SonarQube Integration Project Submission

1. Project Overview

Project Name: AddressBookSystem
Module Included: Address Book Module
Type: Core Java + Eclipse Project
Source Folder: src
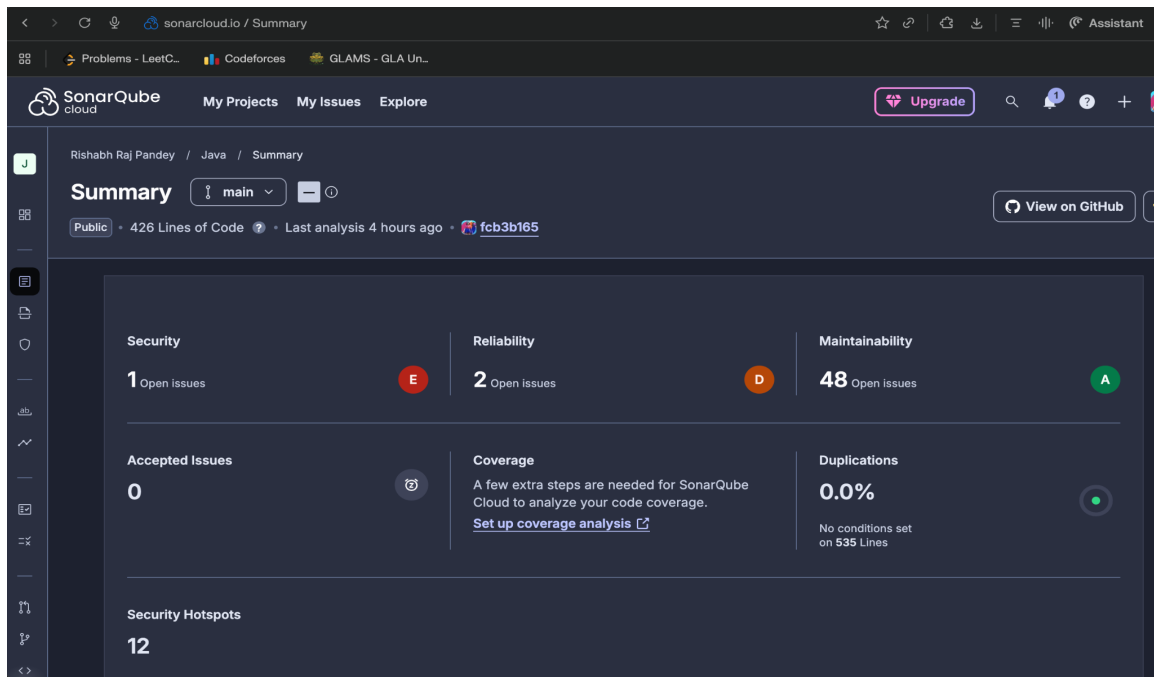Compiled Output Folder: bin

2. SonarQube Integration Procedure

Step 1: Install SonarQube Community Build and start server using StartSonar.bat.
Step 2: Access http://localhost:9000 and login with admin credentials.
Step 3: Create new project with Project Key: address-book and generate token.
Step 4: Install SonarScanner CLI and add it to system PATH.
Step 5: Compile Java project to generate .class files in bin folder.
Step 6: Run SonarScanner from project directory.
Step 7: View analysis report on SonarQube Dashboard.

3. SonarQube Analysis Result Summary

Quality Gate: PASSED
Security Issues: 1
Reliability Issues: 2
Maintainability Issues: 48 Code Smells
Coverage: 0% (JUnit not added)
Duplications: 0.0%

4. Screenshots & Execution Proof

sonarcloud.io / Overview

☆ ⊘ | ⊘ ⊎ | ≡ ⊪⊪ ⓒ Assistant

⬚ | ⇲ Problems - LeetC... | ⬛ Codeforces | 🐢 GLAMS - GLA Un...

SonarQube cloud

My Projects   **My Issues**   **Explore**

⬦ Upgrade 🔍 🔔¹ ❓ + 🌸

J

Rishabh Raj Pandey / Java / Overview

Overview

🐙 **View on GitHub** ⭐

Public • No tags • **426** Lines of Code • Last analysis **4 hours ago**

J   **Java**

━━

● Java ● XML

Main Branch Status

Main Branch Evolution since 4 hours ago

Quality Gate ⓘ

—   **Not computed**

No historical data yet

Rishabh Raj Pandey / Java / Summary

Summary

main

Public · 426 Lines of Code · Last analysis 4 hours ago · fcb3b165

View on GitHub

Security		Reliability		Maintainability	
1 Open issues	E	**2** Open issues	D	**48** Open issues	A

Accepted Issues		Coverage	Duplications	
0		A few extra steps are needed for SonarQube Cloud to analyze your code coverage. **Set up coverage analysis**	**0.0%** No conditions set on 535 Lines	

Security Hotspots

12

```
15:56:30.450 INFO  Load branch configuration (done) | time=7ms
15:56:30.471 INFO  Load quality profiles
15:56:31.279 INFO  Load quality profiles (done) | time=808ms
15:56:31.315 WARN  SCM provider autodetection failed. Please use "sonar.scm.provider" to define SCM of
your project, or disable the SCM Sensor in the project settings.
15:56:31.328 INFO  Create analysis
15:56:31.929 INFO  Create analysis (done) | time=601ms
15:56:32.548 INFO  Load active rules
15:56:34.341 INFO  Load active rules (done) | time=1794ms
15:56:35.142 INFO  Organization key: flyingcrab31
15:56:35.214 INFO  Preprocessing files...
15:56:35.448 ERROR Error during SonarScanner Engine execution
java.lang.IllegalStateException: Failed to preprocess files
        at org.sonar.scanner.scan.filesystem.ProjectFilePreprocessor.processModuleSources(ProjectFilePr
eprocessor.java:170)
        at org.sonar.scanner.scan.filesystem.ProjectFilePreprocessor.processModule(ProjectFilePreproces
sor.java:143)
        at org.sonar.scanner.scan.filesystem.ProjectFilePreprocessor.processModulesRecursively(ProjectF
ilePreprocessor.java:131)
        at org.sonar.scanner.scan.filesystem.ProjectFilePreprocessor.execute(ProjectFilePreprocessor.ja
va:99)
        at org.sonar.scanner.bootstrap.SpringScannerContainer.doAfterStart(SpringScannerContainer.java:
466)
        at org.sonar.core.platform.SpringComponentContainer.startComponents(SpringComponentContainer.ja
va:210)
        at org.sonar.core.platform.SpringComponentContainer.execute(SpringComponentContainer.java:189)
        at org.sonar.scanner.bootstrap.SpringGlobalContainer.doAfterStart(SpringGlobalContainer.java:14
6)
        at org.sonar.core.platform.SpringComponentContainer.startComponents(SpringComponentContainer.ja
va:210)
        at org.sonar.core.platform.SpringComponentContainer.execute(SpringComponentContainer.java:189)
        at org.sonar.scanner.bootstrap.ScannerMain.runScannerEngine(ScannerMain.java:143)
        at org.sonar.scanner.bootstrap.ScannerMain.run(ScannerMain.java:58)
```