**Player Manual: Text Adventure Game**

**Objective:**

Explore different locations, collect items, fight monsters, and discover the ancient book to defeat the final enemy and win the game!

**How to Play:**

* **Start**: The game begins in a forest clearing.
* **Navigation**: Use the direction buttons to move: Go North, Go South, etc.
* **Items**:
  + Pick up items by clicking **Take [item name]**.
  + Use or drop items from your inventory using **Use** and **Drop** buttons.
* **Combat**:
  + If there’s a monster, click **Fight** to attack.
* **Game Controls**:
  + **Look**: Repeats the room description.
  + **Inventory**: Shows what you're carrying.
  + **Quit**: Ends the game.
  + **Reset Game**: Restarts from the beginning.

**Locations:**

* **Clearing**: Starting point, has basic supplies.
* **Cave**: Home to a giant spider; contains weapons and a key.
* **River**: Has a locked path; unlock it with the **silver key**.
* **Tower**: Final location with the **shadow beast** and the **ancient book**.

**Strategy Tips:**

* Collect useful items before entering dangerous areas.
* Use the **silver key** to unlock the bridge to the tower.
* **Use** the **ancient book** to win the game if you're in the tower.

**What I Did – Explanation for Professor:**

I built a simple text-based adventure game with a frontend in **React** and a backend using **Python Flask**.

**Frontend (React):**

* File: App.js
* Displays the game UI with:
  + Room descriptions
  + Inventory and items
  + Action buttons (move, take, use, fight)
* It sends and receives data from the Flask backend through HTTP requests (/start, /command, /reset).
* Styles are customized in styles.css for a cleaner layout and visual feedback.

**Backend (Python Flask):**

* File: app.py
* Handles game logic:
  + Movement between rooms
  + Item interactions
  + Combat with enemies
  + Game state and reset
* Rooms are hardcoded with enemies, items, and locked paths.
* Maintains a player state with location, health, inventory, etc.

**Communication:**

* Frontend and backend talk via API:
  + GET /start – starts the game
  + POST /command – processes player commands
  + GET /reset – resets the game to its initial state