Technical Specifications

Overview:

The web application is implemented using React.js (JSX), Material UI and CSS for the frontend, Java and SpringBoot for the backend, and MongoDB as the NoSQL database solution.

Client Architecture:

- The user interacts with the software via a web application.
- Initially, the home page is displayed, with a brief description of the game and software and buttons that allow users to choose which game they want to play.
- A navigation menu allows guest users to log-in to an existing account or sign-up and create a new account. The menu also allows signed-in users to sign-out of their account and view the leaderboard.
- When a user clicks the button to play either of the games, they are redirected to the game screens. On these screens they can play the timed game and create new high scores.
- When a signed-in user clicks the button to view the leaderboard, they can see their high-score as well as the top 10 high-scores for each game. A guest user will not be able to see the leaderboard or be added onto it.
- Material UI and CSS will be used to build components and maintain a cohesive design for the game.

Server Architecture:

- REST APIs will be implemented to handle all requests on our client-facing web application.
- NoSQL and Java will be used to run all CRUD operations on the game and user data.

Database Architecture:

- Two collections will be created in MongoDB: 'Player' and 'Leaderboard'.
- The 'Player' collection will store each player as a document. These will include a player's username, password, high scores for each of the four games, and cumulative score across games.
- The 'Leaderboard' collection will store each game's leaderboard as a document. These will include a tree map of the top 10 players, and will store their username and score to display on the leaderboard screen.

- A collection will not be used to store the questions and corresponding correct answers; these will be randomly generated on the client-side.

Functionality-Specific Technical Specifications:

Home Page:

- The home page is the first page that the user lands on when they access the webpage.
- The page allows the users to navigate through the application through a navigation bar on the top of the page.
- The navigation bar includes links to the login page, the leaderboard, and the games.
- The page also displays the rules for each game under the navigation bar.
- This page can be accessed by both registered and unregistered users with complete functionality.

Authentication:

- A new user can create an account with a unique username and password combination. This data will be stored in the database.
- To log-in to an existing account, the username and password combination inputted by a user on the client-side must match data extracted in the database.
- The username and password will both be stored as strings in the database.

Game Screen:

- There are four games on the website: Addition, Subtraction, Multiplication, and Division.
- For each game, when the start button is pressed, a timer of 60 seconds begins to decrement.
- While the timer is ticking down, a user will be answering a randomly generated set of math questions that are based on which game mode they chose on the home screen. For example, if the user chose the ‘addition’ game, they will be answering a question that tests their addition skills. For each question they answer correctly, their score will increment by one.
- Once a question is answered, a new question will be shown regardless of the previous answer being right or wrong.
- The game will automatically stop once the timer hits zero, and their final score will be shown based on how many questions they answered correctly.

- If their final score is higher than their previously recorded high score that is stored in a NoSQL database, it will update the player's new high score to be the final score.
- However, users who are not signed in will not be able to save their scores.

Leaderboard:

- A tally of the first 10 position holder's details will be displayed for each of the games that we are creating.
- The first 10 position holders will be decided based on how many questions are answered correctly in the 60 seconds given for the game.
- The display will include the position holder's username and score..
- An ArrayList of a custom class object will be created to store the score of users as the and the details of the position holder (username)
- Once the game is over, the leaderboard map will be updated if required. This can happen in two scenarios. First, when a user who isn't on the leaderboard achieves a score that qualifies to be in the top 10 or when an existing user on the leaderboard achieves a score that is better than his previous score and allows him to go higher on the leaderboard. These both will be checked and dealt with appropriately.
- If multiple users have the same high-score, then their names will be displayed in random order but if the score is included in the top 10, then each of the users will be included in the list. The leaderboard will display a maximum of 10 people.