

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій**

ЗВІТ

До індивідуального проєкту №1

на тему «Погода в місті»

з дисципліни «WEB-орієнтовані технології. Frontend розробки»

Виконав(ла): студент(ка) групи Ткаченко Костянтин Олександрович
ІІ-331 (ПІБ)

Перевірів(ла): _____ (Проскура Світлана Леонідівна)

Київ – 2025 року

ВСТУП

У сучасному світі швидкий доступ до актуальної інформації про погоду є необхідністю для планування повсякденного життя, подорожей, роботи та відпочинку. Зростання використання мобільних та веб-додатків для моніторингу погоди

Актуальність теми полягає в потребі зручного, responsive веб-застосунку, який надає реальні дані про погоду в будь-якому місті, працює на всіх пристроях і не залежить від комерційних додатків.

Метою проєкту є розробка адаптивного веб-додатку «Погода в місті» на базі HTML5, CSS3 та JavaScript з пошуком за містом, відображенням поточної погоди та прогнозу, інтерактивними елементами (збереження улюблених міст у localStorage) та інтеграцією з безкоштовним API (наприклад, OpenWeatherMap або Open-Meteo).

1 АНАЛІЗ ТА МОДЕЛЮВАННЯ СИСТЕМИ

Тема WEB-застосунку «Погода в місті» — адаптивний веб-додаток для швидкого перегляду поточної погоди та прогнозу за назвою будь-якого міста світу.

Актуальність теми У 2025–2026 роках доступ до точної та швидкої інформації про погоду став критичним через посилення кліматичних змін, часті аномальні явища (екстремальні морози, спека, сильні опади, ожеледиця), зростання мобільності населення та популярність планування дня/подорожей у реальному часі. За даними різних джерел, понад 70 % користувачів смартфонів щодня перевіряють погоду, а попит на легкі веб-рішення без встановлення додатків постійно зростає.

Мета роботи Розробити адаптивний односторінковий веб-застосунок «Погода в місті», який дозволяє користувачу швидко отримувати поточну погоду та короткий прогноз за введеною назвою міста, зберігає обрані міста в локальному сховищі браузера та забезпечує зручний перегляд на пристроях будь-якого розміру.

Бізнес-логіка роботи Користувач вводить назву міста в поле пошуку → система надсилає запит до безкоштовного погодного API → отримує дані про температуру, відчувається, вологість, швидкість вітру, іконку погоди, опис та прогноз на 5 днів → відображає результат у зрозумілій картці. Користувач може додавати міста до «улюблених» → вони зберігаються в localStorage → при наступному відкритті показуються як швидкі кнопки/картки. При помилці (місто не знайдено, проблеми з мережею) відображається зрозуміле повідомлення. Усі дії відбуваються без перезавантаження сторінки.

Функціональні вимоги

1. Пошук погоди за назвою міста (поле введення + кнопка або Enter).

2. Відображення поточної погоди: температура, відчувається, опис, іконка, вологість, вітер, тиск.
3. Відображення прогнозу на 5 днів (мінімальний набір: дата, іконка, min/max температура).
4. Можливість додавати/видаляти міста зі списку улюблених.
5. Відображення списку улюблених міст як швидких кнопок або карток.
6. Обробка помилок (місто не знайдено, відсутність інтернету).
7. Автоматичне завантаження погоди для останнього переглянутого міста при відкритті сторінки.

Нефункціональні вимоги

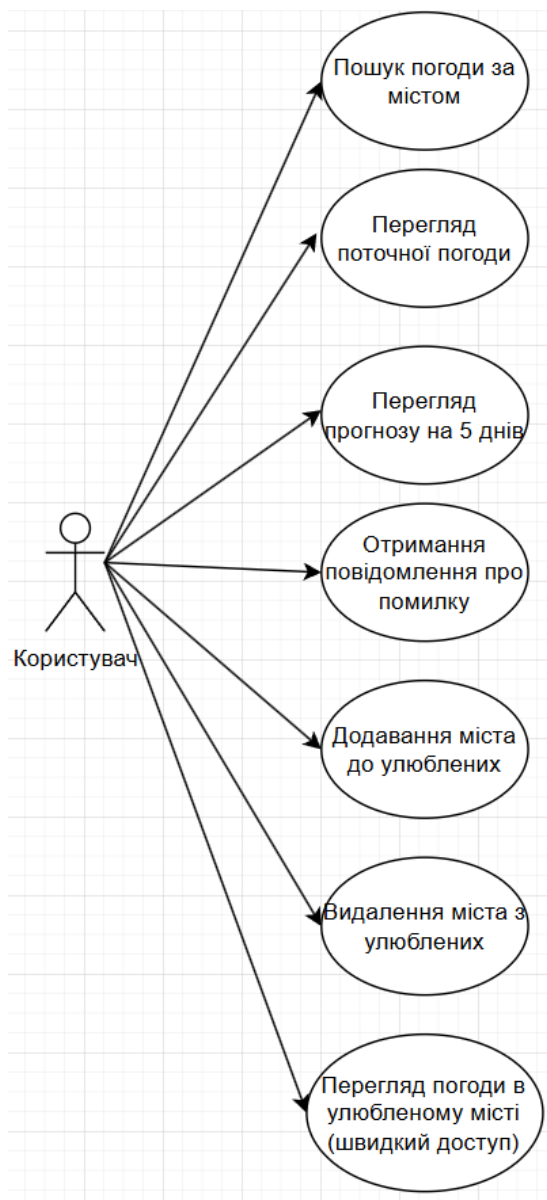
1. Повна адаптивність інтерфейсу під екрани від 320 px (мобільні) до 1920+ px (десктоп).
2. Час відповіді на запит користувача < 2 секунд (за умови стабільного інтернету).
3. Коректна робота в сучасних браузерях: Chrome, Firefox, Safari, Edge (останні 2–3 версії).
4. Використання відносних одиниць вимірювання та медіа-запитів для responsive дизайну.
5. Мінімальна вага сторінки (без важких бібліотек, оптимізовані зображення іконок).
6. Інтуїтивно зрозумілий інтерфейс, контрастність тексту за WCAG 2.1 (рівень AA).
7. Збереження даних у localStorage без обмеження терміну дії.

Use-case діаграма Актори:

- Користувач (основний)

Основні use-case:

1. Пошук погоди за містом
2. Перегляд поточної погоди
3. Перегляд прогнозу на 5 днів
4. Додавання міста до улюблених
5. Видалення міста з улюблених
6. Перегляд погоди в улюбленому місті (швидкий доступ)
7. Отримання повідомлення про помилку



Стек технологій

- HTML5
- CSS3 (flexbox, grid, media queries, анімації)
- JavaScript (ES6+, async/await, fetch API, DOM-маніпуляції)
- Зовнішній сервіс: Open-Meteo API або OpenWeatherMap API (безкоштовний шар)
- Бібліотеки: не використовуються (чистий стек), за бажанням можна додати dayjs для форматування дат

2 РОЗРОБЛЕННЯ АДАПТИВНОГО ІНТЕРФЕЙСУ WEB-ЗАСТОСУНКУ

Посилання на репозиторій - <https://github.com/kostpost/weather-app>

Розроблено адаптивний односторінковий веб-застосунок «Погода в місті» з повністю responsive дизайном, який коректно відображається на пристроях з роздільною здатністю від 320 px (мобільні) до 1920+ px (десктоп). Використано сучасні підходи CSS: flexbox для навігації та вирівнювання, grid для блоку прогнозу, медіа-запити для адаптації, CSS-анімації для плавного відкриття меню та появи контенту.

Структурні компоненти інтерфейсу

1. **Шапка (header)** Містить назву застосунку та пошукову форму. На екранах шириною менше 768 px трансформується в компактне бургер-меню з плавною анімацією відкриття.

Скріншот: (вставити скріншот десктопної версії + мобільної з відкритим бургер-меню)

Фрагмент коду (index.html):

```
<header>
  <div class="container">
    <div class="logo">Погода в місті</div>

    <nav class="nav-menu">
      <form id="search-form" class="search-form">
        <input
          type="text"
          id="city-input"
          placeholder="Введіть місто (наприклад, Київ)"
          autocomplete="off"
          required
        />
        <button type="submit">Пошук</button>
      </form>

      <div class="burger" id="burger-btn">
        <span></span>
        <span></span>
        <span></span>
      </div>
    </nav>
  </div>
</header>
```

Фрагмент коду (style.css — бургер-меню + анімація):

```
header {
  background: rgba(255, 255, 255, 0.95);
  box-shadow: 0 2px 10px rgba(0,0,0,0.1);
  position: sticky;
  top: 0;
  z-index: 100;
  padding: 1rem 0;
}

.burger {
  display: none;
  flex-direction: column;
  gap: 5px;
  cursor: pointer;
}

.burger span {
  width: 28px;
  height: 3px;
  background: #2c3e50;
  border-radius: 2px;
  transition: all 0.3s;
}

@media (max-width: 768px) {
  .nav-menu {
    position: absolute;
    top: 100%;
    left: 0;
    width: 100%;
    background: white;
    flex-direction: column;
    padding: 1rem;
    box-shadow: 0 4px 10px rgba(0,0,0,0.1);
    transform: translateY(-100%);
    transition: transform 0.4s ease;
  }

  .nav-menu.active {
    transform: translateY(0);
  }

  .search-form {
    width: 100%;
    flex-direction: column;
  }

  .search-form input {
    width: 100%;
  }

  .burger {
    display: flex;
  }
}
```


Основний блок (main) Містить картку поточної погоди та сітку прогнозу на 5 днів. Використано CSS Grid для динамічної перебудови: на десктопі — кілька колонок, на планшеті — дві, на смартфоні — одна.

Скріншот: (вставити скріншот десктопної версії + мобільної версії з однією колонкою)

Фрагмент коду (index.html):

```
<main class="container main-content">
  <section class="intro">
    <h2>Дізнайтесь погоду в будь-якому місті світу</h2>
    <p>Введіть назву міста у пошуку вище</p>
  </section>

  <div id="current-weather" class="weather-card hidden">
  </div>

  <div id="forecast" class="forecast-grid hidden">
  </div>

  <div id="favorites" class="favorites-list">
    <h3>Улюблені міста</h3>
    <div id="favorites-buttons"></div>
  </div>
</main>
```

Фрагмент коду (style.css):

```
.weather-card,
.forecast-grid,
.favorites-list {
  background: rgba(255,255,255,0.9);
  border-radius: 12px;
  padding: 1.8rem;
  margin: 2rem 0;
  box-shadow: 0 6px 20px rgba(0,0,0,0.08);
}

.hidden {
  display: none;
}

.forecast-grid {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(160px, 1fr));
  gap: 1.2rem;
}
```

Нижній колонтитул (footer) Містить копірайт, посилання на API та GitHub.

Фрагмент коду (index.html):

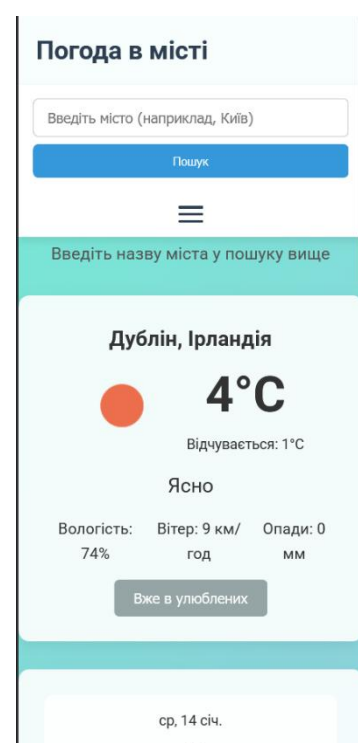
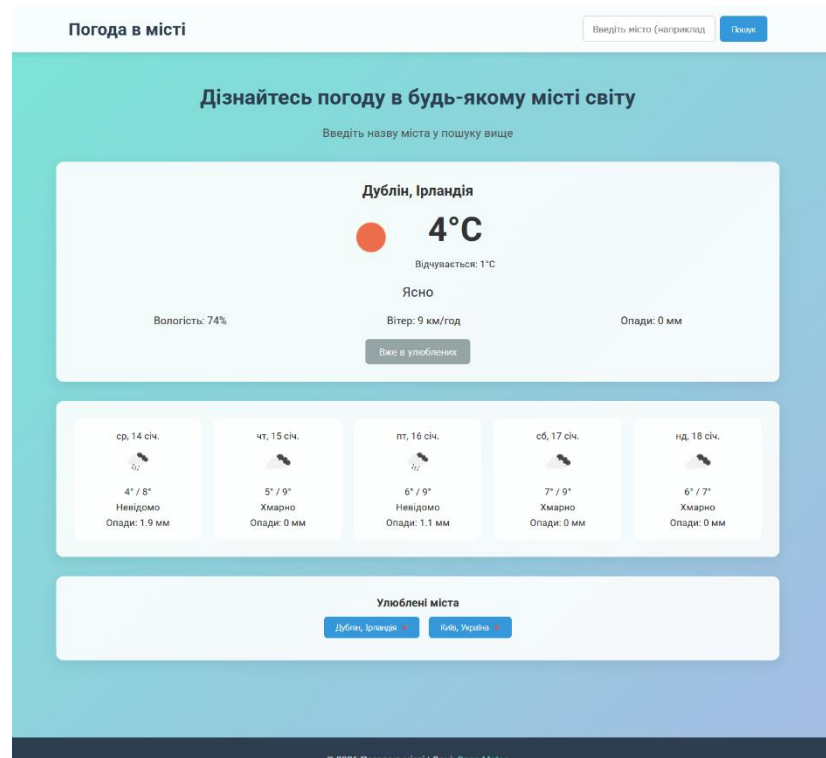
```
<footer>
  <div class="container">
    <p>© 2026 Погода в місті | Дані: <a href="https://open-meteo.com/"
target="_blank">Open-Meteo</a></p>
  </div>
</footer>
```

Фрагмент коду (style.css):

```
footer {
  background: #2c3e50;
  color: white;
  text-align: center;
  padding: 1.5rem 0;
  margin-top: 3rem;
}

footer a {
  color: #74ebd5;
  text-decoration: none;
}

footer a:hover {
  text-decoration: underline;
}
```



3 ПРОГРАМНА ЛОГІКА (JAVASCRIPT)

У проєкті «Погода в місті» JavaScript відповідає за всю інтерактивність, асинхронну роботу з API, маніпуляцію DOM, обробку подій та збереження даних користувача. Логіка побудована на чистому ES6+ без фреймворків, з використанням `async/await` та `fetch`.

3.1. Опис функціоналу

У проєкті реалізовано такі основні функціональні можливості:

- Динамічне оновлення контенту сторінки Дані про погоду та прогноз завантажуються та відображаються без перезавантаження сторінки шляхом маніпуляцій DOM (заміна вмісту елементів `current-weather` та `forecast`).
- Інтерактивність інтерфейсу Реалізовано обробку кліків на кнопках пошуку, додавання/видалення улюблених міст, відкриття бургер-меню, `hover`-ефекти та анімації появи контенту.
- Збереження даних Використано `localStorage` для зберігання списку улюблених міст (масив назв міст). Дані зберігаються між сеансами браузера.
- Асинхронні операції Застосунок виконує два послідовні `fetch`-запити: геокодинг (пошук координат за назвою міста) та отримання прогнозу погоди. Використовується `async/await` для читабельного коду.

3.2. Обробка подій

Обробка подій реалізована за принципами DOM-event handling з використанням `addEventListener`.

Основні події, які були реалізовані:

- Події кліку (`click`)
 - Відкриття/закриття бургер-меню

- Надсилення форми пошуку
 - Додавання/видалення міста з улюблених
 - Перехід до погоди обраного улюбленого міста
- Події форми (submit)
 - Обробка форми пошуку міста з `preventDefault()` для уникнення перезавантаження сторінки
- Події завантаження (load)
 - Ініціалізація списку улюблених міст та перевірка хешу в URL при завантаженні сторінки

3.3. Валідація форм

Валідація форми пошуку міста включає:

- Перевірку обов'язкових полів Поле вводу має атрибут `required`, тому браузер не дозволяє надіслати порожню форму.
- Перевірку формату даних Використовується `trim()` для видалення пробілів на початку/кінці, а також регулярні вирази для очищення від зайвих символів (наприклад, `-ukraina` з хешу).
- Валідацію в реальному часі Не реалізована повна `live`-валідація (можна додати `oninput`), але при помилці API показується повідомлення.
- Підсумкову валідацію перед відправкою Якщо поле порожнє — запит не виконується.
- Обробку серверних помилок При помилці геокодингу або `forecast API` відображається повідомлення типу «Місто не знайдено. Спробуйте іншу назву.».

Клієнтська валідація виконується чистим JavaScript.

3.4. Робота з API (fetch)

Застосунок здійснює асинхронні GET-запити до двох ендпоінтів `Open-Meteo API`:

- GET-запит до геокодингу — отримання координат за назвою міста

- GET-запит до forecast — отримання поточної погоди та прогнозу на 5 днів

Особливості реалізації:

- Використання нативного fetch() з async/await
- Обробка станів: показ курсора «wait» під час завантаження (loading), приховування/показ блоків (success/error)
- Обробка HTTP-помилки: перевірка response.ok не потрібна (API повертає error в JSON), тому перевіряється наявність geoData.results та data.error
- Відображення помилок користувачу через showError()

3.5. Опис бізнес-логіки

Бізнес-логіка визначає правила роботи застосунку:

- Алгоритми роботи з даними
 - Пошук міста → геокодинг → отримання координат → запит прогнозу
 - Очищення назви міста від зайвого (наприклад, -ukraina з хешу)
 - Генерація slug для URL-хешу (toUrlSlug)
- Умовна логіка
 - Показ кнопки «Додати в улюблені» тільки якщо міста ще немає в списку
 - Автоматичне завантаження міста тільки якщо є хеш у URL (#kyiv)
 - При помилці — показ повідомлення замість картки погоди
- Правила взаємодії користувача з UI
 - Якщо поле пошуку порожнє — запит не виконується
 - Якщо місто вже в улюблених — кнопка «Додати» стає неактивною
 - Якщо API повернув помилку — показується зрозуміле повідомлення

- Обробка складних сценаріїв
 - Підтримка прямого доступу за посиланням (#kyiv)
 - Збереження улюблених міст між сеансами
 - Ручне додавання в улюблені (без автододавання)

Бізнес-логіка структурована в окремих функціях (`loadWeather`, `renderCurrentWeather`, `renderFavorites` тощо), що дозволяє легко її розширювати (наприклад, додати геолокацію або теми).

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Оформлення текстових документів у навчальному процесі. Стандарт організації (кафедри) СОУ ІСТ 01-22. Для студентів кафедри інформаційних систем та технологій.
 2. Conventional Commits URL: <https://www.conventionalcommits.org/en/v1.0.0/#commit-message-with-description-and-breaking-change-footer> (дата звернення 08.08.2025).
 - 3.
 4. HTML Підручник. W3Schools українською. URL: <https://w3schoolsua.github.io/html/index.html>.
 5. HTML. Мова для створення веб-сторінок. W3Schools українською. URL: <https://w3schoolsua.github.io/index.html#gsc.tab=0>
 6. HTML Довідник тегів. URL: <https://w3schoolsua.github.io/tags/index.html>
 7. CSS Підручник. W3Schools українською. URL: <https://w3schoolsua.github.io/css/index.html#gsc.tab=0>
 8. HTML. Language for creating web pages. W3Schools (English). URL: https://w3schoolsua.github.io/index_en.html#gsc.tab=0 (дата звернення: 08.08.2025).
 9. HTML Tutorial. W3Schools (English). URL: https://w3schoolsua.github.io/html/index_en.html#gsc.tab=0 (дата звернення: 08.08.2025).
 10. Підручники HTML та CSS. HTMLBook. URL: https://htmlbook.at.ua/news/tutorial_html/1-0-1 (дата звернення: 08.08.2025).
 11. Learn to Code. With the world's largest web developer site. W3Schools. URL: <https://www.w3schools.com/> (дата звернення: 08.08.2025).
 12. Український веб-довідник. URL: <https://css.in.ua/> (дата звернення: 08.08.2025).
 13. Довідник по HTML тегам. URL: <https://css.in.ua/html/tags> (дата звернення: 08.08.2025).
 14. Довідник HTML тегів. HTML-CSS.co.ua. URL: <https://html-css.co.ua/dovidnuk-html-tags/> (дата звернення: 08.08.2025).
 15. HTML конструювання. HTMLBook. URL: <http://htmlbook.in.ua/>; <http://htmlbook.in.ua/tegs-html/>; <http://htmlbook.in.ua/pr01-html/> (дата звернення: 08.08.2025).
- CSS-TRICKS. Guides. Articles. URL: <https://css-tricks.com/> (дата звернення: 08.08.2025).