

Team 4

# WEARE APP

---

F I N A L   P R O J E C T   D E M O



# WEare Project

- What is WEare?
- What is the project about?
- Which are the main functionalities for testing?



# How we started?

01

## Planning

Early planning of the activities , creating Test Plan, high-level test cases

02

## Setting Up the Projects

Setting up the projects in Jira (integration with X-ray), GitHub and Postman

03

## Assigning Tasks

Creating a task planning document

Main challenges and how we overcome them.

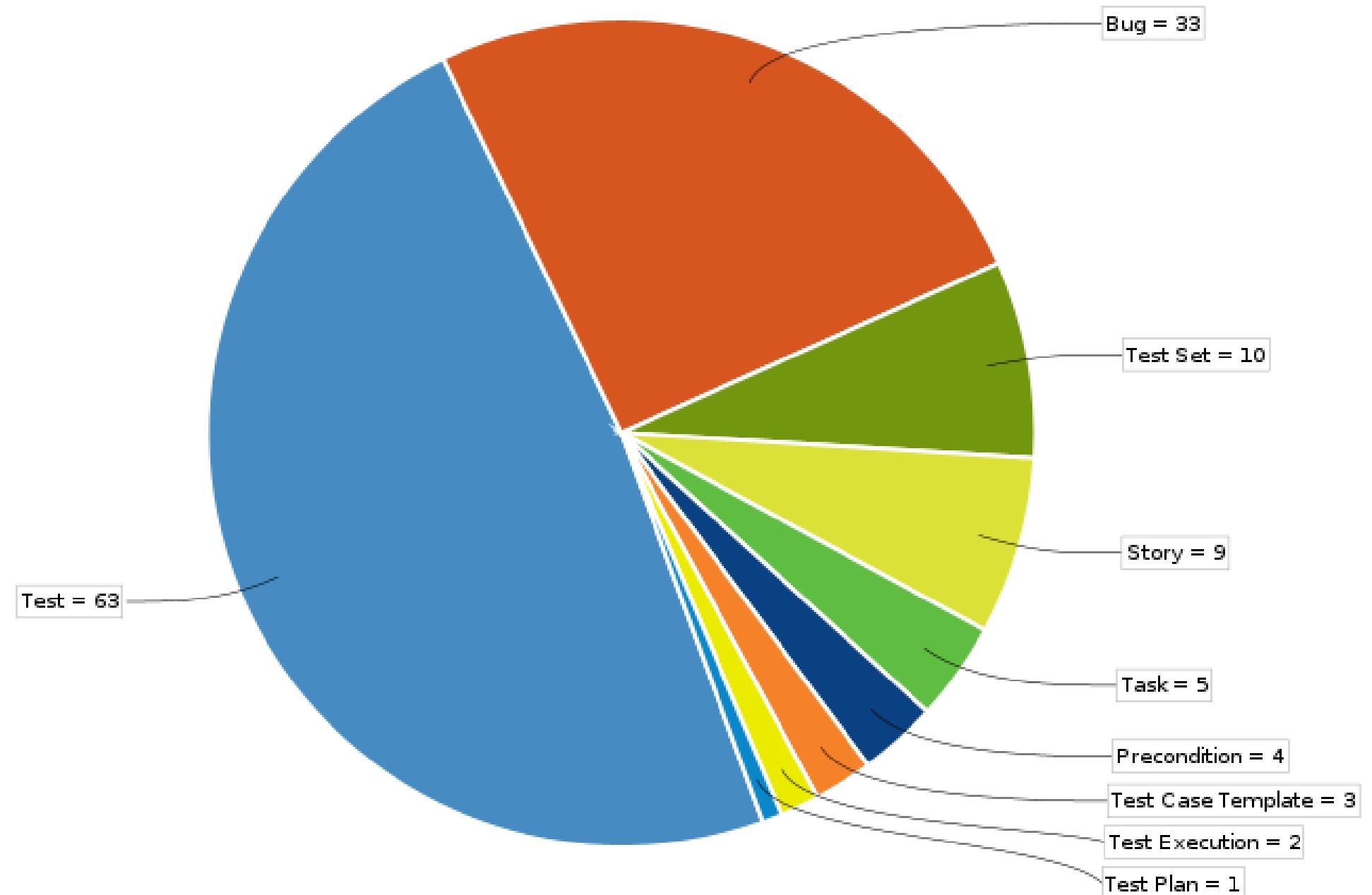
# Hristo's Spotlight

My overall contribution for the project



# Manual Testing

- Creating templates (for test case, bug and preconditions) in Jira
- Organizing the work in Sprints according to the Test Plan
- Assigning tasks
- Creating test cases
- Logging bugs



# Manual Testing

32 PASSED 30 FAILED TOTAL TESTS: 62

Search  Only My Test Runs Filters ▾

Showing 25 of 62 issues

1	WSN-86 [Posting and Interaction] Delete a Comment MANUAL DONE INTERACTION POSTING	PASSED	M	grid	list
2	WSN-87 [Posting and Interaction] Edit a Comment MANUAL DONE INTERACTION POSTING	FAILED	M	grid	list
3	WSN-10 [Posting and Interaction] Create a Comment Successfully MANUAL DONE INTERACTION POSTING	PASSED	M	grid	list
4	WSN-11 [Posting and Interaction] Create a Comment with Blank Content MANUAL DONE INTERACTION POSTING	FAILED	M	grid	list
5	WSN-21 [Posting and Interaction] Comment on the Post with a Long Message MANUAL DONE INTERACTION POSTING	FAILED	M	grid	list
6	WSN-32 [Posting and Interaction] Set up How Many of the Newest Comments Should be visible under Each Post MANUAL DONE INTERACTION POSTING	FAILED	M	grid	list
7	WSN-59 [Posting and Interaction] Attach a file (image, document, etc.) in the comment to a post MANUAL DONE INTERACTION POSTING	FAILED	M	grid	list
8	WSN-25 [Posting and Interaction] Create a Post successfully MANUAL DONE INTERACTION POSTING	PASSED	M	grid	list
9	WSN-26 [Posting and Interaction] Create a Post with Text and Uploaded Picture more than 1MB MANUAL DONE INTERACTION POSTING	FAILED	M	grid	list

# Automation Testing

01

## API Testing

- Creating Postman collection for the main functionalities

02

## Selenium Webdriver Tests

- Creating tests using Java code in IntelliJ



# Postman

WEare Project Demo

A61 - Team 4 Workspace    New Import    Overview    WEare    WEare API

**WEare - Run results**  
Ran today at 09:50:29 · [View all runs](#)

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	WEare API	1	603ms	14	13 ms

All Tests Passed (14) Failed (0) Skipped (0) [View Summary](#)

Iteration 1

**GET** Navigate to the URL  
http://localhost:8081  
PASS Status code is 200    200 OK 11 ms 28.601 KB

**GET** Login  
http://localhost:8081/login  
PASS Status code is 200    200 OK 5 ms 10.828 KB

**POST** Create a New Post  
http://localhost:8081/api/post/auth/creator  
PASS Status code is 200    200 OK 25 ms 528 B

**POST** Filter Posts  
http://localhost:8081/posts  
PASS Status code is 200    200 OK 6 ms 25.886 KB

**GET** Find All Posts  
http://localhost:8081/api/post/  
PASS Status code is 200    200 OK 30 ms 7.987 KB

**PUT** Edit Post  
http://localhost:8081/api/post/auth/editor?postId=90  
PASS Status code is 200    200 OK 5 ms 305 B

**POST** Like a Post  
http://localhost:8081/api/post/auth/likesUp?postId=90  
PASS Status code is 200    200 OK 10 ms 1.733 KB

**DELETE** Delete a Post  
http://localhost:8081/api/post/auth/manager?postId=108  
PASS Status code is 200    200 OK 20 ms 305 B

Online Find and replace Cancel    Postbox Runner Start Brown Cookies Vault Trash

# Postman

WEare Project Demo

The screenshot shows the Postman application interface with the title "WEare Project Demo". On the left, there's a sidebar with sections for Collections, Environments, and History. The main area displays the "WEare - Run results" for a run that ran at 09:50:29. The results table shows the following details:

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	WEare API	1	603ms	14	13 ms

Below the table, it says "All Tests Passed (14) Failed (0) Skipped (0)" and provides a "View Summary" link.

The results list includes the following API calls:

- POST Like a Post**  
http://localhost:8081/api/post/auth/likesUp?postId=90  
Status: PASS Status code is 200
- DELETE Delete a Post**  
http://localhost:8081/api/post/auth/manager?postId=108  
Status: PASS Status code is 200
- POST Create a New Comment**  
http://localhost:8081/api/comment/auth/creator  
Status: PASS Status code is 200
- POST Like a Comment**  
http://localhost:8081/api/comment/auth/likesUp?commentId=152  
Status: PASS Status code is 200
- PUT Edit Comment**  
http://localhost:8081/api/comment/auth/editor?commentId=217&content=fddsf  
Status: PASS Status code is 200
- GET Get All Comments**  
http://localhost:8081/api/comment  
Status: PASS Status code is 200
- DELETE Delete Comment**  
http://localhost:8081/api/comment/auth/manager  
Status: PASS Status code is 200
- POST Authenticate**  
http://localhost:8081/authenticate  
Status: 200 OK 6 ms 28.601 KB

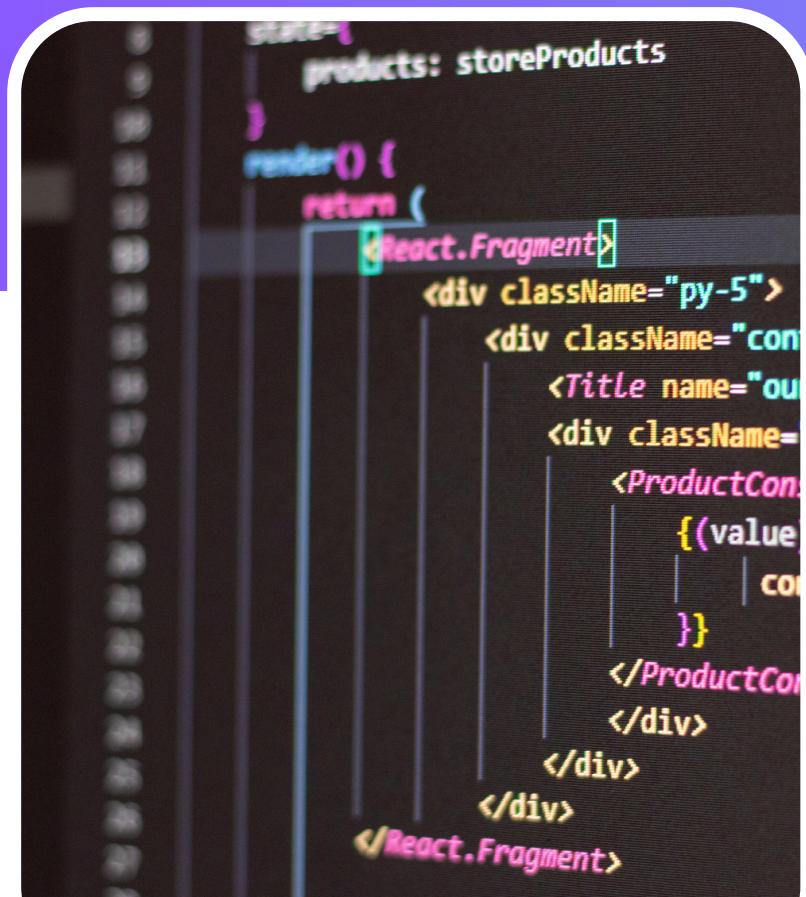
# Mina's Spotlight

My overall contribution for the project



# Test Automation Framework

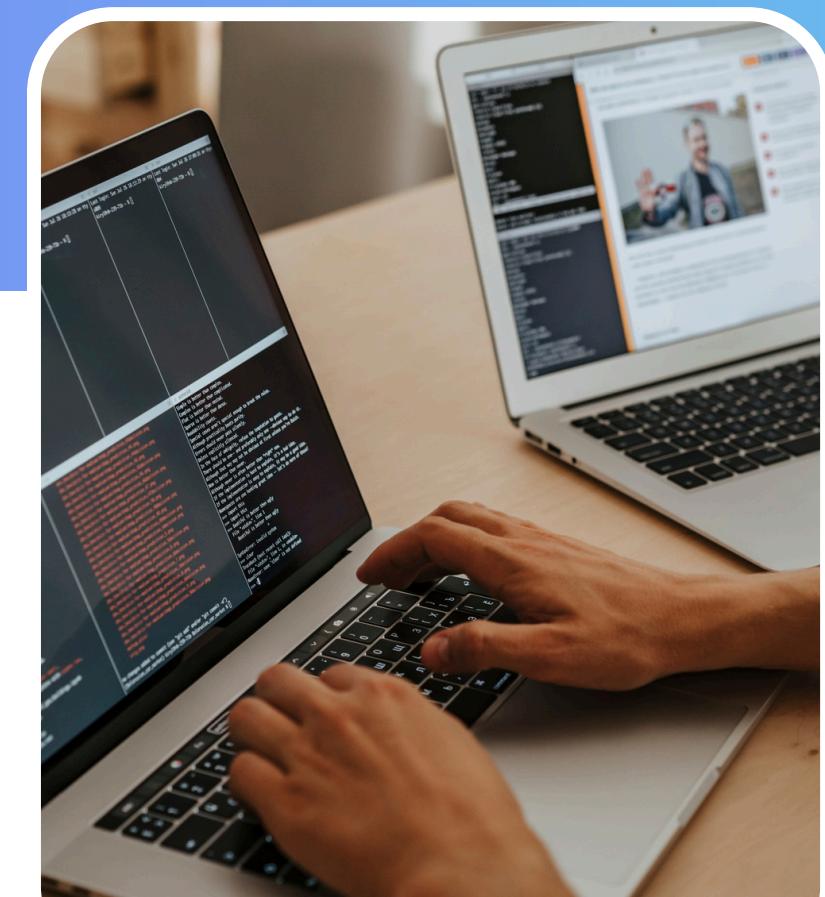
01 Our approach



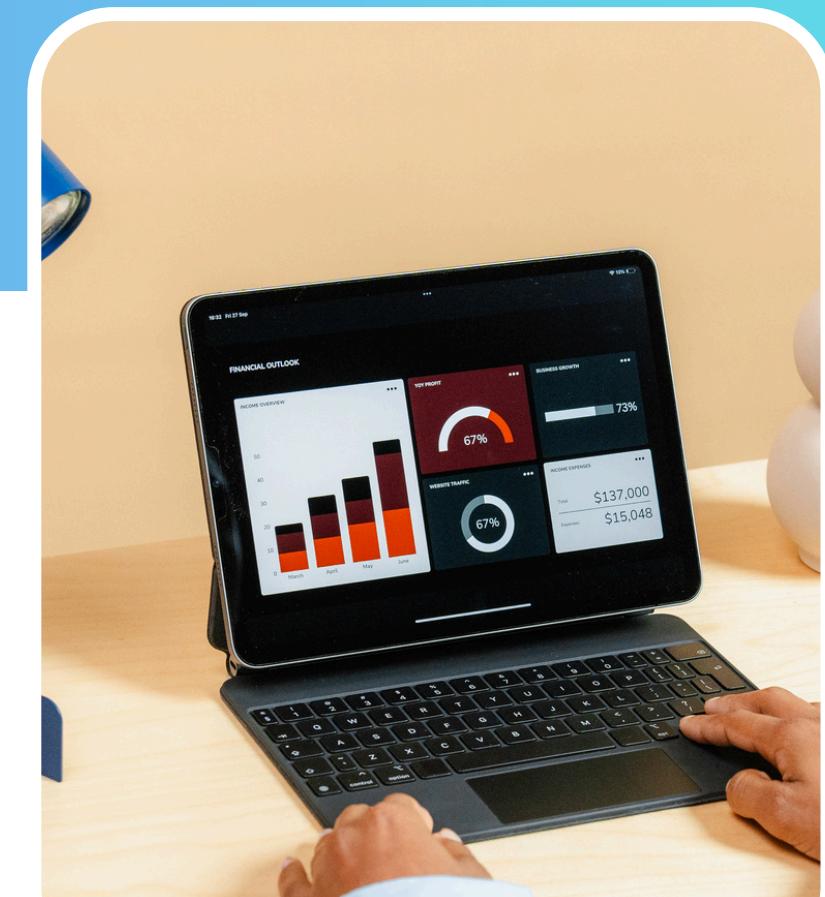
02 How we organized the structure of the project



03 Writing Tests



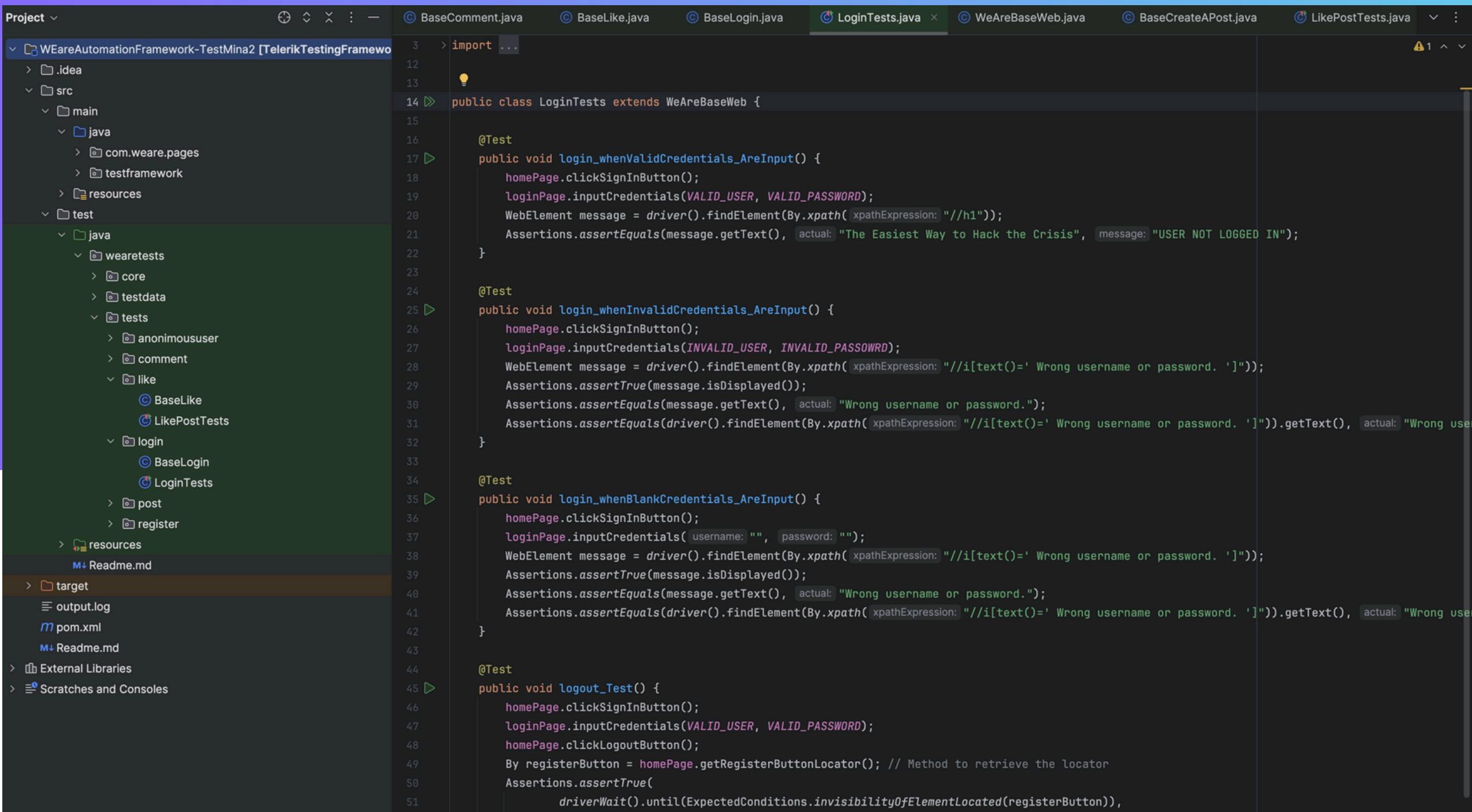
04 Tests execution



# Test Automation Framework

- Main difficulties

- How we overcome them



The screenshot shows an IDE interface with a dark theme. On the left is a project tree for 'WEareAutomationFramework-TestMina2 [TelerikTestingFramework]'. The 'src' folder contains 'main' and 'test' packages. The 'test/wearetests/tests/login' package is selected, showing a file named 'LoginTests.java'. The code in 'LoginTests.java' contains several @Test annotations for testing login functionality, including checks for valid credentials, invalid credentials, and blank credentials. The code uses Selenium WebDriver and Assertions from the 'org.openqa.selenium' and 'org.junit' namespaces.

```
import ...  
public class LoginTests extends WeAreBaseWeb {  
  
    @Test  
    public void login_whenValidCredentials_AreInput() {  
        homepage.clickSignInButton();  
        LoginPage.inputCredentials("VALID_USER", "VALID_PASSWORD");  
        WebElement message = driver().findElement(By.xpath("//h1"));  
        Assertions.assertEquals(message.getText(), actual: "The Easiest Way to Hack the Crisis", message: "USER NOT LOGGED IN");  
    }  
  
    @Test  
    public void login_whenInvalidCredentials_AreInput() {  
        homepage.clickSignInButton();  
        LoginPage.inputCredentials("INVALID_USER", "INVALID_PASSWORD");  
        WebElement message = driver().findElement(By.xpath("//i[text()=' Wrong username or password. ']"));  
        Assertions.assertTrue(message.isDisplayed());  
        Assertions.assertEquals(message.getText(), actual: "Wrong username or password.");  
        Assertions.assertEquals(driver().findElement(By.xpath("//i[text()=' Wrong username or password. ']")).getText(), actual: "Wrong user");  
    }  
  
    @Test  
    public void login_whenBlankCredentials_AreInput() {  
        homepage.clickSignInButton();  
        LoginPage.inputCredentials("", "");  
        WebElement message = driver().findElement(By.xpath("//i[text()=' Wrong username or password. ']"));  
        Assertions.assertTrue(message.isDisplayed());  
        Assertions.assertEquals(message.getText(), actual: "Wrong username or password.");  
        Assertions.assertEquals(driver().findElement(By.xpath("//i[text()=' Wrong username or password. ']")).getText(), actual: "Wrong user");  
    }  
  
    @Test  
    public void logout_Test() {  
        homepage.clickSignInButton();  
        LoginPage.inputCredentials("VALID_USER", "VALID_PASSWORD");  
        homepage.clickLogoutButton();  
        By registerButton = homepage.getRegisterButtonLocator(); // Method to retrieve the locator  
        Assertions.assertTrue(driverWait().until(ExpectedConditions.invisibilityOfElementLocated(registerButton)),  
            "The element is still visible");  
    }  
}
```



# Reporting



# Vladislav's Spotlight

My overall contribution for the project

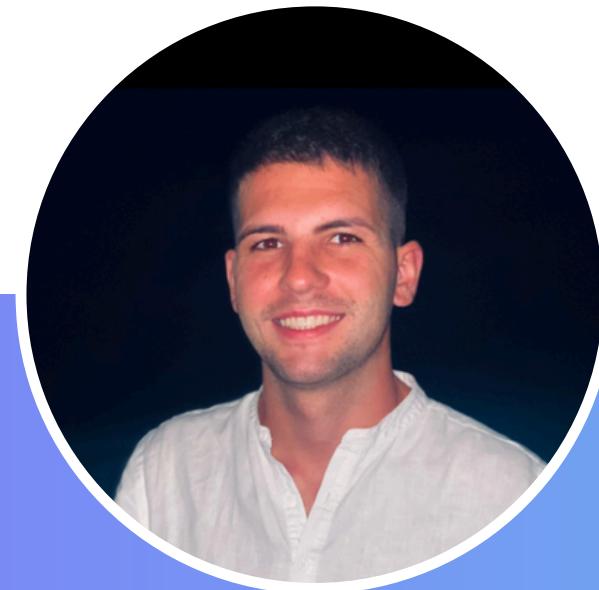


# Meet Our Team



**Mina  
Mladenova**

[mina.mladenova.a61@learn.telerikacademy.com](mailto:mina.mladenova.a61@learn.telerikacademy.com)



**Vladislav  
Adamov**

[vladislav.adamov.a61@learn.telerikacademy.com](mailto:vladislav.adamov.a61@learn.telerikacademy.com)



**Hristo  
Hristov**

[hristo.hristov.a61@learn.telerikacademy.com](mailto:hristo.hristov.a61@learn.telerikacademy.com)

WEare Project Demo

THANK YOU

---