

# Expense Tracker

Aaron Vinodbhai Christian

Initially, my team had a well-defined plan for developing our application. We intended to use the Laravel PHP framework with a MySQL database and a graphical UI built with HTML, CSS, and JavaScript. However, during the second session, we realized that we did not have enough time to complete the application as originally envisioned. As a result, we had to make significant adjustments to our development strategy.

We decided to switch our framework and UI approach, opting for a text-based interface rather than a graphical one. Given that all team members were proficient in C++, we chose to develop the application in C++ and use a CSV flat file for data storage. Task division was well-structured within the group, with each member taking responsibility for their assigned tasks. From my perspective, the team functioned effectively because each person understood their role and expectations, contributing to a successful collaboration.

During the first session, we were tasked with creating a Software Requirement Specification (SRS) document. This document covered functional and non-functional requirements, system requirements, and a flowchart. One challenge we faced was that the TA, acting as the client, was uncertain about the specific requirements. As a result, we had to make assumptions based on system requirements, leading us to initially plan for an application using Laravel and MySQL.

The second session was significantly more demanding. While we managed to develop the basic requirements of the application during the lecture, the rushed development process resulted in several minor bugs. Our team dedicated the rest of the day to debugging and refining the logic of each module. By the end of the day, we successfully uploaded a functional version of our application, free of major bugs and meeting the essential requirements.

Today marked our third and final session, where our primary goal was to finalize the application by addressing any remaining errors and making final improvements. Additionally, we took the opportunity to implement optional functionalities that, while not required, added value to the project. We successfully completed the application, resolved all bugs, and incorporated test cases to ensure functionality.

Through this project, I reinforced a crucial lesson that I have encountered multiple times: the importance of thoroughly gathering system and user requirements. Understanding these aspects early in the development process can save time on architectural design, allowing developers to focus on efficient and scalable application development.