

# РОЗРОБЛЕННЯ ВЕБДОДАТКУ ДЛЯ СИСТЕМНОГО АНАЛІЗУ ТА ПРОГНОЗУВАННЯ ПОГОДНИХ УМОВ ІЗ ЗАСТОСУВАННЯМ ТЕХНОЛОГІЙ ШТУЧНОГО ІНТЕЛЕКТУ

Вінницький національний технічний університет

## **Анотація:**

*У роботі представлено результати розробки вебзастосунок для аналізу та прогнозування погодних умов. Описано архітектуру системи, що включає чотири ключові компоненти: базу даних PostgreSQL, серверну частину на основі Flask, зовнішній інтерфейс OpenWeather API та клієнтський інтерфейс на HTML/CSS/HTMX. Реалізовано сім функціональних модулів: авторизації, поточної погоди, збережених міст, прогнозування, графічної візуалізації, порівняння міст та кліматичної аналітики. Застосовано алгоритм машинного навчання DecisionTreeClassifier для оцінки комфортності погодних умов. Проведено порівняльний аналіз існуючих сервісів, що підтвердив доцільність розробки власного комплексного рішення.*

**Ключові слова:** вебзастосунок, аналіз даних, прогнозування погоди, Flask, PostgreSQL, машинне навчання, HTMX.

## **Abstract:**

*This paper presents the results of developing a web application for weather data analysis and forecasting. The system architecture encompasses four core components: a PostgreSQL database, a Flask-based backend, the OpenWeather API integration layer, and an HTML/CSS/HTMX client interface. Seven functional modules have been implemented: authentication, current weather, saved cities, forecasting, chart visualization, city comparison, and climate analytics. A DecisionTreeClassifier machine learning algorithm is applied to assess weather comfort levels. A comparative analysis of existing services confirmed the feasibility of developing a comprehensive custom solution.*

**Keywords:** web application, data analysis, weather forecasting, Flask, PostgreSQL, machine learning, HTMX.

## **Вступ**

Стрімкий розвиток інформаційних технологій та зростаючий попит на оперативну метеорологічну інформацію обумовлюють актуальність створення інтелектуальних вебсистем для аналізу та прогнозування погодних умов. За даними дослідження, понад 70% користувачів регулярно звертаються до погодних сервісів у повсякденній діяльності — для планування подорожей, сільськогосподарських робіт, спортивних заходів тощо.

Наявні на ринку сервіси, такі як AccuWeather, Weather.com та OpenWeather, орієнтовані переважно на відображення базових погодних показників та короткострокових прогнозів. Більшість із них не забезпечує можливості глибокого аналізу накопичених історичних даних, порівняння погодних умов між містами та застосування методів машинного навчання для оцінки комфортності погодних умов [1, 2, 3].

Метою роботи є розроблення вебзастосунок «WeatherApp» — системи з повним циклом роботи з метеорологічними даними: від їх отримання в реальному часі через OpenWeather API до аналізу, прогнозування та інтелектуальної оцінки. Для досягнення поставленої мети вирішено такі завдання: аналіз існуючих сервісів та виявлення їх недоліків; проектування багатомодульної архітектури системи; реалізація серверної та клієнтської частин; впровадження алгоритму машинного навчання для класифікації комфортності погодних умов [4].

## **Результати дослідження**

У процесі розробки вебзастосунок проведено аналіз предметної області та існуючих аналогів. Аналіз трьох провідних сервісів — AccuWeather, Weather.com та OpenWeather — дозволив виявити спільні недоліки: обмежені можливості аналізу історичних даних, відсутність інтеграції методів машинного навчання, перевантаженість або недостатність інтерфейсу, залежність від сторонніх

сервісів без гнучкого налаштування. Ці спостереження стали основою для формування вимог до розробленої системи та визначили її ключові переваги.

На основі виявлених потреб спроектовано архітектуру системи. Вебзастосунок реалізовано як багаторівневу систему, що включає чотири структурні компоненти: базу даних (PostgreSQL), серверну частину (Flask/Python), інтерфейс взаємодії з зовнішнім сервісом (OpenWeather API) та клієнтський інтерфейс (HTML, CSS, Tailwind, HTMX/Jinja2). Така архітектура забезпечує модульність, масштабованість і зручність підтримки системи (рис. 1) [5].

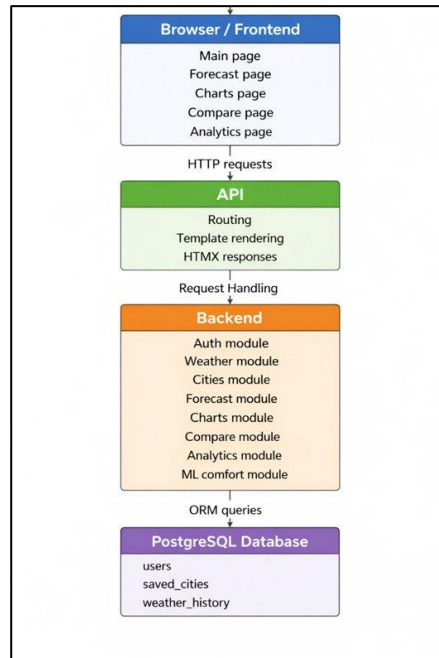


Рисунок 1. Діаграма архітектури веб-системи

Серверна частина реалізована на основі фреймворку Flask та включає сім функціональних модулів: авторизації (auth.py), поточної погоди (weather.py), збережених міст (cities.py), прогнозування (forecast.py), графічної візуалізації (charts.py), порівняння міст (compare.py) та кліматичної аналітики (analytics.py). Для взаємодії з базою даних PostgreSQL використовується ORM-бібліотека SQLAlchemy, що забезпечує логічну цілісність даних та каскадне видалення пов'язаних записів [6,7].

База даних містить три таблиці: users (дані користувача), saved\_cities (персональний список міст) та weather\_history (накопичені історичні погодні спостереження). Зв'язки між таблицями реалізовано за схемою «один до багатьох», що дозволяє зберігати персоналізовані погодні записи для кожного користувача та забезпечує підґрунтя для кліматичної аналітики.

Модуль машинного навчання реалізовано у файлі weather\_ml.py із застосуванням алгоритму DecisionTreeClassifier бібліотеки Scikit-learn [4]. Модуль поєднує два підходи: rule-based оцінку комфортності як початковий рівень розмітки даних та навчання класифікаційної моделі на накопичених у базі даних історичних спостереженнях. Вхідними ознаками є чотири метеорологічні показники: температура повітря (°C), відносна вологість (%), атмосферний тиск (гПа) та швидкість вітру (м/с). Вихідною змінною є категоріальна оцінка рівня комфортності погодних умов.

Процес застосування моделі відбувається у два етапи. На першому — формуються навчальні зразки на основі накопиченої у таблиці weather\_history інформації; rule-based функція присвоює кожному запису мітку комфортності. На другому — навчена модель застосовується до поточних погодних даних і збагачує відповідь сервера оцінкою рівня комфорту, яка відображається у блоці поточної погоди та на сторінці прогнозування. Вибір алгоритму дерева рішень зумовлений його інтерпретованістю, швидкістю навчання та ефективністю на відносно невеликих наборах даних [8].

Клієнтська частина вебзастосунку реалізована з використанням HTML, CSS, Tailwind CSS та HTMX. Завдяки HTMX реалізовано динамічне оновлення окремих блоків сторінки без повного перезавантаження, що підвищує швидкість роботи та зручність взаємодії. Розроблено п'ять сторінок: головну з поточною погодою, прогнозування, графіків, порівняння міст та кліматичної аналітики (рис. 2-5) [9].

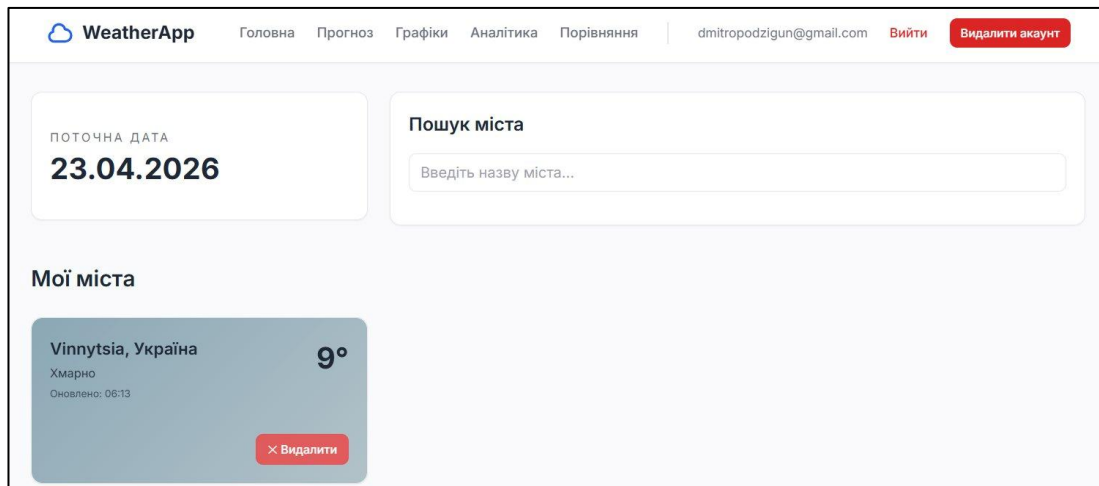


Рисунок 2. Головна сторінка вебдодатку після авторизації

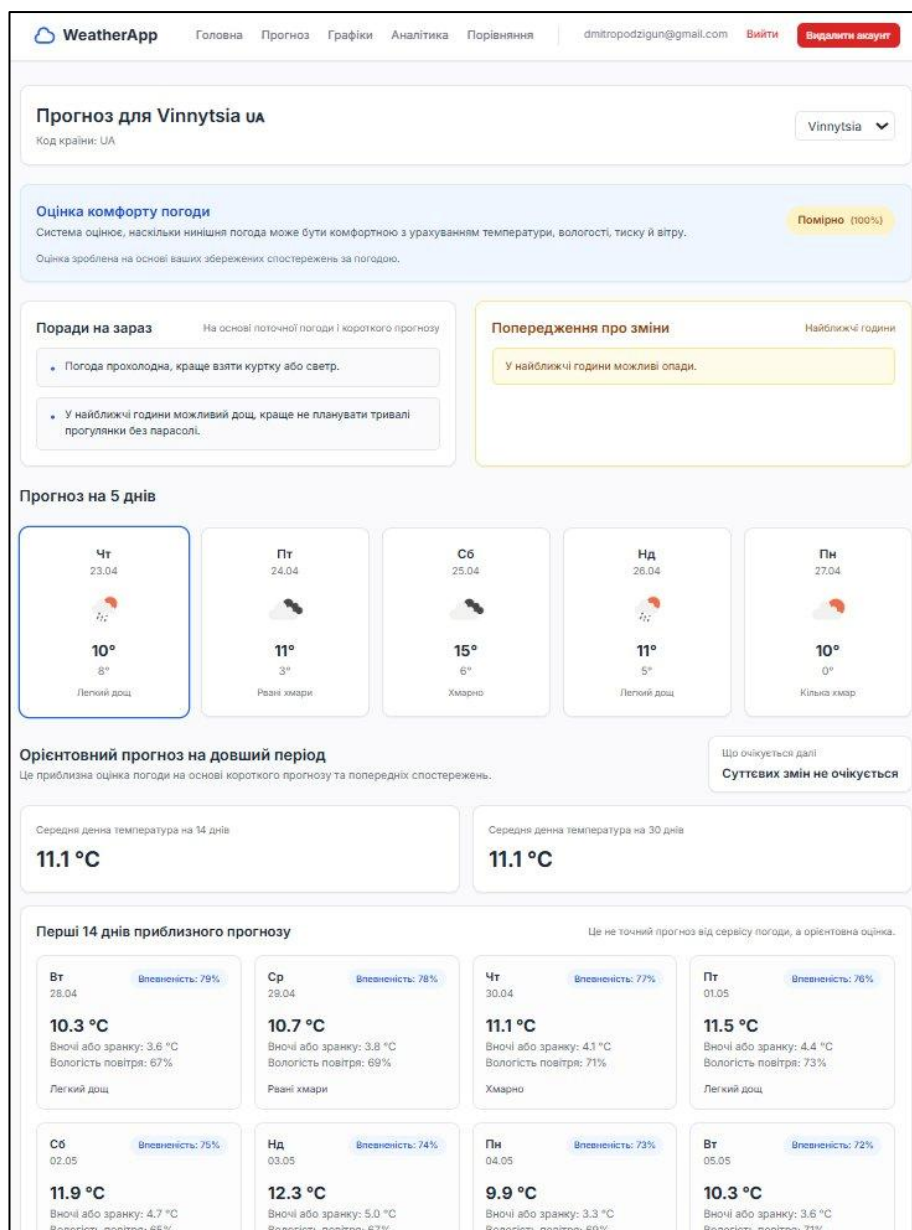


Рисунок 3. Сторінка прогнозування погодних умов

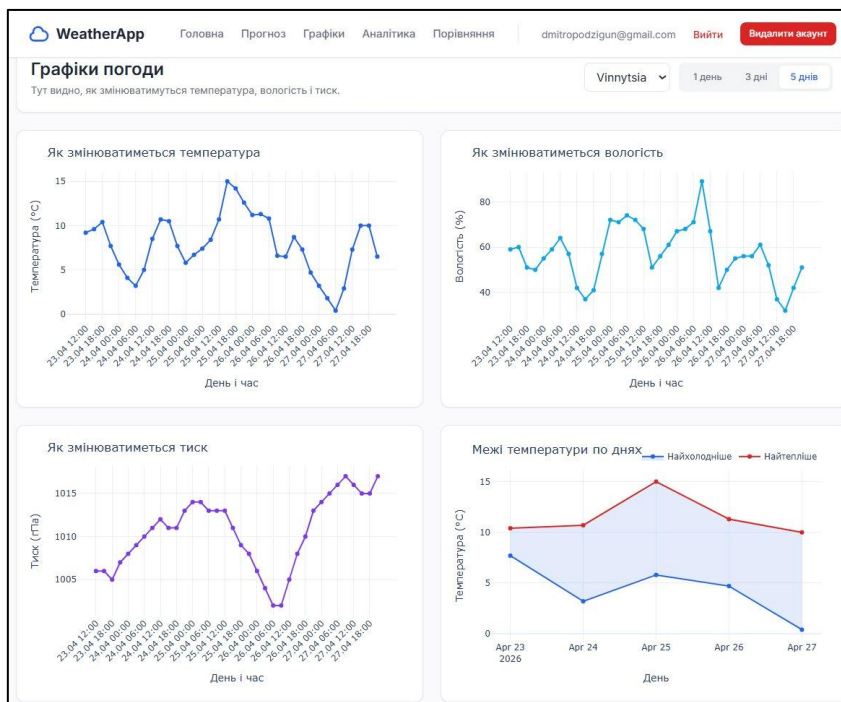


Рисунок 4. Сторінка графічної візуалізації погодних показників на найближчі 5 днів

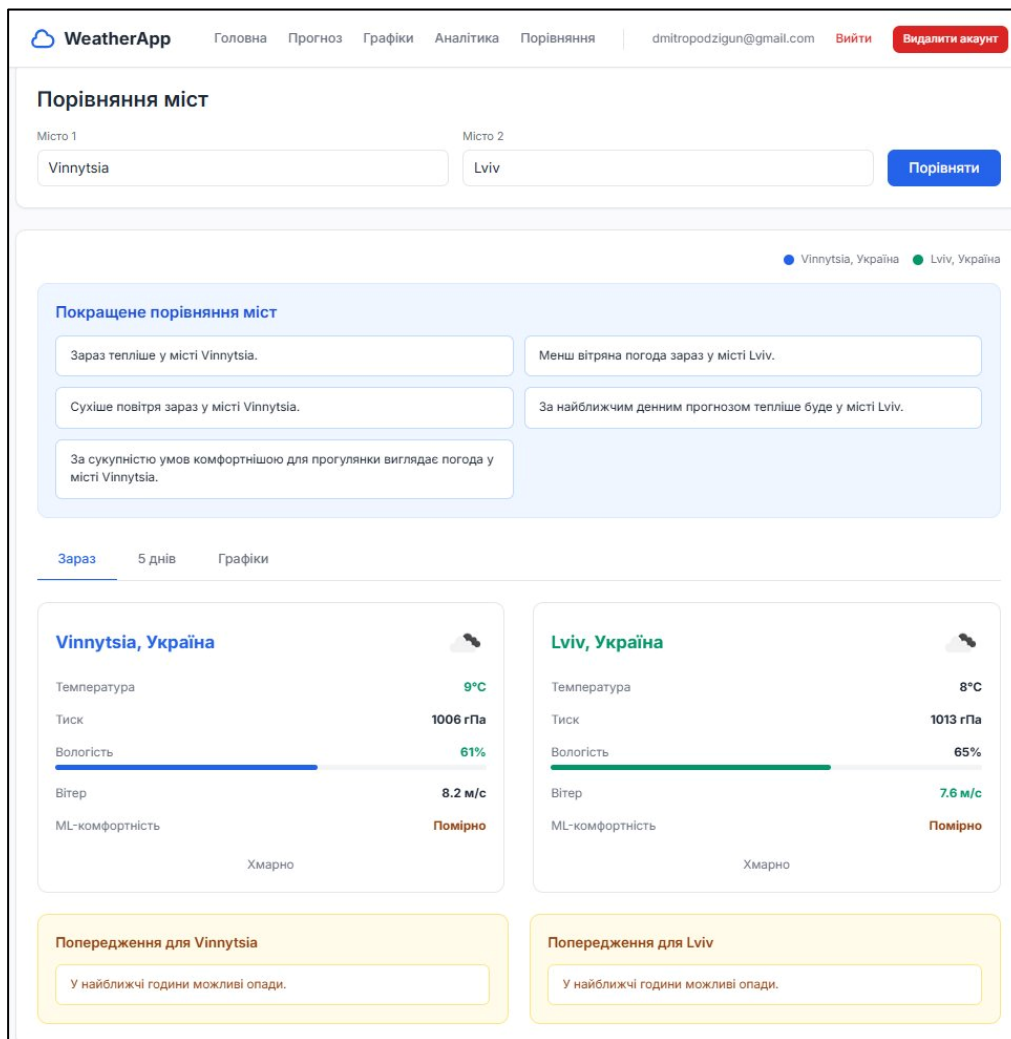


Рисунок 5. Сторінка порівняння погодних умов у двох містах

## Висновки

У результаті виконаної роботи розроблено вебзастосунок для аналізу та прогнозування погодних умов, що забезпечує повний цикл роботи з метеорологічними даними в єдиній екосистемі. Система орієнтована на широке коло користувачів, яким необхідний зручний інструмент для отримання, збереження, аналізу та прогнозування погодних даних.

Проведено аналіз трьох провідних погодних сервісів та визначено ключові недоліки ринкових рішень: відсутність глибокого аналізу накопичених даних, обмежені можливості персоналізації та відсутність інтеграції методів машинного навчання. На основі цього аналізу сформовано вимоги до системи та обґрунтовано доцільність розробки власного рішення.

Спроектовано та реалізовано багаторівневу архітектуру системи з чітким розподілом відповідальності між модулями. Серверна частина на Flask включає сім функціональних модулів, клієнтська частина реалізована з використанням HTMX для динамічного оновлення інтерфейсу без перезавантаження сторінки. Для збереження та обробки метеорологічних даних використано PostgreSQL з ORM SQLAlchemy.

Впроваджено модуль оцінки комфортності погодних умов на основі алгоритму DecisionTreeClassifier, що підвищує практичну цінність системи та розширює її аналітичні можливості порівняно з існуючими аналогами. Вихідний код розміщено у відкритому репозиторії GitHub [10], що забезпечує прозорість розробки та можливість подальшого розвитку проекту.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. OpenWeather API Documentation. URL: <https://openweathermap.org/api>
2. AccuWeather. Офіційний сайт. URL: <https://www.accuweather.com>
3. Weather.com. The Weather Channel. URL: <https://weather.com>
4. Pedregosa F., Varoquaux G., Gramfort A. et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*. 2011. Vol. 12. P. 2825–2830.
5. Flask Documentation. Flask Web Development Framework. URL: <https://flask.palletsprojects.com>
6. SQLAlchemy Documentation. The Python SQL Toolkit and ORM. URL: <https://docs.sqlalchemy.org>
7. PostgreSQL Documentation. URL: <https://www.postgresql.org/docs>
8. Haupt S. E., Pasini A., Marzban C. *Artificial Intelligence Methods in the Environmental Sciences*. Springer, 2009. 424 p.
9. HTMX Documentation. High power tools for HTML. URL: <https://htmx.org/docs>
10. Podzigun D. Weather Forecasting Web Application: GitHub repository. URL: [https://github.com/dimapodzigun/WeatherApp\\_course\\_project\\_web\\_dev](https://github.com/dimapodzigun/WeatherApp_course_project_web_dev)

**Подзігун Дмитро Костянтинович** – студент групи СА-246, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м. Вінниця, e-mail: [dmitropodzigun@gmail.com](mailto:dmitropodzigun@gmail.com).

**Жуков Сергій Олександрович** – к.т.н., доцент кафедри системного аналізу та інформаційних технологій, Вінницький національний технічний університет, e-mail: [sazhukov@gmail.com](mailto:sazhukov@gmail.com).

**Podzigun Dmytro K.** – student of group SA-24b, Faculty of Intelligent Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: [dmitropodzigun@gmail.com](mailto:dmitropodzigun@gmail.com).

**Zhukov Serhii O.** – Ph.D., Assistant Professor of the Department of System Analysis and Information Technologies, Vinnytsia National Technical University, Vinnytsia, e-mail: [sazhukov@gmail.com](mailto:sazhukov@gmail.com).