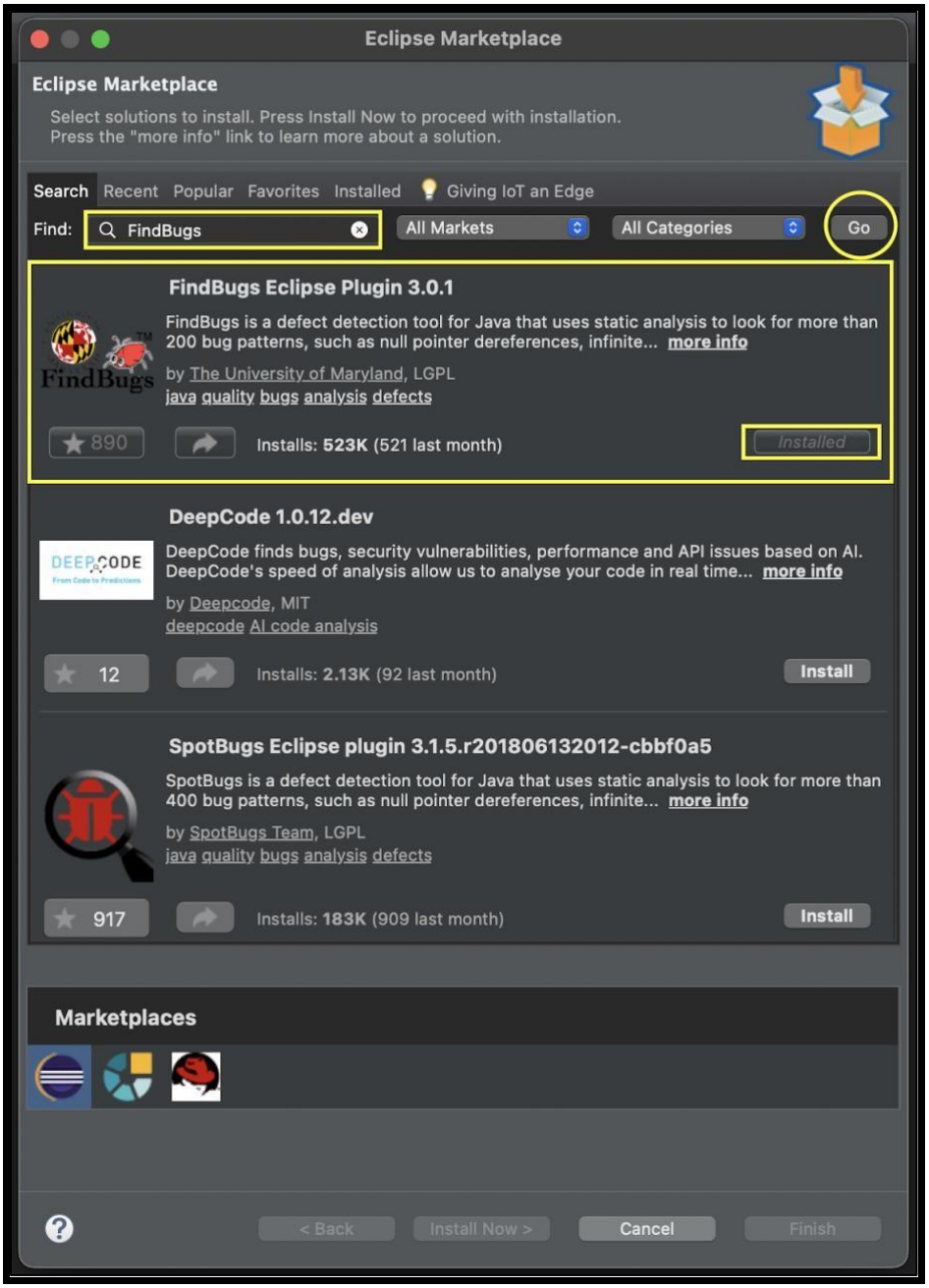
1. The configuration needed to run each tool

For both FindBugs and PMD, we need JAVA Enterprise version.

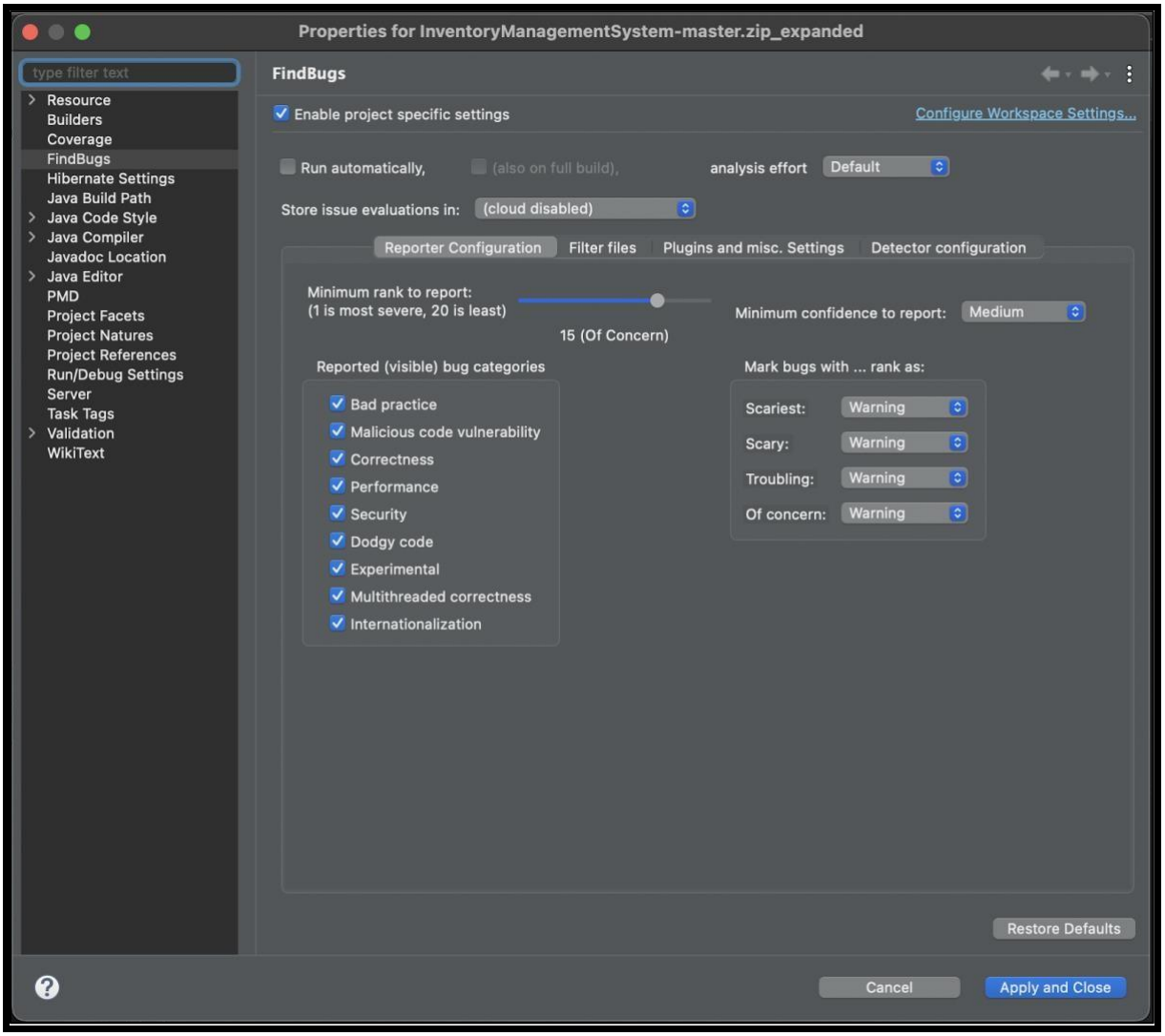
Steps to install FindBugs:

1. Select Help > Eclipse Marketplace.
2. Type FindBugs in Find (as per the picture below).
3. Select Go.
4. Select the First option of Findbugs Eclips Plugin 3.0.1.
5. Select install.



Steps to configure FindBug reported category:

Right-click on Project > Properties > Select FindBug



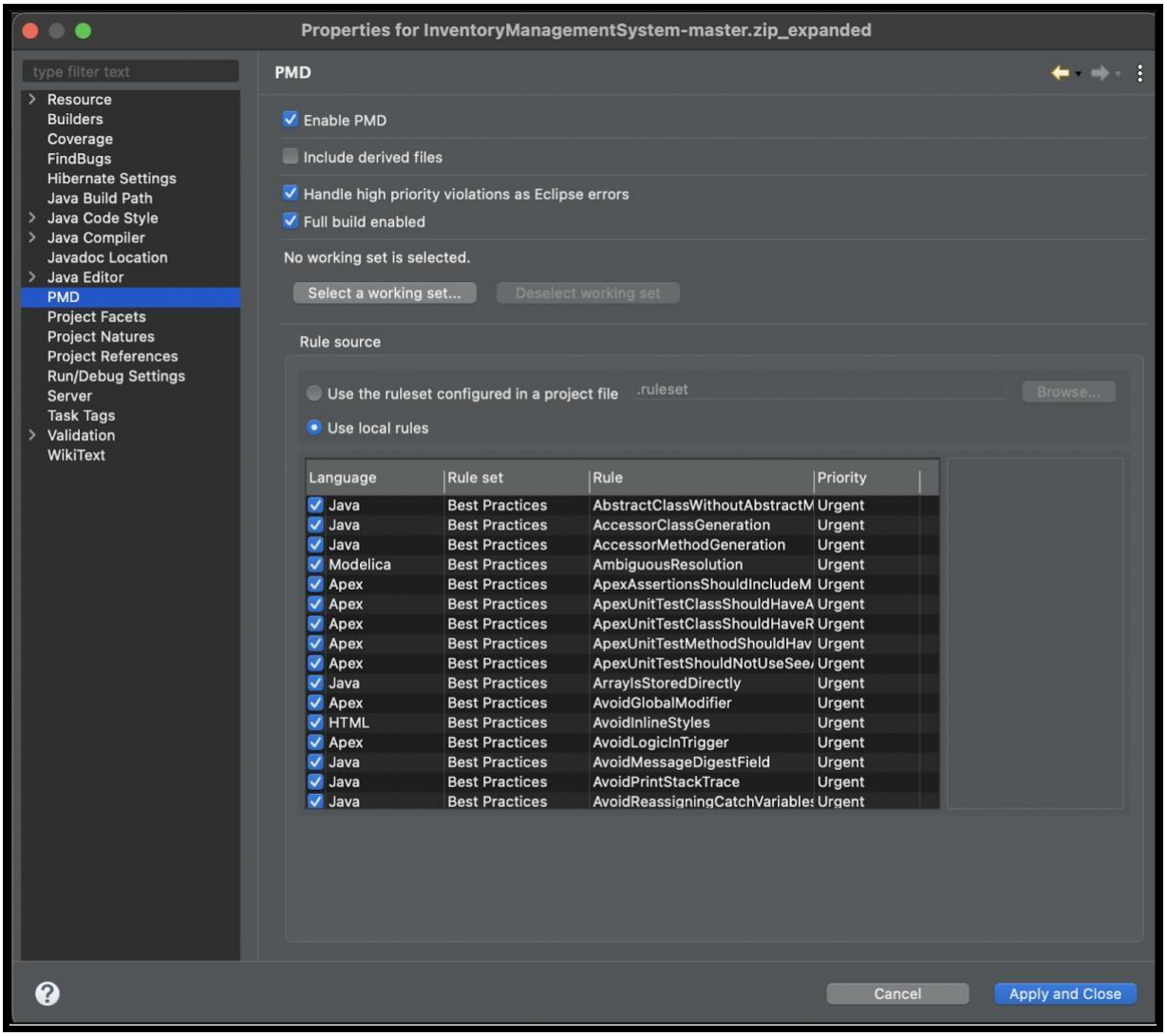
Steps to install PMD:

1. Select Help > Eclipse Marketplace.
2. Type PMD in Find (as per the picture below).
3. Select Go.
4. Select PMD plugin option as per the image.

5. Select install.

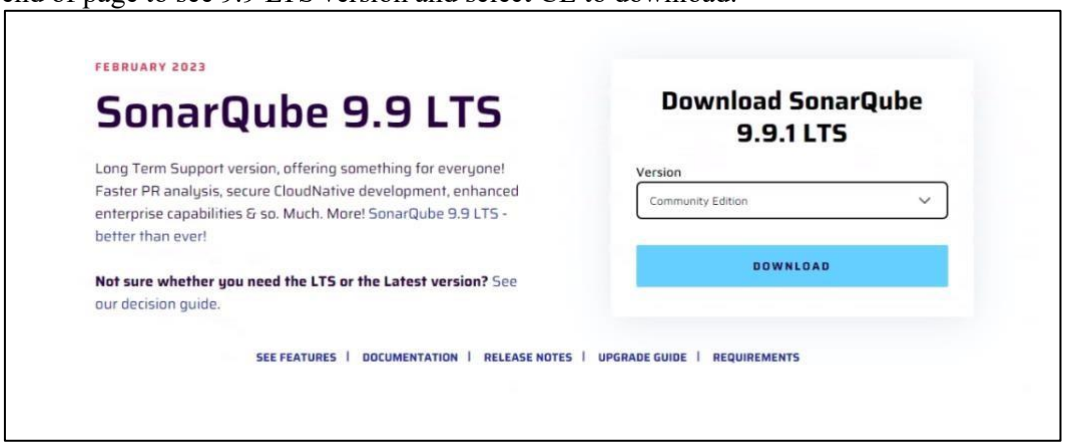
Steps to configure PMD reported category:

Right-click on Project > Properties > Select PMD

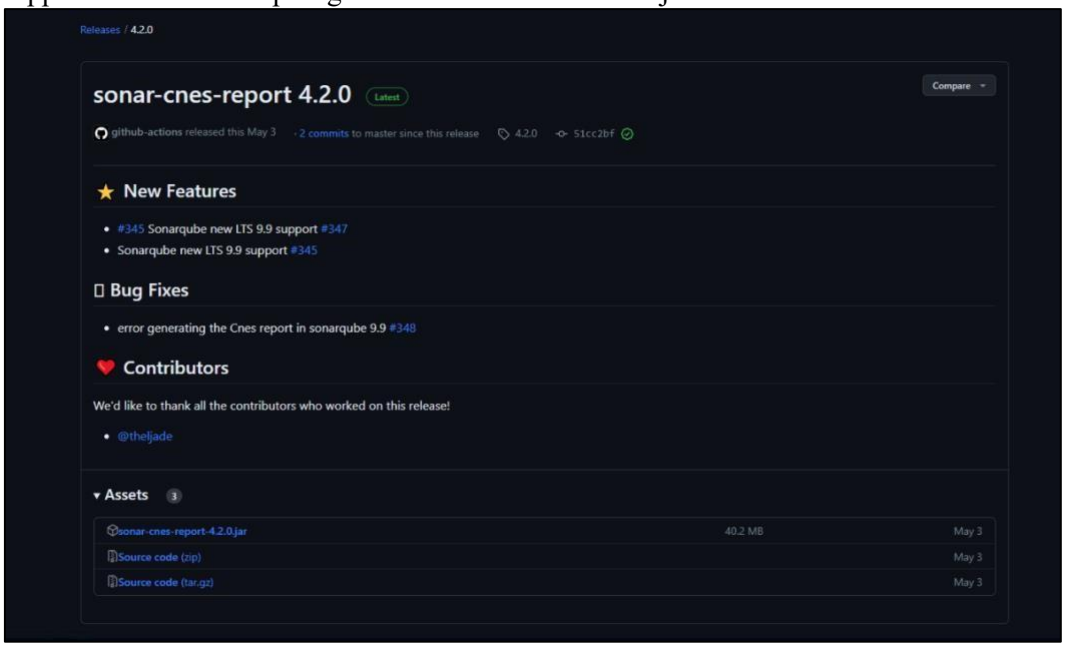


Steps to install SonarQube:

1. Go to https://www.sonarsource.com/products/sonarqube/downloads/ and scroll down to end of page to see 9.9 LTS version and select CE to download.

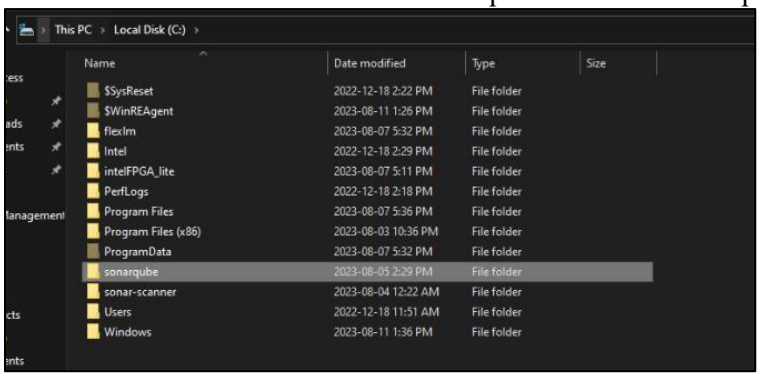


2. Go to https://github.com/cnescatlab/sonar-cnes-report/releases/tag/4.2.0 to find the latest supported version of report generator and download the jar file.

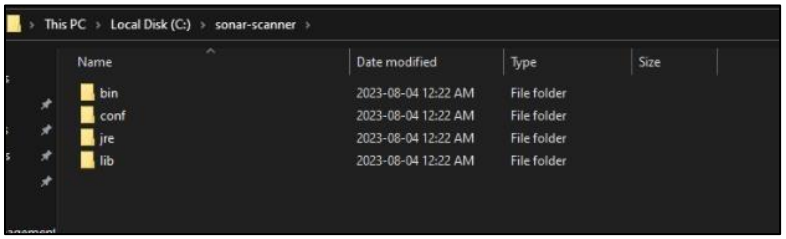


3. Go to https://docs.sonarsource.com/sonarqube/9.9/analyzing-sourcecode/scanners/sonarscanner/ and get the Windows 64-bit file.

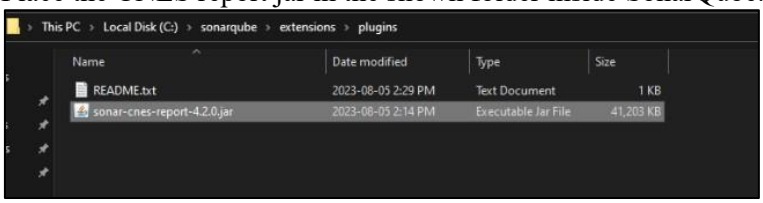
4. Make 2 new Folders in the Local Disk of your windows installation. SonarQube and SonarScanner. Extract the downloaded Zip files inside their respective folders.



Like so:

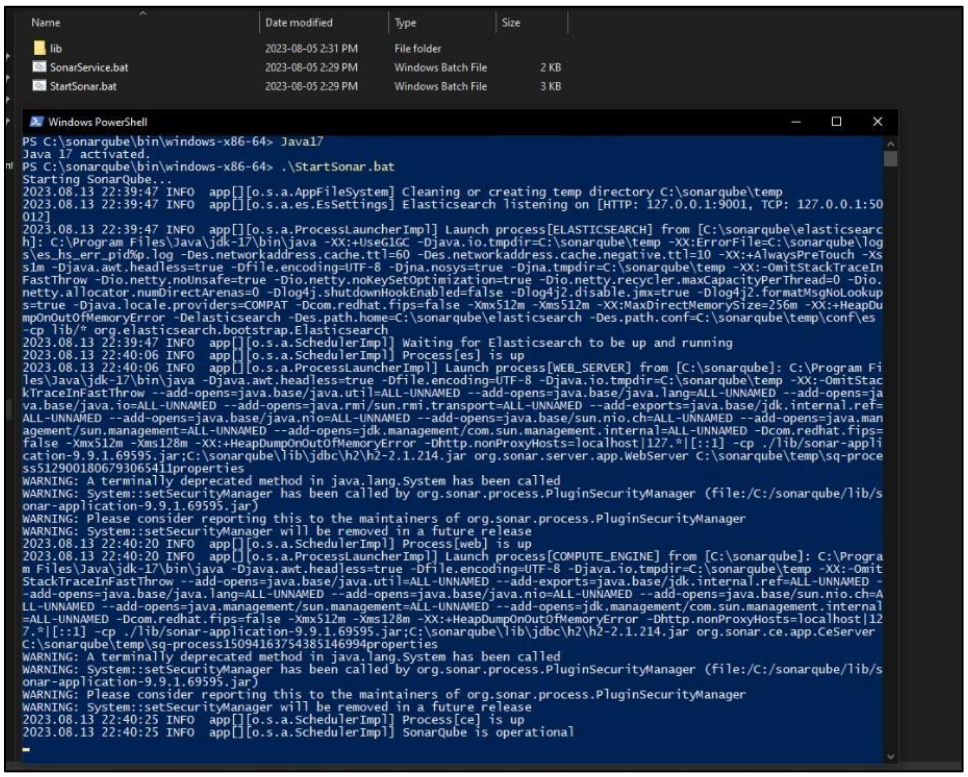


5. Place the CNES report jar in the shown folder inside SonarQube.



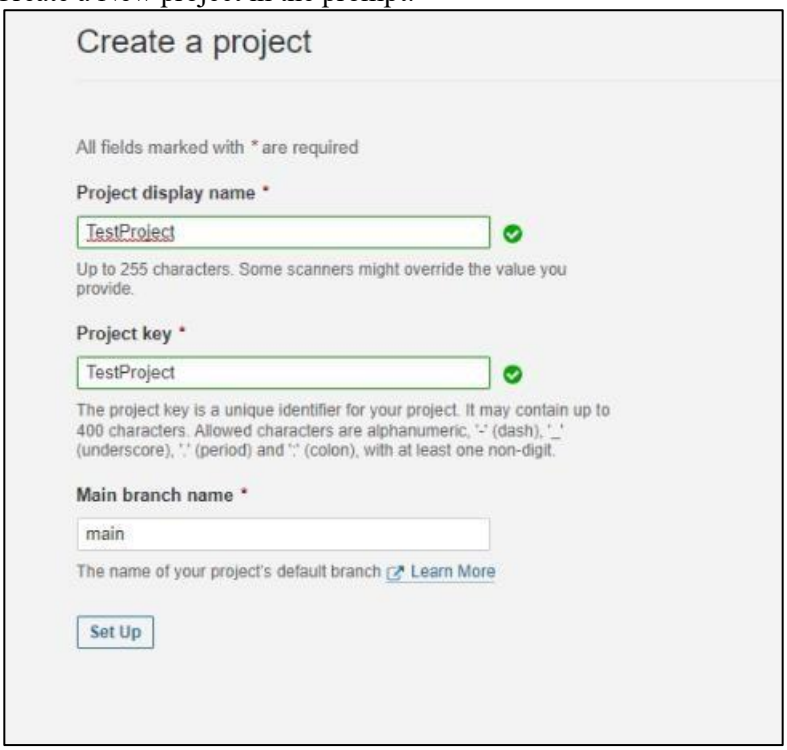
6. Make sure that you set sonar-Scanner in environment variables for the SYSTEM.By going in PATH as so.

7. Open the SonarQube/bin/windows-x86_64/ and open a PowerShell window in the same folder. **CHECK and MAKE SURE the JAVA version is 17.**And run the StartSonar.bat file

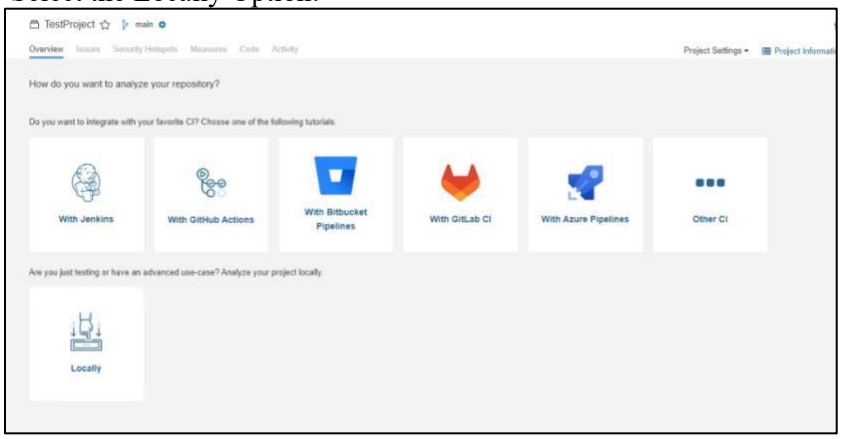


8. Once it is operational, Go to localhost:9000 to find your installation in a web browser. It may ask you to login with default credentials as admin, admin and reset it again. Change the password as you deem fit and login.

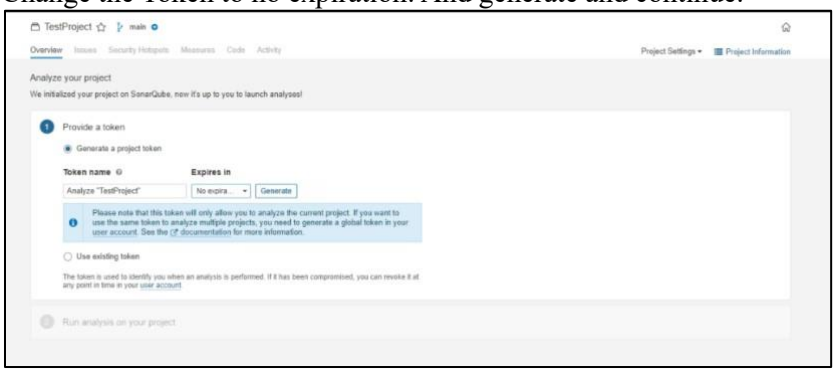
9. Create a New project in the prompt.



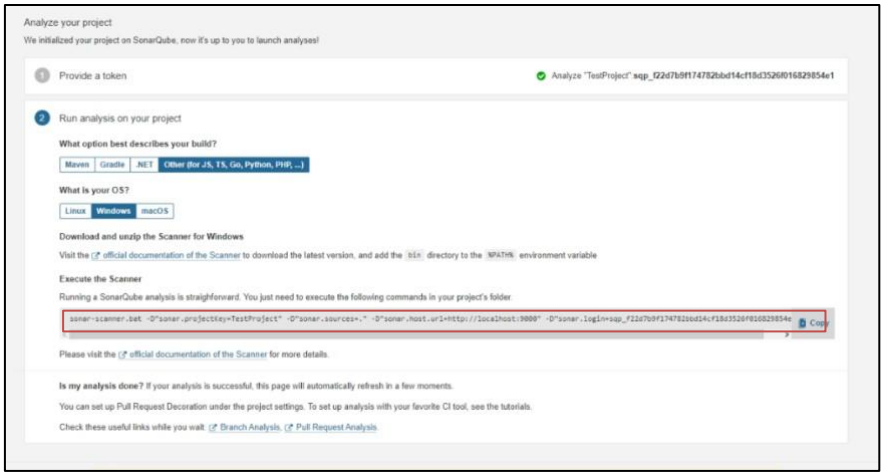
10. Select the Locally Option:



11. Change the Token to no expiration: And generate and continue:

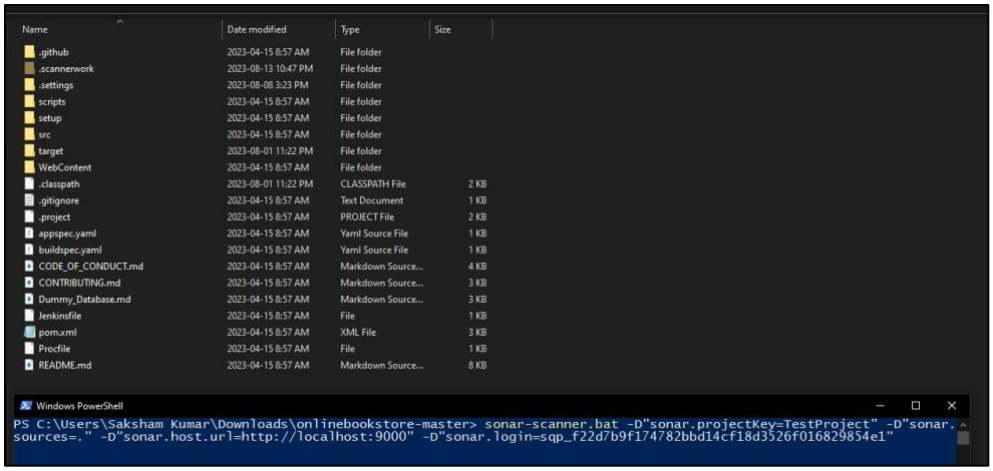


12. Select Other and Windows:



And copy the command provided:

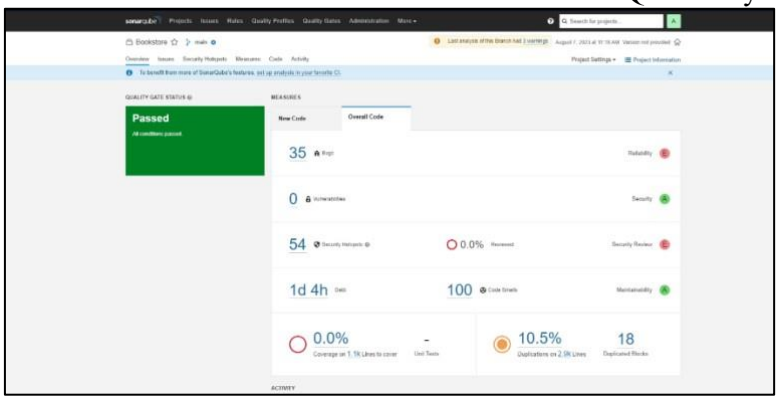
13. Run the Command in a new PowerShell window in the same folder where the project is installed in like so:



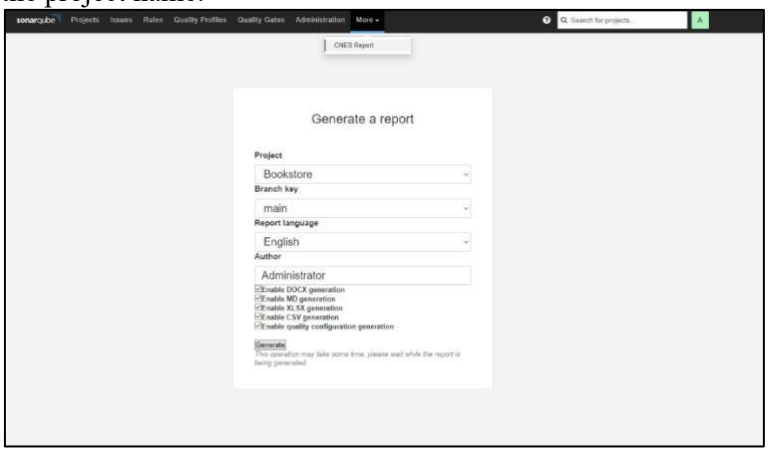
Incase of an error about the classes not being present and only java file, try appending the command with this command and run it:

-D"sonar.java.binaries=target/classes" OR where the ".class" files are stored

14. After a successful run check on the browser for SonarQube and you will see the web report:



15. For downloading the report Go on “More” and “CNES Report” and generate by selecting the project name:



NOTE: If you have any pop-ups for allowing the extension please click YES.

2. Reported bugs

1. FindBug reported a security vulnerability in Train Ticket Reservation System. Using this report we tried performing Cross-Site Scripting injection "<script>alert("ATTACK")</script>" on text input boxes. This injection did produce a security bug.

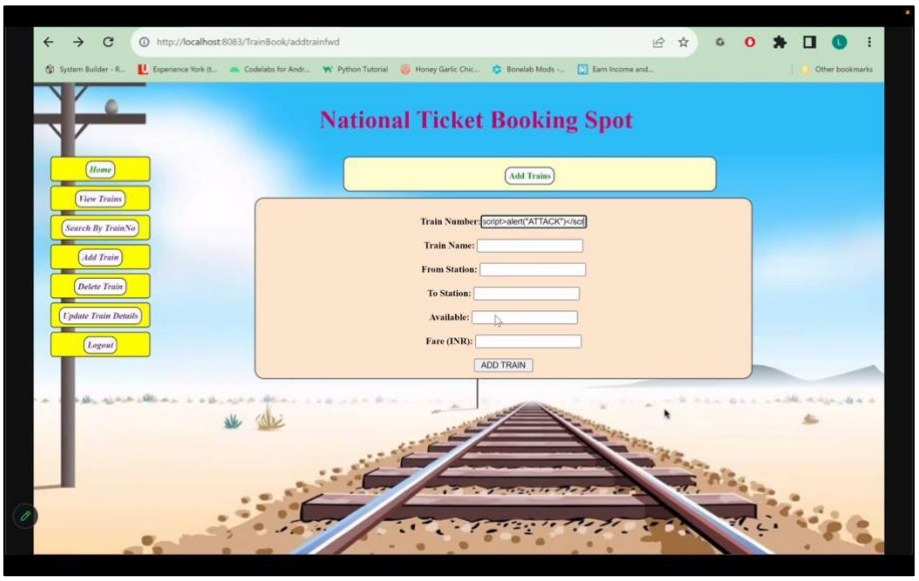
Bug Description:
The website is having a security vulnerability. This could potentially put user accounts at risk and allow malicious activities to take place. I have attached the screenshots of performing Cross-Site Scripting "<script>alert("ATTACK")</script>" on text input boxes.

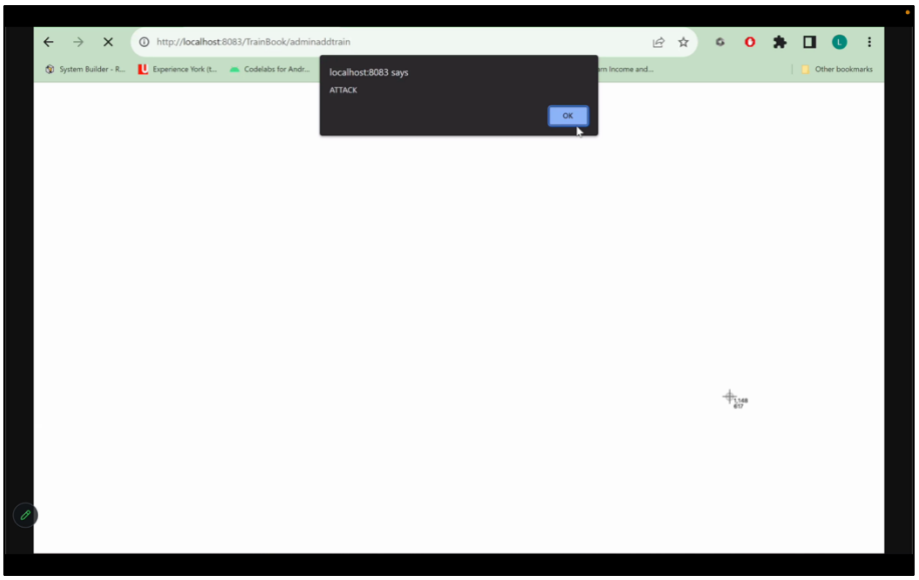
Steps to reproduce the behavior:

1. Go to National Ticket Booking Spot
2. Type <script>alert("ATTACK")</script> in Train Number text box
3. An alert box will pop up (bug).

Expected behavior:
The website should not allow any type Cross-Site Scripting injection. However, it allowed the Cross-Site Scripting injection and an alert box appeared (bug).

Screenshots:





Bug Reported Link:
https://github.com/shashirajraja/Train-Ticket-Reservation-System/issues/11

2. Online Shopping Cart (E-commerce website) and Train Ticket Reservation System projects are from the same repository. So, we decided to perform a Cross-Site Scripting injection on Online Shopping Cart, and we found a security vulnerability.

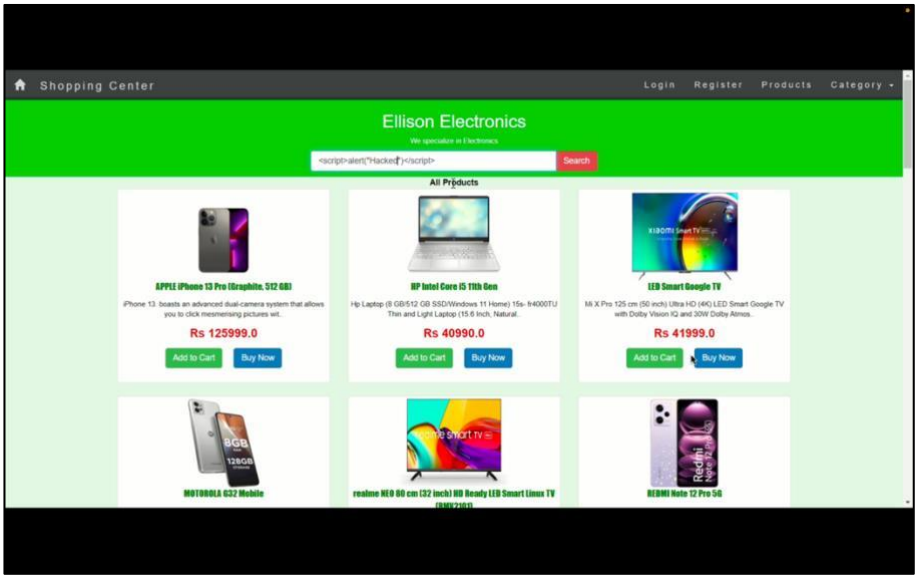
Bug Description:
The website is having a security vulnerability. This could potentially put user accounts at risk and allow malicious activities to take place. I have attached the screenshots of performing Cross-Site Scripting "<script>alert("ATTACK")</script>" on text input boxes.

Steps to reproduce the behavior:
1. Go to HomePage.
2. Type <script>alert("ATTACK")</script> in search items text box.

3. An alert box will pop up (bug).

Expected behavior:
The website should not allow any type Cross-Site Scripting injection. However, it allowed the Cross-Site Scripting injection and an alert box appeared (bug).

Screenshots:



Bug Reported Link:
https://github.com/shashirajraja/shopping-cart/issues/10

3. The inventory management system had a hard-coded SQL query to authenticate the login. This query authenticates only one type of user.

Describe the bug:

Normal users can't login. The reason for the bug is that in ConnectionFactory class the checkLogin method is having hard-coded SQL query for ADMINISTRATOR instead of ADMINISTRATOR or Normal User. Therefore, if a Normal User tries to login into the application. This query returns a NULL result.

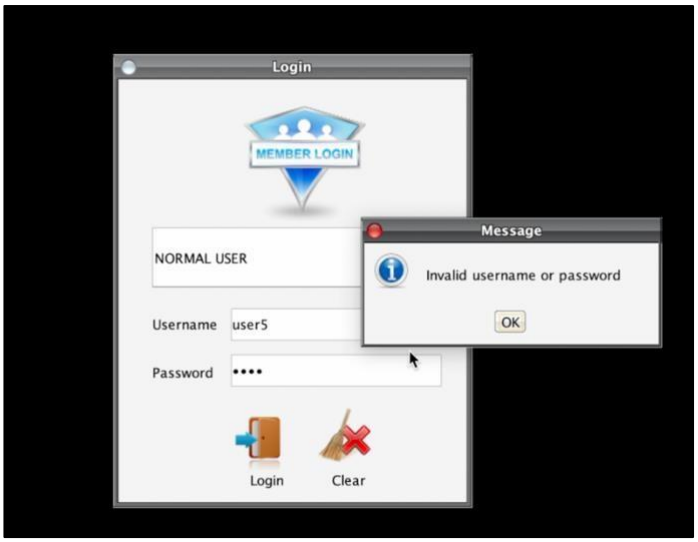
Steps to reproduce the behavior:

1. Enter Username: (Normal User's username) 2.
Enter password: (Normal User's password)
3. User gets login error.

Expected behavior:

Users should allow logging into the system if their Username and password associated with the Username is correct.

Screenshots:



Bug Reported Link:

https://github.com/sazanrjb/InventoryManagementSystem/issues/17