High-Level Design: Online Judge System (MERN Stack)

H.SaiSandeep

1 System Overview

The Online Judge is a web platform that hosts coding challenges, manages competitions, and automatically evaluates submitted solutions. It is built entirely using the MERN stack:

- **MongoDB:** Database for storing problems, solutions, test cases, and user data.
- **Express.js:** Server-side framework for API routes and backend logic.
- **React.js:** Frontend library for building the user interface.
- **Node.js:** JavaScript runtime environment for executing the backend server.

2 Architecture Diagram

3 Component Breakdown

3.1 Frontend (React.js)

- Provides an interactive web interface for participants.
- Screens:
  - **Home Screen:** Problem list, login/signup.
  - **Problem Screen:** Problem details, code editor, language selection, submit button, verdict display.
  - **Profile:** User info and submission history.
  - **Leaderboard (Optional):** Ranking of participants.
- Uses REST API calls to communicate with the backend.

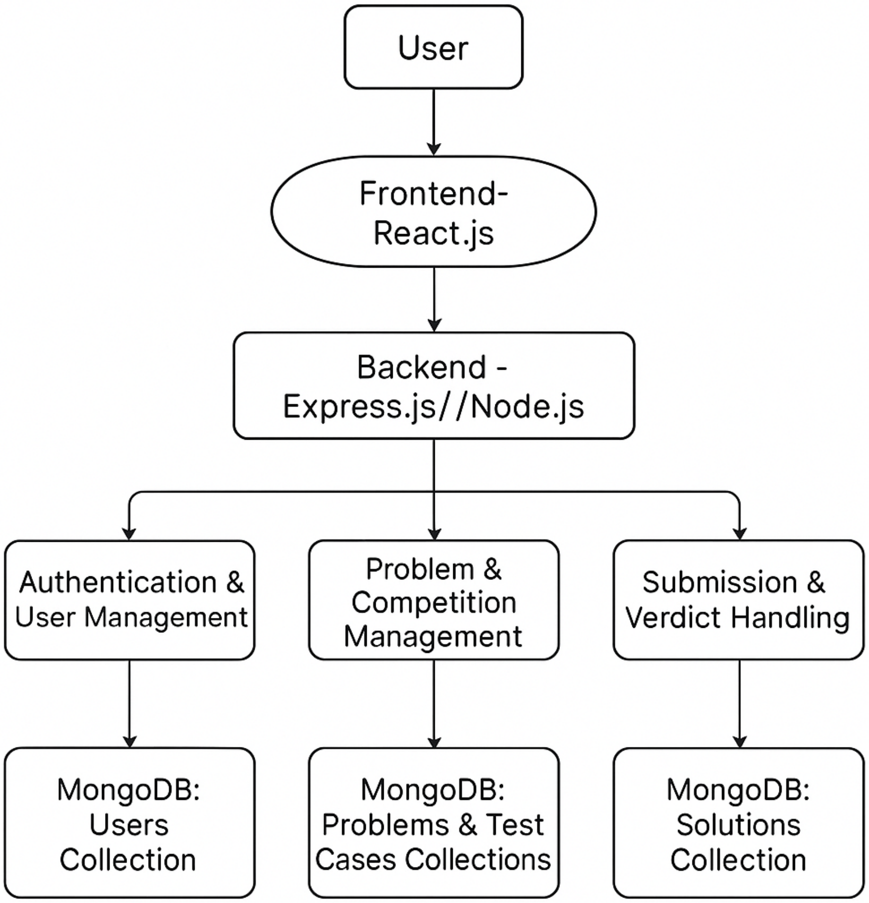
User
Frontend-React.js
Backend - Express.js//Node.js
Authentication & User Management
Problem & Competition Management
Submission & Verdict Handling
MongoDB: Users Collection
MongoDB: Problems & Test Cases Collections
MongoDB: Solutions Collection

Figure 1: Online Judge System High-Level Architecture (MERN Stack)

3.2 Backend (Express.js + Node.js)

- Handles RESTful API endpoints for:
  - User registration and login.
  - Fetching problem lists and individual problem details.
  - Receiving and evaluating code submissions.
  - Returning submission verdicts and leaderboard data.
- Evaluates code submissions on the server side by:
  - Retrieving relevant test cases from MongoDB.
  - Running submitted code using system commands (e.g., `child_process` in Node.js).

  - Comparing output with expected results.
  - Storing the verdict and returning it to the user.

- Handles authentication, session management, and access control.

3.3 Database (MongoDB)

- **Technology Stack:** MongoDB
- **Purpose:** Stores all persistent data for the Online Judge system.
- **Collections:**
  - `problems`:
    - `statement`: string
    - `name`: string
    - `code`: string
    - `difficulty`: string (optional)
  - `solutions`:
    - `problem`: reference to a problem
    - `verdict`: string
    - `submitted_at`: datetime
  - `test_cases`:
    - `input`: string
    - `output`: string
    - `problem`: reference to a problem
  - `users`:
    - `UserId`: string
    - `Password`: string
    - `Email`: string
    - `DOB`: date
    - `FullName`: string

4 Authentication

- Users register with email and password.
- Backend verifies credentials and issues authentication tokens (e.g., JWT).
- Tokens are required for secure API requests.

5 Code Evaluation

- Code is executed by the Node.js server using system calls.
- Outputs are compared with expected outputs stored in MongoDB.
- Verdicts (Accepted, Wrong Answer, etc.) are returned to the user and saved.

6 Key Features

- **Practice Problems:** Accessible anytime for self-paced practice.
- **Leaderboard:** Ranks participants by performance.
- **Profile:** Tracks individual progress and submission history.