**CS 499 Milestone Two: Software Design and Engineering Enhancement Narrative**

**Artifact Overview**  
 The selected artifact for this milestone is a mobile application called **Crypto Tracker**, originally developed in CS 360: Mobile Architecture and Programming. This Android application allows users to view real-time cryptocurrency data through an interactive and responsive UI. The project was originally built using XML-based layouts and a monolithic architecture, with UI logic tightly coupled to data access code. The artifact was created in early 2024 as part of the course’s final project.

**Justification for Inclusion**  
 I selected this artifact for my ePortfolio because it highlights my progress in applying professional software engineering practices, particularly in mobile application development. The enhancements I implemented demonstrate my ability to refactor an existing codebase using modern design principles. Key components that showcase my skills include the introduction of **Clean Architecture** (separating the app into domain, data, and presentation layers), implementation of the **MVVM (Model-View-ViewModel)** pattern, and migration of the UI from XML to **Jetpack Compose**. These improvements make the codebase more modular, scalable, and testable. Additionally, I incorporated **Kotlin Coroutines** for asynchronous data handling and improved error handling with a Result wrapper, showing my ability to deliver production-quality code.

**Outcome Alignment**  
 This enhancement aligns directly with the following CS program outcomes:

* **Design and evaluate computing solutions** using standard practices (refactoring to Clean Architecture)
* **Use well-founded techniques and tools** in computing practices (e.g., Jetpack Compose, MVVM, Coroutines)
* **Develop a security mindset** (by decoupling data sources and handling errors gracefully)

All outcomes planned in Module One for this category have been addressed, and no changes are currently needed.

**Reflection on the Enhancement Process**  
 The enhancement process required me to deeply re-evaluate the architecture and long-term maintainability of the application. I learned how crucial clean separation of concerns is—not just for code clarity, but for scalability and testing as well. Migrating to Jetpack Compose also allowed me to modernize the UI, making it more responsive and easier to maintain. The biggest challenge was restructuring the code while ensuring feature parity and maintaining existing functionality. I also had to debug lifecycle issues introduced by improper state handling, which taught me more about the nuances of Android’s architecture components.

Overall, this enhancement significantly improved the quality of the artifact and deepened my understanding of professional software design principles.