Automatic Testing - Team 21

Third party products used:

- Postman (Backend)
- Mocha (Backend)
- Chai (Backend)
- Jest (Frontend)

How is it implemented:

- Postman was used to perform high speed REST request to test our communication between the client and server.
- Mocha and Chai are used to run http request tests automatically
- For our backend automatic testing, a test.js file is in the same folder as the server.js file
- Jest is used to test frontend data and methods

System Coverage:

- Create Account
  - Invalid email
  - Already taken Username
- Login
  - Invalid username
  - Invalid password
- User Scores
  - User high score
- Change Username
  - Invalid username
  - Invalid password
- Change Password
  - Invalid username
  - Invalid password
- User Trophies
  - Trophy List entries
- Start new game

    - Creation of game entry
- Change game information
    - Normal flow of game logic leading to change in state of game object
- Ending the game
    - Confirming successful saving of scores/trophies and removal of game-related entries in the database
- Frontend Gameplay
    - Rooms will not reappear for a predefined number of subsequent rooms (must go through X different rooms before you have a chance of seeing the “Cave of the Troll” room again)
- Main Menu
    - New game and Continue game buttons correctly populate the game object

How to Run Tests:

Backend server

1. Navigate to the “backend” folder in your terminal
2. Run “npm install mocha”
3. Run “npm install chai”
4. Run “npm test”

Frontend

1. Navigate to “eaui” folder in your terminal
2. Run “npm install --save-dev jest”
3. Run “npm run test”

Needs to be tested Manually:

- Account Recovery
    - Make a user
    - Attempt to recover password using email
    - Temp password sent to the user’s email