

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО”**

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

До захисту допущено:  
**Завідувач кафедри**  
Сергій СТИРЕНКО

(підпис)

“\_\_” \_\_\_\_\_ 2023 р.

**Дипломний проєкт**

**на здобуття ступеня бакалавра**

**за освітньо-професійною програмою “Комп’ютерні системи та мережі”  
спеціальності 123 “Комп’ютерна інженерія”**

на тему: Прогнозувальник погоди на основі технології PWA

Виконав (-ла): студент (-ка) 4 курсу, групи ІВ-93  
(шифр групи)

Лінійчук Данило Васильович

(прізвище, ім’я, по батькові)

(підпис)

Керівник доцент, к.т.н., Павлов В.Г.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Консультант (нормоконтроль)ст. викл. Виноградов Ю.М.

(назва розділу) (посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Рецензент д.т.н., проф. кафедри ІСТ, Корнієнко Б.Я.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Засвідчую, що у цьому дипломному проєкті  
немає запозичень з праць інших авторів без  
відповідних посилань.

Студент \_\_\_\_\_  
(підпис)

Київ – 2023 р.

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО”**

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Рівень вищої освіти – перший (бакалавр)

Освітньо-професійна програма

“Комп’ютерні системи та мережі”

спеціальності 123 “Комп’ютерна інженерія”

**ЗАТВЕРДЖУЮ**

**Завідувач кафедри**

Сергій СТИРЕНКО

\_\_\_\_\_ (підпис)

“\_\_” \_\_\_\_\_ 2023 р.

**ЗАВДАННЯ**

на бакалаврський дипломний проєкт студента

Лінійчука Данила Васильовича

1. Тема проєкту Прогнозувальник погоди на основі технології PWA  
керівник проєкту Павлов В.Г., к.т.н., доцент  
(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)  
затверджені наказом по університету від 31 травня 2023 року No2102-с
2. Термін здачі студентом закінченого проєкту
3. Вихідні дані до проєкту технічна документація, теоретичні дані.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які розробляються)  
Розділ 1. Актуальність проєкту та огляд існуючих рішень.  
Розділ 2. Технології розробки клієнт-серверної системи.  
Розділ 3. Проектування та розробка системи.  
Розділ 4. Огляд реалізованої системи.

5. Перелік графічного матеріалу (з точним позначенням обов'язкових креслень) діаграма взаємодії модулів(структурна схема), модель зберігання даних(функціональна схема), діаграма діяльності системи(принципова схема)
6. Консультанта проєкту, з вказівкою розділів проєкту, які до них вносяться

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Нормконтроль	Винградов Ю.М.		

7. Дата видачі завдання 15.12.2022

#### Календарний план

№ П/П	Найменування етапів дипломного проєкту	Терміни виконання етапів проєкту	Примітки
1.	<i>Затвердження теми проєкту</i>	<i>10.12.2022 - 15.12.2022</i>	
2.	<i>Вивчення та аналіз завдання</i>	<i>15.12.2022 - 15.03.2023</i>	
3.	<i>Розробка архітектури та загальної структури системи</i>	<i>15.03.2023 - 25.03.2023</i>	
4.	<i>Розробка структур окремих підсистем</i>	<i>25.03.2023 - 05.04.2023</i>	
5.	<i>Програмна реалізація системи</i>	<i>05.04.2023 - 15.04.2023</i>	
6.	<i>Оформлення пояснювальної записки</i>	<i>15.04.2023 - 12.05.2023</i>	
7.	<i>Захист програмного продукту</i>	<i>25.04.2023</i>	
8.	<i>Передзахист</i>	<i>23.05.2023</i>	
9.	<i>Захист</i>	<i>23.06.2023</i>	

Студент-дипломник \_\_\_\_\_ Данило ЛІНІЙЧУК  
(підпис)

Керівник проєкту \_\_\_\_\_ Валерій ПАВЛОВ  
(підпис)

## **Анотація**

В дипломній роботі був розроблений сервіс для прогнозування погоди на основі технології PWA. Сервіс призначений для наочного прогнозування погоди в режимі реального часу або у фоновому режимі та для використання в якості альтернативи існуючим рішенням.

Проект був створений на мові TypeScript з використанням бібліотек React та Redux в інтегрованому середовищі розробки Visual Studio Code.

Для надання можливості використання застосунку без участі браузера була розроблена логіка для service-worker для кешування даних та встановлення застосунку на користувацький пристрій.

## **Annotation**

In the thesis, a service for weather forecasting based on PWA technology was developed. The service is intended for visual weather forecasting in real-time or in the background and to be used as an alternative to existing solutions.

The project was created in the TypeScript language using the React and Redux libraries in the Visual Studio Code integrated development environment.

To enable the use of the application without the participation of the browser, a logic was developed for the service-worker to cache data and install the application on the user's device.

справки	Формат	Значення	Найменування	Кіл. листів	№ екземпляр	Додаток
			Документація загальна			
			Знову розроблена			
	<i>A4</i>	<i>ІАЛЦ.467200.001 ОА</i>	Прогнозувальник погоди на основі технології PWA	1		
			Опис альбому			
	<i>A4</i>	<i>ІАЛЦ.467200.002 ТЗ</i>	Прогнозувальник погоди на основі технології PWA	3		
			Технічне завдання			
	<i>A4</i>	<i>ІАЛЦ.467200.003 ПЗ</i>	Прогнозувальник погоди на основі технології PWA	67		
			Пояснювальна записка			
	<i>A4</i>	<i>ІАЛЦ.467200.004 Д1</i>	Прогнозувальник погоди на основі технології PWA	1		
			Діграма взаємодії модулів (структурна схема)			
	<i>A1</i>	<i>ІАЛЦ.467200.005 Д2</i>	Прогнозувальник погоди на основі технології PWA	1		
			Модель зберігання даних(функціональна схема)			
	<i>A1</i>	<i>ІАЛЦ.467200.006 ДЗ</i>	Прогнозувальник погоди на основі технології PWA	1		
			Діаграма діяльності системи (принципова схема)			

					<b><i>ІАЛЦ.467200.001 ОА</i></b>		
<i>Зм</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підп.</i>	<i>Дата</i>			
<b><i>Розроб</i></b>	Лінійчук Д.В				<b><i>Система захисту інформації в бездротових мережах</i></b> <b><i>Опис альбому</i></b>		
<b><i>Перев</i></b>	Павлов В.Г.						
					<b><i>КПІ ім. Ігоря Сікорського</i></b> <b><i>ФІОТ ІВ-93</i></b>		

Технічне завдання на дипломний проєкт  
Зміст

1.	НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ .....	2
2.	ПІДСТАВА ДЛЯ РОЗРОБКИ .....	2
3.	МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ .....	2
4.	ДЖЕРЕЛА РОЗРОБКИ .....	2
5.	ТЕХНІЧНІ ВИМОГИ .....	2
	5.1. ВИМОГИ ДО РОЗРОБЛЮВАНОВОГО ПРОДУКТУ .....	2
	5.2. ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....	2
	5.3. ВИМОГИ ДО АПАРАТНОЇ ЧАСТИНИ .....	2
6.	ЕТАПИ РОЗРОБКИ .....	3

					<b>ІАЛЦ.467200.002 ТЗ</b>		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Лінійчук Д.В.			Літ.	Арк.	Акрушів
Перевір.		Павлов В.Г.			1	3	
Н. Контр.		Виноградов Ю.М.			КПІ ім. Ігоря Сікорського ФІОТ ІВ-93		
Затверд.		Стіренко С.Г.			Технічне завдання		

## 1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

Дане технічне завдання поширюється на розробку веб-додатку для прогнозування погоди а також на подальше вдосконалення та підтримку розробленої системи. Область застосування: альтернатива існуючим сайтам та додаткам погнозування погоди

## 2. ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на виконання бакалаврського проекту по освітньо-професійної програми “Комп’ютерні системи та мережі” спеціальності 123 “Комп’ютерна інженерія”, затверджене кафедрою Обчислювальної техніки Національного технічного Університету України “Київський Політехнічний інститут імені Ігоря Сікорського”.

## 3. МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ

Метою даного проекту є розробка прогнозувальника погоди на основі технології PWA, який надаватиме користувачам зручний спосіб отримання погодової інформації.

## 4. ДЖЕРЕЛА РОЗРОБКИ

Джерелом розробки є публікації в періодичних виданнях, довідники на публікації в Інтернеті щодо даної теми.

## 5. ТЕХНІЧНІ ВИМОГИ

### 5.1. Вимоги до розроблюваного продукту

- Інтеграція технології PWA – реалізація всіх можливостей запропонованих даною технологією.
- Наочне відображення прогнозів погоди – застосунок повинен в повній мірі забезпечити наочність прогнозування погоди та мати простий та зрозумілий інтерфейс
- Використання сучасних засобів та методів розробки веб-застосунків.

### 5.2. Вимоги до програмного забезпечення

- Операційна система Windows, Linux чи Mac
- Мобільні пристрої на операційній системі Android або IOS
- Браузери версій 2018 року та вище

### 5.3. Вимоги до апаратно частити

- Комп’ютер на базі процесора Intel Pentium та вище
- Оперативної пам'яті не менше 512 Мбайт
- Вільний простір жорсткого диску не менше 100 Мбайт

					ІАЛЦ.467200.002 ТЗ	Арк.
						2
Змн.	Арк.	№ докум.	Підпис	Дата		

## 6. ЕТАПИ РОЗРОБКИ

	Дата
6.1. Вивчення літератури	17.04.2023
6.2. Складання та узгодження технічного завдання	01.05.2023
6.3. Проектування програмного забезпечення	08.05.2023
6.4. Програмна реалізація продукту	15.05.2023
6.5. Тестування програмного забезпечення	22.05.2023
6.6. Налаштування та виправлення помилок	29.05.2023
6.7. Оформлення документації дипломної роботи	07.06.2023

					ІАЛЦ.467200.002 ТЗ	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

## Зміст

РОЗДІЛ 1 АКТУАЛЬНІСТЬ ПРОЕКТУ ТА ОГЛЯД ІСНУЮЧИХ РІШЕНЬ.....	4
1.1 АКТУАЛЬНІСТЬ ПРОЕКТУ.....	4
1.2 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ .....	5
1.2.1 ОГЛЯД САЙТІВ ДЛЯ ПРОГНОЗУВАННЯ ПОГОДИ .....	5
1.2.1 ОГЛЯД САЙТІВ ДЛЯ ПРОГНОЗУВАННЯ ПОГОДИ .....	12
1.3 ПОРІВНЯННЯ ІСНУЮЧИХ РІШЕНЬ .....	17
Висновок до розділу 1 .....	21
РОЗДІЛ 2 ТЕХНОЛОГІЇ РОЗРОБКИ КЛІЄНТ-СЕРВЕРНОЇ СИСТЕМИ .....	22
2.1 ОГЛЯД ТЕХНОЛОГІЇ PWA.....	22
2.1.1 Переваги та недоліки PWA.....	22
2.1.2 Складові компоненти PWA .....	26
2.2 РОЗРОБКА СЕРВІСНОЇ ЧАСТИНИ НА ПЛАТФОРМІ FIREBASE .....	27
2.3 ОГЛЯД ТЕХНОЛОГІЙ ДЛЯ СТВОРЕННЯ СЕРВЕРНОЇ ЧАСТИНИ .....	30
2.3.1 Вибір мови програмування.....	30
2.3.2 Бібліотека React js .....	32
2.3.3 Бібліотека Redux.....	36
2.3.4 Препроцесор SCSS .....	38
Висновок до розділу 2 .....	39
РОЗДІЛ 3 ПРОЕКТУВАННЯ ТА РОЗРОБКА СИСТЕМИ .....	41
3.1 ОПИС ВИМОГ ДЛЯ ФУНКЦІОНАЛУ ДОДАТКУ .....	41
3.2 РОЗРОБКА АРХІТЕКТУРИ ВЕБ-ЗАСТОСУНКУ.....	42
3.2.1 Опис архітектурного шаблону .....	42
3.2.2 Розробка архітектури проекту.....	43
3.3 РОЗРОБКА ПРОГРАМНОЇ ЧАСТИНИ.....	45
3.4 МОДЕЛЬ ЗБЕРІГАННЯ ДАНИХ.....	52
3.5 РОЗРОБКА UML-ДІАГРАМ ФУНКЦІОНУВАННЯ СИСТЕМИ.....	54
Висновок до розділу 3 .....	56
РОЗДІЛ 4 ОГЛЯД РЕАЛІЗОВАНОЇ СИСТЕМИ .....	57
4.1 ОГЛЯД РЕАЛІЗОВАНИХ МОЖЛИВОСТЕЙ PWA.....	57
4.2 ОГЛЯД ОСНОВНИХ МОЖЛИВОСТЕЙ СИСТЕМИ.....	61
Висновок до розділу 4 .....	66
ВИСНОВОК .....	67
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ .....	68

					<b>ІАЛЦ.467200.003 ПЗ</b>		
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>			
<i>Розроб.</i>		<i>Лінійчук Д.В.</i>			<i>Літ.</i>	<i>Арк.</i>	<i>Акрушів</i>
<i>Перевір.</i>		<i>Павлов В.Г.</i>				1	68
<i>Н. Контр.</i>		<i>Виноградов Ю.М.</i>			<b>КПІ ім. Ігоря Сікорського</b> <b>ФІОТ ІВ-93</b>		
<i>Затверд.</i>		<i>Стіренко С.Г.</i>					
<b>Прогнозувальник погоди на основі технології PWA</b> <b>Зміст</b>							

## ВСТУП

За останні роки мобільні пристрої та веб-додатки зайняли центральне місце в нашому повсякденному житті. Користувачі все частіше звертаються до мобільних пристроїв для отримання інформації, комунікації та виконання різноманітних завдань. У цьому контексті Progressive Web Applications (PWA) виявляються потужним інструментом, який поєднує переваги веб-сайтів та мобільних додатків, забезпечуючи користувачам зручну та швидку взаємодію. У цьому дипломному проекті розглядається створення сервісу для прогнозування погоди на основі технології PWA. Прогнозування погоди є важливим аспектом нашого повсякденного життя, а доступ до точної та актуальної інформації про погоду є необхідним для планування наших дій та прийняття рішень.

Метою проекту є розробка прогнозувальника погоди на основі технології PWA, який надаватиме користувачам зручний спосіб отримання погодової інформації. Застосунок буде працювати на різних пристроях та браузерах, надаючи гнучкість та доступність. Використання технології PWA дозволить нам забезпечити інтерактивний інтерфейс, швидкий завантаження сторінок та можливість роботи в автономному режимі. Задля досягнення цієї мети сформовано декілька задач які будуть вирішені впродовж створення дипломного проекту:

- Огляд існуючих аналогів ресурсів подібної тематики та їх порівняння.
- Огляд сучасних технологій для розробки веб-сервісів та вибір найкращих.
- Розробка архітектури для нашої системи та програмної частини цієї системи.
- Огляд та тестування прогнозувальника погоди.

					<i>ІАЛЦ.467200.003 ПЗ</i>	Арк.
						2
Змн.	Арк.	№ докум.	Підпис	Дата		

Проект передбачає аналіз та використання зовнішніх сервісів, що надають погодні дані, та їх інтеграцію з веб-додатком. Окрім того, будуть реалізовані механізми кешування та оновлення даних, що дозволить покращити продуктивність та доступність додатку. Очікується, що результатами дипломного проекту буде повнофункціональний та ефективний PWA сервіс для прогнозування погоди.

					<i>ІАЛЦ.467200.003 ПЗ</i>	<i>Арк.</i>
						3
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

# РОЗДІЛ 1 АКТУАЛЬНІСТЬ ПРОЕКТУ ТА ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

## 1.1 Актуальність проекту

Одним з найпопулярніших видів сайтів якими найчастіше користуються люди всього світу це сайти прогнозування погоди. За даними статистики що надає британська компанія яка спеціалізується на маркетингових дослідженнях Cantar CMeter, станом на серпень 2021 року сайти з прогнозування погоди займають одинадцяте та сімнадцяте місце в рейтингу найпопулярніших в Україні[1], а станом на 2022 рік вже посідають десяте місце[2]. Що свідчить про зріст популярності сайтів даної тематики.

Така тенденція обумовлена тим, що все більше користувачів звикли перевіряти прогноз погоди, щоб заздалегіть планувати свої дії. Але разом з цим такі сервіси мають задовільняти декілька важливих умов:

- По-перше, вони мають надавати надійну інформацію, тому чим більш точною та розгорнутою вона буде тим більша кількість користувачів почне використовувати його на постійній основі;
- По-друге, інтерфейс такого ресурсу має бути максимально зручним та інтуїтивно зрозумілим. Потрібно надавати доступ до всіх функцій без зайвих кліків та складних навігаційних систем;
- По-третє, наочність також відіграє важливу роль. Тому подана інформація повинна бути не лише зрозумілою, а й привабливою для користувачів. Чим лаконічніше та простіше буде виглядати сайт тим краще буде для користувача;

Загалом, прогноз погоди - це не просто набір цифр і графіків. Це сервіс, який впливає на наші плани та розпорядок дня. Тому важливо, щоб сервіси прогнозу

					ІАЛЦ.467200.003 ПЗ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

погоди забезпечували достовірну інформацію, мали зручний інтерфейс та привабливий зовнішній вигляд, щоб користувачі могли легко та з задоволенням користуватися ними.

## 1.2 Огляд існуючих рішень

### 1.2.1 Огляд сайтів для прогнозування погоди

На сьогоднішній день існує велика кількість сервісів, які надають прогноз погоди. Ці сервіси відрізняються у способах реалізації та доступній інформації для користувачів. Наприклад, серед зарубіжних веб-ресурсів можна відзначити Weather.com - англomовний ресурс від оператора супутникового телебачення з США, Accuweather.com. Щодо українських веб-сайтів, популярні варіанти включають Sinoptik.ua, Gismeteo.ua та meteo.ua. Розглянемо лише найпопулярніші сервіси на території України.

Gismeteo.ua (Рисунок 1.1) – цей сайт працює з 2000го року, як ресурс який надає інформацію з прогнозу погоди та пропонує своє API за підпискою для інших сервісів.

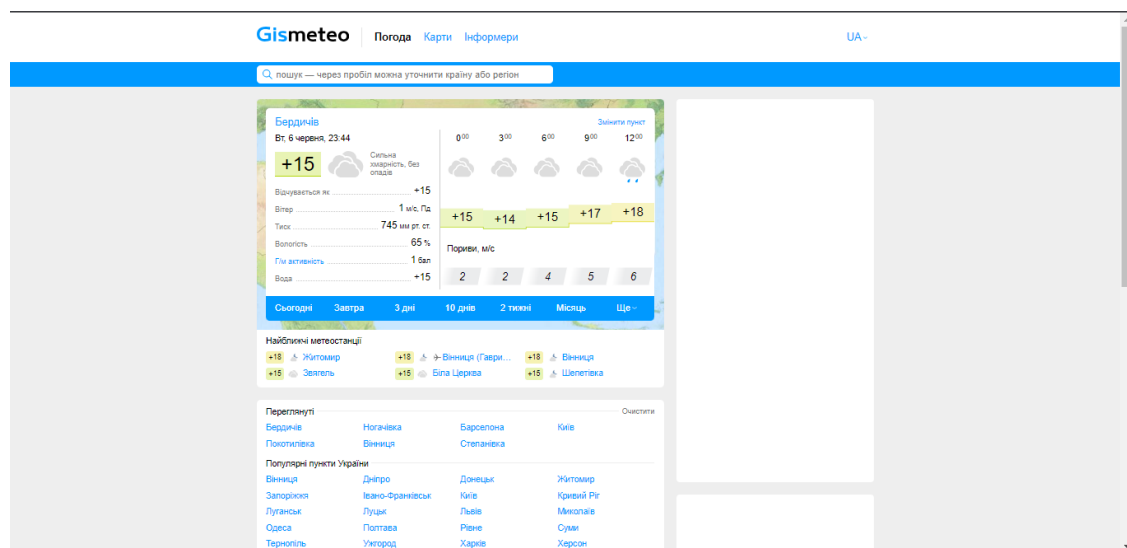


Рисунок 1.1 – сайт Gismeteo.ua

Щодо функціональності цього ресурсу то в користувача є такі можливості:

- Можливість роздрукувати прогноз погоди за допомогою принтера або залишити відгук щодо його точності;
- Перегляд метеорологічних мап(Рисунок 1.2) на який можна відслідкувати зміни в опадах, температурі, руху повітряних мас та хмарності в інтерактивному режимі з інтервалами змін в одну годину;
- Перегляд прогнозу погоди на 1, 3, 10 ,14 та 30 днів з можливістю перегляду змін температури та швидкості вітру в продовж дня. Також для таких прогнозів передбачена зміна параметрів для відслідковування таких як наявність пилку в повітрі, висота снігового покриву і т.д.;

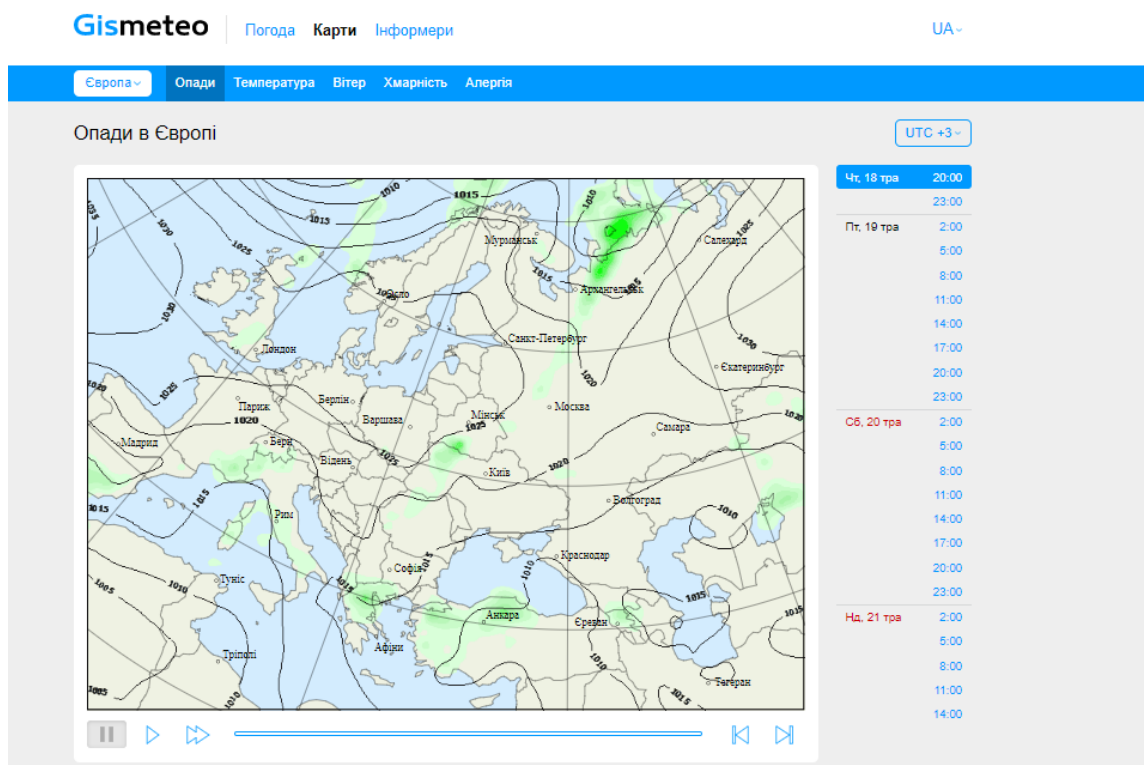


Рисунок 1.2 – метеорологічні мапи

Ретельно ознайомившись з цим ресурсом, можливо виділити як його переваги так і недоліки. Розглянемо список переваг які має цей ресурс:

									Арк.
									6
Змн.	Арк.	№ докум.	Підпис	Дата	ІАЛЦ.467200.003 ПЗ				

- Можливість гнучко налаштовувати прогноз погоди з вибором параметрів які потрібні користувачу;
- Велика кількість інтервалів для отримання прогнозів, аж до прогнозу на цілий місяць;
- Інтуїтивно зрозумілий інтерфейс, та можливість зміни мови на сайті;
- Наявність інтерактивних метеорологічних мап;

Щодо недоліків то можливо виділити такі пункти:

- Надлишкова кількість рекламних банерів на сайті, що частково ускладнює користування ресурсом, особливо при зміні населеного пункту та налаштування параметрів прогнозу погоди;
- Відсутність інтерактивної мапи з можливістю більш зручного вибору населеного пункту або місцевості для якої потрібно зробити прогноз;
- Складність використання інтерактивних метеорологічних мап з огляду на те що вибір місцевості здійснюється за часовим поясом та інколи перестає працювати після декількох змін;
- Відсутня можливість збереження користувацьких налаштувань, адже зміни зберігають в локальній пам'яті браузера, що захищає їх від під час закриття вкладки або вікна браузера але не захищає від випадкового видалення під час очищення пам'яті браузером або користувачем.

З огляду на переваги та недоліки цього ресурсу можна сказати, що цей сайт не націлений на постійне відвідування користувачем адже він має замало корисних функцій які могли б його зацікавити, а також має проблеми зі збереженням користувацьких налаштувань, бо для цього він використовує локальну пам'ять браузера яка ненадійна для постійного збереження даних.

Sinoptik.ua (Рисунок 1.3) – сайт належить українському порталу Ukr.net та працює з 2010 року. Цей ресурс не надає послуг з використання свого API, а зосереджений лише на прогнозуванні погоди.

					<i>ІАЛЦ.467200.003 ПЗ</i>	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		



- Відсутність користувацьких налаштувань. Сайт не надає можливості користувачу отримати більш детальний опис прогнозу погоди або прибрати непотрібні йому налаштування;
- Надмірна кількість рекламних банерів. За рахунок такої кількості реклами інтерфейс частково втрачає свою зрозумілість адже кількість зайвого контенту місцями заважає пошуку активних елементів на сторінці;
- Застарілий, непривабливий дизайн сайту;

Даний ресурс хоча й виконує свою функцію, а саме прогнозує погоду але кількість налаштувань та візуальний вигляд значно гірший ніж у попереднього. Окрім того період для прогнозувань може не влаштувати деяких користувачів, наприклад тих хто планує подорожі або якусь довготривалу діяльність. Візуальна складова сайту хоч і виглядає простою але надлишкова кількість реклами перенасичує головну сторінку, що ускладнює сприйняття інформації.

Meteo.ua (Рисунок 1.4) – сайт надає широкий спектр послуг з прогнозування погоди а також пропонує можливість розміщення своїх інформерів погоди на інших ресурсах.

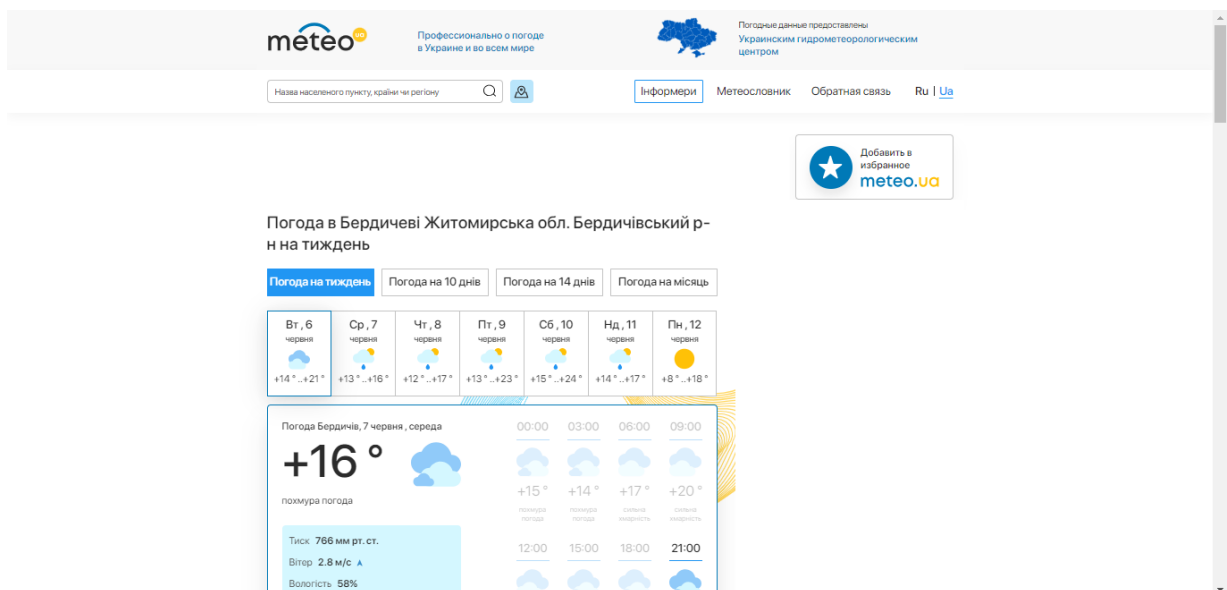


Рисунок 1.4 – сайт Meteo.ua

Даний ресурс надає більшу функціональність ніж попередні, окрім того прогнози погоди набувають певної інтерактивності під час перегляду. Розглянемо список функцій яких цей ресурс надає користувачу:

- Можливість прогнозування погоди з інтервалами в 7, 10, 14 та 30 днів. Зокрема прогнози на день мають можливість погодинного відстеження при якому показується зміна в температурі, тиску швидкості вітру та вологості;
- Можливість перегляду поточної погоди для всіх міст певної області. Для цього на головній сторінці виділений окремий блок з відображенням погодних умов без пошуку міста за назвою;
- Метеословник, який надає можливість користувачу розтлумачити певні терміни прогнозу погоди які він міг не зрозуміти;
- Інтерактивна мапа України на якій відображається поточна погода в обласних центрах;
- Можливість додавати стилізовані інформери прогнозів погоди на свої ресурси;
- Динамічні метеорологічні мапи з можливістю перегляду зміни певних погодних показників в реальному часі або з його плином по всьому світі;

Виходячи з функціональних можливостей даного сайту можна виділити як його сильні так і слабкі сторони. Розглянемо спочатку сильні сторони даного ресурсу :

- Велика кількість вибору інтервалів для отримання прогнозу погоди
- Наочність прогнозу погоди збільшена за рахунок зрозумілих зображень зі статусом погоди та за рахунок можливості інтерактивного погодинного перегляду прогнозу погоди для окремих днів;
- Можливість перегляду поточної погоди в різних містах області в окремому блоці сторінки без потреби пошуку міст за назвою;

					<i>ІАЛЦ.467200.003 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

- Наявність інтерактивних метеорологічних мап з можливістю відслідковування зміни погодних умов в різних регіонах, що може знадобитись деяким користувачам;
- Інтерфейс сайту інтуїтивно зрозумілий та не перешкоджає пошуку потрібної інформації;
- Приємний дизайн;
- Наявність додаткових можливостей таких як налаштування та формування інформерів які можуть бути використані на інших сайтах;
- Наявність історичних даних щодо погоди в певний день та їх рекордні показники;

Виходячи з функціональних можливостей даного сайту можна виділити як його сильні так і слабкі сторони. Розглянемо спочатку сильні сторони даного ресурсу :

- Велика кількість вибору інтервалів для отримання прогнозу погоди
- Наочність прогнозу погоди збільшена за рахунок зрозумілих зображень зі статусом погоди та за рахунок можливості інтерактивного погодинного перегляду прогнозу погоди для окремих днів;
- Можливість перегляду поточної погоди в різних містах області в окремому блоці сторінки без потреби пошуку міст за назвою;
- Наявність інтерактивних метеорологічних мап з можливістю відслідковування зміни погодних умов в різних регіонах, що може знадобитись деяким користувачам;
- Інтерфейс сайту інтуїтивно зрозумілий та не перешкоджає пошуку потрібної інформації;
- Наявність додаткових можливостей таких як налаштування та формування інформерів які можуть бути використані на інших сайтах;

					<i>ІАЛЦ.467200.003 ПЗ</i>	Арк.
						11
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

- Наявність історичних даних щодо погоди в певний день та їх рекордні показники;
- Приємний дизайн;

Розглянувши найпопулярніші сайти для прогнозування погоди в Україні можливо сформулювати спільні слабкі сторони цих ресурсів. Найголовнішою проблемою є надмірна кількість реклами яка некоректно розміщена на сторінках, що створює ефект перенасичення контентом і заважає сприйняттю корисних даних та пошуку активних елементів. Окрім того, ці сайти не підходять для регулярного використання адже немає можливості обирати які саме параметри будуть надаватись в прогнозі та зберігати їх налаштування для окремого користувача. Але є й позитивні сторони серед яких велика кількість періодів для прогнозування та наявність різного роду інтерактивних мап для відслідковування зміни різних погодних параметрів в зручному форматі.

### 1.2.1 Огляд сайтів для прогнозування погоди

Мобільні додатки для відслідковування погоди набули трохи більшої популярності аніж сайти. Це пов'язано з тим що розробники операційних систем для смартфонів заздалегідь додають застосунки для прогнозування погоди в свої розробки, що частково зменшило потреби в користуванні сайтами такої тематики. Але велика кількість людей через обмеженість функціоналу таких рішень користується сторонніми застосунками. Найпопулярнішими серед них є AccuWeather та OpenWeather кількість їх завантажень на платформах Android та IOS перевищує 100млн. Розглянемо слабкі та сильні сторони цих застосунків.

AccuWeather (Рисунок 1.5) – належить американській компанії AccuWeather яка функціонує з 1962 року та надає комерційні послуги по прогнозуванню погоди. Цей застосунок підтримується як на Android так і на IOS що надає йому більшої популярності.

					<i>ІАЛЦ.467200.003 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12



Рисунок 1.5 – застосунок AccuWeather

Даний застосунок має всього 4 активних вкладки в яких й розміщені всі його функції. Окрім того наявне меню в якому користувач має можливість змінювати інформацію свого профілю. Розглянемо функції які надає цей застосунок:

- Можливість перегляду прогнозу погоди на поточний день з деякими додатковими даними, такими як поточна якість повітря активність пилку для людей які страждають від алергії та історичні показники які фіксувались в цей день;
- Надання погодинного прогнозу погоди на чотири дні;
- Надання прогнозу погоди на 10 днів або на декілька місяців ;
- Інтерактивна мапа на якій в реальному часі або з прогнозом на 3 години виділяються погодні явища, такі як дощ сніг обледеніння в різних регіонах та інше;

					ІАЛЦ.467200.003 ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

Після дослідження всіх функцій які надає цей застосунок та покористувавшись ним певний час можливо сформулювати його слабкі та сильні сторони. Спочатку розглянемо переваги:

- Інтерфейс інтуїтивно зрозумілий, майже кожний його елемент надає додаткові дані після натискання. Зокрема дуже сучасний та приємний дизайн застосунку з можливістю змінювати основні кольори в меню налаштувань;
- Можливість змінювати одиниці виміру деяких величин таких як температура швидкість довжина та напрямок вітру . Ці налаштування зберігаються навіть після видалення та повторної інсталяції цього додатку.
- Можливість збереження назв міст для прогнозування погоди та відповідне налаштування сповіщень для зміни погодних умов в цих містах
- Прогнозування відбувається без пошуку назви міста а за допомогою відстеження по геолокації користувача
- Варіативність для прогнозування як і в кількості днів так і по годинно;

Але попри переваги цей застосунок має й декілька недоліків а саме:

- Неможливість перегляду останнього прогнозу погоди при відсутності інтернет з'єднання
- Деякі функції стають доступними лише після купівлі платної підписки, окрім того в безкоштовній версії наявна реклама як у вигляді банерів між корисною інформацією так і у вигляді відеороликів які програються без можливості їх уникнути
- Під час пошуку міста або зміни серед вже збережених застосунків перестав працювати на короткий час іноді можливі самотійні закриття.

					<i>ІАЛЦ.467200.003 ПЗ</i>	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

- Система сповіщень може припинити свою роботу під час того як користувач вимикає геолокацію.

Загалом додаток AccuWeather пропонує користувачу достатній спектр функцій для задоволення його потреб, однак для того щоб користуватись ним в повній мірі потрібно придбати підписку аби розблокувати деякі з них. Користуватися таким додатком зі смартфона набагато простіше ніж використовувати сайт. Але даний ресурс працює тільки на смартфонах і не підтримує інші платформи такі як Windows та Linux тому це не універсальне рішення.

OpenWeather (Рисунок 1.6) – даний застосунок також належить американській компанії яка займається наданням свого API для комерційних проектів. В своєму розпорядженні має велику кількість приватних метеорологічних станцій які забезпечують точність прогнозування погоди.



Рисунок 1.6 – застосунок OpenWeather

Загалом додаток має дуже просту структуру, він складається з основної сторінки додатку на якій розміщена вся інформація щодо прогнозу погоди на поточний день та меню налаштувань. Розглянемо основні функції які надає нам цей застосунок:

- Можливість перегляду прогнозу погоди на поточний день та в даний момент часу. З візуалізацією змін в погодніх умовах за допомогою графіку.
- Можливість отримати прогноз погоди на тиждень також з відповідною візуалізацією змін погодніх умов впродовж дня за допомогою графіку
- Налаштування одиниць вимірювання для показників прогнозу погоди в залежності від потрібних користувачу
- Можливість залишити відгук про некоректність прогнозу погоди та вказати актуальні дані

Тепер з огляду на функціональні можливості застосунку давайте розглянемо його переваги:

- Мінімалістичний дизайн, який сприяє швидкому пошуку потрібної інформації та забезпечує інтуїтивність інтерфейсу.
- Відсутність рекламних банерів
- Гнучка система налаштувань одиниць вимірювання для показників прогнозу погоди
- Можливість наочно спостерігати тенденцію змін погодніх умов впродовж дня за рахунок зрозумілих та простих графіків.

Попри переваги даного застосунку в ньому присутні й деякі недоліки обумовлені простотою даного застосунку :

- Недостатня кількість функцій
- Неможливість перегляду останнього прогнозу погоди при відсутності інтернет з'єднання
- Відсутність прогнозування погоди більш ніж на тиждень

					ІАЛЦ.467200.003 ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

- Відсутність можливості зберегти користувацькі налаштування в своєму профілі, що може знадобитись в разі зміни девайсу або видалення застосунку та повторного його встановлення

Розглянувши найпопулярніші мобільні застосунки для прогнозування погоди прослідковуються основні недоліки та переваги таких рішень. Почнемо з переваг, мобільні додатки для прогнозування погоди простіше використовувати через декілька причин: прогнозування відбувається не тільки через введення та пошук назви міст, а й за допомогою геолокації, що спрощує роботу користувачу, також наявні сповіщення які без участі користувача можуть звітувати про зміни погодніх умов. Але є й мінуси найголовнішим з яких те що застосунки такого роду зазвичай безкоштовні лише частково адже для розблокування іноді важливих функцій потрібно купувати підписку, також проблемою є й те що такі додатки потребують використання сторонніх програм для встановлення та займають чимало пам'яті пристрою.

### 1.3 Порівняння існуючих рішень

В попередньому розділі ми розглянули існуючі рішення для прогнозування погоди серед яких були найпопулярніші сайти та мобільні додатки. Порівняймо їх у таблиці 1.1. З огляду на описані переваги та недоліки рішень описаних в попередніх розділах сформуємо шкалу для оцінювання 1 до 5, де 5 – найкраще задовольняє обрану властивість, а 1 – найгірше задовольняє обрану властивість 3 – це середнє значення. Щодо критеріїв для оцінки опишемо умови для досягнення кращого результату:

- Незалежність від операційної системи та сторонніх програм – для досягнення максимальної оцінки ресурс повинен працювати на будь яких платформах та бути незалежним від сторонніх програм які б забезпечували доступ до нього.

					<i>ІАЛЦ.467200.003 ПЗ</i>	Арк.
						17
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

- Вибірковість прогнозування погоди – для досягнення максимальної оцінки ресурс повинен надавати користувачу можливість погодинного прогнозування для поточного дня, тижневий прогноз погоди та прогнозування на десять днів також він повинен надавати такий список показників погодних умов температура швидкість вітру тиск вологість та ймовірність опадів окрім того забезпечувати можливість додавання або вилучання їх з прогнозу погоди.
- Можливість додавання користувацьких налаштувань – для досягнення максимальної оцінки ресурс повинен надавати користувачу інструменти для налаштування одиниць вимірювання для прогнозування погоди та зберігати ці налаштування в довгостроковій формі
- Наочність прогнозу погоди та його показників – для досягнення максимальної оцінки ресурс повинен використовувати інтуїтивно зрозумілі зображення для показників погодних умов мати в наявності інтерактивну мапу та використовувати графіки для зручного відображення тенденцій змін погодних умов.
- Використання доступних функцій пристрою – для досягнення максимальної оцінки ресурс повинен використовувати геолокацію для автоматичного розпізнавання населеного пункту в якому знаходиться користувач та сповіщення для вчасного інформування про зміну погодних умов тощо.

					<i>ІАЛЦ.467200.003 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

Таблиця 1.1 – Порівняльна характеристика

Назва ресурсу → Властивості ↓	Gismeteo	Sinoptik	Meteo	AccuWeather	OpenWeather
Незалежність від операційної системи пристрою та сторонніх програм	4	4	4	1	1
Вибірковість прогнозування погоди	4	2	3	4	3
Можливість додавання користувацьких налаштувань	1	1	1	5	1
Наочність прогнозу погоди та його показників	3	1	3	5	3
Використання доступних функцій пристрою	1	1	1	5	5

Як можна побачити із порівняльної таблиці ні один з ресурсів не задовольняє всі критерії в повній мірі та не є універсальним рішенням.

Розпочнемо з сайтів для прогнозування погоди. Хоча таке рішення і є незалежним від операційної системи але залежить від браузера адже без нього неможливо отримати доступ до цього ресурсу. Залежність від

браузера впливає й на зручність використання адже користувачу потрібно використовувати сторонню програму та пам'ятати назву чи адресу ресурсу, щоб ним користуватись. Окрім того в такого рішення є проблеми з користувацькими налаштуваннями, оскільки вони не використовують систему авторизації користувача тому й дані про користувацькі налаштування зберігаються лише в браузері пам'ять якого постійно оновлюється та очищується. До того ж сайт не завжди може використовувати внутрішні функції пристрою такі як геолокація, надсилання сповіщень, це залежить від типу та налаштувань браузера.

Щодо мобільних додатків то їх основна проблема полягає в тому що вони працюють лише на тих операційних системах для яких їх створювали, окрім того для їх встановлення користувач повинен використовувати та реєструватись на сторонніх сервісах. Хоча це сприяє на зручність використання такого додатку адже для його використання нічого не потрібно окрім його встановлення також через те що такий ресурс має доступ до внутрішніх можливостей пристрою, він без проблем може використовувати геолокацію та інші можливості. Також збільшується можливість для додавання та зберігання користувацьких налаштувань. Але виникає проблема в тому що таких додаток в різних випадках може займати чимало пам'яті пристрою на відміну від сайтів.

					<i>ІАЛЦ.467200.003 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

## Висновок до розділу 1

В попередніх розділах ми розглянули ресурси для прогнозування погоди на прикладах сайтів та мобільних додатків.

Щодо сайтів то це універсальне рішення яке працює в незалежності від платформи, нескладне в розробці та не займає пам'яті пристрою але обмежене в можливості покращення використання для користувача, тобто додавання користувацьких налаштувань та у використанні внутрішніх функцій пристрою.

Натомість мобільні застосунки з прогнозування погоди хоч й працюють лише на платформах для яких їх створювали, займають чимало пам'яті пристрою та складніші в розробці але все ж простіші у використанні користувачем та мають більший простір можливостей для додавання та зберігання користувацьких налаштувань.

Виходячи з цього, для того щоб створити кращий сервіс для прогнозування погоди потрібно зберегти переваги які надають сайти та мобільні додатки такої ж тематики та прибрати їх недоліки а саме:

- Надати можливість перегляду останнього прогноз погоди без доступу в інтернет;
- Зробити сервіс незалежним від операційної системи та сторонніх програм;
- Надати йому можливість використовувати внутрішні функції пристрою такі як геолокація і тд;
- Забезпечити створення та зберігання користувацьких налаштувань;

					ІАЛЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		21

# РОЗДІЛ 2 ТЕХНОЛОГІЇ РОЗРОБКИ КЛІЄНТ-СЕРВЕРНОЇ СИСТЕМИ

## 2.1 Огляд технології PWA

PWA(Progressive Web Application) – це додаток, створений за допомогою веб-технологій і є гібридом звичайного веб-сайту, доступ до якого здійснюється через браузер (в тому числі і браузер мобільного додатку). Такі гібридні веб-додатки імітують досвід використання нативних додатків і мають максимально наближений до них зовнішній вигляд і особливості використання [3]. Ще однією особливістю цієї технології є те що вона розв’язує проблему обмеженого місця на смартфоні адже розмір веб-дodatка не перевищує 1-3 мб а переважну більшість даних PWA зберігає в хмарному сховищі. Окрім того такий додаток зберігає автономність, тобто доступ може отримуватись як з доступом до інтернету так і без нього.

За допомогою цієї технології майже будь який сайт можна перетворити в веб-додаток при виконанні деяких умов але потрібно звертати увагу на доцільність такого переходу. Найпопулярнішими варіантами сайтів які створюються на основі технології PWA є інтернет магазини, інформаційні ресурси та сайти візитки. Хоча навіть великі інтернет ресурси такі як Pinterest, Tinder та Twitter використовують технологію PWA в своїх розробках.

### 2.1.1 Переваги та недоліки PWA

Звичайно, окрім PWA, існують різні технології та підходи для створення веб-додатків які б працювали як нативні застосунки на мобільних пристроях і не тільки. Для того щоб порівняти їх потрібно сформулювати ключові переваги та недоліки технології PWA.

					ІАЛЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		22

## Розглянемо переваги Progressive Web App:

- Користування без браузера. Функція надає клієнту миттєвий доступ до інформації, щоб користувач мав можливість в будь-який час відвідати платформу;
- Кросплатформність. В той час коли стандартний мобільних застосунок працює тільки на платформах для яких його розробляли PWA може працювати майже на всіх апаратних платформах що знижує вартість розробки такого застосунку тому що не потрібно створювати різні версії додатку для різних платформ;
- Автономний режим. Без підключення до Інтернету у клієнта залишається можливість переглянути інформацію та надіслати HTML-форму. При з'єднанні з Інтернетом, дані з заповненої форми, відправляються на сервер завдяки зберіганню на пристрою. Наприклад, якщо у вас пропаде інтернет при потраплянні в тунель, на звичайному веб-сайті, дані пропадуть після натискання "Надіслати", в той час як PWA завдяки зберіганню самостійно відправляє за першої можливості [4];
- Поліпшення конверсії. Під час роботи з PWA збільшується можливість повторного відвідування сайту а також збільшується час який користувач проводить на ресурсі;
- Спрощена процедура внесення оновлень. Користувачу не потрібно використовувати магазин програм для оновлення такого додатку;
- Непрямий вплив на SEO. PWA працює тільки із захищеним протоколом HTTPS, що позитивно впливає на ранжуванні сторінок. Також прогресивні програми мають кращі поведінкові фактори ніж звичайні, оскільки клієнти активніше заходять на сторінку, тому менше відмов для сайту, що також вплине на ранжування [4];

					<i>ІАЛЦ.467200.003 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

- Підвищена продуктивність. Швидкість роботи PWA підвищується завдяки тому, що він може ефективно функціонувати, працювати як веб-сайт, кешувати та обслуговувати текст, зображення та інший контент. Цей тип програмного забезпечення покращує не лише швидкість завантаження сторінок, але й користувацький досвід.

Не зважаючи на переваги, PWA технологія також має недоліки, наприклад:

- Відсутність максимальної універсальності. Хоча майже всі сучасні браузері підтримують цю технологію в застарілих версіях ця підтримка може бути відсутньою;
- Можливість довгого встановлення. Це обумовлено тим що при першому відвідуванні сайту відсутня інформація про ресурс в кеші браузеру. Тому при повільному з'єднанні з інтернетом встановлення може зайняти певний час.
- Можливість втрати даних при видаленні кешу. При роботі з сайтом через браузер при видаленні кешу безповоротно будуть втрачені данні потрібні для роботи з застосунком без інтернету. Проблема буде наявна допоки не з'явиться інтернет і дані не перезапишуться в кеш;
- Складність впровадження в готовий сайт. Незважаючи на легкість написання PWA впровадити його у вже готовий сайт може бути не просто оскільки сайт для цього потрібно максимально оптимізувати щоб не перевантажувати кеш браузера;
- Обмеження працездатності під час роботи в режимі "інкогніто". Під час роботи в браузері в режимі "інкогніто" обмежується використання файлів cookie, кешу та різних можливостей пристрою(геолокація, мікрофон тощо) від який залежить коректна робота деяких функцій PWA;

					<i>ІАЛЦ.467200.003 ПЗ</i>	Арк.
						24
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

- Складність впровадження в готовий сайт. Незважаючи на легкість написання PWA впровадити його у вже готовий сайт може бути не просто оскільки сайт для цього потрібно максимально оптимізувати щоб не перевантажувати кеш браузера;
- Обмеження працездатності під час роботи в режимі “інкогніто”. Під час роботи в браузері в режимі “інкогніто” обмежується використання файлів cookie, кешу та різних можливостей пристрою(геолокація, мікрофон тощо) від який залежить коректна робота деяких функцій PWA;

З огляду на переваги й недоліку цієї технології можна сказати що PWA можливо застосувати практично для всіх веб-сайтів, але рішення про перехід або застосування цієї технології при розробці сайту має ґрунтуватися на його розмірі, призначенні та можливості оптимізації використовуваних файлів та скриптів.

Однак, PWA призначені не лише для невеликих компаній і стартапів із обмеженими фінансовими ресурсами. Навпаки, багато глобальних корпорацій бачать величезні результати від переходу на стратегію PWA на користь своїх користувачів. Одним з основних брендів, які привернули багато уваги, є Twitter, який розробив свій додаток Twitter Lite з урахуванням PWA. Після запуску Twitter Lite кількість твітів у Twitter збільшилася на 75%[6], кількість сторінок за сеанс зросла на 65%, а показник відмов зменшився на 20%. Також слід зазначити що якщо сайт не відповідає умовам для переходу на цю технологію це не означає що це неможливо але на реалізацію потрібно витратити значно більше часу та ресурсів ніж з невеликими проектами.

					<i>ІАЛЦ.467200.003 ПЗ</i>	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

## 2.1.2 Складові компоненти PWA

Кожен PWA застосунок має містити наступні компоненти: маніфест веб-додатку та Service Worker.

Розпочнемо з Service Worker це технічний елемент, який підтримує одну з головних особливостей прогресивних веб-додатків - автономний режим роботи, фонову синхронізацію та push-сповіщення, характерні для мобільних додатків. Service Worker - це JavaScript-файл, що запускається окремо від веб-сторінки/додатку. Він реагує на взаємодію користувача з додатком, у тому числі на мережеві запити, що надходять зі сторінок, які він обслуговує[4].

Оскільки цей файл починає працювати лише після обробки певної події його існування коротке. Окрім того цей файл дозволяє кешувати оболонку додатку, тому при повторних відвідинах він завантажується миттєво. Необхідний динамічний контент оновлюється щоразу, коли з'єднання відновлюється. Ця механіка дозволяє забезпечити продуктивність додатку та покращити користувацький досвід. Та на останок Service Worker відповідає за надсилання так званих Push-сповіщень які можуть надсилатись під час того як браузер чи застосунок були закриті та за фонову синхронізацію. Він затримує дії до відновлення стабільного з'єднання. Наприклад, повідомлення, надіслане безпосередньо перед розривом з'єднання, буде позначено як відправлене, а потім буде доставлене, щойно з'єднання відновиться. Таким чином, сервери можуть надсилати періодичні оновлення додатку, що дозволить йому оновитися, коли з'єднання буде відновлено.

Маніфест веб-додатку – це файл у форматі json який відповідає за надання PWA вигляду нативного додатку. За допомогою маніфесту розробник може контролювати спосіб відображення програми користувачеві (тобто повноекранний режим без видимого рядка URL-

					<i>ІАЛЦ.467200.003 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

адреси) і спосіб її запуску. Файл дозволяє розробнику знайти централізоване місце для метаданих

веб-програми. Маніфест зазвичай містить початкову URL-адресу, повну та коротку назву програми, посилання на іконки, а також розміри, тип і розташування іконок.

Окрім того для PWA застосунок повинен передавати всі свої ресурси по HTTPS протоколу тому дуже важливо щоб в додатку не розміщувались посилання на незахищені ресурси навіть якщо це посилання на зображення тому їх треба зберігати в себе на сервері.

## 2.2 Розробка сервісної частини на платформі Firebase

Firebase — це платформа розробки мобільних та веб-додатків. Firebase надає базу даних та бекенд у режимі реального часу як послугу. Сервіс надає розробникам додатків API для синхронізації даних програми між клієнтами та зберігання їх у хмарі Firebase. Компанія також пропонує клієнтські бібліотеки, що дозволяють інтегруватися з програмами Android, iOS, JavaScript/Node.js, Java, Objective-C та Swift. База даних також доступна через API REST і може бути пов'язана з різними фреймворками JavaScript, такими як AngularJS, React і Ember.js[6].

Основні можливості Firebase:

- Робота в режимі реального часу (Realtime) передбачає використання хмарної бази даних, де дані зберігаються у форматі JSON і синхронізуються в режимі реального часу з кожним підключеним клієнтом. При створенні кросплатформових додатків для iOS, Android та використання JavaScript SDK всі клієнти використовують одну базу даних Realtime, отримуючи автоматичні оновлення з найсвіжішими даними.
- Режим автономної роботи (Offline) також підтримується в додатках Firebase, оскільки SDK Firebase Realtime Database кешує поточні

					<i>ІАЛЦ.467200.003 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		27

дані на пристрої. Після відновлення підключення до Інтернету, клієнтський пристрій отримує пропущені зміни та синхронізує їх з поточним станом сервера.

- Доступ до Firebase Realtime Database можна отримати безпосередньо з мобільних пристроїв або веб-браузерів, не потрібно писати серверний код для додатків. Забезпечення безпеки та перевірки даних здійснюється за допомогою правил безпеки Firebase Realtime Database, які базуються на гнучкій мові виразів. Ці правила виконуються під час читання або запису даних.
- Firebase Cloud Messaging (FCM) – це універсальний механізм для обміну повідомленнями, який дозволяє відправляти повідомлення на пристрої. Використовуючи FCM, ви можете сповіщати клієнтські програми про наявність нових електронних або інших даних для синхронізації. Ви також можете надсилати повідомлення, щоб залучити та утримати увагу користувача. Для випадків, коли потрібен обмін миттєвими повідомленнями, повідомлення може містити корисне навантаження до 4 КБ, яке передається до клієнтської програми.
- Firebase Storage – це хмарне сховище, розроблене для додатків. Cloud Storage for Firebase є потужним та легким у використанні сервісом для зберігання медіа-даних. В рамках безкоштовного тарифного плану надається до 10 ГБ безкоштовного простору в хмарі. За допомогою SDK розробник може зчитувати та зберігати аудіофайли, зображення, відео або будь-який інший формат даних, які використовуються для відображення контенту. Завантажені файли можна переглянути в Firebase Console.
- Аутентифікація Firebase – це важливий етап для багатьох додатків, який дозволяє ідентифікувати користувача. Цей процес дозволяє додатку безпечно зберігати та структурувати дані, які призначені для конкретного користувача в хмарному середовищі. Крім того,

					<i>ІАЛЦ.467200.003 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		28

завдяки аутентифікації дані можуть бути синхронізовані на різних пристроях з різними операційними системами, що використовує користувач. Ідентифікація також дозволяє відображати релевантні дані, спеціально підібрані для конкретного користувача.

Головними перевагами використання технології Firebase є:

- Швидкість роботи. Набір засобів розробки Firebase включає інтуїтивно зрозумілі API, які дозволяють швидко та легко створювати високоякісні програми. Крім того, у вас є всі необхідні інструменти для розширення бази користувачів та збільшення доходів: просто оберіть ті, які відповідають вашим цілям;
- Готова інфраструктура. Не потрібно створювати складну інфраструктуру або працювати з декількома панелями управління. Натомість ви можете зосередитися на потребах своїх користувачів;
- Статистика. В основі Firebase є безкоштовний аналітичний інструмент, розроблений спеціально для мобільних пристроїв. Google Analytics для Firebase дозволяє отримувати дані про активність користувачів та вживати негайних заходів за допомогою додаткових функцій;
- Кросплатформність. Firebase працює на будь-якій платформі з пакетами розробки для Android, iOS, JavaScript та C++. Доступ до Firebase можна також отримати через серверні бібліотеки або REST API;
- Масштабованість. Якщо програма стає популярною і навантаження збільшується, немає необхідності змінювати код сервера або використовувати додаткові ресурси: Firebase зробить це за вас. Більше того, більшість функцій Firebase безкоштовні і залишаються такими, незалежно від розміру проекту. Доступні чотири платні функції. Вони включають безкоштовний пробний період та два тарифні плани;

					<i>ІАЛЦ.467200.003 ПЗ</i>	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

Оцінивши можливості даної платформи ми будемо використовувати її можливості як бази даних та її системою аутентифікації.

## 2.3 Огляд технологій для створення серверної частини

### 2.3.1 Вибір мови програмування

Для розробки клієнтської частини програми було обрано мову програмування TypeScript. Це мова програмування створена компанією Microsoft та позиціонує себе як можливість розширити функціональність JavaScript. Важливо зазначити що сучасні браузерери не мають змоги розпізнавати код написаний на цій мові тому він компілюється в JavaScript та оптимізується. Це дозволяє взаємодіяти з користувачем на стороні клієнта, керувати браузером, здійснювати асинхронний обмін даними із сервером та змінювати структуру та зовнішній вигляд веб-сторінки.

Щодо синтаксису то будь який код написаний на JavaScript буде оброблюватись компілятором як валідний але TypeScript має й додаткові можливості серед яких:

- анотація типів та їх перевірка під час компіляції що дає можливість розробнику власноруч зазначити типи даних для змінних та функцій але й є можливість довірити цю роботу компілятору для автоматичного визначення типів
- підтримка класів і успадкування TypeScript надає більш повноцінну підтримку класів і успадкування, ніж JavaScript. Ви можете визначати класи з явними типами полів і методів, а також використовувати модифікатори доступу, абстрактні класи, інтерфейси та інші концепції об'єктно-орієнтованого програмування.
- Підтримка модулів та просторів імен що дає змогу створювати ізольовані та краще організовані компоненти для нашого проекту

					ІАЛЦ.467200.003 ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

- Наявність великої кількості абстракцій для спрощення типізації коду

Тепер розглянемо основні типи та конструкції які надає нам ця мова. До примітивних типів належать: any(будь-який тип даних), number(64-бітні числа подвійної точності з плаваючою комою), string(являє собою послідовність символів у кодуванні Unicode UTF-16), boolean(логічний тип представлений значеннями true та false), symbol(унікальні ідентифікатори) та bigint(числа що виходять за межі типу number) окрім цих загальноприйнятих типів до розряду примітивних також можна віднести null, undefined, void, never, unknown однак ці типи більше використовуються для пояснення семантики коду а не для роботи з ними. Окрім примітивних типів є й конструкції які дають змогу типізувати класи об'єкти та масиви. До них можна віднести:

- enum – це конструкція яка являє собою набір логічно пов'язаних констант, як значення яких можуть виступати як числа, так і рядки
- union - механізм, що дає змогу створювати з безлічі наявних типів логічну умову, за якою дані можуть належати тільки до одного із зазначених типів
- interface - це синтаксична конструкція, призначена для опису відкритої частини об'єкта без реалізації клас або об'єкт, що реалізує інтерфейс, зобов'язаний реалізувати всі описані в ньому члени. Тому інтерфейс є гарантією наявності описаних у ньому характеристик у об'єкта, що його реалізує.
- types alias – або псевдонім типу дає змогу дати будь якому значенню свій тип.
- tuple – кортеж дають змогу зазначити розмір масиву та типи даних які будуть в ньому зберігатись
- abstract Class – абстрактний клас як й interface дає змогу описати структуру класу який буде його наслідувати.

Найголовніша причина використання цієї мови полягає в тому що вона не обмежує можливості JavaScript а доповнює їх що дає змогу прискорити розробку та за рахунок статичної типізації зменшити кількість багів та помилок. Окрім того переважна більшість фреймворків та бібліотек що дають змогу створювати оптимізований швидкодіючий та багатофункціональний сайт використовуючи лише можливості JavaScript чудово адаптовані для роботи й з TypeScript. Отже, вона забезпечує простоту в освоєні та стабільність роботи додатка, а також надзвичайну швидкість розробки без урізання функціоналу.

### 2.3.2 Бібліотека React js

React.js — це декларативна, ефективна і гнучка бібліотека, призначена для створення інтерфейсів користувача. Вона дозволяє компонувати складні інтерфейси з невеликих окремих частин коду — “компонентів”[8].

React підтримує віртуальний DOM (Document Object Model), а не покладається виключно на DOM браузера. Таким чином розробник працює зі сторінкою, вважаючи що вона оновлюється вся, але бібліотека самостійно вирішує які компоненти сторінки потрібно оновити. Це працює за рахунок паттерну “наглядача” який використовує React. При використанні цієї бібліотеки кожний елемент користувацького інтерфейсу який може змінюватись зберігає свій стан й при його зміні React оновлює віртуальний DOM після цього він порівнює його нову версію з попередньою після того як знаходить який елемент зазнав змін він оновлює його в реальному DOM сторінки що значно підвищує продуктивність веб-застосунку.

					<i>ІАЛЦ.467200.003 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		32





- `componentDidUpdate` - це метод життєвого циклу, який викликається одразу після оновлення. Цей метод не викликається під час першого рендерингу. Він використовується для роботи з DOM під час оновлення компонента. Це також зручний метод для виконання веб-запитів при порівнянні поточних та попередніх пропсів (наприклад, веб-запит може бути не потрібен, якщо пропси не змінились).
- `componentWillUnmount` – це метод життєвого циклу, який викликається безпосередньо перед розмонтуванням та знищенням компонента. Тут можна зробити будь-яку очистку, наприклад, скасувати таймери, мережеві запити або підписки створені у `componentDidMount()`;

Важливо пам'ятати що ці методи можливо використовувати лише в компонентах у вигляді класів.

На даний момент такі компоненти вважаються застарілими але все одно використовуються в деяких випадках. Актуальним видом компонентів є функціональні компоненти. Вони створюються як звичайні функції які як аргумент приймають об'єкт `props` і завжди повинні використовувати ключове слово `return` в якому описується верстка даного компонента.

Керування поведінкою функціональних компонентів здійснюється за допомогою так званих хуків – це спеціалізовані функції які дають змогу керувати змінами та поведінкою компоненту. Ми не будемо розглядати їх всі а лише основні серед яких :

- `useState` – це хук який повертає значення початкового стану компонента та функцію для його зміни. Саме за допомогою цієї функції ми маємо змогу керувати станами в функціональному компоненті.
- `useEffect` – єдиний хук який заміняє всі методи життєвого циклу. Цей хук приймає масив значень після зміни яких він буде викликатись та виконувати прописані нами дії. В одному компоненті таких хуків може бути використано декілька

- useRef – дає можливість помітити елемент верстки що використовується в компоненті для подальшого звернення до нього та зміни.

Ми розглянули базові поняття та можливості цієї бібліотеки які дадуть нам змогу створити та керувати клієнтською частиною нашого проекту.

### 2.3.3 Бібліотека Redux

Для кращого керування клієнтською частиною проекту було обрано бібліотеку Redux. Ця бібліотека надає підхід для зберігання та оновлення стану додатка таким чином, щоб цей процес був прогнозованим та детермінованим. Її використання в даному проекті обумовлене тим що при використанні стандартних методів для збереження станів компонентів які надає бібліотека React виникає проблема з масштабованістю адже внутрішні стани компоненті починають зберігати досить велику кількість даних які згодом за допомогою props передаються до інших компонентів як вверх так і вниз по ієрархії що ускладнює логіку обміну даними.

Натомість бібліотека Redux вирішує цю проблему адже стан додатку зберігається в централізованому об'єкті “Store”. Розглянемо основні конструкції що надає нам ця бібліотека:

- Store - це централізоване сховище що містить в собі стан додатку й напряду доступне лише для зчитування інформації.
- Actions - об'єкти які описують зміни стану. Вони містять в собі тип дії та додаткові дані які необхідні для зміни стану
- Reducers – функції які регулюють зміни стану відповідно до отриманих об'єктів Action. Вони отримують попередній стан застосунку дію яка характеризує що саме повинно змінитись та повертають новий стан
- Dispatch – метод який ініціалізує оновлення стану приймає в себе Action та передає його для обробки

					<i>ІАЛЦ.467200.003 ПЗ</i>	Арк.
						36
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Розглянемо базовий алгоритм роботи цієї бібліотеки. Головним компонентом Redux є store це об'єкт який приймає початковий стан застосунку та надає доступ до нього тільки для читання змінити стан застосунку може лише спеціальна функція reducer в яку за допомогою методу dispatch передається спеціальний об'єкт action в якому зберігається тип операції та якісь корисні дані. Потік даних завжди односпрямований. Отже при взаємодії з користувацьким інтерфейсом якщо потрібно оновити стан застосунку чи окремого компонента викликається метод dispatch який передає спеціальний об'єкт в якому міститься інформація про тип дії що потрібно виконати та може містити корисні дані до функції reducer яка приймає стан компоненту та дію і в результаті повертає змінений стан в наш store.

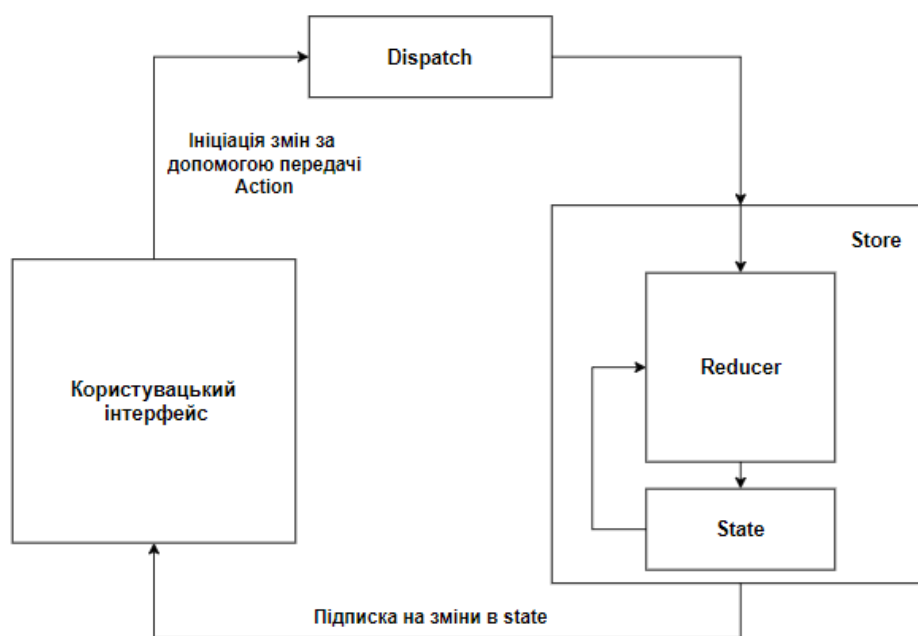


Рисунок 2.3 – принцип роботи Redux

В нашому проекті використовується дві допоміжні бібліотеки які допомагають працювати з redux а саме redux-toolkit та redux-persist. Перша з них надає певні кострукції для спрощення маніпуляції зі store та асинхронними операціями такими як запити до серверу.

Також вона надає можливість розділяти логіку роботи з певними частинами нашого стану застосунку що забезпечує модульність коду та незалежність певних частин користувацького інтерфесу. Друга бібліотека надає можливість зберігати дані необхідні для роботи застосунку в так званій localStorage(внутрішня пам'ять браузера дані в якому зберігаються безстроково) що вкрай необхідно для роботи PWA без підключення до інтернету. Окрім того ця бібліотека запобігає втрату даних під час оновлення сторінки або закриття вікна браузера.

### 2.3.4 Препроцесор SCSS

SCSS препроцесор - це надбудова над CSS, яка надає нові функції і можливості для CSS, за допомогою доданих синтаксичних конструкцій. Мета препроцесорів полягає в наданні зручних синтаксичних конструкцій для веб-розробника, щоб спростити і прискорити розробку і підтримку стилів в проектах. CSS препроцесори роблять код чистим і дійсним.[9] Загалом різниця між CSS та SCSS полягає лише в деяких змінах синтаксису всі правила що прописуються в SCSS файлі компілюються в звичайний файл тобто він використовується лише для розробки проекту та не приймає участі в роботі самого проекту. Розглянемо відмінності SCSS від CSS:

- Наявність вкладеності – SCSS дозволяє вкладати правила одне в одного що зменшує їх кількість та полегшує наслідування
- Наявність змінних та міксинів – за допомогою цього окремі групи правил які постійно повторюються можливо записати в міксин та імпортувати в різних ділянках коду а також можна зберегти окремі значення такі як відступи та кольори щоб використовувати їх як змінну
- Наявність функцій, математичних операцій та циклів

Це всі зміни що відрізняють препроцесор SCSS від мови стилів CSS.

					ІАЛЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		38

## Висновок до розділу 2

Firebase - це потужний та простий у використанні інструмент. Він полегшує життя розробників, дозволяючи їм зосередитись на логіці програми, а не на складній обробці даних. Firebase дає змогу легко керувати даними з декількома рядками коду. Також є можливість оновлення даних в автономному режимі, і зміни будуть автоматично синхронізовані, як тільки користувач повернеться до мережі. Оскільки дані зберігаються у форматі JSON, їх легко обробляти та змінювати.

Typescript є неперевершеним засобом, що поєднується безпосередньо з HTML та CSS і має широку підтримку усіма відомими браузерами. Це робить його незамінним для розробки веб-інтерфейсів. За рахунок того що це модифікована версія мови JavaScript вона використовує всі найкращі аспекти даної мови та надає додаткові переваги серед яких наявність статичної типізації та можливість опису користувацьких типів що дає змогу дізнатись про помилки виконання програми на етапі компіляції а не роботи проекту. Та значна оптимізація коду за рахунок компіляції типізованого коду в JavaScript.

Дана мова пропонує розширені можливості для взаємодії з користувачем, анімації, маніпулювання DOM-елементами та валідації форм. Завдяки широкій базі знань та розширеним можливостям, TypeScript дозволяє розробникам реалізувати найскладніші ідеї та створити інтерактивні, високоефективні веб-додатки.

Однією з ключових переваг використання React JS є можливість повторного використання компонентів. Це дозволяє розробникам економити час, оскільки вони можуть використовувати готові компоненти замість написання коду з нуля для кожного елемента. React JS був спеціально створений для досягнення високої продуктивності.

					<i>ІАЛЦ.467200.003 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		39

Його ядро використовує віртуальний DOM і можливість рендерингу на стороні сервера, що дозволяє швидко виконувати складні програми. React дозволяє розробникам використовувати компоненти як на стороні клієнта, так і на стороні сервера, що розширює можливості розробки та покращує продуктивність додатків.

					<i>ІАЛЦ.467200.003 ПЗ</i>	<i>Арк.</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		40

## РОЗДІЛ 3 Проектування та розробка системи

### 3.1 Опис вимог для функціоналу додатку

Увесь функціонал додатку надається навіть неавторизованому користувачу реєстрація та авторизація надає можливість для збереження даних користувача в хмарному середовищі для можливості користуватись додатком на різних пристроях та отримувати push-сповіщення від застосунку.

Так як наш веб-додаток розроблюється на основі технології PWA він повинен реалізовувати такі функції:

- Встановлення додатку на початковий екран будь якого пристрою для роботи без безпосереднього використання браузера користувачем.
- Кешування даних для забезпечення можливості перегляду останніх прогнозів погоди за відсутності інтернет з'єднання.
- Мати в наявності адаптовану верстку для забезпечення коректної роботи та відображення на всіх видах моніторів.
- Оновлення даних без допомоги користувача.

Що стосується функцій які не стосуються технології PWA повинні бути реалізованими такі можливості:

- Перегляду поточного прогнозу погоди на головній сторінці застосунку з використанням геолокації
- Погодинне прогнозування погоди на теперішній день з постійним оновленням згідно поточного часу
- Наявність меню з спеціалізованими показниками такими як ультрафіолетовий індекс відчутна температура та інше.
- Перегляд тижневого прогнозу погоди

					ІАЛЦ.467200.003 ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

- Наявність пошукової стрічки за автодоповненням тексту та різними функціями відповідно до активної сторінки застосунку
- Створення списку відстежуваних міст за допомогою пошукової стрічки з відображенням поточної погоди та погодинним і триденним прогнозом погоди
- Наявність мапи з можливістю додавання інтерактивних маркерів з відображенням поточних погодних умов та додавання цих міст до списку відстежуваних міст
- Перегляд статистики з відображенням на графіку змін мінімальних та максимальних температур за останніх три місяці, кількісної характеристики сумарних опадів та годин з дощем на кругових діаграмах а також статистика погодних умов.
- Вибір одиниць вимірювання для погодних показників для температури, швидкості вітру, тиску та опадів
- Виклик стандартного меню дозволів для геолокації та сповіщень за допомогою перемикачів на сторінці.

## 3.2 Розробка архітектури веб-застосунку

### 3.2.1 Опис архітектурного шаблону

Даний застосунок будується за архітектурним шаблоном MVC(Model-view-controller). Модель(model) компонент який представляє собою дані нашого застосунку а саме отримання обробку зміну та збереження даних застосунку та ніяк не залежить від інших компоненті та не зберігає дані про те як ці дані будуть відображатись цим компонентом є Redux. Представлення(view) відповідає за відображення даних моделі користувачу цей компонент надає користувацький інтерфейс та забезпечує отримання

					<i>ІАЛЦ.467200.003 ПЗ</i>	Арк.
						42
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

користувачем даних та їх відображення при цьому не змінює їх таким компонентом в нашому випадку виступає React. Останнім компонентом в цьому архітектурному шаблоні є контроллер(controller) він відповідає за взаємодію між моделлю та представленням так як ці компоненти повністю відокремлені один від одного контроллер оброблює користувацькі дії та координує взаємодію між цими компонентами він отримує вхідні дані від представлення інтерпритує їх та передає моделі і навпаки в нашому випадку за це відповідає бібліотека React-redux

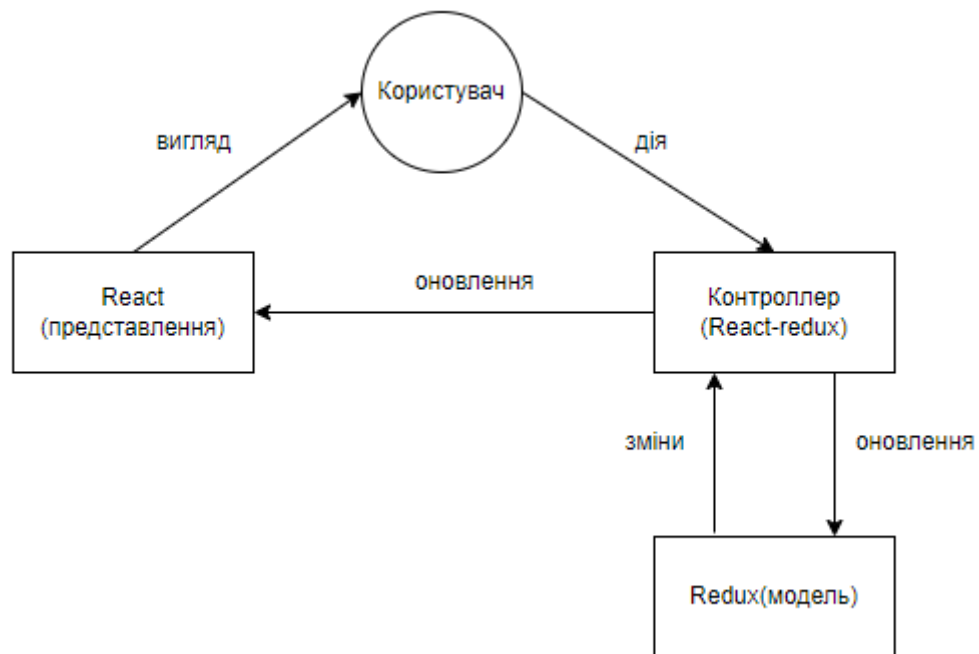


Рисунок 3.1 – Архітектура MVC з Redux та React

### 3.2.2 Розробка архітектури проекту

Даний проект складається з вкладених в App.tsx сторінок в які вкладаються компоненти що відповідають за відображення певних елементів сторінки та складаються з модулів. Модулі це функція або декілька функцій які відповідають за окремі задачі пов'язані з функціонуванням компоненту. Ось список модулів які наявні в більшості компонентів:

- Модуль роботи з користувачем
- Модуль роботи з внутрішнім сховищем Redux
- Модуль інтерфейсної частини
- Модуль роботи з ефектами(етапами життєвого циклу компонента)
- Модуль попередньої обробки даних

Загальна схема взаємодії компонентів та модулів системи зображена на рисунку 3.2.

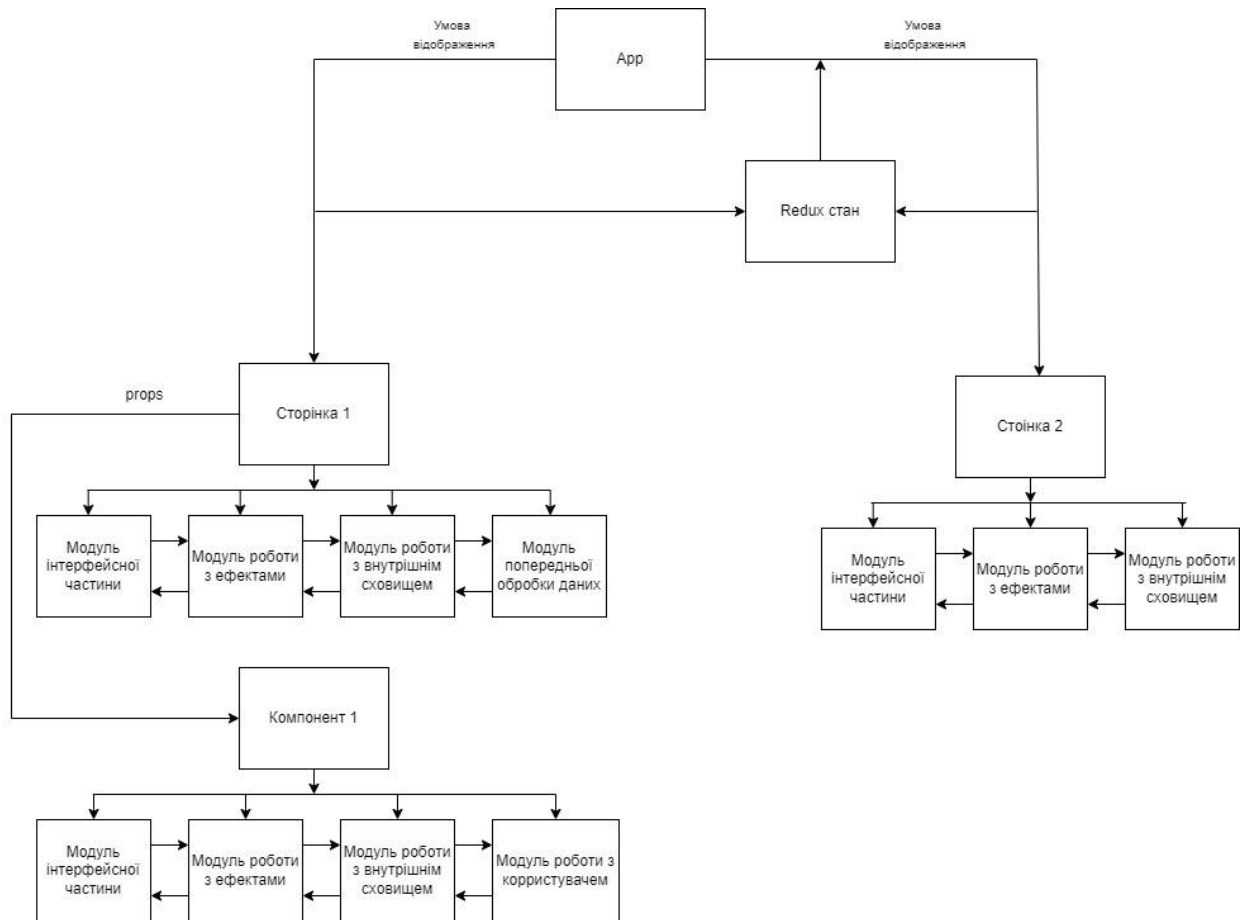


Рисунок 3.2 – Схема взаємодії компонентів та модулів системи

Модуль роботи з користувачем забезпечує взаємодію користувача та веб-застосунку. До його складових частин належать власноруч розроблені функції які замінюють стандартну поведінку активних елементів на сторінці та викликаються в наслідок взаємодії з такими елементами. В нашому проекті такі функції в основному спричиняють виклик оновлення Redux сховища в наслідок чого змінюється відображення на сторінці, перехід по

сторінках застосунку зміну користувацьких налаштувань обробку запитів користувача та появу інтерактивних елементів маркерів тощо.

Модуль роботи з внутрішнім сховищем Redux відповідає за отримання оновлення або зміну даних внутрішнього стану застосунку.

В цих модулях описується логіка з оновлення або отримання даних в залежності від запиту користувача сторінки на якій цей модуль викликається та характеру змін.

Модуль роботи з ефектами гарантує коректне відображення елементів сторінки в залежності від характеру змін компонента сторінки на якій цей модуль викликається розміру екрану пристрою на якому працює система та наявності даних необхідних для відображення. Він викликається завжди при першому відображенні компонента .

Модуль інтерфейсної частини відповідає за зовнішній вигляд та структуру компонента в цьому модулі прописується html розмітка з відповідними назвами класів по яких будуть застосовувати css правила які забезпечують зовнішній вигляд. Окрім того за допомогою React відбувається динамічне розміщення змісту компонента та постановка умови для відображення цих даних якщо це потрібно.

Модуль попередньої обробки даних забезпечує переведення даних в зручний для відображення формат його сортування або за потреби часткове доповнення. Зміни цих даних відбуваються локально тому ніяк не впливають на сховище Redux.

### **3.3 Розробка програмної частини**

Для початку розробки програмної частини проекту потрібно створити шаблон React застосунку за допомогою команди: `npx create-react-app my-app --template cra-template-pwa-typescript` ця команда створить базову структуру проекту

					<i>ІАЛЦ.467200.003 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		45

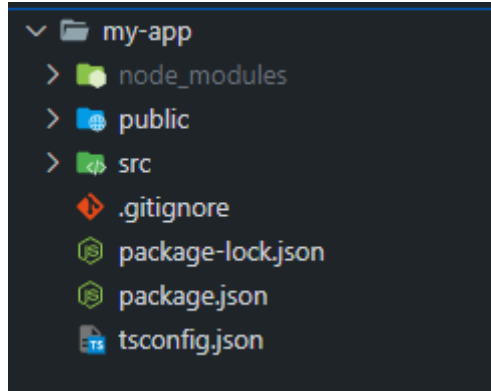


Рисунок 3.3 – Базова структура проекту

- Папка `node_modules` – містить в собі базові бібліотеки React застосунку та бібліотеки завантажені розробником
- Папка `public` – зберігає статичні файли такі як `index.html` `manifest.json` та зображення доступ до яких доступний через браузер без участі серверу
- Папка `src` – містить в собі всі файли для розробки застосунку такі як скрипти і тд згодом ці файли використовуються для фінального складання проекту.
- Файли `package.json` та `package-lock.json` – використовуються для опису проекту а саме його назви версій бібліотек що використовуються в фінальній версії та лише для розробки та можуть містити скрипти
- Файл `tsconfig.json` – файл налаштувань для компілятора TypeScript.

Файлова структура готового проекту виглядає наступним чином:

					ІАЛЦ.467200.003 ПЗ	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		

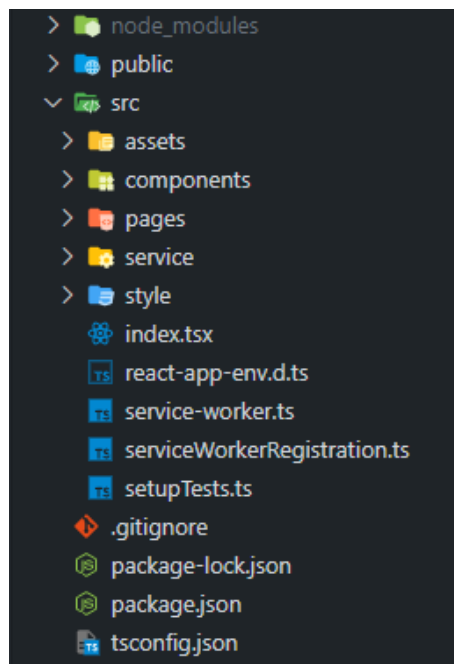


Рисунок 3.4 – Фінальна структура проекту

Окрім базових файлів створених на самому початку було створено такі файли та папки:

- assets – папка в якій зберігаються всі медіа-файли що використовуються для відображення в певних компонентах застосунку
- style – папка в якій зберігаються файли зі стилями окремих компонентів та медіа правила необхідні для адаптивної верстки сайту
- service – папка в якій зберігаються файли які не стосуються логіки роботи React а саме вся логіка роботи бібліотек Redux та Firebase
- components – папка з усіма React компонентами що використовуються в роботі веб-застосунку
- pages – папка зі сторінками нашого веб-застосунку в окремі файли сторінок імпортуються компоненти з яких складаються ці сторінки
- service-worker.ts – файл який відповідає за сервісну частину нашого застосунку в ньому міститься вся логіка для кешування даних та роботи застосунку без доступу до інтернету

					ІАЛЦ.467200.003 ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

- `serviceWorkerRegistration.ts` – в цьому файлі описана логіка реєстрації сервісного працівника так як не кожна версія браузерів підтримує цю технологію тут відбувається перевірка на сумісність та безпосередня реєстрація.
- `index.tsx` – файл який являє собою точку для входу в наш веб застосунок в цей файл імпортується вся клієнтська частина застосунку підключення до Redux та викликається реєстрація `service-worker`.

Основними структурними частинами застосунку являються `react` компоненти які імпортуються та компонуються в окремих файлах сторінок а сторінки в свою чергу імпортуються в файл `App.tsx` який регулює що відображається в даний момент користувачу. Ось список всіх сторінок та компонентів які вони використовують для відображення користувачу.

Сторінка `MainPage.tsx` відображає головну сторінку веб додатку на якій міститься поточний прогност погодинний прогност погоди та тижневий прогност погоди відповідно до міста в якому знаходиться користувач у випадку якщо він не змінював його за допомогою пошукової стічки дана сторінка імпортує в себе такі компоненти:

- `Total.tsx` – відповідає за відображення поточних погодних умов
- `WeekForecast.tsx` – відповідає за відображення тижневого пррогнозу погоди з урахуванням погодних умов та пиблизної темпеатурри повітря
- `TodayForecast.tsx` – відповідає за погодинне погнозування погоди вповодж дня
- `SearchPanel.tsx` – стрічка для пошуку назв міст з автодоповненням тексту на даній сторінці допомагає змінювати місто для прогнозування з поточного на те яке обере користувач

					<i>ІАЛЦ.467200.003 ПЗ</i>	Арк.
						48
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

- AirConditios.tsx – відповідає за формування та відображення опціональних погодних показників таких як ультрафіолетовий індекс відчутна температура та інше
- Spinner.tsx – універсальний компонент який відображається у випадку коли браузер очікує відповіді користувача або застосунок ще не отримав відповідь від стороннього API

Сторінка CitiesPage.tsx відображає список обраних користувачем міст для відстеження та погодинний і триденний прогноз погоди для цих міст в залежності від вибору міста користувачем, якщо цей список пустий відображається прогнозування для поточного міста. На даній сторінці використовуються такі компоненти:

- CityForecast.tsx – частково використовує функціонал компонентів Total.tsx Today.tsx WeekForecast.tsx для відображення погоди для окремого міста
- Spinner.tsx
- SearchPanel.tsx – на даній сторінці не змінює поточне місто користувача а додає міста в список для відстеження

Сторінка MapPage.tsx відображає для користувача інтерактивну мапу та зменшену версію списку з містами зі сторінки CitiesPage.tsx. Компоненти що використовуються:

- Map.tsx – компонент який використовує бібліотеку react-leaflet для відображення та налаштування мапи. Він містить логіку по взаємодії користувача з мапою а саме пошук геолокації точки яку обрав користувач та розміщення й наповнення маркерів.
- CityItem.tsx – компонент який відповідає за оформлення кожного міста;
- Spinner.tsx
- SearchPanel.tsx – на цій сторінці відповідає за перенесення фокусу мапи на обране місто та автоматичне розташування маркера

					ІАЛЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		49

Сторінка `StatsPage.tsx` відображає графіки зі статистикою погодних умові відповідно до міста в якому знаходиться користувач також має можливість змінити місто за допомогою пошукової стічки. Дана сторінка використовує такі компоненти:

- `SeasonChart.tsx` – компонент відображає лінійний графік із показниками мінімальних та максимальних температур за кожен день впродовж останніх трьох місяців. За формування графіків відповідає бібліотека `Chart.js` та її версія для React `react-chart-2`.
- `WeatherConditions.tsx` – відображає три кругових діаграми з кількісними показниками погодних умов кількості опадів та годин під час яких йшов дощ за останніх три місяці
- `SearchPanel.tsx` – на цій сторінці відповідає за міну міста по якому відображається статистика

Сторінка `SettingsPage.tsx` відображає меню налаштувань для одиниць вимірювання погодних показників та для дозволів користувача для використання геолокації та надсилання сповіщень окрім того тут відбувається реєстрація та авторизація користувача. Дана сторінка використовує такі компоненти:

- `SettingsUnits.tsx` – компонент з перемикачами для вибору одиниць вимірювання для всіх сторінок на сайті
- `Switchers.tsx` – відображає перемикачі для дозволів користувача
- `Advertise.tsx` – відображає корисну інформацію для користувача та блок з реєстрацією та авторизацією.

За коректне відображення інформації про застосунок такої як назва застосунку логотип колір фону та теми відповідає файл `manifest.json` який зображено на рисунку:

					<i>ІАЛЦ.467200.003 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		50

```

public > {--} manifest.json > ...
 2   "name": "DL Weather",
 3   "short_name": "Weather PWA",
 4   "theme_color": "#202b3b",
 5   "background_color": "#0b131e",
 6   "display": "standalone",
 7   "orientation": "portrait",
 8   "start_url": ".",
 9   "icons": [
10     {
11       "src": "icons/16.png",
12       "sizes": "16x16"
13     },
14     {
15       "src": "icons/32.png",
16       "sizes": "32x32"
17     },
18     {
19       "src": "icons/57.png",
20       "sizes": "57x57"
21     },
22     {
23       "src": "icons/64.png",
24       "sizes": "64x64"
25     },
26     {
27       "src": "icons/114.png",
28       "sizes": "114x114"
29     },
30     {
31       "src": "icons/128.png",
32       "sizes": "128x128"
33   }

```

Рисунок 3.5 – Файл manifest.json

Аби наш проект міг відображати банер з пропозицією інсталювати застосунок та працювати як повноцінний PWA застосунок він повинен відповідати таким критеріям:

- Сайт повинен працювати через HTTPS з'єднання – це забезпечується розміщенням нашого проекту на платформі Netlify яка забезпечує підтримку https
- Наявність зареєстрованого service-worker що забезпечує кешування даних

					ІАЛЦ.467200.003 ПЗ	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

- Файл manifest.json повинен містити щонайменше назву коротку назву початковий адрес та значення для режиму відображення на всіх пристроях

Відповідність цим критеріям забезпечує коректну роботу PWA.

### 3.4 Модель зберігання даних

В даному проєкті для його роботи не використовується реляційна та нереляційна бази даних. Натомість використовується Redux який зберігає всі необхідні дані для роботи всередині store та дублює ці дані в localStorage браузера для оновлення або повторного використання. За обробку та зберігання даних відповідають так звані зрізи (slice) які створюються за допомогою функції createSlice імпортованої з бібліотеки redux-toolkit ці зрізи зберігають в собі частини глобального стану застосунку та надають інструменти для отримання та модифікації даних. Загалом всі дані проєкту зберігаються в чотирьох зрізах:

- MainSlice – зберігає дані про поточну сторінку такі як її назва та розмір вікна браузера для адаптивного відображення верстки. Окрім того тут зберігаються всі користувацька налаштування які були обрані на сторінці з налаштуваннями
- ForecastSlice – окрім даних про ключі доступу до Арі та статусів запитів зберігає всі дані що стосуються погнозів погоди на всіх сторінках застосунку окрім тих що стосуються маркерів на мапі
- GeocodingSlice – зберігає дані що стосуються геолокацій міст розташування маркерів та можливих назві міст при використанні пошукової стрічки з автодоповненням тексту.
- StatsSlice – зберігає дані що стосуються всієї статистики що відображається в застосунку.

					<i>ІАЛЦ.467200.003 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		52



### 3.5 Розробка UML-діаграм функціонування системи

Для візуального моделювання системи та документування певних аспектів проекту були побудовані так звані UML-діаграми. Такі діаграми створюються для опису візуальних моделей різних аспектів систем та дозволяють розумати проектувати документувати та комунікувати архітектуру та функціональність програмних аспектів. Існує велика кількість типів таких діаграм серед яких були обрані діаграма діяльності системи та діаграма компонентів.

Діаграма діяльності системи (activity diagram) – використовується для моделювання послідовності дій системи та надає можливість візуалізувати ці сценарії. Ця діаграма описує алгоритм роботи нашої системи та описана в додатку 2. Така діаграма представлена графом діяльності основними елементами якої є:

- Вузли – відображають конкретні дії які виконує система або стани в яких перебуває система
- Ребра – відображають переходи між вузлами або зв'язки між ними вони можуть мати умови або вирази які визначають за рахунок чого система переходить до наступного вузла
- Точки вибору – елементи діаграми де система приймає рішення про перехід до іншого вузла на підставі умови або виразу
- Паралельність – показує одночасне виконання декількох дій вона використовується для відображення ситуації коли дії виконуються паралельно без залежності один від одного

Діаграма взаємодії модулів – це тип діаграм який візуалізує структуру компонентів системи та взаємодії між ними вона показує логіку організацію компонентів та залежності між ними. Дана діаграма приведена в додатку 3. Основними компонентами такої діаграми є:

					<i>ІАЛЦ.467200.003 ПЗ</i>	Арк.
						54
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

- Компоненти – основні частини системи або підсистеми кожен компонент виконує певну функцію або надає послуги
- Залежності – показують які компоненти залежать від інших або взаємодіють з ними вони можуть бути напрямленими або ненапрямленими та вказують які компоненти використовують функціональність інших
- Модулі – частини компонентів відповідальні за виконання певних функцій.

					<i>ІАЛЦ.467200.003 ПЗ</i>	Арк.
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		55

## Висновок до розділу 3

У даному розділі був представлений шаблон архітектури, який використовувався під час проектування системи. Крім того, була детально описана архітектура проекту, включаючи склад проекту, його компоненти та модулі. Також була наведена діаграма взаємозв'язків між цими модулями та компонентами, що допомагає зрозуміти взаємодію між ними.

Далі була розроблена програмна частина проекту, де було описано файлову структуру проекту. Це дозволяє отримати загальне уявлення про основні файли, що використовуються у проекті.

Також було розроблено ряд UML-діаграм, включаючи діаграму діяльності системи та діаграму компонентів. Ці діаграми моделюють різні аспекти роботи системи, надаючи візуальне уявлення про послідовність подій та взаємозв'язки між компонентами системи.

На завершення була сформована модель зберігання даних, де було описано компоненти сховища, їх структуру та логіку зміни та оновлення даних. Це допомагає зрозуміти, як дані зберігаються та як вони можуть бути змінені.

					ІАЛЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		56

# РОЗДІЛ 4 ОГЛЯД РЕАЛІЗОВАНОЇ СИСТЕМИ

## 4.1 Огляд реалізованих можливостей PWA

Так як даний додаток розроблений на основі технології PWA варто приділити увагу можливості роботи додатку на пристроях різних операційних систем розглянемо можливість встановлення додатку на різних операційних пристроях та відображення користувацького інтерфейсу на цих платформах.

На операційній системі Windows при використанні браузера в правій стороні пошукової стрічки з'являється зображення комп'ютера при натисканні на яке браузер пропонує встановити додаток на свій пристрій це продемонстровано на рисунку 4.1.

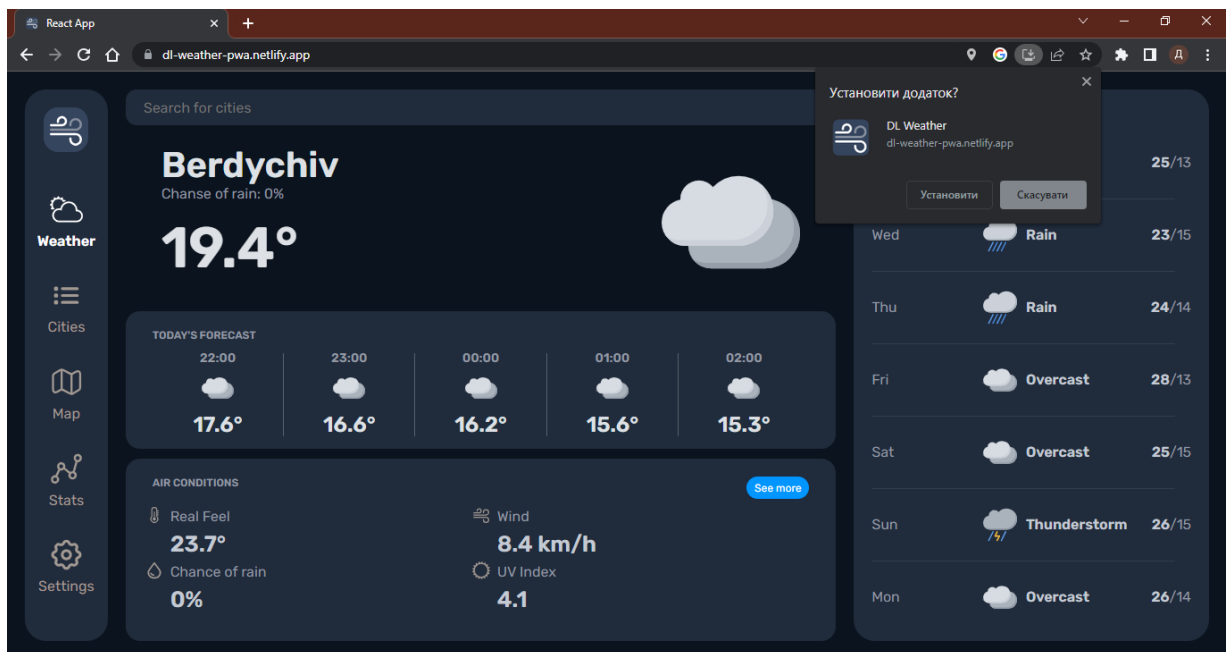


Рисунок 4.1 – Пропозиція встановлення додатку на Windows

Після встановлення додатку на робочому столі з'являється ярлик застосунку при натисканні на який він відкривається без участі браузера дані зміни зображені на рисунку 4.2.

					ІАЛЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		57

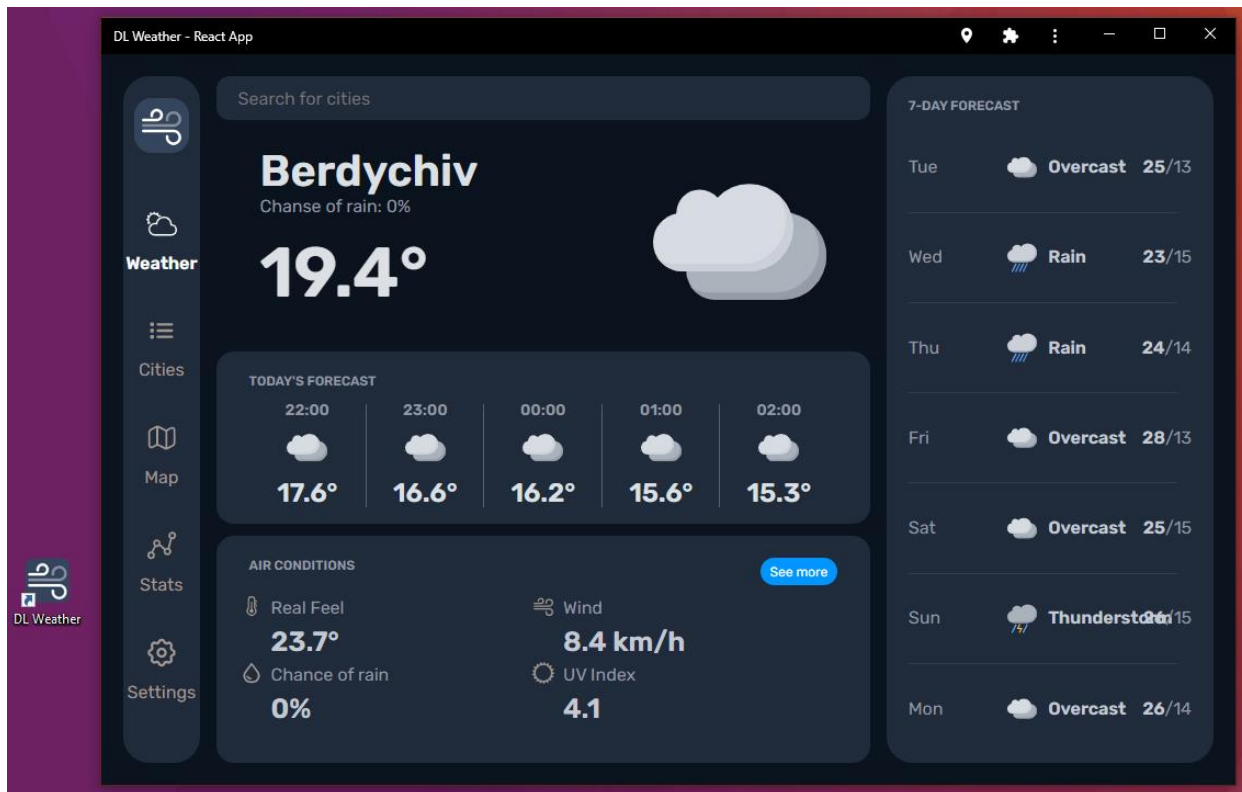


Рисунок 4.2 – Вигляд встановленого застосунку та його ярлика

На мобільних операційних системах Android та IOS при відкритті застосунку в браузері при використанні усіх браузерів окрім Safari він сам запропонує встановити додаток на головний екран пристрою та створить відповідний ярлик це продемонстровано на рисунку. У випадку якщо користувач використовує браузер Safari на телефоні аби встановити додаток потрібно перейти в меню поширити та обрати опцію становити на головний екран.

					ІАЛЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		58



Рисунок 4.3 – встановлення додатку на мобільних пристроях

Те що застосунок можна встановити на головний екран пристрою свідчить про те що наш service-worker був зареєстрований тому залишилось перевірити роботу застосунку при умові відсутності інтернет з'єднання для цього потрібно вимкнути застосунок та доступ до інтернету та ще раз увімкнути наш додаток щоб побачити які файли та дані зберіглись у пам'яті. Як видно з рисунку 4.4 застосунок зберігає свій зовнішній вигляд та відображає останні отримані дані.

					ІАЛЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		59



Рисунок 4.4 – Вигляд застосунку без доступу до інтернету

Для того щоб остаточно впевнитись в тому що цей застосунок використовує в повній мірі технологію PWA був використаний спеціальний інструмент від компанії google під назвою Lighthouse. Цей інструмент аналізує додаток та формує заключення з результатом інтеграції технології PWA у додаток на етапі розробки. Заключення щодо інтеграції та оптимізації цієї технології зображене на рисунку 4.5.

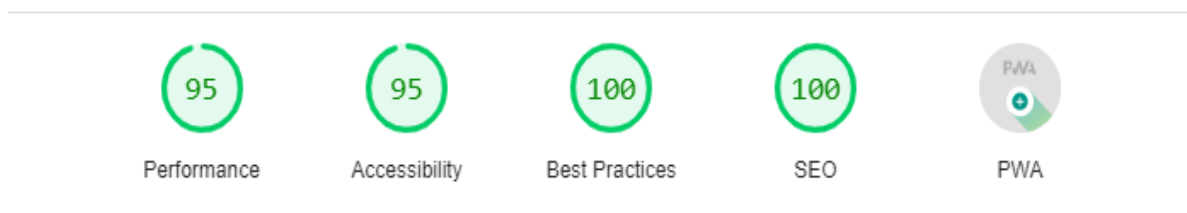


Рисунок 4.5 – Оцінка інтеграції PWA

## 4.2 Огляд основних можливостей системи

При першому використанні додатку або в разі його використання через анонімне вікно браузера додаток автоматично запитає доступ до геолокації у користувача та замінить блоки з даними на інтерактивні елементи “спінери” які сигналізують про очікування дій від користувача або про підготовку даних для відображення. Запит на використання геолокації та елементи які сигналізують про очікування зображені на рисунк 4.6 головна сторінка з отриманим дозволом про використання геолокації зображена на рисунку 4.7.

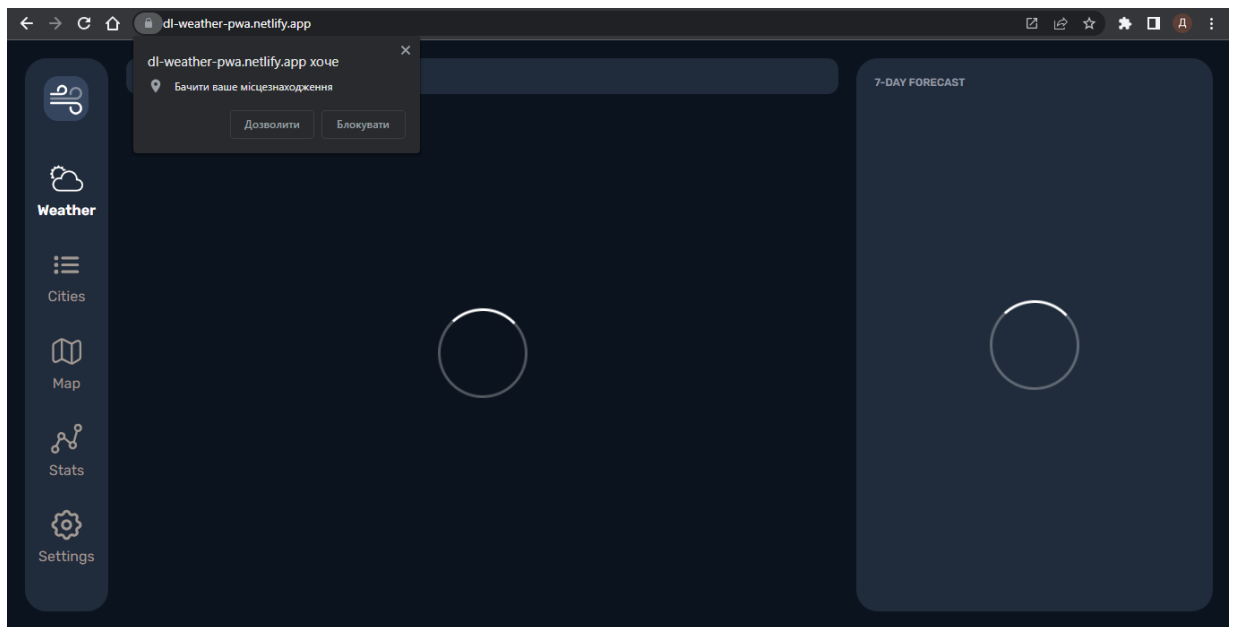


Рисунок 4.6 – Вигляд застосунку під час очікування дій користувача

					ІАЛЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		61

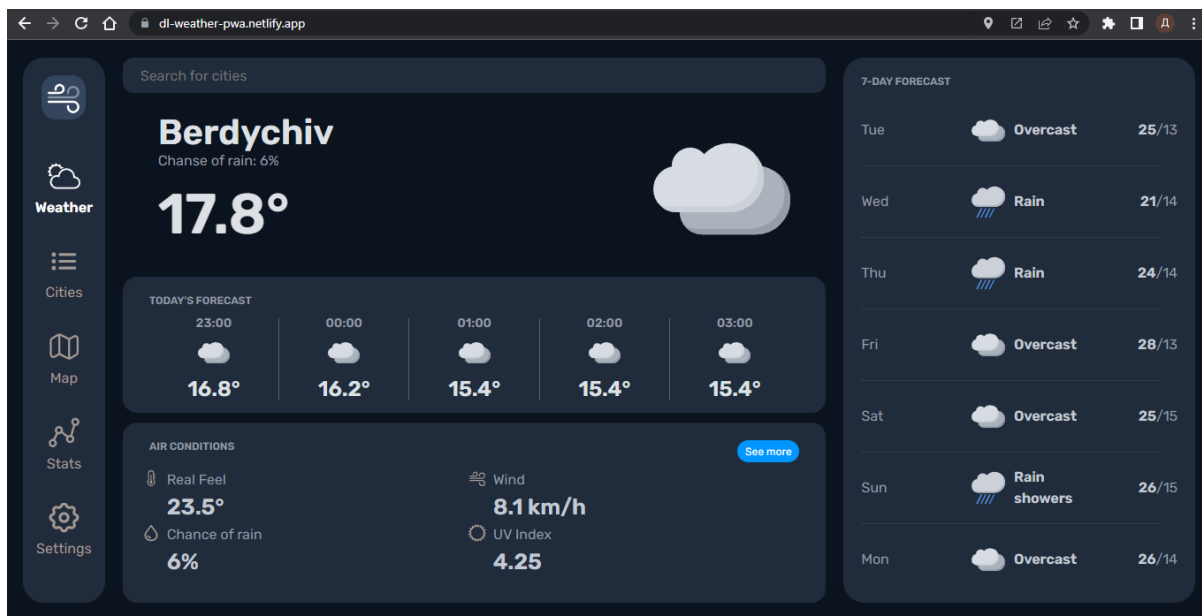


Рисунок 4.7 – Головна сторінка застосунку

На цій сторінці поле вводу використовується для зміни населеного пункту та для автоматичного доповнення тексту що вводить користувач. Пошукова стрічка пропонує лише назви міст та населених пунктів.

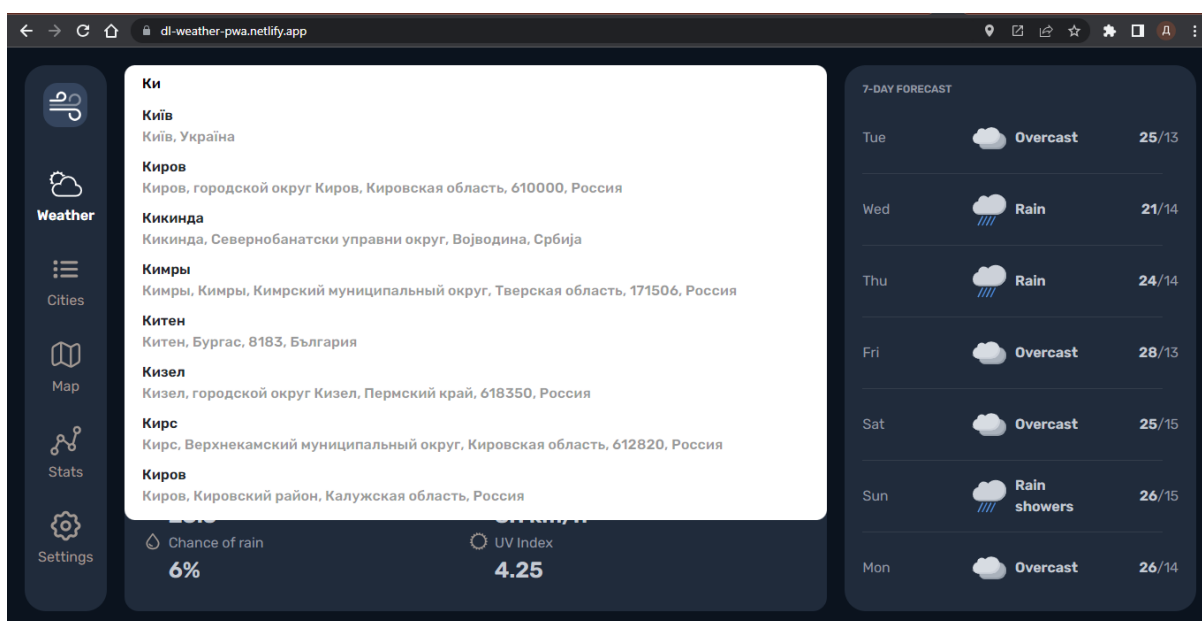


Рисунок 4.8 – Автодоповнення пошукової стрічки

Для навігації по сторінках використовується бічна панель із зображеннями та назвами сторінок. В разі використання додатку на мобільних пристроях або на пристроях з шириною екрану меншою аніж 575

					ІАЛЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		62

пікселів цей блок прихований для його активації потрібно натиснути на ЛОГОТИП

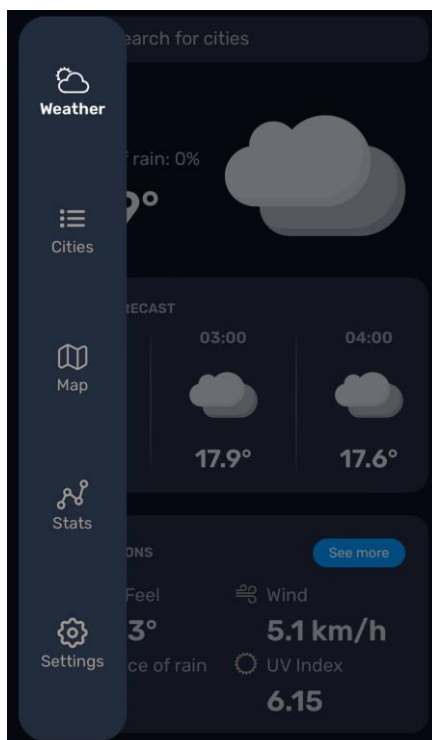


Рисунок 4.9 – Адаптація навігаційного блоку

На другій сторінці застосунку знаходиться список міст який користувач може доповнювати за допомогою пошукової стрічки при натисканні на блок з містом в правій стороні вікна користувачу відображається прогноз погоди для відповідного міста це можна побачити на рисунку 4.10.

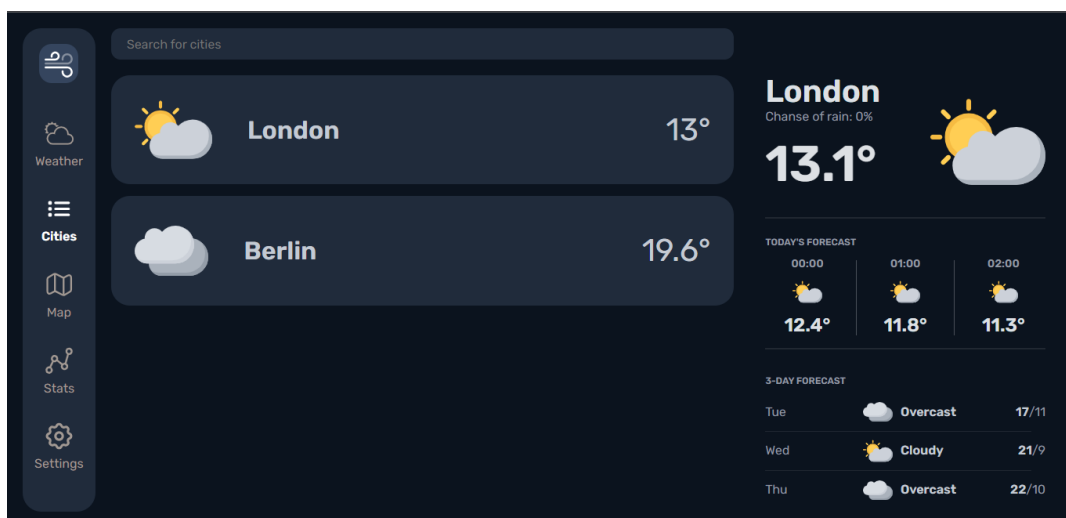


Рисунок 4.10 – Список міст для відстеження

На третій сторінці застосунку розташована інтерактивна мапа з можливістю додавати свої маркери міста які були відмічені відображаються в правій частині сторінки та безпосередньо на мапі. Дані маркери інтерактивні при натисканні на них відображається поточна погода в місті також його можна видалити за допомогою кнопки delete пошукова стрічка на цій сторінці переводить фокус мапи на обране місто.

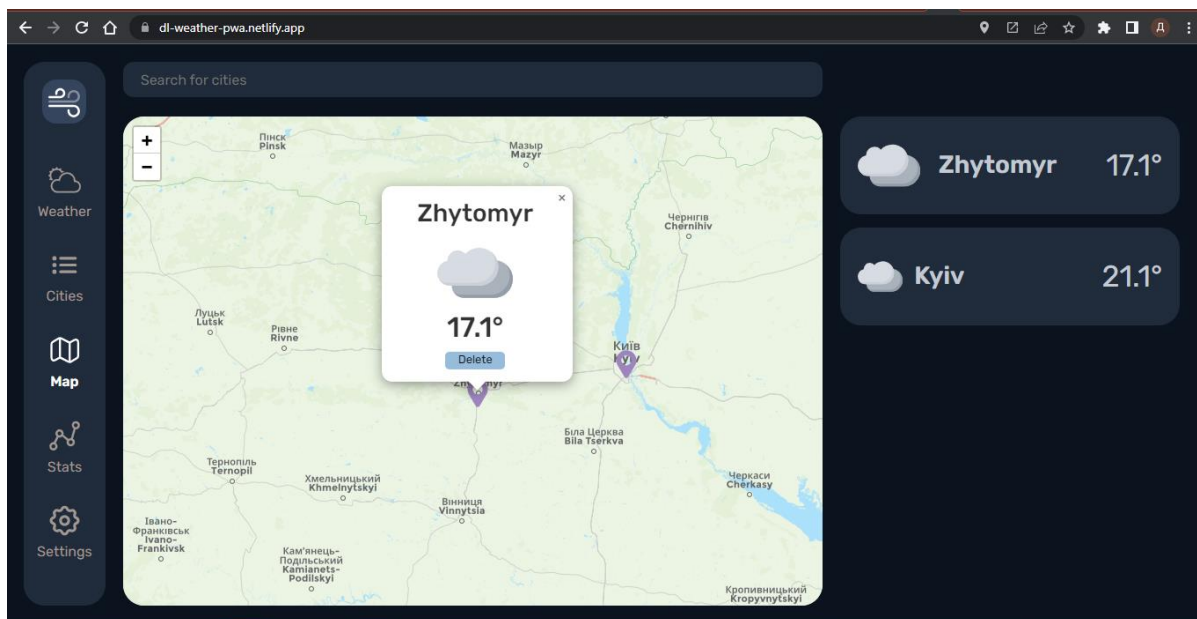


Рисунок 4.11 – Робота інтерактивної мапи

На четвертій сторінці застосунку відображається статистика по показниках погоди за останніх три місяці пошукова стрічка змінює місто для якого відображається статистика. Сторінка містить три кругових діаграми при наведенні на зони яких відображається кількісна характеристика. Написи показників та колір який їх відображає також є інтерактивними при натисканні на них можна прибрати або поновити відображення показника. Лінійний графік відображає показники мінімальної та максимальної температур за кожен день останніх трьох місяців.



## Висновок до розділу 4

В четвертому розділі дипломної роботи представлений готовий веб додаток. Було розглянуто реалізовані можливості PWA застосунку серед яких встановлення застосунку на пристрої з різними операційними системами, адаптація верстки в залежності від виду пристрою та кешування даних для можливості перегляду останнього прогнозу погоди без доступу до інтернету. Окрім того були продемонстровані всі сторінки застосунку з описом їх функціональних можливостей в якості прогнозувальника погоди.

					ІАЛЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		66

## ВИСНОВОК

В рамках даної дипломної роботи був розроблений сервіс для прогнозування погоди на основі технології Progressive Web Applications (PWA). Метою проекту було створення зручного та швидкого інструменту, який надаватиме користувачам актуальну та точну інформацію про погоду.

Для формування вимог щодо функціоналу даного сервісу було оглянуто та проаналізовано найпопулярніші рішення для прогнозування погоди. Внаслідок порівняння цих сервісів сформувався перелік позитивних аспектів які потрібно було реалізувати в даному проекті та перелік проблем які потрібно було вирішити в нашому сервісі.

Під час розробки були використані сучасні технології та інструменти, зокрема React, Redux та Typescript. React дозволив створити ефективну та гнучку інтерфейсну частину, в той час як Redux забезпечив керування станом додатку та однозначну передачу даних. Використання Typescript додало перевагу статичної типізації та покращило безпеку та якість коду.

Сервіс реалізовує різні функціональності, зокрема отримання погодових даних зовнішніми сервісами, відображення прогнозу погоди на зручному інтерфейсі, кешування даних для забезпечення швидкості та доступності, а також можливість роботи в автономному режимі.

Результати дипломної роботи показали, що використання технології PWA дозволяє створити потужний та ефективний сервіс для прогнозування погоди. Даний веб додаток забезпечує користувача усім необхідним для зручного та наочного прогнозування погоди. При відповідному масштабуванні та вдосконаленні, сервіс може стати дуже популярним та затребуваним.

					ІАЛЦ.467200.003 ПЗ	Арк.
						67
Змн.	Арк.	№ докум.	Підпис	Дата		

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Найвідвідуваніші сайти України в 2021 році [Електронний ресурс] – Режим доступу до ресурсу: <https://www.unian.ua/techno/communications/podivisya-v-interneti-top-25-saytiv-yaki-naychastishe-vidviduyut-ukrajinci-11653705.html>.
2. Найвідвідуваніші сайти України в 2022 році [Електронний ресурс] – Режим доступу до ресурсу: <https://konkurent.ua/publication/100330/25-saytiv-yaki-ukraintsi-naychastishe-vidvidovali-v-chervni/>.
3. Що таке Progressive Web App? [Електронний ресурс] – Режим доступу до ресурсу: <http://www.vtei.com.ua/doc/2023/konf1011/zb1.pdf>.
4. Переваги та недоліки PWA [Електронний ресурс] – Режим доступу до ресурсу: <https://arto.agency/ua/pwa/>.
5. Progressive Web Apps: Core Features, Architecture, Pros and Cons [Електронний ресурс] – Режим доступу до ресурсу: <https://www.altexsoft.com/blog/engineering/progressive-web-apps/>.
6. Firebase [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/Firebase>.
7. JavaScript [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/JavaScript>.
8. Що таке React [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.legacy.reactjs.org/tutorial/tutorial.html#what-is-react>.
9. CSS-препроцесори і перевага їх використання. препроцесори [Електронний ресурс] – Режим доступу до ресурсу: <https://virtuapc.ru/tablets/css-preprocessor-i-preimushchestvo-ih-ispolzovaniya-preprocessor/>.

					ІАЛЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		68

# **ДОДАТОК 1**

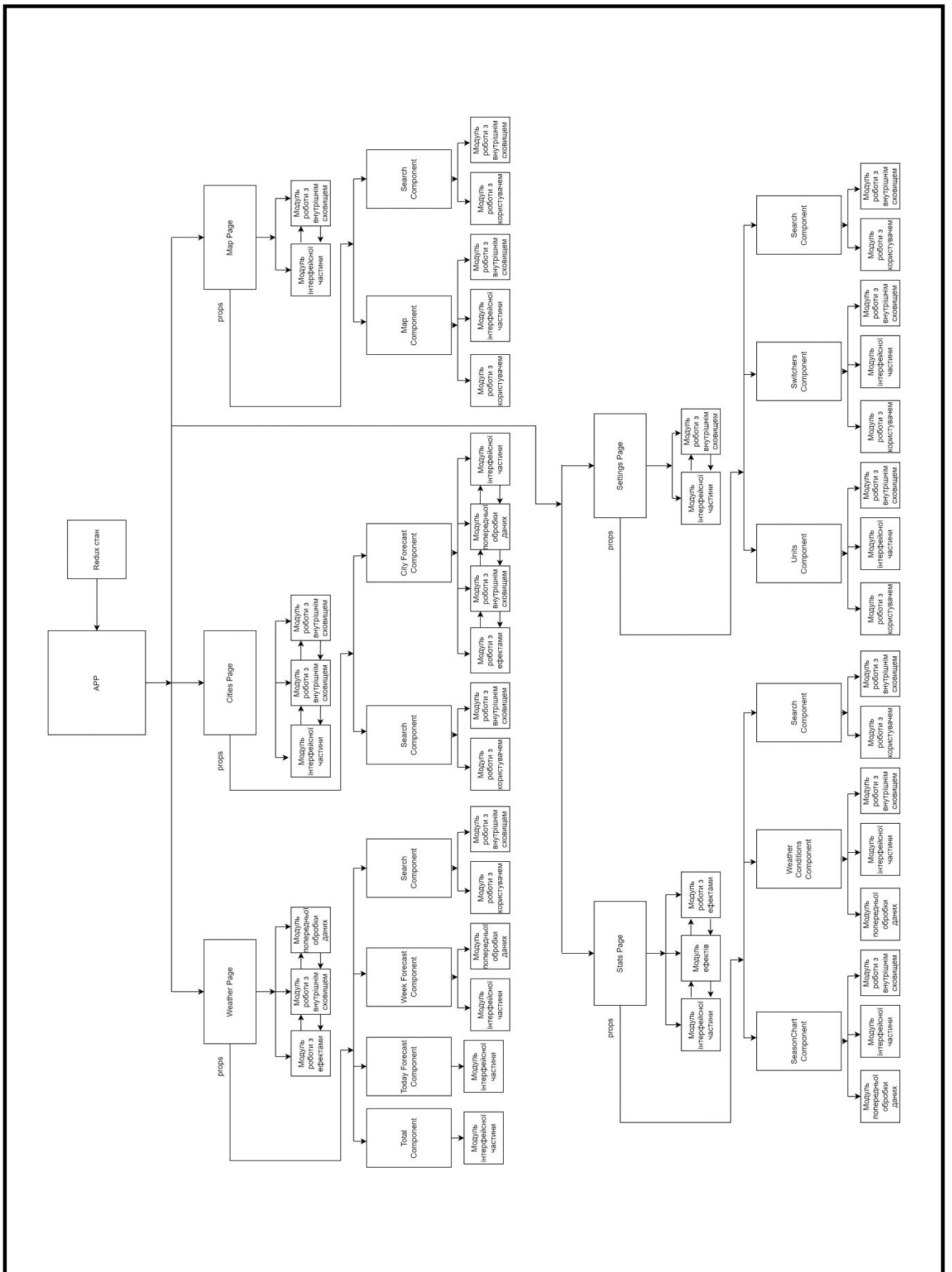
**Прогнозувальник погоди на основі технології PWA**

**Діаграма взаємодії модулів (структурна схема)**

**ІАЛЦ.467200.004 Д1**

**Аркушів 1**

**Київ 2023 р**



Змін.	Арк.	№ докум.	Підпис	Дата
Розробив		Лінійчук Д.В.		
Перевір.		Павлов В.Г.		
Н. контр.		Виноградов Ю.М.		
Затверд.		Стіренко С.Г.		

ІАЛЦ.467200.004 Д1

Прогнозувальник погоди на основі технології PWA  
 Діаграма взаємодії модулів(структурна схема)

Літ.	Аркуш	Аркушів
	1	1
КПІ ім. Ігоря Сікорського ФІОТ ІВ-93		

## **ДОДАТОК 2**

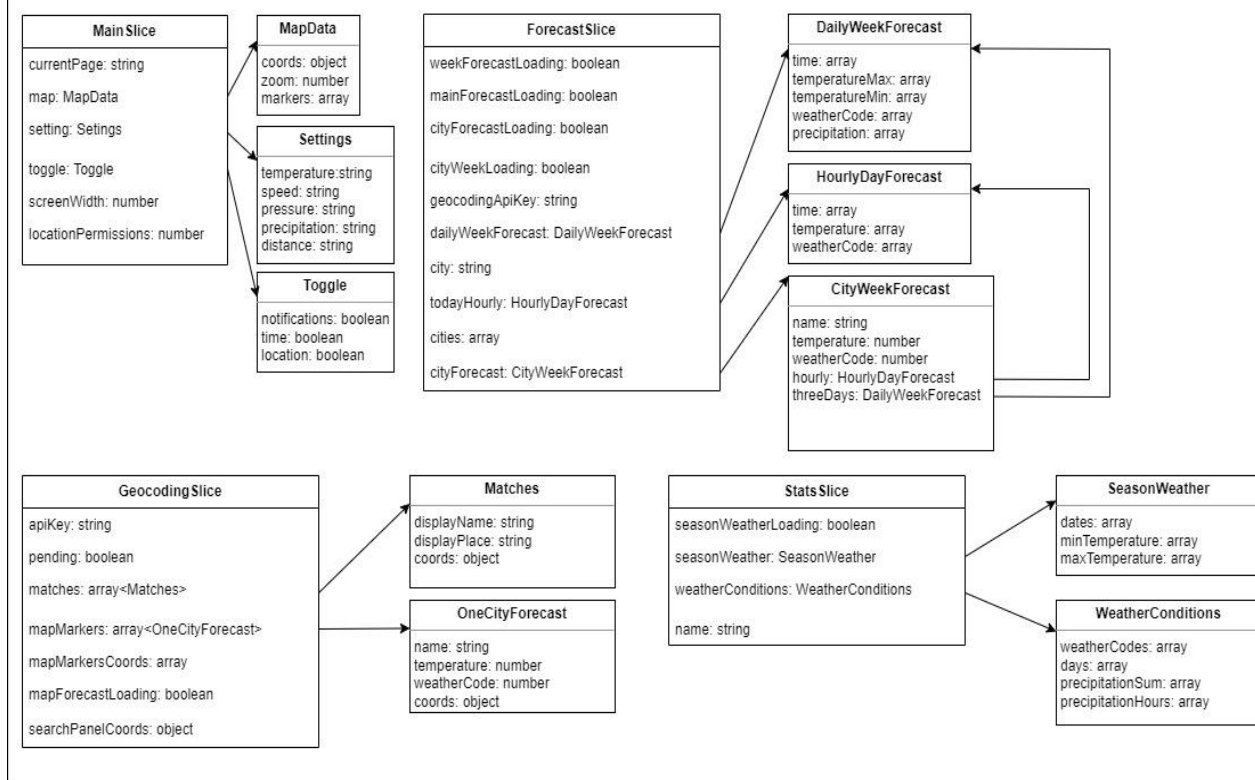
**Прогнозувальник погоди на основі технології PWA**

**Модель зберігання даних(функціональна схема)**  
ІАЛЦ.467200.005 Д2

Аркушів 1

**Київ 2023 р**

## Redux



<b>ІАЛЦ.467200.005 Д2</b>				
<i>Змін.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>
<i>Розробив</i>		<i>Лінійчук Д.В.</i>		
<i>Перевір.</i>		<i>Павлов В.Г.</i>		
<i>Н. контр.</i>		<i>Виноградов Ю.М.</i>		
<i>Затверд.</i>		<i>Стіренко С.Г.</i>		
<b>Прогнозувальник погоди на основі технології РWA Модель зберігання даних(функціональна схема)</b>			<i>Лім.</i>	<i>Аркуш</i>
			1	1
<b>КПІ ім. Ігоря Сікорського ФІОТ ІВ-93</b>				

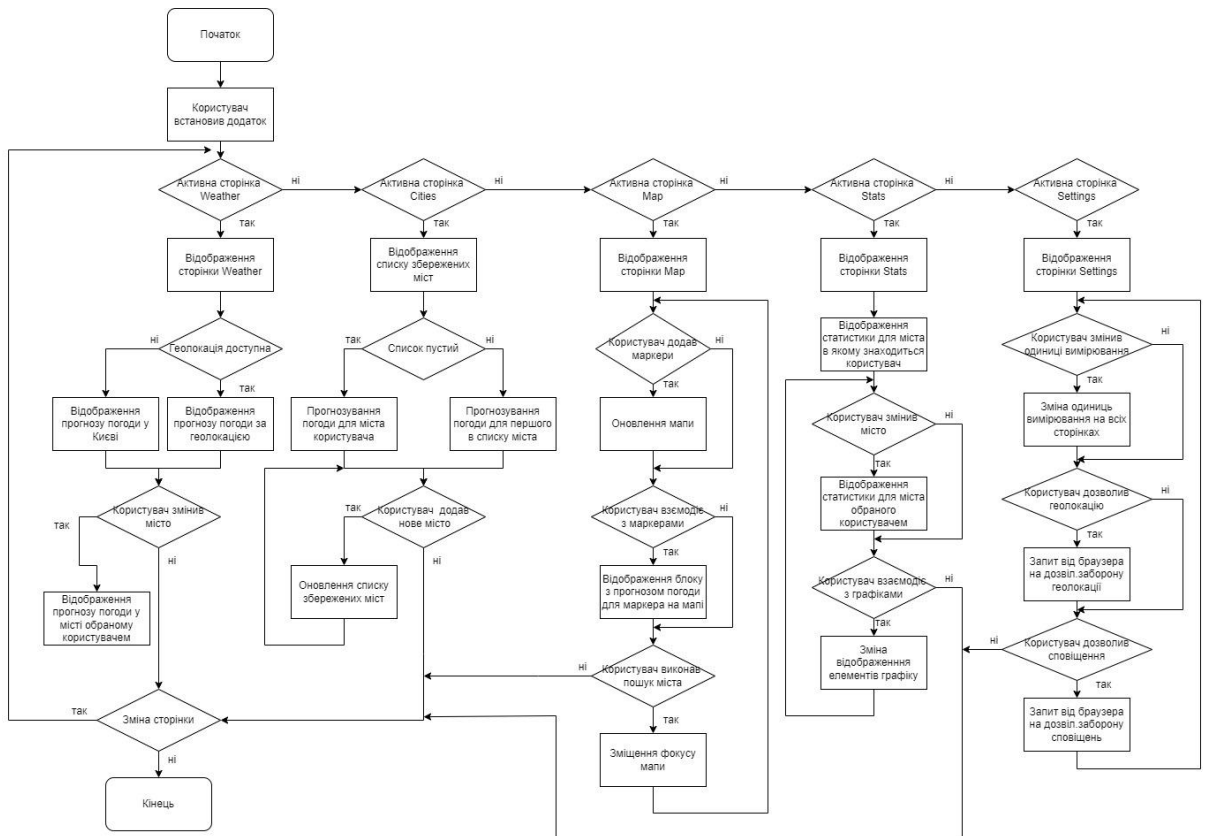
## **ДОДАТОК 3**

**Прогнозувальник погоди на основі технології PWA**

**Діаграма діяльності системи(принципова схема)  
ІАЛЦ.467200.006 ДЗ**

**Аркушів 1**

**Київ 2023 р**



<b>ІАЛЦ.467200.006 ДЗ</b>				
Змін.	Арк.	№ докум.	Підпис	Дата
Розробив		Лінійчук Д.В.		
Перевір.		Павлов В.Г.		
Н. контр.		Виноградов Ю.М.		
Затверд.		Стіренко С.Г.		
Прогнозувальник погоди на основі технології PWA Діаграма діяльності системи (принципова схема)			Літ.	Аркуш
				1
			КПІ ім. Ігоря Сікорського ФІОТ ІВ-93	