System Overview and Rationale

My goal was to build a robust, scalable vertical slice, treating the assignment as a professional task. I chose a 3D environment and selected art assets from Kenney and KayKit to establish a cohesive medieval fantasy theme.

The project is built on several key architectural decisions: Inventory System: I implemented the Model-View-Presenter (MVP) pattern, ensuring a clean separation between data, UI, and logic. This decoupling was crucial for maintainability and simplified serializing data for the save/load feature.

For UI input, I used Unity's optimized EventSystem for pointer interactions, reserving the newer Input System for global shortcuts to avoid unnecessary complexity.

Save/Load System: The save system ensures data persistence by serializing the inventory to a JSON file. It saves the GUID of each ItemData ScriptableObject, not raw data, which prevents bloat and keeps saves valid even if assets are renamed.

Personal Performance Assessment

I am pleased with the final result, having implemented all core requirements with a focus on scalable, maintainable code. The inventory and input systems are particular highlights that demonstrate a strong architectural foundation.

My main challenge was time management. I spent considerable time exploring movement systems, which, while leading to a polished result, reduced time for secondary features. Consequently, the equipment code did not receive the same level of refinement and is a clear area for future improvement.

This project reinforced the importance of prioritizing core tasks and building upon a solid architectural base. I am confident that the final product effectively demonstrates my technical skills and problem-solving abilities.