

# **High-Level Design (HLD): Online Judge Platform**

**Devanand P**

---

## **1. Purpose**

The Online Judge platform is a full-stack web application where users can solve programming problems, submit their code, view verdicts, and track their problem-solving progress. It supports code execution in multiple languages within secure Docker containers and includes basic competitive coding features like a leaderboard, AI hint generation and hidden test case evaluation.

---

## **2. Key Features**

- Submit code in Python, C, C++
- AI-based hint generator (Gemini API integration)
- Hidden test case evaluation for fair judging
- Leaderboard based on number of problems solved
- Admin can add/edit problems with difficulty tags
- Submission verdicts displayed in real-time
- Containerized using Docker
- Deployed on AWS EC2 with Docker Compose

### 3. Technology Stack

#### Frontend:

- HTML, CSS
- JavaScript

#### Backend:

- Django
- Django REST Framework

#### Database:

- SQLite (Development)

#### Deployment:

- AWS EC2
  - Docker
- 

### 4. Execution Flow

1. User logs in or registers
2. User sees the list of available problems on the dashboard
3. User selects a problem → redirected to problem page
4. User writes and runs/submits code in the online editor
5. Custom inputs are processed and passed into the Docker container
6. For **Run**, output is shown; for **Submit**, hidden test cases are evaluated
7. Verdict is saved and displayed; leaderboard updated if solved
8. Submission history is stored in the database

## **5. UI Pages (Frontend)**

### **Landing Page:**

- Login/Register buttons

### **Dashboard:**

- List of all problems
- Solved problems shown with green tick

### **Problem Page:**

- Title, description, sample I/O, constraints
- Code editor with language dropdown
- Run and Submit buttons
- Output window for verdict or errors

### **Submissions Page:**

- View all submissions for a given problem
- Status, timestamp, and language shown

### **Leaderboard Page:**

- Ranks users by problems solved

### **Admin Add Problem Page:**

- Create new problems
  - Add sample and hidden test cases
  - Add constraints and description
- 

## **6. Code Execution System**






Each submission runs inside a **Docker container** (Python/C/C++)

Memory and time limits applied

Custom inputs passed via `subprocess.run()`

No network access (for security)

Verdicts returned:

-  Accepted (AC)
  -  Wrong Answer (WA)
  -  Time Limit Exceeded (TLE)
  -  Runtime/Error (RE)
  -  Compilation Error (CE)
-

## 7. Security Measures

- CSRF protection (via Django)
  - Authenticated views (`@login_required`)
  - Docker-based execution sandboxing
  - Timeout protection for code execution
  - Users can't view others' hidden test cases or submissions
- 

## 8. Deployment & DevOps

- Dockerfile and `docker-compose.yml` used to containerize app
  - Deployment to AWS EC2 Ubuntu instance
  - Docker image pulled from private AWS ECR repository
  - Environment variables stored securely in `.env` (ignored by Git)
  - Sensitive files (`keys/`, `.env`, `.pem`) ignored in `.gitignore`
- 

## 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.