Lab Assignment 7

AIM: To perform static analysis on Python programs using SonarQube SAST process.

LO4: To identify and remediate application vulnerabilities earlier and help integrate security in the development process using SAST Techniques.

THEORY:

SonarQube:

Overview: SonarQube is an open-source platform for continuous inspection of code quality. It is used to analyze and measure code quality and security issues in a codebase.

Features:

Static Code Analysis: SonarQube scans source code to identify bugs, code smells, and security vulnerabilities.

Continuous Integration: It integrates seamlessly with CI/CD pipelines, providing automated code analysis during the development process.

Security Analysis: While it primarily focuses on code quality, it also has some security rules to catch common security issues.

Maintainability Metrics: SonarQube provides maintainability metrics and helps teams understand code complexity and maintainability.

Dashboard and Reporting: It offers dashboards and reports for tracking code quality and issues over time.

Use Case: SonarQube is used for improving code quality, maintainability, and to catch some common code security issues. It's more about general code quality and development best practices.

SAST (Static Application Security Testing):

Overview: SAST is a security testing method that analyzes source code, bytecode, or binary code for vulnerabilities without executing the application. It is primarily focused on identifying security issues and vulnerabilities in the code.

Features:

Code Scanning: SAST tools examine the source code or compiled code to identify potential security vulnerabilities, such as SQL injection, cross-site scripting, and more.

Early Detection: SAST is used early in the development process to find security issues before they can be exploited.

Language Support: SAST tools support various programming languages and frameworks.

Integration: They can be integrated into CI/CD pipelines to automatically scan code before deployment.

Use Case: SAST is used for finding and fixing security vulnerabilities in code. It helps secure applications by identifying potential security threats early in the development lifecycle.

1. INSTALL sonarqube (docker images) and sonarscanner zip file from https://docs.sonarsource.com/sonarqube/latest/analyzing-sourcecode/scanners/sonarscanner/ and set up config file as given in docs.

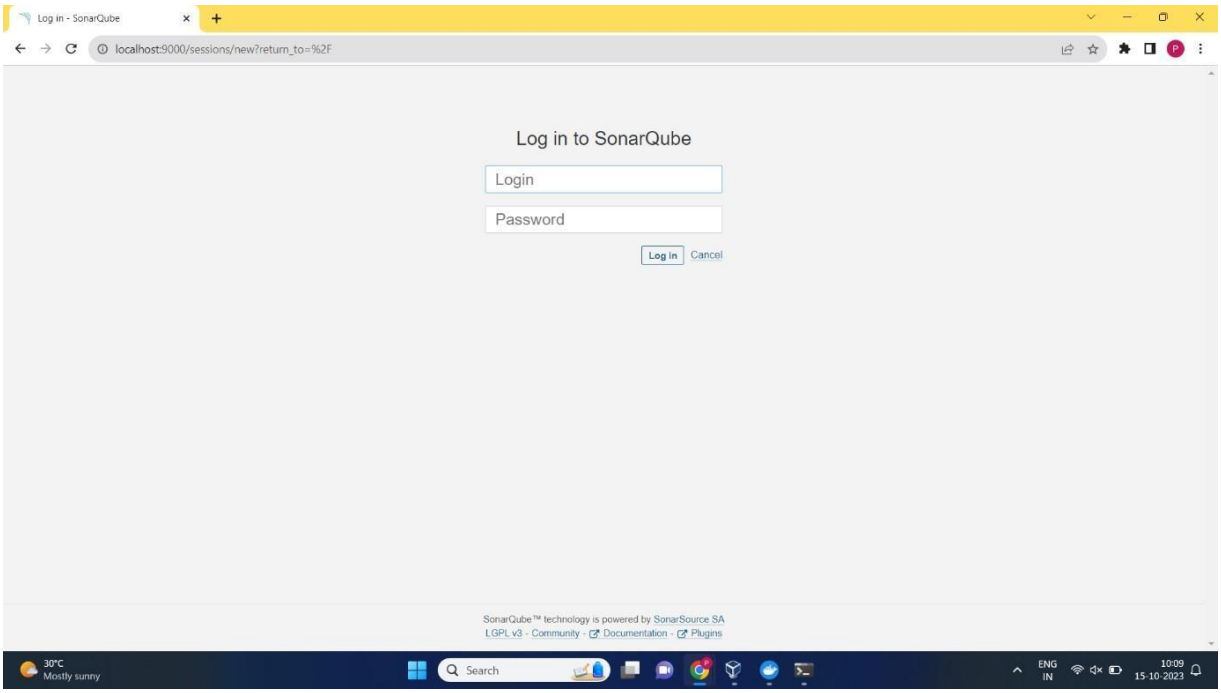
```
Command Prompt
Microsoft Windows [Version 10.0.22621.2428]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Pratik Arote>docker pull sonarqube
Using default tag: latest
latest: Pulling from library/sonarqube
43f89b94cd7d: Pull complete
50431c77a77b: Pull complete
dfd8e860e672: Pull complete
637e2db99ae6: Pull complete
7de1c2853278: Pull complete
d2152ffce821: Pull complete
519cf218564f: Pull complete
Digest: sha256:c6c8096375002d4cb2ef64b89a2736ad572812a87a2917d92e7e59384b9f6f65
Status: Downloaded newer image for sonarqube:latest
docker.io/library/sonarqube:latest

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview sonarqube

C:\Users\Pratik Arote>docker pull sonarsource/sonar-scanner-cli
Using default tag: latest
latest: Pulling from sonarsource/sonar-scanner-cli
9398808236ff: Pull complete
4f4fb700ef54: Pull complete
3cd77fb28e46: Pull complete
f78b288abc31: Pull complete
Digest: sha256:494ecc3b5b1ee1625bd377b3905c4284e4f0cc155cff397805a244dee1c7d575
Status: Downloaded newer image for sonarsource/sonar-scanner-cli:latest
docker.io/sonarsource/sonar-scanner-cli:latest

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview sonarsource/sonar-scanner-cli
```

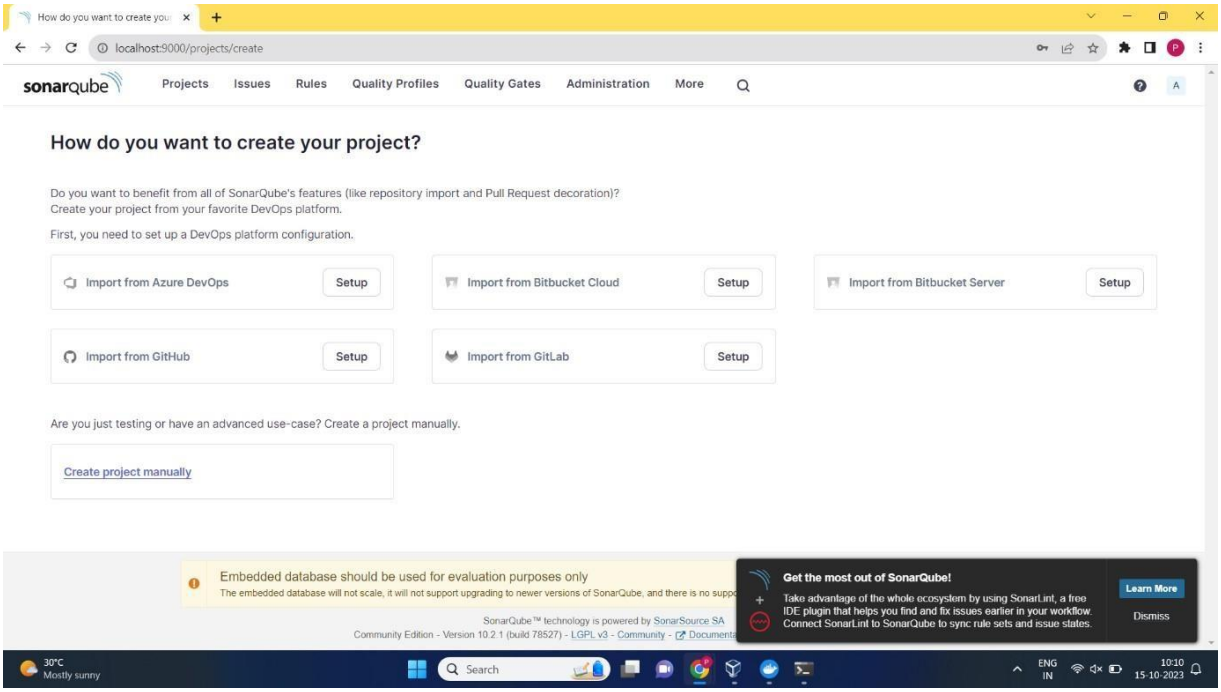
2. Spin up the container

```
C:\Users\Pratik Arote>docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
f3630dbc2ffa6e5598ad922085026400a1f9f1564416b0606b5348000f6d1377

C:\Users\Pratik Arote>docker images
REPOSITORY                     TAG       IMAGE ID       CREATED        SIZE
sonarqube                      latest    3183d6818c6e   42 hours ago   716MB
sample-web-app                 latest    713c7cdaaf78   2 weeks ago    42.7MB
myimage                        latest    438bb56a50a3   2 weeks ago    122MB
sonarsource/sonar-scanner-cli  latest    2f384fb1bbd5   5 weeks ago    358MB
ubuntu                         latest    c6b84b685f35   8 weeks ago    77.8MB
hello-world                    latest    9c7a54a9a43c   5 months ago   13.3kB

C:\Users\Pratik Arote>docker ps
CONTAINER ID   IMAGE              COMMAND                  CREATED          STATUS          PORTS                    NAMES
f3630dbc2ffa   sonarqube:latest   "/opt/sonarqube/dock…"   27 minutes ago   Up 27 minutes   0.0.0.0:9000->9000/tcp   sonarqube
```

3. Open http://localhost:9000 on the browser. Enter login and password both as “admin” and then set up new password.

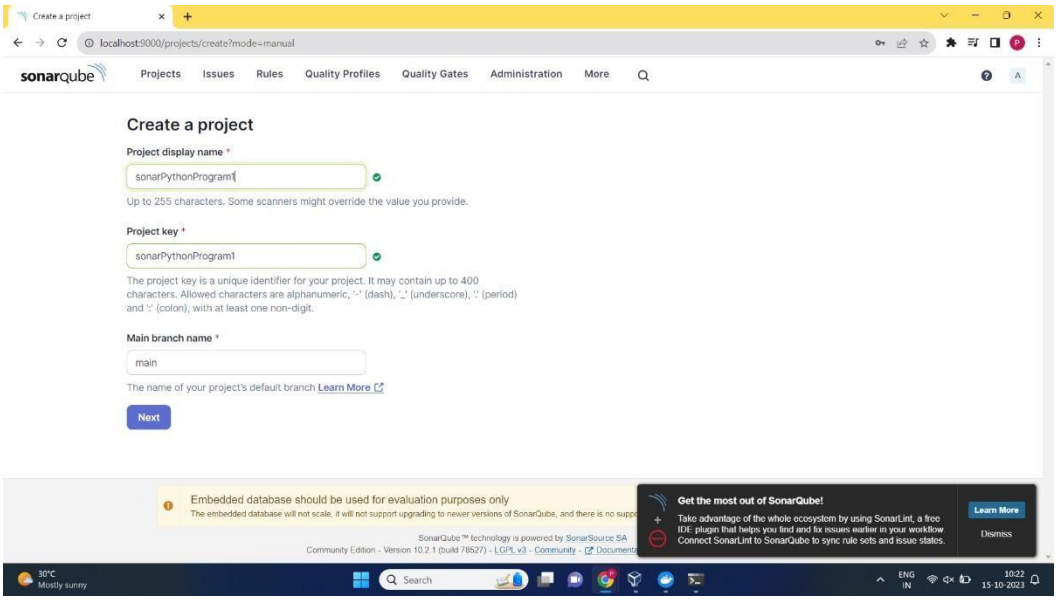
Anmol Tripathi T23 138

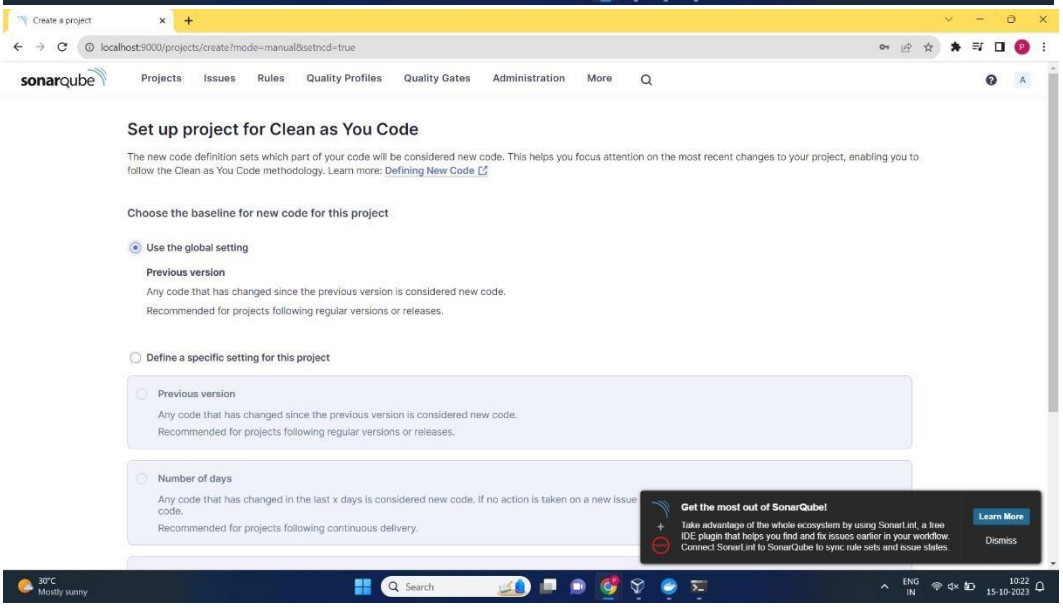




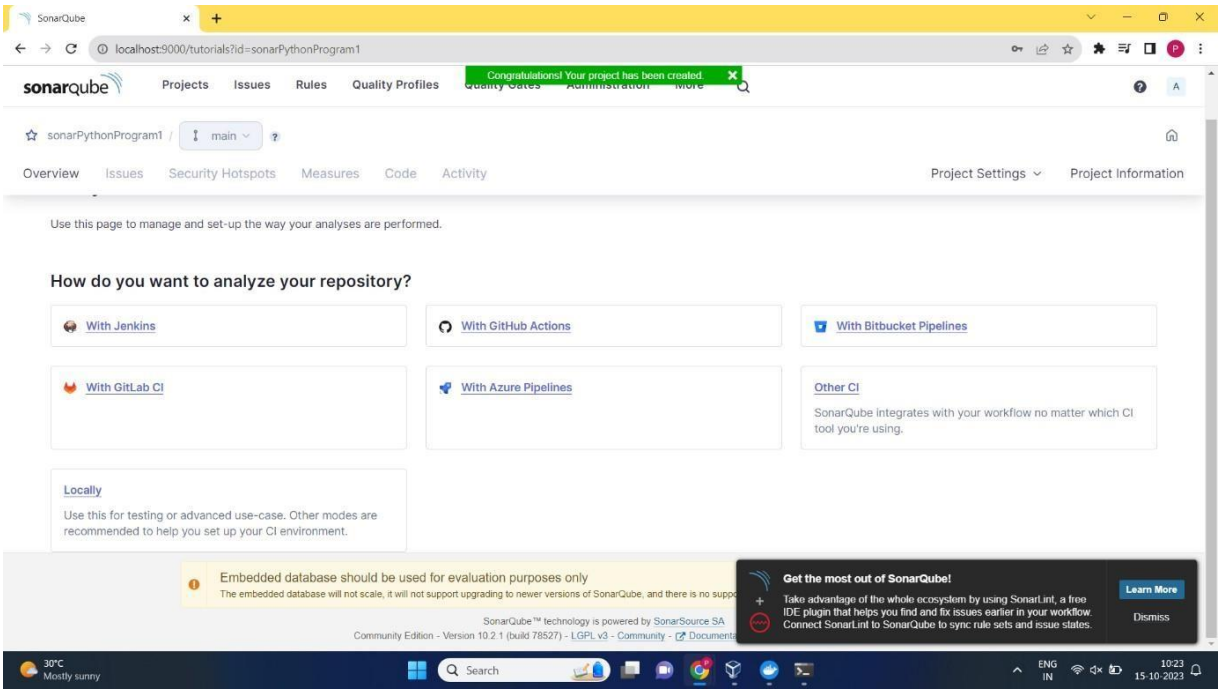
4. Create a project

Anmol Tripathi T23 138

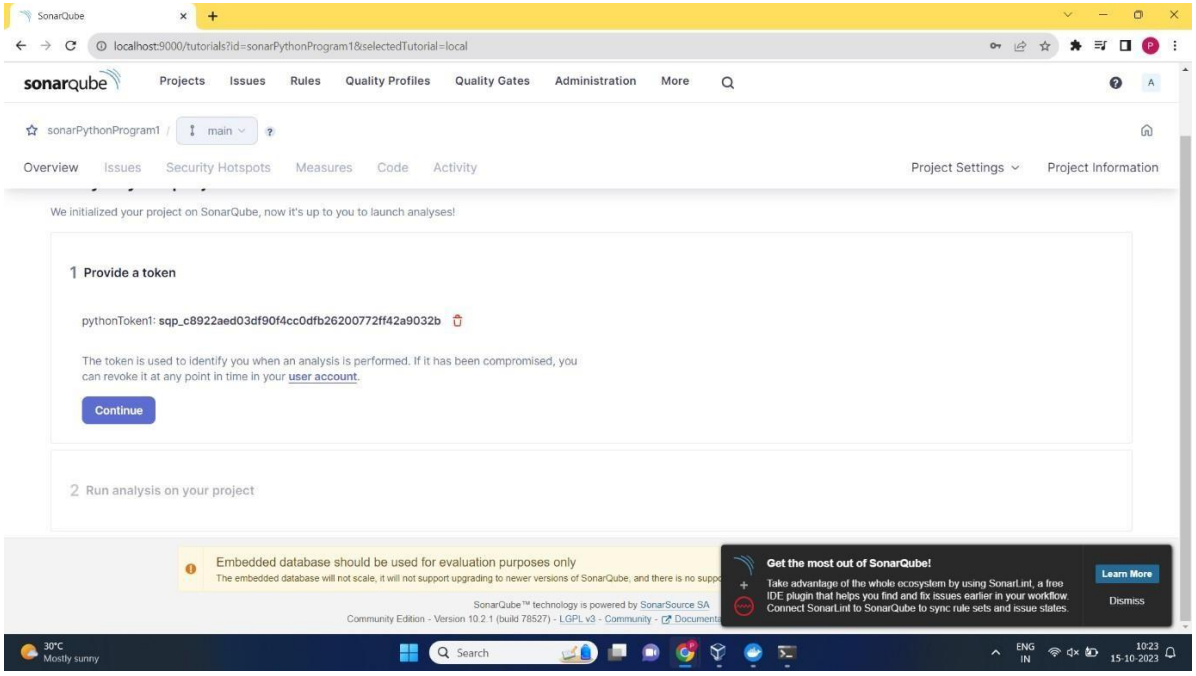


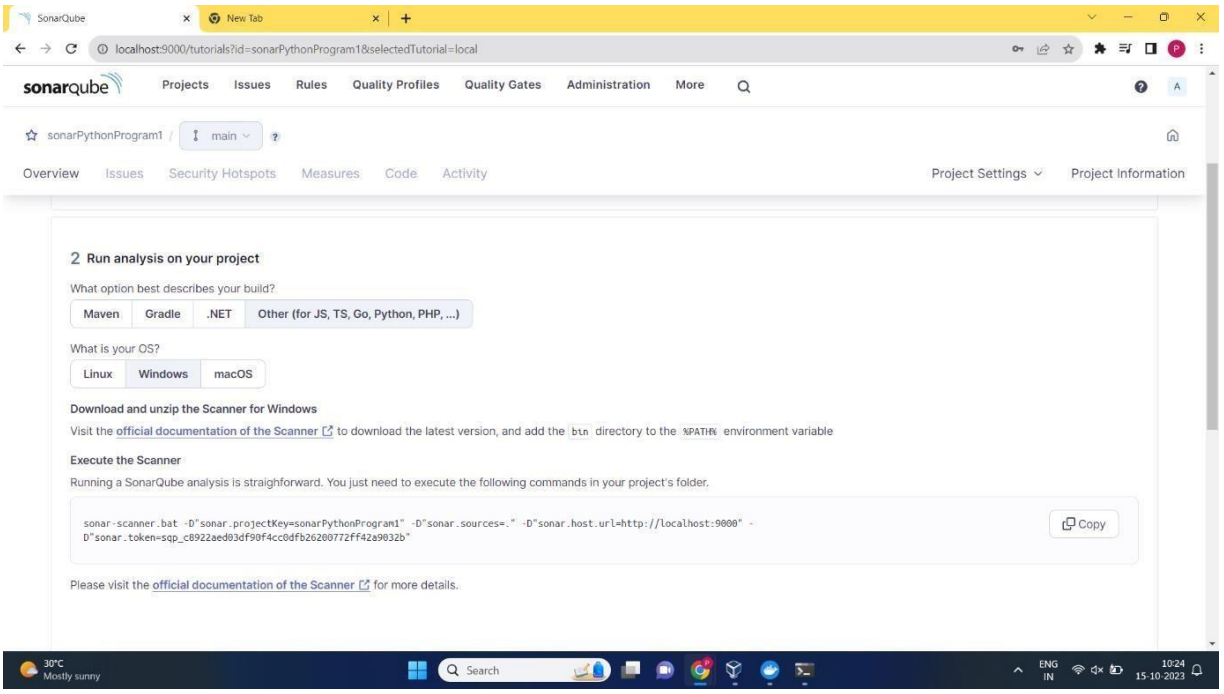


Anmol Tripathi T23 138

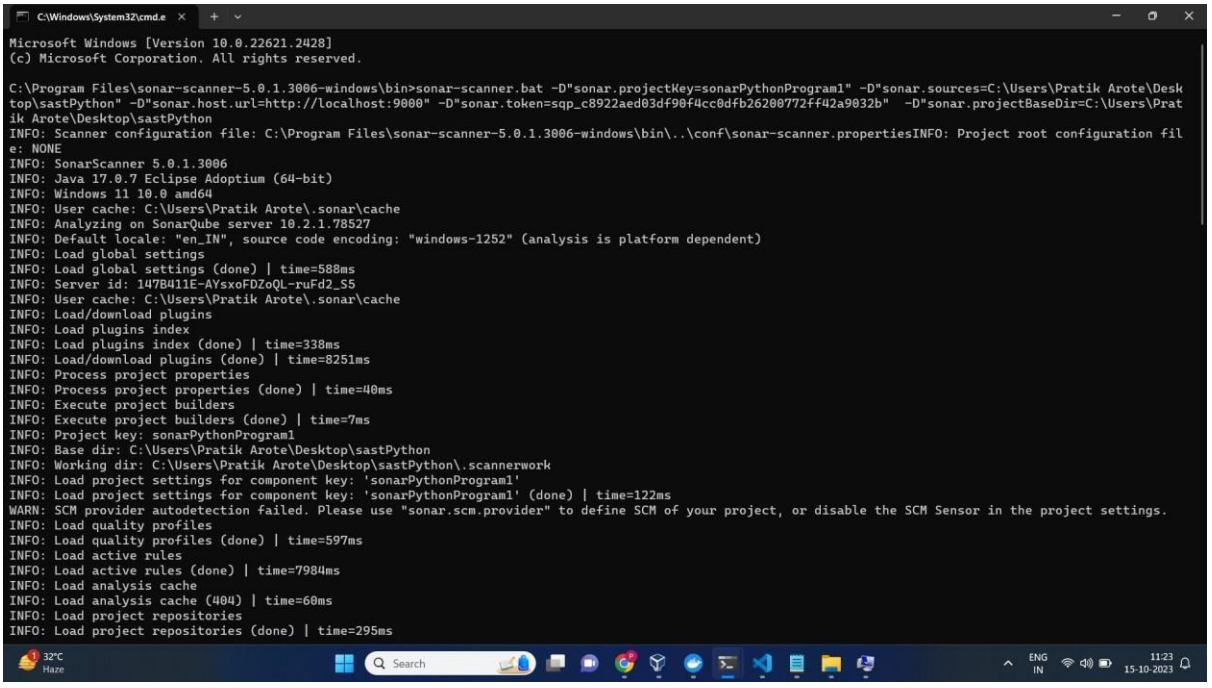


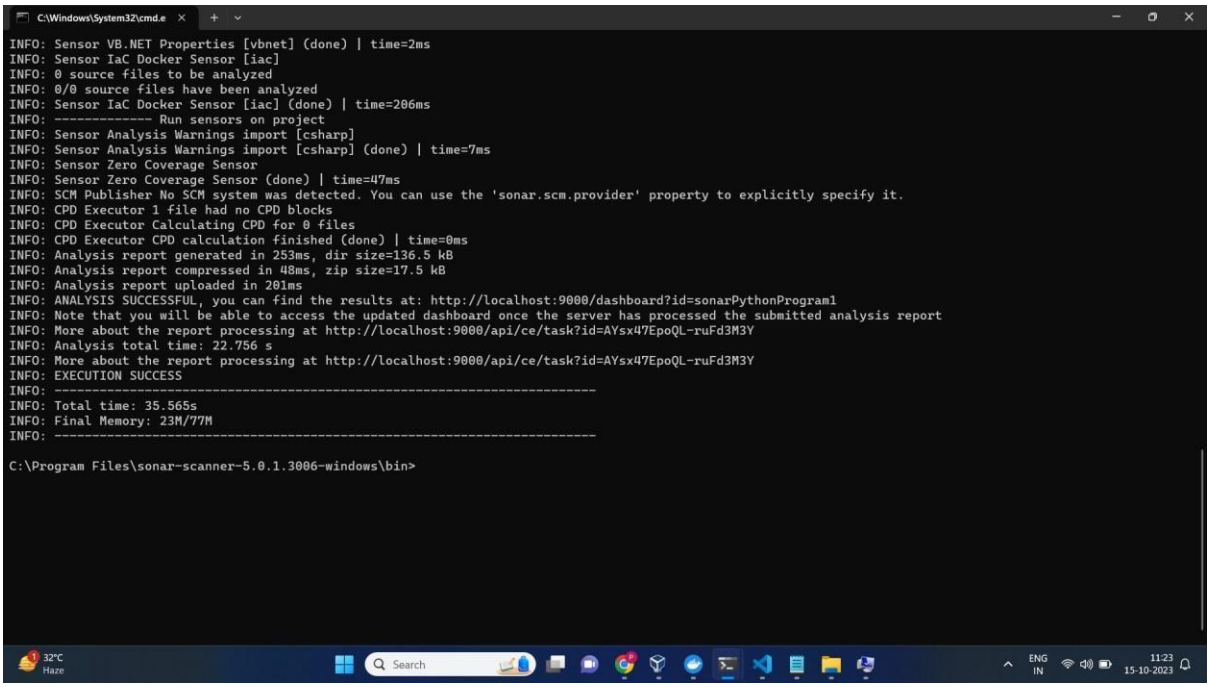
5. Provide token



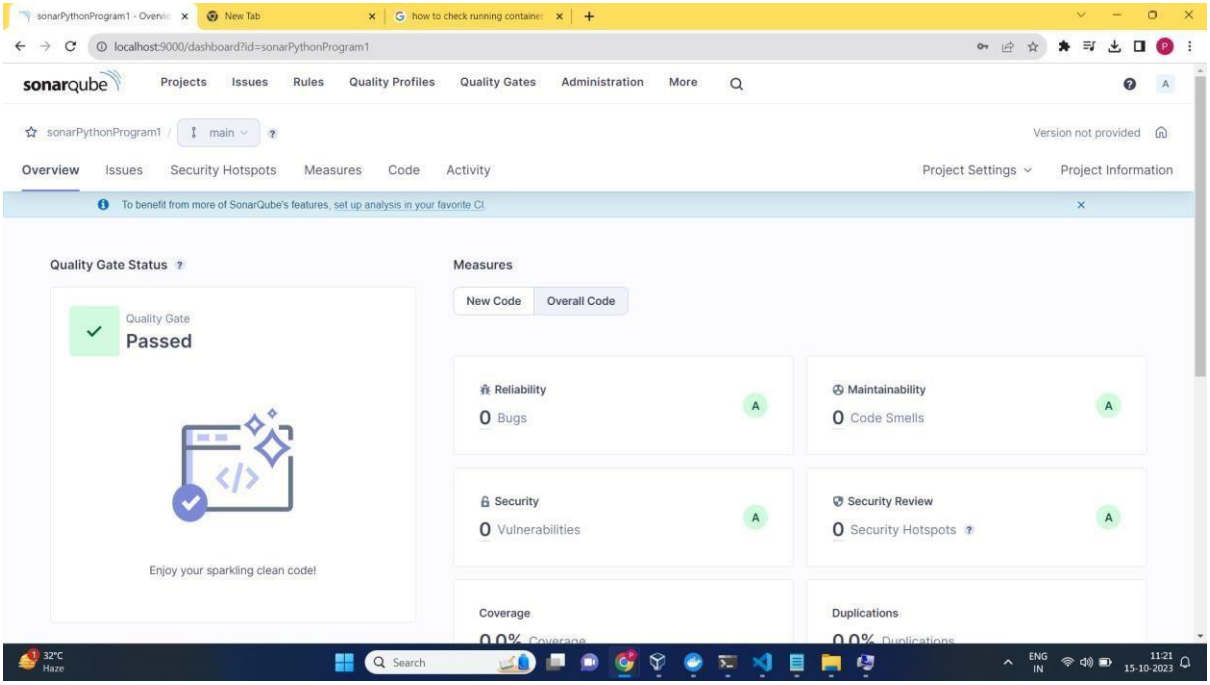


6. Enter the following command





7. See the result of the test



CONCLUSION:

Here we have successfully performed static analysis of python programs.