LEADERBOARD SYSTEM – FULLSTACK TASK SUBMISSION

Presented by Raeen Fatima

INTRODUCTION

The Leaderboard System is a full-stack web application built using ReactJS (frontend) and Node.js with Express (backend), with MongoDB as the database. The system allows users to be added dynamically, claim points through randomized logic, and view live ranking updates on a real-time leaderboard.

This project simulates a competitive points-based ranking mechanism suitable for gamified platforms, employee rewards, hackathons, or any event that involves dynamic scoring. It emphasizes responsiveness, real-time data updates, user interaction, and clean UI/UX.

PROJECT OVERVIEW

1.

Frontend (ReactJS + TailwindCSS):
- Responsive, modern UI with podium-style leaderboard.
- Add users, claim random points, view rankings in real-time.
- Clean component structure using React hooks and animations (Framer Motion).

2.

Backend (Node.js + Express):
- REST APIs for user creation, point claiming, leaderboard fetching.
- Business logic for random point generation and rank calculation.
- Maintains claim history in a separate collection for traceability.

3.

Database (MongoDB + Mongoose):
- Stores user data (name, totalPoints).
- Maintains claim history with timestamps.
- Schema validation and efficient querying using Mongoose.

PROCESS

Planning & Requirements Analysis

1.
- Understood task scope and features from the problem statement.
- Defined schema structure for Users and ClaimHistory.

Backend Setup

2.
- Initialized Node.js with Express.
- Created API endpoints for:
  - Creating a user (POST /api/users)
  - Claiming points (POST /api/users/:id/claim)
  - Fetching leaderboard (GET /api/leaderboard)
  - Fetching claim history (GET /api/history)
- Integrated MongoDB using Mongoose with clean schema definitions.

Responsive UI and Styling

5.
- Applied clean UI with mobile-first design.
- Highlighted Top 3 users in a visually distinct podium section.
- Added toast notifications for user actions (success/error).

Frontend Development

3.
- Set up React project with TailwindCSS and Framer Motion.
- Built components:
  - AddUserForm, UserCard, ClaimModal, Leaderboard, Pagination, TopThreePodium.
- Handled API integration using axios.

Testing & Deployment

6.
- Tested all features manually (add user, claim points, check ranks)
- Deployed:
  - Frontend on Vercel
  - Backend on Render
  - MongoDB on MongoDB Atlas

Real-Time Updates & Sorting Logic

4.
- Implemented leaderboard sort logic based on totalPoints.
- Used slice/sort/map patterns to ensure proper ranking and pagination.

RESULT

- A fully functional Fullstack Leaderboard System was successfully developed within the deadline.
- Users can be added dynamically, and points can be claimed with real-time leaderboard updates.
- Top 3 users are highlighted in a responsive podium-style UI, with ranks and points auto-updating.
- A separate Claim History collection is maintained in MongoDB for tracking user actions.
- The application was deployed:
- 🔗 Frontend on [Vercel](#)
- 🚀 Backend on [Render](#)
- 🗄 Database on MongoDB Atlas (secured)
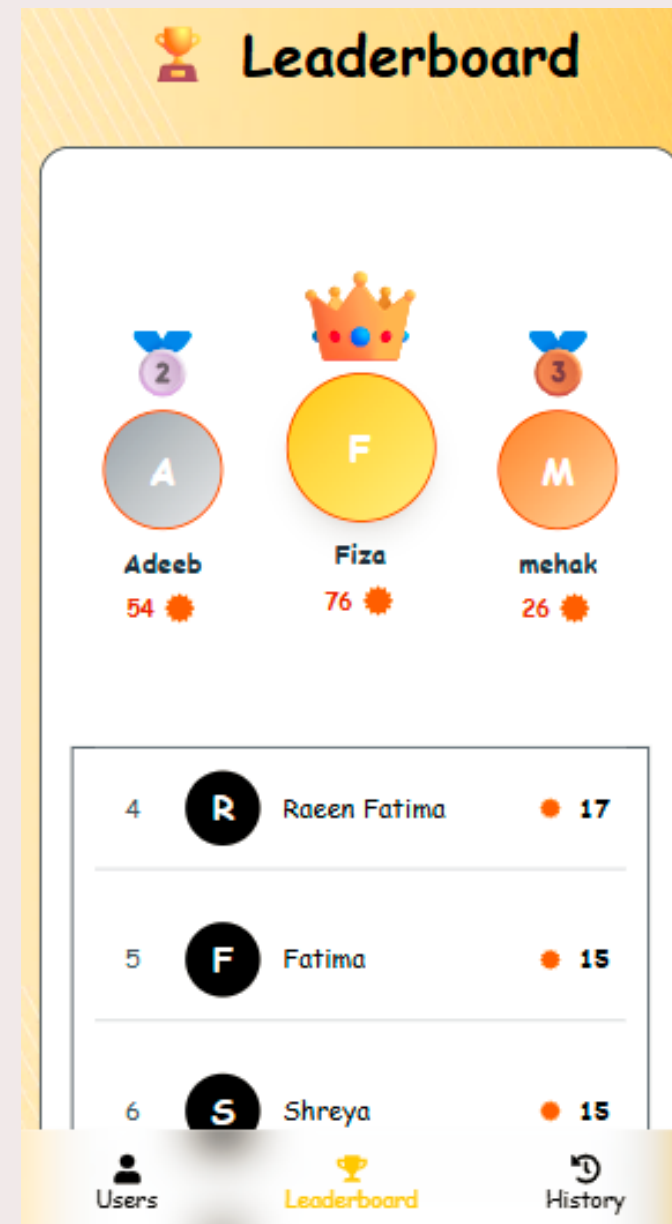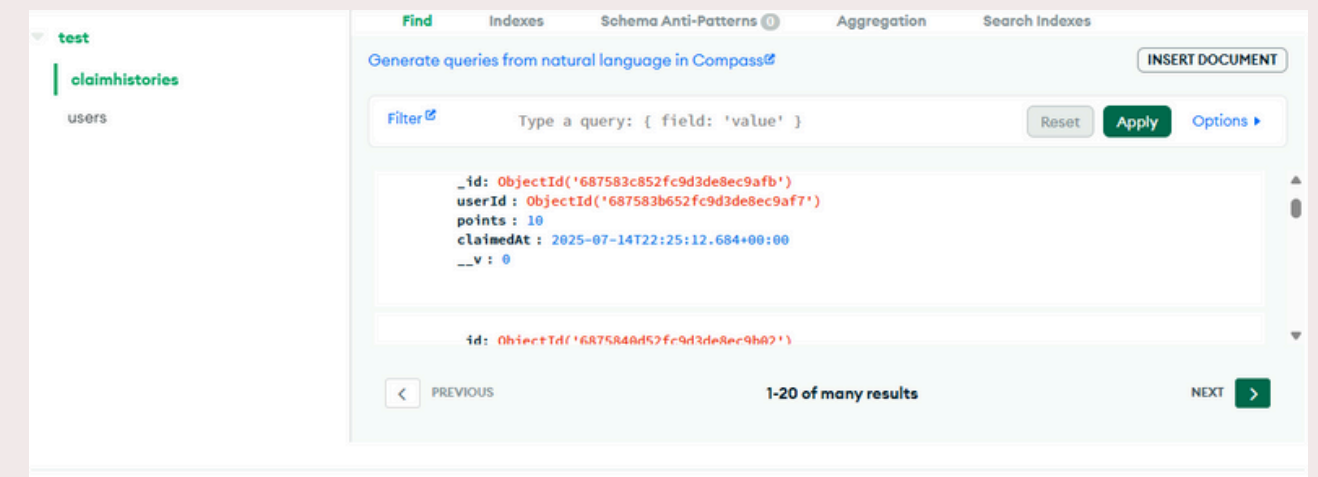- 🔗GitHub [Repository](#)
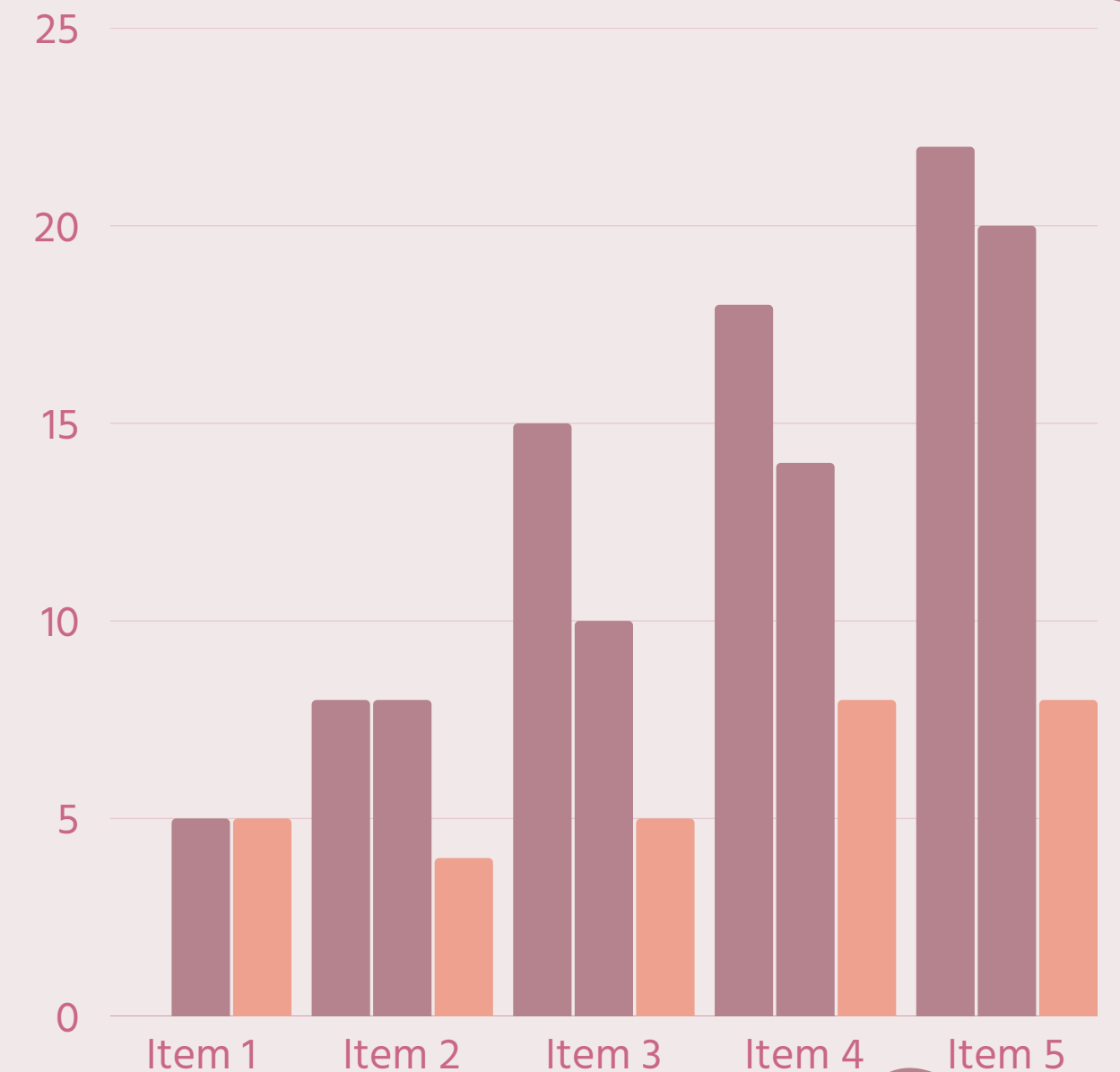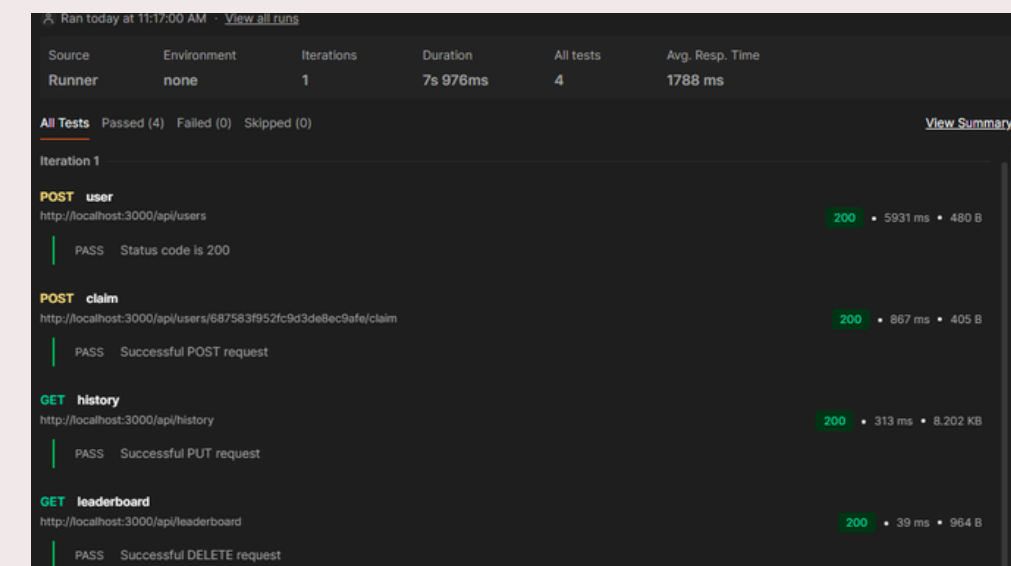
FILE STRUCTURE
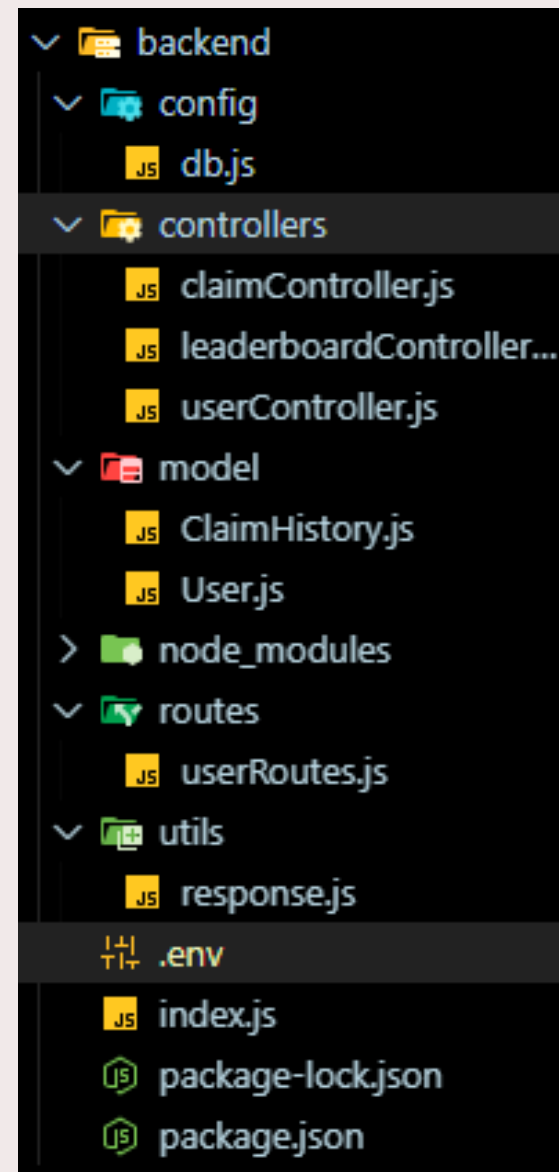
Backend
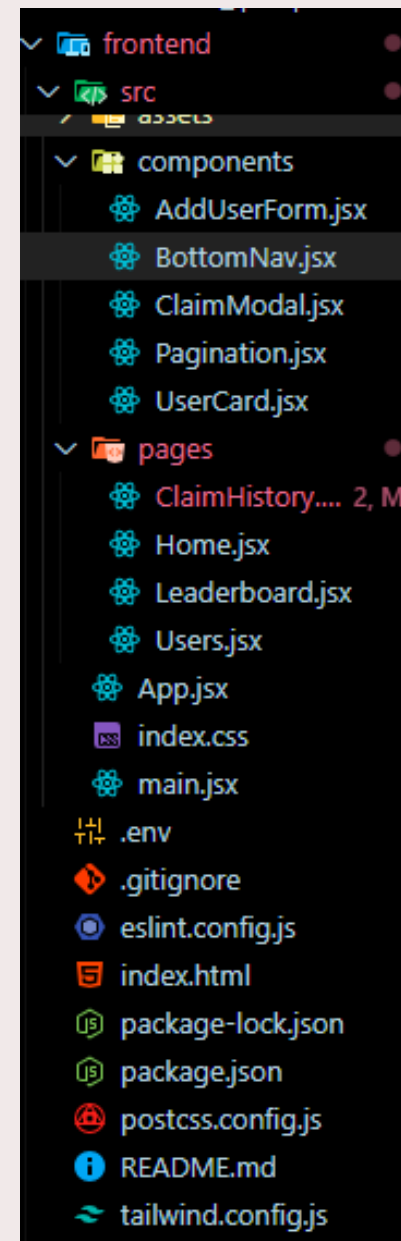
Frontend

UI view

Database

API testing through postman

CONCLUSION

The Leaderboard System project demonstrates the successful implementation of a full-stack application using modern web technologies. With ReactJS on the frontend, Node.js and Express on the backend, and MongoDB for data storage, the system delivers a seamless experience for managing users, tracking claim history, and displaying dynamic rankings in real-time.

This system is scalable, maintainable, and built with best practices, making it a strong foundation for any gamified or points-based user engagement application.

THANK YOU

GMAIL   raeenfatimahere@gmail.com

FRONTEND URL   Leaderboard System

GITHUB REPO   GitHub Repo

by

RAEEN FATIMA