1️⃣ Problem Structure & Submission Process

Each problem should have:

✅ **Problem Statement** – Clear description, constraints, and input/output format.
✅ **Difficulty Level** – Easy, Medium, Hard (assign points accordingly).
✅ **Test Cases** – Public (visible to users) + Private (hidden for evaluation).
✅ **Time & Space Constraints** – To prevent inefficient solutions.
✅ **Allowed Programming Languages** – Java, Python, C++, JavaScript, etc.

💡 **Submission Process:**

1️⃣ User submits their code.
2️⃣ The system compiles and executes it inside a **secure Docker container**.
3️⃣ Code is tested against multiple **test cases**.
4️⃣ Result is generated – Accepted, Wrong Answer, Time Limit Exceeded, Runtime Error, etc.
5️⃣ Score is assigned based on performance.

2️⃣ Judging Criteria & Marking Scheme

Criteria	Points Allocation	Notes
Correctness	✅ Full points if all test cases pass. ✅ Partial points if some pass.	- Key factor in evaluation.
Efficiency (Time Complexity)	✅ Bonus points for optimal solutions. ❌ Penalty for exceeding time limits.	- Helps prevent brute-force solutions.
Memory Usage	✅ Lower memory solutions get bonus.	- Important for large input cases.
Edge Case Handling	✅ Bonus for handling tricky cases. ❌ Penalty for failing extreme test cases.	- Tests robustness.
Code Style & Readability (Optional)	✅ Bonus for clean, readable code.	- Encourages good coding practices.

3️⃣ Test Case Weightage & Partial Marking

Each problem should have **multiple test cases** categorized into:

- **Basic Cases (20%)** – Simple inputs to check basic logic.
- **Intermediate Cases (40%)** – Regular scenarios.
- **Edge Cases (40%)** – Large inputs, boundary conditions, tricky logic.

Marking System

- **All test cases pass** → Full score (e.g., 100 points).

- **Partial test cases pass** → Proportional score (e.g., if 6/10 pass, user gets 60%).
- **Fails all test cases** → Zero score.

4️⃣ Leaderboard System & Ranking

Scoring Based on Difficulty

- **Easy problems** → 50 points
- **Medium problems** → 100 points
- **Hard problems** → 200 points

Ranking Criteria:
✅ **Total points scored** in a contest.
✅ **Submission time** (Tie-breaker: Faster submissions rank higher).
✅ **Number of attempts** (More attempts → Lower ranking).

Example:

User	Total Score	Time Taken	Attempts	Rank
A	350	1h 30m	3	🥇 1st
B	350	1h 45m	2	🥈 2nd
C	320	1h 20m	1	🥉 3rd

5️⃣ Cheating Prevention (Anti-Plagiarism System)

✅ **Code Similarity Checker** – Use **MOSS (Measure of Software Similarity)** or **JPlag** to detect copied code.
✅ **IP & Session Tracking** – Prevent multiple accounts from submitting the same solution.
✅ **Randomized Test Cases** – Different test sets per user.
✅ **Time-based Submission Monitoring** – Too many submissions in a short time raise flags.

6️⃣ Bonus Features (Advanced)

🚀 **Dynamic Difficulty Adjustment** – AI-based analysis to update problem difficulty.
🚀 **Live Editorials & Hints** – Users unlock hints based on points deducted.
🚀 **Adaptive Test Cases** – Generate random edge cases to challenge users.

💡 Next Steps

1. **Implement Code Execution System** using **Docker**.
2. **Develop Leaderboard & Ranking Algorithm**.
3. **Setup MOSS/JPlag for Plagiarism Detection**.
4. **Design Submission UI in React** with **Monaco Editor**.