

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
«ПОЛТАВСЬКА ПОЛІТЕХНІКА ІМЕНІ ЮРІЯ КОНДРАТЮКА»**

**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ ТА РОБОТОТЕХНІКИ**

**КАФЕДРА КОМП'ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І
СИСТЕМ**

**КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА
спеціальність 122 «Комп'ютерні науки»**

на тему:

**«Розробка мобільного додатку для збирання та аналізу метеорологічної
інформації»**

Студента групи 601-ТН Поспелова Іллі Сергійовича

Керівник роботи
кандидат технічних наук,
доцент Здоренко Ю.М.

Завідувач кафедри
кандидат технічних наук,
доцент Двірна О.А.

РЕФЕРАТ

Кваліфікаційна робота магістра: 62 с., 10 рисунків, 4 додатки, 29 джерел.

Об'єкт дослідження: процес збору та аналізу метеорологічної інформації.

Мета роботи: удосконалення процесу збору та аналізу метеорологічної інформації шляхом використання програмних рішень.

Методи: системний аналіз, метод аналогій, інформаційне моделювання, абстрагування, порівняння, групування.

Ключові слова: мобільний додаток, метеорологічний аналіз, інтерактивний інтерфейс, кліматичні дані, візуалізація даних.

Результати роботи апробовані в рамках наукової конференції здобувачів вищої освіти за результатами щорічної студентської наукової конференції в 2021 році, XVIII щорічного міждисциплінарного семінару «Студентські роботи за науковою тематикою кафедри інформаційних систем та технологій» в 2021 році, The III International Scientific and Practical Conference «Education and Science of Today: Intersectoral Issues and Development of Sciences», 2022.

ANNOTATION

Qualification work of master's degree: 62 p., 10 pictures, 4 applications, 29 sources.

Object of study: the process of collecting and analyzing meteorological information.

The goal of the work: improvement of the process of collecting and analyzing meteorological information by using software solutions.

Methods: system analysis, method of analogies, information modeling, abstraction, comparison, grouping.

Keywords: mobile application, meteorological analysis, interactive interface, climate data, data visualization.

The results of the work were tested as part of the scientific conference of higher education seekers based on the results of the annual student scientific conference in 2021, the XVIII annual interdisciplinary seminar "Student works on scientific topics of the Department of Information Systems and Technologies" in 2021, The III International Scientific and Practical Conference "Education and Science of Today: Intersectoral Issues and Development of Sciences", 2022.

ЗМІСТ

ВСТУП.....	5
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ЗАСОБІВ АВТОМАТИЗОВАНОГО ЗБИРАННЯ ТА АНАЛІЗУ МЕТЕОРОЛОГІЧНОЇ ІНФОРМАЦІЇ З ВИКОРИСТАННЯМ ПРОГРАМНИХ РІШЕНЬ	8
1.1 Аналіз основних понять та предмету дослідження метеорології	8
1.2 Представлення метеорологічної інформації	11
1.3 Класифікація програм метеорологічної інформації	13
1.4 Огляд основних типів програмних продуктів взаємодіючих із метеоданими на прикладі існуючих на ринку популярних рішень	14
1.5 Загальна постановка задачі дослідження	20
РОЗДІЛ 2 РОЗРОБКА АРХІТЕКТУРИ МОБІЛЬНОЇ ВЕРСІЇ ПРОГРАМНОГО ДОДАТКУ ДЛЯ ЗБИРАННЯ ТА АНАЛІЗУ МЕТЕОРОЛОГІЧНОЇ ІНФОРМАЦІЇ. ВИБІР СПОСОБІВ ДЛЯ ЙОГО РОЗГОРТАННЯ.	22
2.1 Проектування архітектури додатку	22
2.2 Розробка компонентів і взаємодії у додатку	26
2.3 Методи розгортання додатку	34
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ДОДАТКУ ТА ОЦІНКА ЕФЕКТИВНОСТІ ЗАПРОПОНОВАНОГО РІШЕННЯ.	38
3.1 Концептуальна модель додатку	38
3.2 Опис функціональності додатку	40
3.3 Оцінка ефективності запропонованого рішення.....	44
ВИСНОВКИ	46
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	48
ДОДАТОК А ПОРІВНЯЛЬНА ХАРАКТЕРИСТИКА БАЗ ДАНИХ	51
ДОДАТОК Б ДІАГРАМА ПРЕЦЕДЕНТІВ	52
ДОДАТОК В ДІАГРАМА ПОСЛІДОВНОСТЕЙ.....	53

ДОДАТОК Г ПУБЛІКАЦІЇ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ ЗА МАТЕРІАЛАМИ КВАЛІФІКАЦІЙНОЇ РОБОТИ	54
---	----

ВСТУП

Сучасний розвиток інформаційних систем і технологій дозволив автоматизувати процеси збору, обробки, зберігання та візуалізації різноманітної метеорологічної інформації. Метеорологічні дані мають вагоме значення для людської діяльності, що зумовило появу численних прикладних напрямків, таких як агрометеорологія, біометеорологія, радіометеорологія, супутникова та авіаційна метеорологія тощо.

Аграрний сектор заслуговує на особливу увагу, адже кліматичні умови є ключовими факторами, що впливають на продуктивність сільськогосподарських культур. Збір та аналіз агрометеорологічних даних дозволяють здійснювати точне прогнозування, від якого залежить оптимізація зрошення, визначення норм і методів внесення добрив, прогнозування поширення хвороб рослин, строки боротьби зі шкідниками, планування сівби, оцінка умов перезимівлі озимих культур та багато інших технологічних процесів. Коректне виконання цих заходів суттєво впливає на кінцеву урожайність і економічну ефективність виробництва.

Збір даних може здійснюватися за допомогою стаціонарних і локальних метеорологічних станцій, мобільних датчиків, супутникових знімків та інших засобів. Для сільськогосподарських потреб основну роль відіграють станції локального та стаціонарного типу, які здатні збирати значний обсяг даних у великому радіусі. Це дозволяє забезпечити більш точні оцінки та прогнози стосовно конкретних полів або земельних ділянок.

Отже, розробка програмного продукту, який дозволяє відстежувати метеорологічні показники в будь-яких точках країни з урахуванням їхнього історичного аналізу, а також зручно та наочно представляти ці дані у вигляді графіків, порівняльних таблиць і готових аналітичних звітів, є надзвичайно актуальною задачею.

Об'єктом дослідження виступає процес збору та аналізу метеорологічної інформації.

Метою роботи є удосконалення процесу збору та аналізу метеорологічної інформації шляхом використання програмних рішень.

Для досягнення поставленої мети необхідно було виконати ряд завдань:

- аналіз існуючих технологій для створення інформаційних систем та засобів візуалізації інформації зі збору та аналізу метеорологічної інформації на основі використання веб-технологій, обґрунтування необхідності створення інформаційної системи для збору та аналізу метеорологічної інформації;

- формування вимог до апаратного та програмного забезпечення для створення інформаційної системи збору та аналізу метеорологічної інформації на основі веб-технологій;

- розробка архітектури програмної компоненти інформаційної системи, вибір необхідних модулів, розробка алгоритму, структурної схеми додатку;

- програмна реалізація мобільного додатку інформаційної системи збору та аналізу метеорологічної інформації з використанням веб-технологій та створення керівництва для розгортання та роботи з розробленим програмним рішенням;

- оцінка ефективності запропонованого рішення.

Предметом дослідження є програмні платформи на основі веб-технологій для створення інформаційних систем з автоматизованого збору та аналізу метеорологічної інформації.

Методи дослідження:

- системний аналіз – для встановлення структурних зв'язків між змінними або елементами досліджуваної системи;

- метод аналогій – для встановлення відношення еквівалентності між двома системами, що розглядаються, за деякими ознаками;

- інформаційне моделювання: уявне, матеріальне – для забезпечення дослідження інформаційно довідковими матеріалами;

- абстрагування – з метою нехтування несуттєвими властивостями предметів і виявленні найважливіших рис предмета дослідження;
- порівняння – для зіставлення даних у динаміці.

Практична значущість полягає у створенні додатку для збору та аналізу метеорологічної інформації. Такий додаток може використовуватися науковими та навчальними установами, а також підприємствами різних форм власності для узагальнення даних і їх подальшого застосування в аналізі та прогнозуванні природних і виробничих процесів у різних галузях народного господарства.

Результати роботи апробовані в рамках наукової конференції здобувачів вищої освіти за результатами щорічної студентської наукової конференції в 2021 році, XVIII щорічного міждисциплінарного семінару «Студентські роботи за науковою тематикою кафедри інформаційних систем та технологій» в 2021 році, The III International Scientific and Practical Conference «Education and Science of Today: Intersectoral Issues and Development of Sciences», 2022.

Обсяг та структура роботи. Кваліфікаційна робота складається із вступу, трьох розділів основної частини, висновків, списку використаних джерел та додатків. Вона викладена на 62 сторінках формату А4 та ілюстрована 10 рисунками. В роботі використано 29 науково-технічних джерел.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ЗАСОБІВ АВТОМАТИЗОВАНОГО ЗБИРАННЯ ТА АНАЛІЗУ МЕТЕОРОЛОГІЧНОЇ ІНФОРМАЦІЇ З ВИКОРИСТАННЯМ ПРОГРАМНИХ РІШЕНЬ

1.1 Аналіз основних понять та предмету дослідження метеорології

У наш час більшість даних про атмосферні процеси, клімат і погоду отримують через спостереження. Отримані дані використовуються для встановлення причинно-наслідкових зв'язків, що виникають в атмосфері. Результати спостережень включають інструментальні вимірювання числових значень метеорологічних параметрів та їхніх змін, а також візуальні оцінки якісних характеристик атмосферних явищ. Зібрані дані обробляються за допомогою спеціальних правил і алгоритмів, що дозволяє створювати прогнози на майбутнє [1].

Для збору та відображення узагальнених метеорологічних даних використовуються спеціалізовані програми. Стандартна програма метеорологічних спостережень охоплює реєстрацію таких параметрів: температура повітря, атмосферний тиск, відносна вологість повітря, швидкість і напрямок вітру, хмарність, кількість та типи опадів (дощ, сніг, град тощо), наявність і інтенсивність опадів, що утворюються на поверхні землі, горизонтальна видимість, кількість сонячних хвилин, температура на поверхні ґрунту і на різних глибинах, стан поверхні ґрунту, щільність і висота снігового покриву.

Також фіксуються різні атмосферні явища, зокрема: метеорологічні (хуртовини, шквали, смерчі, імла, пилові бурі), електричні (грози, тихі електричні розряди, полярні сяйва), оптичні (райдуга, гало, вінця навколо світил, міражі тощо). Ці дані є основою для аналізу та прогнозування змін у атмосфері.

Метеорологія — це наука про атмосферу, яка вивчає її склад, властивості, а також фізичні та хімічні процеси, що в ній відбуваються. Вона є частиною ширшої науки — геофізики, яка досліджує явища та процеси в атмосфері, на поверхні суші та в товщі ґрунту [2].

Через складність досліджень і потреби різних галузей було виділено декілька спеціалізованих напрямків метеорології:

- синоптична метеорологія — досліджує закономірності атмосферних процесів для їх прогнозування;
- динамічна метеорологія — вивчає сили, які діють в атмосфері, та їхній рух;
- актинометрія — аналізує енергію Сонця та Землі, її вплив на температурний режим, хмарність, погоду й клімат;
- кліматологія — вивчає закономірності формування клімату на Землі та в окремих регіонах.

Прикладні напрямки метеорології:

- агрометеорологія — оцінює вплив погодних умов на сільське господарство;
- авіаційна метеорологія — забезпечує прогнозування погодних умов для авіації;
- супутникова метеорологія — використовує дані з супутників для аналізу погоди й клімату.

Агрометеорологія має ключове значення для сільськогосподарського виробництва, зокрема:

- дослідження взаємозв'язків між погодними умовами та врожайністю культу;
- аналіз гідрометеорологічних умов у просторі й часі;
- розробка методів оцінки впливу погоди на стан ґрунтів, ріст і розвиток рослин, а також поширення шкідників і хвороб;
- розробка методів агрометеорологічних прогнозів;

- агрокліматичне районування та впровадження нових сортів культур і порід тварин;
- обґрунтування заходів меліорації земель і зміни мікроклімату;
- впровадження індустріальних технологій у рослинництві;
- дослідження попиту на агрометеорологічну інформацію, популяризація знань у цій сфері.

Ці завдання допомагають адаптувати аграрне виробництво до змін клімату та підвищити ефективність його технологічних процесів.

Ці та інші завдання вирішуються агрометеорологією, спрямованою на вдосконалення усіх форм оперативного агрометеорологічного забезпечення аграрного сектору країни інформацією [3].

Не менш важливою є авіаційна метеорологія, яка має тісний зв'язок із загальною метеорологією. Це прикладна галузь, що вивчає вплив метеорологічних умов на авіаційну техніку та діяльність авіації в цілому. Вона також розробляє теоретичні та методичні основи для метеорологічного забезпечення польотів [4].

Основні завдання авіаційної метеорології включають:

- забезпечення економічності повітряних перевезень;
- гарантування безпеки польотів;
- забезпечення регулярності авіасполучення.

Ці напрями досліджень є важливими для розвитку як аграрного, так і транспортного секторів, адже від якості метеорологічного забезпечення залежить ефективність і безпека їхньої діяльності.

Важливість супутникової метеорології важко переоцінити, адже завдяки цій галузі, яка розробляє методи отримання і використання метеорологічної інформації за допомогою апаратури, встановленої на метеорологічних супутниках, стало можливим глибше досліджувати структуру та розподіл хмарного покриву Землі, а також визначати температуру хмарної поверхні.

Дослідження великомасштабних структур хмарного покриву і встановлення їх зв'язку з процесами формування погоди стали основою для

спутникового аналізу хмарності. Цей аналіз включає дешифрування зображень хмарності з метою отримання більш точного уявлення про синоптичну ситуацію. Супутникові зображення є особливо цінними для своєчасного розпізнавання, прогнозування та вивчення штормів і ураганів, моніторингу крижаного та снігового покривів, аналізу кліматичних змін. Супутникова метеорологія значно підвищила точність прогнозів і забезпечила ефективніший моніторинг глобальних атмосферних процесів [5].

Значущість метеорологічних даних залишається великою як для окремих регіонів, так і для всієї країни в цілому. Актуальним є питання взаємозв'язку гідрометеорологічних вимірювань з глобальними кліматичними процесами та подальшого прогнозування небезпечних стихійних явищ. Прогнозування погоди є одним із найскладніших завдань фізики атмосфери, і на сьогодні метеорологічні служби використовують різноманітні методи для збору погодної інформації. Це включає використання повітряних куль з вимірювальним обладнанням, супутників, наземних станцій та інших технологій. Ці способи дозволяють отримувати дані, які є необхідними для прогнозування змін в атмосфері, а також для своєчасного попередження стихійних лих і забезпечення безпеки.

У наш час існує безліч методів прогнозування метеорологічних явищ, серед яких можна виділити кількісні, синоптичні та статистичні підходи. Однак жоден з них, навіть у поєднанні, не гарантує абсолютно точного прогнозу. Рішення таких завдань полягає у необхідності обробки великих обсягів даних, що вимагає використання сучасних обчислювальних засобів та програмного забезпечення. Це дозволяє зменшити похибки прогнозів і підвищити їхню точність, зокрема шляхом інтеграції різних методів та технологій для більш детального аналізу атмосферних процесів [6].

1.2 Представлення метеорологічної інформації

Правильність аналізу атмосферних процесів та успішність подальших прогнозів значною мірою залежать від якості первинної метеорологічної інформації. Первинна інформація включає дані про поточний стан погоди, отримані під час метеорологічних спостережень. Крім того, існує вторинна інформація, яка включає дані у вигляді синоптичних карт, вертикальних розрізів, аерологічних діаграм та інших структурованих відомостей.

Для забезпечення точності прогностичних моделей, первинна метеорологічна інформація повинна відповідати кільком вимогам, зокрема:

- глобальність, тобто охоплення всіх необхідних територій;
- тривимірність, яка дозволяє враховувати вертикальні та горизонтальні зміни в атмосфері;
- комплексність, що забезпечує збирання різноманітних параметрів погоди;
- синхронність, тобто одночасний збір даних із різних джерел;
- регулярність, що передбачає збір інформації у встановлені інтервали;
- оперативність, що дає змогу швидко отримувати та обробляти дані для актуальних прогнозів.

Системи збору метеорологічної інформації постійно удосконалюються на міжнародному та національному рівнях. До таких систем відносяться гідрометеорологічні та метеорологічні радіолокаційні станції, авіаційна розвідка погоди, буйкові морські станції та інші технології для збору, аналізу та прогнозування метеорологічних явищ [7].

Сучасний розвиток інформаційних систем і технологій дозволяє значно автоматизувати процеси аналізу, збору, зберігання та представлення метеорологічної інформації. Розроблені технічні комплекси забезпечують автоматичну обробку вхідних метеоданих, що дозволяє ефективно управляти великими обсягами інформації. Отримані дані поділяються на різні типи і проходять через форматний та логічний контроль для виявлення можливих помилок. Після цього дані перетворюються в зручний для подальшого аналізу вигляд і зберігаються в базі даних системи, яка регулярно оновлюється з

урахуванням нової інформації. Такий підхід дозволяє значно підвищити точність і оперативність прогнозів, а також забезпечити ефективне управління метеорологічними даними.

Під час створення систем для обробки метеорологічних даних активно використовуються основні математичні методи для прогнозування погоди. До таких методів відносяться метод лінійної регресії, метод за місцевими ознаками, а також метод професора Броунова, який застосовується для прогнозування опадів, хмарності, температури та заморозків. Поповнення бази даних метеорологічними показниками відбувається через підключення програми до системи отримання первинної інформації, що дозволяє автоматично оновлювати дані. Подальше використання цієї метеорологічної бази є ефективним лише за умови достатньої якості наявних даних, зокрема, їх однорідності та мінімальної кількості грубих помилок. Якість даних прямо впливає на точність прогнозів та можливість їх використання для прийняття важливих рішень у різних сферах, таких як сільське господарство, авіація та інші галузі, що залежать від погодних умов [8].

1.3 Класифікація програм метеорологічної інформації

Програми, що займаються збором, обробкою та збереженням метеорологічної інформації, мають широкий спектр напрямків, що залежать від цілей використання програмного продукту. Для комерційних програм характерний зручний інтерфейс і велика кількість метеорологічних показників, а також можливість прогностичних оцінок на місяць уперед. Однак для наукових досліджень таких даних може бути недостатньо. Тому існують вузькоспеціалізовані програми, які надають максимально точні та актуальні метеорологічні дані, зібрані відповідно до конкретної ситуації.

Програми для розрахунку кліматичних моделей і географічних зображень використовуються для глобальних кліматичних та метеорологічних прогнозів з

урахуванням спостережень по всій Землі, часто із значним історичним терміном. Це дозволяє оцінювати та прогнозувати основні кліматичні тренди, що допомагає оптимізувати діяльність людини в різних регіонах, а також враховувати можливі зміни клімату.

Широко застосовуються програми прогнозування кліматичних умов для транспортних систем, зокрема авіації та морських перевезень. Ці програми розраховують оптимальні маршрути для транспорту і попереджають про небезпечні атмосферні явища, такі як фронти, можливі урагани, шторми чи інші природні катастрофи, які можуть загрожувати безпеці перевезень.

Останнім часом активно розвиваються програми для прогнозування метеоумов у сільському господарстві, що базуються на актуальних та історичних метеоданих. Ці програми допомагають створювати моделі для прогнозування оптимальних строків обробітку ґрунту, сівби, догляду за посівами і збору врожаю, а також попереджають аграріїв про небезпечні кліматичні явища, такі як посухи, зливи, приморозки чи пилові бурі.

1.4 Огляд основних типів програмних продуктів взаємодіючих із метеоданими на прикладі існуючих на ринку популярних рішень

Аналіз програмних продуктів, які мають подібну функціональність або належать до однієї предметної області, є важливим етапом у плануванні та розробці нової системи. Такий аналіз дозволяє оцінити існуючі рішення з точки зору їх ефективності, зручності використання та виявлення можливих недоліків у реалізації. Завдяки такому аналізу можна визначити сильні та слабкі сторони різних продуктів, що дозволить уникнути повторення помилок та адаптувати найкращі підходи в новій системі.

Збір даних про конкурентні програмні продукти дозволяє спроектувати більш ефективну архітектуру майбутнього додатку, а також запровадити цікаві та корисні функції, які підвищать зручність і продуктивність користування. Хороший аналіз також допомагає визначити потреби користувачів, на яких

орієнтований продукт, що дозволяє створити додаток, який буде не лише функціональним, але й зручним для кінцевих користувачів.

1.4.1 Meteoprog. Проєкт, який надає детальну інформацію про погодні умови та прогнози погоди в режимі реального часу, має значну актуальність для користувачів, які потребують точних і своєчасних погодних даних для планування своєї діяльності. Прогнозні карти, метеограми та таблиці, представлені на такому ресурсі, забезпечують користувачів повною картиною метеорологічної ситуації в їхніх населених пунктах на найближчі дні чи тижні.

Такі ресурси здебільшого є прогностичними, тобто вони спрямовані на прогнозування погодних умов на основі поточних та минулих даних. Вони зазвичай орієнтовані на широке використання в повсякденному житті для отримання актуальної інформації про погоду в певному місці, що дозволяє людям планувати свою діяльність із урахуванням майбутніх погодних змін.

Для розрахунку прогнозів погоди на таких платформах використовуються числові моделі прогнозу погоди, такі як «Weather Research and Forecasting» (WRF) [9]. Ця модель є однією з найсучасніших та найпоширеніших у метеорології, дозволяючи створювати точні прогнози з високим рівнем деталізації. Використання потужних обчислювальних ресурсів, таких як високопродуктивний комп'ютерний кластер, забезпечує обробку великих обсягів даних, що необхідні для створення точних прогнозів на різних масштабах.

До основних функцій Meteoprog можна віднести наступні:

- перегляд прогнозу погоди на певний період, від 1 до 30 днів починаючи з поточного дня, відповідно вибору користувача, можливість погодинно переглянути прогноз на наступні 3 дні;
- перегляд погодних показників за допомогою зображення їх на інтерактивних мапах;

- наявність конструктора, що надає можливість створення власного інформера та його подальше використання на своїх ресурсах.

Присутня вкладка новин, де можна прочитати статті на теми, пов'язані з погодою. Інтерфейс Meteorog виглядає сучасним і зручним, з інтуїтивно зрозумілим розташуванням контенту та кнопок. Сайт також має мобільну версію. На головній сторінці легко переглядати прогноз погоди, графіки температури, опадів, тиску та температуру по містах України. Головну сторінку розглянутого ресурсу можна переглянути на рис. 1.1.

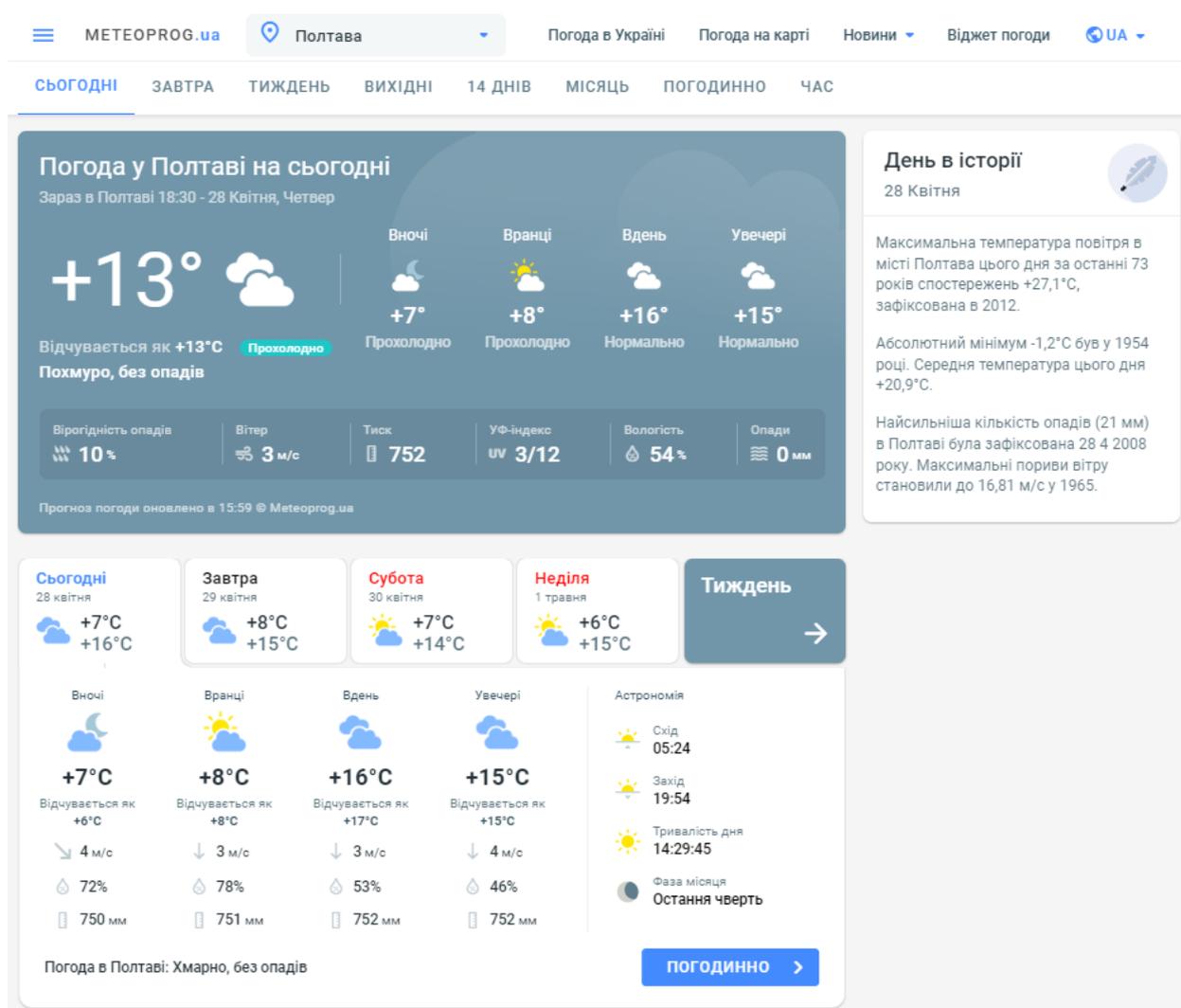


Рисунок 1.1 – Головна сторінка Meteorog.ua

1.4.2 WeeWX. Безкоштовна програмна платформа з відкритим кодом, написана на мові програмування Python, яка дозволяє інтегрувати персональні

метеостанції з різними типами пристроїв для збору, обробки та візуалізації метеорологічних даних. Основною метою цієї системи є створення графіків, звітів та формування HTML-сторінок для публікації погодної інформації. Платформа підтримує інтеграцію з більшістю версій операційних систем, зокрема Linux, BSD, Solaris та macOS, що робить її доступною для широкого кола користувачів [10].

WeeWX особливо корисна для тих, хто використовує персональні метеостанції для збору точних погодних даних в реальному часі. Це дозволяє створювати персоналізовані погодні звіти з різних показників, таких як температура повітря, вологість, атмосферний тиск, швидкість вітру та інші. Платформа пропонує безліч налаштувань для публікації даних на різних погодних сервісах або на власному вебсервері. Користувачі можуть інтегрувати дані на своїх вебсайтах для моніторингу актуальної ситуації або архівування історичних даних.

Один із цікавих аспектів — це гнучкість налаштувань, які дозволяють користувачам детально налаштувати вигляд і формат отримуваних метеоданих. Платформа також дозволяє інтегруватися з різними зовнішніми сервісами, які надають готові панелі для моніторингу, керування та архівування даних. Одним із таких сервісів є PWSweather, який безкоштовно надає свої можливості для публікації погодних даних в обмін на використання інформації, що надходить з метеостанцій своїх користувачів. Цей сервіс використовує ці дані для збагачення своєї бази і подальшого аналізу, сприяючи таким чином створенню більш точної та надійної погодної інформації для великої кількості користувачів.

За допомогою WeeWX користувачі можуть не лише отримувати актуальні дані, але й створювати власні прогнози, зберігати архіви даних для подальшого аналізу або надавати ці дані для більш ширшого використання в рамках громади метеорологів або ентузіастів. Це робить платформу потужним інструментом для досліджень та моніторингу погодних умов, а також ідеальним

рішенням для тих, хто хоче поглиблено працювати з метеорологічними даними на локальному рівні.

Ключовими особливостями WeeWX є підтримка багатьох популярних апаратних рішень, проста інсталяція для користувачів Debian (Ubuntu) і Red Hat (CentOS), можливість роботи з базами даних sqlite або MySQL. Система дозволяє створювати чи змінювати зовнішній вигляд ресурсу, підтримує локалізацію та використовує розширювану систему шаблонів. WeeWX забезпечує фільтрацію аномальних значень, внесення поправок калібрування датчиків метеостанції, а також має розширювану архітектуру мікроядра. Крім того, існує можливість доповнення системи новими сервісами.

1.4.3 Meteo Farm. Спеціалізований метеосервіс, який надає користувачам детальну інформацію про агропогодні умови, орієнтуючись на конкретні поля, визначені самими користувачами. Цей сервіс дозволяє аграріям отримувати актуальні метеорологічні дані, що постійно оновлюються, завдяки підключенню до системи власних метеостанцій, що встановлені безпосередньо на земельних ділянках [11].

Основною перевагою Meteo Farm є можливість точно моніторити погодні умови на окремих полях, що дає можливість аграріям оперативно реагувати на зміни кліматичних умов, відстежувати температуру, вологість, опади, атмосферний тиск, швидкість вітру та інші фактори, що впливають на стан сільськогосподарських культур. Система дає змогу отримати доступ до потужної аналітичної бази, яка включає не лише поточні погодні дані, але й прогнози на найближчий період, а також історичну інформацію, що дозволяє робити висновки про тенденції і особливості кліматичних умов для кожної конкретної місцевості.

Завдяки цьому сервісу користувачі можуть точніше планувати агрономічні роботи, такі як сівба, обробка ґрунту, догляд за посівами та збирання врожаю, оскільки мають доступ до більш точних і локалізованих метеорологічних даних. Крім того, Meteo Farm також дозволяє попереджати

про небезпечні погодні явища, такі як посухи, зливи чи приморозки, що є надзвичайно важливим для планування і прогнозування в сільському господарстві.

Головними особливостями є можливість додавання власного поля в будь-якій точці країни, прогнозування погодних умов на 7 днів, починаючи з поточного, та відстеження фактичних показників метеостанцій поблизу поля. Система дозволяє калькулювати суму активних і ефективних температур, а також ГТК, налаштовувати сповіщення про поточний стан поля та зберігати історію показників за наявності поруч метеостанцій компанії. Також можна створювати віджети на основі даних зі створених полів.

Інтерфейс Meteo Farm відрізняється лаконічністю та зручністю у використанні. Вигляд сторінки із показниками поля зображені на рис. 1.2.

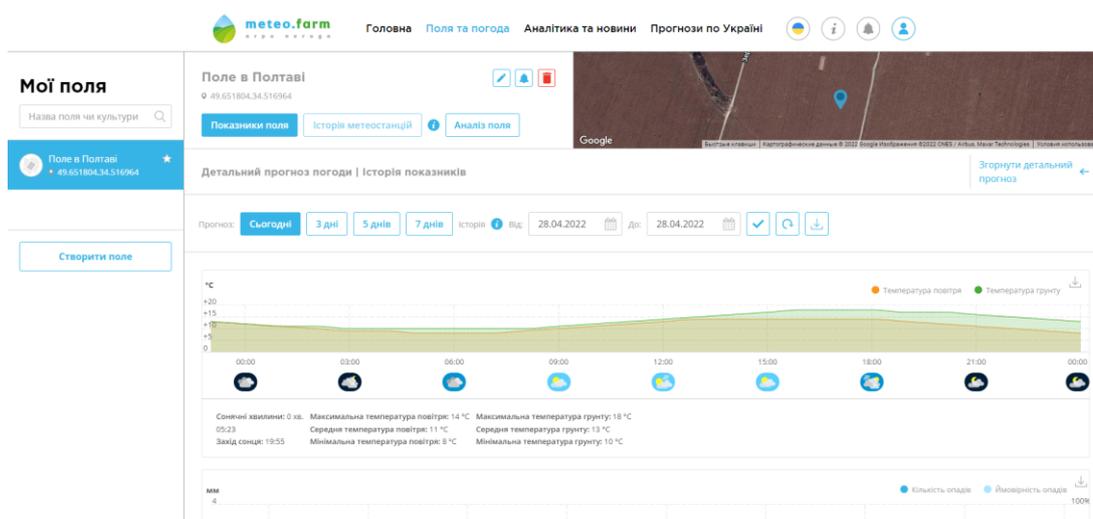


Рисунок 1.2 – Сторінка із показниками з створеного поля

На головній сторінці Meteo Farm користувачі мають можливість переглядати актуальний прогноз погоди, а після реєстрації з'являються додаткові функції. Користувач може налаштувати сповіщення, створити власні віджети та поле, а також переглядати релевантну метеорологічну інформацію, специфічну для їхнього регіону або земельної ділянки. Такий персоналізований підхід дозволяє користувачам отримувати точні прогнози та прогностичні дані в зручному вигляді для прийняття обґрунтованих рішень.

Програми, які взаємодіють з метеостанціями, є необхідними для налаштування збору даних та їх передачі кінцевим користувачам. Це дозволяє аграріям або іншим користувачам безперервно отримувати дані, що стосуються погодних умов, для подальшого використання в роботі або дослідженнях.

Прогностичні ресурси мають великий попит серед звичайних користувачів, які прагнуть отримати точну інформацію про погодні умови для планування своїх повсякденних або професійних дій. Багато ресурсів надають доступ до вже зібраних метеоданих, які часто використовуються для аналізу та прийняття важливих рішень, наприклад, у сільському господарстві, транспорту, енергетиці чи будівництві. В таких випадках метеорологічна інформація є платною, оскільки її збирання та обробка потребує значних ресурсів.

Сервіси, що працюють на найвищому рівні абстракції, як правило, надають користувачам зручний доступ до обробленої метеорологічної інформації, яка вже пройшла необхідний аналіз і об'єднання даних з різних джерел. Це робить такі ресурси корисними для специфічних галузей, де точність і швидкість обробки інформації критично важливі для ефективного прийняття рішень.

1.5 Загальна постановка задачі дослідження

В результаті проведеного аналізу вищезгаданих програмних продуктів можна зробити висновок, що існують різноманітні системи, які мають прямий контакт із метеорологічною інформацією і відрізняються за метою та сферою застосування. Кожен із цих продуктів має чітко визначені цілі та задачі, орієнтовані на конкретні потреби користувачів. На основі цього аналізу була сформульована основна задача дослідження, що полягає у розробці власного програмного продукту для збору та обробки метеорологічної інформації, який відповідає специфічним вимогам користувачів та забезпечує ефективну обробку даних.

Відповідно до розв'язуваної задачі, в роботі необхідно вирішити наступні часткові задачі досліджень:

- аналіз існуючих технологій для створення інформаційних систем та засобів візуалізації інформації зі збору та аналізу метеорологічної інформації на основі використання веб-технологій, обґрунтування необхідності створення інформаційної системи для збору та аналізу метеорологічної інформації;

- формування вимог до апаратного та програмного забезпечення для створення інформаційної системи збору та аналізу метеорологічної інформації на основі веб-технологій;

- розробка архітектури програмної компоненти інформаційної системи, вибір необхідних модулів, розробка алгоритму, структурної схеми додатку;

- програмна реалізація мобільного додатку інформаційної системи збору та аналізу метеорологічної інформації з використанням веб-технологій та створення керівництва для розгортання та роботи з розробленим програмним рішенням;

- оцінка ефективності запропонованого рішення.

Об'єктом дослідження виступає процес збору та аналізу метеорологічної інформації.

Предметом дослідження є програмні платформи на основі веб-технологій для створення інформаційних систем з автоматизованого збору та аналізу метеорологічної інформації.

Таким чином, у цьому розділі було здійснено аналіз основної наукової, методичної та нормативної літератури, що стосується організації метеорологічних досліджень, а також проведено загальне порівняння основних видів програмних продуктів, які взаємодіють із метеорологічними даними. На основі цього аналізу було визначено основні напрямки для подальшого дослідження та сформульовано чітке завдання для розробки відповідних рішень у цій сфері.

РОЗДІЛ 2

РОЗРОБКА АРХІТЕКТУРИ МОБІЛЬНОЇ ВЕРСІЇ ПРОГРАМНОГО ДОДАТКУ ДЛЯ ЗБИРАННЯ ТА АНАЛІЗУ МЕТЕОРОЛОГІЧНОЇ ІНФОРМАЦІЇ. ВИБІР СПОСОБІВ ДЛЯ ЙОГО РОЗГОРТАННЯ.

2.1 Проєктування архітектури додатку

Додаток реалізовано за принципами клієнт-серверної архітектури. У цій обчислювальній моделі сервер забезпечує зберігання, постачання та управління ресурсами й послугами, якими користується клієнт. На відміну від традиційного підходу, де сервер надсилає клієнту готові HTML-сторінки з вбудованими даними, наш додаток використовує сучасний метод обміну інформацією: сервер передає дані у форматі JSON-об'єктів. Це дозволяє клієнтській частині самостійно визначати спосіб представлення отриманих даних, забезпечуючи більшу гнучкість у відображенні інтерфейсу.

Загальний дизайн додатка відповідає принципам SPA (Single Page Application), які є сучасною еволюцією клієнт-серверної архітектури. Основна відмінність полягає в тому, що повне оновлення сторінки браузера не відбувається [12]. У SPA презентаційна логіка повністю переноситься на клієнтську частину, а сервер відповідає лише за передачу даних. Такий підхід значно оптимізує швидкодію додатка та зменшує обсяг трафіку між клієнтом і сервером.

SPA — це тип вебдодатків, у якому більша частина коду завантажується на початку роботи, а всі наступні оновлення інтерфейсу здійснюються динамічно, без перезавантаження сторінки. Це дозволяє економити час і ресурси, оскільки однакові елементи інтерфейсу не завантажуються повторно, що значно покращує користувацький досвід.

Додатково, використання SPA сприяє кращій продуктивності завдяки тому, що основна частина бізнес-логіки виконується на стороні клієнта. Це

дозволяє створювати інтерактивні додатки, які швидко реагують на дії користувачів.

Головними особливостями SPA є:

- універсальність. Використовувати вебдодаток можна практично з будь-якого пристрою, якщо на ньому є доступ до інтернету;
- можливість залучення великих обсягів даних. Розмір програми й використовувані нею дані не обмежені пам'яттю пристрою;
- SPA легко налагоджувати за допомогою Chrome, оскільки є можливість контролювати роботу мережі, досліджувати елементи сторінки та пов'язані з нею дані;
- швидкодія роботи. Одна сторінка, що містить весь необхідний інтерфейс не тільки економить час на повторне завантаження даних, а й підвищує продуктивність роботи з вебдодатком.

Хоча наш додаток побудований на основі Next.js і має чітко визначену структуру сторінок, завантаження даних здійснюється виключно на клієнтській стороні. Це робить його архітектуру функціонально ідентичною SPA (Single Page Application).

Проте використання Next.js надає додаткові переваги завдяки підтримці статичної генерації сторінок (SSG). У рамках цього підходу базова структура сторінок створюється під час збирання production версії проєкту. Це дозволяє зменшити час початкового завантаження та покращити SEO-оптимізацію, оскільки базовий HTML вміст генерується заздалегідь.

Таким чином, наш додаток поєднує переваги архітектури SPA з перевагами Next.js, забезпечуючи високу продуктивність, адаптивність і покращений досвід для користувачів. Схематичне зображення архітектури наведено на рис. 2.1.

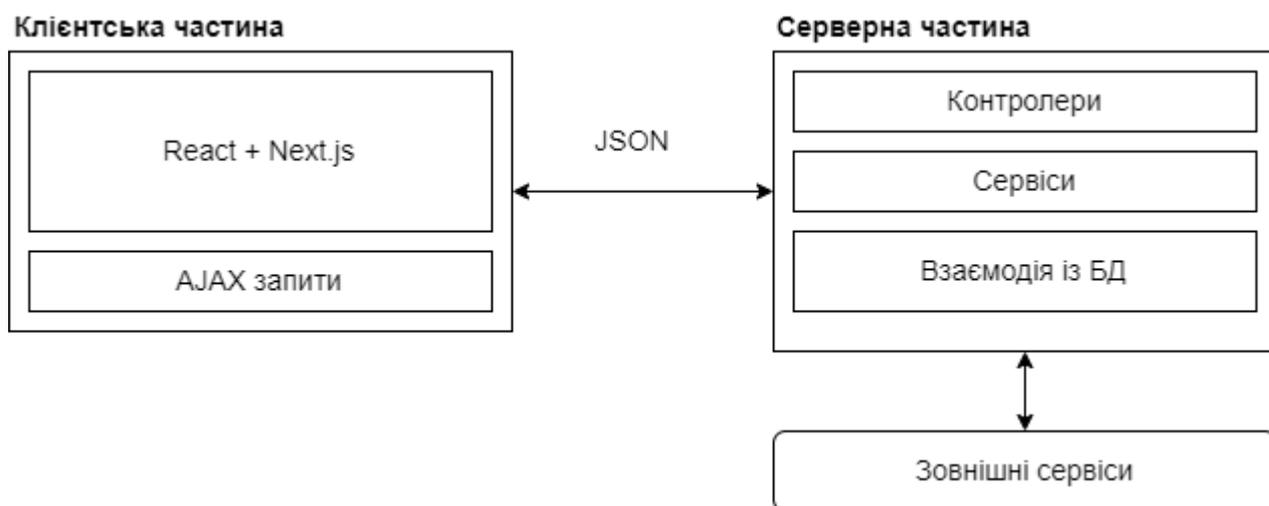


Рисунок 2.1 – Схематичне зображення клієнт-серверної архітектури додатку

Серверна частина додатка розроблена з використанням Node.js та фреймворку Express, які забезпечують масштабованість і ефективність вебсервера. У кореневій теці проєкту розміщуються файли налаштувань, необхідні для коректної роботи застосунку, сторонні модулі, а також дві основні теки: `src` та `dist`.

Оскільки проєкт написано мовою TypeScript, усі файли з розширенням `.ts` необхідно транспілювати у `.js`, щоб забезпечити їх сумісність із вебсервером. Для цього використовується транспілятор, який перетворює TypeScript-код у JavaScript. Транспільовані файли розміщуються у теці `dist`, яка містить готову до виконання серверну логіку.

Тека `src` є основною для розробки серверної частини додатка. Вона містить вихідний код і реалізує основну логіку роботи сервера. Зокрема, у теці `src` розташовані такі компоненти:

- контролери, обробляють запити від клієнта, взаємодіють із бізнес-логікою та повертають відповідь;
- сервіси, реалізують бізнес-логіку та допоміжні функції;
- маршрутизація – визначає кінцеві точки API, які використовуються для взаємодії між клієнтом і сервером;

- middlewares – функції-посередники, що мають доступ до об'єктів запиту та відповіді й виконують свій код перед логікою, що описана в контролері;

- моделі, що описують структуру даних та взаємодію з базою даних, створенні за допомогою пакету mongoose, в яких описуються колекції та їх типи полів у MongoDB.

Така структура забезпечує зрозумілу організацію коду, спрощує підтримку та масштабування серверної частини додатка.

Клієнтська частина створена за допомогою Next.js. Фреймворк має базову структуру файлів, оптимізує кінцеву версію продукту й забезпечує комфорт під час розробки.

У кореневій теці проєкту розташовуються конфігураційні файли, необхідні сторонні модулі, а також три основні теки: public, pages та src.

Тека public використовується для обслуговування статичних файлів, таких як зображення, шрифти чи файли favicon. Вміст цієї теки доступний безпосередньо через вебсервер, без додаткової обробки.

Тека pages слугує для організації маршрутизації в проєкті. Кожен файл у цій теці автоматично стає окремим маршрутом у вебдодатку. Next.js забезпечує маршрутизацію на основі файлової системи, що значно спрощує розробку.

Тека src містить вихідний код клієнтської частини додатка. Основні складові цієї теки:

- компоненти — усі частини користувацького інтерфейсу, які використовуються у застосунку, включаючи повторно вживані елементи (кнопки, форми, заголовки тощо);

- спеціальні функції-хуки — React-хуки, що дозволяють працювати з методами життєвого циклу функціональних компонентів, станом додатка та побічними ефектами;

- логіка роботи з HTTP-запитами — налаштування спеціалізованого екземпляра бібліотеки axios для виконання запитів до сервера. Тут також

реалізовані перехоплювачі запитів і відповідей для обробки помилок чи додавання токенів авторизації;

- моделі даних — визначення типів для об'єктів, що використовуються в проєкті, забезпечуючи точність і безпеку роботи з даними завдяки TypeScript;
- глобальне сховище стану — центральне місце для управління станом додатка, яке дозволяє синхронізувати дані між різними компонентами;
- базові стилі та теми — файли, що містять загальні стилі для додатка, зокрема кольорові схеми, шрифти та налаштування тем.

Така структура спрощує розробку, підтримку та масштабування проєкту, забезпечуючи чітке розділення обов'язків між різними частинами коду.

2.2 Розробка компонентів і взаємодії у додатку

Для розробки клієнтської та серверної частин додатка було обрано JavaScript. Це дозволяє створювати повноцінний вебдодаток без залучення інших мов програмування. JavaScript є інтерпретованою мовою програмування з динамічною типізацією, яка підтримує кілька парадигм, зокрема об'єктно-орієнтований та функціональний стилі [12].

Стандартом мови є специфікація ECMAScript, кожна версія якої має свої особливості. Наразі рекомендується використовувати специфікацію не нижче ECMAScript 6 (ES6), оскільки цей стандарт привніс багато нових можливостей, що значно сприяло розвитку мови та її популяризації серед розробників. Найновішою версією на сьогодні є ECMAScript 12 (ES14). Однак не всі сучасні браузери підтримують новітній синтаксис останніх версій стандарту, тому під час розробки часто використовують Babeljs. Це інструмент, який дозволяє перетворювати код, написаний за стандартом ECMAScript 6 і вище, у зворотно сумісну версію JavaScript, що стабільно працює в поточних і застарілих браузерах чи середовищах [13].

У нашому проєкті, окрім JavaScript, використовуємо TypeScript — строго типізовану мову програмування, побудовану на основі JavaScript. Мова була

розроблена компанією Microsoft, але з самого початку стала OpenSource і активно розвивається спільнотою. TypeScript додає до JavaScript додатковий синтаксис, зокрема механізм статичної типізації, що допомагає зменшити кількість помилок під час написання коду. TypeScript транспілюється в JavaScript, що дозволяє виконувати його в будь-якому браузері чи сервері, оснащеному JavaScript-середовищем.

TypeScript успадковує всі основні переваги JavaScript, але також надає додаткові переваги, зокрема завдяки статичній типізації та інших концепціях, характерних для TS. Ці можливості стають особливо корисними при роботі з великою кодовою базою або в командах розробників, які працюють над одним проєктом. Статична типізація дозволяє зменшити кількість помилок на етапі розробки, покращує автодоповнення та рефакторинг коду, а також полегшує взаємодію між різними частинами програми, що робить розробку більш ефективною та надійною.

JavaScript є мовою з динамічною типізацією, що означає, що перевірка типів і виявлення помилок типізації відбувається лише під час виконання коду. Цей підхід надає більшу гнучкість, дозволяючи компонентам програми адаптуватися і змінюватися на льоту. Однак у великих проєктах, де змінних і типів стає все більше, це може призвести до труднощів у відстеженні потенційних помилок, оскільки неправильне використання або відсутність чіткої типізації може спричинити помилки, які виявляються лише в процесі виконання [14].

TypeScript пропонує необов'язкову статичну типізацію, що дозволяє змінним зберігати фіксований тип після їх оголошення, і вони можуть приймати лише певні значення відповідно до визначеного типу. Компілятор TypeScript попереджає про помилки типізації на етапі компіляції, що дозволяє уникнути їх потрапляння у кінцеву версію проєкту. Важливо зазначити, що TypeScript не вимагає обов'язкового оголошення типів у всіх частинах проєкту — розробники можуть визначати типи лише в необхідних місцях. Цей підхід

відрізняє TypeScript від інших статично типізованих мов програмування і дозволяє досягти правильного балансу між гнучкістю та типобезпекою.

Інформація про типи значно покращує роботу редакторів коду та інтегрованих середовищ розробки (IDE), роблячи їх набагато кориснішими. Завдяки цьому, IDE можуть пропонувати такі функції, як навігація по коду та автодоповнення, що дають точні варіанти вибору. Це значно полегшує написання та підтримку коду, підвищуючи загальну продуктивність розробки. Важливо зазначити, що TypeScript продовжує активно розвиватися, і під час розробки нашого проекту актуальною була версія 5.7.

Після того як вибору мови програмування, розглянемо основні технології, що були використані для розробки клієнтської частини вебдодатку.

Основним фундаментальним елементом при розробці користувацького інтерфейсу був вибір бібліотеки React, основною метою якої є спрощення роботи з великою кількістю даних, що постійно оновлюють інтерфейс, а також їх оптимізація. React дозволяє ефективно керувати станом компонентів та оновлювати лише ті частини інтерфейсу, які змінюються, що покращує продуктивність. Розробкою React в основному займається невелика команда, що працює у Facebook, хоча також свій внесок роблять розробники з усього світу [15].

До основних ключових особливостей React можна віднести наступні:

- JSX синтаксис: Це надбудова над JavaScript, що дозволяє використовувати синтаксис HTML-тегів для зручного створення компонентів. JSX дозволяє писати компоненти у вигляді, що нагадує звичайний HTML, що спрощує процес розробки;

- багаторазове використання компонентів: Вебдодаток складається з кількох компонентів, кожен з яких має свою логіку та елементи управління. Ці компоненти відповідають за рендеринг невеликої частини HTML, яку можна повторно використовувати у різних частинах додатка;

- підвищення швидкодії інтерфейсу: React використовує механізм віртуального DOM, що дозволяє оптимізувати процес оновлення інтерфейсу.

Віртуальний DOM працює швидше, оскільки змінює лише ті елементи DOM, які зазнали змін, замість того, щоб повторно рендерити все дерево DOM;

- React API верхнього рівня: Це набір функцій, що дозволяє працювати з компонентами, контролюючи їх стан без необхідності безпосереднього звернення до DOM-елементів.

Щоб розгорнути середовище для розробки на React, можна скористатися офіційним інструментом від команди розробки — `prx create-react-app`, що забезпечує налаштовану середу з базовими залежностями та конфігурацією. Альтернативою є налаштування власної конфігурації за допомогою Webpack, або використання інших інструментів, таких як Parcel, Snowpack та інші схожі рішення для побудови та розгортання вебдодатків.

Звичайно існують інші інструменти для ефективної розробки клієнтської частини вебзастосунків, наприклад такі як Vue, Angular, Svelte. Порівняльна характеристика даних засобів наведена у таблиці 2.1.

Таблиця 2.1 – Порівняльна характеристика технологій front-end

Критерії	Назви технологій			
	React	Vue	Svelte	Angular
Швидкість роботи	4	4,5	5	3,5
Легкість у вивченні	4,5	4,5	5	3,5
Популярність	5	4	3	4
Наявність розвинутої спільноти та ресурсів	5	4	3	4

Для зручності розробки було вирішено використовувати Next.js — фреймворк, побудований на основі React, який вирішує багато проблем, що виникають під час налаштування проєкту з нуля та подальшої роботи з ним. Основними перевагами Next.js перед іншими способами розгортання проєкту є:

- зручна система маршрутизації;
- професійні налаштування проєкту;
- можливість налаштування візуалізації певних сторінок;

- автоматичне розділення коду для пришвидшеного завантаження;
- підтримка різноманітних способів стилізації проєкту;
- можливість легкого розгортання проєкту в інтернеті за допомогою використання Vercel (платформа для frontend фреймворків й статичних сайтів);
- можливість створення API для легкої взаємодії із сервером.

Особливу увагу у огляді Next.js необхідно приділити опціям візуалізації, адже саме це є основним аргументом у використанні даного фреймворку. Короткий опис в цілому наявних видів візуалізації зображено на рис. 2.2.

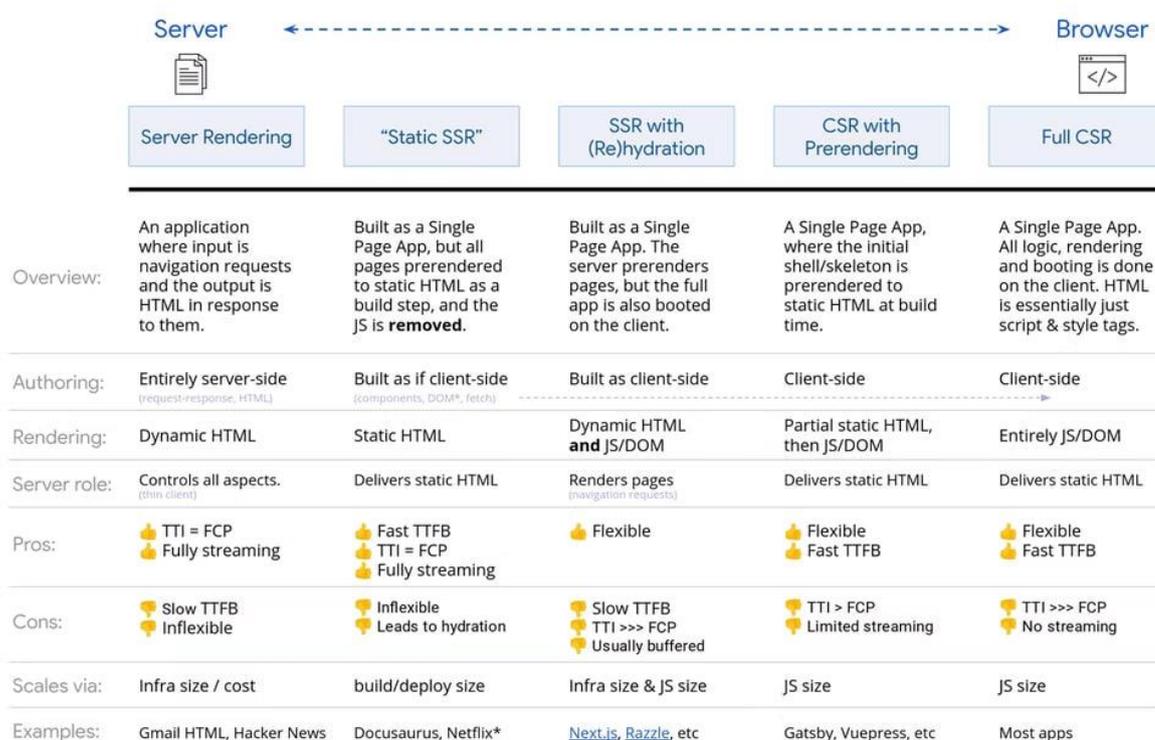


Рисунок 2.2 – Основні види візуалізації у вебзастосунках [16]

SSR (Server-side rendering) – здатність вебдодатку створювати HTML-файли на сервері та відправляти повністю відтворені HTML-сторінки до клієнта. Браузер подає запит на інформацію від сервера, який миттєво відповідає, надсилаючи клієнту повністю відтворену сторінку. Пошукові системи можуть сканувати та індексувати вміст цих файлів, що корисно для пошукової оптимізації [17].

SSG (Static-site generation) – здатність генерувати повний статичний HTML вебсайт на основі необроблених даних і набору шаблонів. По суті, статична генерація сайтів готує HTML-сторінки для показу користувачам на стадії створення будови проєкту. Оскільки ці сторінки попередньо створені, вони дуже швидко завантажуються у браузерях користувачів та не потребують налаштування власного вебсервера. Такий підхід ще іноді називають pre-rendering [18].

CSR (Client-side rendering) – клієнтська візуалізація має на увазі зображення сторінок прямо у браузері за допомогою JavaScript. Уся логіка отримання даних, шаблонізація та маршрутизація обробляються на стороні клієнта, а не на сервері [19].

ISR (Incremental static regeneration) – підхід, що дозволяє оновлювати сторінки, що були побудовані за допомогою SSG. Коли робиться запит на статично генеровану сторінку, то зображується кешована сторінка. Розробник може поставити час, через який Next.js повинен буде регенерувати сторінку, у разі оновлення початкових даних, з якими вона була генерована [20].

Next.js дозволяє комбінувати вище описані техніки візуалізації та генерації сторінок, що надає важливу гнучкість під час розробки.

Для стилізації проєкту використовується CSS-фреймворк Tailwind. Ця бібліотека розроблена для швидкого і зручного створення стилів, застосовуючи унікальний підхід до утилітарних класів без необхідності написання окремих CSS-файлів. Tailwind гармонійно інтегрується в процес розробки компонентів, забезпечуючи високу продуктивність і гнучкість [21].

Особливу увагу заслуговує підхід mobile-first, який лежить в основі Tailwind. Фреймворк сприяє створенню адаптивного дизайну, починаючи з базового вигляду для мобільних пристроїв і поступово додаючи стилі для більших екранів. Це забезпечує логічну та ефективну організацію стилів, оптимізованих для сучасних пристроїв.

Основні переваги використання Tailwind у порівнянні зі звичайним CSS:

- mobile-first адаптивність. Tailwind надає простий і зрозумілий механізм роботи з медіа-запитами через модифікатори, що дозволяє легко адаптувати дизайн під різні розміри екранів;
- мінімізація дублювання стилів. Використання утилітарних класів зменшує обсяг CSS-коду та виключає дублювання стилів;
- швидкість розробки. Завдяки готовим класам, розробники можуть реалізувати складні макети значно швидше, концентруючись на функціональності;
- контроль над стилями. Tailwind компілює лише ті стилі, які використовуються у проєкті, що позитивно впливає на продуктивність та швидкість завантаження сторінок;
- кросбраузерна сумісність. Фреймворк автоматично додає необхідні префікси для підтримки сучасних CSS-властивостей у різних браузерах.

Підхід mobile-first у поєднанні з потужністю Tailwind робить цей фреймворк ідеальним вибором для створення сучасного, адаптивного та швидкого у розробці дизайну.

Основою для реалізації серверної частини проєкту є програмна платформа Node.js - середовище виконання JavaScript коду, що побудоване на основі рушія JavaScript Chrome V8, який дозволяє транслювати виклики мовою JavaScript у машинний код [22].

Node.js працює в одному потоці, що дозволяє уникнути необхідності створювати окремі потоки для кожного запиту. Відповідно, це дозволяє зменшити витрати на ресурси та спрощує управління сервером. Однак, на відміну від традиційних серверів, Node.js використовує асинхронні операції вводу-виводу, завдяки чому він здатний ефективно обробляти велику кількість запитів одночасно, не блокуючи виконання коду.

У стандартній бібліотеці Node.js передбачено багато асинхронних примітивів, таких як зворотні виклики (callbacks) і проміси (promises), що дозволяють реалізувати неблокуючий ввід-вивід. Це означає, що коли Node.js виконує операції вводу-виводу (наприклад, при зчитуванні даних з бази даних

або файлової системи), він не блокує потік виконання, а негайно повертається до інших операцій, поки не отримає відповідь на запит.

Цей підхід дозволяє Node.js обробляти тисячі одночасних підключень без створення паралельних потоків, що знижує ймовірність помилок, пов'язаних з багатозадачністю, та значно підвищує ефективність роботи сервера.

Одною зі значних переваг слід зазначити можливість написання коду мовою JavaScript чи TypeScript, що значно поширює коло розробників, які мають можливість писати серверну частину без необхідності у вивченні іншої мови програмування.

У тандемі із Node.js проєкт використовує фреймворк Express.js, що являє собою найпопулярніше рішення при роботі із Node.js [23]. Його використання надає наступні можливості:

- написання обробників подій для запитів з різними HTTP-методами, за різними URL-адресами (маршрутами);
- встановлення загальних параметрів для вебдодатків;
- додавання додаткових middlewares в будь-якій точці в рамках потоку обробки запитів.

Express був випущений у листопаді 2010 року та наразі має актуальну версію 4.21.2. Хоча сам Express досить мінімалістичний, спільнота створила сумісні пакети програмного забезпечення для вирішення практично будь-якої проблеми веброзробки. Існують бібліотеки для роботи з файлами cookie, сесансами, входами користувачів, параметрами URL-адресів, даними, заголовками безпеки та багато іншого.

У якості бази даних для проєкту використовується MongoDB — документоорієнтована NoSQL система керування базами даних з відкритим вихідним кодом. Вона широко застосовується для зберігання великих обсягів даних завдяки своїй гнучкості та масштабованості. На відміну від реляційних баз даних, де дані організовані у вигляді таблиць з рядками та стовпцями, MongoDB використовує концепцію колекцій та документів. Документи є основною одиницею зберігання і представляють собою набір пар "ключ-

значення", що дозволяє працювати зі складними структурами даних. Колекції виступають аналогами таблиць і містять множину документів, при цьому схема документів може бути створена на льоту без попереднього визначення [24].

Ця гнучкість дозволяє MongoDB зберігати різноманітні дані та забезпечує високу продуктивність і простоту масштабування. Структура документів відповідає звичайним JavaScript об'єктам, що полегшує інтеграцію з сучасними вебдодатками. Завдяки можливості зберігати масиви та складні структури, MongoDB ефективно підтримує ієрархічні зв'язки. Середовище цієї бази даних легко розширюється та може містити велику кількість колекцій і документів. Крім того, MongoDB пропонує сервіс «MongoDB Atlas» — хмарне рішення, яке дозволяє безкоштовно проводити дослідження та тестування.

У проєкті MongoDB інтегровано з ODM (Object Document Mapper) Mongoose. Це забезпечує зручне створення моделей і залежностей між ними, використовуючи спеціальне API для взаємодії з базою даних. Такий підхід значно спрощує управління даними у проєктах, що потребують швидкої обробки великих обсягів змінюваних даних. Основні відмінності між NoSQL і SQL СКБД наведені в додатку А.

2.3 Методи розгортання додатку

У процесі розробки додатку однією з ключових стадій є розгортання, тобто перехід від розробки до повноцінної роботи продукту на сервері. Вибір методу розгортання безпосередньо впливає на стабільність, швидкість роботи та зручність підтримки додатку. Для проєкту, що використовує Next.js, одним з найбільш ефективних рішень є використання платформ, спеціально налаштованих для розгортання додатків, створених за допомогою цього фреймворку.

Одним з найпопулярніших варіантів для розгортання вебдодатків на базі Next.js є платформа Vercel [25]. Це хмарне середовище, розроблене спеціально для оптимізації розгортання вебдодатків, зокрема побудованих за допомогою

Next.js. Vercel підтримує всі основні можливості Next.js, такі як SSR (Server-Side Rendering), SSG (Static-Site Generation), ISR (Incremental Static Regeneration) та інші, що робить її ідеальним варіантом для розгортання сучасних вебдодатків.

Основні переваги використання Vercel:

- простота налаштування: Однією з головних переваг Vercel є його інтеграція з GitHub, GitLab або Bitbucket. Кожен раз, коли змінюється код у репозиторії, Vercel автоматично запускає процес побудови додатку та розгортає його;

- швидкість: Завдяки використанню CDN (Content Delivery Network), додаток доставляється користувачам з максимальною швидкістю, що позитивно впливає на час завантаження сторінок та загальний досвід користувачів;

- підтримка статичних та динамічних сторінок: Vercel ефективно працює як зі статичними сторінками (генерація під час побудови), так і з динамічними (з використанням SSR або ISR). Це дозволяє комбінувати різні стратегії рендерингу, що дає гнучкість в управлінні контентом;

- автоматичне масштабування: Платформа автоматично масштабує сервери для обробки великої кількості одночасних запитів, що забезпечує безперебійну роботу навіть при великому навантаженні.

Vercel є зручним вибором для проєктів, які активно використовують Next.js, оскільки цей фреймворк повністю підтримує всі функціональні можливості платформи.

Попри зручність Vercel, існує кілька інших варіантів для розгортання Next.js додатків, кожен з яких має свої особливості та може бути вибраний в залежності від вимог проєкту.

Netlify є ще однією популярною платформою для хостингу статичних вебсайтів і динамічних вебдодатків. Як і Vercel, Netlify дозволяє легко налаштувати безперервну інтеграцію (CI) та автоматичне розгортання через Git репозиторії. Платформа підтримує такі функціональності, як автоматичне масштабування, глобальне розповсюдження через CDN, а також надає

можливість налаштування серверних функцій за допомогою Netlify Functions (що є зручним для обробки запитів та виконання серверної логіки) [26].

Основні переваги:

- інтеграція з Git: Автоматичне розгортання додатку після кожного коміту в Git репозиторії;
- підтримка серверних функцій: Легке додавання серверних функцій для обробки динамічних запитів без необхідності налаштування окремого сервера;
- підтримка різних стратегій рендерингу: Підтримка SSG, SSR та інших методів генерації сторінок, аналогічно Vercel.

Для великих проєктів, де потрібен високий рівень контролю та масштабованості, AWS пропонує потужне середовище для розгортання вебдодатків. Для розгортання Next.js додатку можна використати AWS Amplify, AWS Lambda, або EC2 для більш складних налаштувань. За допомогою AWS Amplify можна автоматизувати процеси побудови, тестування та розгортання додатків.

До переваг можна віднести:

- масштабованість: Підтримка масштабування як для невеликих, так і для великих додатків;
- потужні інструменти для розробки та хостингу: AWS пропонує різноманітні інструменти для налаштування серверів, баз даних та управління додатками;
- інтеграція з іншими сервісами AWS: Можливість використання таких сервісів, як S3, CloudFront, RDS, Lambda для створення комплексної інфраструктури.

Heroku є хмарною платформою, що дозволяє розгортати додатки з підтримкою різних мов програмування, зокрема Node.js. Heroku простий у використанні і добре підходить для невеликих та середніх проєктів. Для Next.js можна використовувати просту інтеграцію з Git репозиторієм, що дозволяє автоматично розгортати додатки після кожного коміту [27].

Основні переваги:

- легкість налаштування: Простий процес розгортання через Git;
- масштабованість: Можливість масштабування через додавання нових динамічних або статичних ресурсів;
- інтеграція з іншими додатками та базами даних: Heroku підтримує інтеграцію з численними сторонніми сервісами та базами даних.

При виборі платформи для розгортання важливо враховувати такі фактори:

- простота налаштування: Для невеликих проєктів, де важлива швидкість розгортання, варто обрати рішення, яке має мінімальні вимоги щодо конфігурації (наприклад, Vercel чи Netlify);
- масштабованість: Для великих проєктів із високим трафіком варто вибирати платформи, які можуть автоматично масштабувати сервери, такі як AWS або Heroku;
- контроль над інфраструктурою: Якщо потрібен повний контроль над сервером та налаштуванням, можна обрати DigitalOcean або AWS.

Вибір методу розгортання додатку залежить від потреб проєкту. Для більшості випадків, пов'язаних з розгортанням Next.js додатків, оптимальними рішеннями є Vercel та Netlify, оскільки вони пропонують зручні інструменти для автоматизації розгортання і масштабування, а також мають високу продуктивність завдяки використанню CDN. Для більш складних проєктів з високими вимогами до контролю та налаштування інфраструктури підійдуть AWS або DigitalOcean.

РОЗДІЛ 3

РЕАЛІЗАЦІЯ ПРОГРАМНОГО ДОДАТКУ ТА ОЦІНКА ЕФЕКТИВНОСТІ ЗАПРОПОНОВАНОГО РІШЕННЯ.

3.1 Концептуальна модель додатку

Одним із очікуваних результатів нашого дослідження було створення власного мобільного додатку для збирання та аналізу метеорологічної інформації. Розроблений додаток є інструментом для здійснення метеорологічних спостережень на основі сформованих користувачем звітів за введеними даними локації та проміжку часу. Користувач може налаштовувати вигляд звіту відповідно до своїх конкретних потреб.

Зважаючи на те, що багато людей у наш час використовують мобільні пристрої значно частіше, ніж комп'ютери, додаток був розроблений з акцентом на mobile-first підхід. Це дозволяє забезпечити зручний доступ та коректну роботу додатку на смартфонах та планшетах, що особливо важливо для користувачів, які знаходяться в полях чи інших мобільних умовах.

Однією з областей застосування є агрономія, де отримані дані можуть використовуватись для:

- прогнозування поширення шкідливих організмів, зокрема збудників хвороб;
- визначення інтегрального агрокліматичного показника, такого як гідротермічний коефіцієнт, для якого потрібно знати середні температури й кількість опадів за однаковий проміжок часу;
- виявлення фаз розвитку сільськогосподарських культур, що важливо для визначення технічних робіт, необхідних для вирощування культур, та планування системи захисту цих культур;
- прогнозування часу збирання врожаю, де важливим показником є сума ефективних температур.

Під час розробки користувацького інтерфейсу основними принципами були: легкість, мінімалізм, сучасний дизайн, можливість подальшої адаптації під мобільні пристрої та інтуїтивно зрозуміле використання. Приклад однієї зі сторінок показано нижче на рис. 3.1.

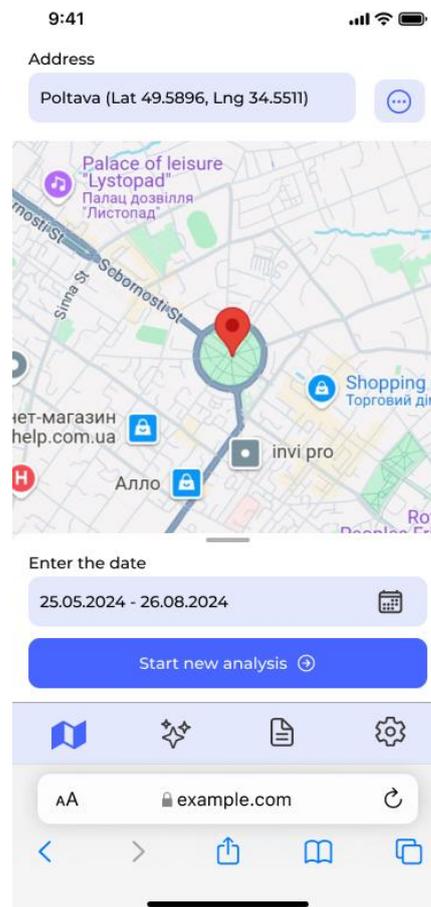


Рисунок 3.1 – Приклад сторінки із додатку

На етапі проєктування системи було розроблено UML діаграми прецедентів та послідовностей, що представлені в додатках Б та В відповідно. Ці діаграми допомогли чітко визначити основні функції додатку та показати, як взаємодіють різні модулі системи.

Діаграма прецедентів зображує відносини між акторами та варіантами використання, що визначають функції, які система надає користувачам. Вона допомагає описати, які дії система виконує під час взаємодії з актором, без вказівок на конкретну реалізацію чи логіку роботи модулів. UML використовує

різні типи зв'язків, такі як включення, асоціації, узагальнення та розширення, для точного опису цих взаємодій [28].

Діаграма послідовностей зосереджена на часових аспектах взаємодії між об'єктами системи, що робить її важливим доповненням до діаграми прецедентів. Вона детально описує, як об'єкти обмінюються повідомленнями протягом процесу виконання сценарію. Об'єкти відображаються у вигляді прямокутників з підкресленими іменами, повідомлення — лініями зі стрілками, а відповіді — пунктирними лініями [29].

В результаті цього процесу було сформовано ясне уявлення про функціональні можливості, структуру та дизайн майбутньої системи.

3.2 Опис функціональності додатку

Розроблений додаток орієнтований на використання на мобільних пристроях та забезпечує зручний доступ до метеорологічних даних, їх аналізу та формування аналітичних звітів. Його функціональність охоплює всі необхідні аспекти роботи з сучасними метеорологічними системами, включаючи збирання даних, інтерактивний аналіз та збереження результатів у зручних форматах. Це дозволяє користувачам легко отримувати актуальну інформацію про погодні умови та використовувати її для особистих чи професійних потреб.

Після відкриття додатку користувач потрапляє на сторінку авторизації або реєстрації. Реєстрація доступна через стандартну форму, яка включає основні поля, такі як електронна пошта та пароль. Для безпеки, всі облікові записи вимагають підтвердження через лист, який надсилається на вказану електронну адресу. Це гарантує, що кожен користувач проходить базову перевірку даних. Після цього користувачу необхідно ввести код підтвердження, який надійшов на електронну пошту. На цьому процес реєстрації буде успішно завершено. Зображення процесу реєстрації зображено на рис. 3.2.

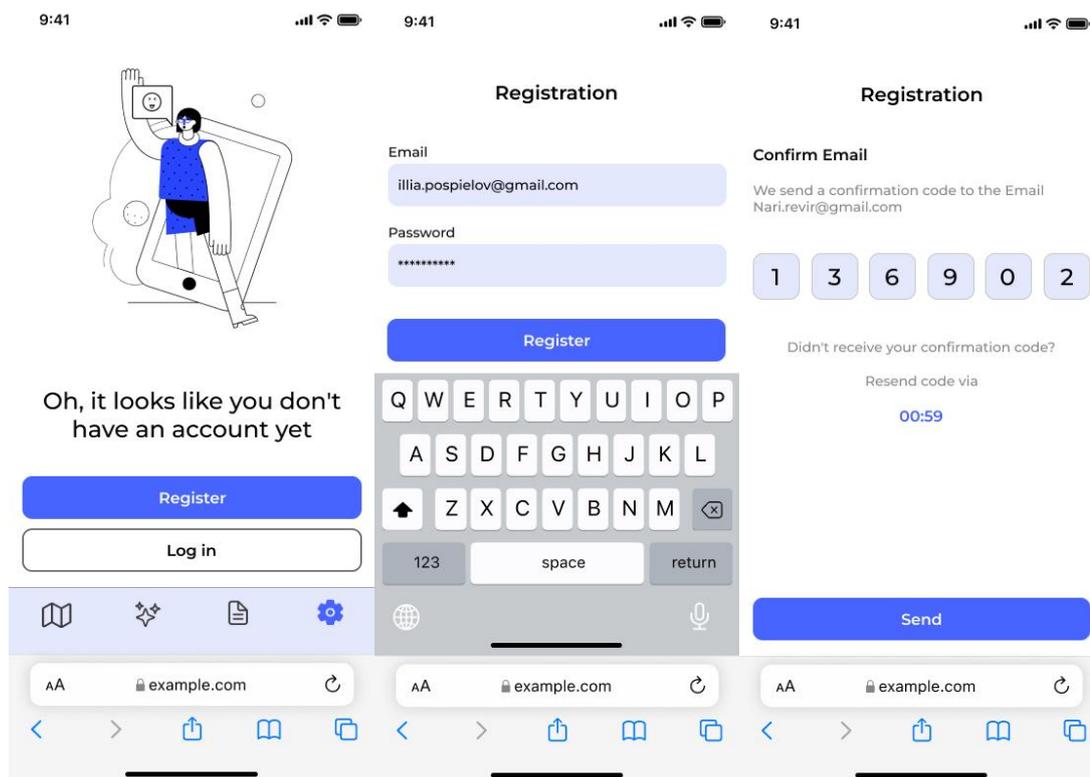


Рисунок 3.2 – Процес реєстрації

Авторизовані користувачі одразу потрапляють на головну сторінку аналізу даних та мають можливість формування звітів на основі заданих параметрів. Інтерфейс включає інтерактивну карту для вибору локації та форму для обрання періоду збору даних. Після заповнення форми, дані обробляються, і користувач отримує доступ до результатів у вигляді звіту із можливістю перегляду графіків, аналітики та усереднення даних. Зображення головної сторінки показано на рис. 3.3.

Інтерфейс мобільного додатку спроектований таким чином, щоб забезпечити зручність роботи навіть для нефахівців. Його структура є логічною та інтуїтивно зрозумілою, що дозволяє легко виконувати складні аналітичні операції без додаткового навчання. Крім того, можливість експорту даних у популярні формати, такі як XLSX та PDF, значно спрощує їх подальше використання у зовнішніх системах або звітності.

Таким чином, функціональність додатку відповідає сучасним вимогам до інформаційних систем, забезпечуючи користувачам потужний інструмент для роботи з метеорологічними даними.

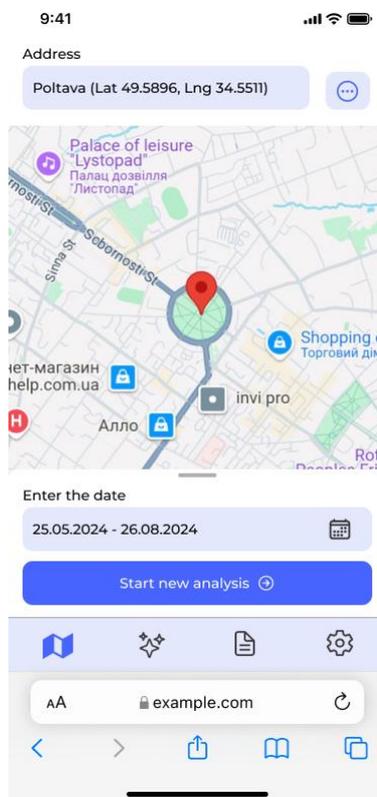


Рисунок 3.3 – Сторінка «Головна сторінка аналізу даних»

Після заповнення полів і натискання кнопки аналізу, користувач отримує доступ до сформованого звіту, який представлений у двох основних блоках:

- загальні показники — містить таблицю з ключовими метеорологічними даними та динамічними графіками, які користувач може налаштовувати;
- аналітика — включає розраховані параметри, такі як сума активних і ефективних температур, гідротермічний коефіцієнт та інші аналітичні індикатори, які мають практичну цінність, особливо для аграрного сектору.

Зображення частин звіту наведено на рис. 3.4 та 3.5.

Крім того, користувач має можливість переглянути раніше сформовані звіти перейшовши до пункту меню «Звіти» де перед ним з’явиться відповідна таблиця. Дана сторінка зображена на рис. 3.6.

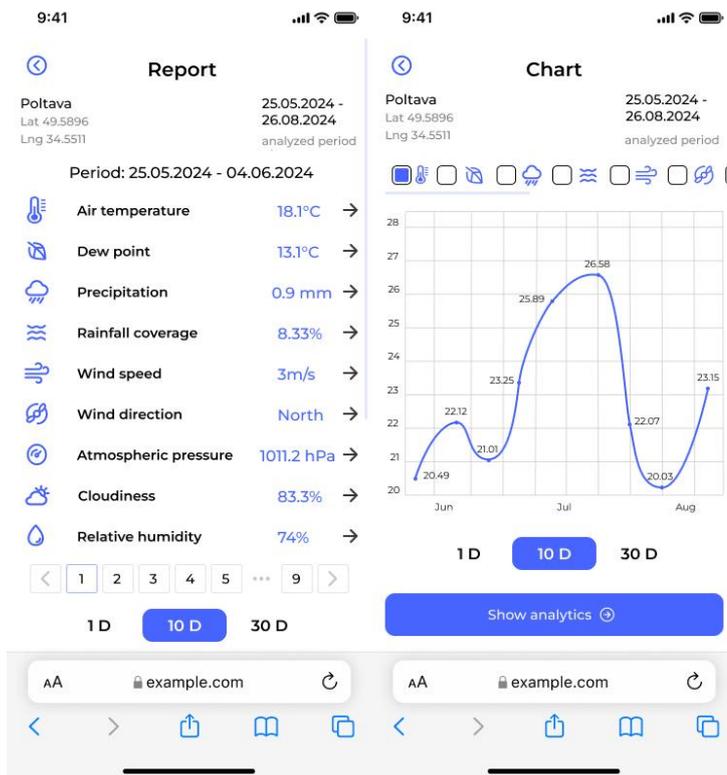


Рисунок 3.4 – Загальні показники

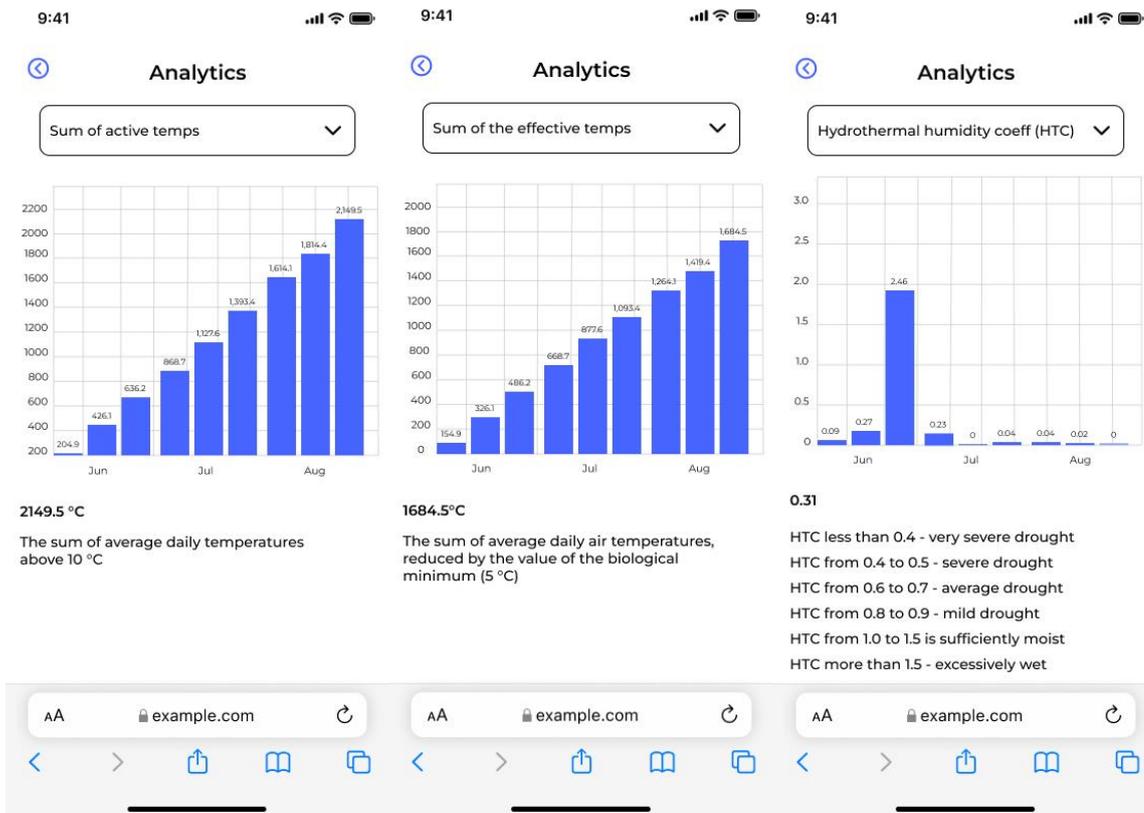


Рисунок 3.5 – Аналітика

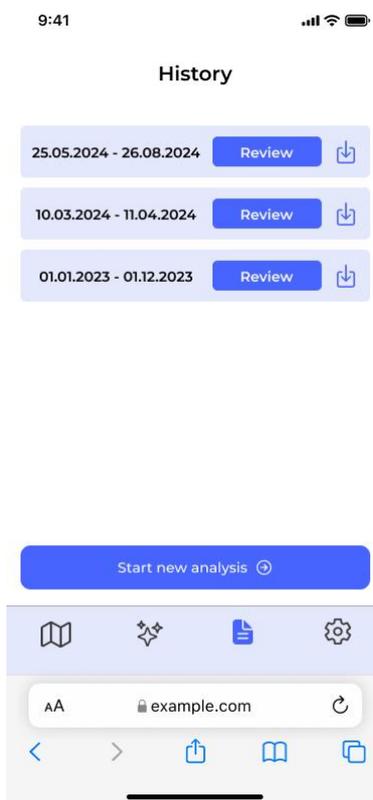


Рисунок 3.6 – Сторінка сформованих звітів

Створений додаток має потенціал для подальшого вдосконалення, що може значно покращити його функціональність і зручність використання. У перспективі планується реалізація додаткових можливостей, зокрема впровадження нових методів аналізу метеорологічної інформації, розширення функціоналу для взаємодії з користувачами через персоналізовані повідомлення та інші вдосконалення, спрямовані на покращення загального досвіду роботи із застосунком.

3.3 Оцінка ефективності запропонованого рішення

Розроблений мобільний додаток для збору та аналізу метеорологічної інформації повністю відповідає поставленим цілям і завданням, що підтверджено тестуванням і апробацією. Додаток успішно виконує збирання метеорологічних даних із заданої локації та за визначений проміжок часу, надає можливість інтерактивного відображення даних на карті, а також формує звіти у вигляді графіків, таблиць і аналітичних висновків. Особливо корисною є

можливість користувача налаштовувати вигляд звіту відповідно до своїх потреб, що робить додаток актуальним для аграрних підприємств, науково-дослідних установ та освітніх організацій.

Інтерфейс додатку створено з урахуванням принципів сучасного UI/UX дизайну, що забезпечує простоту та інтуїтивність у використанні. Чітка структура інтерфейсу, логічно розділені функціональні блоки та можливість швидкого доступу до основних функцій сприяють ефективній роботі користувачів. Додаток має адаптивний дизайн, що дозволяє однаково зручно працювати як на комп'ютерах, так і на мобільних пристроях. Мінімалістичний, але сучасний стиль оформлення сприяє легкому сприйняттю інформації, а тестування показало, що навіть користувачі без спеціальної підготовки швидко розбираються у його функціоналі.

Технічна сторона реалізації забезпечує високу продуктивність і стабільність роботи. Завдяки використанню сучасних технологій, таких як React для клієнтської частини, Node.js для серверної сторони та MongoDB для зберігання великих обсягів даних, програмний продукт ефективно виконує обробку та аналіз даних. Архітектура додатку дозволяє обробляти значні обсяги інформації та масштабувати систему у разі розширення функціоналу або збільшення кількості користувачів.

Оптимізація запитів до бази даних і застосування асинхронного програмування значно знижують час відповіді додатку навіть за умов високого навантаження. Крім того, було враховано сучасні вимоги до інформаційної безпеки: забезпечено захист з'єднань через протокол HTTPS, що гарантує конфіденційність і цілісність переданих даних.

Загалом мобільний додаток демонструє високу ефективність, поєднуючи зручність використання, стабільність роботи та широкий спектр функціональних можливостей, що робить його перспективним інструментом для використання у різних галузях.

ВИСНОВКИ

В результаті виконаної роботи була досягнута поставлена мета роботи та вирішено ряд завдань. Було проведено аналіз наукової, методичної та нормативної літератури з теми розробки мобільних додатків і організації метеорологічних досліджень, а також обґрунтовано вдосконалення якості та зручності аналізу зміни метеорологічних показників у заданих локаціях протягом обраного періоду з урахуванням вимог користувачів.

Аналіз інтернет-ресурсів показав наявність великої кількості сайтів, які надають прогностичні дані, базуючись на метеорологічних вузлах, і дають змогу отримати актуальну інформацію про погоду та її прогноз на декілька днів уперед. Додатки, що дозволяють збирати та аналізувати метеорологічні показники в історичному контексті, зазвичай надаються на платній основі.

Користувацький інтерфейс мобільного додатка розроблений з використанням бібліотеки React, а серверна частина реалізована на платформі Node.js.

Особливу увагу було приділено створенню мобільної версії додатка, яка забезпечує доступ до функціоналу системи з використанням смартфонів та планшетів. Мобільна версія дозволяє користувачам отримувати метеорологічні дані, проводити їх аналіз та візуалізацію незалежно від місця знаходження, що значно підвищує зручність та доступність системи.

Таким чином, результатом роботи є розроблений додаток, який збирає метеорологічні дані за допомогою стороннього API, обробляє їх та надає у зручному для користувача форматі. Графічне представлення даних у вигляді графіків дозволяє ефективно аналізувати зміни метеорологічних показників за обраний період.

Створена система може бути використана в наукових, навчальних установах та підприємствах різних форм власності для узагальнення отриманої інформації, її аналізу та прогнозування природних і виробничих процесів у

різних

галузях

народного

господарства.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Тлумачний словник з метеорології та кліматології: навч. посіб. для студ. неметеорол. спец. / В. І. Затула, Л. М. Титаренко ; Черкаський національний ун-т ім. Богдана Хмельницького. Черкаси : [Вид. від. ЧНУ ім. Б. Хмельницького], 2009. 75 с.
2. Метеорологія і кліматологія як науки. URL: <http://www.geograf.com.ua/meteorology/873-meteorologiya-i-klimatologiya-yak-nauki-istoriya-jikh-rozvitku> (дата звернення 20.10.2024).
3. Основи агрометеорології: Підручник / Польовий А.М., Божко Л.Ю., Вольвач О.В.; Одеський державний екологічний університет, Одеса : Видавництво ТЕС, 2012. 250 с.
4. Авіаційна метеорологія. URL: <https://cutt.ly/utCVNgc> (дата звернення 20.11.2024).
5. Супутникова метеорологія. URL: <https://cutt.ly/itC2hCc> (дата звернення 20.10.2024).
6. Ситник В. Ф. Інтелектуальний аналіз даних (дейтамайнінг): навч. посібник. Київ: КНЕУ, 2007. - 376 с.
7. Нажмудінова О.М. Синоптична метеорологія: Конспект лекцій. – Одеса, 2010. – 77 с.
8. Комендат О. Роль сучасних інформаційних технологій в аналізі метеорологічних даних. Матеріали X Всеукраїнської студентської науково - технічної конференції / В 2 т. Тернопіль: Тернопільський національний технічний університет ім. І. Пулюя (м. Тернопіль, 25-26 квітня 2017р.), 2017. С. 53-54.
9. Meteorprog. URL: <https://www.meteorprog.ua/ua/> (дата звернення 20.10.2024).
10. WeeWX. URL: <https://weewx.com/> (дата звернення 28.10.2024).

11. Meteo Farm. URL: <https://www.meteo.farm/> (дата звернення 20.10.2024).
12. Single-page application. URL: <https://livebook.manning.com/book/spa-design-and-architecture/chapter-1/> (дата звернення 22.11.2024).
13. What is Babel. URL: <https://babeljs.io/docs/en/> (дата звернення 22.11.2024).
14. The Good and the Bad of TypeScript. URL: <https://www.altexsoft.com/blog/typescript-pros-and-cons/> (дата звернення 22.11.2024).
15. React a JavaScript library for building user interfaces. URL: <https://reactjs.org/> (дата звернення 22.11.2024).
16. Rendering on the Web. URL: <https://web.dev/rendering-on-the-web/> (дата звернення 22.11.2024).
17. Server-side Rendering Definition. URL: <https://www.heavy.ai/technical-glossary/server-side-rendering> (дата звернення 24.11.2024).
18. What is a static site generator. URL: <https://www.cloudflare.com/learning/performance/static-site-generator/#:~:text=A%20static%20site%20generator%20is,to%20users%20ahead%20of%20time> (дата звернення 24.11.2024).
19. Client-side rendering. URL: <https://rockcontent.com/blog/client-side-rendering-vs-server-side-rendering/> (дата звернення 24.11.2024).
20. Incremental Static Regeneration. URL: <https://nextjs.org/docs/basic-features/data-fetching/incremental-static-regeneration> (дата звернення 24.11.2024).
21. Tailwind. URL: <https://tailwindcss.com/docs/installation> (дата звернення 24.11.2024).
22. Introduction to Node.js. URL: <https://nodejs.dev/learn> (дата звернення 24.11.2024).
23. Express. URL: <https://expressjs.com/> (дата звернення 24.11.2024).
24. What is MongoDB? Introduction, Architecture, Features & Example. URL: <https://www.guru99.com/what-is-mongodb.html> (дата звернення 24.11.2024).

25. Vercel. URL: <https://vercel.com/docs/getting-started-with-vercel> (дата звернення 28.11.2024).
26. Netlify. URL: <https://docs.netlify.com/> (дата звернення 28.11.2024).
27. Heroku. URL: <https://devcenter.heroku.com/> (дата звернення 28.11.2024).
28. USE CASE DIAGRAMS. URL: <https://www.quality-assurance-group.com/use-case-diagrams/> (дата звернення 30.11.2024).
29. Sequence diagram. URL: https://www.wikiwand.com/en/Sequence_diagram (дата звернення 30.11.2024).

ДОДАТОК А

ПОРІВНЯЛЬНА ХАРАКТЕРИСТИКА БАЗ ДАНИХ

Таблиця А.1 - Порівняльна характеристика NoSQL та SQL баз даних

Критерії	Представники NoSQL та SQL баз даних	
	MongoDB	MySQL
Швидкість роботи	Має перевагу при діях пов'язаних із вставленням та оновленням великої кількості записів.	Дозволяє швидко обирати велику кількість записів з таблиць.
Зручність у використанні	Зберігає дані в колекціях без примусового створення чіткої схеми даних.	Потребує розуміння принципів нормалізації, цілісності та дизайну реляційної бази даних.
Можливість до розширення	Налаштування розподіленого кластеру дозволяє налаштування його окремих частин як набору реплік. У розподіленому кластері дані розподіляються між багатьма серверами, що надає майже безмежний простір до розширення.	Можливості до розширення значно менші, є 2 варіанти: вертикальне розширення або додавання реплік читання. Вертикальне розширення передбачає додавання додаткових ресурсів до існуючого сервера бази даних.
Гнучкість	Дизайн документів із необов'язковою чіткою схемою дозволяє створювати та покращувати програми без необхідності запускати складні та дорогі процеси міграції схем.	Повільний перенос схем та збережених процедур, що можуть залежати від оновлення схем даних створюють додаткові труднощі.
Безпека	Використовує модель доступу на основі ролей з гнучким набором дозволів. Комунікація шифрується за допомогою TLS.	Підтримує ті самі функції шифрування, що й MongoDB; модель автентифікації схожа, але надає не тільки ролі, а й привілеї, що модерують доступ до певних операції чи наборів даних.

ДОДАТОК Б

ДІАГРАМА ПРЕЦЕДЕНТІВ

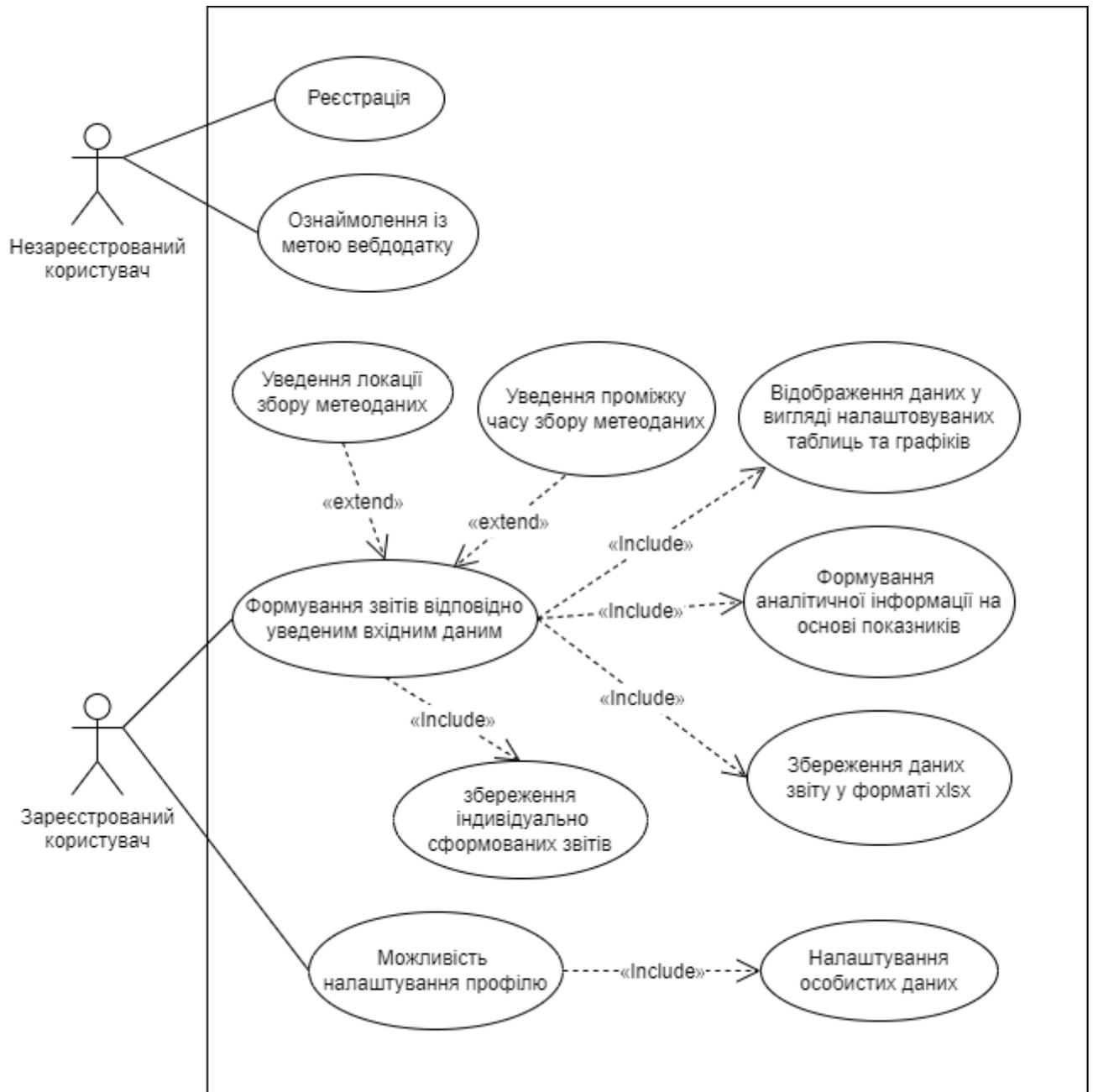


Рисунок Б.1 – Діаграма прецедентів

ДОДАТОК В

ДІАГРАМА ПОСЛІДОВНОСТЕЙ

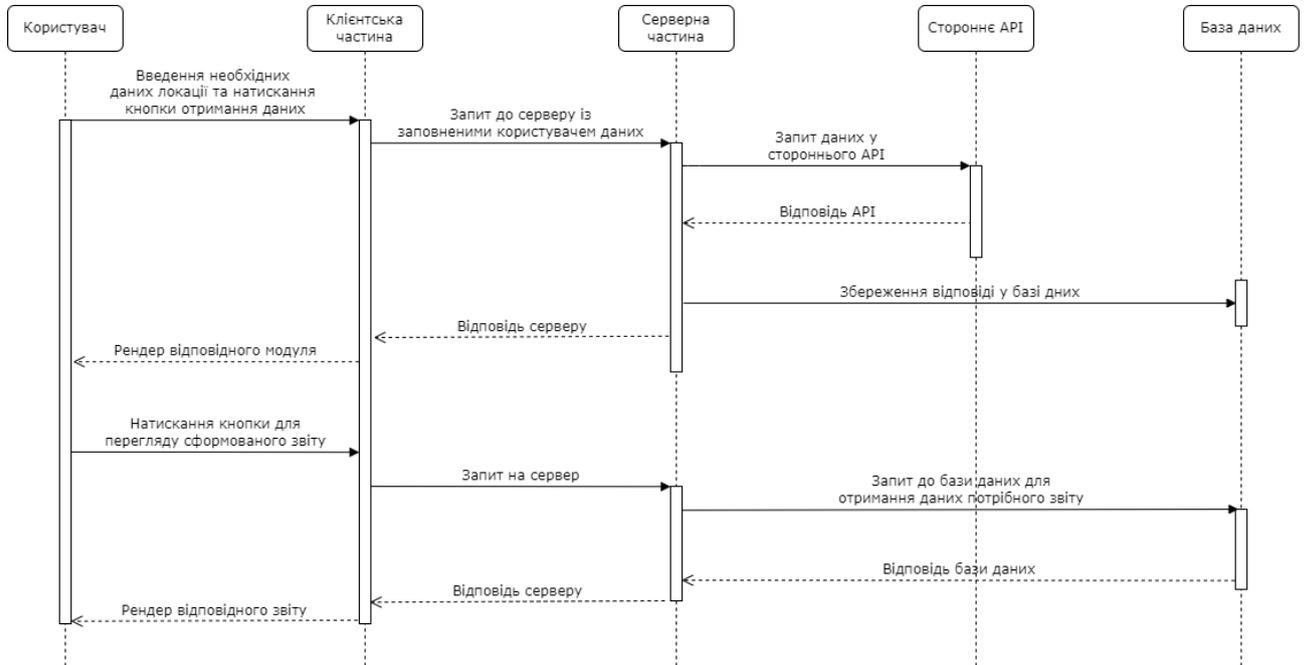


Рисунок В.1 – Діаграма послідовностей

ДОДАТОК Г
ПУБЛІКАЦІЇ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ
ЗА МАТЕРІАЛАМИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ПОЛТАВСЬКИЙ ДЕРЖАВНИЙ АГРАРНИЙ УНІВЕРСИТЕТ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЕКОНОМІКИ,
УПРАВЛІННЯ, ПРАВА ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ



МАТЕРІАЛИ

XVIII щорічного міждисциплінарного семінару

«СТУДЕНТСЬКІ РОБОТИ
ЗА НАУКОВОЮ ТЕМАТИКОЮ
КАФЕДРИ ІНФОРМАЦІЙНИХ
СИСТЕМ ТА ТЕХНОЛОГІЙ»

2 грудня 2021 року

Полтава – 2021

РЕДАКЦІЙНА КОЛЕГІЯ

Юрій УТКІН	- к.т.н., доцент, завідувач кафедри інформаційних систем та технологій, доцент кафедри;
Антоніна КАЛІНІЧЕНКО	- д.с.-г.н., професор, професор кафедри;
Юрій ПОНОЧОВНИЙ	- д.т.н., с.н.с., професор кафедри;
Вадим СЛЮСАР	- д.т.н., професор, професор кафедри;
Олена КОПШИНСЬКА	- к.ф.-м.н., доцент, професор кафедри;
Олег ОДАРУЩЕНКО	- д.т.н., доцент, професор кафедри;
Леонід ФЛЕГАНТОВ	- к.ф.-м.н., доцент, професор кафедри;
Юлія ВАКУЛЕНКО	- к.с.-г.н., доцент, доцент кафедри;
Лариса ДЕГТЯРЬОВА	- к.т.н., доцент, доцент кафедри;
Тетяна ДУГАР	- к.е.н., доцент, доцент кафедри;
Сергій ІВКО	- к.т.н., доцент кафедри;
Марина МАВРІНА	- к.т.н., доцент кафедри;
Олена ОДАРУЩЕНКО	- к.т.н., доцент кафедри;
Надія ПРОТАС	- к.с.-г.н., доцент, доцент кафедри;
Мирослав РЯБИЙ	- к.т.н., доцент кафедри;
Ігор СЛЮСАРЬ	- к.т.н., доцент, доцент кафедри;
Олексій ТИРТИШНІКОВ	- к.т.н., доцент, доцент кафедри;
Юрій УТКІН	- к.т.н., доцент, завідувач кафедри;
Наталія САЗОНОВА	- асистент.

Матеріали XVIII щорічного міждисциплінарного семінару «Студентські роботи за науковою тематикою кафедри інформаційних систем та технологій». Полтава: ПДАУ, 2 грудня 2021 р. 44 с.

У збірнику надруковані матеріали міждисциплінарного семінару студентських робіт за науковою тематикою кафедри інформаційних систем та технологій Полтавського державного аграрного університету.

Тези наводяться без змін та редагування. Відповідальність за зміст та редакцію тез несуть автори та наукові керівники.

Для студентів, аспірантів та викладачів вищих навчальних закладів.

© Полтавський державний аграрний університет (ПДАУ)

© Кафедра інформаційних систем та технологій

Ілля Поспелов
спеціальність «Інформаційні системи та технології»
Науковий керівник – д.т.н., с.н.с. Юрій Поночовний

ІНТЕРФЕЙС ВИХІДНИХ ДАНИХ ВЕБДОДАТКУ ДЛЯ ЗБИРАННЯ ТА АНАЛІЗУ МЕТЕОРОЛОГІЧНОЇ ІНФОРМАЦІЇ

Клімат та погода є важливими складовими, які впливають на діяльність людини. Складність предмету досліджень метеорології та потреби різних галузей економіки призвели до необхідності виникнення різних галузевих дисциплін, серед них виділяють наступні:

- синоптична метеорологія (прогнозування погоди);
- динамічна метеорологія (теоретичні основи науки);
- актинометрія (вивчення режимів надходження потоків сонячної радіації);
- кліматологія [1].

На сьогодні більшу частину інформації про процеси в атмосфері, погоду і клімат отримують шляхом спостережень. Результати спостережень використовуються для встановлення причинно-наслідкових зв'язків, які виникають в атмосфері. Їх проведення має на меті вивчення клімату і його змін та складання прогнозу погоди. Для збору та показу узагальнених метеорологічних даних використовують спеціальні програмно-апаратні комплекси. Доступ до метеорологічних даних можуть забезпечуватися спеціальними програмами або вебдодатками, які умовно можна розділити на ті, що збирають та обробляють дані, та ті, що надають користувачеві уже наявну інформацію у зручному вигляді, що дає змогу дослідити та зробити певні висновки [2].

Метою даної роботи є розробка вебдодатку для узагальнення та показу метеорологічних даних відповідно до запиту користувача. Для досягнення поставленої мети необхідно виконати ряд завдань:

- дослідити предметну область;
- провести аналіз програмних продуктів схожого призначення;
- обрати інструментальні засоби для розробки та доступні джерела, для отримання даних;
- створити демоверсію програмного продукту для тестування й отримання зворотнього зв'язку.

Оскільки розроблюваний додаток буде використовувати наявні дані за допомогою отримання їх по відкритому API, то має сенс розглядати додатки другої групи (що надають користувачеві наявну інформацію).

Заради отримання робочого MVP максимально швидко, було прийнято рішення використовувати MERN стек, що означає використання наступних технологій [3]:

- Mongo DB – документо-орієнтована система базами даних, яка не потребує опису схеми таблиць, а оперує даними у форматі ключ-значення,

– Node.js – міжплатформове середовище виконання JavaScript, що дозволяє створювати серверні і мережеві додатки;

– Express.js – вебфреймворк, що працює всередині середовища виконання Node.js [4];

– React.js – бібліотека для створення користувацьких інтерфейсів.

Для отримання даних використовується умовно безплатний API від Dark Sky [5].

Розроблений вебдодаток збирає інформацію про погодні умови за допомогою стороннього API, потім обробляє її та у зручному форматі показує користувачеві (рис. 1).

Графічне зображення у вигляді графіків, дозволяє проаналізувати зміну метеорологічних показників за обраний проміжок часу в історичному розрізі (рис. 2, 3).



Рис. 1. Приклад збору даних.

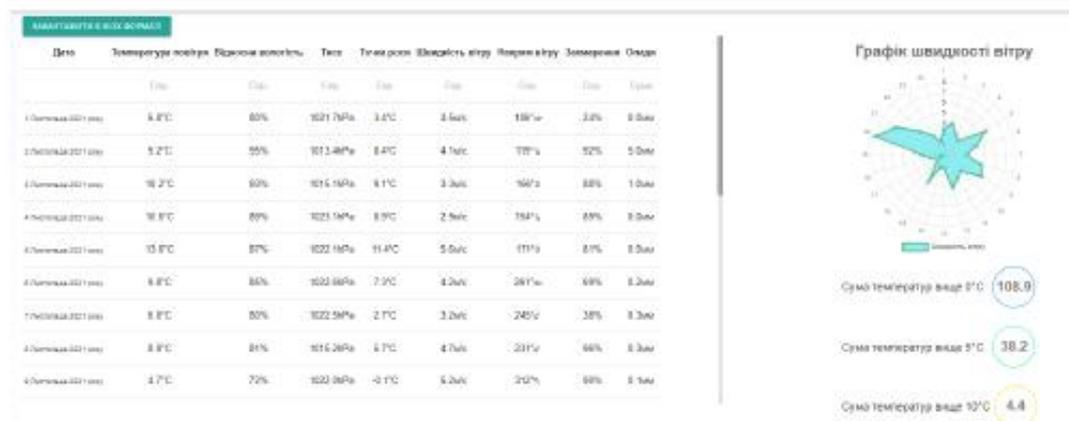


Рис. 2. Показ зібраних даних.

Створена система потенційно може бути використана у наукових, навчальних та виробничих галузях. На цей час програмний продукт має явні недоліки, які були виявлені реальними користувачами на етапі альфа-тестування [6].

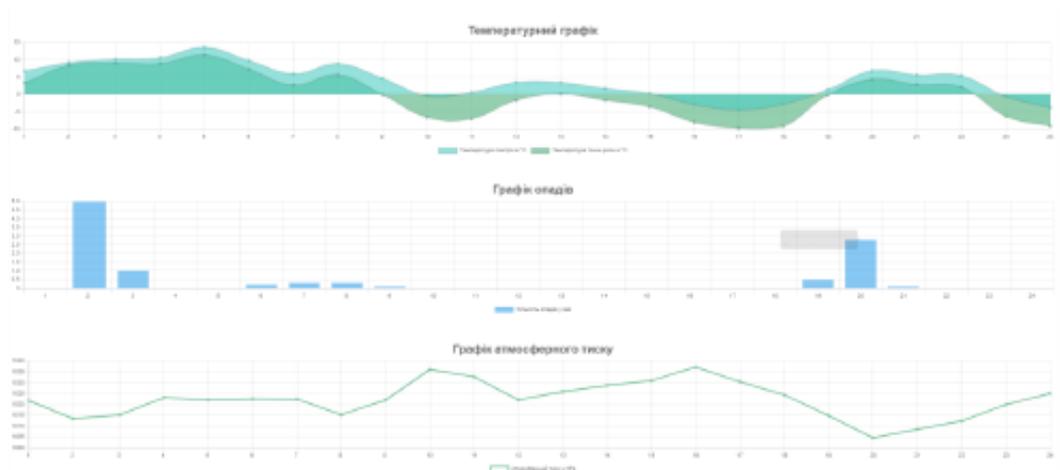


Рис. 3. Показ зібраних даних у вигляді графіків.

Список використаних джерел

1. Метеорологія і кліматологія: Підручник / Під редакцією д.ф.-м.н., професора Степаненка С.М. Одеса, 2008. 533 с.
2. Нажмудінова О.М. Синоптична метеорологія: Конспект лекцій. Одеса, 2010. 77 с.
3. Правове забезпечення Національної інноваційної системи у сучасних умовах: монографія / [С. В. Глібка, О. В. Розгон, Ю. В. Георгієвський та ін.]; за ред. С. В. Глібка, О. В. Розгон. Харків: НДІ прав. забезп. інновац. розвитку НАПрН України, 2020. 360 с.
4. Express/Node introduction – MDN web docs. URL: https://developer.mozilla.org/enUS/docs/Learn/Server-side/Express_Nodejs/Introduction (дата звернення 24.11.2021).
5. Dark Sky API. URL: <https://darksky.net/dev> (дата звернення 24.11.2021).
6. Авраменко А.С., Авраменко В.С., Косенюк Г.В. Тестування програмного забезпечення. Навчальний посібник. Черкаси: ЧНУ імені Богдана Хмельницького, 2017. 284 с.

DOI 10.36074/logos-20.05.2022.033

DEPENDENCE OF GROWTH PROCESSES OF (*ECHINACEA PURPUREA* (L.) MOENCH) OF THE FIRST YEAR OF VEGETATION ON AGROMETEOROLOGICAL FACTORS

ORCID ID: 0000-0003-0433-2996

Pospelov Sergii Viktorovych
Doctor of Agricultural Sciences, Professor,
Head of the Department of Agriculture and Agrochemistry named after V.I. Sazanov
Poltava State Agrarian University

ORCID ID: 0000-0002-8030-1166

Pospelova Ganna Dmytrivna
PhD in Agricultural Sciences,
Associate Professor of the Department of Plant Protection
Poltava State Agrarian University

Pospelov I.
student
Poltava State Agrarian University

UKRAINE

In recent decades, members of the genus *Echinacea* Moench. gained well-deserved popularity among those who grow medicinal plants. Natural reserves of raw materials in their homeland, North America, are quite limited, which prompted its introduction into culture in the world and Ukraine [1]. An analysis of world experience shows that among the nine species *Echinacea purpurea* (L.) Moench. is a leader in the scale of study and use [5,7].

In recent years, Ukraine has been actively studying the medicinal, fodder and honey properties of echinacea, systematic work on breeding, phytochemistry, in order to develop regional technologies for its cultivation to create a raw material base [3,4,6]. That is why the aim of our work was an in-depth study of ecological and biological features of *Echinacea purpurea* in the first year of the growing season. Morphometric studies were conducted in the conditions of industrial plantations of Poltava region, agrometeorological indicators according to meteorological stations with the help of a special program.

The ontogenesis of echinacea has certain patterns [2]. According to long-term data, the mass of aboveground part of one plant in the period from germination to June developed very slowly - the increase was only 0.8 g. From June to July the growth rate increased and the above figure was 10.9 g. Then it took place more intensively - in August the mass was 42.4 g, and in September - 74.5 g. A similar pattern was inherent in the development of the root system. During the period from germination to June, the mass was 0.1 g. In the following months, the rate of growth and development gradually increased: in July - 1.3, August - 8.9, September - 19.1 g. Thus, echinacea begins to grow rapidly, starting from August, and these processes are actively occurring until the end of the growing season.

It was also found that environmental factors affect the growth processes of echinacea. It was found that the mass of the aboveground part (MAP) significantly correlates with the amount of precipitation from the beginning of the calendar year

($r=0.459$) and from the date of sowing ($r=0.589$). The root system was more sensitive to changes in soil moisture. Its mass (MRS) was closely related to precipitation ($r=0.499$ and $r=0.640$, respectively). The effect of air temperature on the MAP was more significant. The correlation coefficient with temperatures above $0\text{ }^{\circ}\text{C}$ was $r=0.664$, above $5\text{ }^{\circ}\text{C}$ $r=0.675$ and above $10\text{ }^{\circ}\text{C}$ $r=0.684$, which indicates the importance of this factor for the development of echinacea. Characteristically, the relationship between the MRS and temperatures was not much greater, but more stable ($r=0.691-0.698$).

Regression analysis of experimental data was performed taking into account the reliability of correlations. Regarding the dependences of the growth of the aboveground part on the amount of precipitation since sowing, they have the form: $Y=0.3705e^{0.0015X}$, $R^2=0.5592$; $Y=0.0013X^{1.7458}$, $R^2=0.5385$. The equations that characterize the dependence of MRS growth on the sum of temperatures above $10\text{ }^{\circ}\text{C}$ have the following form: $Y=0.2787 e^{0.0044X}$, $R^2=0.7871$; $Y=3E-07X^{2.6923}$, $R^2=0.8599$. It can be concluded that the development of the aboveground part is more related to the temperature of the air during the growing season than to the amount of precipitation.

Similar patterns are characteristic of the growth of the root system. Dependence on precipitation is illustrated by models: $Y=0.0317 e^{0.0189X}$, $R^2=0.5377$; $Y=4E-05X^{2.0672}$, $R^2=0.5377$. More reliable were the equations that characterize the increase in the MRS depending on the sum of temperatures above 10°C : $Y=0.0207e^{0.0053X}$, $R^2=0.7661$; $Y=4E-09X^{3.0539}$, $R^2=0.7623$.

The identified patterns allow to optimize the technological process of growing echinacea in order to obtain medicinal raw materials.

References:

- [1] Porada A. A. (1998). Experience in growing *Echinacea purpurea* in the Forest-Steppe of Ukraine. *The study and use of echinacea: mater.* International Scientific Conf., Poltava. September 21-24, 86-89.
- [2] Pospelov S.V., Samorodov V.N. Kravchenko S. A. (2000). Dynamics of development of the aerial part of *Echinacea purpurea* in the first year of vegetation. *Visnik Poltava. of State Agrarian Institute*, 2, 19-21.
- [3] Pospelov S.V., Samorodov V.N., Mishchenko O.V. (2002). Features of the accumulation of hydroxycinnamic acids in *Echinacea purpurea* in the first year of vegetation. *Visnik PDAA*, 4, 34-38.
- [4] Pospelov S.V., Samorodov V.N., Pospelova A.D. (2013). Qualitative assessment of raw materials of echinacea varieties of PDAA breeding. *Innovative approaches to the study of echinacea: mater.* International scientific conf., Poltava, June 25-27, 180-186.
- [5] V. N. Samorodov and S. V. Pospelov (1999). Echinacea in Ukraine: half a century of experience of introduction and cultivation, 52 p.
- [6] V. N. Samorodov and S. V. Pospelov (2000). Echinacea at the turn of the XXI century: problems, trends, prospects (based on the materials of the conference in Kansas City, USA). *Bulletin of the Poltava State Agricultural Institute*, 3, 90-97.
- [7] Foster S. (1991). *Echinacea Nature's Immune Enhancer*. Rochester- Vermont, 150 p.



EUROPEAN
SCIENTIFIC
PLATFORM

ΛΟΓΟΣ

COLLECTION OF SCIENTIFIC PAPERS

E
S
P

WITH PROCEEDINGS OF THE III INTERNATIONAL
SCIENTIFIC AND PRACTICAL CONFERENCE

**«EDUCATION AND SCIENCE OF
TODAY: INTERSECTORAL ISSUES
AND DEVELOPMENT OF SCIENCES»**

MAY 20, 2022 • CAMBRIDGE, UK

Cambridge, United Kingdom
«P.C. Publishing House»
2022

Vinnytsia, Ukraine
«Yevropeiska naukova platforma»
2022

UDC 001(08)
E 25

<https://doi.org/10.36074/logos-20.05.2022>



Chairman of the Organizing Committee: Holdenblat M.

Responsible for the layout: Bilous T.

Responsible designer: Bondarenko I.



The conference is included in the catalog of International Scientific Conferences; certified by Euro Science Certification Group (Certificate № 22361 dated April 24th, 2022).

Conference proceedings are publicly available under terms of the Creative Commons Attribution 4.0 International License (CC BY 4.0).



Bibliographic descriptions of the conference proceedings are indexed by CrossRef, ORCID, Google Scholar, ResearchGate, OpenAIRE and OUCI.

E 25 **Education and science of today: intersectoral issues and development of sciences:** Collection of scientific papers «ΛΟΓΟΣ» with Proceedings of the III International Scientific and Practical Conference, Cambridge, May 20, 2022. Cambridge-Vinnytsia: P.C. Publishing House & European Scientific Platform, 2022.

ISBN 978-617-8037-80-2
ISBN 978-1-8380555-4-7 (PDF)

«European Scientific Platform», Ukraine
«P.C. Publishing House», United Kingdom

DOI 10.36074/logos-20.05.2022

Papers of participants of the III International Scientific and Practical Conference «Education and science of today: intersectoral issues and development of sciences», held in Cambridge, May 20, 2022, are presented in the collection of scientific papers.

UDC 001 (08)

ISBN 978-617-8037-80-2
ISBN 978-1-8380555-4-7 (PDF)

© Participants of the conference, 2022
© European Scientific Platform, 2022
© P.C. Publishing House, 2022
© Cambridge Data Science LTD, 2022