

# High-Level Design Document: Online Judge (OJ)

## Problem Statement

A coding challenge is a competitive event where participants solve problems within a set timeframe. The Online Judge (OJ) platform provides the infrastructure to host these challenges, collect and evaluate submissions, assign verdicts, and maintain leaderboards. Examples include Codeforces, CodeChef, and LeetCode.

## Project Overview

Design and develop a full-stack Online Judge using the MERN Stack. The platform will accept code submissions from users, run the code in isolated environments, evaluate them against test cases, and return verdicts like 'Accepted', 'Wrong Answer', or 'Runtime Error'.

## Key Features

- User Registration & Authentication
- Code Submission Interface
- Practice & Contest Problem Sets
- Leaderboards
- Submission Logs
- Docker-based Sandbox Execution
- Plagiarism Detection (MOSS)
- Intelligent Test Case Generator (NEW)
- Auto Backup & Crash Recovery (NEW)
- Multi-language Support for Code (C++, Python, Java, etc.) (NEW)

## Challenges and Solutions

1. Thundering herd (simultaneous submissions) -> Use Message Queues (e.g., RabbitMQ).
2. Malicious code -> Use Docker containers for sandboxing and memory/cpu limits.
3. Verdict tampering -> Secure API using JWT, server access logs, and RBAC.

## High-Level System Architecture

Frontend (React.js + Tailwind CSS):

- Home Page
- Problem List
- Code Editor with Language Selector
- Submission Logs & Verdict
- Leaderboard Page

Backend (Node.js + Express.js):

- REST API for Problems, Submissions, Leaderboard
- Middleware for auth and role-based access

Database (MongoDB):

- Users, Problems, Solutions, Test Cases collections

## **Evaluation Engine**

- Uses Docker for safe and isolated code execution
- Language-based compilers inside containers
- Child Process / Worker Threads to handle execution
- Resource and timeout limits enforced per container

## **Flow of Code Submission**

1. User submits code -> API call
2. Code is added to a queue
3. Evaluator picks the job, compiles & executes in a Docker container
4. Output is compared with expected output
5. Verdict is returned to frontend & saved to DB

## **Leaderboard Design**

- Sorted by score and submission time
- Filtered by problem or contest
- Endpoint returns last 10 submissions for quick access

## **New Enhancements**

- MOSS-based Plagiarism Detection for fairness
- Smart Test Case Generator using AI models
- Support for C++, Java, Python, JS (multi-language engine)
- Auto-Backup & Recovery for submissions
- Admin Dashboard for managing contests & reviewing logs