This prototype implements a multiplayer cooking game using Unreal Engine 5, where players interact with kitchen stations to prepare ingredients and serve dishes. The gameplay is centered around interactable systems: players pick up ingredients, use stations like cookware and dispensers, and track progress through dynamic UI widgets.

From the beginning, I focused on modularity and network consistency. Each core mechanic—interacting, cooking, holding items—was built around a clean architecture using interfaces, components, and replicated actors. For instance, `UCookableComponent` handles the state and timing of the cooking process, while `ACookwareStation` manages ownership and triggers player montages for immersive feedback. Widgets like the progress bar are only visible to the interacting player and are updated client-side using replicated state and world-time functions.

During development, I prioritized replicating gameplay over visuals. Challenges included ensuring item mesh visibility on clients, syncing animations using Montages, and avoiding physics or collision issues on held actors. Debugging network behavior required a careful balance between server authority and local visual feedback.

I personally feel this prototype demonstrates strong architectural decisions and an understanding of networked gameplay. While some polish remains—such as animation-driven prop spawning and refining edge-case replication—it's a solid foundation for a scalable multiplayer game. I'm proud of the modularity, visual feedback systems, and overall synchronization achieved within the scope of the task.