# Project Requirements

You are required to develop a multiplayer online game similar to Diablo II within one month. The frontend must be developed using the Unity engine, while the backend technology is unrestricted. The game should support at least four players simultaneously and include a simple quest system. Additionally, the game must track players' progress and logout positions, allow players to log in as guests, and enable them to continue their game progress upon re-entering the game.

## Frontend Requirements

- Restrictions: Do not use third-party frameworks or plugins, except for the map editor.

- Engine: Develop the game using the Unity engine.

- Gameplay Mechanics: The game should include basic character movement (8 directions), attacking (no need for complex formulas), and interaction functionalities.

- User Interface (UI): Design a simple UI, including the main menu, in-game interface, and login screen.

## Backend Requirements

- Restrictions: Do not use third-party frameworks or plugins, except for Database and Cache systems.

- Technology Stack: Backend technology is unrestricted. You may choose from Node.js, Golang, Java, C#, etc.

- Multiplayer Functionality: Implement multiplayer connectivity supporting at least four players online simultaneously.

- Quest System: Design and implement a simple quest system, including quest acceptance, completion, and reward distribution.

- Player Data Storage: Implement player data storage functionalities, including game progress, character status, and logout positions.

- Guest Login Support: Allow players to log in as guests and restore their previous state upon re-entering the game.

## Functional Requirements

- Map: The map should include walkable and non-walkable areas, as well as interactive objects such as treasure chests.

- Multiplayer Gameplay: Enable real-time interactions between players, such as synchronized character movements.

- Quest System: Design at least one simple quest that rewards players upon completion (e.g., experience points, items).

- Data Saving: Automatically save the player's current progress and position upon logout, allowing them to continue the game upon re-login.

## Submission Contents

- Complete project code, including the frontend Unity project and backend code.

- Brief technical documentation.

## Evaluation Criteria

- Functionality Completeness: Whether all specified functional requirements have been implemented.

- Technical Implementation: Code quality, architectural design, and the rationale behind technology choices.

- User Experience: Smoothness of gameplay, UI friendliness, and overall experience.

- Multiplayer Interaction: Stability of connections and synchronization effectiveness in multiplayer gameplay.

- Creativity and Innovation: Any additional innovative features or designs beyond the specified requirements.

## Bonus Points

1. Explanation of system architecture, technology choices, and usage instructions.

2. Unit Tests.

3. Chat System.

4. Skill System.