Prototype II

Strengths:

Modular Design: The architecture of this version of the weather app is built upon a modular design . HTML,CSS handle the look and feel, Javascript is responsible for fetching the data from the php and populating the HTML tags, PHP is responsible for providing the data for the Javascript, and making queries to the database. Thus, everything is utilized in a way that each component handles a specific aspect of the application making it easier to build, maintain and debug.

Efficient Data Handling: Using PHP, along with MySQL database, enables efficient data handling. Storing each searched city's data in the database allows for quick retrieval when needed and refreshing the data if it is older than 2 hours ensures accurate data readings. By storing data locally, the app reduces reliance on external APIs for every user request, thus improving performance.

Weaknesses:

Over-reliance on External API: This version of weather app is solely reliant on OpenWeatherMap for its data. This, while making things simple, could be a very big problem in peak usage hours or API Downtime causing the application to be slow and sluggish. One possible solution could be to integrate additional weather API's alongside OpenWeatherMap to diversify data sources and reduce reliance on a single service.

Performance Overhead with Database Calls: Making a call to the database for each search query can lead to increased database load and latency, especially as the number of users grows. This can result in slower response times and decreased overall application performance. To mitigate this in memory caching mechanisms could be implemented using server side tools like Redis.