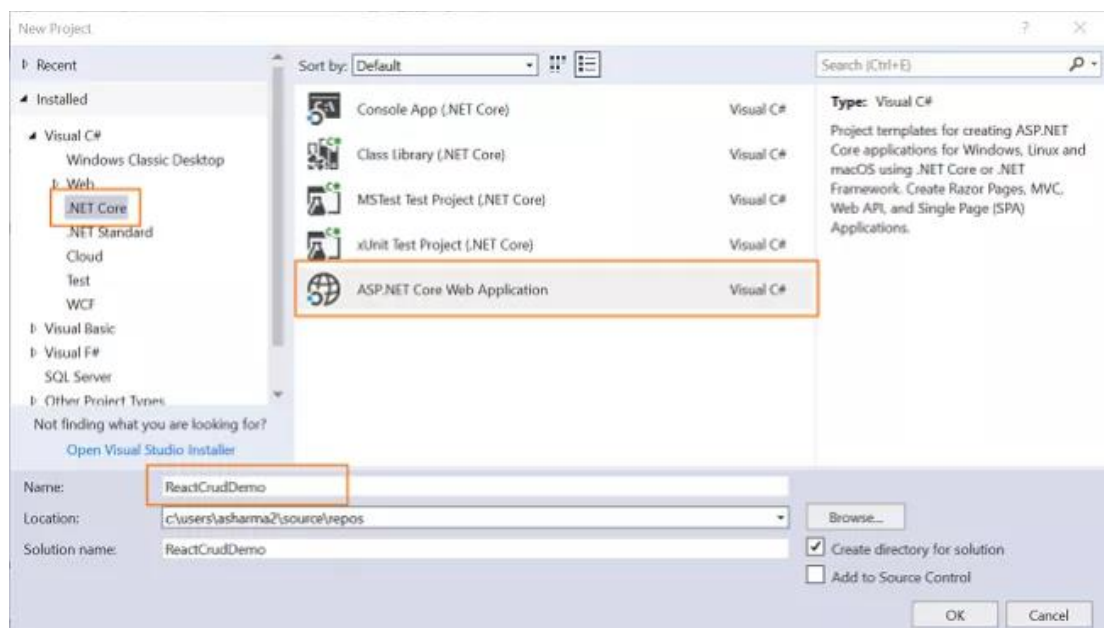


Рис. 2.3. Створення ASP .NET Web Applications

В методі Seed ініціалізуємо базу даних початковими даними. Тепер нам потрібно створити саму міграцію.

Для цього в консолі Package Manager Console вводимо команду: PM, після неї команду Add-Migration та MigrateDB.

Після цього Visual Studio автоматично згенерує клас міграції. В методі Up за допомогою виклику методу CreateTable створюються таблиця "dbo.Users" та відбувається її налаштування: створення стовпців, установка ключів.

Метод Down видаляє стовпці та таблицю на випадок, якщо вони існують. Щоб виконати міграцію, застосуємо цей клас, набравши в консолі команду: PM, після якої Update-Database. Після цього отримано структуру бази даних, зображену на рис. 2.4.

Наступним етапом розробки бекенд частини системи для аналізу та візуалізації кліматичних даних є додавання Web API Controller до нашої аплікації. Для цього відкриваємо папку Controllers та виконуємо наступну послідовність дій: Add та додати New Item.

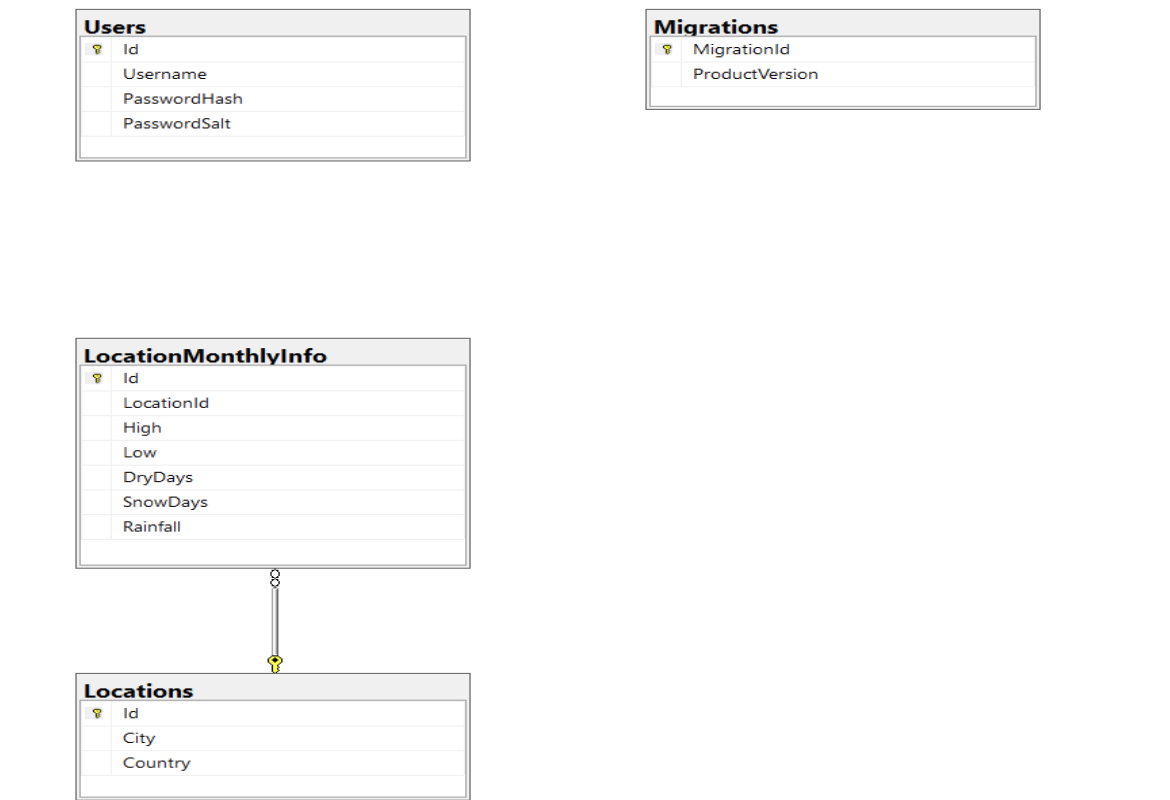


Рис. 2.4. Структура бази даних

Обираємо ASP.NET з меню лівої панелі, після чого обираємо API Controller Class і даємо йому назву UsserController, як показано на рис. 2.5.

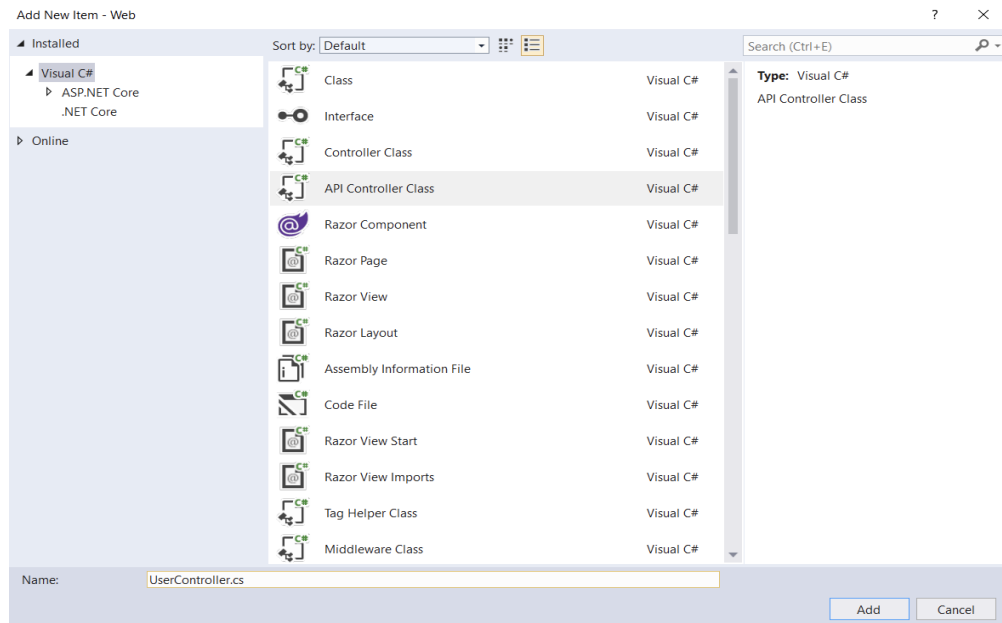


Рис. 2.5. Створення контролера UsserController

Даний контролер буде мати метод POST для реєстрації користувача.

В контролері AuthController.cs додаємо логіку логування користувачів в нашій веб-аплікації.

Також створюємо контролер для отримання наборів даних для аналізу та візуалізації даних, де додаємо два методи: перший [HttpGet] для отримання всього сету даних (по всіх доступних локаціях) та другий [HttpGet], що приймає параметр query string з LOCATION ID обраного географічного об'єкта з вкладеними місячними наборами даних (рис. 2.6).


```

1  using System.Collections.Generic;
2  using System.Threading.Tasks;
3  using Business.Dtos;
4  using Business.Services;
5  using Microsoft.AspNetCore.Mvc;
6
7  namespace ClimateAnalyticsApp.Controllers
8  {
9      [ApiController]
10     [Route(template: "[controller]")]
11     public class LocationController : ControllerBase
12     {
13         private readonly ILocationService locationService;
14
15         public LocationController(ILocationService locationService)
16         {
17             this.locationService = locationService;
18         }
19
20         [HttpGet]
21         public Task<List<LocationDtos>> Get()
22         {
23             return locationService.Get();
24         }
25
26         [HttpGet]
27         public Task<LocationDtos> GetItems([FromQuery]int id)
28         {
29             return locationService.GetItems(id);
30         }
31     }
32 }

```

Рис. 2.6. API методи для отримання даних

На наступній ітерації розробки програми при підготовці клієнтської частини необхідно створити JavaScript-файл з назвою Service.js, за допомогою способу fetch(). Fetch API – це інтерфейс JavaScript для роботи із HTTP запитами та відповідями. Метод fetch(), дозволяє легко і логічно асинхронно отримувати ресурси по мережі.

Подібна функціональність раніше досягається за допомогою XMLHttpRequest, проте fetch() – це набагато потужніший інструмент. fetch() повертає об'єкт Promise, який при отриманні позитвної відповіді від серверної частини виконує задану функцію, в якій передається об'єкт response як аргумент. Для отримання усього набору даних відповідно до локацій потрібно додати відповідний метод, як зображено на рис. 2.7.

```

getRequest = () => {
  let url = 'location/get';
  fetch(url).then(response => response.json()).then((repos) => {
    this.setState( state: {
      repos: repos
    });
  });
};

```

Рис. 2.7. Отримання даних за допомогою методу fetch

2.5. Висновки до розділу

У даному розділі проаналізовано мови програмування, які використовуються у процесі розробці програми для аналізу та візуалізації даних, та було обрано сучасні засоби розробки: Microsoft SQL Server для розгортання бази даних, Microsoft .NET фреймворк з мовою розробки C# для серверної частини аплікації, REST API для комунікації між серверною і клієнтською частинами застосунку. Для написання клієнтської частини веб-сайту застосовано фреймворк ReactJS з мовою програмування JavaScript. Для візуального оформлення веб-клієнта застосуємо CSS та фреймворк React Bootstrap.

Вказані вище технології у підсумку дадуть можливість створити ефективний швидкодіючий інструмент для візуалізації наборів даних для системи аналізу та візуалізації кліматичних показників, що забезпечує досягнення поставлених у роботі цілей.

А також у даному розділі розроблено серверну частину комп'ютерної системи для аналізу та візуалізації кліматичних даних. Досліджено використання MS SQL Server 2017 для розгортання бази даних, за допомогою мови програмування C# та фреймворка .NET створено бекенд частину експериментальної аплікації, зокрема, розроблено сервіси та контролери для

реалізації функціоналу логування користувачів, вивчено та реалізовано REST API для взаємодії з клієнтською частиною програми.

Усе вищесказане забезпечило хороше масштабування та взаємодію компонентів системи, спільність програмних інтерфейсів, що в результаті покращить швидкодію системи та її безпеку.

РОЗДІЛ 3

РОЗРОБКА КЛІЄНТСЬКОЇ ЧАСТИНИ СИСТЕМИ АНАЛІЗУ ТА ВІЗУАЛІЗАЦІЇ КЛІМАТИЧНИХ ДАНИХ

3.1. Розробка клієнтської частини аплікації

Для фронтенд розробки нам потрібно встановити Node версії ≥ 8.10 та npm версії ≥ 5.6 , як показано на рис. 3.1.

Завантаження

Поточна версія: 12.13.0 (Містить npm 6.12.0)

Завантажте початковий код Node.js або інсталятор для вашої платформи та почніть розробку сьогодні.

LTS	Поточна	
Рекомендовано для більшості	Найновіші можливості	
 Інсталятор для Windows <small>node-v12.13.0-x86.msi</small>	 Інсталятор для macOS <small>node-v12.13.0.pkg</small>	 Вихідний код <small>node-v12.13.0.tar.gz</small>
Інсталятор для Windows (.msi)	32-bit	64-bit
Бінарний файл для Windows (.zip)	32-bit	64-bit
Інсталятор для macOS (.pkg)	64-bit	
Бінарний файл для macOS (.tar.gz)	64-bit	
Бінарні файли для Linux (x64)	64-bit	
Бінарні файли для Linux (ARM)	ARMv7	ARMv8
Вихідний код	node-v12.13.0.tar.gz	

Рис. 3.1. Встановлення Node.js

Після здійснення вищевказаної установки в обраній папці викликаємо Terminal нашої операційної системи. Для створення проекту виконуємо ініціалізаційні команди (рис. 3.2).

```
npx create-react-app my-app
cd my-app
npm start
```

Рис. 3.2. Команди для створення проекту

Додано основу клієнтської частини веб-застосунку, створену за допомогою Javascript фреймворка React, проте цей клієнт не опрацьовує логіку бекенду та баз даних, а лише надає команди для побудови фронтенду, тому його буде використано зі створеним раніше бекендом [8].

При вивченні створеного автоматично згенерованого файлу `package.json`, нам буде доступна структура пакетів програми, як показано на рис. 3.3.

```
1  {
2    "name": "my-app",
3    "version": "0.1.0",
4    "private": true,
5    "dependencies": {
6      "react": "^16.11.0",
7      "react-dom": "^16.11.0",
8      "react-scripts": "0.9.x"
9    },
10   "devDependencies": {},
11   "scripts": {
12     "start": "react-scripts start",
13     "build": "react-scripts build",
14     "test": "react-scripts test --env=jsdom",
15     "eject": "react-scripts eject"
16   }
17 }
```

Рис. 3.3. Структура файлу `package.json`

В цьому JSON файлі в об’єкті “dependencies” вказані залежності, які необхідні для функціонування нашого застосунку та їх версії.

Об’єкт “scripts” – це оголошені наші основні прт-команди, необхідні в процесі розробки веб-сайту.

Також в нашій IDEA отримана структура проекту вказана на рис. 3.4.

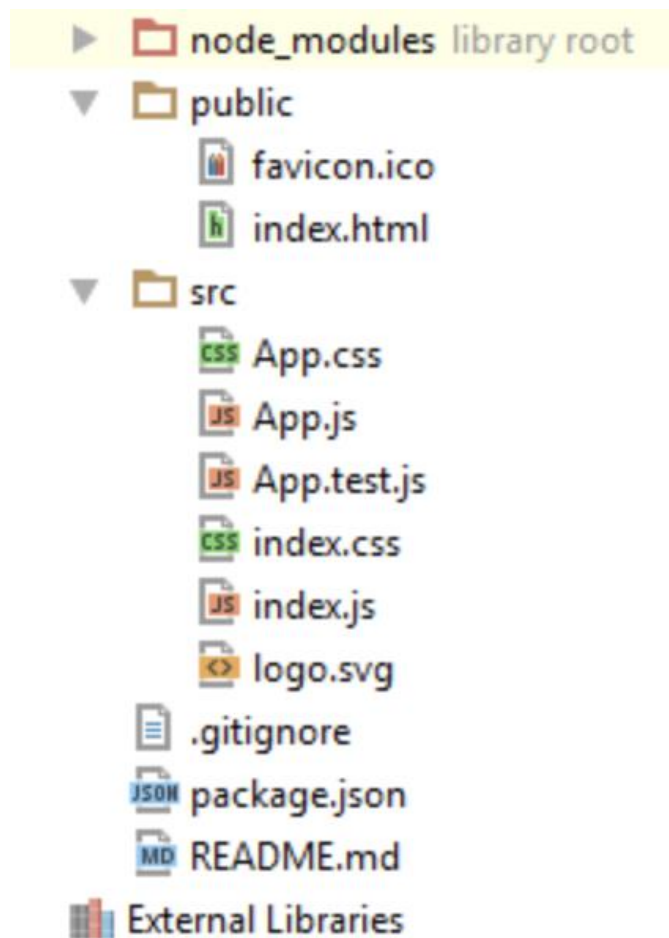


Рис. 3.4. Структура проекту

`node_modules/` – в цій папці розміщені всі залежності проекту, вказані в файлі `package.json` і які встановлюються при команді `npm install`.

`public/` – вміст цієї папки – це те, що потрібно буде відрендерити на сторінку веб-застосунку: `public/index.html` – шаблон веб-сайту і `favicon.ico` – фавіконка, яка буде відображатись на вкладці браузера.

`src/` – тут розміщені всі вихідні файли, з якими буде відбуватися безпосередньо взаємодія та зміни.

`src/index.js` – це вхідна точка створеного проекту.

Після ініціалізації проекту цей файл матиме наступний вигляд, що відображено на рис. 3.5.

```
1  import React from 'react';
2    import ReactDOM from 'react-dom';
3    import App from './App';
4  import './index.css';
5
6    ReactDOM.render(
7      <App />,
8      document.getElementById( elementId: 'root' )
9    );
```

Рис. 3.5. Вигляд файлу після ініціалізації

`import` модуль `/React/` `from 'react'` та `import /ReactDOM/` модуль `from 'react-dom'` – це підключення модулів `React` та `ReactDOM` із встановлених раніше `node`-модулів.

Далі підключається компонент `App.js`, використовується `import App` (модуль) `from './App'`.

Стрічка `import './index.css'` підключає `CSS`-файл з базовими стилями проекту.

Частина коду:

```
«ReactDOM.render(
  <App />,
  document.getElementById('root')
);»
```

містить декілька частин:

`render()` – метод модуля `ReactDOM`, який приймає два аргументи: компонент, який потрібно відрендерити, та `HTML` “обгортку”, в яку необхідно розмістити результат. В нашому проекті елемент з `id=“root”`, який був зазначений в `public/index.html`.

`<App />` – це компонент `App.js`, оголошення при допомозі **JSX** – (або JavaScript Syntaxes Extensions) (розширення синтаксису JavaScript). Це HTML-подібний синтаксис для створення реакт-компонентів.

Синтаксис його схожий на звичайний HTML, проте є декілька відмінностей, які було враховано в роботі над проектом:

- 1 компонент – це 1 нода, тобто вся розмітка компоненту повинна бути розміщена в межах одного тегу;
- непарні теги в JSX завжди потрібно закривати;
- замість атрибута `class` в JSX використовується `className` [8].

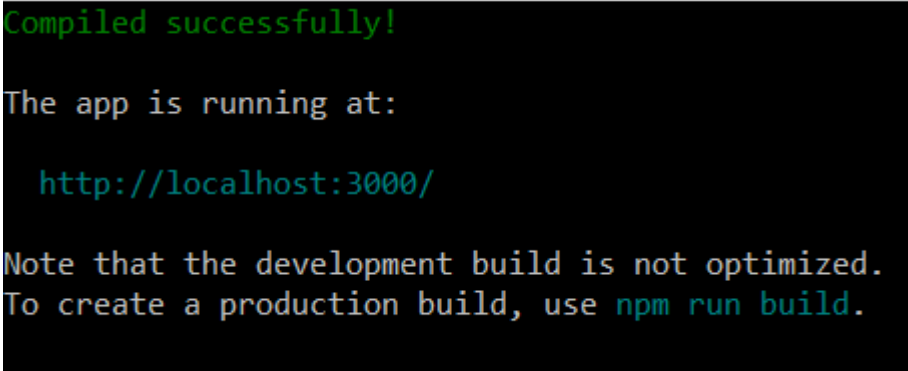
Для перевірки коректності установки базових пакетів та налаштування нашої системи для аналізу та візуалізації даних в папці проекту в терміналі виконуємо команду, як показано на рис. 3.6.



```
npm start
```

Рис. 3.6. Команда запуску аплікації

В результаті чого отримуємо наступний вивід (рис. 3.7):



```
Compiled successfully!  
The app is running at:  
  
  http://localhost:3000/  
  
Note that the development build is not optimized.  
To create a production build, use npm run build.
```

Рис. 3.7. Повідомлення про успішний старт програми

Також відбудеться запуск проекту у браузері за адресою <http://localhost:3000>.

В середовищі для розробки працює livereload, тобто всі зміни відображаються на сторінці без її перезагрузки.

Наступним етапом розробки клієнтської частини системи аналізу та візуалізації кліматичних даних є створення JavaScript-компонентів та CSS-файлів стилізації відповідно до задач, поставлених перед нашим веб-додатком (рис. 3.8).

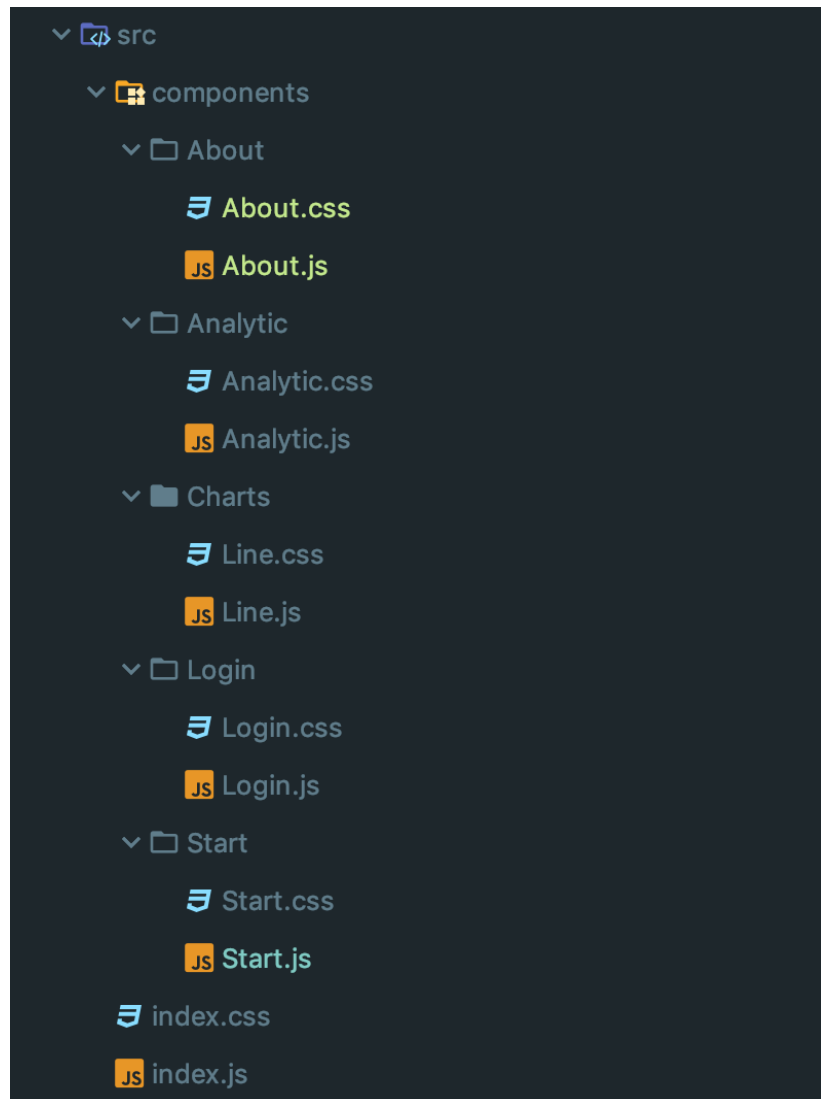


Рис. 3.8. Створені компоненти

У папці Start розміщений React-компонент, який відобразитиметься на головній сторінці нашого сайту. Компонент Login відповідатиме за клієнтську частину процесу логування користувачів у програмі. У файлах Analytic зосереджується основна частина логічних операцій для аналізу та візуалізації

кліматичних даних. Папка Charts містить код для кастомізації та налаштування обраних графіків для відображення опрацьованих масивів даних. About – це компонент, у якому буде розміщено форму зворотнього зв'язку користувачів програми з розробниками та загальна інформація про автора веб-додатку.

Наступним етапом розробки системи аналізу та візуалізації кліматичних даних є підключення модуля React Bootstrap версії 4.3. Bootstrap – це фреймворк для стилізації динамічних веб-додатків. Для цього у терміналі виконуємо наступну команду для додавання модуля до проекту, як показано на рис. 3.9.

```
npm install react-bootstrap bootstrap
```

Рис. 3.9. Встановлення модуля React Bootstrap

Наступним кроком додаємо стрічку `import 'bootstrap/source/dists/css/bootstrap.min.scss'` в файл `src/index.js` для того, щоб функціонал підключеного клієнтського фреймворку був доступний у нашому проєкті. Після цього додаємо зовнішні стилі від React Bootstrap, як видно з рис. 3.10.

```
<link
  rel="stylesheet"
  href="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
  integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQU0hcWr7x9JvoRxT2MZw1T"
  crossorigin="anonymous"
/>
```

Рис. 3.10. Додавання зовнішніх стилів

Найкращий варіант для того, щоб на проєкті були підключені найновіші версії стилів, використати як джерело ресурсних файлів CDN (Content delivery network), яке отримує регулярні офіційні оновлення від розробників.

Після завершення даного етапу як результат було отримано локально розгорнутий базовий проект за веб-адресою <http://localhost:3000> зі встановленими залежностями для подальшої розробки, зокрема фреймворк React версії 16.10.2 та React Bootstrap версії 4.3.

Наступною стадією розробки веб-застосунку для аналізу та візуалізації кліматичних даних є верстка клієнтських компонентів та написання логіки аналізу для отриманих з бекенду наборів даних.

При верстці стартової сторінки використовуємо підключений раніше клієнтський фреймворк Bootstrap, зокрема імпортуємо компонент кнопки – `import` компонент `Button` із модуля `'react-bootstrap/Button'` і за допомогою JSX синтаксису додаємо її на сторінку та стилізуємо цю частину веб-сайту за допомогою CSS у відповідному файлі.

Далі створюємо веб-розмітку для нашої форми логування користувачів програми, як зображено на рис. 3.11.

```
render(){
  return (
    <div className="login-form-wrapper">
      <div className="login-form">
        <Form onSubmit={this.handleLogin}>
          <Form.Group controlId="formBasicEmail">
            <Form.Label>Email address</Form.Label>
            <Form.Control type="email" placeholder="Enter email"/>
            <Form.Text className="text-muted">
              We'll never share your email with anyone else.
            </Form.Text>
          </Form.Group>

          <Form.Group controlId="formBasicPassword">
            <Form.Label>Password</Form.Label>
            <Form.Control type="password" placeholder="Password"/>
          </Form.Group>

          <Form.Group controlId="formBasicCheckbox">
            <Form.Check type="checkbox" label="Check me out"/>
          </Form.Group>
          <Button variant="primary" type="submit">
            Submit
          </Button>
        </Form>
      </div>
    </div>
  );
}
```

Рис. 3.11. Створена веб-розмітка

Як помітно у наведеному вище рисунку крім базових тегів також імпортовані компоненти Bootstrap - `import {Button, Form} from "react-bootstrap"`.

У компоненті `Analytic.js UI (user interface)` частина клієнтської програми буде представлена елементами, які дають змогу користувачам взаємодіяти з логікою веб-сайту, зокрема `select` (рис. 3.12).

```
<select name="" id="" onChange={this.change} value={value.id}>
  {this.citiesData.forEach(item =>
    <option key={item.id} value={item.id}>{item.city}, {item.country}</option>
  )}
</select>
```

Рис. 3.12. Елемент `select`

Для вибору конкретного набору інформації для аналізу та відображення використаємо також елементи `Switcher` та `Checkbox`, які дадуть нам змогу відобразити інформацію після фільтрування та сортування за обраними показниками.

Завершальним компонентом нашої системи аналізу та візуалізації кліматичних даних є створення форми зворотного зв'язку у компоненті `About`, яка відтворена на рис. 3.13.

Write to us

Your name

Your email

Subject

Your message


SEND 

Рис. 3.13. Форма зворотного зв'язку

Отже, на даному етапі розробки програми ми виконали верстку основних компонентів клієнтської частини програми та виконали поставлені перед собою завдання, отримавши зручний та зрозумілий користувацький інтерфейс.

На наступному етапі розробки нашої системи аналізу та візуалізації кліматичних даних ми додаємо логічну частину операцій, які повинні будуть здійснюватися на клієнтській частині веб-застосунку.

У фреймворку React існує два підходи оголошення компонентів – функціональний та класовий. У нашому проекті ми використовуємо класовий метод оголошення, тому що він базується на найновішій версії мови JavaScript EcmaScript 6, що нам дозволить мати розширені можливості при написанні коду.

Для реалізації динамічної взаємодії з користувачем спочатку додаємо можливість навігації по нашому веб-сайту. Для цього використовуємо модуль react-router-dom. Після його додавання у файл package.json виконуємо команду npm install та імпортуємо його у наш головний файл index.js та налаштуємо роутер відповідно до структури нашого проекту (рис. 3.14).

```
import { Route, BrowserRouter as Router } from 'react-router-dom'  
  
const routing = (  
  <Router>  
    <div>  
      <Route path="/main" component={Start} />  
      <Route path="/analytics" component={Analytics} />  
      <Route path="/login" component={Login} />  
      <Route path="/about" component={About} />  
    </div>  
  </Router>  
)  
);
```

Рис. 3.14. Роутінг проекту

Далі стартовій сторінці на кнопку “SIGN IN” додаємо метод `handleClick` для переходу на сторінку логування.

На сторінці для логування було додано наступні методи:

`handleEmailChange` – для обробки введених даних в текстовому полі електронної пошти користувача;

`handlePasswordChange` – відповідає за збір та обробку введеного паролю;

`handleLogin` – формує об’єкт із зібраних даних та відправляє їх на сервер та в разі відповіді про успішне логування перенаправляє користувача на сторінку, де він може працювати з аналітикою.

Наступний етап – додавання методів на компонент з аналітикою. Для аналізу отриманих із сервера кліматичних даних, які надходять нам у вигляді масивів, використовуємо доступні в JavaScript методи:

метод `arr.find` підійде нам для пошуку інформації за конкретним унікальним ідентифікатором, його синтаксис наступний:

```
let result = arr.find(function(item, index, array) {  
  // якщо true – повертає поточний елемент масиву та зупиняє перебір  
  // якщо на всіх ітераціях результат false, повертає undefined  
});
```

Метод `find` шукає один (перший зустрічний) елемент, на якому функція-колбек поверне `true`.

На той випадок, якщо знайдених елементів може бути більше одного, ми використаємо метод `arr.filter(fn)` (фільтрування за певним обраним показником).

Синтаксис цього методу схожий з `find`, але `filter` повертає масив із всіх елементів, які задовільняють вказану у методі умову.

Для ітераційного перебору та опрацювання кожного наступного елемента масиву кліматичних історичних даних ми використовуємо методи доступні у обраній мові програмування:

- `forEach` – цей JavaScript-метод «`arr.forEach(callback() [current, thisArg])`» застосовується для поступового опрацювання масиву. Цей метод для кожного ітерованого елемента масиву буде викликати передану функцію `callback`.

- `map` – метод «`arr.map(callback() [current, thisArg])`» використовується для трансформації масиву. Він утворює новий екземпляр масиву, який буде складатися із результатів виклику `callback`-функції з аргументами `item, i` та `array`) для кожного елемента масиву кліматичних даних.

- `every` – засіб для роботи з масивом «`arr.every(callback() [current, thisArg])`» видає результат `true`, якщо використання `callback`-функції поверне `true` для кожного елемента нашого масиву даних.

- `some` – вказний метод використовує синтаксис «`arr.some(callback[current, thisArg])`», що результатом поверне `true`, якщо застосування `callback` -функції поверне `true` для хоча б одного елемента досліджуваного масиву.

- `reduce` – функція «`arr.reduce(callback[, initialValue])`» може бути використана для поступового опрацювання всіх елементів масиву із запам'ятовуванням тимчасового розрахункового результату. Вона приймає `callback`-функцію способом ітерації до кожної одиниці масиву, зберігаючи в процесі тимчасові результати обчислення. Функція-`callback` включає такі аргументи: `previousValue`, `current`, `indexElement` та власне опрацьований масив. Крім `callback` розглянутому методу можна передавати аргумент `initialValues` [5], так як це зображено на рис. 3.15.

```

getSnowDaysPerYear() {
  return this.selectedData.monthlyAvg.reduce(function (sum, current) {
    return sum + current.snowDays;
  }, 0);
}

getDryDaysPerYear() {
  return this.selectedData.monthlyAvg.reduce(function (sum, current) {
    return sum + current.dryDays;
  }, 0);
}

getRainfallPerYear() {
  return this.selectedData.monthlyAvg.reduce(function (sum, current) {
    return sum + current.rainfall;
  }, 0);
}

```

Рис. 3.15. Обробка даних за допомогою javascript-методів роботи з масивами

Отже, було отримано веб-застосунок в якому користувач після здійснення процесу ідентифікації отримує доступ до інструментів аналізу історичних кліматичних даних та отримати вивід у вигляді текстової інформації. Тому наступним етапом розробки є додавання можливості візуалізації.

3.2. Підключення ApexCharts та створення діаграм для візуалізації даних

Для візуалізації даних нашої системи аналізу кліматичних даних використаємо ApexCharts – це сучасна JavaScript бібліотека з відкритим вихідним кодом для побудови графічного образу даних на основі SVG (Scalable Vector Graphics) графіки, головну сторінку веб-сайту подано на рис. 3.16.

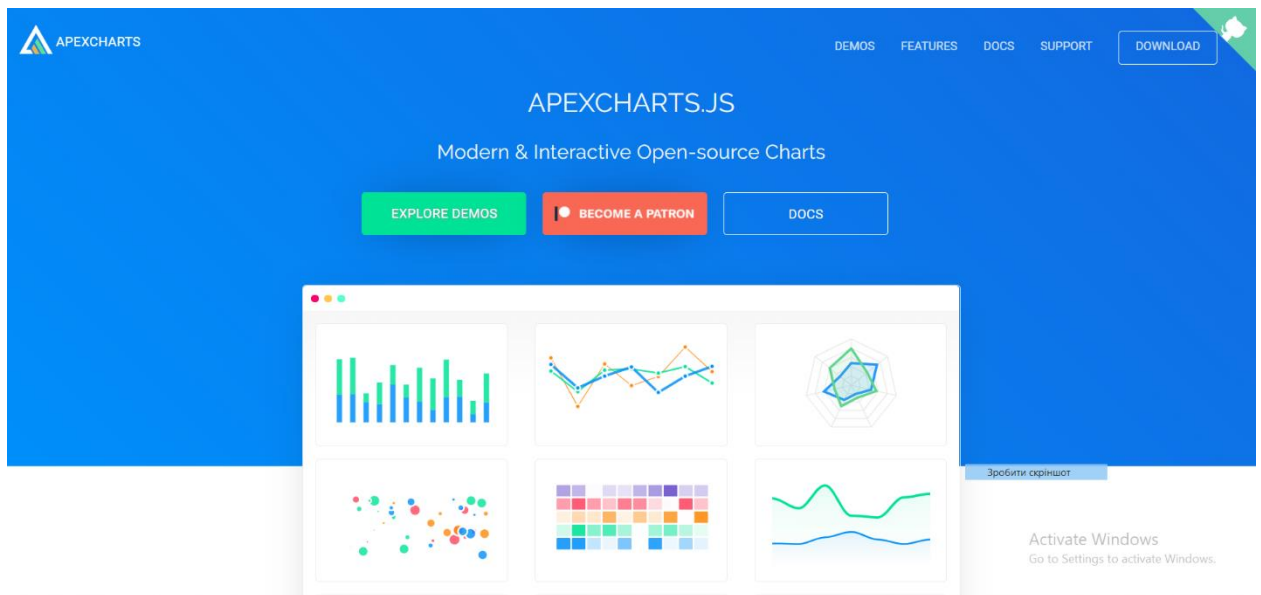


Рис. 3.16. Головна сторінка ApexCharts

Вона є безкоштовною у користуванні та поширюється з ліцензією типу MIT. Великою перевагою при виборі ApexCharts був доступ до багатьох видів графіків для візуалізації даних та можливості для їх комбінування та кастомізації для зручності роботи користувача та виконання специфічних завдань.

Спочатку додаємо модуль "apexcharts": "^3.10.1" та "react-apexcharts": "^1.3.3" у файл package.json та у терміналі виконуємо відповідну команду, як показано на рис. 3.17.

```
npm install --save react-apexcharts apexcharts
```

Рис. 3.17. Інсталяція пакетів ApexCharts

Після завантаження та встановлення модулів імпортуємо модуль у наш компонент, додавши у стрічку import компонент Chart із бібліотеки "react-apexcharts".

React-ApexCharts – це обгортка для компоненту ApexCharts, що дає можливість його підключити як модуль react.js.

Для уніфікації та кастомізації календарних дат завантажимо та підключимо бібліотеку "moment.js". На рис. 3.18. зображено головну сторінку бібліотеки.

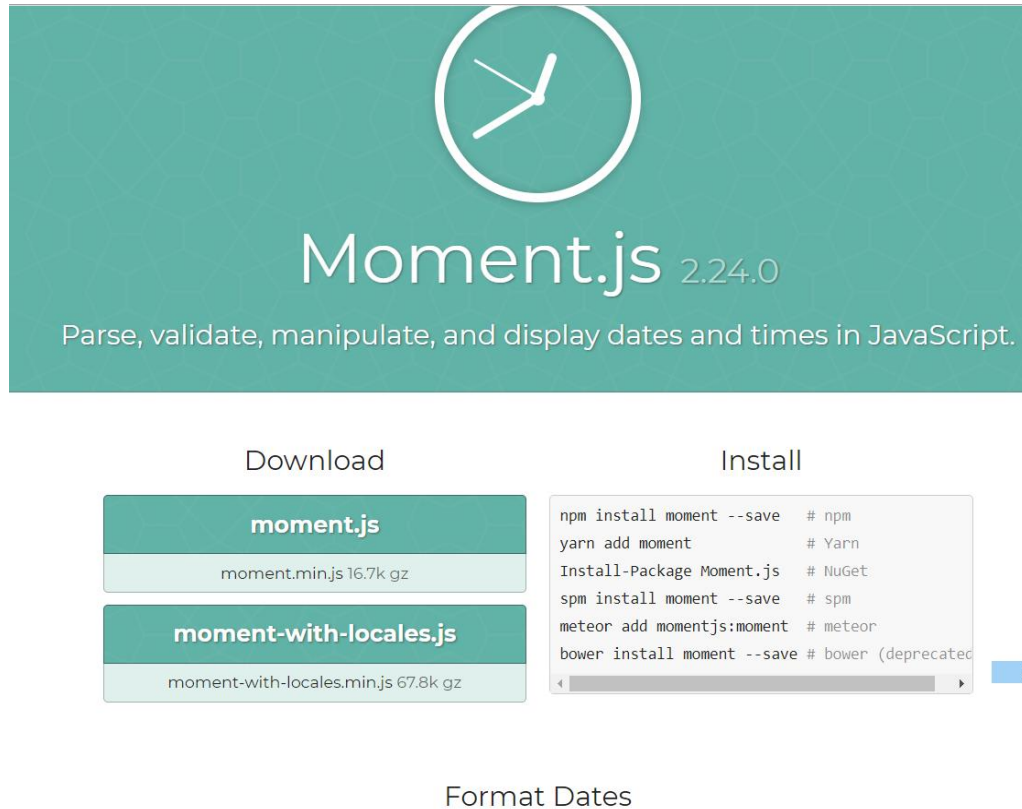


Рис. 3.18. Головна сторінка бібліотеки moment.js

Ця бібліотека є оболонкою для стандартного об'єкта JavaScript - Date. Вона дозволяє обробляти дати та час та проводити арифметичні операції та порівняння з ними [6].

Користуючись офіційною документацією та наведеними демонстраційними прикладами діаграм, доступних на сайті розробника за адресою d3p5cqs32ag3o.cloudfront.net/moment.js/docs, розпочинаємо створення графіків для виконання завдань візуалізації набору даних отриманого з серверної частини проекту.

Для відображення набору даних максимальних та мінімальних температур використовуємо лінійний тип графіка з двома наборами даних та мітками на лініях. На рис. 3.19 зображено візуалізацію масиву даних, що демонструє помісячні максимальні та мінімальні значення температури

повітря для вибраної локації [9]. Даний тип діаграми має високу можливість налаштування (відображення тільки найвищих або найнижчих показників) та кастомізації (зміна кольорової гами, розмірів шрифту, швидкості анімації і т.д.).

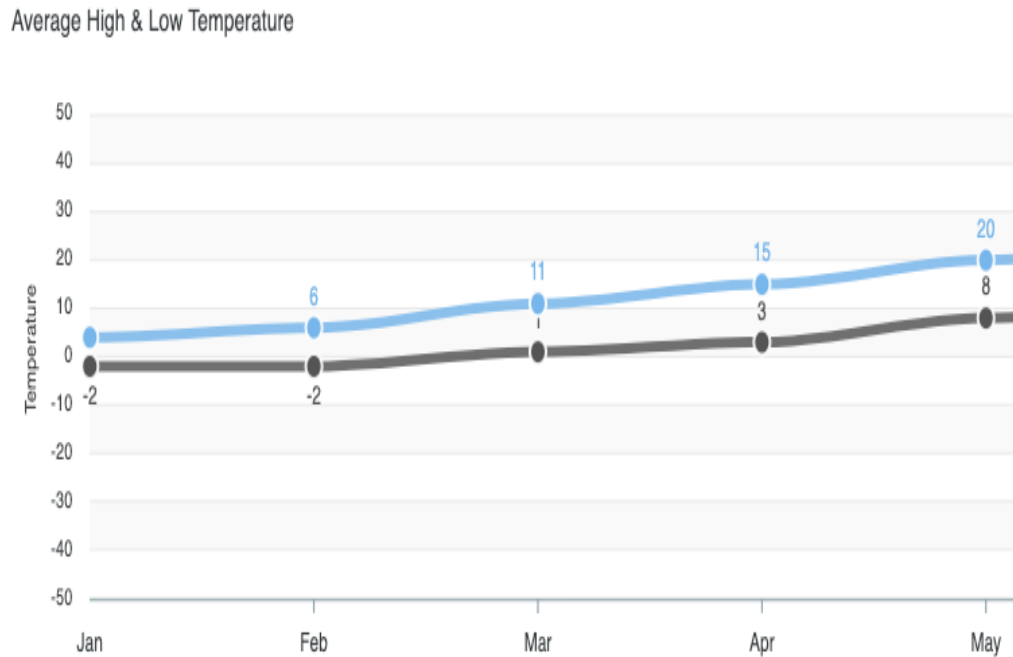


Рис. 3.19. Графік для візуалізації температурних значень

Для опрацювання наборів даних для наведеного графіку додаємо новий метод, як це видно з рис. 3.20.

```

createDate(data){
  let high = [];
  let low = [];
  data.monthlyAvg.forEach(
    item=>{
      high.push(item.high);
      low.push(item.low);
    }
  );
  this.setState( state: { series: [
    {
      name: "High",
      data: high
    },
    {
      name: "Low",
      data: low
    }
  ]});
}

```

Рис. 3.20. Метод для обробки аналізованих даних

Для візуалізації кількості опадів створюємо стовпцевий графік, що демонструє тенденції сезонності у кількості опадів для різних контрольних географічних точок. Зокрема, на наведеному на рис. 3.21 стовпчастому графіку зображений річний розподіл опадів у вибраному географічному районі.

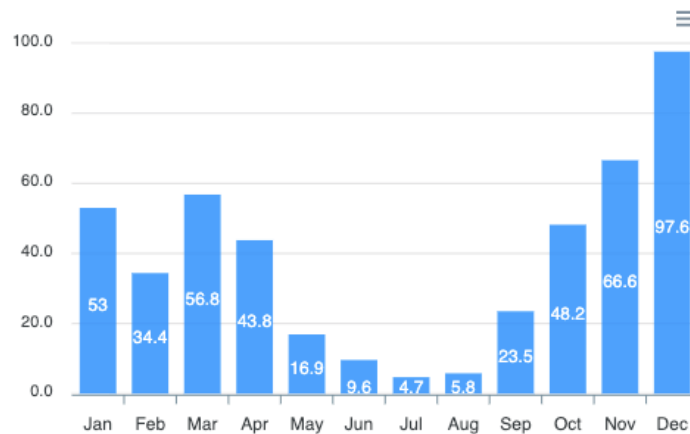


Рис. 3.21. Стовпцевий графік кількості опадів

Для демонстрації співвідношення річної кількості опадів у вигляді снігу та дощу використаємо групований горизонтальний стовпчатий графік, як на рис. 3.22.

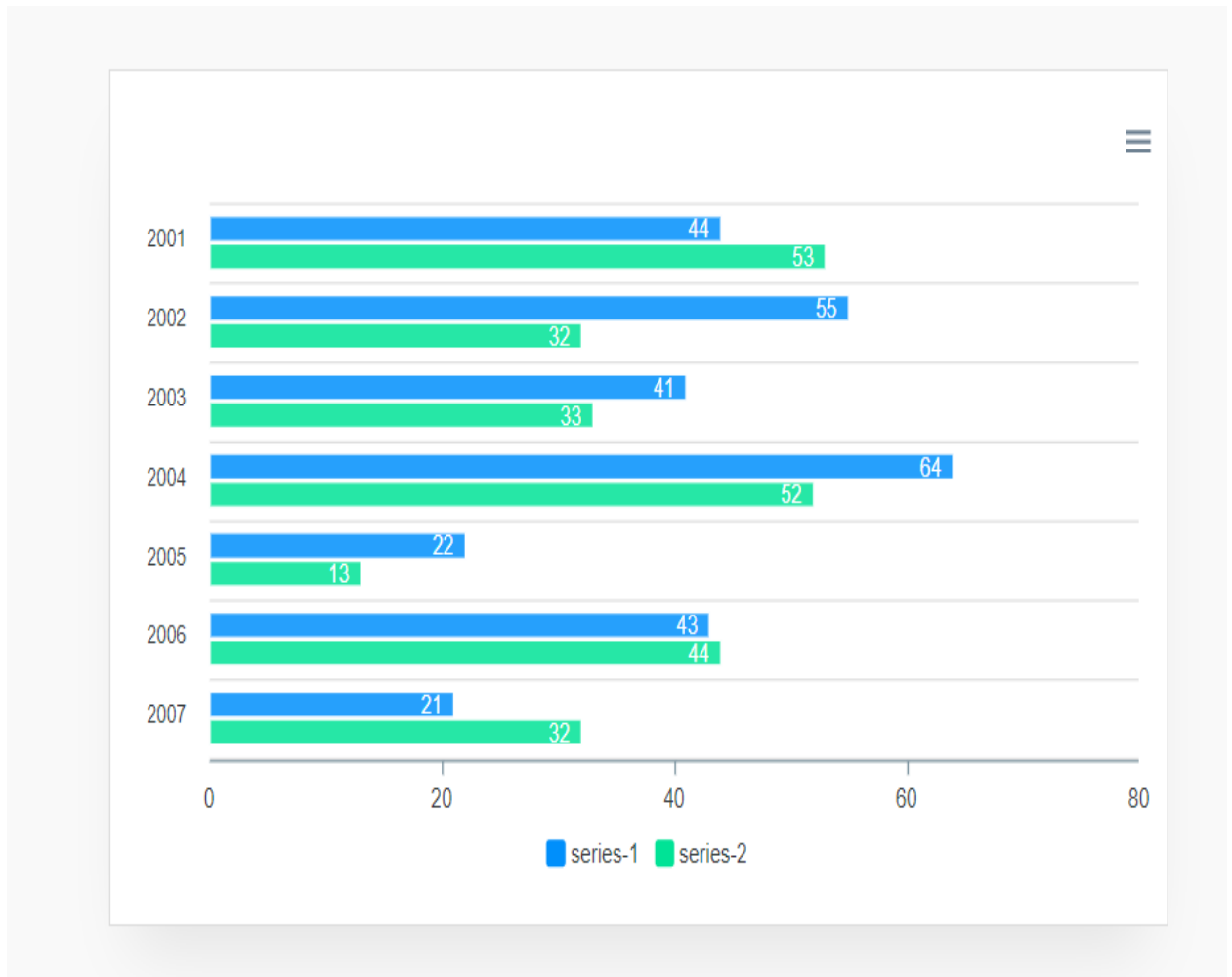


Рис. 3.22. Співвідношення річної кількості опадів

Для прослідкування загальної тенденції рівня температури по географічних місяцях створюємо діаграму (рис. 3.23), що демонструє розкидання температури відповідно до географічних місць:

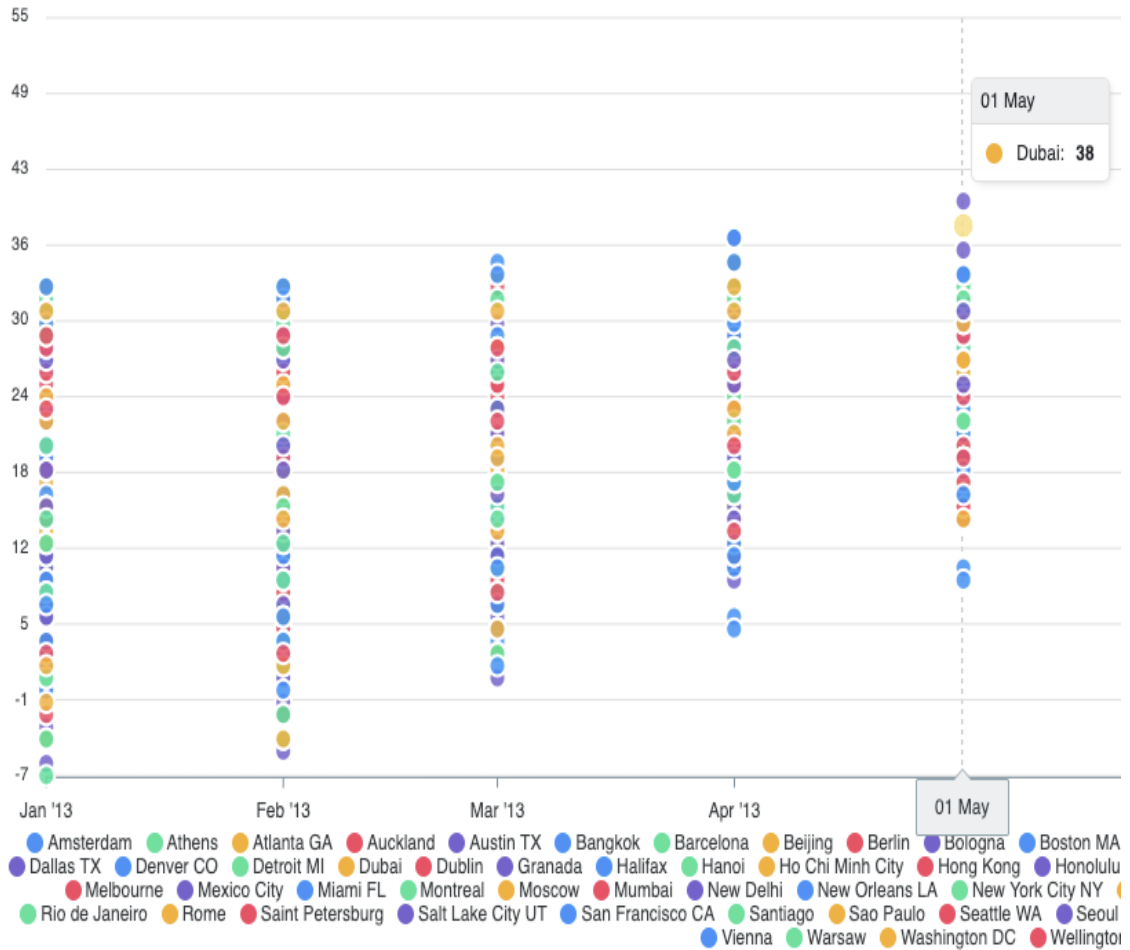


Рис. 3.23. Діаграма розкиду температури відповідно до географічних місць

Дана діаграма демонструє кліматичний розподіл кліматичного показника – середньомісячна температура повітря у багатьох географічних об'єктах, що дає можливість зробити висновки про найбільш типові значення та крайні точки [9].

Також слід відзначити, що можна використовувати змішані типи інфографік для вивчення взаємозв'язків між різними кліматичними показниками.

Важливою перевагою даної системи візуалізації даних є те, що незважаючи на різноманітність створених графіків, способи їх налаштування та кастомізації досить схожі.

Спочатку налаштуємо зовнішній вигляд полотна графіка, як показано на рис. 3.24.

```
shadow: {
  enabled: true,
  color: '#000',
  top: 18,
  left: 7,
  blur: 10,
  opacity: 1
},
toolbar: {
  show: false
}
},
colors: ['#77B6EA', '#545454'],
dataLabels: {
  enabled: true,
},
stroke: {
  curve: 'smooth'
},
```

Рис. 3.24. Налаштування зовнішнього вигляду полотна графіка

Далі налаштуємо осі графіку для візуалізації даних, як видно на рис. 3.25.

```
xaxis: {
  categories: moment.monthsShort(),
  title: {
    text: 'Month'
  }
},
yaxis: {
  title: {
    text: 'Temperature'
  },
  min: -50,
  max: 50
},
```

Рис. 3.25. Налаштування осей графіку

Для передачі відображення опрацьовані та проаналізовані дані передаються за допомогою серій даних (рис. 3.26).

```
series: [  
  {  
    name: "High",  
    data: []  
  },  
  {  
    name: "Low",  
    data: []  
  }  
],
```

Рис. 3.26. Серії даних

Після здійснення всіх необхідних налаштувань ми отримуємо потужний інструмент для візуалізації наборів даних для системи аналізу кліматичних показників.

Бібліотека ApexCharts дала нам кросбраузерне та адаптивне рішення, яке ми використали та кастомізували для того, щоб візуалізувати закономірності рівня максимальної та температури в залежності від географічного розташування досліджуваних об'єктів, а також створили діаграму, яка демонструє сезонність кількості опадів.

В подальшому розвитку нашої аплікації ми будемо створювати нові графіки для демонстрації аналізованих даних залежно від потреб користувача та доступних наборів даних.

3.3 Висновки розділу

Отже, у даному розділі досліджено та розроблено клієнтську частину комп'ютерної системи аналізу та візуалізації кліматичних даних. Для розробки використано мову програмування JavaScript (стандарт ECMAScript 6) та базований на ній фреймворк ReactJS v.16.12.0.

У процесі стилізації аплікації та створенні кроссбраузерного і адаптивного рішення, що відповідає сучасним вимогам UI/UX дизайну, було проаналізовано та застосовано CSS фреймворк React Bootstrap v4.0.1.

Функціонал взаємодії із API серверної частини реалізовано за допомогою інтерфейсу Fetch API, що розширює можливості у порівнянні з використанням стандартного об'єкта XMLHttpRequest.

Для розробки візуалізації отриманих та проаналізованих даних було використано JavaScript бібліотеку ApexCharts, яка використовує SVG графіку для відмальовування діаграм, що в результаті дало можливість отримати високий рівень кастомізації та адаптивності динамічних графіків.

Особливістю роботи є те, що основну кількість логічних операцій із аналізу історичних кліматичних показників ми зосредили у клієнтській частині, що пришвидшило процес візуалізації інформації для користувача у порівнянні з класичним підходом до розробки клієнт-серверних програм, де основна бізнес-логіка зосереджена на серверній частині, таким чином було підтверджено наукову новизну та високу актуальність науково-дослідної роботи.

РОЗДІЛ 4

ОБҐРУНТУВАННЯ ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ

4.1. Розрахунок витрат на розробку та впровадження проектного рішення

1) Витрати на розробку і впровадження програмного продукту (К) визначаються за формулою 4.1:

$$K_{\text{заг}} = K_1 + K_2, \quad (4.1)$$

де K_1 – витрати на розробку програмного продукту, грн.;

K_2 – витрати на відлагодження і дослідну експлуатацію програмного продукту на ЕОМ, грн.

Витрати на розробку програмного продукту включають в себе [10]:

1. Витрати на оплату праці розробників (B_{on});
2. Єдиний соціальний внесок ($B_{есв}$);
3. Вартість додаткових виробів, що закуповуються (B_o);
4. Транспортно-заготівельні витрати (B_{mp});
5. Накладні витрати (B_n);
6. Інші витрати ($B_{ин}$).

Для проведення розрахунків витрат на оплату праці необхідно визначити категорії працівників, які беруть участь у процесі проектування, середньоденну заробітну плату спеціаліста відповідної категорії та трудомісткість робіт у людино-днях (людино-годинах), а також їх чисельність.

Середньоденна заробітна плата i -го розробника ($ЗП_{Di}$) розраховується за формулою 4.2.

(4.2)

$$ЗП_{ди} = \frac{ЗП_i}{\Phi_M},$$

де $ЗП_i$ – основна місячна заробітна плата розробника i -ої спеціальності, грн.;

Φ_M – місячний фонд робочого часу, днів.

У проекті беруть участь 4 працівники:

- 1) front-end програміст – 17 000 грн./міс.;
- 2) back-end програміст – 17 000 грн./міс.;
- 3) бізнес-аналітик – 12 000 грн./міс.;
- 4) UI-UX веб-дизайнер – 10 000 грн./міс.

Середня кількість робочих днів у місяці становить 22, а при 40-годинному робочому тижні загальна кількість робочих годин – 176.

Трудомісткість робіт над проектом становить для:

- 1) front-end програміста – 20 людино-днів;
- 2) back-end програміста – 20 людино-днів;
- 3) бізнес-аналітика – 15 людино-днів;
- 4) UI-UX веб-дизайнера – 17 людино-днів.

Денна заробітна плата становить для:

- 1) front-end програміста:

$$ЗП_{ди} = 17000/22 = 772,73 \text{ грн./день};$$

- 2) back-end програміста:

$$ЗП_{ди} = 17000/22 = 772,73 \text{ грн./день};$$

- 3) бізнес-аналітика:

$$ЗП_{ди} = 12000/22 = 545,45 \text{ грн./день};$$

- 4) UI-UX веб-дизайнера:

$$ЗП_{ди} = 10000/22 = 454,55 \text{ грн./день}.$$

Розрахунок витрат на оплату праці усіх розробників проекту обчислюємо за формулою 4.3.

(4.3)

$$B_{оп} = \sum_{i=1}^N n_i \cdot t_i \cdot ЗП_{дi}$$

де n_i – чисельність розробників проекту і-ої спеціальності, осіб;

t_i – час, витрачений на розробку проекту працівником і-ої спеціальності, дні;

$ЗП_{дi}$ – денна заробітна плата розробника і-ої спеціальності, грн.

$$B_{оп} = 1 \cdot 20 \cdot 772,73 + 1 \cdot 20 \cdot 772,73 + 1 \cdot 15 \cdot 545,45 + 1 \cdot 17 \cdot 454,55 = 46818,3 \text{ грн.}$$

Розрахунок витрат на оплату праці розробників наведено у табл. 4.1.

Таблиця 4.1

Розрахунок витрат на оплату праці

Спеціальність розробника	Кількість розробників, чол.	Час роботи, дні	Денна заробітна плата розробника, грн.	Витрати на оплату праці, грн.
Front-end програміст	1	20	772,73	15454,6
Back-end програміст	1	20	772,73	15454,6
Бізнес-аналітик	1	15	545,45	8181,75
UI-UX веб-дизайнер	1	17	454,55	7727,35
Всього				46818,3

2) Витрати на оплату праці працівникам включають в себе додаткові зобов'язання підприємства по оплаті єдиного соціального внеску (ЄСВ) $B_{есв}$.

Ставка єдиного соціального внеску встановлюються у відсотках до бази нарахування цього внеску відповідно до класів професійного ризику

виробництва. Видання програмного забезпечення, консультування з питань інформатизації, комп'ютерне програмування, діяльність із керування комп'ютерним устаткуванням та інша діяльність у сфері інформаційних технологій та комп'ютерних систем належать до 2 класу професійного ризику, відповідно до якого ставка ЄСВ складає 36,77 % [11].

Отже, витрати на оплату ЄСВ:

$$V_{есв} = 46818,3 \cdot 0,3677 = 17215,09 \text{ грн.}$$

3) Витрати на додаткові вироби, що закупаються (B_d) (папір, диски тощо) визначаються за їх фактичними цінами з врахуванням найменування, номенклатури та їх необхідної кількості у проекті. Транспортно-заготівельні витрати ($B_{тр}$) приймаються як 10 % від суми витрат на додаткові вироби, що закупаються. Вихідні дані та результати розрахунків наведено у таблиці 4.2.

Таблиця 4.2

Розрахунок витрат на куповані вироби

Найменування купованих виробів	Марка, тип	Кількість на розробку, шт.	Ціна за од., грн.	Сума витрат, грн.	Сума витрат з урахуванням транспортно-заготівельних витрат, грн.
Папір	Maestro Standart A4, 500 арк.	2	90	180	200
Всього					200

4) Витрати на придбання спецобладнання (B_{co}) розраховують у тому випадку, якщо для розроблення та впровадження проектного рішення необхідне придбання додаткових технічних засобів. Оскільки під час розроблення проектного рішення у придбанні спецобладнання потреби не було, тому його вартість рівна 0.

5) Накладні витрати (B_n) включають витрати на управління, загальногосподарські та невиробничі витрати. Приймаємо накладні витрати на рівні 25 % від витрат на оплату праці. Обчислюємо накладні витрати:

$$B_n = 46818,3 \cdot 0,25 = 11704,58 \text{ грн.}$$

5) Інші витрати (B_{in}) – це витрати, які не враховані у попередніх статтях витрат. Вони розраховуються за встановленими відсотками до витрат на оплату праці (11 %).

$$B_{in} = 46818,3 \cdot 0,11 = 5150,01 \text{ грн.}$$

6) Витрати на розроблення проектного продукту розраховуємо за формулою 4.4.

$$K_1 = B_{оп} + B_{ссв} + B_{д} + B_n + B_{in}, \quad (4.4)$$

$$K_1 = 46818,3 + 17215,09 + 200 + 11704,58 + 5150,01 = 81087,98 \text{ грн.}$$

7) Витрати на відлагодження і дослідну експлуатацію розраховуємо за формулою 4.5.

$$K_2 = S_{м.г.} \cdot t_{від}, \quad (4.5)$$

де $S_{м.г.}$ – вартість однієї години роботи ПК, грн./год.;

$t_{від}$ – кількість годин роботи ПК на відлагодження програми, год.

Тривалість дослідної експлуатації системи та відлагодження розробниками і-ої спеціальності становить 3 доби при 8-годинному робочу дні. Для 4-х розробників на відлагодження програми кількість годин роботи ПК становить:

$$t_{від} = 3 \cdot 8 \cdot 4 = 96 \text{ год.}$$

Тариф на електроенергію становить 1,4314 грн./кВт, ПК споживає 0,035 кВт/год. під час роботи. Тому вартість однієї години роботи ПК:

$$S_{м.г.} = 0,035 \cdot 1,4314 = 0,05 \text{ грн./год.}$$

Витрати на дослідну експлуатацію системи та відлагодження становлять:

$$K_2 = 0,05 \cdot 96 = 4,8 \text{ грн.}$$

Результати розрахунків наведено у таблиці 4.3.

Таблиця 4.3

Кошторис витрат на розробку програмного продукту

Найменування елементів витрат	Сума витрат, грн.
Витрати на розробку програмного продукту, у т.ч.:	
витрати на оплату праці	46818,30
сплата єдиного соціального внеску	17215,09
витрати на додаткові вироби, що закупаються	200,00
витрати на придбання спецобладнання	0,00
накладні витрати	11704,58
інші витрати	5150,01
Витрати на дослідну експлуатацію системи та відлагодження	4,80
Всього	81092,78

4.2. Визначення комплексного показника якості

Комплексний показник якості (P_j) визначається шляхом порівняння вибраного аналогу та показників якості проектованої системи.

Вибір показників якості здійснюють експертним методом. Для визначення P_j доцільно використовувати систему показників якості та технічного рівня, що включає в себе такі групи:

1. Показники призначення (функціональності):
 - 1.1) ступінь новизни;
 - 1.2) універсальність;

- 1.3) актуальність.
- 2. Показники надійності:
 - 2.1) стійкість до помилок;
 - 2.2) захищеність;
 - 2.3) відновлюваність.
- 3. Зручність застосування:
 - 3.1) зрозумілість;
 - 3.2) оперативність.
- 4. супроводжуваність:
 - 4.1) стабільність;
 - 4.2) аналізованість.

Комплексний показник якості проектованої системи визначається методом арифметичного середньозваженого за формулою 4.6.

$$P_{\text{я}} = \sum_{i=1}^m C_i \times q_i, \quad (4.6)$$

де m – кількість одиничних параметрів (показників), прийнятих для оцінки якості проектованої системи;

q_i – коефіцієнт вагомості кожного з параметрів щодо їх впливу на технічний рівень та якість проектованої системи (який встановлюється експертним шляхом).

Коефіцієнт вагомості визначається:

$$\sum_{i=1}^m q_i = 1, 0. \quad (4.7)$$

Часткові показники якості, визначені порівнянням числових значень одиничних показників проекрованої системи й аналога розраховуються за формулами:

$$C_i = \frac{P_{npi}}{P_{ai}} \quad \text{або} \quad C_i = \frac{P_{ai}}{P_{npi}}, \quad (4.8)$$

де P_{npi} , P_{ai} – кількісні значення i -го одиничного показника якості відповідно проекрованої аналога й системи.

Результати розрахунку комплексного показника якості наведено у таблиці 4.4.

Таблиця 4.4

**Визначення комплексного показника якості проекрованої системи
(аналога)**

Показники	Числове значення показників, бали		Відносний показник якості, C_i	Коефіцієнт вагомості, q_i	$C_i \times q_i$
	Аналог	Розроблене проектне рішення			
1	2	3	4	5	6
Показники призначення					
Актуальність	5	9	1,80	0,1	0,18
Універсальність	4	7	1,75	0,2	0,35
Ступінь новизни	4	9	2,25	0,05	0,11
Показники надійності					
Стійкість до помилок	6	10	1,67	0,2	0,33
Відновлюваність	8	7	1,14	0,1	0,11
Захищеність	8	8	1,00	0,05	0,05
Зручність застосування					
Зрозумілість	9	10	1,11	0,05	0,06
Оперативність	7	9	1,29	0,1	0,13
Супроводжуваність					
Аналізованість	5	8	1,60	0,05	0,08
Стабільність	8	9	1,13	0,1	0,11
Всього				1,00	1,51

Комплексний показник якості дорівнює:

$$P_{\text{я}} = 0,18 + 0,35 + 0,11 + 0,33 + 0,11 + 0,05 + 0,06 + 0,13 + 0,08 + 0,11 = 1,51.$$

Отриманий результат показує, що розроблена система для аналізу та візуалізації кліматичних даних за багатьма показниками є кращою у порівнянні з аналогом.

4.3. Визначення експлуатаційних витрат

Під час порівняння програмних продуктів вартість підготовки даних (E_1) і вартість годин роботи ПК (E_2) включають в експлуатаційні витрати. Одноразові експлуатаційні витрати визначаються за формулою:

$$E_{П(A)} = E_{1П(A)} + E_{2П(A)}, \quad (4.9)$$

де $E_{П(A)}$ – одноразові експлуатаційні витрати на проектне рішення (аналог), грн.;

$E_{1П(A)}$ – вартість підготовки даних для експлуатації проектного рішення (аналогу), грн.;

$E_{2П(A)}$ – вартість машино-годин роботи ПК для проектного рішення (аналогу), грн.

Вартість підготовки даних для експлуатації проектного рішення (аналогу) (E_1) визначаються за формулою 4.10.

$$E_1 = \sum_{i=1}^N n_i \cdot t_i \cdot 3Пe_i, \quad (4.10)$$

де i – номери категорій персоналу, які беруть участь у підготовці даних;

n_i – кількість співробітників i -ої категорії, осіб;

t_i – тривалість роботи співробітників i -ої категорії, год.;

$ЗП_{2i}$ – середньогодинна ставка працівника і-ої категорії з врахуванням оплати єдиного соціального внеску, грн./год.

Середньогодинну ставку розраховується за формулою:

$$ЗП_{2i} = \frac{ЗП_{2oi}(1+b)}{\Phi_2}, \quad (4.11)$$

де $ЗП_{2oi}$ – основна місячна зарплата працівника і-ої категорії, грн.;

b – коефіцієнт, який враховує оплату єдиного соціального внеску;

Φ_2 – місячний фонд робочого часу, год.

Отже, визначаємо середньогодинну ставку:

- для проектного продукту:

1) front-end програміст

$$ЗП_{2i} = 17000(1+0,3677)/176 = 132,11 \text{ грн./год.};$$

2) back-end програміст

$$ЗП_{2i} = 17000(1+0,3677)/176 = 132,11 \text{ грн./год.};$$

3) бізнес-аналітик

$$ЗП_{2i} = 12000(1+0,3677)/176 = 93,25 \text{ грн./год.};$$

4) UI-UX веб-дизайнер

$$ЗП_{2i} = 10000(1+0,3677)/176 = 77,71 \text{ грн./год.};$$

- для аналога:

1) front-end програміст

$$ЗП_{2i} = 19000(1+0,3677)/176 = 147,65 \text{ грн./год.};$$

2) back-end програміст

$$ЗП_{2i} = 19000(1+0,3677)/176 = 147,65 \text{ грн./год.};$$

3) бізнес-аналітик

$$ЗП_{2i} = 13000(1+0,3677)/176 = 101,02 \text{ грн./год.};$$

4) UI-UX веб-дизайнер

$$ЗП_{2i} = 11000(1+0,3677)/176 = 85,48 \text{ грн./год.}$$

Обчислюємо вартість підготовки даних для експлуатації проектного рішення та аналогу:

$$E_{1П} = 1 \cdot 20 \cdot 132,11 + 1 \cdot 20 \cdot 132,11 + 1 \cdot 15 \cdot 93,25 + 1 \cdot 17 \cdot 77,71 = 8004,22 \text{ грн.}$$

$$E_{1A} = 1 \cdot 21 \cdot 147,65 + 1 \cdot 20 \cdot 147,65 + 1 \cdot 14 \cdot 101,02 + 1 \cdot 19 \cdot 85,48 = 9092,05 \text{ грн.}$$

Тариф на електроенергію становить 1,4314 грн./кВт. Під час роботи ПК споживає 0,035 кВт/год. Тому вартість однієї години роботи ПК становить 0,05 грн./год.

Вартість машино-годин роботи ПК для проектного рішення та аналогу буде становити:

$$E_{2П} = 0,05 \cdot (20 + 20 + 15 + 17) = 3,6 \text{ грн.}$$

$$E_{2A} = 0,05 \cdot (21 + 20 + 14 + 19) = 3,2 \text{ грн.}$$

Одноразові експлуатаційні витрати становлять:

$$E_{П} = 8004,22 + 3,6 = 8007,82 \text{ грн.}$$

$$E_{A} = 9092,05 + 3,2 = 9095,25 \text{ грн.}$$

Річні експлуатаційні витрати визначаються за формулою 4.12.

$$B_{(e)П(A)} = E_{П(A)} \cdot N_{П(A)}, \quad (4.12)$$

де $B_{(e)П(A)}$ – експлуатаційні річні витрати проектного рішення (аналогу), грн.;

$N_{П(A)}$ – періодичність експлуатації проектного рішення (аналогу), разів/рік.

Річні експлуатаційні витрати для проектного рішення (аналогу) становлять, враховуючи те, що періодичність експлуатації проектного рішення 12 разів/рік:

$$B_{(e)П} = 8007,82 \cdot 12 = 96093,84 \text{ грн.}$$

$$B_{(e)A} = 9095,25 \cdot 12 = 109143,00 \text{ грн.}$$

Вихідні дані та результати розрахунків витрат для експлуатації проектного рішення наведено у таблиці 4.5.

Таблиця 4.5

Розрахунок витрат на підготовку даних для роботи на ЕОМ

Категорія персоналу	Чисельність співробітників і-ої категорії, чол.	Час роботи співробітників і-ої категорії, год.	Середньогодинна ЗП співробітника і-ої категорії, грн.	Витрати на підготовку даних, грн.
Проектне рішення				
front-end програміст	1	20	132,11	2642,20
back-end програміст	1	20	132,11	2642,20
бізнес-аналітик	1	15	93,25	1398,75
UI-UX веб-дизайнер	1	17	77,71	1321,07
Всього				8004,22
Аналог				
front-end програміст	1	21	147,65	3100,65
back-end програміст	1	20	147,65	2953,00
бізнес-аналітик	1	14	101,02	1414,28
UI-UX веб-дизайнер	1	19	85,48	1624,12
Всього				9092,05

Проектне рішення було розроблено із меншими витратами у порівнянні з аналогом, тому розроблена система є більш рентабельною та дешевшою, ніж аналог.

4.4. Розрахунок ціни споживання проектного рішення

Ціна споживання (C_C) – це витрати на придбання і експлуатацію проектного рішення за весь строк його служби:

$$C_{C(P)} = C_P + B_{(E)NPV}, \quad (4.13)$$

де C_P – ціна придбання проектного рішення, грн.;

$B_{(E)NPV}$ – теперішня вартість витрат на експлуатацію проектного рішення, грн.

Ціна придбання проектного рішення розраховується за формулою 4.14.

$$C_{II} = K \cdot \left(1 + \frac{P_P}{100}\right) \cdot (1 + C_{ПДВ}) + K_0 + K_k, \quad (4.14)$$

де P_P – норматив рентабельності;

$C_{ПДВ}$ – ставка податку на додану вартість (20 %);

K_0 – витрати на прив'язку та освоєння проектного рішення на конкретному об'єкті, грн.;

K_k – витрати на доукомплектування технічних засобів на об'єкті, грн.

Витрат на освоєння та доукомплектування не передбачається ($K_0 = 0$ грн., $K_k = 0$ грн.). Приймається норматив рентабельності $P_P = 30$ %.

Ціна придбання проектного рішення буде становити:

$$C_{II} = 81092,78 \cdot (1 + 0,3) \cdot (1 + 0,2) = 126504,74 \text{ грн.}$$

Ціна придбання аналогу:

$$C_A = 130\,000 \text{ грн.}$$

Теперішня вартість витрат на експлуатацію проектного рішення розраховується за формулою:

$$B_{(e)NPV} = \sum_{t=1}^T \frac{B_{(E)Nt}}{(1 + R)^t}, \quad (4.15)$$

де $B_{(E)Nt}$ – річні експлуатаційні витрати у t-ому році, грн.;

T – строк служби проектного рішення, років (1 рік);

R – річна ставка проценту банків (25 %).

$$B_{(e)NPV} = 96093,84 / (1 + 0,25) = 76875,07 \text{ грн.}$$

Теперішня вартість витрат на експлуатацію аналогу становить:

$$B_{(e)NPV} = 109143,00 / (1 + 0,25) = 87314,40 \text{ грн.}$$

Тоді ціна споживання проектного рішення становить:

$$Ц_{СП} = 126504,74 + 76875,07 = 203379,81 \text{ грн.}$$

Ціна споживання аналогу:

$$Ц_{СА} = 130\ 000 + 87314,40 = 217314,40 \text{ грн.}$$

4.5. Визначення показників економічної ефективності

1) Показник конкурентоспроможності розраховуємо за формулою 4.16.

$$K_{кс} = \frac{Ц_{c(n)} \cdot П_{я}}{Ц_{c(a)}}, \quad (4.16)$$

$$K_{КС} = 203379,81 \cdot 1,51 / 217314,40 = 1,41.$$

2) Економічний ефект у сфері експлуатації (грн.) становить:

$$E_{екс} = B_{(e)a} - B_{(e)n}, \quad (4.17)$$

$$E_{екс} = 109143,00 - 96093,84 = 13049,16 \text{ грн.}$$

3) Економічний ефект у сфері проектування (грн.) розраховується:

$$E_{np} = Ц_a - Ц_n, \quad (4.18)$$

$$E_{np} = 130\ 000 - 126504,74 = 3495,26 \text{ грн.}$$

Оскільки $E_{np} > 0$ та $E_{екс} > 0$, тому розраховуємо:

5) Додатковий економічний ефект у сфері експлуатації (грн.):

$$E_{ексД} = \sum_{i=1}^T E_{екс} (1 + R)^{T-i}, \quad (4.19)$$

$$E_{ексД} = 13049,16 \cdot (1 + 0,25)^{1-1} = 16311,45 \text{ грн.}$$

5) Додатковий економічний ефект у сфері проектування (грн.):

$$E_{прД} = E_{пр} \cdot (1 + R)^T, \quad (4.20)$$

$$E_{прД} = 3495,26 \cdot (1 + 0,25)^1 = 4369,08 \text{ грн.}$$

б) Термін окупності витрат на проектування рішення (років):

$$T_{ок} = K / Ц_n \quad (4.21)$$

$$T_{ок} = 81092,78 / 126504,74 = 0,64 \text{ роки.}$$

Результуючі показники економічної ефективності приведені у таблиці

4.6.

Таблиця 4.6

Показники економічної ефективності проектного рішення

Найменування показників	Одиниці вимірювання	Значення показників	
		Аналог	Проектне рішення
1. Капітальні вкладення	грн.		81092,78
2. Ціна придбання	грн.	130 000,00	126504,74
3. Річні експлуатаційні витрати	грн.	109143,00	96093,84
4. Ціна споживання	грн.	217314,40	203379,81
5. Економічний ефект у сфері експлуатації	грн.		13049,16
6. Додатковий економічний ефект у сфері експлуатації	грн.		16311,45
7. Економічний ефект у сфері проектування	грн.		3495,26
8. Додатковий економічний ефект у сфері проектування	грн.		4369,08
9. Термін окупності витрат на проектування рішення	роки		0,64
10. Коефіцієнт конкурентоспроможності			1,41

4.6. Висновки до розділу

У цьому розділі проведено економічну оцінку проектного рішення. Розраховано витрати на розроблення і впровадження проектного рішення, показник якості та експлуатаційні витрати, ціну споживання проектного рішення та аналогу.

На основі розрахунків економічного ефекту у сфері експлуатації (13049,16 грн.) та сфері проектування (3495,26 грн.) встановлено, що розроблення системи аналізу та візуалізації кліматичних даних є економічно доцільним. Коефіцієнт конкурентоспроможності становить 1,41.

Капітальні вкладення становлять 81092,78 грн. Ціна придбання проектного рішення – 126504,74 грн., а аналогу – 130 000 грн. Термін окупності витрат на проектування рішення – 0,64 роки. Отже, розроблення системи аналізу та візуалізації кліматичних даних є економічно вигідним.

РОЗДІЛ 5

ОХОРОНА ПРАЦІ ТА БЕЗПЕКИ В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

Питання охорони праці та безпеки в надзвичайних ситуаціях розглянуті для етапу проектування й розробки системи аналізу та візуалізації кліматичних даних.

Охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності. Умови праці на робочому місці, безпека технологічних процесів, машин, механізмів, устаткування та інших засобів виробництва, стан засобів колективного та індивідуального захисту, що використовуються працівником, а також санітарно-побутові умови повинні відповідати вимогам законодавства. Працівник має право відмовитися від дорученої роботи, якщо створилася виробнича ситуація, небезпечна для його життя чи здоров'я або для людей, які його оточують, або для виробничого середовища чи довкілля. Він зобов'язаний негайно повідомити про це безпосереднього керівника або роботодавця. Факт наявності такої ситуації за необхідності підтверджується спеціалістами з охорони праці підприємства за участю представника профспілки, членом якої він є, або уповноваженої працівниками особи з питань охорони праці (якщо професійна спілка на підприємстві не створювалася), а також страхового експерта з охорони праці [12]. Завдання охорони праці – звести до мінімуму ушкодження та захворювання працівника з одночасним забезпеченням комфорту при максимальній продуктивності праці. Основними цілями охорони праці є формування в спеціалістів необхідних знань і практичних навичок по правових і організаційних питаннях охорони праці, виробничій санітарії, техніці безпеки, пожежній безпеці.

5.1. Загальна характеристика приміщення і робочого місця

Розробка системи аналізу та візуалізації виконується в приміщенні, яке знаходиться на четвертому поверсі восьмиповерхового будинку з загальним та місцевим освітленням. В приміщенні одностороннє освітлення, вікна орієнтовані на схід, на вікнах є ролети. Стеля білого кольору з коефіцієнтом відбиття 0,7, стіни цегляні світлого кольору з коефіцієнтом відбиття 0,5. В приміщенні працює 4 людини, відповідно до цього отримуємо вхідні дані для аналізу потенційно-небезпечних і шкідливих виробничих факторів, які наведено в табл. 5.1.

Таблиця 5.1

Вхідні дані

Параметри приміщення	Значення
Довжина x ширина x висота	6,6 x 6,1 x 2,7 м
Площа	40,26 м ²
Об'єм	108,70 м ³
Номер робочого місця	Специфіка роботи
I робоче місце	Front-end програміст (спеціаліст з розробки клієнтської частини веб- застосунків)
II робоче місце	Back-end програміст (спеціаліст з розробки серверної частини веб застосунків та проектування баз даних)
III робоче місце	Бізнес-аналітик (також виконує роль менеджера продукту)
IV робоче місце	UI-UX веб-дизайнер
Технічні засоби (кількість)	Назва та характеристики
Монітор (4 шт.)	HP 22Xi/21,5"/1920x1080px/IPS
Комп'ютер (4 шт.)	HP ProBook 440 G6, екран 14" IPS (1920x1080) Full HD, Intel Core i7-8565U (1.8 - 4.6 ГГц)/RAM 16 ГБ/SSD 256 ГБ
Підлоговий кулер (1 шт.)	CRYSTAL YLR3-5V208
Кондиціонер (1 шт.)	DEKKER DSH105R/G/26м ² /2,65кВт-2,9кВт/25x74,5x19,5см/9 кг
Світильники загального призначення (3 шт.)	Світильник растровий вмонтований 4x18W
Світильники місцевого призначення (4 шт.)	DeLux Décor TF-05 / 1 x 40Вт

Згідно НПАОП 0.00-7.15-18 [14] площа S' , виділена для одного робочого місця з персональною ЕОМ, повинна бути не менше 6 м^2 і об'єм – не менше 20 м^3 . У приміщенні розташовано 4 робочі місця, що повністю відповідає необхідним нормам.

Розрахуємо фактичні значення цих показників, розділивши об'єм приміщення та загальну площу на кількість працюючих.

Отже, виходячи з отриманих результатів за характеристиками площі та об'єму, приміщення відповідає нормам.

Таблиця 5.2

Характеристики робочого місця

№	Найменування параметру	Значення	
		фактичне	нормативне
1.	Висота робочої поверхні, мм	780	680 – 800
2.	Ширина робочої поверхні, мм	1500	не менше 600
3.	Глибина робочої поверхні, мм	750	не менше 600
4.	Висота простору для ніг, мм	750	не менше 600
5.	Ширина простору для ніг, мм	800	не менше 500
6.	Глибина простору для ніг, мм	750	не менше 450
7.	Висота поверхні сидіння, мм	480	400 – 500
8.	Ширина сидіння, мм	500	не менше 400
9.	Глибина сидіння, мм	500	не менше 400
10.	Висота опорної поверхні спинки, мм	550	не менше 300
11.	Ширина поверхні спинки, мм	470	не менше 380
12.	Довжина підлокітників, мм	300	не менше 250
13.	Ширина підлокітників, мм	60	50 – 70
14.	Відстань від очей до екрану, мм	650	600 – 700

Можна зробити висновок, що розміри робочого місця програміста відповідають встановленим нормам, виходячи з заданих параметрів.

5.2. Аналіз потенційно небезпечних і шкідливих виробничих факторів на робочому місці

При створенні системи аналізу та візуалізації робота виконується сидячи без фізичних зусиль, тому відноситься до категорії легка Ia [15].

Приміщення для роботи мають бути обладнані системами опалення, кондиціонування повітря або припливно-витяжною вентиляцією відповідно до ДБН В.2.5-67:2013. Нормовані параметри мікроклімату, іонного складу повітря, вмісту шкідливих речовин відповідають вимогам ДСН 3.3.6.042-99, ГН 2152-80, ГОСТ 12.1.005-88, ДСТУ ГОСТ 12.0.230:2008 та ДСТУ ГОСТ 12.4.041:2006. Під вентиляцією розуміють сукупність заходів та засобів, призначених для забезпечення на постійних місцях та зонах обслуговування приміщень метеорологічних умов та чистоти повітряного середовища, що відповідають гігієнічним та технічним вимогам. Основне завдання вентиляції – вилучити із приміщення забруднене, вологе або нагріте повітря та подати чисте свіже повітря.

Джерелами шуму в приміщенні є вентилятор системного блоку, ноутбуку та кондиціонер. Звук, що створюється вентилятором та кондиціонером, можна класифікувати як постійний.

Відповідно до ДБН В.2.5-28:2018 робота відноситься до розряду зорових робіт. Передбачається використання природного, штучного та змішаного освітлення.

ЕОМ є однофазним споживачем електроенергії, що живиться від змінного струму 220В від мережі із заземленою нейтраллю. IBM PC відноситься до електроустановок до 1000В закритого виконання, всі струмопровідні частини знаходяться в кожухах. За способом захисту людини від ураження електричним струмом, ЕОМ і периферійна техніка повинні відповідати 1 класу захисту.

Технічні методи захисту від ураження струмом зводиться до застосування струму безпечної напруги, захисту у випадку випадкового

доторкання до струмоведучих частин і від надмірних струмів, захисту у випадку переходу напруги на неструмоведучі металеві частини установок.

Безпечну напругу одержують від сітки підвищеної напруги (110-120 В) за допомогою знижувальних трансформаторів.

Захисту від доторкання до струмоведучих частин установки досягають за допомогою ізоляції, відгородження застосування блокуючих пристроїв запобіжної сигналізації та неприступності розташування установок.

Розподільні щитки поміщають у закриті металеві кожухи-ящики.

Запобіжну сигналізацію застосовують у вигляді плакатів і надписів. Найкращими світловими сигналізаціями є подвійні, яких при наявності напруги горить червона лампочка, а при її відсутності - зелена.

Захист від надмірних струмів – короткого замикання і струмів перевантаження, які можуть спричинити займання ізоляції, здійснюється запобіжниками й автоматичними вимикачами, а захист від переходу напруги на струмоведучі частини за допомогою захисного заземлення і захисного вимикання.

Запобігання пожежі досягається виключенням утворення джерел загорянь і горючого середовища.

В цьому приміщенні можливі пожежі таких класів: А – горіння твердих речовин, Е – горіння електроустановок під напругою.

5.3. Безпека в надзвичайних ситуаціях

Великі аварії і катастрофи можуть призвести до загибелі людей і завдати відчутної шкоди. Тому забезпечення безаварійної роботи на підприємстві потрібно розглядати як важливе завдання, що вимагає уваги керівників усіх рівнів та інженерно-технічного персоналу. Аварії можуть відбутися в результаті стихійних лих, допущених прорахунків у проектуванні, будівництві й устаткуванні підприємств; введення в експлуатацію об'єктів з великими недоробками і відступами від проектів; прийняття в експлуатацію

вентиляційних систем без випробування їх на ефективність роботи; недоробок з техніки безпеки й охорони праці; незадовільного оснащення контрольно-вимірювальною, захисною, блокуючою апаратурою і недостатньої герметичності технологічного устаткування. Вони можуть бути також наслідком технологічних процесів, несправності електропроводки та відсутності надійних систем пожежогасіння.

Кожна конкретна аварія викликається сукупністю ряду причин і несприятливих факторів у результаті низького рівня обізнаності персоналу, допущеної недбалості, порушені правил техніки безпеки. Вивчення причин аварій і всебічна оцінка ступеня небезпеки дозволяють правильно визначити заходи щодо їх попередження, передбачити необхідні заходи захисту людей і зниження збитків.

Основними заходами щодо ліквідації наслідків великих аварій є: оповіщення про небезпеку робітників та службовців; комплексна розвідка об'єкта, на якому відбулася аварія; порятунок людей з-під завалів, зі зруйнованих і пошкоджених будинків та споруд; надання медичної допомоги постраждалим і евакуація їх у лікувальні установи; гасіння пожеж; локалізація аварій на комунально-енергетичних мережах, які перешкоджають веденню рятувальних робіт; улаштування проїздів і проходів до місць аварії; обвалування нестійких конструкцій, розбирання завалів, демонтаж збереженого устаткування, якому загрожує небезпека.

Швидке проведення рятувальних робіт і оперативна ліквідація наслідків аварії вимагають значних сил і засобів, для цих цілей залучаються спеціальні та територіальні формування загального призначення.

Рятувальні роботи в місцях аварії проводяться в умовах загазованості, а при пожежах – задимленості і високих температур, щоб забезпечити безперервність роботи з наростаючим темпом, ресурси поділяють на зміни і виділяють резерви.

Рятувальні роботи та допомога потерпілим організуються негайно після виникнення аварії. До місця аварії першими повинні прибувати

протипожежні команди, підрозділи міліції, машини швидкої медичної допомоги.

Ліквідація наслідків аварії може здійснюватися одночасно на всьому об'єкті чи на окремих ділянках у тих випадках, коли мається достатня кількість сил і засобів, роботи проводяться відразу на всій площі. Якщо сил недостатньо, роботи повинні проводитися послідовно, в першу чергу їх починають там, де необхідно надати допомогу людям, і на ділянках, які становлять найбільшу небезпеку [18].

Перша медична і лікарська допомога надається постраждалим, які знаходяться в стані шоку, а також звільненим з-під завалів і уламків. Витягування людей з-під великих завалів здійснюється з дотриманням заходів безпеки, їм надається невідкладна медична допомога з наступною евакуацією в лікувальні установи.

Для організації робіт з ліквідації наслідків аварій і катастроф на об'єкті створюється постійно діюча надзвичайна оперативна група під керівництвом головного інженера. У надзвичайних умовах вона працює під загальною координацією районної (міської) надзвичайної комісії.

Ліквідація наслідків аварії проводиться в чотири етапи:

1. Вживання екстрених заходів (попередня оцінка обстановки, надання допомоги потерпілим, вживання екстрених заходів по захисту робітників, службовців, населення, локалізація аварії та організація розвідки).

2. Оперативне планування (розвідка, уточнення обстановки, розрахунок необхідних сил і засобів, оцінка масштабів збитків, планування робіт з ліквідації наслідків аварії).

3. Рятувальні роботи (розшук потерпілих, їх витягування з-під завалів, з палаючих будинків, евакуація людей із зони аварії, надання першої медичної й інших видів допомоги постраждалим).

4. Ліквідація наслідків (заходи щодо створення умов для забезпечення життєдіяльності населення в районі аварії, відновлення функціонування [19].

5.4. Висновки до розділу

Аналіз умов праці в розглянутому робочому приміщенні показав, що умови праці з ПЕОМ відповідають нормативам.

Інструктаж і навчання всіх працюючих безпечних методів роботи – основна передумова різкого зниження і навіть повної ліквідації травматизму.

Головне завдання навчання та інструктажу полягає в тому, щоб робітники до вступу на роботу одержали необхідні знання прийому безпечної роботи і вивчили правила техніки безпеки.

В зв'язку з тим, що даний вид праці є шкідливим через напруженість, розробнику ПЗ рекомендується роботи перерви в роботі із виконанням незначних гімнастичних вправ та вправ для очей.

Також ми розглянули питання про особливості проведення рятувальних й інших невідкладних робіт при ліквідації насадків великих виробничих аварій і катастроф на об'єктах господарювання, де були наведені основні причини виникнення великих виробничих аварій і катастроф на об'єктах та розглянуті заходи щодо ліквідації цих наслідків.

РОЗДІЛ 6 ЕКОЛОГІЯ

6.1. Актуальність екологічних проблем

Сучасна екологія – це наукова база для розробки тактики та стратегії поведінки людства. Її закони навчають, що людство є частиною природи й своїм існуванням залежить від функціонування природних систем.

Суспільству потрібно усвідомити необхідність орієнтування не на боротьбу з наслідками, які руйнують природу, а на усунення самих причин знищення природи. Закони екології лежать в основі не лише біологічного, а й соціального буття. З розширенням екологічної кризи вони сьогодні стають більш актуальними.

Спричинений політичними, економічними помилками та серйозними екологічними прорахунками теперішній стан природного середовища України оцінюється фахівцями як критичний, коли вже неможливі його самовідновлення й самоочищення: відбуваються активна деградація й небезпечне знищення останніх природних ресурсів. Ліквідація глобальної екологічної кризи є на сьогодні найважливішим завданням людства. Для його вирішення, перш за все, необхідні зміни екологічної стратегії й тактики, організація всебічної екологічної освіти, виховання екологічної свідомості всього населення Землі.

В зв'язку з глобальною екологічною кризою в Україні розпочато екологічну реформу в 1991 році прийняттям закону “Про охорону навколишнього середовища”, було створено відповідний комітет, який згодом реформували в Міністерство екології та природних ресурсів України.

Згідно Законів України “Про охорону навколишнього природного середовища” (від 25.06.91 р. із змінами від 05.03.98...14.12.99 р.) та “Про екологічну експертизу” (від 09.02.95 р.) кожен проект, що розробляється,

необхідно проаналізувати з точки зору негативного впливу на довкілля і відшукати шляхи зменшення цього впливу.

6.2. Етапи та техніка збору та обробки екологічної інформації

Екологічна інформація являє собою сукупність даних про динаміку кількісних та якісних змін стану природних об'єктів довкілля, їх взаємозв'язок і закономірності розвитку. Ця сукупність даних є базою для проведення оцінки екологічного стану навколишнього середовища та прийняття рішень в області екології [20]. Накопичена екологічна інформація створює банки еколого-економічних даних, які є одним із ефективних засобів розуміння законів і закономірностей екологічного стану навколишнього середовища.

Екологічну інформацію про навколишнього середовище та його стан одержують з різних джерел:

1) джерела первинної інформації (результати досліджень через спостереження, експеримент та під час експедицій). Вони є вагомою частиною усього матеріалу;

2) джерела вторинної інформації (це вже зведена інформація про стан здоров'я людей і навколишнього середовища, стан екологічної безпеки господарської діяльності та загалом всі екологічні ситуації на окремих об'єктах і регіонах).

3) джерела науково-теоретичної інформації (таблиці, карти, описи тощо);

4) джерела правової інформації.

Екологічна інформація має різний характер – аналітичний, синтетичний і оперативний.

Аналітичний характер інформації зумовлюється великим об'ємом різнопланових і децентралізованих даних, які мають бути приведені в порівнянний вигляд.

Синтетичний характер інформації відіграє велике значення для глобального впливу на масштабні екосистеми шляхом обліку обставин, які належать до охорони природного середовища та раціонального використання ресурсів.

Оперативний характер виконує попереджувальні функції в різних напрямках підтримки охорони та рівноваги навколишнього середовища, його відтворення.

Потрібно враховувати різні аспекти при зборі і обробці екологічної інформації:

- інерційність інформації;
- вплив фонових факторів;
- новизну і розширення масштабів екологічної статистики;
- багатоступінний збір статистичних даних і нормативних параметрів.

Екологічна інформація є двох типів – первинна та похідна. Первинна інформація міститься в статистичній звітності, яка укладається організаціями та підприємствами, що пов'язані з шкідливим впливом на навколишнє середовище, експлуатацією природних ресурсів чи здійсненням природоохоронних функцій. Похідна, тобто вторинна екологічна інформація міститься в еколого-економічному паспорті підприємства чи організації.

Екологічні дослідження вимагають постійного дотримання чотирьох послідовних етапів:

1. Спостереження.
2. Формулювання вже на основі спостережень теорії про закономірність досліджуваного явища.
3. Перевірка теорії наступними спостереженнями.
4. Спостереження за тим чи є правдивими передбачення, які базуються на цій теорії.

Всі факти, які відносяться до конкретного питання, називають даними. Факти базуються на прямих або непрямих спостереженнях. Спостереження можуть бути якісними (смак, колір, форма, вигляд тощо) або кількісними

(точніші). Внаслідок спостережень отримують так званий “сирий матеріал”, на основі якого вже формулюється гіпотеза.

Гіпотеза – це науково обґрунтоване припущення, що базується на спостереженнях, за допомогою якого можна пояснити те чи інше явище.

Системний підхід є методологічною базою екологічної статистики.

В екології найпоширенішими є польові біометричні методи та експерименти. Збір екологічної інформації здійснюється за допомогою різних методів:

1. Ландшафтно-екологічний підхід (він дає змогу виділяти природні, штучні біогеоценози, прогнозувати сукцесії, досліджувати їх генезис, здійснювати екологічний моніторинг).

2. Ландшафтно-індикаційні спостереження (виконуються для виявлення характерних зовнішніх особливостей місцевості).

3. Польовий метод (здійснюються фенологічні спостереження, агрохімічні, агрофізичні, мікробіологічні дослідження ґрунтів, ботанічні, фізіологічні та біохімічні дослідження рослин).

4. Біохімічні спостереження (виконуються для вивчення речовинного складу рослинності).

5. Гідрохімічні спостереження (здійснюються для вивчення підземних вод).

6. Гідрогеологічні спостереження (проводяться для вивчення гідродинамічних, гідрохімічних і гідрофізичних особливостей стану підземних вод за допомогою природних джерел, криниць і гідрогеологічних свердловин).

7. Ґрунтово-газові спостереження (виконуються для дослідження активних зон тектонічних порушень, вивчення летючих забруднювачів, техногенних забруднень вуглеводами підземних вод чи порід).

8. Геохімічні спостереження ландшафтів (здійснюються для вивчення геохімічних характеристик різних компонентів навколишнього середовища).

9. Радіоекологічні спостереження (виконуються для на відбору проб ґрунту, повітря, води та біоти).

10. Дистанційні спостереження (дають змогу отримувати інформацію про стан окремих компонентів природного середовища і їх перетворення під впливом техногенезу, активності геологічних процесів тощо).

Полевий експеримент є практично неконтрольованим через неймовірну кількість природних факторів, які діють на об'єкт. Лабораторний експеримент є життєво контрольованим. У цьому і полягає між ними основна різниця. Екологічний експеримент є ефективним лише в поєднанні з третім методом екології - методом моделювання.

6.3. Висновки до розділу

У даному розділі розглянуто основні принципи збору, обробки екологічної інформації. Проведено аналіз важливості дотримання чинних вимог законодавства. Для прийняття обґрунтованих управлінських рішень в галузі екології та охорони навколишнього середовища велике значення має створення бази банків даних екологічної інформації. Ліквідація глобальної екологічної кризи на сьогодні є найважливішим завданням людства. Для його вирішення, перш за все, необхідні зміни екологічної стратегії й тактики, організація всебічної екологічної освіти, виховання екологічної свідомості всього населення Землі.

ВИСНОВКИ

У процесі дослідження методів і засобів створення комп'ютерної системи аналізу та візуалізації даних було проаналізовано кліматичні та метеорологічні показники. У дослідженні вперше розглянуто та визначено сильні та слабкі сторони існуючих систем для аналізу кліматичних даних.

На основі отриманих даних для розробки клієнт-серверної аплікації, методами аналізу та узагальнення обрано MS SQL Server для створення бази даних для створення серверної частини обрано C#, мову JavaScript (ECMAScript 6) та базований на ній фреймворк ReactJS v.16.12.0 для клієнта для аплікації. Функціонал взаємодії із API серверної частини реалізовано за допомогою інтерфейсу Fetch API.

У процесі стилізації аплікації та створенні кроссбраузерного і адаптивного рішення, що відповідає сучасним вимогами UI/UX дизайну, було проаналізовано та застосовано CSS фреймворк React Bootstrap v4.0.1.

Для розробки візуалізації отриманих та проаналізованих даних було використано бібліотеку ApexCharts, яка використовує SVG графіку для створення інфографік.

Методом експерименту та узагальнення доведено наукову новизну дослідження, зокрема, особливістю розробленої аплікації є те, що вперше здійснено виконання логічних операцій із аналізу історичних кліматичних показників у клієнтській частині програми, що пришвидшило процес візуалізації інформації для користувача у порівнянні з класичним підходом до розробки клієнт-серверних програм, де основна бізнес-логіка зосереджена на серверній частині.

У дослідженні обґрунтовано економічну ефективність впровадження комп'ютерної системи аналізу та візуалізації кліматичних даних. На основі аналізу вимог з охорони праці, техніки безпеки та екології визначено шляхи мінімізації негативного впливу шкідливих виробничих факторів.

БІБЛІОГРАФІЯ

1. С.І. Решетченко. Метеорологія та кліматологія: навчальний посібник. Х.: ХНУ імені В. Н. Каразіна, 2015. 220 с.
2. Метеорологія і кліматологія: Навчально-методичний посібник. Державний вищий навчальний заклад НЛТУ України. Укладач С.П. Мельничук. Львів: ННЛТУ України, 2018. 148 с.
3. Володарський Є., Кошева Л. Статистична обробка даних. К: НАУ-друк, 2008. 308 с.
4. С. Donald Ahrens and Others. Meteorology Today. Cengage Learning, 2016. 662 p.
5. Флэнаган Д. JavaScript: карманный справочник. Диалектика-Вильямс, 2016. 320 с.
6. Никсон Робин. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. Питер, 2016. 768 с.
7. Бенкс А., Порселло Е. React и Redux. Функциональная веб-разработка. Питер, 2016. 336 с.
8. Рихтер Дж. P55 CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C#. СПб.: Питер, 2013. 896 с.
9. Гладун О. Візуалізація інформації: інфографіка. Ольга Гладун. Вісник ХДАДМ, 2012, №4. С.11-14.
10. Методичні вказівки щодо виконання економічної частини дипломного проекту для студентів денної форми навчання зі спеціальності 7.091501 – “Комп’ютерні системи і мережі” В.О. Мосьпан, І.І. Постіл. Кременчук: Видавничий відділ КДУ ім. М. Остроградського, 2013. 39 с.
11. Розмір єдиного внеску в залежності від класу професійного ризику URL: <http://buhgaltera.net.ua/rozmir-edinogo-vnesku-v-zalezhnosti-vid-klasu-profesijnogo-riziku/> (дата звернення: 16.10.2019).
12. Кулаков М.А., Ляпун В.О., М’який В.О. та ін. Цивільна оборона: навчальний посібник. Харків: Факт, 2008.

13. Шоботов В.М. Цивільна оборона: навчальний посібник. К.: Центр навчальної літератури, 2004.
14. Норми визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною безпекою. НАПБ Б.03.002-2007. (затверджено наказом МНС України від 03.12.2007 № 833).
15. ДСН 3.3.6.042-99. Санітарні норми мікроклімату виробничих приміщень [Текст]. К., 2000. 16 с.
16. Гігієнічна класифікація праці за показниками шкідливості та небезпечності факторів виробничого середовища, важкості та напруженості трудового процесу (затв. наказом МОЗ України від 12.08.2014 р. № 248).
17. Правила охорони праці під час експлуатації електронно-обчислювальних машин. НПАОП 0.00-1.28-10 (затверджено наказом Державного комітету України з промислової безпеки, охорони праці та гірничого нагляду від 26.03.2010 р. № 65).
18. Гігієнічна класифікація праці за показниками шкідливості та небезпечності факторів виробничого середовища, важкості та напруженості трудового процесу (затверджено наказом МОЗ України від 08.04.2014 № 248).
19. Русаловський А. В. Правові та організаційні питання охорони праці: Навч. посіб. К.: Університет “Україна”, 2009. 295 с.
20. Тарасова В.В. Екологічна статистика. К.: Центр учбової літератури, 2008. 392 с.

Додаток А
Тези конференцій