Experiment no.

Aim: To describe SonarQube basic concepts/terminologies and install it windows or Linux machine

LOs: LO1, LO4

Theory:

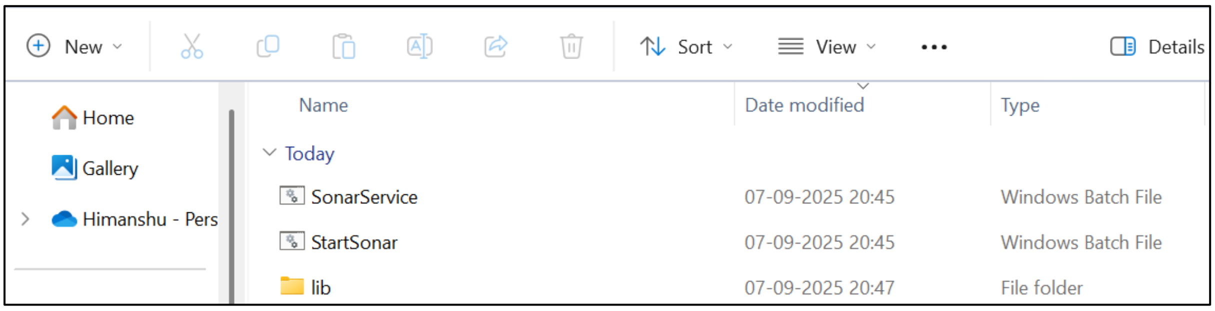
SonarQube: Basic Concepts & Terminology
At its core, SonarQube is a central server that processes code analysis reports. A scanner tool analyzes your code, and the results are sent to the server for visualization and tracking.

Key Terms
- **Project**: The codebase you are analyzing. A project in SonarQube is a collection of source files, configuration, and analysis history.
- **Issues**: An issue is a problem in your code that SonarQube has detected. Issues are categorized into three types:
  - **Bug**: A mistake in the code that will likely cause an error or unexpected behavior at runtime.
  - **Vulnerability**: A piece of code that is open to attack (e.g., SQL injection, cross-site scripting).
  - **Code Smell**: A confusing or poorly structured part of the code that makes it hard to maintain. It's not a bug, but it indicates a weakness in design that could lead to problems later.
- **Technical Debt**: This is a metaphor representing the implied cost of rework caused by choosing an easy (limited) solution now instead of using a better approach that would take longer. Code smells are a primary source of technical debt. SonarQube quantifies this, often in terms of time (e.g., "3 days of effort to fix all code smells").
- **Quality Gate**: A set of conditions your project must meet to be considered ready for release. It's the core of SonarQube's philosophy. A Quality Gate can fail if, for example, the code has new critical issues, insufficient test coverage, or too much duplication. This provides a clear "go/no-go" signal.
- **Quality Profile**: A collection of **Rules** used to analyze a specific language. For example, you have a Quality Profile for Java and another for Python. You can customize these profiles by activating or deactivating rules to fit your team's coding standards.
- **Rules**: A single coding standard or practice that SonarQube checks for. There are thousands of rules for various languages, covering everything from simple formatting to complex security vulnerabilities.
- **SonarScanner**: The tool that actually analyzes your source code. The scanner runs in your build environment (e.g., on a developer's machine, in a Jenkins pipeline, or with GitHub Actions). After analysis, it sends a detailed report to the SonarQube server.

Part 1: Install and Run the SonarQube Server
This part sets up the local server that will store and display the analysis results.

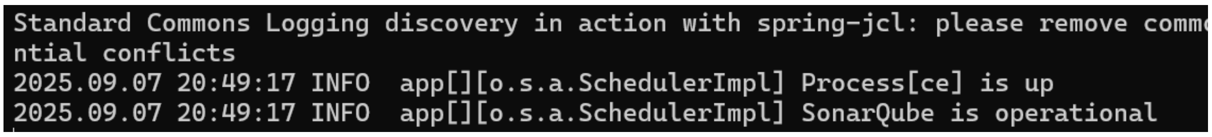
1. **Download SonarQube:**
   - Go to the official website: https://www.sonarsource.com/products/sonarqube/downloads/

- Download the **Community Edition**.



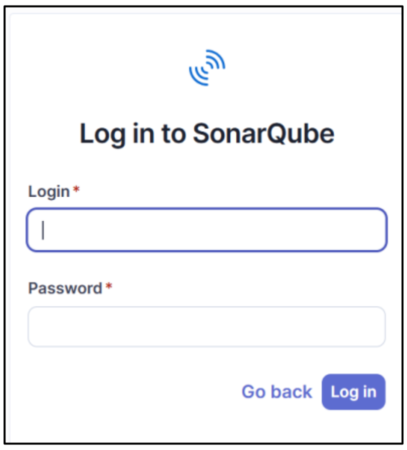
Name	Date modified	Type
Today		
SonarService	07-09-2025 20:45	Windows Batch File
StartSonar	07-09-2025 20:45	Windows Batch File
lib	07-09-2025 20:47	File folder

2. **Create Folder and Extract:**
   - Create a new folder on your C: drive named SONARQUBE.
   - Extract the contents of the downloaded ZIP file into C:\SONARQUBE. You will now have a folder structure like C:\SONARQUBE\sonarqube-10.1.0.73491.
3. **Run the Server:**
   - Navigate to the bin directory for your Windows system. The path will be: C:\SONARQUBE\sonarqube-10.1.0.73491\bin\windows-x86-64
   - In this folder, double-click the file named StartSonar.bat.
   - A command prompt window will open. Wait for the server to start. This may take a few minutes. You will know it's ready when you see a line that says **"SonarQube is operational"**.
   - **Important:** Keep this command prompt window open. Closing it will shut down the SonarQube server.



```
Standard Commons Logging discovery in action with spring-jcl: please remove comm
ntial conflicts
2025.09.07 20:49:17 INFO  app[][o.s.a.SchedulerImpl] Process[ce] is up
2025.09.07 20:49:17 INFO  app[][o.s.a.SchedulerImpl] SonarQube is operational
```

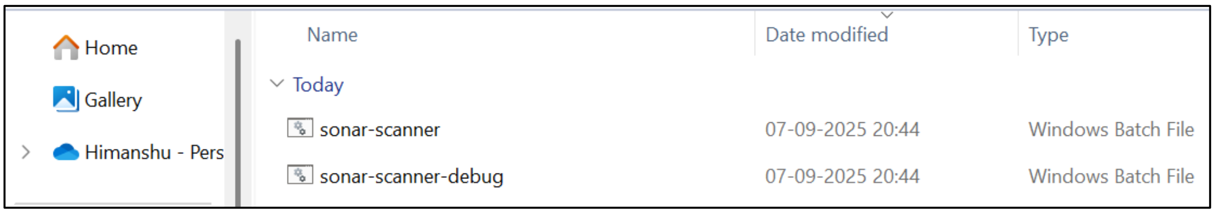
4. **Access SonarQube Dashboard:**
   - Open your web browser and go to: http://localhost:9000
   - You will be prompted to log in. Use the default credentials:
     - **Login:** admin
     - **Password:** admin
   - SonarQube will force you to change the password. As per your document, change it to admin123.



Part 2: Set up the SonarScanner

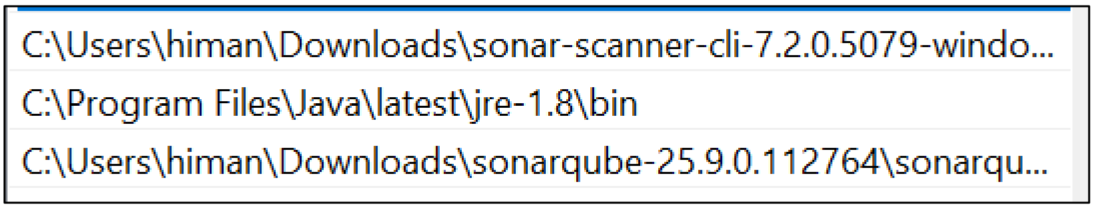
This is the command-line tool that actually scans your code.

1. **Download SonarScanner:**
   - From the SonarQube dashboard or the official website, download the SonarScanner for Windows.
2. **Create Folder and Extract:**
   - Create a new folder on your C: drive named SONARSCANNER.
   - Extract the contents of the downloaded scanner ZIP file into C:\SONARSCANNER. You will now have a folder structure like C:\SONARSCANNER\sonar-scanner-5.0.1.3006-windows.



Name	Date modified	Type
Today		
sonar-scanner	07-09-2025 20:44	Windows Batch File
sonar-scanner-debug	07-09-2025 20:44	Windows Batch File

3. **Add Scanner to Environment Path (Recommended):**
   - This step allows you to run the scanner command from any directory.
   - Copy the path to the scanner's bin directory: C:\SONARSCANNER\sonar-scanner-5.0.1.3006-windows\bin
   - Search for "Edit the system environment variables" in the Windows Start Menu.
   - Click on "Environment Variables...".
   - Under "System variables", find the Path variable, select it, and click "Edit...".
   - Click "New" and paste the path you copied.
   - Click OK on all windows to save the changes.

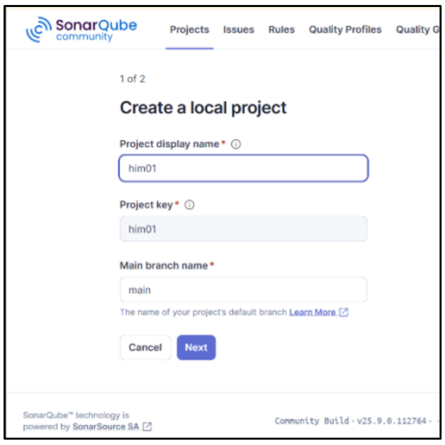


```
C:\Users\himan\Downloads\sonar-scanner-cli-7.2.0.5079-windo...
C:\Program Files\Java\latest\jre-1.8\bin
C:\Users\himan\Downloads\sonarqube-25.9.0.112764\sonarqu...
```

Part 3: Create a Project and Generate an Analysis Token

Now, let's create a project within SonarQube to hold the analysis results.

1. **Create a New Project:**
   - On the SonarQube dashboard (http://localhost:9000), click the "Create a project" button.
   - Select the option to create a project **"Manually"**.
   - Enter a **Project display name** (e.g., him01) and a **Project key** (e.g., him01).
   - Click **"Next"** or **"Set Up"**.



1 of 2

Create a local project

Project display name *

him01

Project key *

him01

Main branch name *

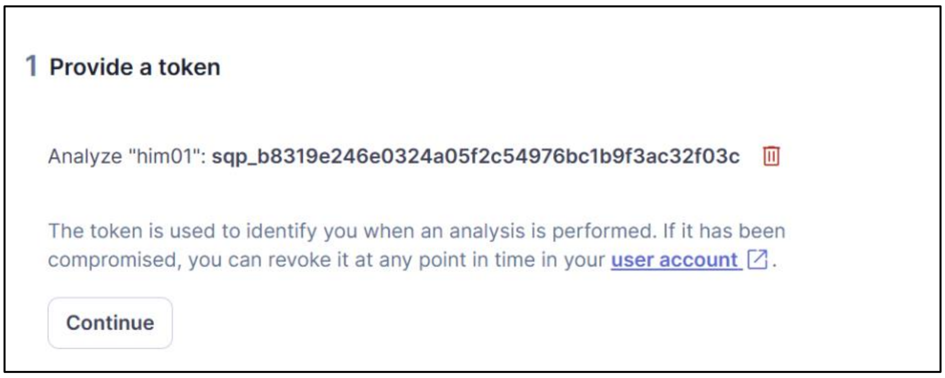
main

The name of your project's default branch Learn More

SonarQube™ technology is powered by SonarSource SA

Community Build - v25.9.0.112764 -

2. **Generate an Analysis Token:**
   - On the next screen, under "Analyze your project," choose the **"Locally"** option.
   - Click the **"Generate a token"** button.
   - You will be shown a new token. It
   - **Crucial:** Copy this token and save it somewhere safe, like a Notepad file. You will need it to run the scan. You will not be able to see it again after you leave this page.
   - Click **"Continue"**.

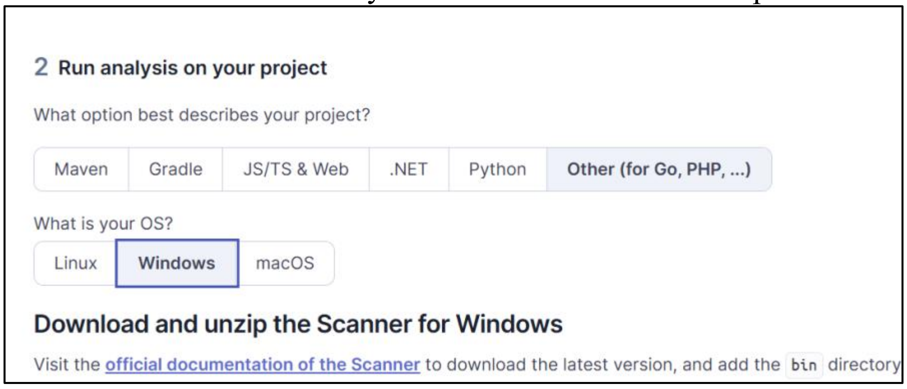


1 Provide a token

Analyze "him01": sqp_b8319e246e0324a05f2c54976bc1b9f3ac32f03c

The token is used to identify you when an analysis is performed. If it has been compromised, you can revoke it at any point in time in your user account.

3. **Get the Scanner Command:**
   - Select your operating system (**Windows**) and you will be shown an example command to run the analysis. We will use this as a template in Part 5.



2 Run analysis on your project

What option best describes your project?

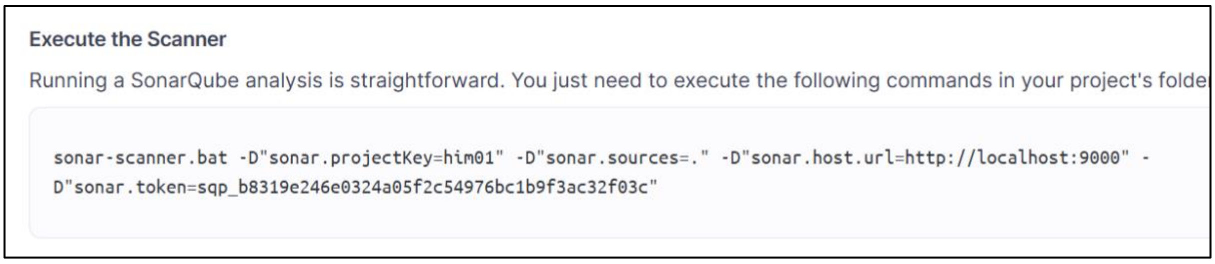
Maven | Gradle | JS/TS & Web | .NET | Python | [✓ Other (for Go, PHP, ...)]

What is your OS?

Linux | [✓ Windows] | macOS

Download and unzip the Scanner for Windows

Visit the official documentation of the Scanner to download the latest version, and add the `bin` directory



Execute the Scanner

Running a SonarQube analysis is straightforward. You just need to execute the following commands in your project's folde

```
sonar-scanner.bat -D"sonar.projectKey=him01" -D"sonar.sources=." -D"sonar.host.url=http://localhost:9000" -
D"sonar.token=sqp_b8319e246e0324a05f2c54976bc1b9f3ac32f03c"
```

Part 4: Prepare Your Code Project

You need a folder with some source code to analyze.

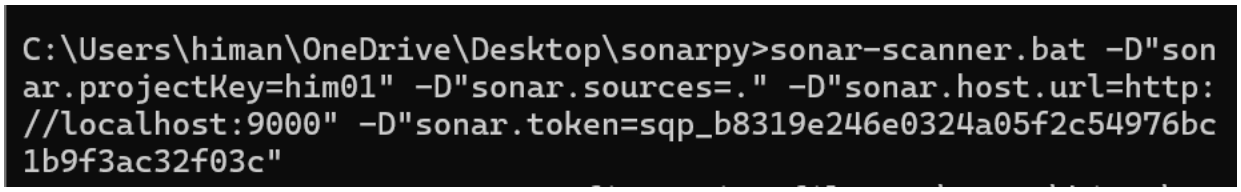
1. **Create a Project Folder:** Create a folder anywhere on your computer for your source code. For this example, let's create it at C:\my-python-project.
2. **Add Source Files:** Place the code you want to analyze into this folder. As your document suggests, you can download a sample Python project or create your own .py files and put them in C:\my-python-project.

sonarpy ✓ 07-09-2025 21:51

Part 5: Run the Analysis

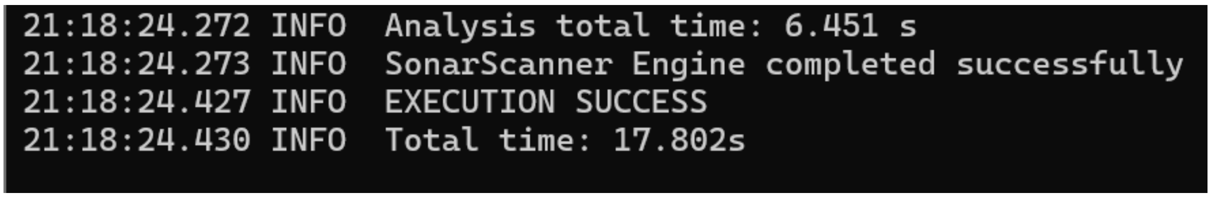
This is the final step where you execute the scan.

1. **Open Command Prompt in Your Project Folder:**
   - Navigate to your project folder (C:\my-python-project) in File Explorer.
   - In the address bar at the top, type cmd and press **Enter**. This will open a command prompt window directly in that folder.
2. **Execute the Scanner Command:**
   - In the command prompt window, paste and execute the following command. Make sure to replace the projectKey and token values with your own.



```
C:\Users\himan\OneDrive\Desktop\sonarpy>sonar-scanner.bat -D"son
ar.projectKey=him01" -D"sonar.sources=." -D"sonar.host.url=http:
//localhost:9000" -D"sonar.token=sqp_b8319e246e0324a05f2c54976bc
1b9f3ac32f03c"
```

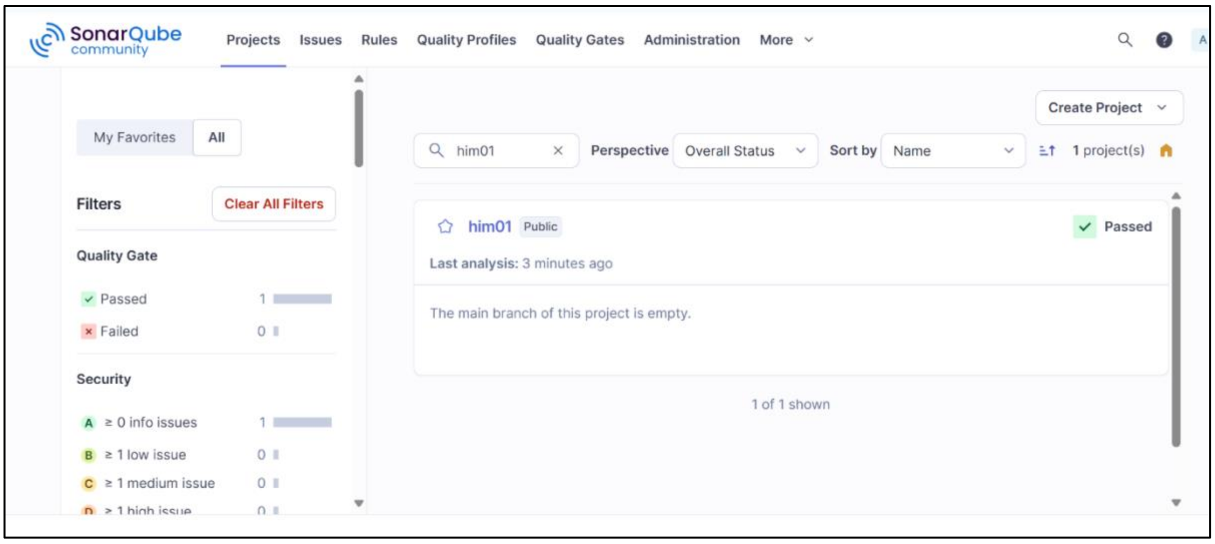
3. **Wait for Completion:** The scanner will now analyze your code. You will see a lot of output in the command prompt. Wait until you see a message like **"EXECUTION SUCCESS"**.



```
21:18:24.272 INFO  Analysis total time: 6.451 s
21:18:24.273 INFO  SonarScanner Engine completed successfully
21:18:24.427 INFO  EXECUTION SUCCESS
21:18:24.430 INFO  Total time: 17.802s
```

Part 6: View the Analysis Report

1. Go back to your SonarQube dashboard in your web browser (http://localhost:9000).
2. Click on your project ("Jyotsna").
3. You will now see a detailed report of your code's quality, including bugs, vulnerabilities, code smells, cyclomatic complexity, and other metrics.



Conclusion: In conclusion, SonarQube is a vital tool for development teams aiming to enhance code quality and security. It operates by analyzing code for bugs, vulnerabilities, and code smells, helping to manage technical debt effectively. The installation on both Windows and Linux is straightforward, providing a central server for analysis results. However, its true power is realized when the SonarScanner is integrated into a CI/CD pipeline, enforcing quality standards through automated Quality Gates. Adopting SonarQube ultimately fosters a culture of continuous improvement and cleaner, more secure code.

LOs achieved: LO1 and LO4

POs achieved: PO1-PO5, PO8, PO10, PO12.