

Leaderboard Task Explanation

Task Objective:

Create a system that allows a user to select one of ten users, claim random points for that selected user, and dynamically display user rankings based on total points. The backend will be developed using NodeJS, and the frontend will be built using ReactJS (Basic Ui).

Features Overview:

- **User Selection:** A list of 10 users will be displayed on the UI. and You can also add more users from frontend and that user will store in the database as well.
 - **Claim Button:** A button that, when clicked, awards random points (1 to 10) to the selected user.
 - **Database:** Points will be stored in the database, and each user's total points will be updated accordingly.
 - **Database (MongoDB):** There should be a Claim Points history collection.(On every claim it will create a history in the collection.
 - **Dynamic Ranking:** Rankings will be displayed in real-time, with the leaderboard adjusting based on users' total points.
-

Backend (NodeJS):

1. **User Collection:** Create a collection of 10 users in the database. Each user will have the following structure:
 - Example users:
 - Rahul
 - Kamal
 - Sanak
 - When a user clicks Claim Button the server will award random points (1 to 10) to the user with the provided userId and update the database.
 -
 - **Random Points Logic:** Ensure that the points are generated randomly between 1 to 10 when the claim button is pressed.
 - **Ranking Calculation:** Sort users by total points in descending order. Assign ranks based on this sorting.
 - **Real-Time Updates:** Implement real-time ranking updates to fetch updated rankings after each point claim.

Frontend (ReactJS):

1. **User List UI:**

- Create a dropdown or list component to display the 10 users. The user should be able to select any user from the list\
- User adding option..
- 2. **Claim Button:**
 - Add a "Claim" button next to the user selection list. When clicked:
 - The app should call the `claim point` API, passing the selected user's ID.
 - Display the assigned random points to the user in real-time.
- 3. **Leaderboard:**
 - Create a component to display user rankings dynamically.
 - The leaderboard should show each user's **name**, **total points**, and **rank**.
 - The leaderboard updates automatically when the user claims points and reflects the new ranking
 - There should be a collection of all user point history. (when ever user claim point it should generate a history Data in history collection)

Ui Example: [Click Here](#)



Time Limit



2 Days from the time the task is shared with you



Submission Guidelines



Fill out the Round 1 Task Submission Form

[\(TASK FORM LINK\)](#)



Bonus Points For:

- Clean and modern UI
- Responsive and optimized layout
- Efficient pagination logic
- Well-structured and reusable code
- Code comments and best practices

Need Help?

If you face any issues or have doubts, feel free to reach out via email hr@triplewsols.com.

!