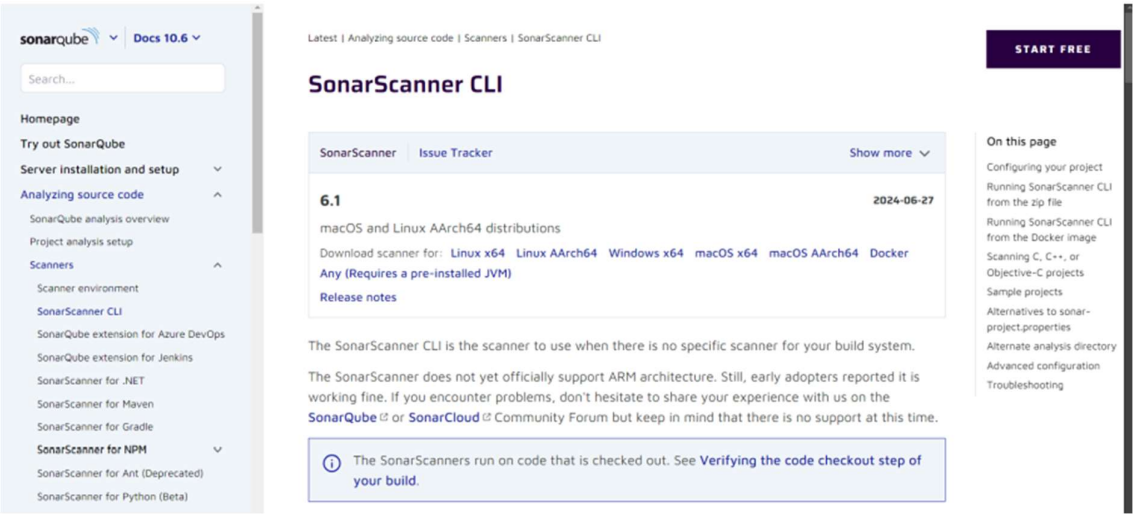
Name : Sandeshkumar.M.Yadav **Div : D15C** **Roll : 61**

Experiment No: 8

AIM: Create a Jenkins CI/CD Pipeline with SonarQube / GitLab Integration to perform a static analysis of the code to detect bugs, code smells, and security vulnerabilities on a sample Web / Java / Python application.

PREREQUISITES:

Step 1: Download sonar scanner https://docs.sonarsource.com/sonarqube/latest/analyzing-source-code/scanners/sonarscanner/ . Visit this link and download the sonarqube scanner CLI



Step 2: Docker Run docker -v command .If docker is not installed so install it

```
C:\Users\Sandesh>docker -v
Docker version 27.2.0, build 3ab4256
```

Step 3: Install sonarqube image Command: docker pull sonarqube

```
C:\Users\Student>docker pull sonarqube
Using default tag: latest
latest: Pulling from library/sonarqube
762bedf4b1b7: Pull complete
95f9bd9906fa: Pull complete
a32d681e6b99: Pull complete
aabdd0a18314: Pull complete
5161e45ecd8d: Pull complete
aeb0020dfa06: Pull complete
01548d361aea: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:bb444c58c1e04d8a147a3bb12af941c57e0100a5b21d10e599384d59b
Status: Downloaded newer image for sonarqube:latest
docker.io/library/sonarqube:latest

What's next:
    View a summary of image vulnerabilities and recommendations → docker
```

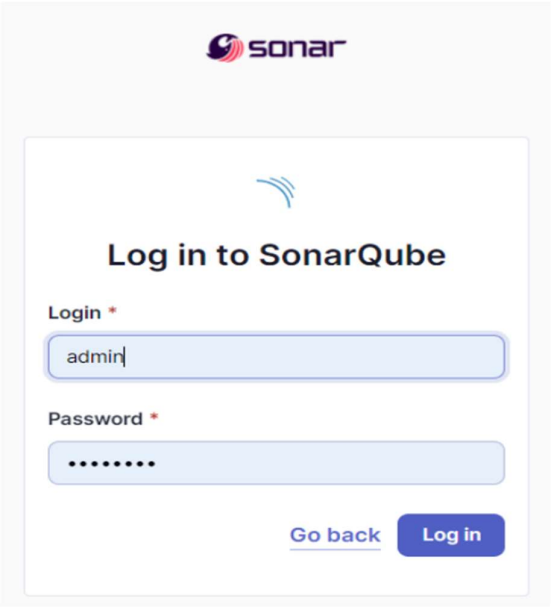
Step 4: Keep Jenkins installed on your system.

EXPERIMENT STEPS:

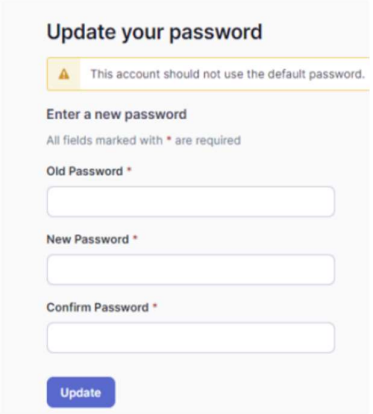
Step1: Run SonarQube image docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest .This command will run the SonarQube image that was just installed using docker.

```
C:\Users\Student>docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
83330c33cd961d8d659f362c5f62c6cd1ff87f31ec99da134350b9b419370561
```

Step 2: Once the SonarQube image is started, you can go to http://localhost:9000 to find the SonarQube that has started

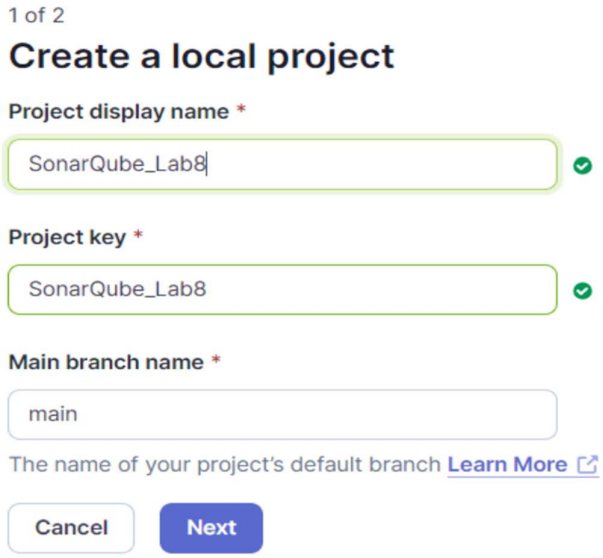


Step 3: On this interface, login with username = ‘admin’ and password = ‘admin’. Once logged in successfully, SonarQube will ask you to reset this password. Reset it and remember this password.



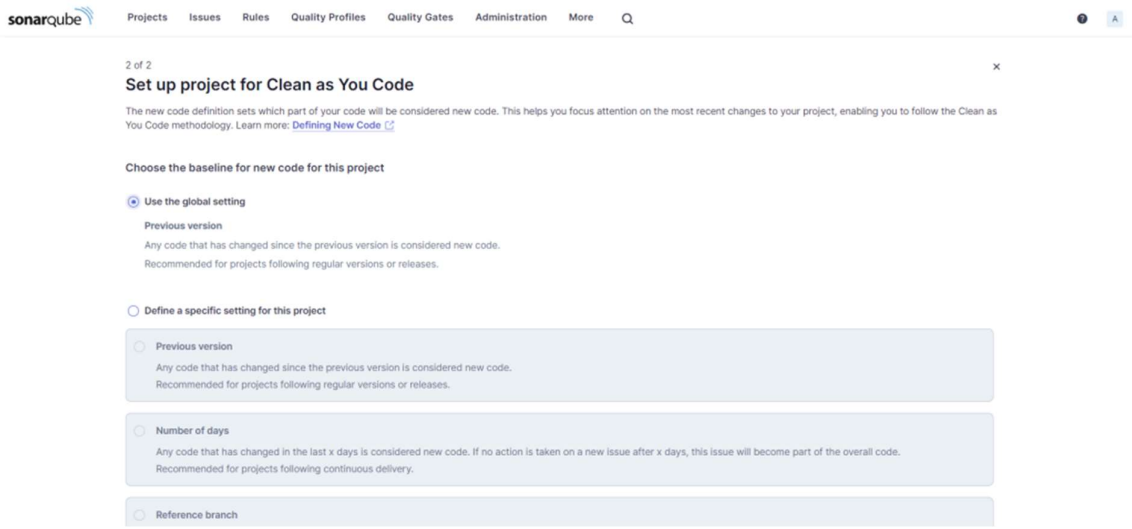
Step 4: After changing the password, you will be directed to this screen. Click on Create a Local Project. Give the project a display name and project key

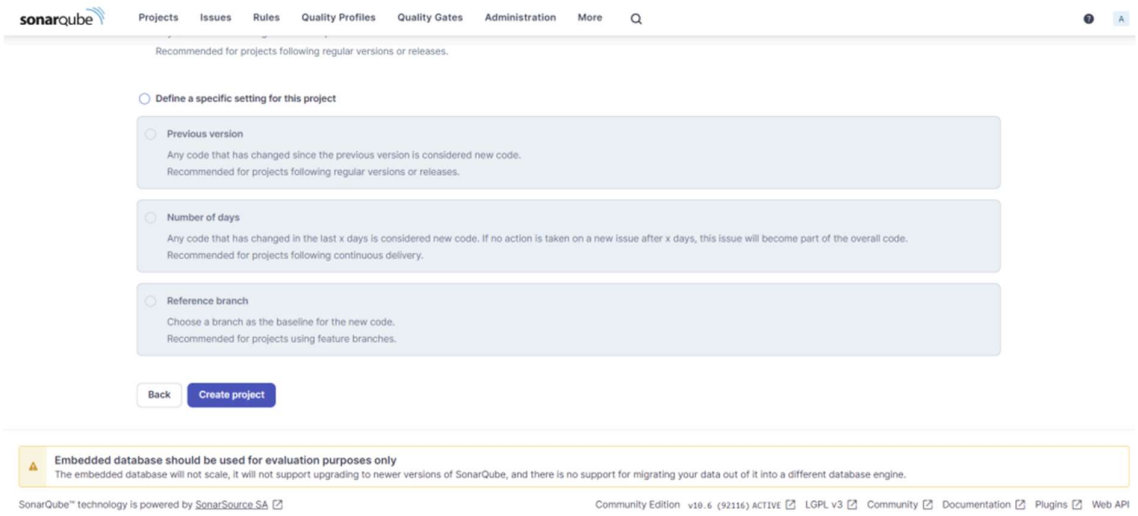
Click on Create Project



Set up the project as required and click on create.

In the Step 2 while creating the project,Sonarqube ask you regarding which code should be considered as the new code for examining it .

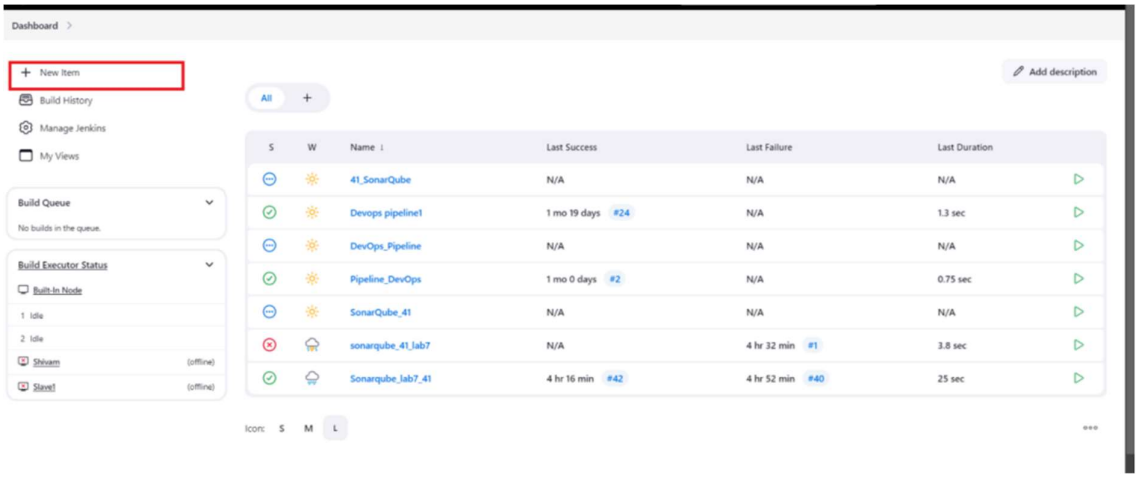




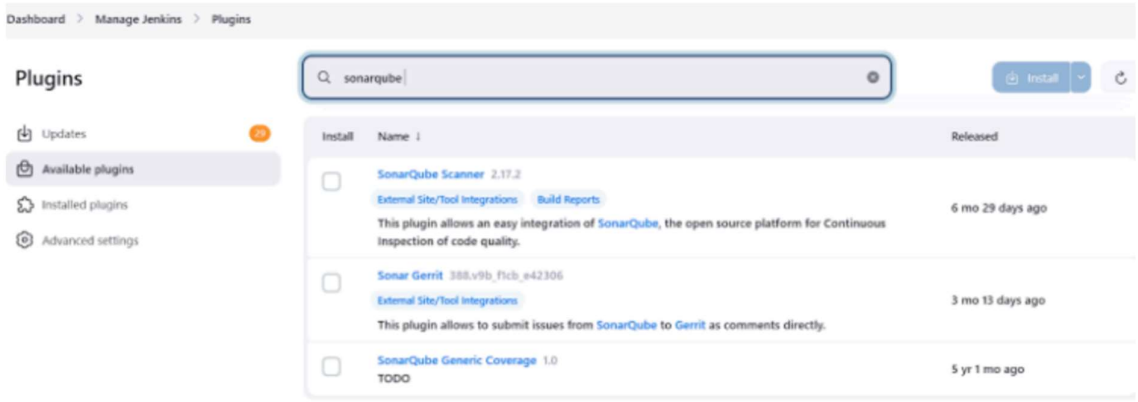
Click on Create

Project is created

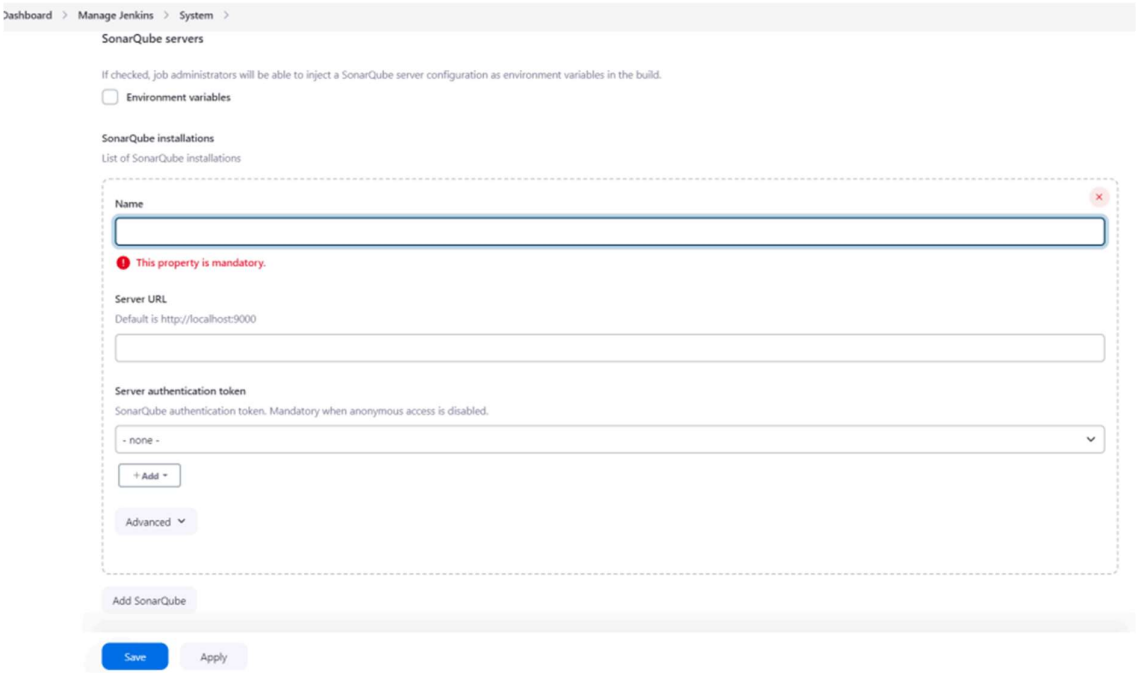
Step 5: Open Jenkins on whichever port it is installed. (http://loaclhost:). Go to the new item



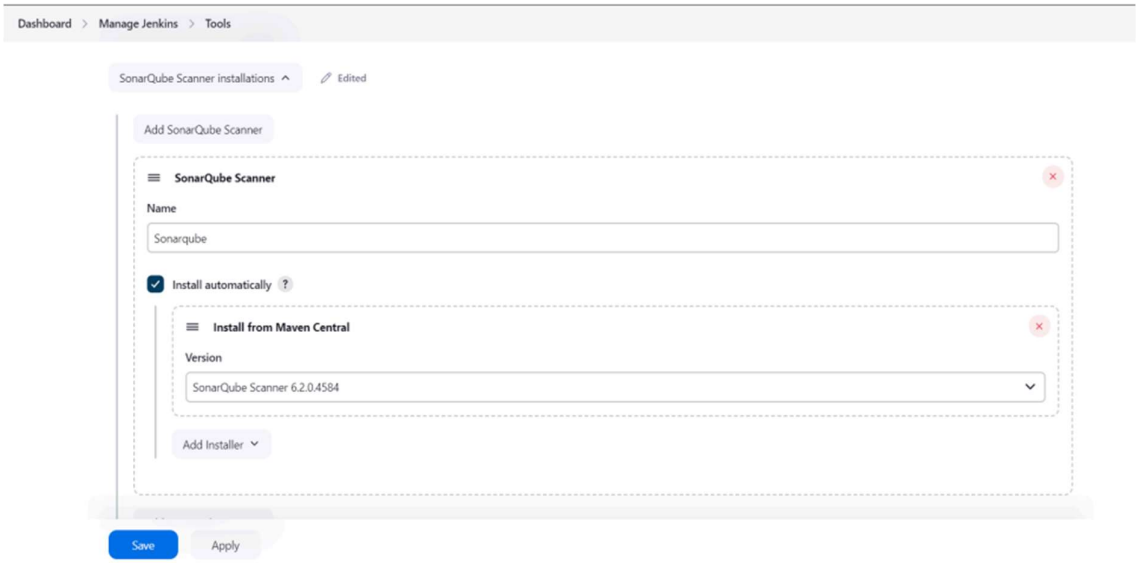
Step 6: Go to manage jenkins →available plugins then Search for Sonarqube Scanner for Jenkins and install it

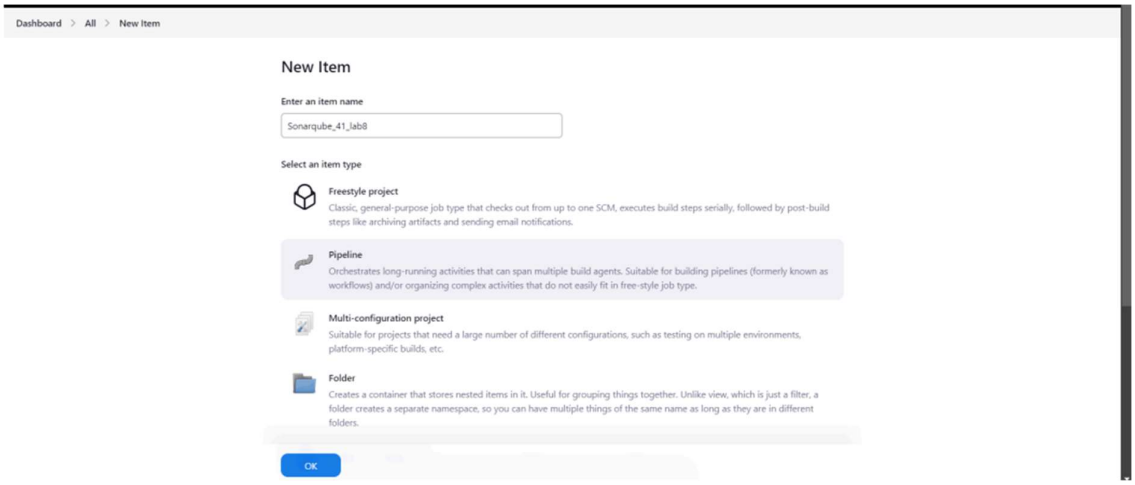


Step 7: Now, go to Manage Jenkins → System. Under Sonarqube servers, add a server. Add server authentication token if needed.



Step 8: Go to Manage Jenkins → Tools. Go to SonarQube scanner, choose the latest configuration and choose to install automatically.

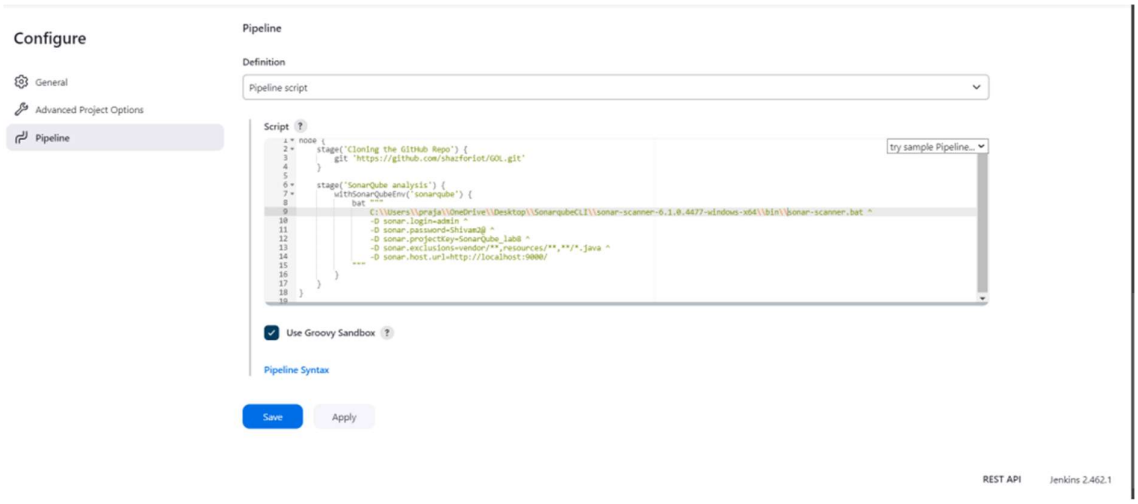




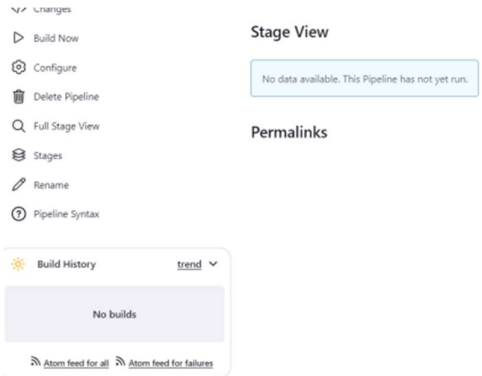
Step 10: Under Pipeline script, enter the following:

```
node {
stage('Cloning the GitHub Repo') {
git 'https://github.com/shazforiot/GOL.git
}
stage('SonarQube analysis') {
withSonarQubeEnv('sonarqube') {
bat """

C:\\Users\\Sandesh\\Downloads\\SonarqubeCLI\\sonar-scanner-6.1.0.4477-windows-x64\\bi
n\\sonar-scanner.bat ^
-D sonar.login=admin ^
-D sonar.password=Sandesh2@ ^
-D sonar.projectKey=SonarQube_Lab8 ^
-D sonar.exclusions=vendor/**,resources/**,**/*.java ^
-D sonar.host.url=http://localhost:9000/
"""
}
}
}
```



Click on save.



This is a Java sample project with many repetitive sections and coding issues that SonarQube will be able to detect during analysis.

Step 11: Go back to jenkins. Go to the job you had just built and click on Build Now.

Stage View

	Cloning the GitHub Repo	SonarQube analysis
Average stage times: (Average full run time: ~15min 45s)	2s	1min 26s
#14 Sep 23 09:19 No Changes	3s	15min 39s

The problem was C:\windows\system32 was not there so we need to add in our environment variable .

Now Check the console output once

```
Keep only the first 100 references.
09:31:43.178 WARN  Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/assertions/gui/package-summary.html for block at line 40.
Keep only the first 100 references.
09:31:43.194 WARN  Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/mail/sampler/package-summary.html for block at
line 39. Keep only the first 100 references.
09:31:43.194 WARN  Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/mail/sampler/package-summary.html for block at
line 40. Keep only the first 100 references.
09:31:43.194 INFO  CPD Executor CPD calculation finished (done) | time=170824ms
09:31:43.213 INFO  SCM revision ID 'ba799ba7e1b576f04a4612322b0412c5e6e1e5e4'
09:34:16.341 INFO  Analysis report generated in 4552ms, dir size=127.2 MB
09:34:33.584 INFO  Analysis report compressed in 17210ms, zip size=29.6 MB
09:34:34.392 INFO  Analysis report uploaded in 807ms
09:34:34.395 INFO  ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=SonarQube_Lab8
09:34:34.395 INFO  Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
09:34:34.395 INFO  More about the report processing at http://localhost:9000/api/ce/task?id=7dfe78a1-793d-47f9-bf86-4636bb755c69
09:34:45.065 INFO  Analysis total time: 15:27.329 s
09:34:45.070 INFO  SonarScanner Engine completed successfully
09:34:45.929 INFO  EXECUTION SUCCESS
09:34:45.932 INFO  Total time: 15:36.252s
[Pipeline] }
[Pipeline] // withSonarQubeEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

REST API Jenkins 2.462.1

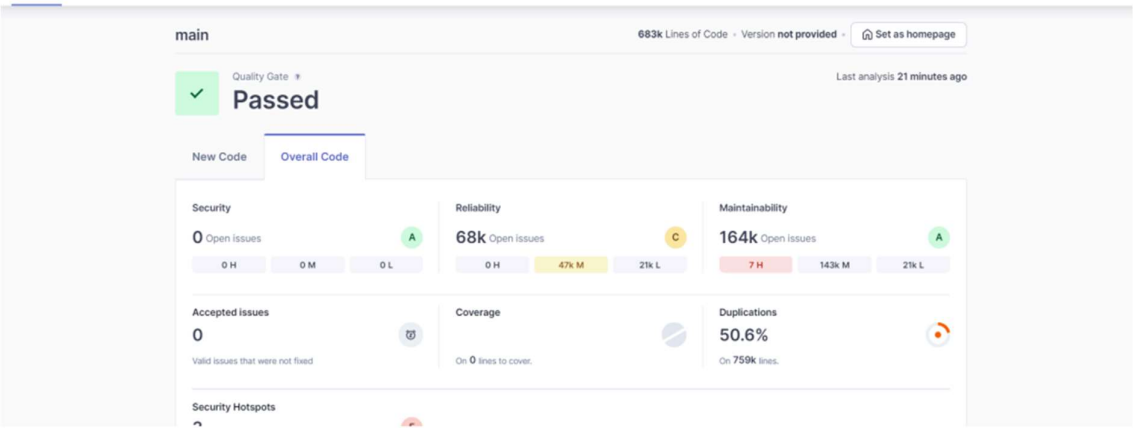
Successfully BUILD

Step 12: After the build is finished, return to SonarQube and review the linked project in detail.

SonarQube_Lab8 PUBLIC — Passed

Last analysis: 19 minutes ago · 683k Lines of Code · HTML, XML, ...

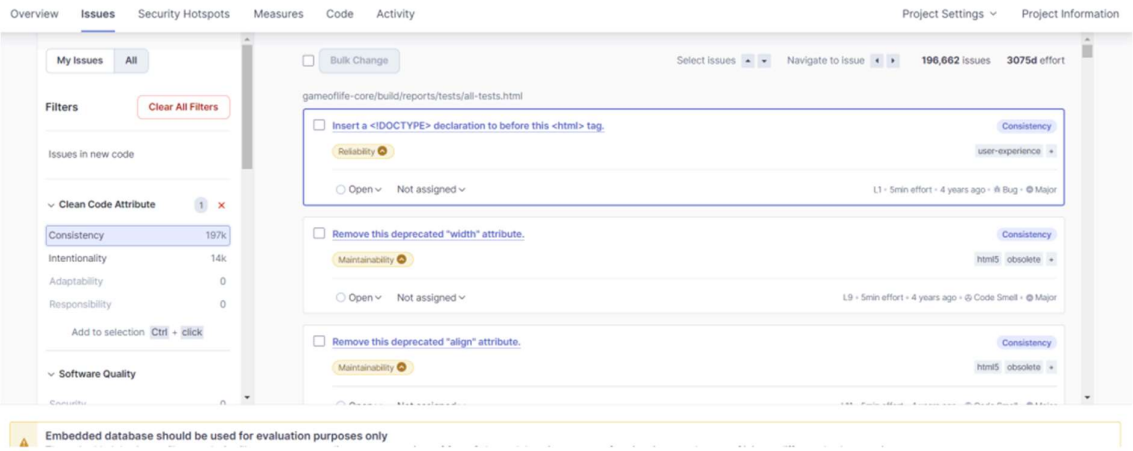
A 0	C 68k	A 164k	E 0.0%	—	50.6%
Security	Reliability	Maintainability	Hotspots Reviewed	Coverage	Duplications



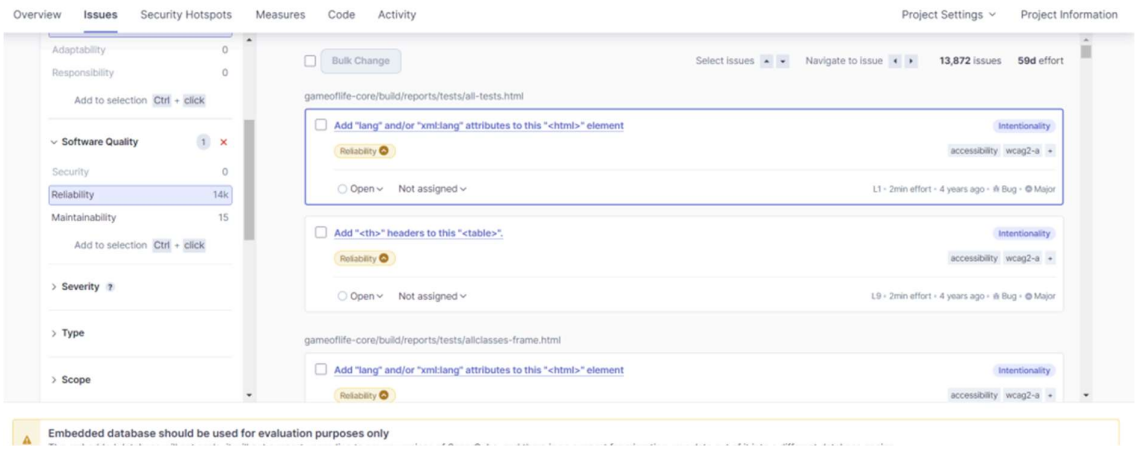
Under different options on the navbar , we can check all the issues with the code.

UNDER ISSUES:

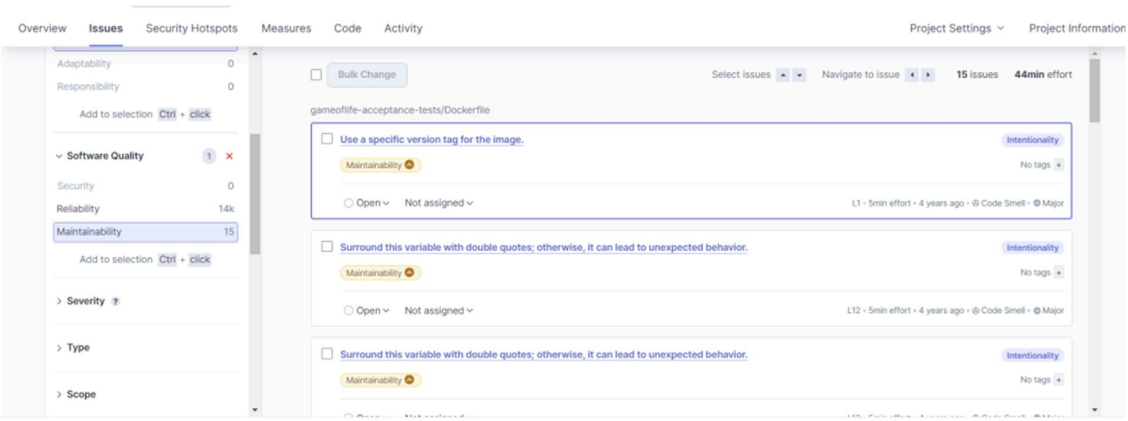
1) Consistency



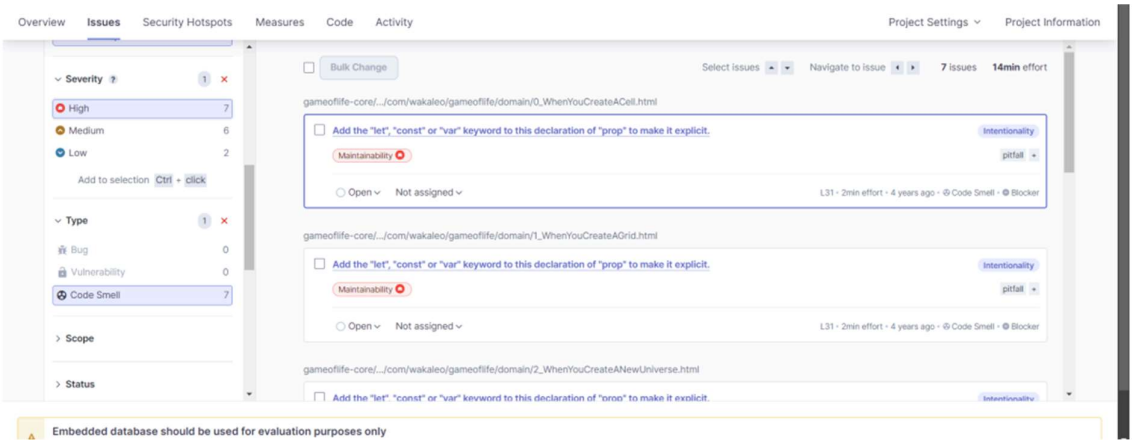
2) Reliability



3) Maintainability

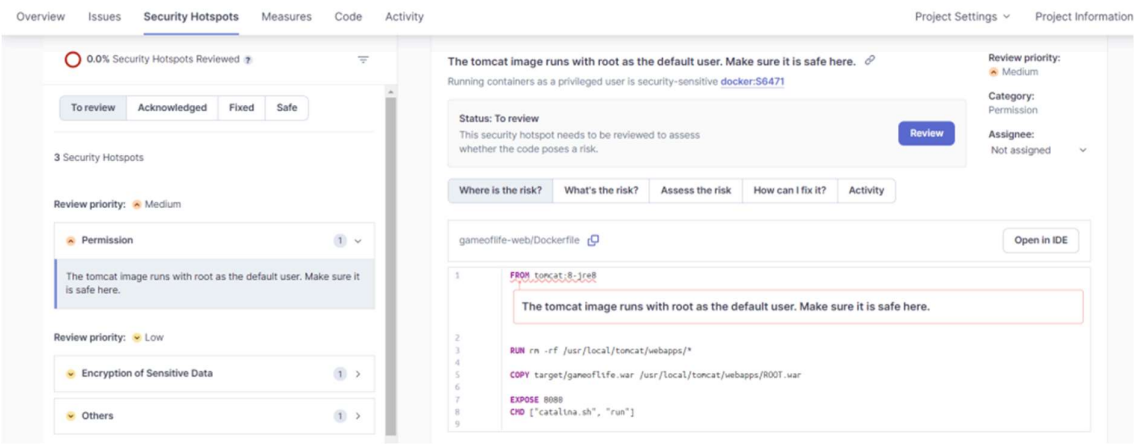


4) Severity

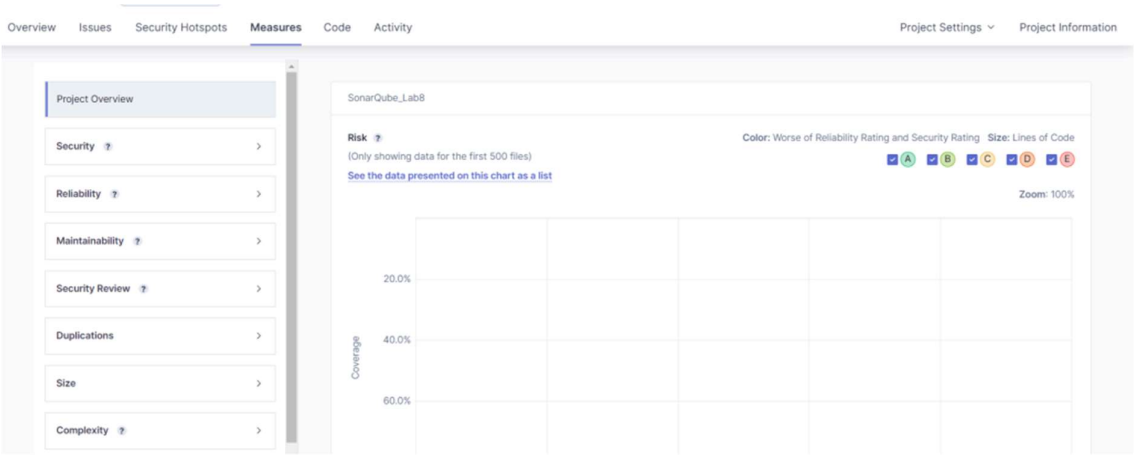


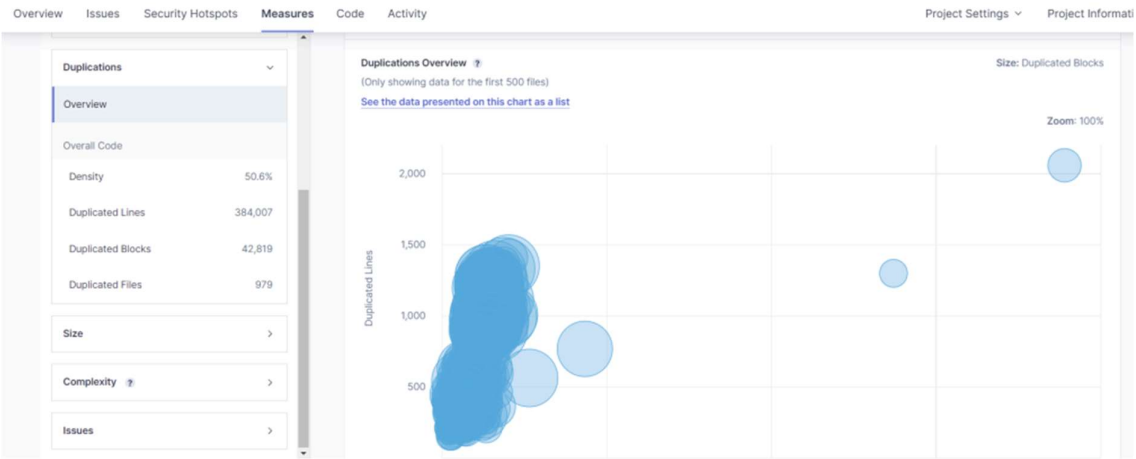
UNDER SECURITY HOTSPOT:

Name : Sandeshkumar.M.Yadav **Div : D15C** **Roll : 61**



UNDER MEASURES:





CONCLUSION: