



An-Najah National University
Faculty of Engineering
Computer Engineering Department

First Semester 2023/2024.

Students name:

Wala' Essam Ashqar.

ID: 12027854.

Doaa Jararaa

ID: 12029152.

Supervisor:

Dr. Haya Samaana.

- **Introduction:**

This project is Car accessories company, we made 7 features.

- **Wala' Essam Ashqar achievements:**

1. Everything has to do with the admin, Customer, Product catalog and sign in, (5 Features, Testing, and source code).
2. CI/CD/Sonar/Coverage/Refactor which are included in this report.

- **Doaa Jararaa achievements:**

Everything has to do with installer, Notification (2 Features, Testing, and the source code).

- **Discussion:**

- **Admin menu:**

1. Show product categories.
2. Add product listings.
3. update product listings.
4. Delete product listings.
5. Add a new category.
6. Search for a product.
7. View customer accounts.
8. Manage customer accounts.
9. Delete a customer account.
10. Schedule and manage installation appointments.
0. Exit

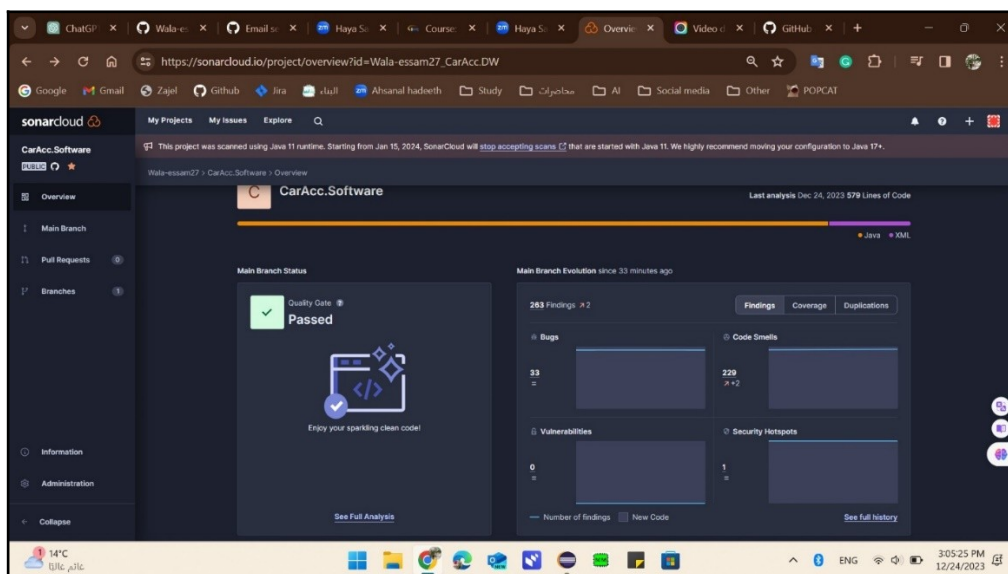
➤ Customer Menu:

1. View my profile.
2. Edit my profile or delete it.
3. View products.
4. Search for products.
5. Purchase products.
6. View order history and installation requests.
7. Delete an order.
0. Exit.

I have created an account on the GitHub (is a platform and cloud-based service for software development and version control using Git, allowing developers to store and manage their code), and I add repository to it to push my project, in order to ensure the quality of the project, and I have connected it with the sonar cloud to check the bad smells, so that I can fix it, and it checks maintainability, reliability, and security for it therefore improving the overall quality for project.

➤ In the first analysis:

Master Branch Evolution:



***There is a problem in coverage, so I want to get the value from eclipse.

Summary - CarAcc.Softw... Projects - Wala-essam27... ChatGPT... Messenger | Facebook... CarAcc.DW/src/main/jav... +

https://sonarcloud.io/organizations/wala-essam27/projects

Google Gmail Zajel Github Jira البقاء Ahsanal hadeeth Study محاضرات AI Social media Other POPCAT

sonarcloud My Projects My Issues Explore

Wala-essam27 PUBLIC Key: wala-essam27

Projects Quality Profiles Rules Quality Gates Members Billing & Upgrade Administration

Filters

Quality Gate

Passed 1

Failed 0

Reliability

A 0

B 0

C 1

D 0

E 0

Security

A 1

Search for projects Perspective Overall Status Sort by Name 1 projects

CarAcc.Softw NEW PUBLIC Passed

Last analysis: 12/25/2023, 3:47 PM - 575 Lines of Code - Java, XML

33 Bugs 0 Vulnerabilities 100% Hotspots Reviewed 228 Code Smells 0.0% Coverage 0.0% Duplications

1 of 1 shown

© 2018-2023 SonarSource SA. All rights reserved. Terms Pricing Privacy Cookie Policy Security Community Documentation Contact us Status About

12°C سماء صافية

Wala-essam27 Course: البرمجيات Haya Samaana's P... Summary - CarAcc... Create main2.yml Summary - CarAcc... +

https://sonarcloud.io/summary/overall?id=Wala-essam27_CarAcc.DW

Google Gmail Zajel Github Jira البقاء Ahsanal hadeeth Study محاضرات AI Social media Other POPCAT

sonarcloud My Projects My Issues Explore

CarAcc.Softw PUBLIC

Overview

Main Branch

Pull Requests

Branches

Information

Administration

Collapse

This project was scanned using Java 11 runtime. Starting from Jan 15, 2024, SonarCloud will stop accepting scans that are started with Java 11. We highly recommend moving your configuration to Java 17+.

Wala-essam27 > CarAcc.Softw > master

Summary Issues Security Hotspots Measures Code Activity

579 Lines of Code Version 0.01-SNAPSHOT Last analysis 3 hours ago 7e47d432 Create main2.yml

Quality Gate Passed

New Code Overall Code

Reliability 33 Bugs

Maintainability 229 Code Smells

Security 0 Vulnerabilities

Security Review 1 Security Hotspots 0.0% Reviewed

Coverage 0.0% Coverage

Duplications 0.0% Duplications

© 2018-2023 SonarSource SA. All rights reserved. Terms Pricing Privacy Cookie Policy Security Community Documentation Contact us Status About

14°C غائم عاصف

➤ After refactoring:

The SonarCloud dashboard for the CarAcc.DW project shows a failed quality gate. The project was scanned using Java 11 runtime, starting from Jan 15, 2024. A warning message states: "This project was scanned using Java 11 runtime. Starting from Jan 15, 2024, SonarCloud will stop accepting scans that are started with Java 11. We highly recommend moving your configuration to Java 17+." The last analysis was performed 1 minute ago by user da814ae9. The dashboard displays 588 Lines of Code, 2 Failing Conditions, and a New Code button. The Quality Gate is Failed. The metrics are as follows:

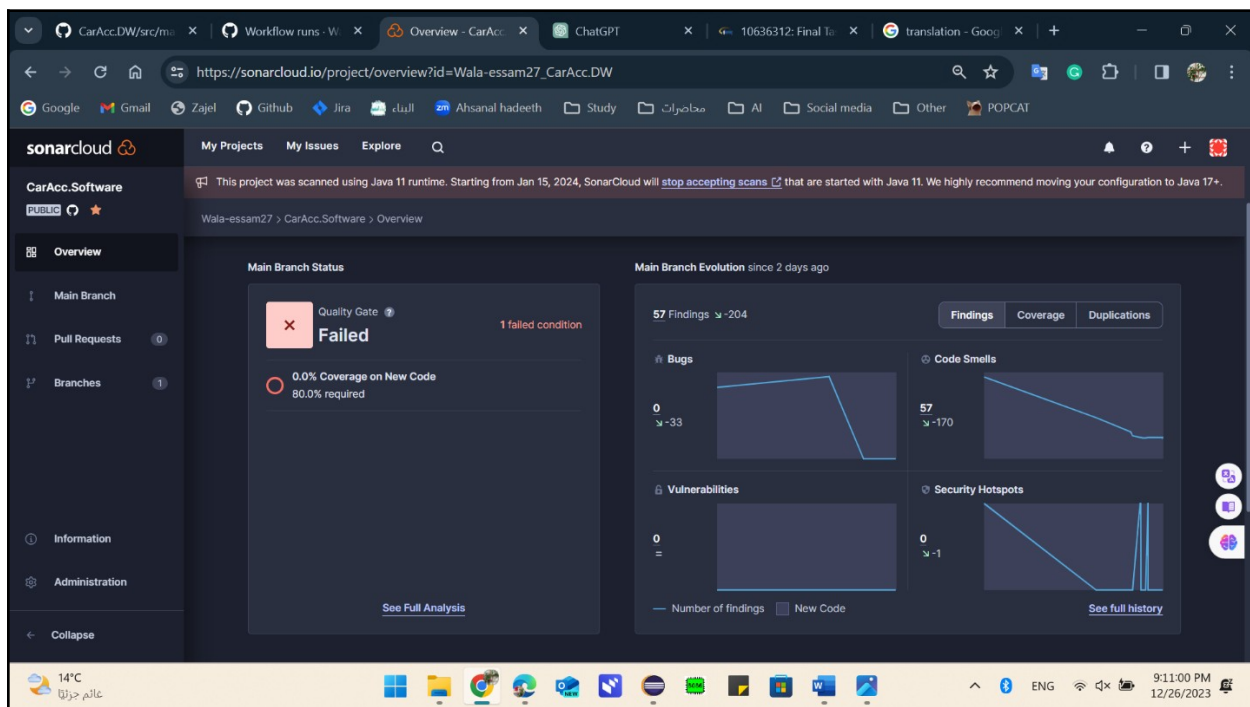
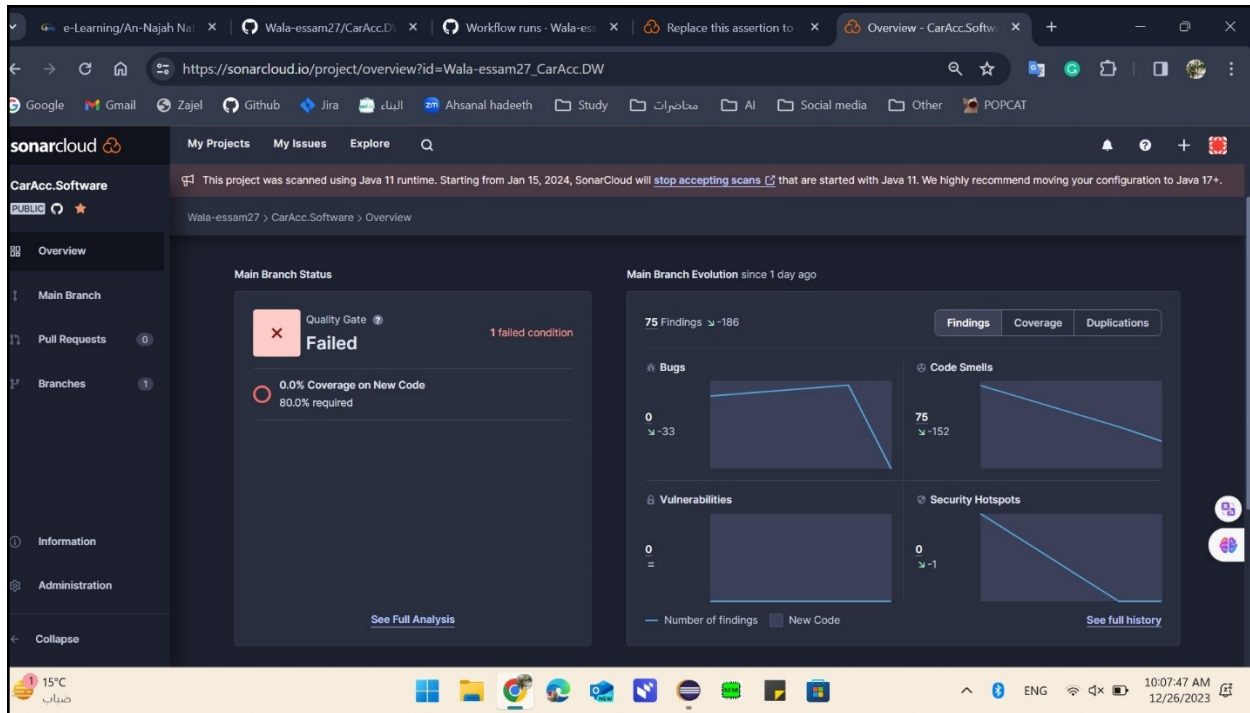
Metric	Value	Status
Reliability	38 Bugs	Failed (C)
Maintainability	114 Code Smells	Failed (A)
Security	0 Vulnerabilities	Passed (A)
Security Review	0 Security Hotspots	Passed (A)
Coverage	0.0% Coverage	Failed (C)
Duplications	0.0% Duplications	Passed (A)

The SonarCloud dashboard for the CarAcc.DW project shows a failed quality gate. The project was scanned using Java 11 runtime, starting from Jan 15, 2024. A warning message states: "This project was scanned using Java 11 runtime. Starting from Jan 15, 2024, SonarCloud will stop accepting scans that are started with Java 11. We highly recommend moving your configuration to Java 17+." The last analysis was performed 3 hours ago by user 2585ced9 commit142. The dashboard displays 612 Lines of Code, 1 Failing Condition, and a New Code button. The Quality Gate is Failed. The metrics are as follows:

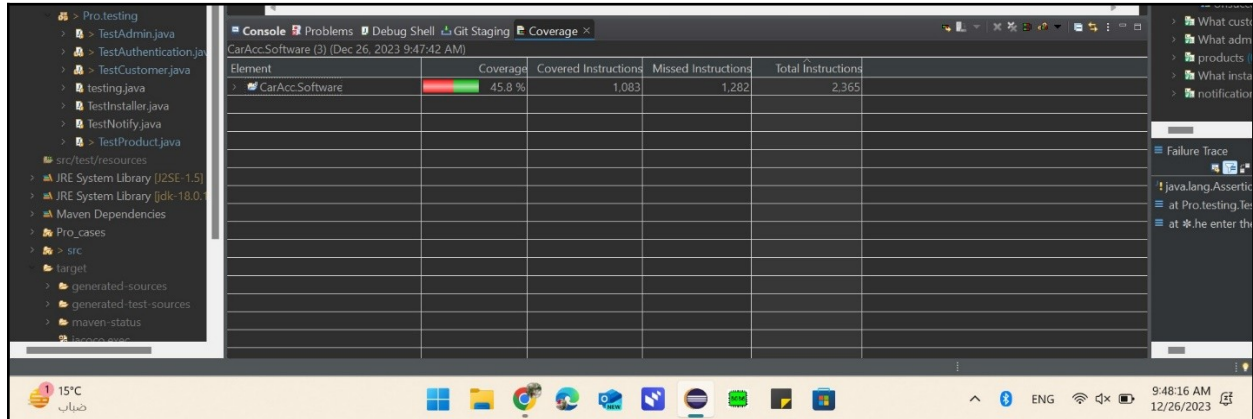
Metric	Value	Status
Reliability	0 Bugs	Passed (A)
Maintainability	47 Code Smells	Failed (A)
Security	0 Vulnerabilities	Passed (A)
Security Review	0 Security Hotspots	Passed (A)
Coverage	0.0% Coverage	Failed (C)
Duplications	0.0% Duplications	Passed (A)

The SonarCloud dashboard for the CarAcc.DW project shows a failed quality gate. The project was scanned using Java 11 runtime, starting from Jan 15, 2024. A warning message states: "This project was scanned using Java 11 runtime. Starting from Jan 15, 2024, SonarCloud will stop accepting scans that are started with Java 11. We highly recommend moving your configuration to Java 17+." The last analysis was performed 10:29:26 PM on 12/26/2023. The dashboard displays 10636312 Lines of Code, 1 Failing Condition, and a New Code button. The Quality Gate is Failed. The metrics are as follows:

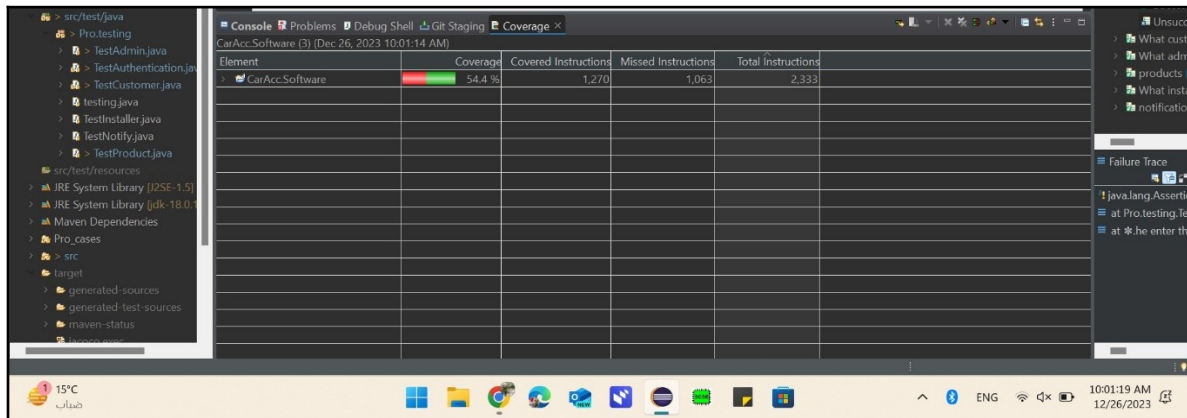
Metric	Value	Status
Reliability	0 Bugs	Passed (A)
Maintainability	0 Code Smells	Passed (A)
Security	0 Vulnerabilities	Passed (A)
Security Review	0 Security Hotspots	Passed (A)
Coverage	0.0% Coverage	Failed (C)
Duplications	0.0% Duplications	Passed (A)



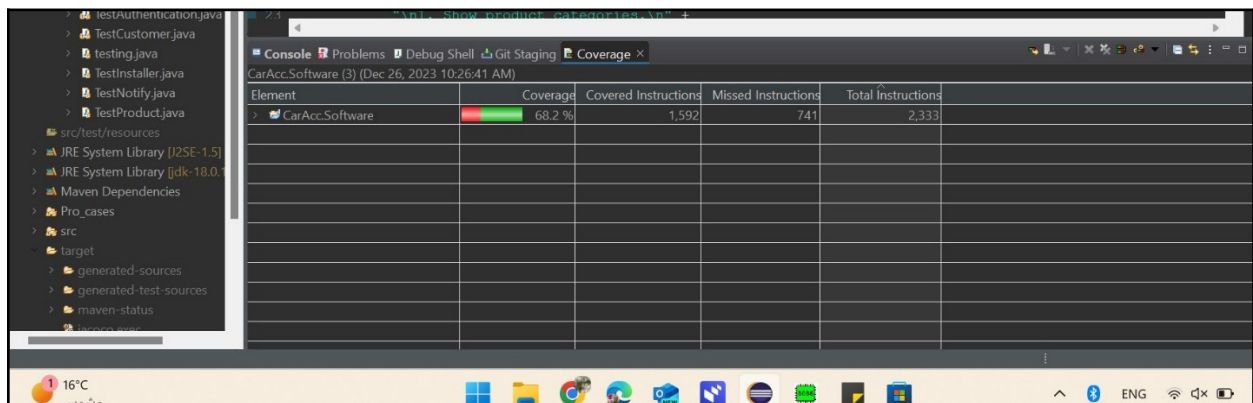
➤ The first value of coverage:

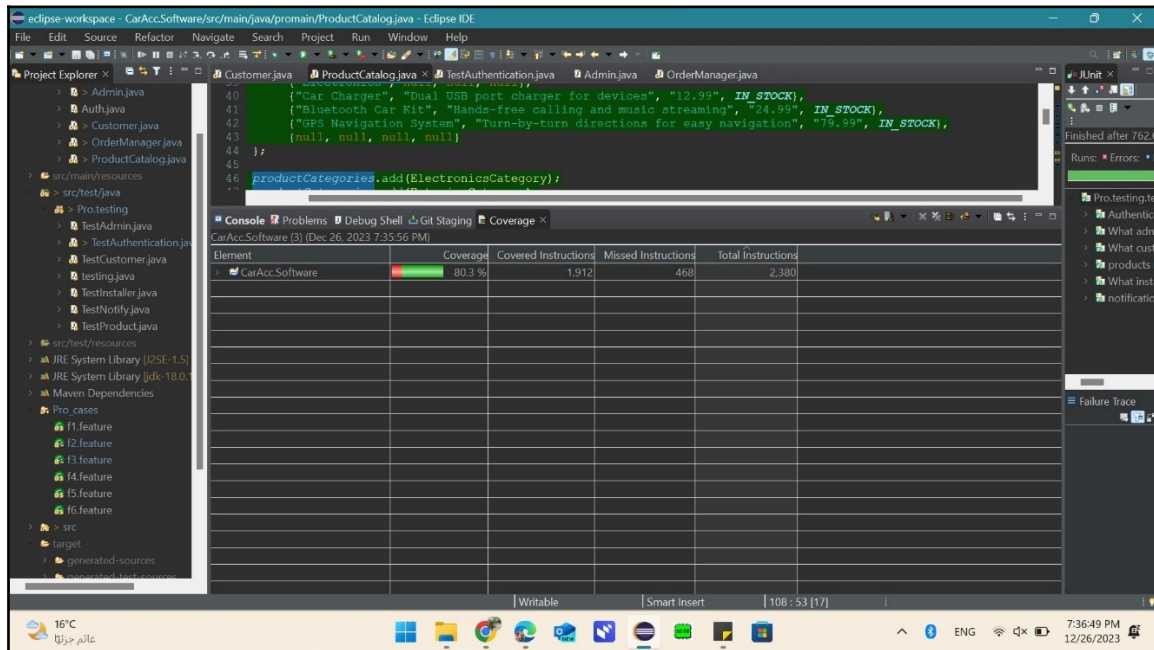


➤ After some changes:



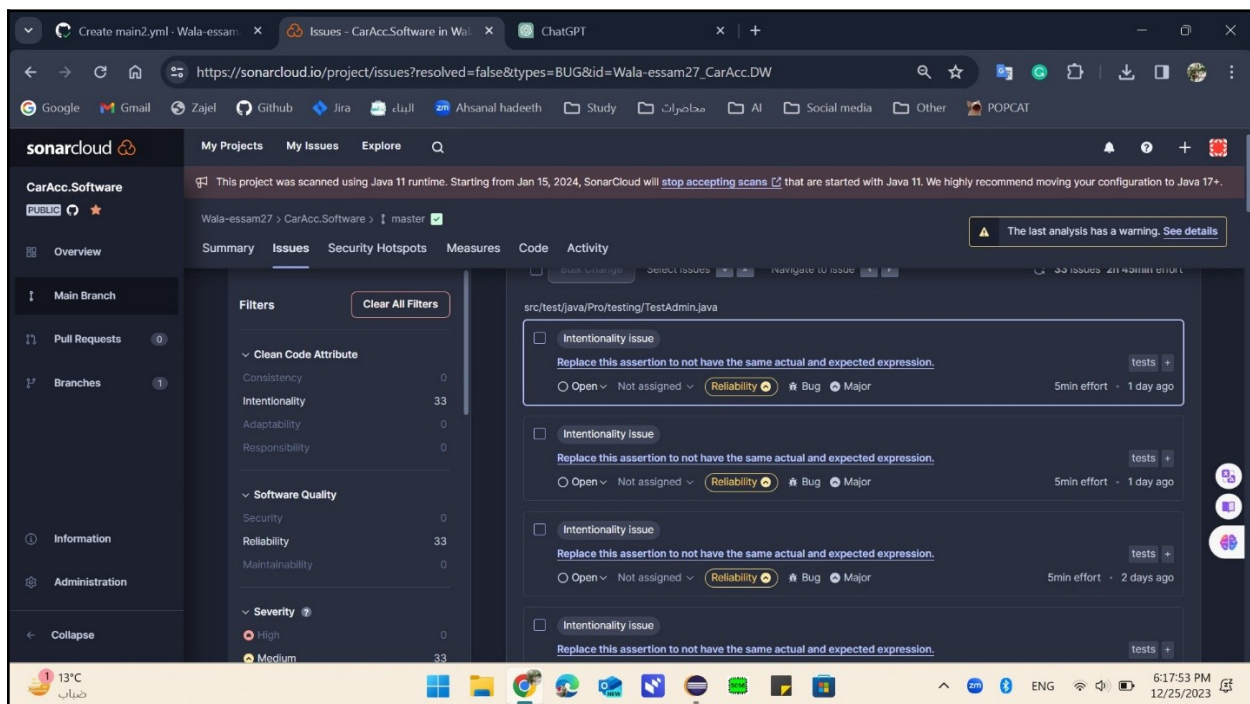
➤ The last percentage after some changes:

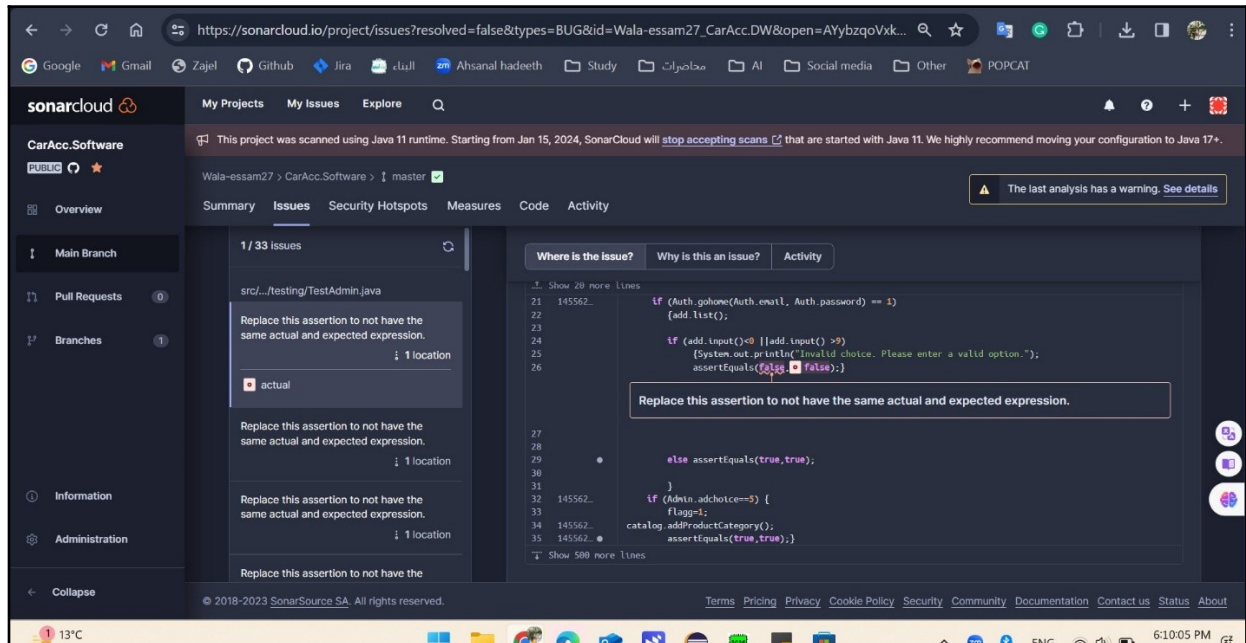




➤ Here I will add some of issues, bad smells, and security hotspot:

1. Before: there were 33 bug from these because I was used it in wrong way.





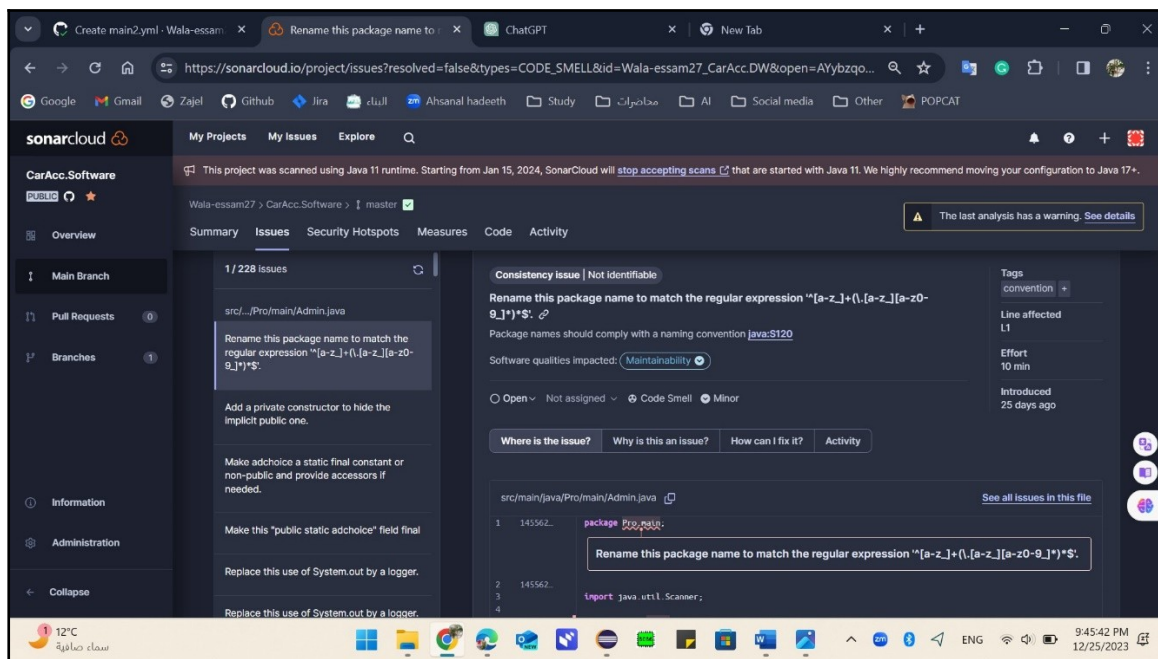
After:

```

• @Then("done deleting")
public void done_deleting() {
    if (flagg==1) assertEquals(true,catalog.deleteProduct());
}

```

2. Before:



After:

```
> src/main/java
> promain
3
4
```

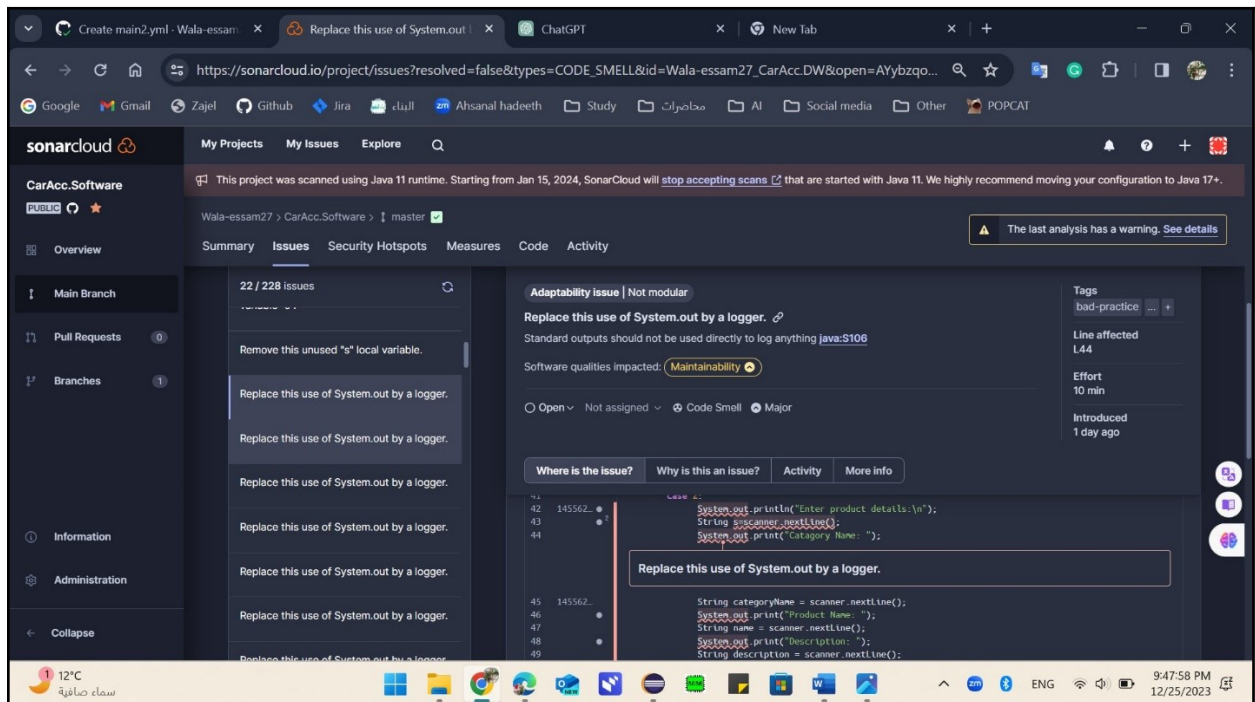
3. Before:



After:

```
static int flag=0;
public static int custchoice=0;
private static final Logger LOGGER = Logger.getLogger(Customer.class);
```

4. Before: built-in formatting to construct the argument for the `LOGGER.info()`.



After:

```
ProductCatalog catalog = new ProductCatalog();
Scanner scanner = new Scanner(System.in);
while ( true) {
    Customer cust = new Customer();
    System.out.println("\n Customer Menu: \n");
    System.out.println("1. View profile.");
    System.out.println("2. Edit profile.");
    System.out.println("3. View products.");
    System.out.println("4. Search products.");
    System.out.println("5. Purchase products.");
    System.out.println("6. View order history and installation requests.");
    System.out.println("7. Delete an order.");
    System.out.println("0. Exit");

    System.out.print("\nEnter your choice: ");

    custchoice = scanner.nextInt();
}
```

```
LOGGER.info("\n===== Customer Menu: =====\n" +
    "1. View profile.\n" +
    "2. Edit profile.\n" +
    "3. View products.\n" +
    "4. Search products.\n" +
    "5. Purchase products.\n" +
    "6. View order history and installation requests.\n" +
    "7. Delete an order.\n" +
    "0. Exit"+
    "\n\nEnter your choice: ");
```

5. Before:

Delete the notes because it is a code smell, but I forget to screenshot this issue.

After:

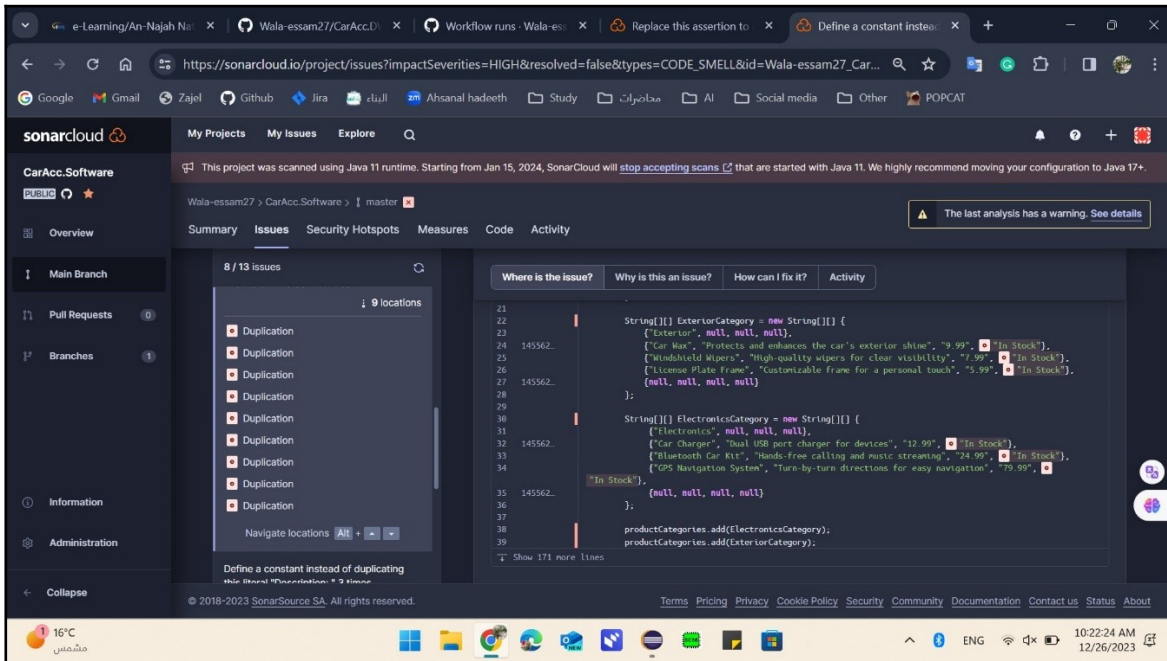
```

}

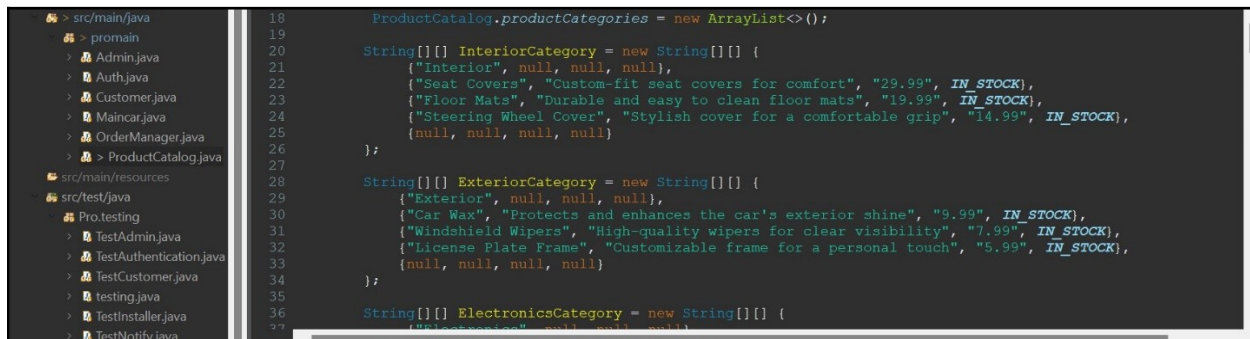
if (foundProducts.isEmpty()) {
    System.out.println("No products found matching the keyword: " + keyword);
} else {
    System.out.println("\nFound Products Matching '" + keyword + "':");
    for (String[] product : foundProducts) {
        //System.out.println("Category: " + product[0]);
        System.out.println("\nProduct: " + product[0]);
        System.out.println("Description: " + product[1]);
        System.out.println("Price: " + product[2]);
        System.out.println("Availability: " + product[3]);
        System.out.println("-----");
    }
}
```

```
if (product != null && product[0] != null && product[0].equalsIgnoreCase(productName)) {
    // Found the product, set its information to null
    product[0] = null;
    product[1] = null;
    product[2] = null;
}
```

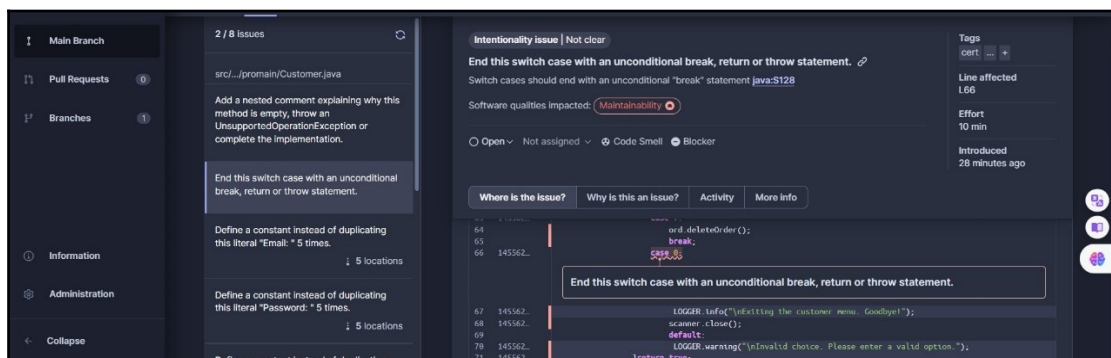
6. Before: Make the duplicates constant.



After:



7. Before:

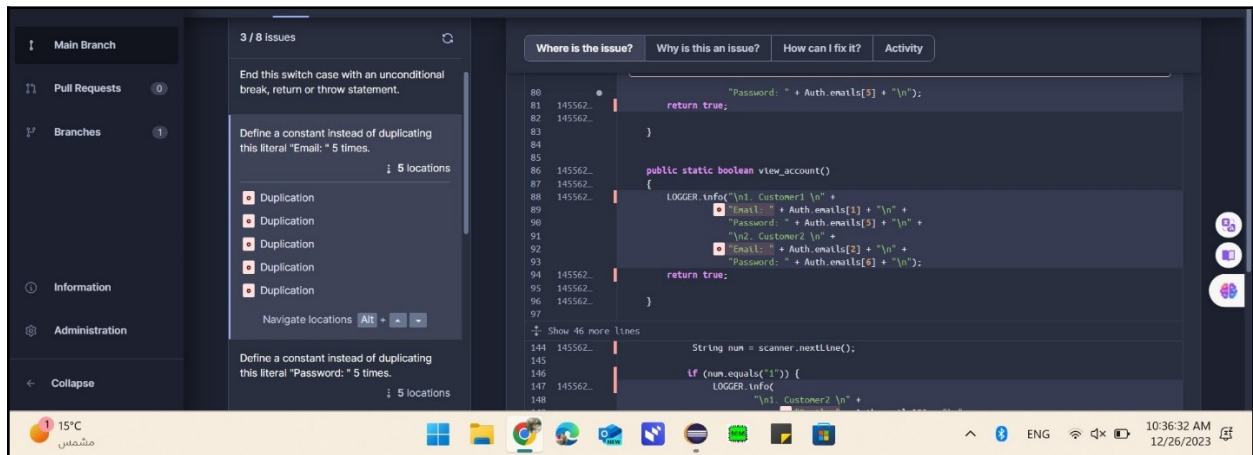


After:

```
        break;
    case 7:
        ord.deleteOrder();
        break;
    case 0:
        LOGGER.info("\nExiting the customer menu. Goodbye!");
        scanner.close();
        return true;
    default:

```

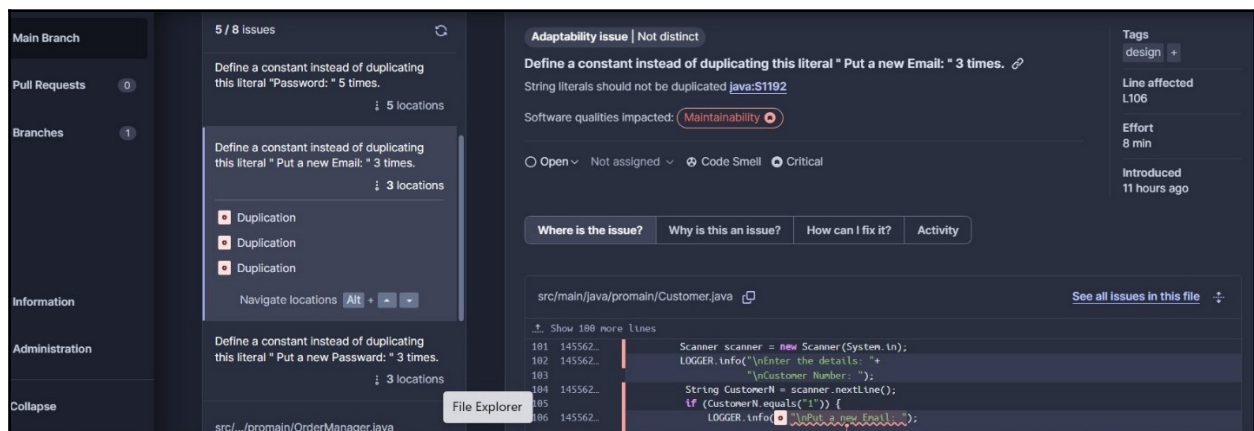
8. Before: Make the duplicates constant.



After:

```
public static boolean view_account()
{
    LOGGER.info("\n1. Customer1 \n" +
        EMAIL + Auth.emails[1] + "\n" +
        PASSWORD + Auth.emails[5] + "\n" +
        "\n2. Customer2 \n" +
        EMAIL + Auth.emails[2] + "\n" +
        PASSWORD + Auth.emails[6] + "\n");
    return true;
}
```

9. Before:



After:

```
78 String CustomerN = scanner.nextLine();
79 if (CustomerN.equals("1")) {
80     LOGGER.info(PUT_A_NEW_EMAIL);
81     String em = scanner.nextLine();
82     LOGGER.info("\nPut a new Password: ");
83     String pass = scanner.nextLine();
84     Auth.emails[1]=em;
85     Auth.emails[5]=pass;
86 }
87 else if (CustomerN.equals("2")) {
88     LOGGER.info(PUT_A_NEW_EMAIL);
89     String em = scanner.nextLine();
90     LOGGER.info("\nPut a new Password: ");
```

10.Before:

```
case 2:
    LOGGER.info("\nEnter product details:\n");
    String s=scanner.nextLine();
```

Remove this useless assignment to local variable "s".

After:

```
LOGGER.info("\nEnter product details:\n");
scanner.nextLine();
LOGGER.info("\nEnter product details:\n");
```

11. Before: the security hotspot.

```
34     return true;
35 }
36
37 145562... public static int gohome(String email,String password) {
```

Remove this unused method parameter "password".

'password' detected in this expression, review this potentially hard-coded password. [🔗](#)

Hard-coded passwords are security-sensitive [java:S2068](#)

Status: To Review

This Security Hotspot needs to be reviewed to assess whether the code poses a risk.

Review

Review priority: 🔴 High

Category: Authentication

Assignee: Not assigned ▼

Where is the risk?	What's the risk?	Assess the risk	How can I fix it?	Activity
<ul style="list-style-type: none"> • Store the credentials in a database. • Use your cloud provider's service for managing secrets. • If a password has been disclosed through the source code: change it. 				

Sensitive Code Example

```
String username = "steve";
String password = "blue";
Connection conn = DriverManager.getConnection("jdbc:mysql://localhost/test?" +
    "user=" + username + "&password=" + password); // Sensitive
```

After:

```
}
else if (email.equals(emails[3]) && p.equals("pass")) {
    LOGGER.info("\nWelcome Our INSTALLER!");
}
```

➤ Then we build in GitHub actions:

This screenshot shows the GitHub Actions interface for a workflow named 'Create main2.yml'. The job 'Build and analyze' is highlighted in the left sidebar and is shown as successful. The main panel displays the job's steps and their durations:

Step	Duration
Set up job	1s
Run actions/checkout@v3	1s
Set up JDK 17	6s
Cache SonarCloud packages	0s
Cache Maven packages	0s
Build and analyze	35s

The interface includes a search bar for logs, a 'Re-run all jobs' button, and a sidebar with links to Summary, Jobs, Run details, Usage, and Workflow file.

This screenshot shows the GitHub Actions interface for a workflow named 'build-on-ubuntu'. The job is shown as successful. The main panel displays the job's steps and their durations:

Step	Duration
Set up job	1s
Step 1 - Checkout main branch from GitHub	0s
Step 2 - Set up JDK 1.11	11s
Step 3 - Build Maven Project	7s
Step 4 - run test cases	3s

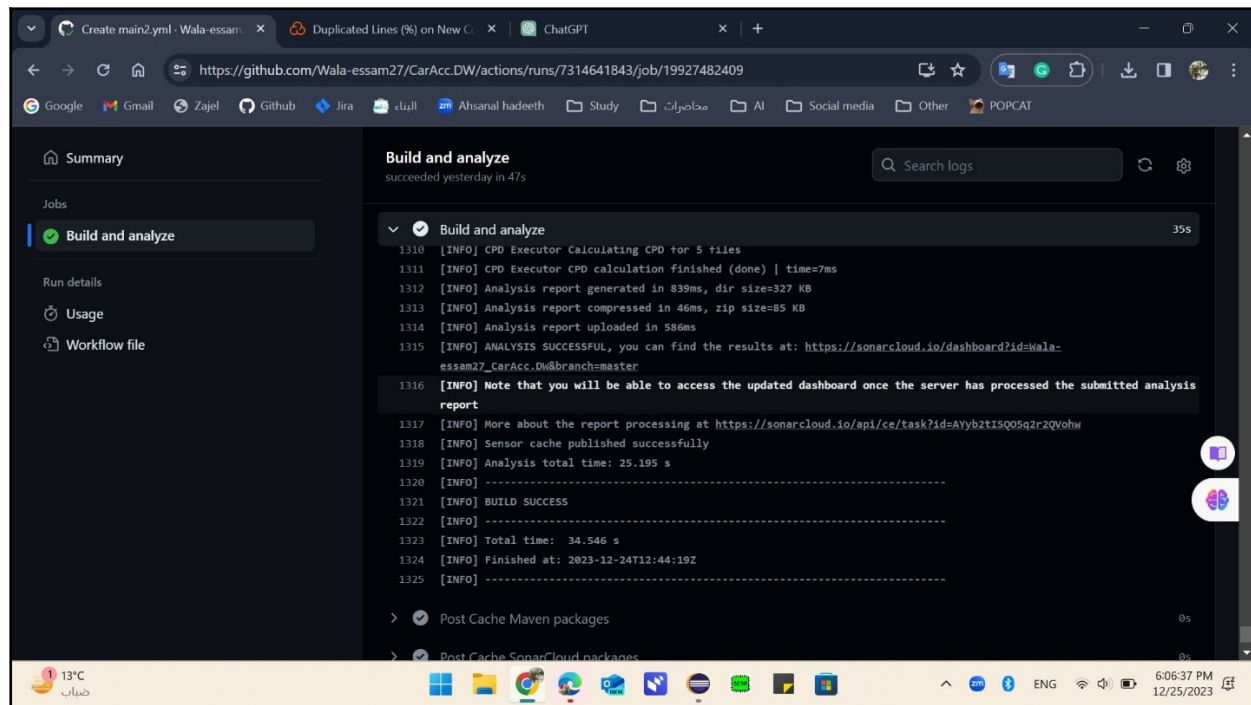
The interface includes a search bar for logs and a sidebar with links to Summary, Jobs, Run details, Usage, and Workflow file.

This screenshot shows the GitHub Actions interface for a workflow named 'Build and analyze'. The job is shown as successful. The main panel displays the job's steps and their durations:

Step	Duration
Set up job	1s
Run actions/checkout@v3	1s
Set up JDK 17	6s
Cache SonarCloud packages	0s
Cache Maven packages	0s
Build and analyze	35s
Post Cache Maven packages	0s
Post Cache SonarCloud packages	0s
Post Set up JDK 17	0s
Post Run actions/checkout@v3	0s
Complete job	0s

The interface includes a search bar for logs and a sidebar with links to Summary, Jobs, Run details, Usage, and Workflow file.

➤ console output:



```
1310 [INFO] CPD Executor Calculating CPD for 5 files
1311 [INFO] CPD Executor CPD calculation finished (done) | time=7ms
1312 [INFO] Analysis report generated in 839ms, dir size=327 KB
1313 [INFO] Analysis report compressed in 46ms, zip size=85 KB
1314 [INFO] Analysis report uploaded in 586ms
1315 [INFO] ANALYSIS SUCCESSFUL, you can find the results at: https://sonarcloud.io/dashboard?id=Wala-essam27_CarAcc.DW&branch=master
1316 [INFO] Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
1317 [INFO] More about the report processing at https://sonarcloud.io/api/ce/task?id=AYyb2tISQ05q2r2QVohw
1318 [INFO] Sensor cache published successfully
1319 [INFO] Analysis total time: 25.195 s
1320 [INFO] -----
1321 [INFO] BUILD SUCCESS
1322 [INFO] -----
1323 [INFO] Total time: 34.546 s
1324 [INFO] Finished at: 2023-12-24T12:44:19Z
1325 [INFO] -----

> Post Cache Maven packages 0s
> Post Cache SonarCloud packages 0s
```

Thank you.