Assignment-3

Convert assignment-2 to a multi-tiered client-server architecture.

1. **Convert the Static Web Game to a client-server application**.
   Take your previously built static web-based game and refactor it into a server-client model. You may use any language, server libraries for your backend.
2. **Present scores and leaderboard.** Every time a player starts a game, they enter a name (on the front end). Every time the game ends, the score is displayed and then saved. The player's scores are also shared against a leaderboard (top 10 scores, including any prior score by the player). This requires adding a database, use any database of your choice.
3. **Keep a clean architecture.** Use any frameworks as needed. Have guidelines on what code goes where. Keep minimal integration points, so testing is easier.
4. **Testing & deployment:** Refactor tests as needed, add new tests to anything you add. Add an integration test each to test client to server, and server to database integrations. Build up coverage, ensure it is 75%. All code linting should hold just as well. Adjust deployment scripts too as needed.
5. **Turn in an architecture & test pyramid diagram**. Add this to your GitHub codebase under a separate documentation folder, as a png or some such. It is okay if it is paper pencil drawn. It needs to be legible though!

What to turn in?

1. On HelloIITK, turn in a PDF (one per team).
2. The PDF should contain:
   a. Team mates names and roll numbers.
   b. Link to github repo.
   c. Link to the deployed version of the game.
   d. Link to the architecture diagram, and the test pyramid.
   e. A 1-2 liner on each of the above items to tell us where to find what. Think of this as documentation (OK if it is in README.md, in that case, specify so).

Due Date: 10th October 2024, 11:59pm.