**Scope Description for a Level 4-5 Developer**

**Project Title:** **Basic Text-Based RPG Game**

**Objective:**  
Develop a simple text-based RPG game where players can engage in combat with monsters, manage a basic inventory, and navigate through different game locations. The game should be easy to understand and interact with, providing immediate feedback based on player actions.

**Core Features:**

1. **Player Attributes:**
   * Track player stats such as experience points (xp), health, and gold.
   * Implement a simple system for increasing or decreasing these stats based on player actions.
2. **Weapons and Inventory:**
   * Create a basic inventory system where players can acquire a limited number of weapons.
   * Each weapon should have a name and a power level that affects combat outcomes.
   * Allow players to purchase new weapons or upgrade their current weapon using in-game gold.
3. **Monsters:**
   * Define a few types of monsters, each with basic attributes like name, level, and health.
   * Implement simple combat logic where the player can attack monsters and receive damage in return.
4. **Locations:**
   * Set up a few key locations (e.g., town square, store, cave) that the player can visit.
   * Each location should provide specific actions (e.g., buying items in the store, fighting monsters in the cave).
   * Implement a straightforward navigation system that allows players to move between these locations.
5. **Combat System:**
   * Implement a basic turn-based combat system where the player and the monster take turns attacking each other.
   * Use simple randomization to determine combat outcomes (e.g., whether an attack hits or misses).
   * Display the results of each combat round to the player.
6. **User Interface and Interaction:**
   * Use basic DOM manipulation to update the game interface based on player actions.
   * Implement buttons and event listeners to handle user inputs (e.g., attacking, buying items).
   * Provide clear, text-based feedback to the player for each action they take.
7. **Game States:**
   * Handle basic game states such as winning the game, losing the game, and restarting.
   * Ensure that when the game is lost, the player's stats are reset, and they can start again.

**Technical Requirements:**

* **JavaScript**: Use basic JavaScript (ES6) to handle game logic and DOM updates.
* **HTML/CSS**: Develop a simple web interface using basic HTML and CSS. The design should focus on functionality and clarity.
* **Simplicity**: The code should be straightforward, with functions broken down into manageable chunks. Focus on readability and ease of understanding.

**Guidance for Development:**

* **Modular Code**: Break down the code into small, reusable functions. This will help in understanding and debugging.
* **Progressive Development**: Start with core features (e.g., basic combat and navigation) before adding more complex elements like inventory management.
* **Testing**: Regularly test the game as you develop each feature. Ensure that player actions are reflected correctly in the game state.
* **Basic Error Handling**: Implement simple checks to prevent issues like purchasing items without enough gold or attacking when health is too low.

**Timeline:**  
Estimate 1 week for development, focusing on learning and implementing each feature step by step.

**Deliverables:**

* A working game with the core features implemented.
* Well-commented code to explain the purpose of each function and how different parts of the game interact.
* A basic README file explaining how to play the game and any known issues.

**Additional Notes:**

* The focus should be on building a functional game while practicing and reinforcing fundamental JavaScript concepts.
* Don’t worry about making the game too complex—keep it simple and manageable.
* Use online resources and tutorials to help with any concepts or features that are challenging.

This scope is designed to provide a clear, achievable project for a developer at a level 4-5 skill level, emphasizing fundamental programming practices and a straightforward approach to building a text-based RPG game.