Student number: 2427221

Student name: Abduqahhorov Arslonbek

Reflective Report on App Architecture

The architecture of my weather application is meticulously crafted to efficiently deliver real-time weather data to users. A notable strength lies in the utilization of a relational database, which enables efficient storage and retrieval of weather information. The implementation of prepared statements in PHP enhances the security of the application by safeguarding against SQL injection attacks. Additionally, the architecture's clear separation of server-side logic in PHP and client-side interactivity in JavaScript adheres to industry best practices, ensuring maintainability and scalability.

One of the significant improvements in the architecture is the implementation of a caching mechanism. This strategic addition optimizes data retrieval, reducing server load and enhancing responsiveness. The caching mechanism effectively minimizes redundant API calls, thereby improving the overall performance and user experience.

However, despite these strengths, the architecture does have some areas that require attention. A primary weakness is the limited robustness in error handling and exception management. The absence of comprehensive error handling can result in unexpected disruptions in the user experience due to unhandled exceptions from API requests or database operations. Moreover, the synchronous data fetching from the OpenWeatherMap API, even with caching in place, may still lead to performance bottlenecks under high user load.

In conclusion, while the architecture of my weather application effectively serves its primary purpose and includes beneficial features like caching, there are still areas for improvement in error handling, further performance optimization, and scalability. Future iterations of the application should focus on addressing these weaknesses to enhance overall reliability, efficiency, and user satisfaction.