

Lab Assignment 6

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 and sonarscanner zip file from <https://docs.sonarsource.com/sonarqube/latest/analyzingsourcecode/scanners/sonarscanner/> and set up config file as given in docs.

The screenshot shows a web browser displaying the SonarQube documentation page for installing a local instance. The page title is "Installing a local instance of SonarQube". The left sidebar contains a navigation menu with links to "Homepage", "Try out SonarQube", "Requirements", "Setup and upgrade", "Analyzing source code", "DevOps platform integration", "User guide", "Project administration", "Instance administration", "Extension Guide", "Sonar Home", "SonarQube", "Community", "Clean Code", "Twitter", and "News". The main content area includes a search bar, a "Docs 9.9 LTS" dropdown, and a "Search..." input field. The main text explains that users can evaluate SonarQube using a traditional installation with the zip file or by spinning up a Docker container. It provides a list of steps for installing from the zip file: 1. Download and install Java 17, 2. Download the SonarQube Community Edition zip file, 3. As a non-root user, unzip it in, for example, C:\sonarqube or /opt/sonarqube, and 4. As a non-root user, start the SonarQube server. It also provides the commands to execute on Windows and other operating systems. A note states that if the instance fails to start, users should check their logs. The bottom of the page shows a "From the Docker image" section.

Once you're ready to set up a production instance, take a look at the [Install SonarQube](#) documentation.

Installing a local instance of SonarQube

You can evaluate SonarQube using a traditional installation with the [zip file](#) or you can spin up a Docker container using one of our [Docker images](#). Select the method you prefer below to expand the installation instructions:

From the zip file

1. Download and install [Java 17](#) on your system.
2. [Download](#) the SonarQube Community Edition zip file.
3. As a **non-root user**, unzip it in, for example, `C:\sonarqube` or `/opt/sonarqube`.
4. As a **non-root user**, start the SonarQube server:

```
# On Windows, execute:  
C:\sonarqube\bin\windows-x86-64\StartSonar.bat  
  
# On other operating systems, as a non-root user execute:  
/opt/sonarqube/bin/<OS>/sonar.sh console
```

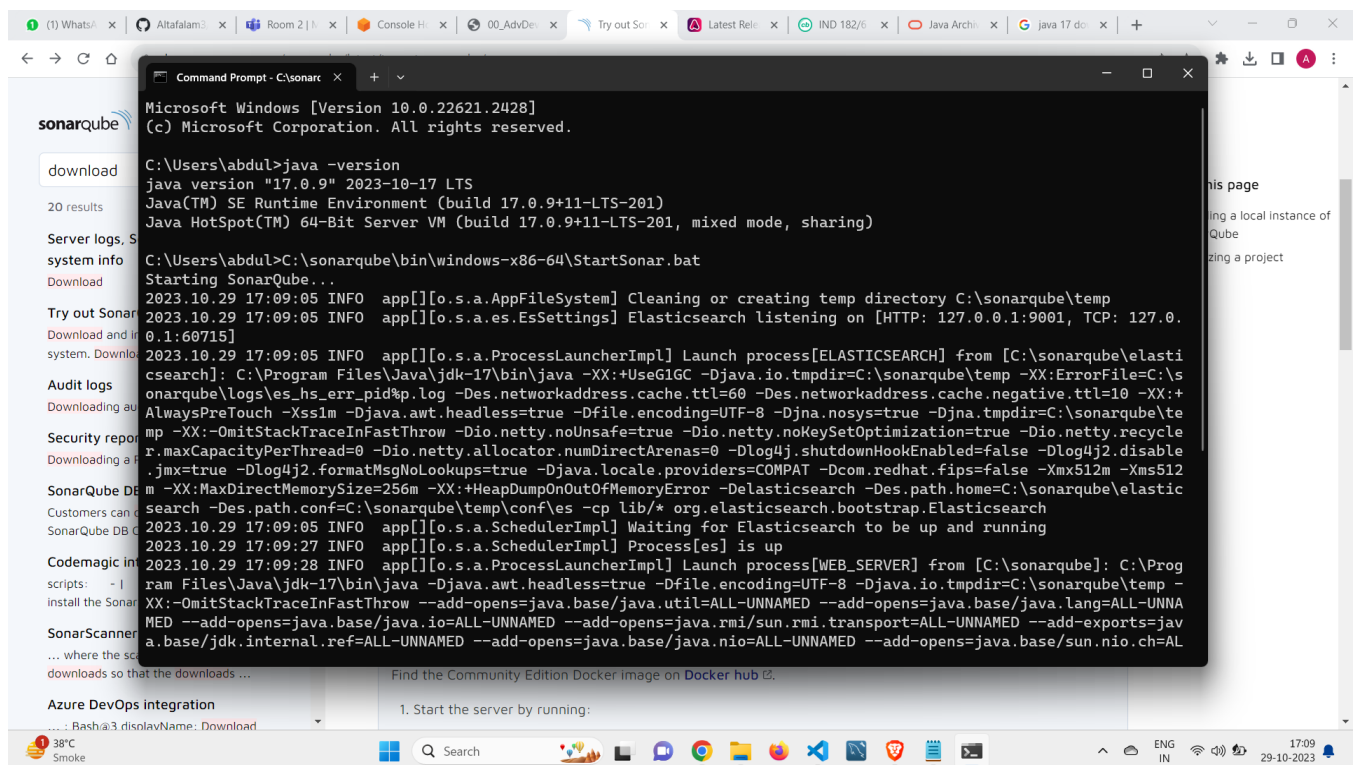
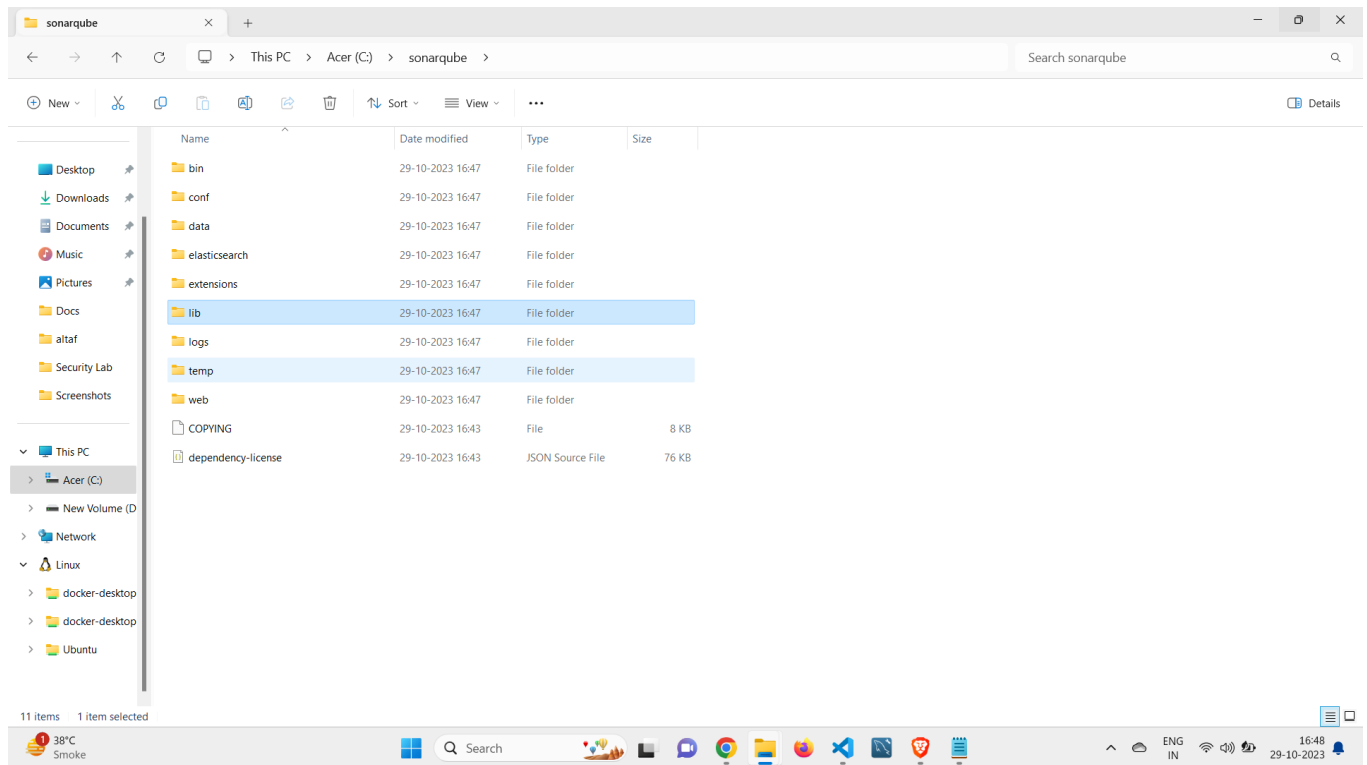
If your instance fails to start, check your [logs](#) to find the cause.

From the Docker image

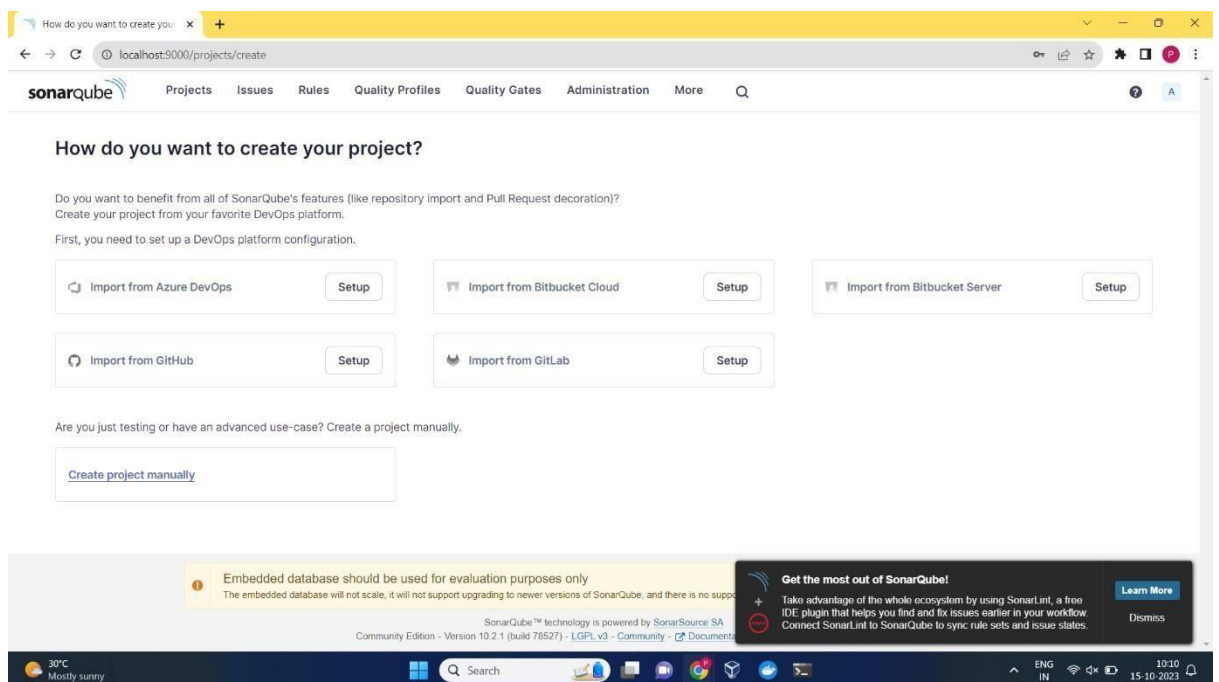
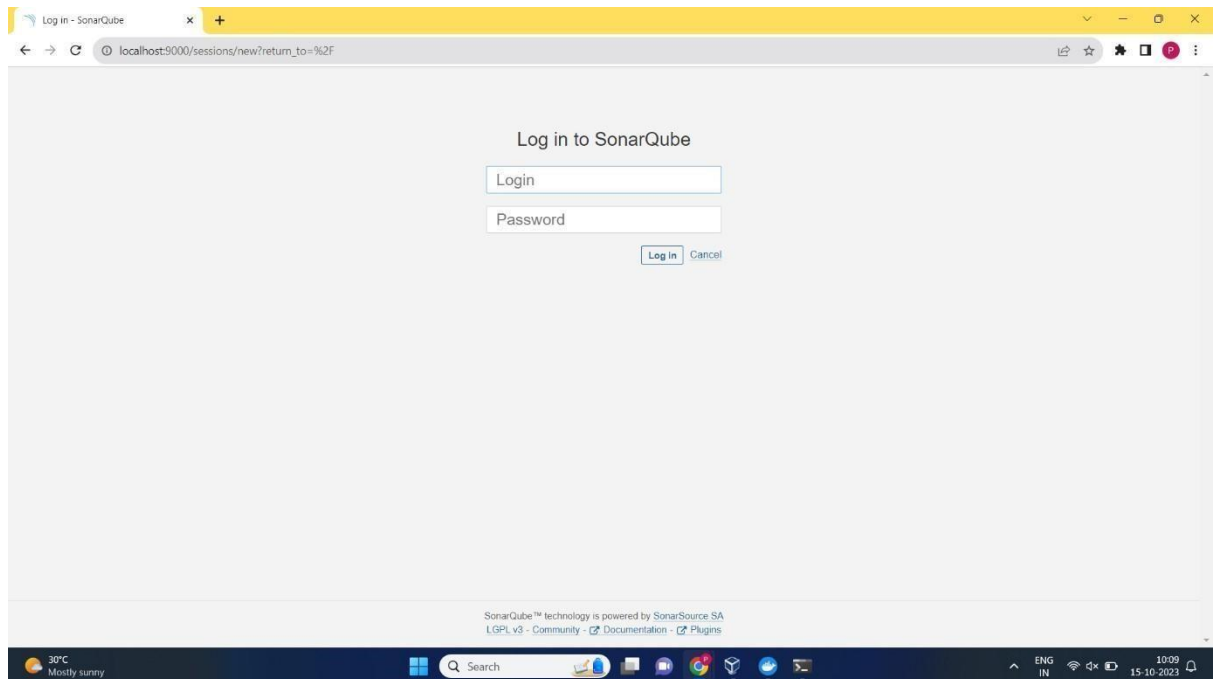
On this page

- Installing a local instance of SonarQube
- Analyzing a project

Unzip and save in C:/sonarqube



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



3. Create a project

Create a project

localhost:5000/projects/create?mode=manual

sonarqube

ProjectsIssuesRulesQuality ProfilesQuality GatesAdministrationMore

Create a project

Project display name *

sonarPythonProgram1

Up to 255 characters. Some scanners might override the value you provide.

Project key *

sonarPythonProgram1

The project key is a unique identifier for your project. It may contain up to 400 characters. Allowed characters are alphanumeric, '-' (dash), '_' (underscore), ':' (period) and ':' (colon), with at least one non-digit.

Main branch name *

main

The name of your project's default branch [Learn More](#)

Next

Embedded database should be used for evaluation purposes only

The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for it.

Get the most out of SonarQube!

Take advantage of the whole ecosystem by using SonarLint, a free IDE plugin that helps you find and fix issues earlier in your workflow. Connect SonarLint to SonarQube to sync rule sets and issue states.

[Learn More](#)

Dismiss

30°C Mostly sunny

Search

ENG IN 10:22 15-10-2023

Create a project

localhost:5000/projects/create?mode=manual&setncd=true

sonarqube

ProjectsIssuesRulesQuality ProfilesQuality GatesAdministrationMore

Set up project for Clean as You Code

The new code definition sets which part of your code will be considered new code. This helps you focus attention on the most recent changes to your project, enabling you to follow the Clean as You Code methodology. [Learn more: Defining New Code](#)

Choose the baseline for new code for this project

☒ Use the global setting

Previous version

Any code that has changed since the previous version is considered new code.

Recommended for projects following regular versions or releases.

☐ Define a specific setting for this project

☐ Previous version

Any code that has changed since the previous version is considered new code.

Recommended for projects following regular versions or releases.

☐ Number of days

Any code that has changed in the last x days is considered new code. If no action is taken on a new issue...

Recommended for projects following continuous delivery.

Get the most out of SonarQube!

Take advantage of the whole ecosystem by using SonarLint, a free IDE plugin that helps you find and fix issues earlier in your workflow. Connect SonarLint to SonarQube to sync rule sets and issue states.

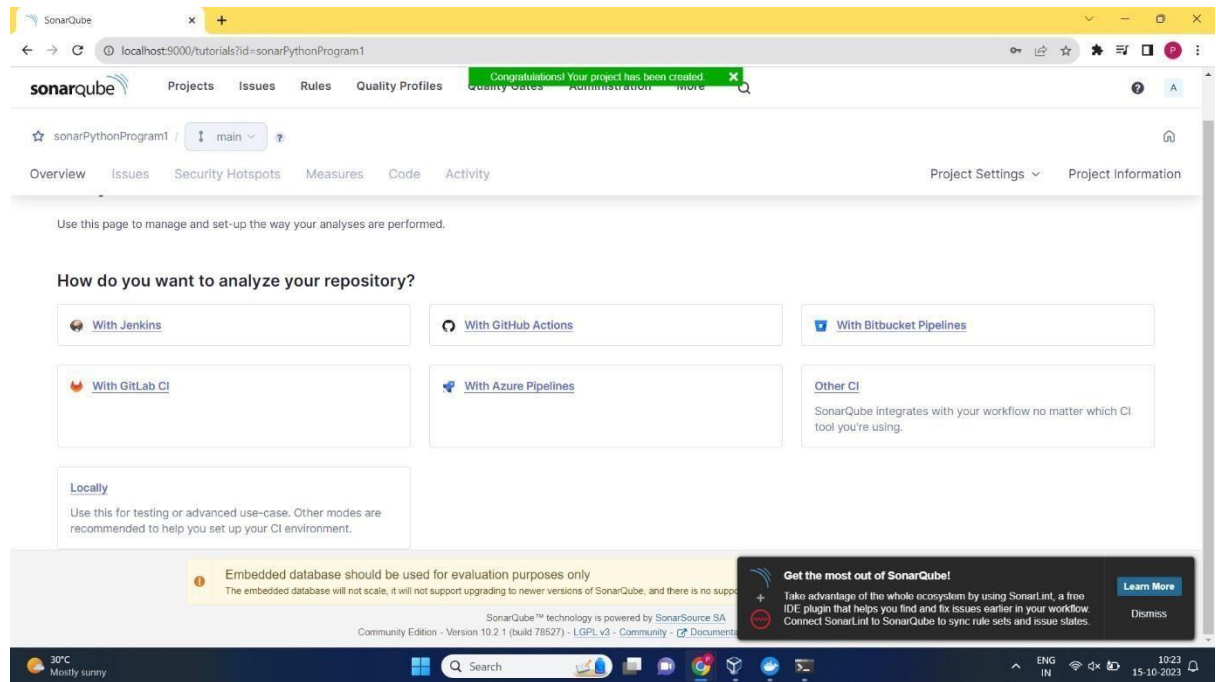
[Learn More](#)

Dismiss

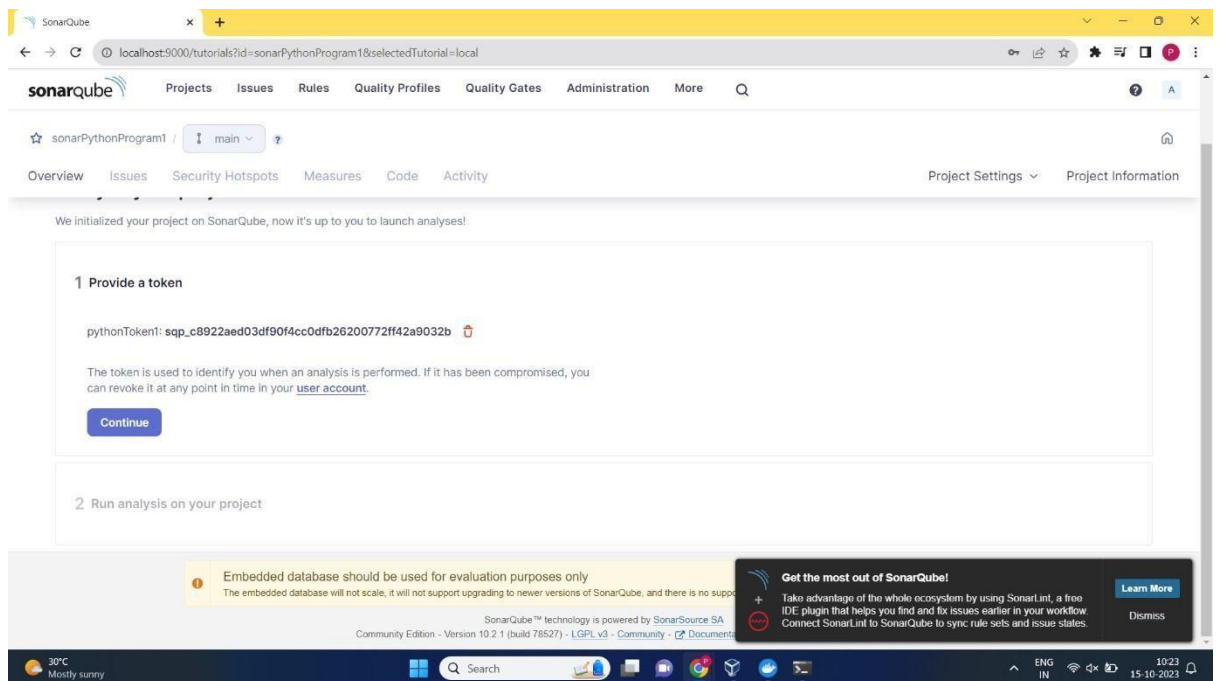
30°C Mostly sunny

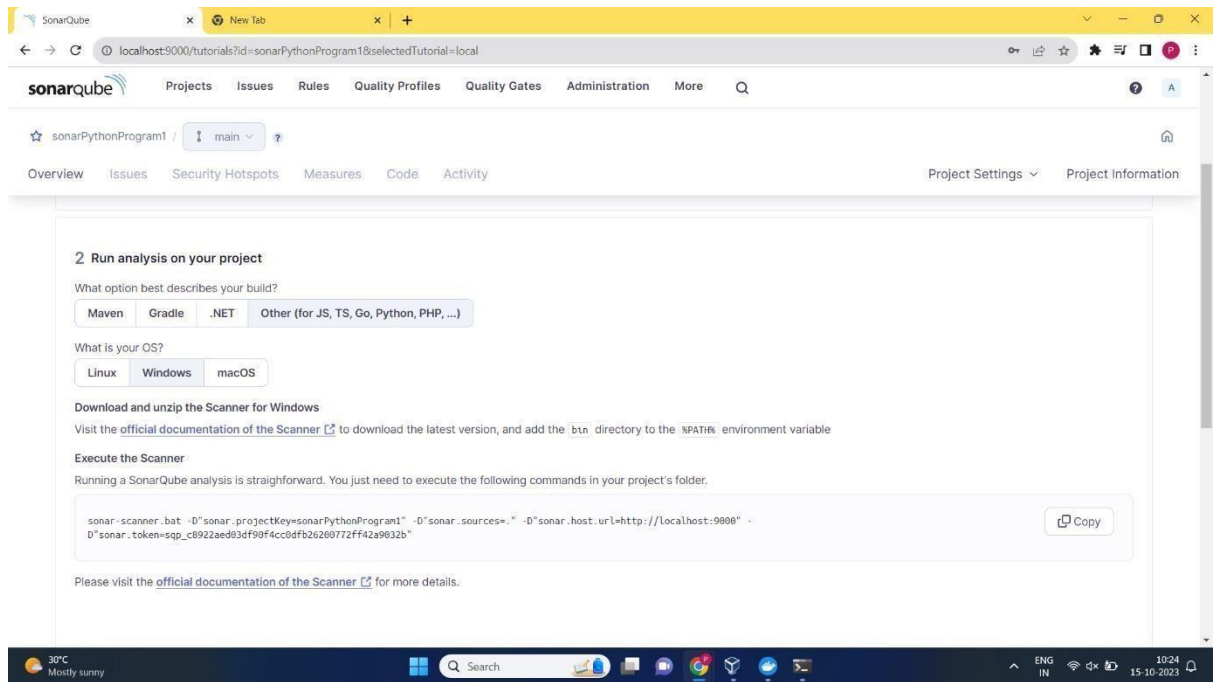
Search

ENG IN 10:22 15-10-2023

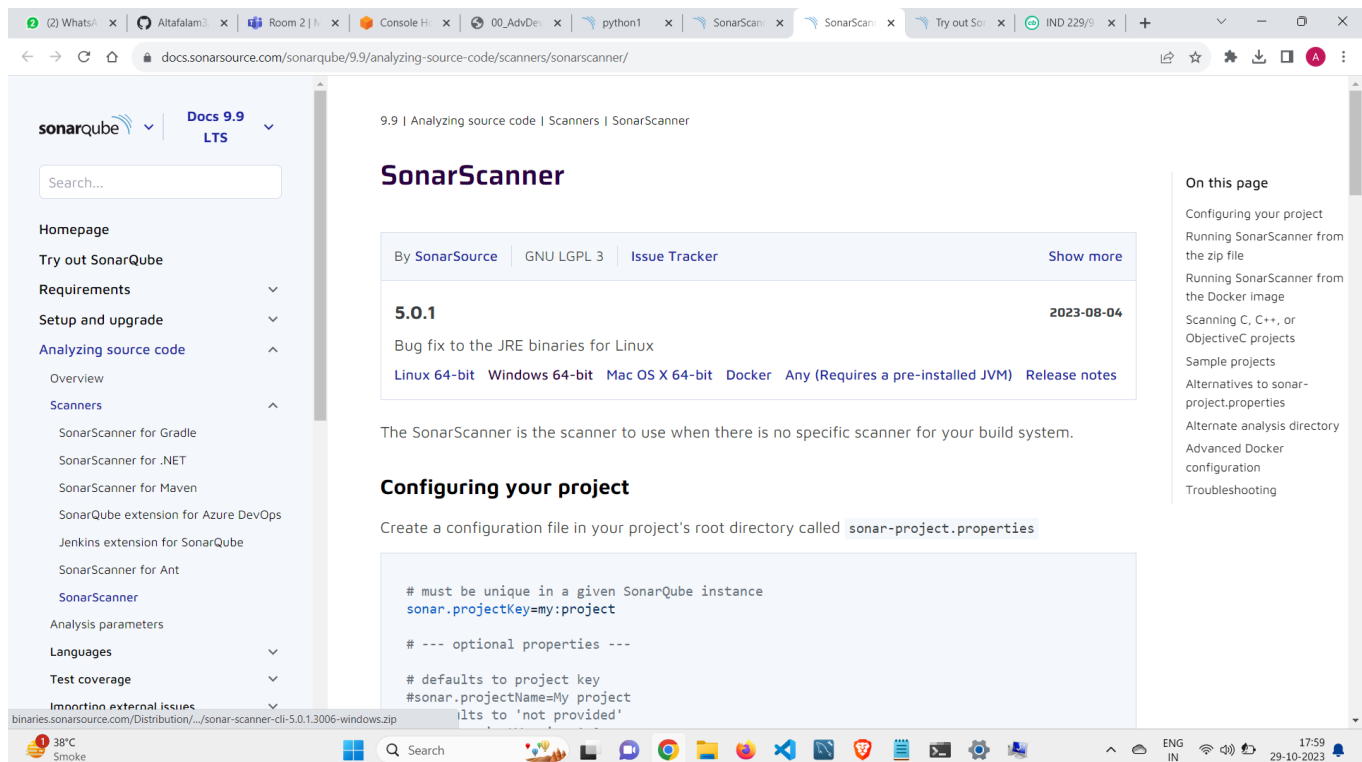


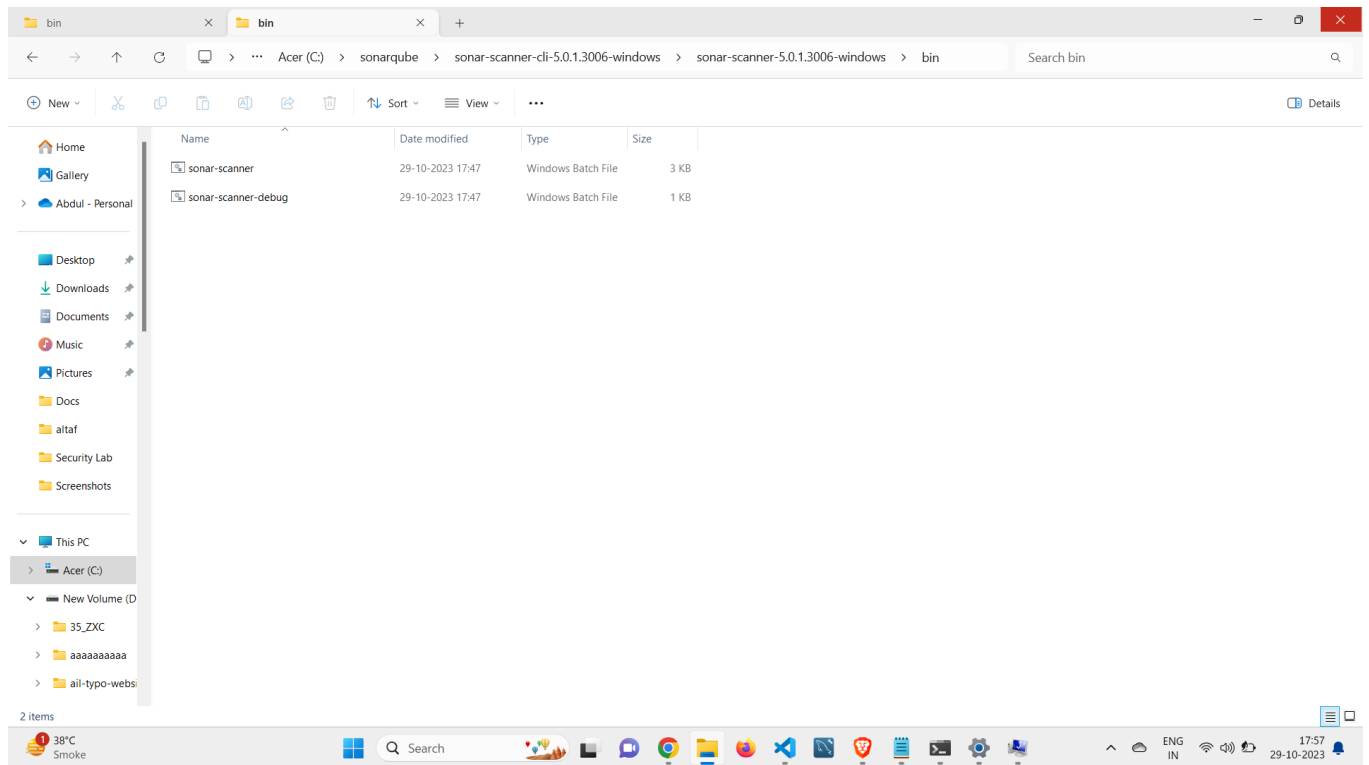
4. Provide token



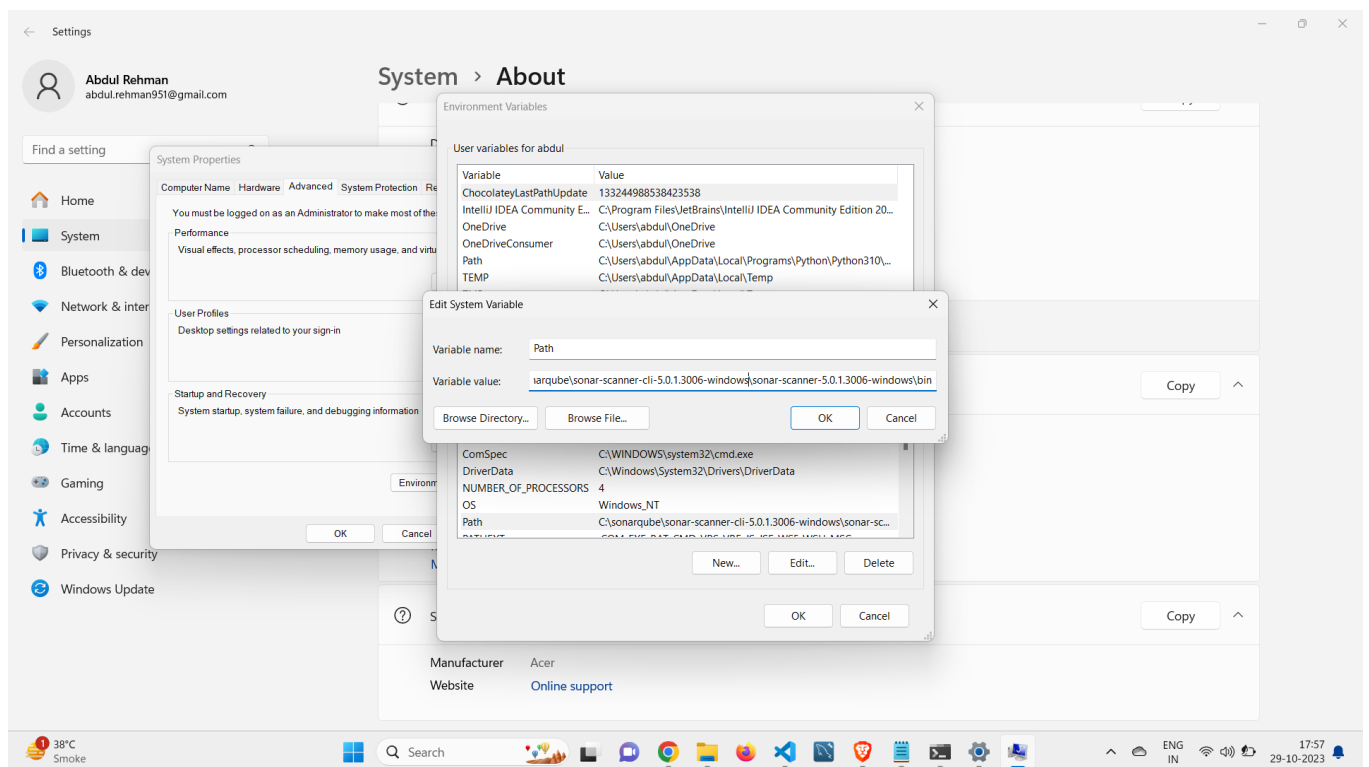


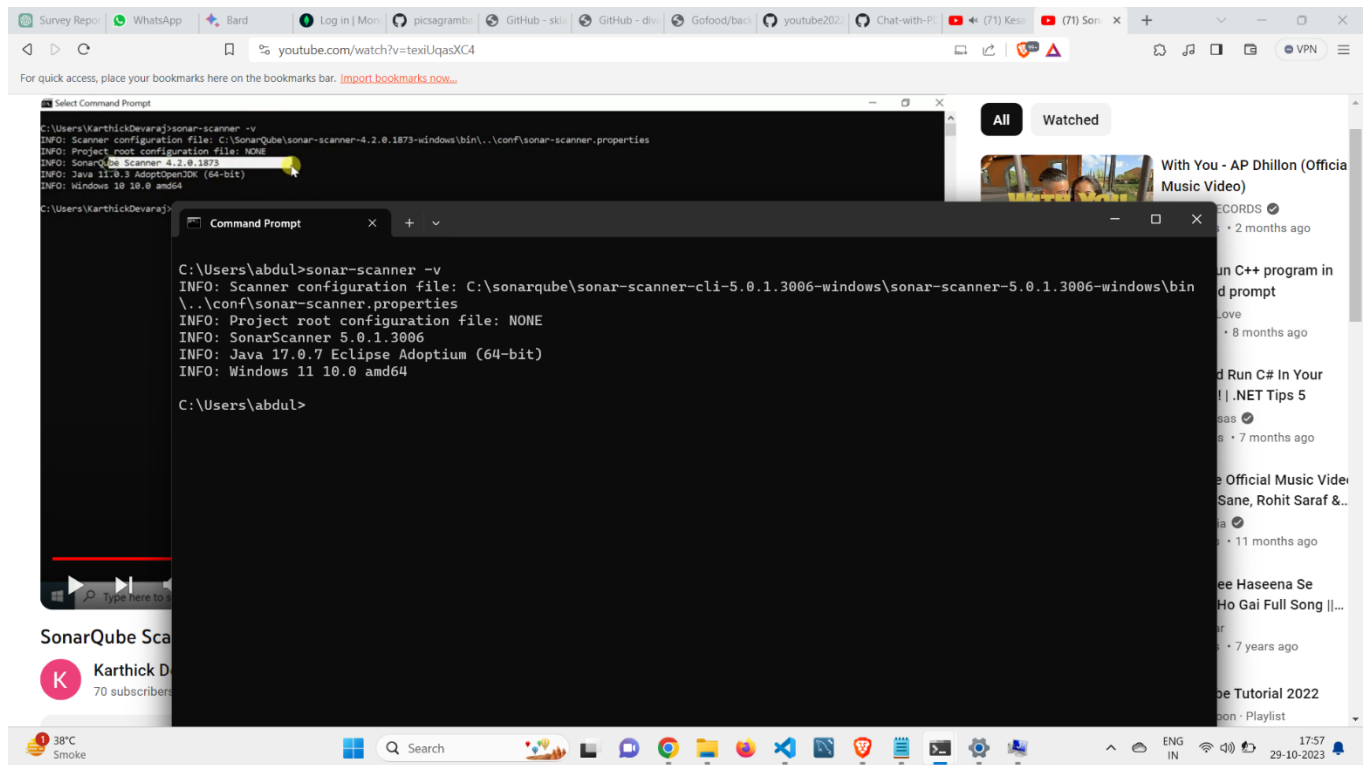
5. Install and setup sonar scanner cli



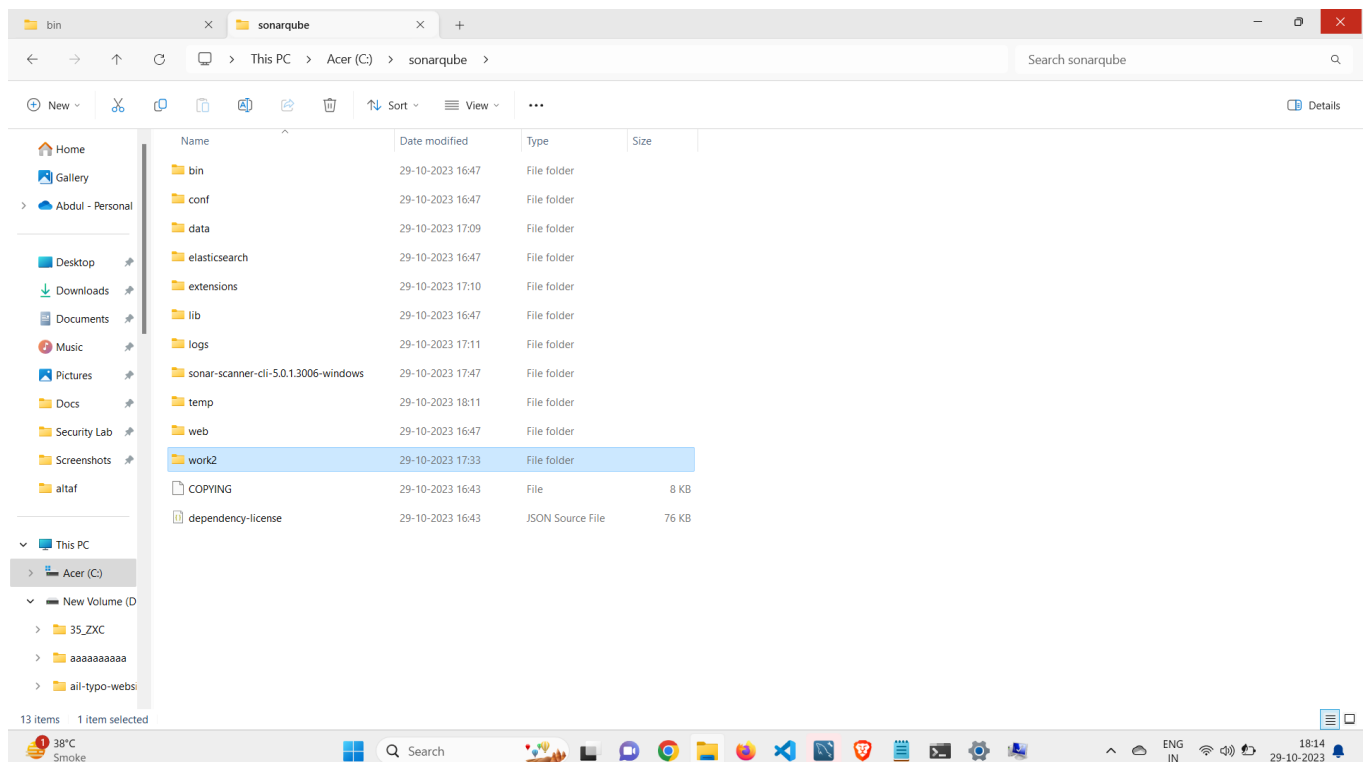


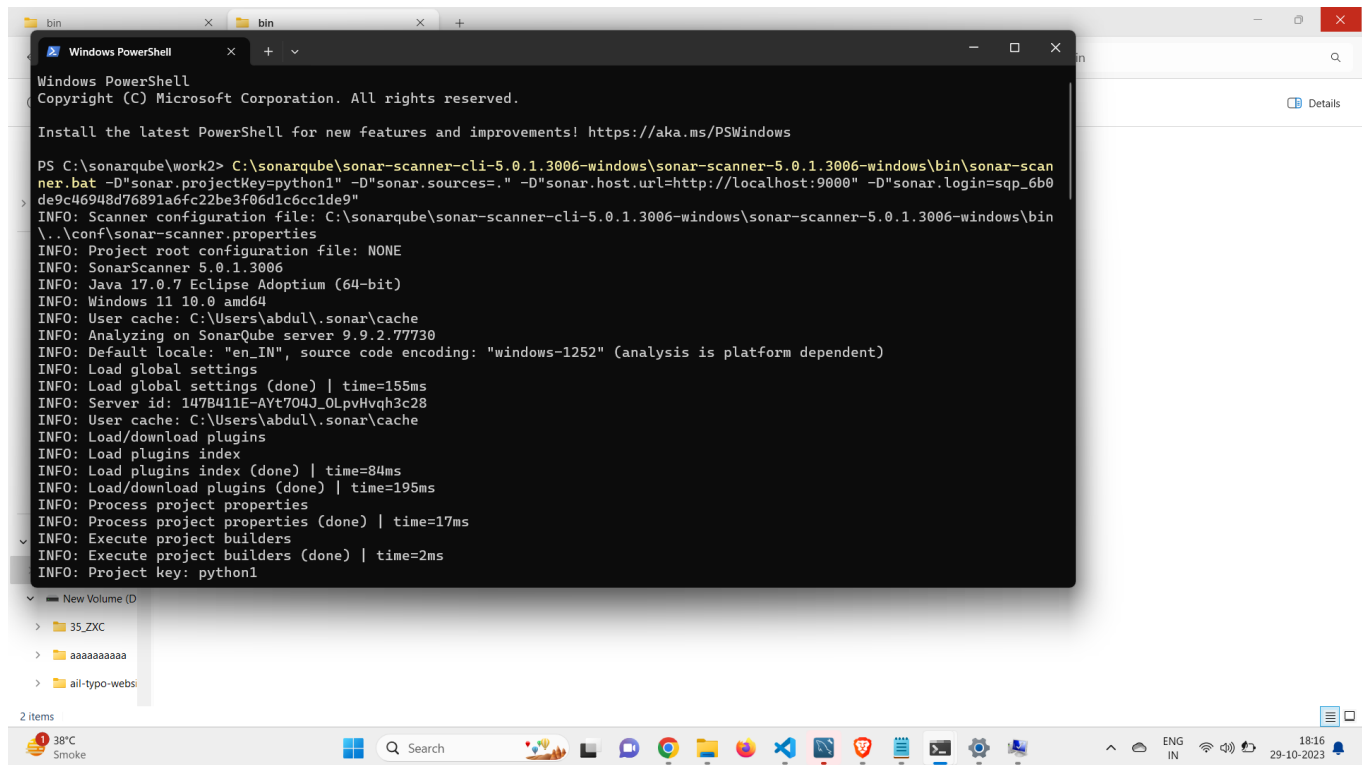
Unzip it in sonarqube folder and take this bin location and save in env variable system properties



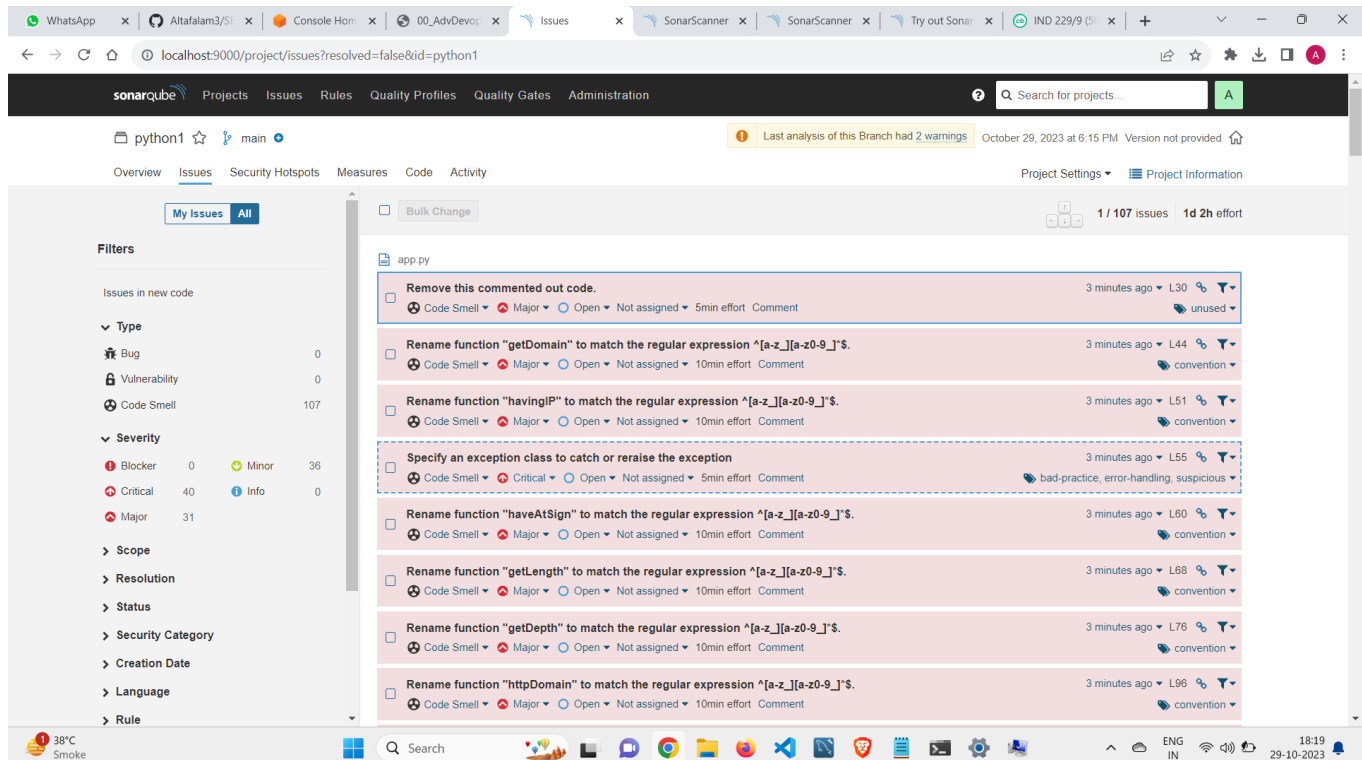


Bin location + the link of bat in snarqube project run in python directory where testing is needed





6. See the result of the test



CONCLUSION:

Here we have successfully performed static analysis of python programs.

