**Weather Information Web Application Documentation**

**Table of Contents**

1. **Project Overview**
2. **Backend (API) Implementation**
   * Architecture
   * Endpoints
   * Error Handling & Validations
   * AI Tool Usage
3. **Frontend Implementation**
   * Architecture
   * User Interface
   * Validations
   * Exception Handling
   * AI Tool Usage
4. **Deployment & Integration**
   * Git Integration
   * Publishing to GitHub
5. **Conclusion**

**1. Project Overview**

The Weather Information Web Application is designed to fetch and display weather-related data, including current weather, 5-day forecast, and air quality information. Users can retrieve weather data based on city names or pincodes. The application is developed using .NET for the backend API and Angular for the frontend UI.

**2. Backend (API) Implementation**

**Architecture**

The backend is built using ASP.NET Core Web API. The application follows a standard architecture with Controllers to handle HTTP requests and a Repository layer to manage data fetching from external services (OpenWeatherMap API).

**Endpoints**

* **GET /api/weather/current?city={city}**: Fetches current weather data for the specified city.
* **GET /api/weather/forecast?city={city}**: Retrieves a 5-day weather forecast for the specified city.
* **GET /api/weather/airquality?city={city}**: Fetches air quality information for the specified city.
* **GET /api/weather/currentbypincode?pincode={pincode}**: Fetches current weather data based on the provided pincode.

**Error Handling & Validations**

* Each endpoint is wrapped in a try-catch block to handle exceptions such as network errors or invalid API responses.
* Validation is performed on the input parameters (e.g., checking if the city name is provided).

**AI Tool Usage**

* AI assistance was used to generate and refine the code for API endpoints, especially for error handling and validation.
* AI tools helped automate the creation of the repository pattern and simplified the integration with the OpenWeatherMap API.

**3. Frontend Implementation**

**Architecture**

The frontend is built using Angular, with a component-based architecture. The main components include:

* **AppComponent**: Manages the main application logic and user interactions.
* **CurrentWeatherComponent**: Displays current weather information.
* **WeatherForecastComponent**: Displays the 5-day weather forecast.
* **AirQualityComponent**: Displays air quality information.
* **WeatherByPincodeComponent**: Handles weather data based on pincode input.

**User Interface**

The UI includes input fields for city names and pincodes, buttons to trigger data fetching, and sections to display weather data.

**Validations**

* The city name input is validated to ensure it's not empty.
* The pincode input is validated to ensure it's exactly 6 digits.
* Validation errors are displayed to the user before making API calls.

**Exception Handling**

* API errors are caught and displayed to the user with a friendly message.
* Angular's reactive error handling was leveraged to manage HTTP errors gracefully.

**AI Tool Usage**

* AI was used to suggest the Angular component structure and optimize data binding practices.
* Validation logic for the frontend was refined using AI suggestions, ensuring robust user input handling.

**4. Deployment & Integration**

**Git Integration**

* Git was used for version control. The project was initialized with git init in Visual Studio Code.
* The project version was checked using git --version, and commits were made regularly to track progress.

**Publishing to GitHub**

* The project was pushed to GitHub using the command line interface, allowing for remote collaboration and backup.
* AI assisted in setting up Git integration and provided guidance on resolving any Git-related errors.

**5. Conclusion**

The development of this weather information web application leveraged both modern development frameworks and AI tools to enhance productivity and code quality. The backend API was constructed to be robust and secure, while the frontend was designed with user-friendly validations and error handling. AI tools were instrumental in optimizing the development process, providing code suggestions, and helping troubleshoot issues.