High-Level Design (HLD) - Online Judge System

Prepared By: S Naveen

GitHub updates:- https\://github.com/Sajjan-Naveen-87/Online-Judge.git

---

Purpose

The Online Judge is a web-based platform intended to:

- Enable solving and submission of coding problems
- Host contests and leaderboards
- Allow problem contribution by admin or group
- Monitor user progress and submission history
- Enable group-based collaborative problem creation
- Support email-based user verification
- Foster community-driven preparation (e.g., FAANG)

Technologies

Component	Technology Used
Backend	Django
Frontend	HTML, CSS, JavaScript , (Bootstrap CSS)
Database	SQLite (extensible to MySQL)
Containerization	Docker (Planned)
AI Integration	Smart feedback & complexity analysis
Hosting	AWS EC2

Frontend Pages Overview

Page	URL	Access
Dashboard (Home)	`/`	All users
Problem List	`/problist/`	All users
Problem Details	`/probdisp/<int:pk>`	All users
Add/Edit Problem	`/addprob/`, `/update/<int:pk>`	Admin/Group
Test Case List	`/testcaselist/<int:pk>`	Admin/Group

Page	URL	Access
Add/Edit Test Case	`/addtestcase/`, `/updatetestcase/`	Admin/Group
Solutions List	`/solutionlist/<int:pid>`	Authenticated
Register	`/register/`	Public
Login	`/login/`	Public
Profile (Planned)	`/profile/`	Authenticated
Explore (Planned)	`/explore/`	All users
Contest (Planned)	`/contest/`	All users
Discuss (Planned)	`/discuss/`	All users
Leaderboard (Planned)	`/leaderboard/`	All users
Group Dashboard	`/group/`	Members/Admin

Core Backend Functionalities

User Management

- `register_user()` with email verification
- `verify_email()` via token link
- `login_user()` only for verified users
- `logout_user()`

Problem Management

- `problist()`
- `probdisp(pk)`
- `add_problem()` (admin/group)
- `update_problem(pk)`
- `delete_problem(pk)`
- `upvote_problem(pk)`

Test Case Management

- `add_testcase(pk)`
- `testcase_list(pk)`
- `update_testcase(pid, cid)`
- `delete_testcase(pid, cid)`

Solution Management

- `add_solution(pid)` (includes Docker logic - planned)
- `solution_list(pid)`
- Analyze and store time/space complexity

Group Feature

- `create_group()`
- `add_group_member()`
- `create_group_problem()`
- `group_problem_list()`
- `submit_group_solution()`
- `set_group_privacy()`
- `view_public_groups()`

Database Schema (Simplified)

users

- id, username, email, password_hash
- date_joined, is_admin, is_verified, verification_token

problems

- id, name, statement, difficulty
- written_by (FK), group_id (nullable), upvotes

test_cases

- id, problem_id, input, output, written_by

solutions

- id, problem_id, code, verdict
- time_complexity, space_complexity, written_by, submitted_at

groups

- id, name, created_by, is_public

group_members

- id, group_id, user_id

Evaluation Workflow (Planned)

1. User submits code
2. Backend queues job for Docker or subprocess
3. Code is executed securely
4. Output is compared with test cases
5. Verdict + Time/Space complexity are returned
6. Verdict is saved and shown

AI Integration (Planned)

- Smart feedback on incorrect solutions
- Hint generation for users
- Estimate and display Time and Space Complexity

Hosting & Deployment

- Hosted on AWS EC2
- Scalable deployment using Docker only

End of HLD Document