🔧 How to Execute Code Submissions?

You have two main options:

1️⃣ Using a Secure Docker-Based Execution System (Recommended)

You can create a **Dockerized execution system** that runs user-submitted code inside **isolated containers**. Here's how it works:

✅ **Step 1: User submits code** (Python, Java, C++, etc.).
✅ **Step 2: Backend (Spring Boot) spins up a Docker container** with the required runtime.
✅ **Step 3: The code executes inside the container** with limited CPU/memory/time.
✅ **Step 4: Capture the output and compare it with expected results.**
✅ **Step 5: Return verdict** (Accepted, Wrong Answer, TLE, Runtime Error, etc.).

🔹 Example Workflow in Spring Boot

- Store code submission in a temp directory.
- Run `docker run --rm -v /submissions:/code ubuntu bash -c "javac Code.java && java Code"`
- Capture output and match with expected results.
- Delete the container after execution.

⚡ Pros

✔️ Secure execution inside an isolated environment.
✔️ Supports multiple languages by using different Docker images.
✔️ Prevents malicious code from affecting the system.

⚠️ Cons

❌ Slight overhead in creating & destroying containers.
❌ Requires setting up Docker & managing resource limits.

2️⃣ Using an Existing Online Code Execution API (Faster, but Less Control)

If you **don’t want to manage execution infrastructure**, you can use **third-party APIs** like:

1. **JDoodle API** – Supports 70+ languages.
2. **Sphere Engine** – A robust API for executing & evaluating code.
3. **HackerEarth API** – Used for coding platforms.

🔹 Example API Call

{ "language": "java", "source_code": "public class Main { public static void main(String[] args) { System.out.println(\"Hello, World!\"); } }", "input": "", "time_limit": 2, "memory_limit": 256 }