Experiment 7

Aim :- To perform static analysis on python programs using sonarqube SAST process

Theory :-

SonarQube is a universal tool for static code analysis that has become more or less the industry standard. Keeping code clean, simple, and easy to read is also a lot easier with SonarQube.

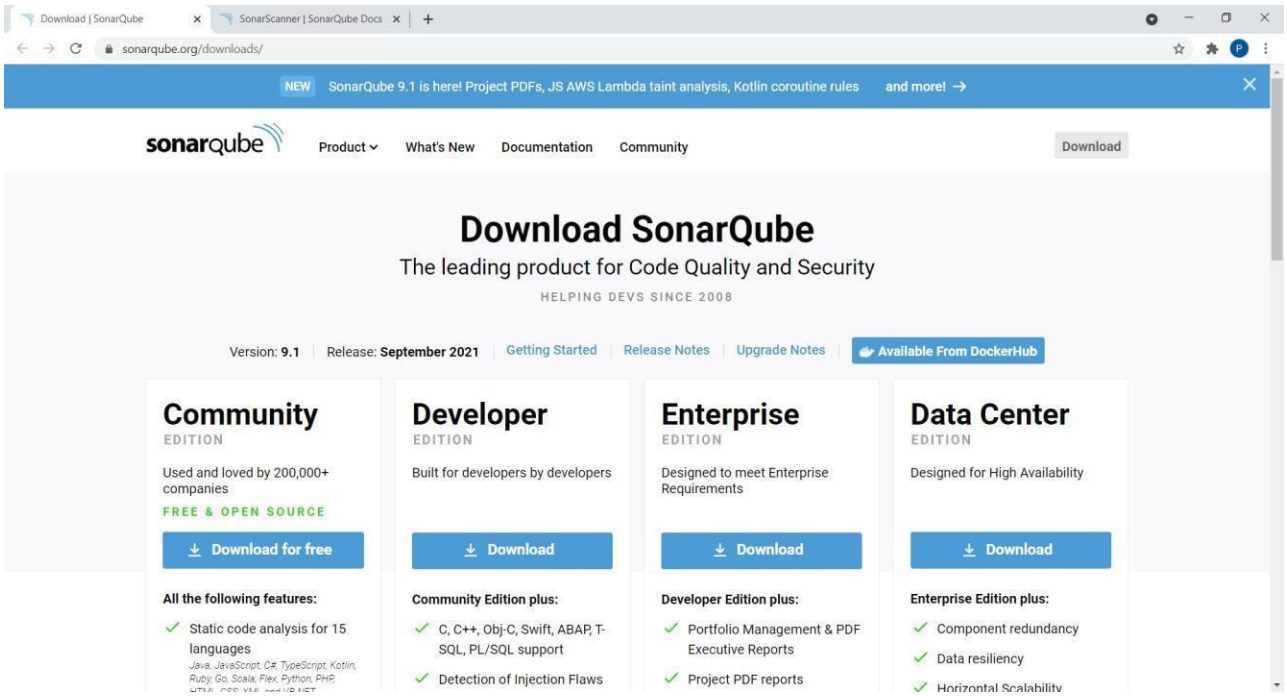
What is SonarQube?
SonarQube is an open-source platform developed by SonarSource for continuous inspection of code quality. Sonar does static code analysis, which provides a detailed report of bugs, code smells, vulnerabilities, code duplications. It supports 25+ major programming languages through built-in rulesets and can also be extended with various plugins.

Benefits of SonarQube
Sustainability - Reduces complexity, possible vulnerabilities, and code duplications, optimising the life of applications. Increase productivity - Reduces the scale, cost of maintenance, and risk of the application; as such, it removes the need to spend more time changing the code
Quality code - Code quality control is an inseparable part of the process of software development.
Detect Errors - Detects errors in the code and alerts developers to fix them automatically before submitting them for output.
Increase consistency - Determines where the code criteria are breached and enhances the quality
Business scaling - No restriction on the number of projects to be evaluated
Enhance developer skills - Regular feedback on quality problems helps developers to improve their coding skills

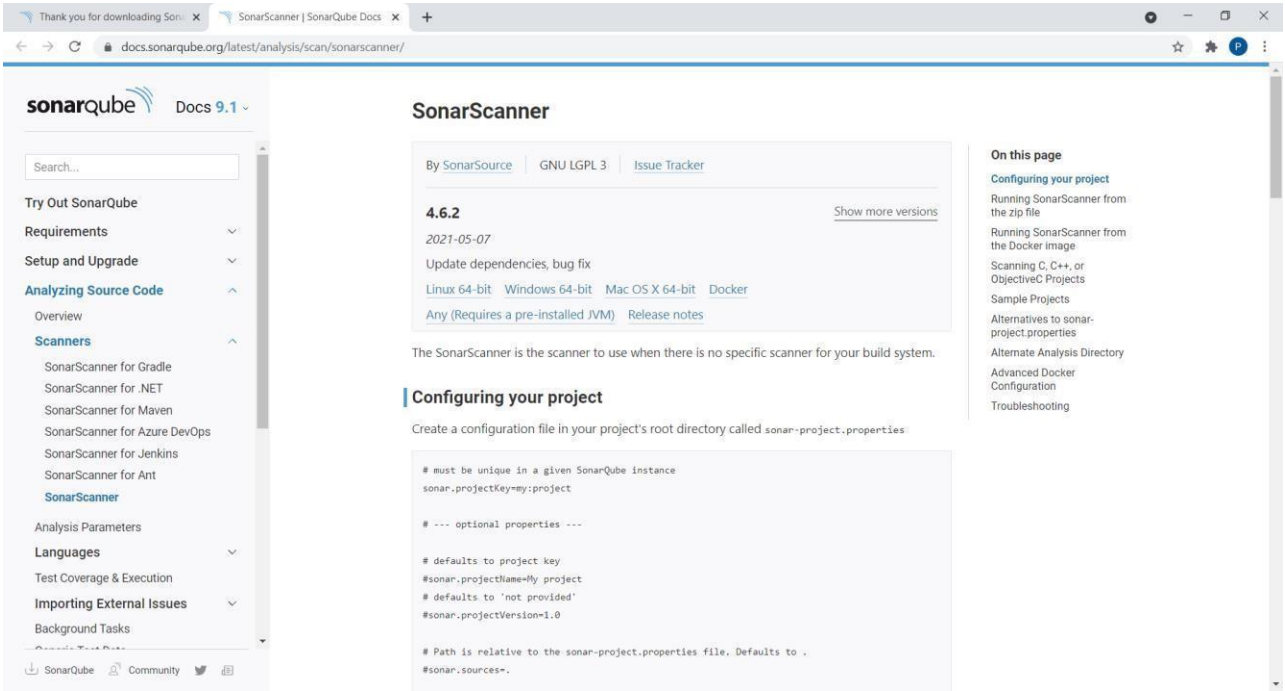
Why SonarQube?
Developers working with hard deadlines to deliver the required functionality to the customer. It is so important for developers that many times they compromise with the code quality, potential bugs, code duplications, and bad distribution of complexity. Additionally, they tend to leave unused variables, methods, etc. In this scenario, the code would work in the desired way.
To avoid these issues in code, developers should always follow the good coding practice, but sometimes it is not possible to follow the rules and maintain the good quality as there may be many reasons.
In order to achieve continuous code integration and deployment, developers need a tool that not only works once to check and tell them the problems in the code but also to track and control the code to check continuous code quality. To satisfy all these requirements, here comes SonarQube in the picture.

Steps :-

1)Download sonarqube

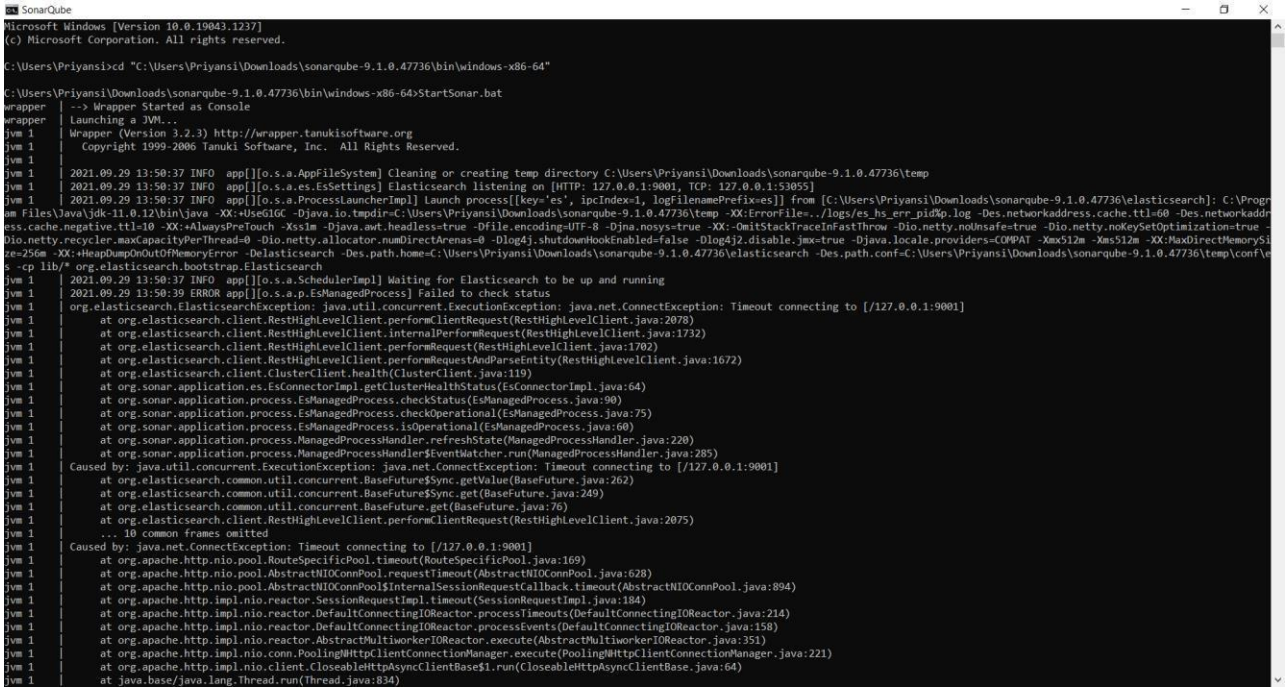


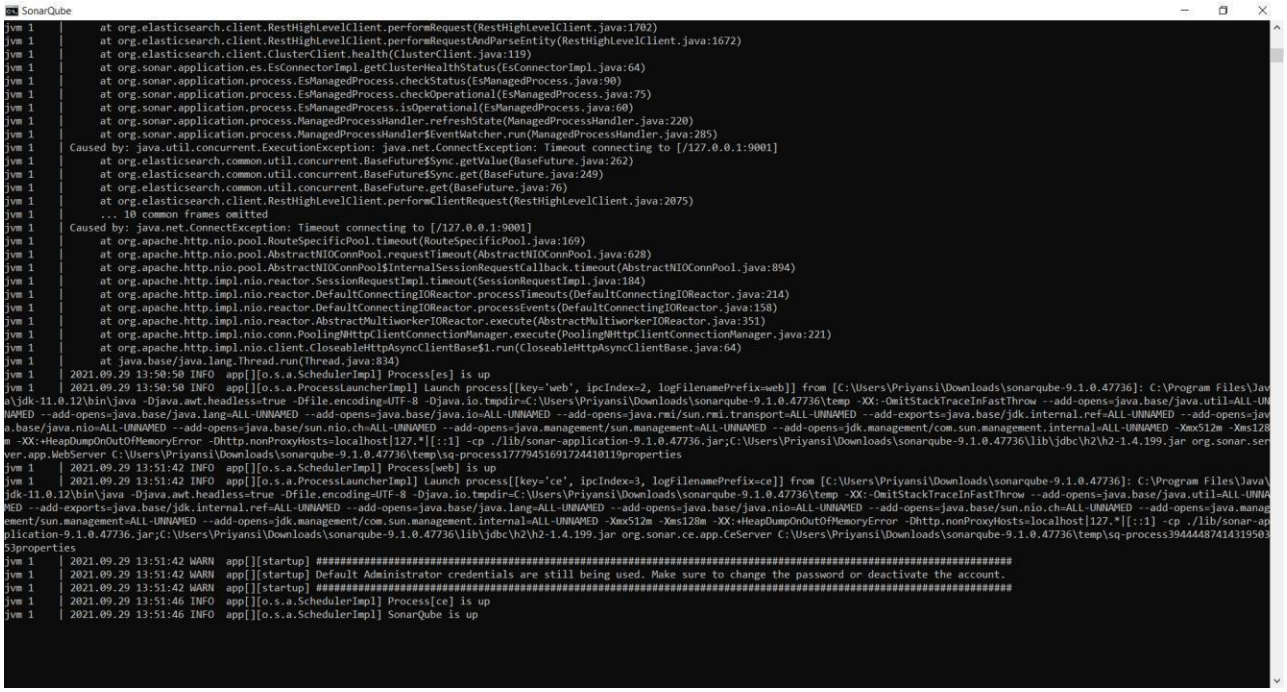
2) After downloading, set Environment Variables. Add "sonarqube-9.1.0.47736\bin" to Path.



3) Open command prompt. Run commands:

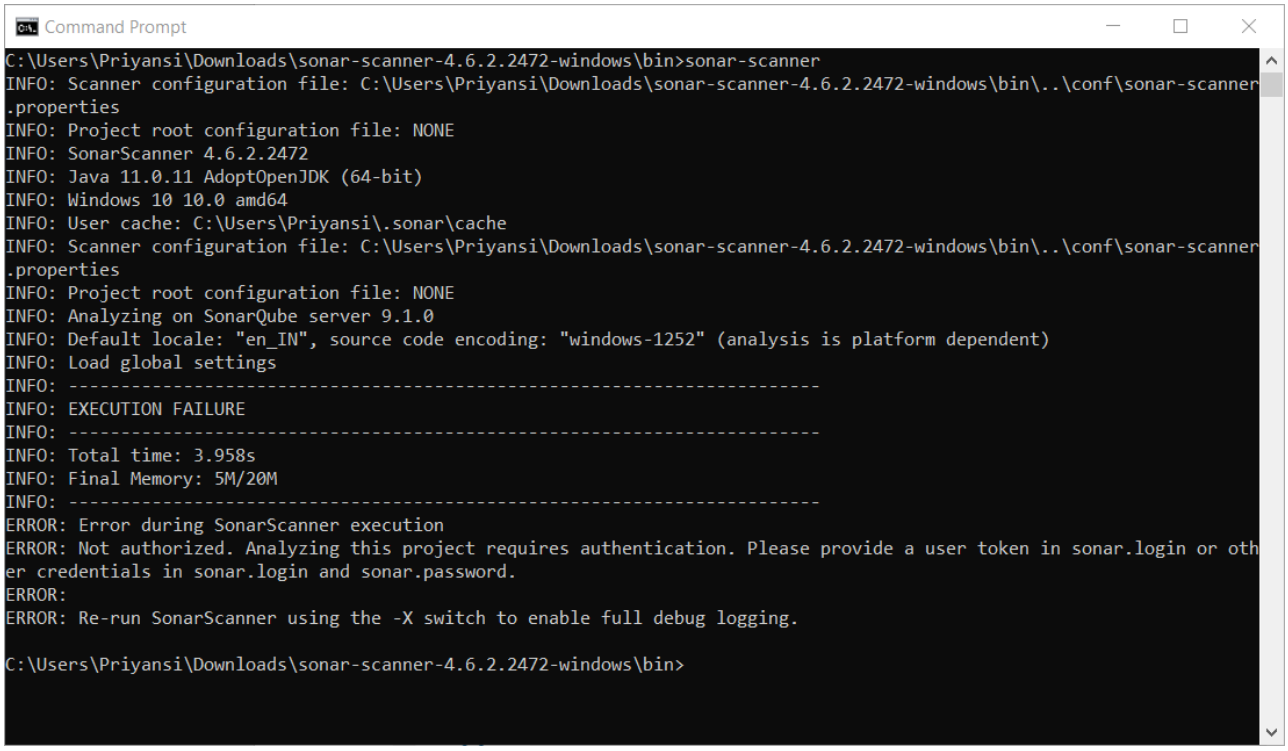
- **cd “sonarqube-9.1.0.47736\bin\windows-x86-64”**
- **StartSonar.bat**



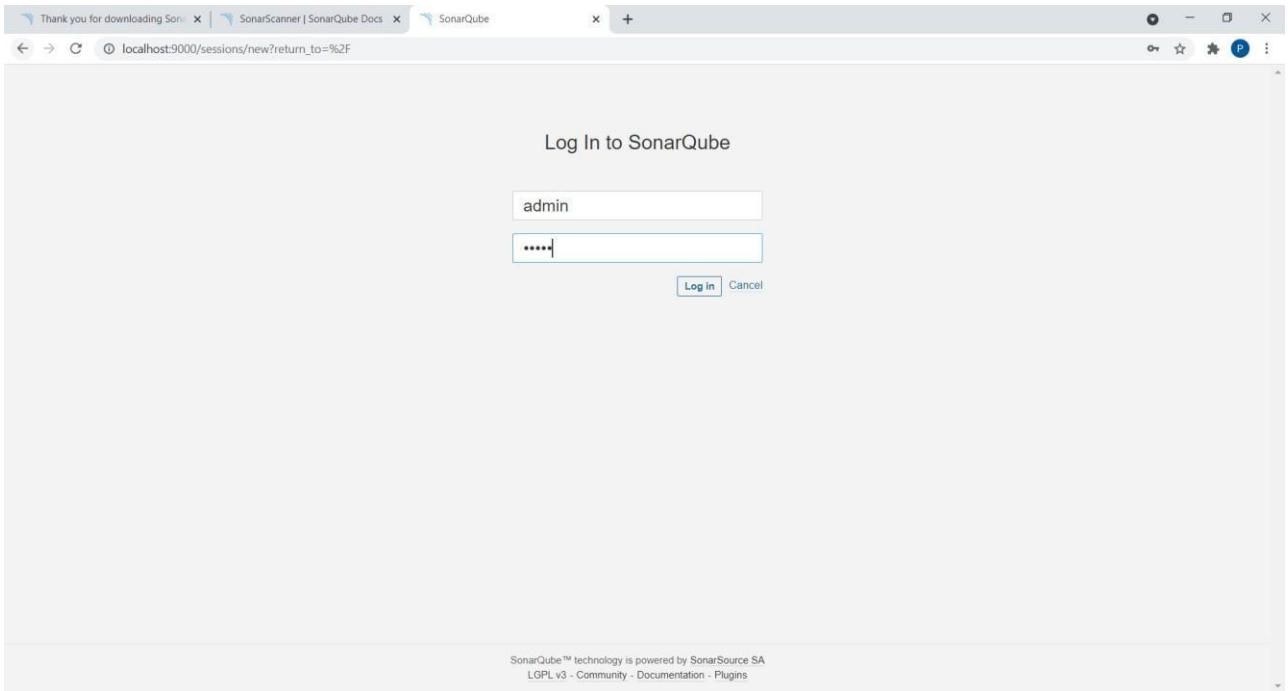


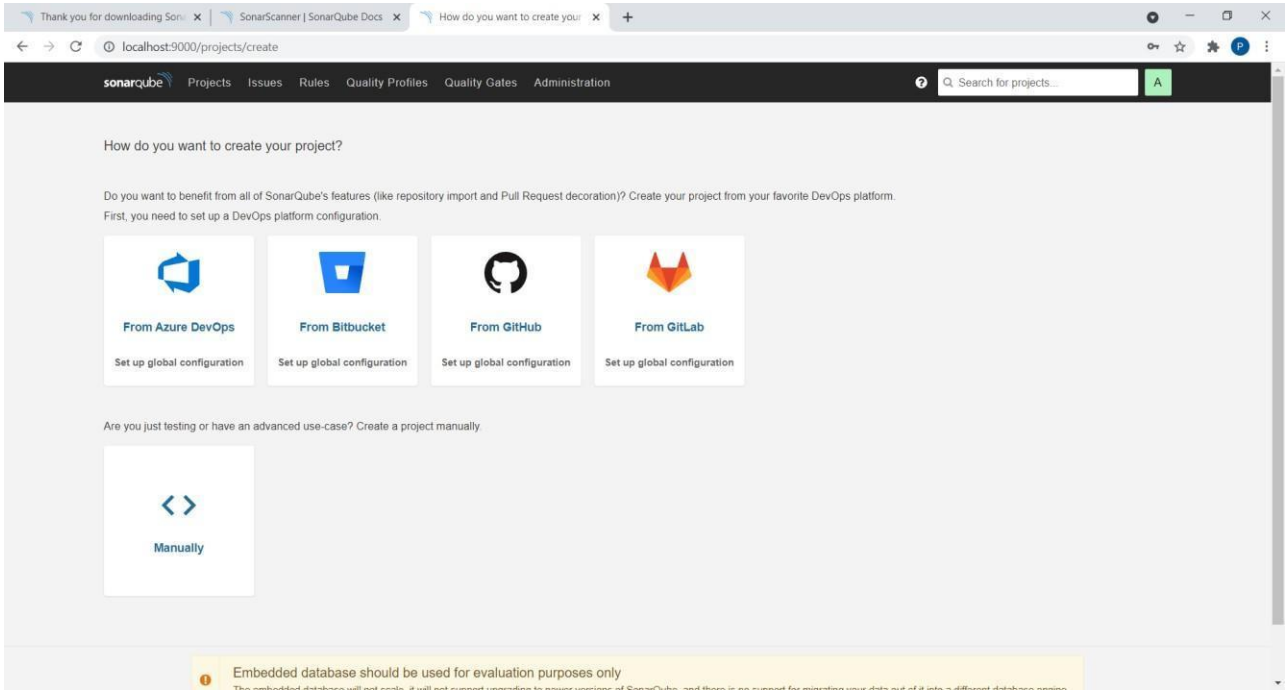
4) Open another command prompt. Run command:

- **cd “sonar-scanner-4.6.2.2472-windows\bin”**
- **sonar-scanner**

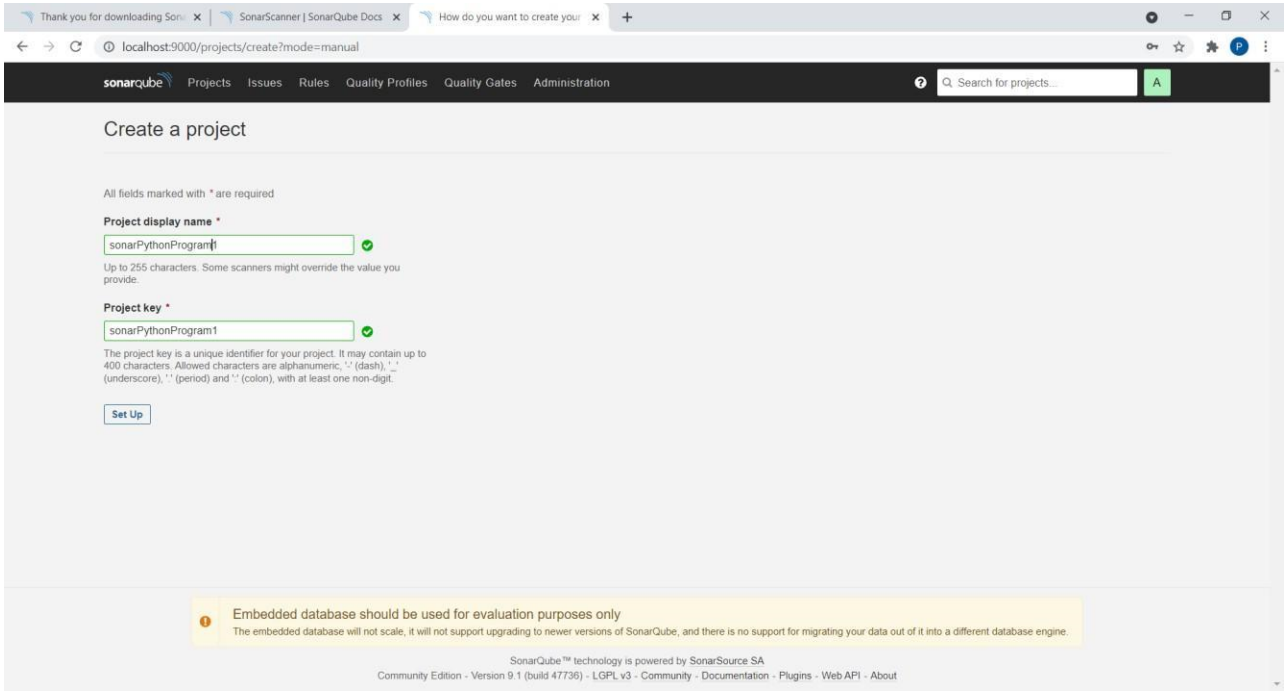


5) Server up and running on localhost:9000
Login using credentials as User: admin and Password: admin and Set a new password

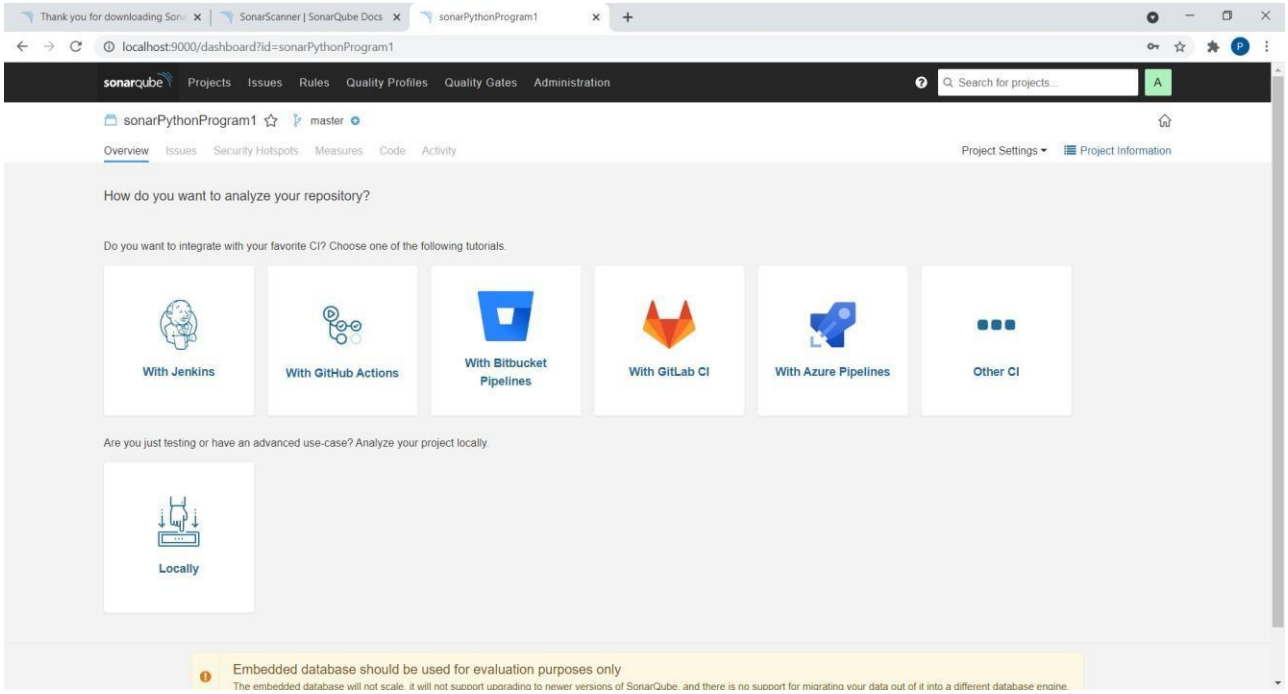




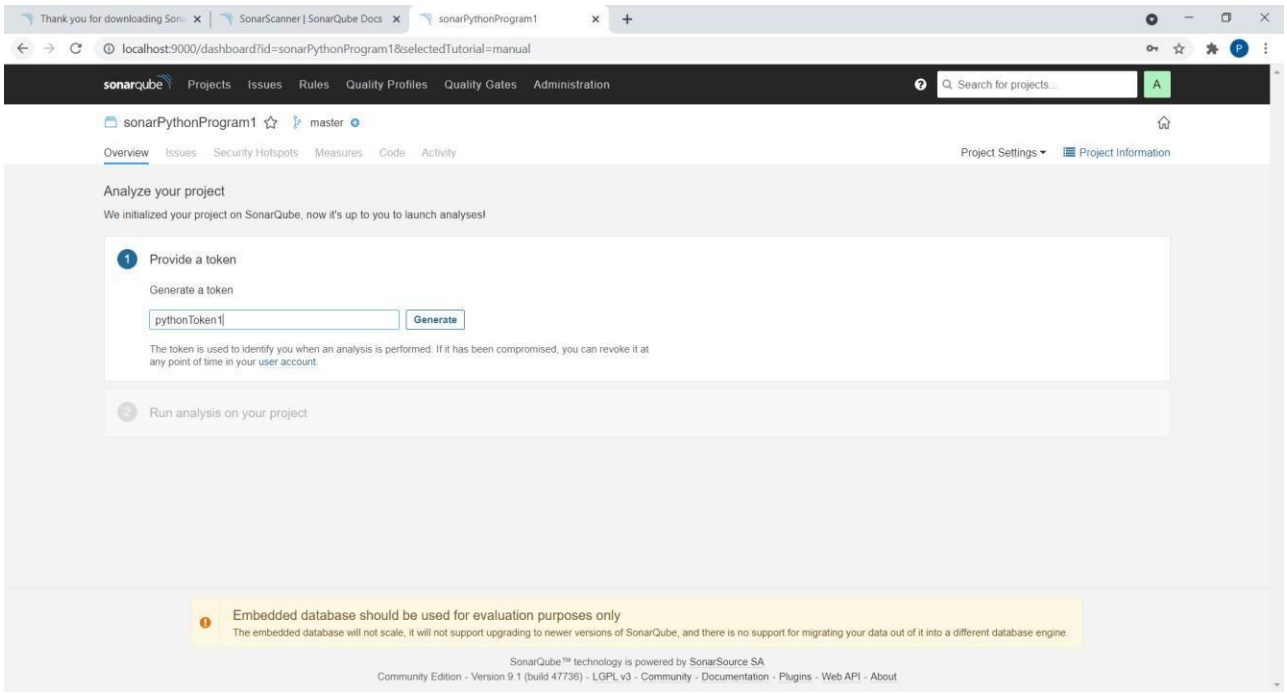
6) Click on Create a project Manually.



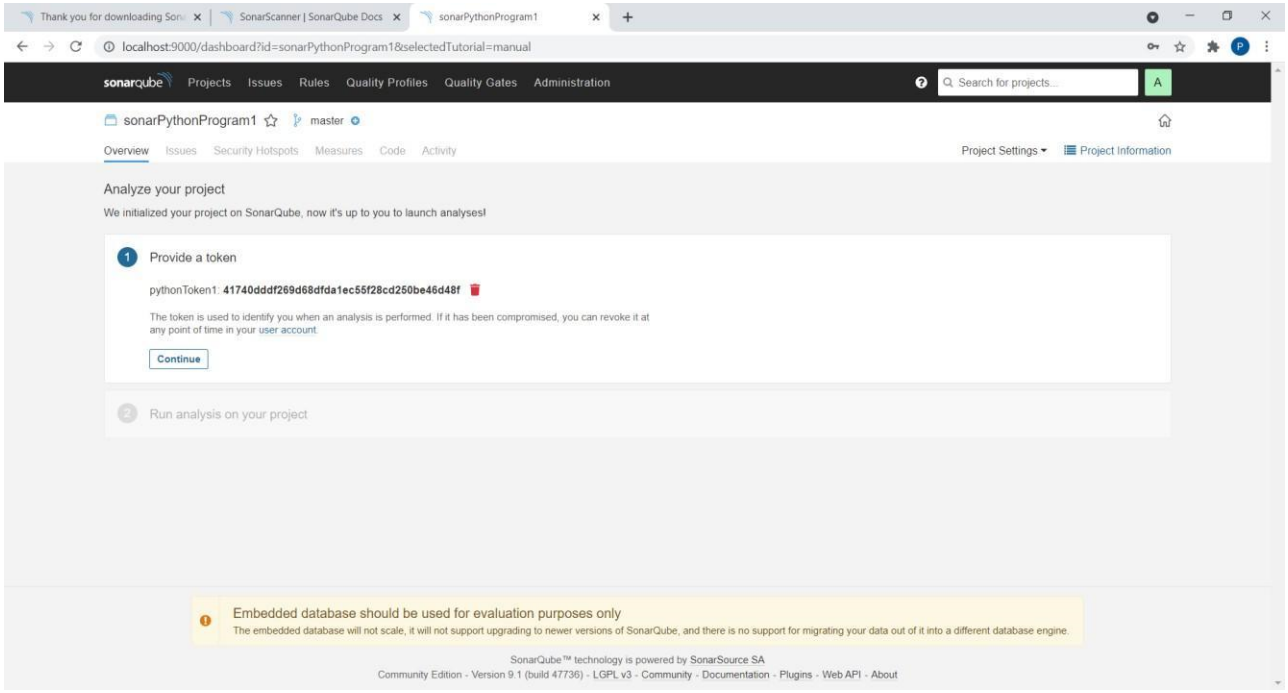
7) Give any Project display name.



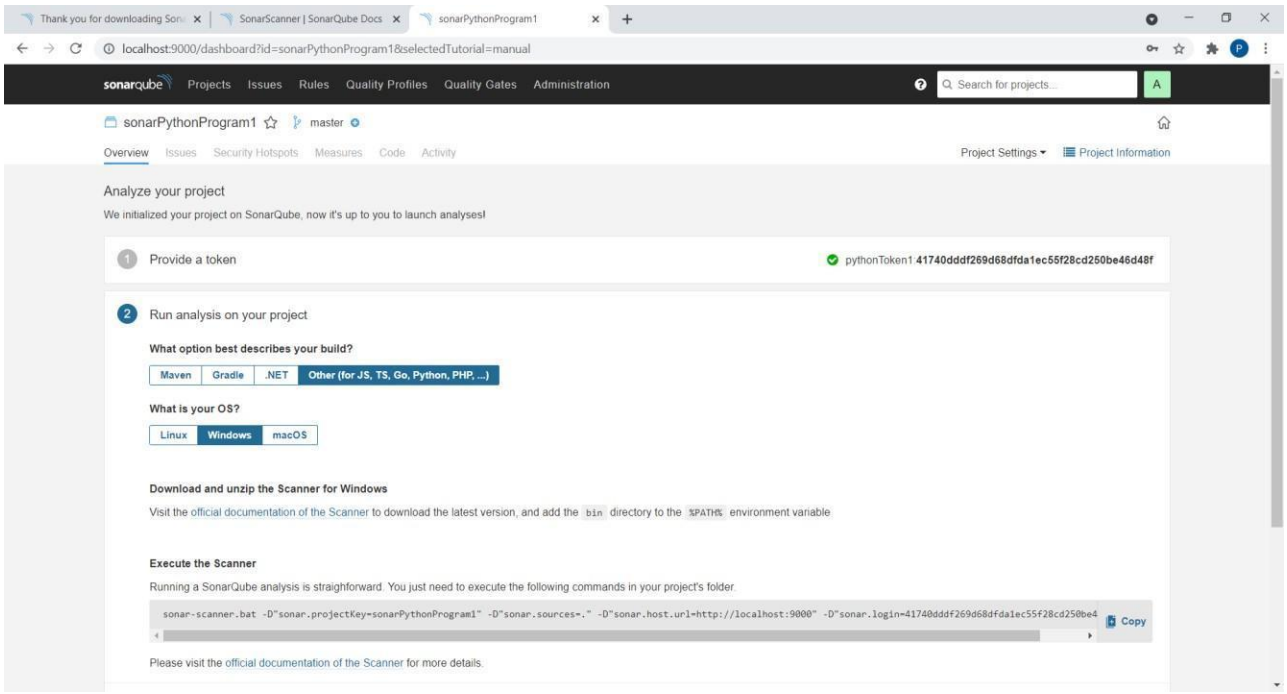
Click on Locally.



9) Give any name to token and click on Generate.



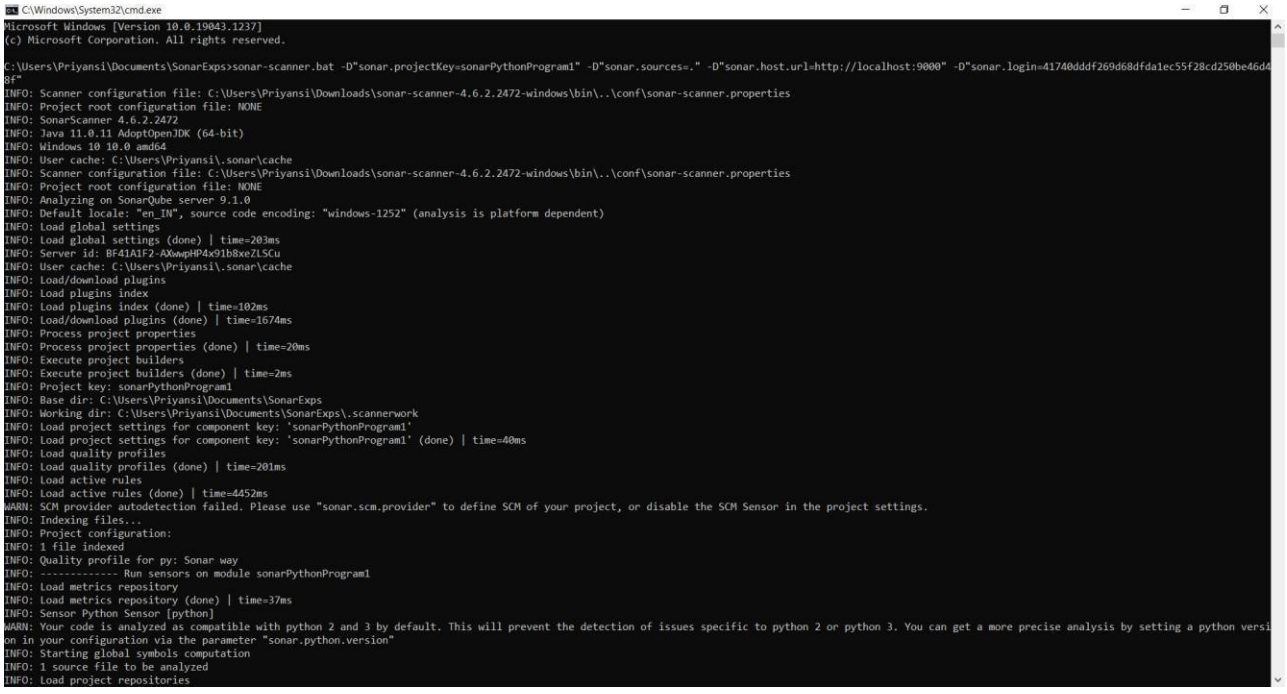
Click on Continue.

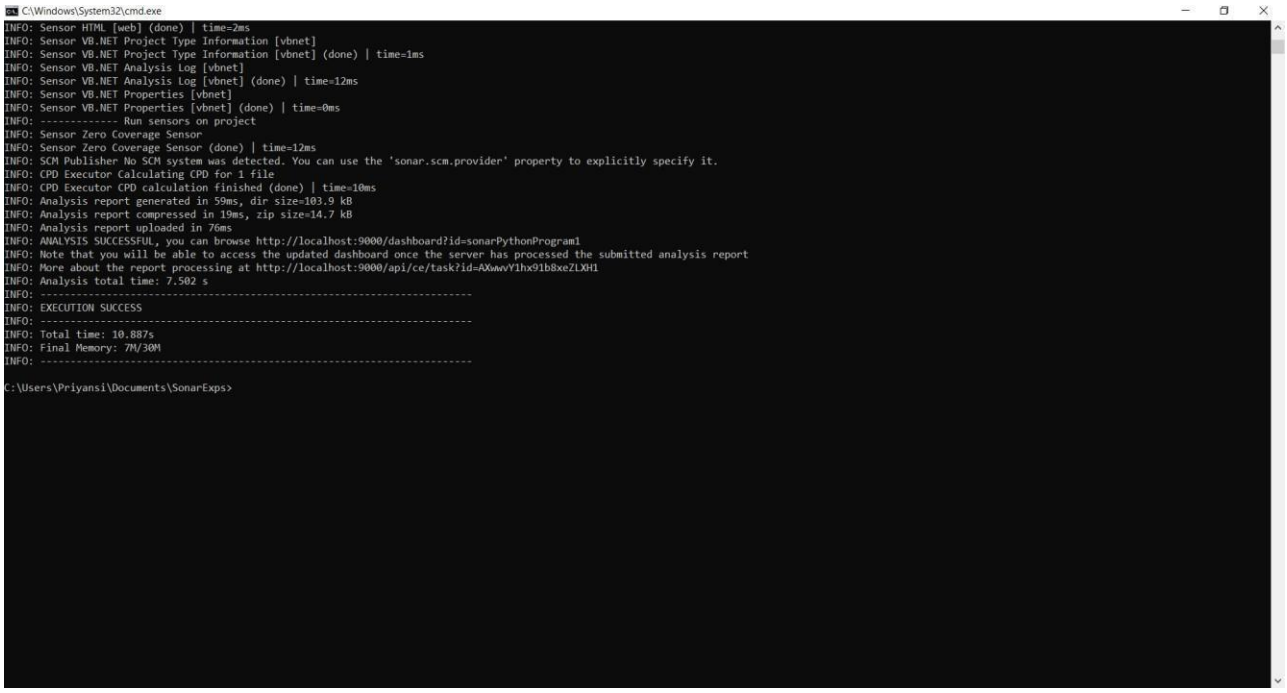


10) Save a Python program in a folder. class Solution(object):
def romanToInt(self, s):

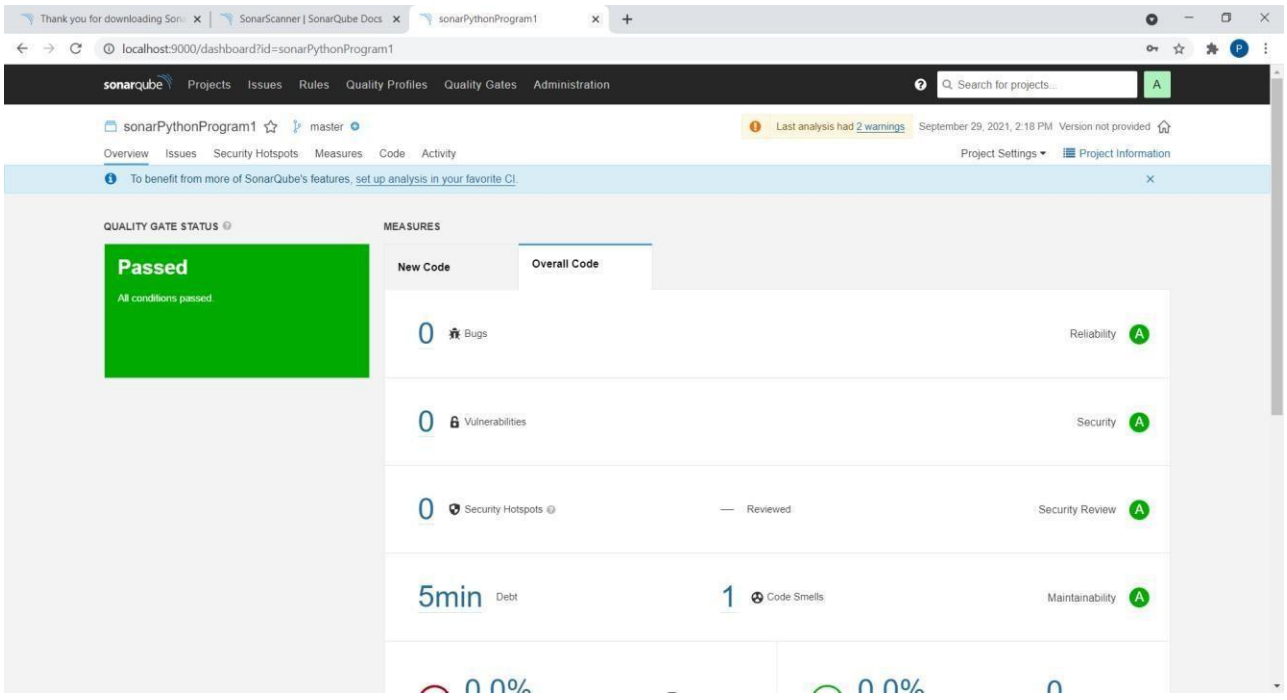
```
roman =
 {'I':1,'V':5,'X':10,'L':50,'C':100,'D':500,'M':1000,'IV':4,'IX':9,'XL':40,'XC':90,'CD':400,'CM':
 900}
      i = 0
      num = ''
      while i < len(s):
        if i+1<len(s) and s[i:i+2] in roman:
          num+=roman[s[i:i+2]]
          i+=2
        else:
          #print(i)
          num+=roman[s[
          i]] i+=1
      return num
 ob1 =
 Solution()
 print(ob1.romanToInt("III"))
 print(ob1.romanToInt("CDXL
 III"))
```

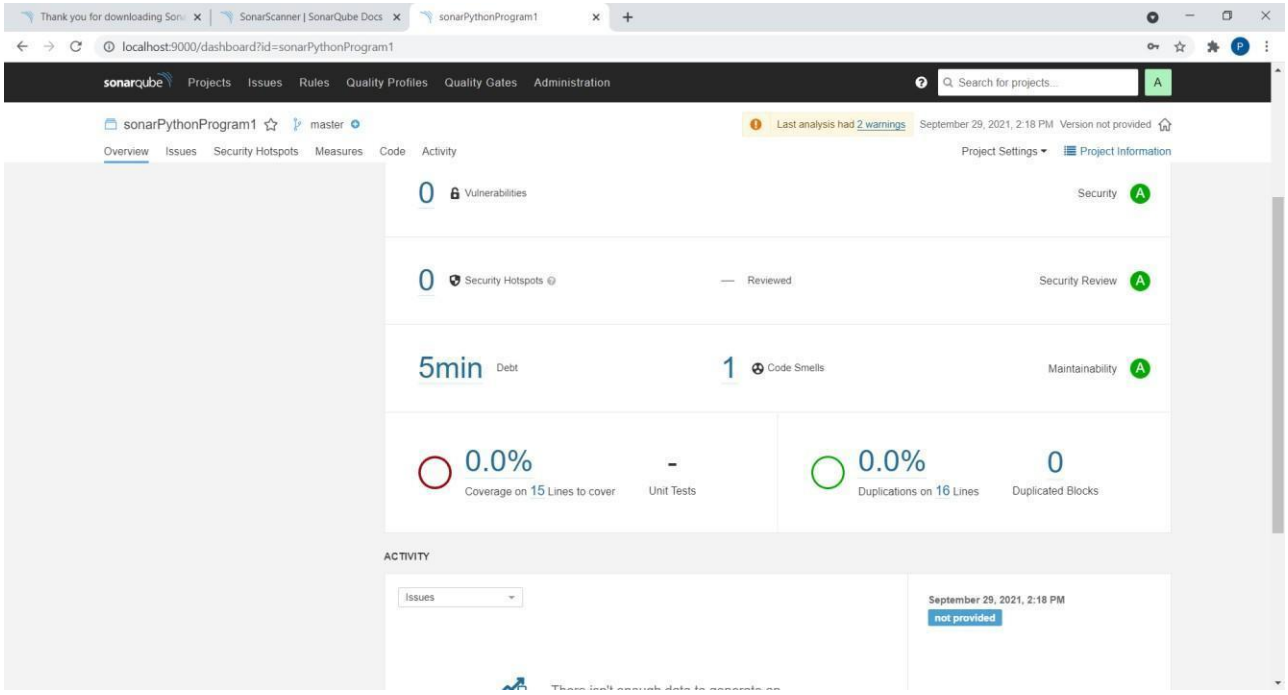
11) Open command prompt in this folder and Run program using copied command. “sonar-scanner.bat -D"sonar.projectKey=sonarPythonProgram1" -D"sonar.sources=." -D"sonar.host.url=http://localhost:9000" -D"sonar.login=41740dddf269d68dfda1ec55f28cd250be46d48f"

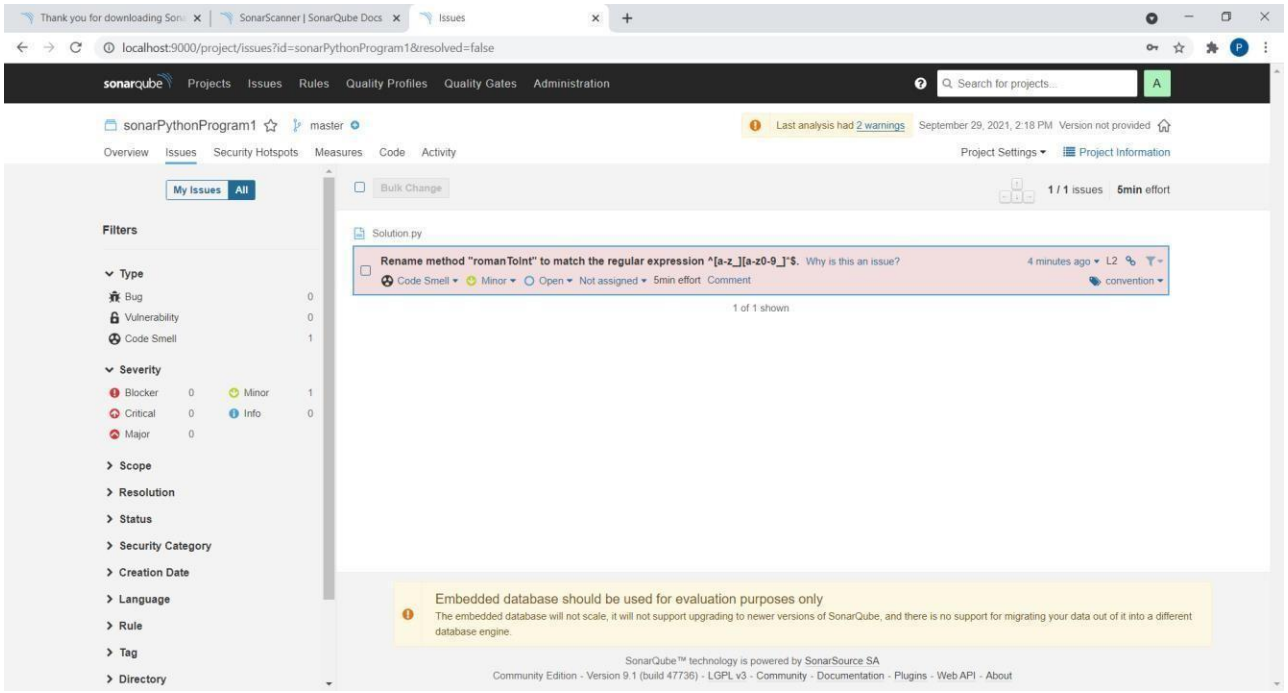




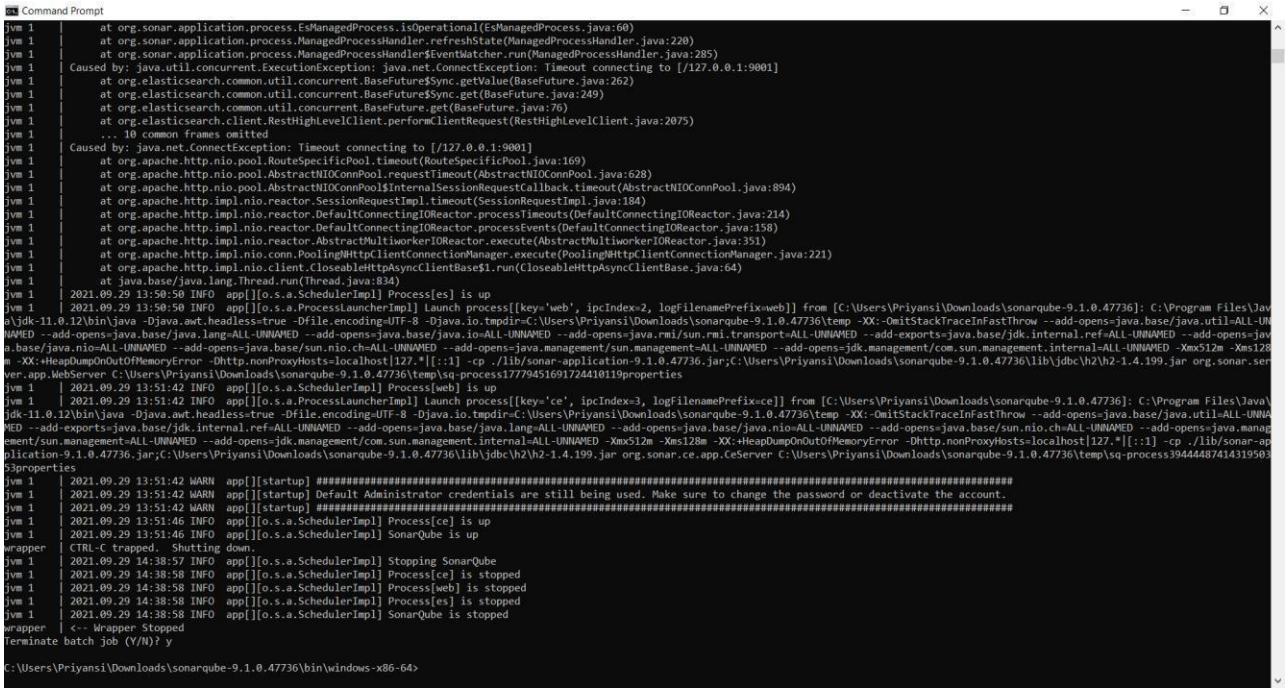
13) Given below is the inspection of code quality to perform automatic reviews with static analysis of code to detect bugs, code smells, and security vulnerabilities.







Press “Ctrl + C” to stop the server.



Conclusion :-

The SonarQube SAST process is a powerful tool for performing static analysis on Python programs. It can help you to identify and fix security vulnerabilities and code quality issues, improving the overall security and quality of your software.