HLD for Online Judge

BY: Pawni Yadav

PURPOSE

An **online judge (OJ)** is a web-based platform used primarily for practising data structures and competitive programming.
It'll allow the user to:

- **Submit code** solutions to programming problems in **various languages** (6+)
- Participate, host **contests** and view their rankings on the **leaderboard**
- Send a request to the admin for **problem/question addition**
- Track their submission history and streak
- Maintaining a **to-do list** for task tracking (e.g., pending, in progress, solved)
- Support a **discussion forum** for the entire OJ

TECH STACK

- Frontend: React & Tailwind CSS, HTML(till implementation of backend)
- Backend: Django
- Database: SQLite (Later PostgreSQL or MongoDB)
- Containerization: Docker
- AI: Deepseek API
- Hosting: AWS EC2/ECR

EXECUTION FLOW

1. User lands on the login/register (homepage)
2. If the user is found in the database, authenticate
   Else prompt signup
3. The user lands on the dashboard.
4. Navbar has a link to the discussion forum, Contest (with a leaderboard with global ranking), a timer(planned), a debug zone (planned) and a link to the profile page.
5. The Dashboard has a list of problems. Green problems have been solved.
6. User clicks the problem and lands on that specific problem page
7. The user writes code and clicks Submit.
8. Frontend sends /submit with code, language, and problem ID.
9. The backend stores the submission.
10. Backend calls the code execution service.
11. The service runs the code in a Docker container and captures output.
12. Update submission with result and status. The verdict is returned and shown to the user

FRONTEND

1. Homepage/landing page

URL: /
Visible to: All (unauthenticated and authenticated users)

Features:

- App name/Logo
- Welcome message
- “Login” and “Register” buttons
- If logged in: “Go to Dashboard” button

Fields:

- Navigation bar with links (Dashboard, contests, discussion, etc.)

2. Register Page

URL: /register
Visible To: Unauthenticated users

Features:

- ‘Register’ button
- Redirect to login on success
- Messages for success/failure, existing email and/or weak password

Fields:

- Username
- Email
- Password
    - Confirm password

3. Login Page

URL: /login
Visible To: Unauthenticated users

Fields:

- Email / Username
- Password

Features:

- Login button
- Link to "Register"
- Redirect to ‘homepage’
- Error handling for incorrect credentials

4. **Problem Dashboard (Problems List)**

URL: /problems
Visible To: Authenticated users

Features:

- List of all problems (title, difficulty, solved/unsolved)
- Search bar
- Filters (difficulty, tags, solved/unsolved)
- Click on the problem to go to the details page
- NavBar (Dashboard, contests(planned), discussion, add a problem, profile (view profile, logout), debug zone(planned), timer)

5. Problem Detail Page

URL: /problems/<int:pid>
Visible To: Authenticated users

Features and fields:

- Problem Title
- Full Description with supported media (images)
- Constraints
- Sample Input/Output
- Time & Memory Limit

Code Editor:

- Code editor
- Language dropdown (C++, Python, Java, etc.)
- Submit Button
- Run button
- Button to view all submissions
- Verdict display

6. Submissions page

URL: /submissions
Visible To: Authenticated users

Features:

- Table of past submissions: problem name, status, language, execution time, date
- Filters (status, language, date)
- Click to view detailed result

7. Submission detail page

URL: /submissions/<int:id>
Visible To: Authenticated users

Shows:

- Problem Title
- Submitted Code (read-only)
- Language
- AI evaluation (planned)
- Execution Result:
    - Status (Success/Fail/Time Limit Exceeded)
    - Output
    - Error message (if any)
    - Time taken
- Option to "Retry the problem" (redirects to problem page of this pid)

8. Add a Problem Page

URL: /addprob
Visible To: Authenticated admin users
Features/Sections/Fields:

- Problem form (title, statement, difficulty, testcases)
- “Suggest” button sends the suggestion to the admin
- Automatically assigns the current user as the author

9. Profile Page

URL: /profile/<int:uid>
Visible To: Authenticated user
Features:

- Username, email, join date
- List of problems contributed
- List of problems solved
- “update profile” option (if uid==current uid)
- To-do list option (planned)
- Streak board (planned)

10. Discuss Page (planned)

URL: /discuss/
Visible To: All users
Features:

- Forum for Q&A
- “Ask a question” button
- Form for questions: Subject, Tag (create new tag, etc), description
- Threads per problem or tag

Planned Pages:

1. Contest page with leader board and upcoming contests
2. Debug zone (planned)(for practicing debugging skills)

BACKEND

User

- user_exists() – if the user already exists in the DB
- register_user() – email-based registration(new addition).
- login_user() – Authenticates with username or email and password.
- logout_user() – Logs out.

Problem

- problem_list() – Displays all problems.
- problem_display(pid) – Fetches the problem and shows its details.
- add_problem() – Authenticated users add a new problem.
- delete_problem(pid) – Authenticated authors can delete their own problems

Test Case

- add_testcase(pid) – Adds custom test cases to a specific problem.
- testcase_list(pid) – Lists all test cases for a problem.
- update_testcase(pid, cid) – Edit a specific test case. (admin only)
- delete_testcase(pid, cid) – Deletes a specific test case.(admin only)

Submission

- add_solution(pid) – Accepts user submissions.
- solution_list(pid) – Shows all submissions for a problem.

Database schema

1. Users (user_id, username, email, password_hashed, join_date)

2. Problems (problem_id, title, description, media, difficulty, tags, author(fk))

3. TestCases (testcase_id, problem_id (fk), input, expected_output,author(fk), is_hidden)

4. Submissions (submission_id, user_id(fk), problem_id(fk), language, code, result, time_used, memory_used, timestamp)

AI Integration (Planned)

- Feedback on all types of solutions
- Hint generation for a problem
- Ask follow-up questions (special button for this on problem details page)
- Used in the Debug zone to generate bug-ridden code

DEPLOYMENT

Using AWS EC2/ECR