Suvam Kharel

CS 499 Capstone

04 April 2025

**Module Five Narrative**

The artifact I chose for this assignment was the Event Tracking App that was a project I worked on CS 360: Mobile Architecture and Programming course, which was a course I took last year. This app allows users to manage and track events, with features such as adding events, storing descriptions, viewing them in a list, and deleting events. In Milestone Four, I focused on enhancing the backend infrastructure of the app. Specifically, I replaced the local SQLite database with a Flask-based API that communicates with a MySQL database hosted locally through XAMPP and managed using phpMyAdmin. I selected this artifact for my ePortfolio because this enhancement demonstrates my ability to integrate backend web services with a native Android application. It showcases my skills in full-stack development, API consumption, database design, and Android-API integration using Volley.

The specific enhancements I completed include:

* Created and configured a MySQL database using XAMPP and phpMyAdmin to replace SQLite for better scalability.
* Developed a Flask backend API (server.py) to handle registration, login, event retrieval, insertion, and deletion using HTTP endpoints.
* Updated the Android app to make HTTP requests via Volley instead of local database queries.
* Replaced database helper logic with real-time interaction with the Flask API, allowing dynamic updates to the UI.
* Verified functionality end-to-end: successful registration/login, event add/delete, and real-time event display sorted by name/date using backend filtering.

Overall, this artifact was enhanced by replacing the local storage with scalable backend system for proper data management.

In module one, I had planned to make the following changes, and all the enhancement categories were met:  
1. Create an events table in MySQL to store event details.

2. Write a backend API to handle event retrieval, insertion, and deletion using MySQL.

3. Replace SQLite queries with HTTP requests to interact with MySQL through the API.

4. Ensure event data is correctly stored, retrieved, and displayed in the app.

During the enhancement, I faced several challenges and learned important lessons. One issue was ensuring the emulator could communicate with the Flask server running on localhost. This required configuring the correct IP and allowing cleartext HTTP traffic. Another learning point was handling MySQL datetime formats and making sure the API returned event data in a consistent structure to parse on the Android side.

I also learned how to test endpoints using Postman and ensured the application handled all CRUD operations properly (Screenshots included below). Implementing server-side sorting and connecting it to client-side controls was a major improvement over static sorting in the original version.

Overall, these enhancements significantly modernized the architecture of the application, making it more realistic and production-ready for real-world deployment. This experience deepened my understanding of client-server interactions, REST APIs, and backend integration in mobile apps.

Applicable Screenshots:

*New update with adding events:*

A screenshot of a calendar

AI-generated content may be incorrect.

*MySQL table showing stored events:*

A screenshot of a computer

AI-generated content may be incorrect.

*Adding new event in UI to test:*

A screenshot of a computer

AI-generated content may be incorrect.

*Flask Update to show the database updated:*A screenshot of a computer screen

AI-generated content may be incorrect.

*Updated database:*A screenshot of a computer

AI-generated content may be incorrect.

*Using PostMan to directly add events:*A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

*UI utilizing sort feature on the client side:*A screenshot of a computer

AI-generated content may be incorrect.

*Showing live updates with search functionality:*A screenshot of a phone

AI-generated content may be incorrect.