**High-Level Design Document: Online Judge Platform (MERN + Docker)**

*1. Overview*

The Online Judge Platform is a web-based application where users can solve coding problems, run or submit code in multiple programming languages, and get automated verdicts. It supports secure and isolated code execution using Docker containers, a role-based authentication system, and AI-enhanced features such as code review and alternative suggestions.

*2. Tech Stack*

|  |  |
| --- | --- |
| Frontend | React.js, Tailwind css |
| Backend | Node.js, Express.js |
| Database | MongoDB |
| Authentication | JWT(admin/user) |
| Code Execution | Docker for isolation |
| AI Integration | GPT based APIs |

*3. Database Design*

a. Users Collection:- id, name, email, password, role(user/admin).

b. Problems Collection:- id, title, company tags, level of difficulty, topics, Description, constraints, hints, sample inputs, sample outputs.

c. Submission Collection:-id, u\_id, p\_id, code, language used, verdict, submission time, runtime, memory used.

d. Testcases Collection:-id, p\_id, input, output.

*4. Backend (Restful APIs)*

a. Authentication

|  |  |  |
| --- | --- | --- |
| **Method** | **Route** | **Description** |
| POST | /api/auth/register | Register new user/admin |
| POST | /api/auth/login | Authenticate and return JWT |
| POST | /api/auth/logout | Logout and clear session |

b. User Profile

|  |  |  |
| --- | --- | --- |
| **Method** | **Route** | **Description** |
| GET | /api/user/profile/:id | View user profile info and stats |

c. Problems

|  |  |  |
| --- | --- | --- |
| **Method** | **Route** | **Description** |
| GET | /api/problems | Get list of all problems |
| GET | /api/problems/:id | Get details of a single problem |
| POST | /api/problems | Admin only: Add a new problem |
| PUT | /api/problems/:id | Admin only: Modify problem |
| DELETE | /api/problems/:id | Admin only: Delete problem |

d. Submisssions

|  |  |  |
| --- | --- | --- |
| **Method** | **Route** | **Description** |
| POST | /api/submissions/run | Run code with sample inputs |
| POST | /api/submissions/submit | Submit code for full evaluation |
| GET | /api/submissions/user/:uid | Get all submissions of a user |
| GET | /api/submissions/:id | View a specific submission |

*5. Frontend (UI Screens)*

a. Authentication Screens

* Register: Form for name, email, password, role.
* Login: Secure login using email and password.
* Authenticated routes redirect based on role

b. Home Page

* General overview
* Navigation to Problems, Submissions, Profile, Admin (if role=admin)

c.Problem List Page

* Displays list of problems
* Filters by difficulty, tags

d. Problem Page

* Full problem details: title, description, constraints, hints
* Code editor with language selector
* Buttons: Run, Submit

e. Submissions Page

* View previous submissions with verdicts, language, timestamps

f. Profile Page

* User info, problems solved

g. Admin Panel

* Add/Edit/Delete problems
* Add/edit test cases
* Only accessible to admin role via protected routes

*6. Docker Usage*

Docker is used to securely execute user-submitted code inside isolated containers. This ensures safety, scalability, and control over resource usage.

* Prevents code from accessing system resources
* Handles multi-language support in isolated environments
* Ensures fair and safe evaluation

*7. AI Enhancements*

AI Code Review, Alternate Method Suggestions, Time & Space Complexity Estimation.