# High-Level Design (HLD) - Online Judge System

**Prepared By:** S Naveen

## 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 |
| 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**