1 Project Links

- **GitHub Repository:** https://github.com/Mundheanil84/Weather-App
- **Live Demo:** Weather App

2 Project Goal

The goal of this project is to build a **clean and simple single-page web application** that allows users to search and view the **current weather conditions** for any city worldwide.

The application focuses on:

- Simplicity
- Responsiveness
- Performance
- User experience

3 Core Features

1. **City Search Bar** – Enter a city name to fetch real-time weather data.
2. **Weather Display** – Shows temperature, weather condition (e.g., Clouds, Rain), and humidity.
3. **Loading Indicator** – Provides feedback while fetching data.
4. **Error Handling** – Displays clear messages if the city is not found.
5. **Responsive Design** – Works seamlessly on desktop, tablet, and mobile.

4 Bonus Features

- **Persistent Storage:** Saves the last searched city in `localStorage`.
- **5-Day Forecast:** Provides extended weather insights.
- **Advanced State Management:** Handles loading, error, and success states effectively.
- **User-Friendly UI:** Smooth animations, minimal dependencies, and clean styling.

5 Technology Stack

- **Frontend Framework:** React.js (with Vite for fast builds)
- **Styling:** CSS Modules, Flexbox, Grid, CSS Variables
- **Icons:** Lucide React

- **API:** OpenWeatherMap API
- **Testing Framework:** Jest + React Testing Library
- **Tools:** Git, GitHub, VS Code

6 Project Structure

```
weather-app/
          public/
          src/
                components/
                      SearchBar.jsx
                      WeatherDisplay.jsx
                      CurrentWeather.jsx
                      Forecast.jsx
                App.jsx
                index.js
                styles/
          package.json
          README.md
          LICENSE
```

Listing 1: Project Directory Structure

7 Setup and Installation

7.1 Prerequisites

- Node.js (v14 or higher)
- npm or yarn

7.2 Steps to Run Locally

```
# Clone repository
git clone https://github.com/yourusername/weather-app
cd weather-app

# Install dependencies
npm install

# Start development server
npm run dev
```

Listing 2: Local Setup Commands

Open http://localhost:3000 in your browser.

7.3 Production Build

```
npm run build
npm run preview
```

Listing 3: Production Build Commands

8 Testing Instructions

8.1 Running Tests

```
npm test
npm run test:watch
npm run test:coverage
```

Listing 4: Test Execution Commands

8.2 Test Coverage

- **Unit Tests:** Component rendering, state management, user interactions, local storage persistence.
- **Integration Tests:** Search functionality, weather data fetching & display, error handling.
- **Manual Testing:** Valid/invalid searches, responsive design on devices, persistence after refresh.

9 API Integration

9.1 OpenWeatherMap API

- **API Key:** Required (free tier available)
- **Endpoints Used:**
  - Current Weather Data: `api.openweathermap.org/data/2.5/weather`
  - 5-Day Forecast: `api.openweathermap.org/data/2.5/forecast`
- **Data Format:** JSON responses with weather metrics

10 Assumptions and Design Choices

- **API Integration:** OpenWeatherMap chosen for global coverage and free tier.
- **State Management:** React Hooks (`useState`, `useEffect`) for simplicity.
- **Persistent Storage:** `localStorage` for saving last searched city.
- **Performance:** Code splitting with Vite, efficient re-renders, minimal dependencies.
- **Styling:** CSS Modules, Flexbox/Grid, mobile-first design.

11 Evaluation Criteria

- **Skills & Code Quality:** Clear component structure, effective state handling, clean UI code.
- **Completion:** Fetches and displays weather data with loading and error states.
- **Bonus Points:** Persistent city search, 5-day forecast, smooth animations.

12 Future Enhancements

- Geolocation-based weather fetching
- Interactive weather maps
- Weather alerts & notifications
- Multi-language support
- Dark/Light theme switching
- Offline support with Service Workers