

## Project Presentation: Weather Application

Introduction: Welcome everyone to the presentation of our weather application project. In this presentation, we will discuss the features we have completed, the tasks that are still in progress, potential bugs, and further improvements to enhance our application.

### Completed Features:

#### Weather Data Management:

Fetching current weather data by city name.

Inserting weather forecast data for a specified city.

Querying weather data by city name and timezone offset.

Fetching weather data using coordinates.

Removing weather entries from the database.

#### Historical Weather Data:

Fetching historical weather data for cities.

Querying historical weather data by city name.

#### Database Operations:

Establishing a connection to the MySQL database.

Executing queries and handling database operations.

#### Express.js Server Configuration:

Configuring routes for weather-related functionalities.

Handling middleware for request parsing and error handling.

#### Frontend Integration:

Integration with frontend components to display weather data.

### Unfinished Promised Tasks:

#### Frontend Integration:

Further integration of backend functionalities with frontend components for a seamless user experience.

#### User Authentication and Authorization:

Implementing user authentication and authorization mechanisms for user-specific actions and data security.

#### UI/UX Improvements:

Enhancing the user interface with interactive elements, search functionalities, error notifications, and overall user experience enhancements.

#### Potential Bugs:

##### Error Handling:

Incomplete error handling could lead to unhandled exceptions or unclear error messages for users.

Redirects in error cases might not provide clear feedback to users.

##### Input Validation:

Lack of comprehensive input validation in some endpoints could result in potential security vulnerabilities or unexpected behavior.

#### Frontend-Backend Sync:

Synchronization issues between frontend and backend components could cause data inconsistencies or display errors.

#### Further Improvements:

##### Geocoding Integration:

Utilizing Geocoding API to fetch coordinates for cities, enhancing location-based weather data retrieval.

##### Weather API Integration:

Interacting with Weather API for current weather and forecast data, handling responses and data transformations.

#### Database Management:

Managing weather data storage in the MySQL database, including insertion, deletion, and querying operations.

#### Middleware and Routing:

Implementing middleware for request parsing and error handling in Express.js.

Defining routes and handling various endpoints for weather data operations.

Asynchronous Operations:

Utilizing asynchronous operations with `async/await` for API calls and database interactions, ensuring non-blocking execution.

Client-Side Timezone Handling:

Calculating local time based on the client's timezone offset for accurate weather data retrieval.

Data Presentation:

Formatting and presenting weather data in JSON format for client-side consumption.

Potential implementation of frontend components to display weather information graphically or interactively.

Conclusion: In conclusion, our weather application has made significant progress in implementing core functionalities. However, there are still tasks to be completed and areas for improvement. With continued efforts and enhancements, we aim to deliver a robust and user-friendly weather application. Thank you for your attention. Now, we welcome any questions or feedback.