High-Level Design (HLD): Online Judge Platform

Devanand P

---

1. Purpose

The Online Judge platform is a full-stack web application designed for users to solve programming problems, track their progress, and participate in competitive contests. It supports code submissions in multiple languages, automatic evaluation via secure containers, and interactive user experiences such as leaderboards, discussion forums, and AI-powered assistance.

---

2. Key Features

- Submit code in 6+ programming languages
- Host or join coding contests with real-time leaderboards
- View and manage submission history and verdicts
- Maintain a personal to-do list for problem-solving
- Admin portal for adding and reviewing new problems
- Integrated discussion forums for collaborative learning
- AI assistance for debugging, hints, and solution evaluation *(planned)*

3. Technology Stack

Frontend: React.js, Tailwind CSS, Vite

Backend: Django, Django REST Framework

Database: SQLite (Dev), PostgreSQL (Prod)

Code Execution: Docker (Language-specific images)

AI Assistance: OpenAI / DeepSeek APIs (planned)

Deployment: AWS EC2 with Docker Compose

---

4. Execution Flow

1. User visits homepage (Login/Register)
2. Authenticated user is directed to Dashboard
3. Dashboard shows problem set with filters
4. User selects a problem, writes code, submits
5. Submission sent via API with code, language, and problem ID
6. Backend saves submission, triggers Celery task
7. Code is executed in a Docker container
8. Verdict is computed and updated in database
9. Verdict is displayed to the user

5. UI Pages (Frontend)

1. Landing Page:

- Buttons to Login/Register
- App overview

2. Login/Register:

- Authentication with validation
- JWT token stored for session

3. Dashboard:

- List of problems
- Filters: Difficulty, Tags, Status
- Color indicators for solved/unsolved

4. Problem Page:

- Full description, sample I/O, constraints
- Online code editor with language dropdown
- Submit and Run buttons
- Verdict display on submission

5. Submissions Page:

- History of attempts
- Columns: Problem, Status, Time, Language
- Click to view code and details

6. Profile Page:

- Username, email, join date
- Problems solved/contributed
- To-do list, streak tracker *(optional)*

7. Add Problem Page (Admin):

- Form to create a problem
- Test cases (visible and hidden)
- Difficulty tags

8. Discussion Forum (Planned):

- Post questions, reply threads
- Tags, upvotes, and moderation

9. Contest Page (Planned):

- Live contests with timers
- Leaderboard rankings

---

6. <u>Code Execution System</u>

- Secure Docker containers per language
- Memory, CPU, and time constraints applied
- Network disabled for security
- Results parsed and returned via Celery
- Verdict options: AC, WA, TLE, RE, CE

---

7. <u>Unique Enhancements (Planned)</u>

- **Tab Switch Detection** during contests
- **AI Debugger** using LLMs to give hints and explanations
- **Auto-grading support** for classroom setups
- **Gamification**: Badges, streaks, points system

8. Security Measures

- JWT Auth and CSRF protection
- Docker sandboxing to prevent malicious code
- Rate-limiting on submissions
- Input sanitization and validation
- Admin-only routes protected by role checks

---

9. Conclusion

This high-level design lays out the technical and functional foundation of an Online Judge platform suited for educational and competitive use cases. It emphasizes security, modularity, and user engagement while allowing for scalability and future enhancements like AI debugging and real-time contest monitoring.