

WIPRO NGA Program – Product Testing

Capstone Project Presentation – 3rd and 4th Sept
2025

Project Title Here - Zero Bank Automation Project

Presented by – Bairi Rakesh Goud

Table of contents

- Project Overview
- Tools and Technologies
- Project Architecture
- Implemented Features
- Test Cases
- Challenges Faced
- Conclusion

Project overview

- Automate key functionalities of the Zero Banking application using **Selenium with Python** to ensure accurate and efficient testing of banking operations.
- Helps in **reducing manual testing effort**, improves accuracy, and ensures faster validation of banking workflows.

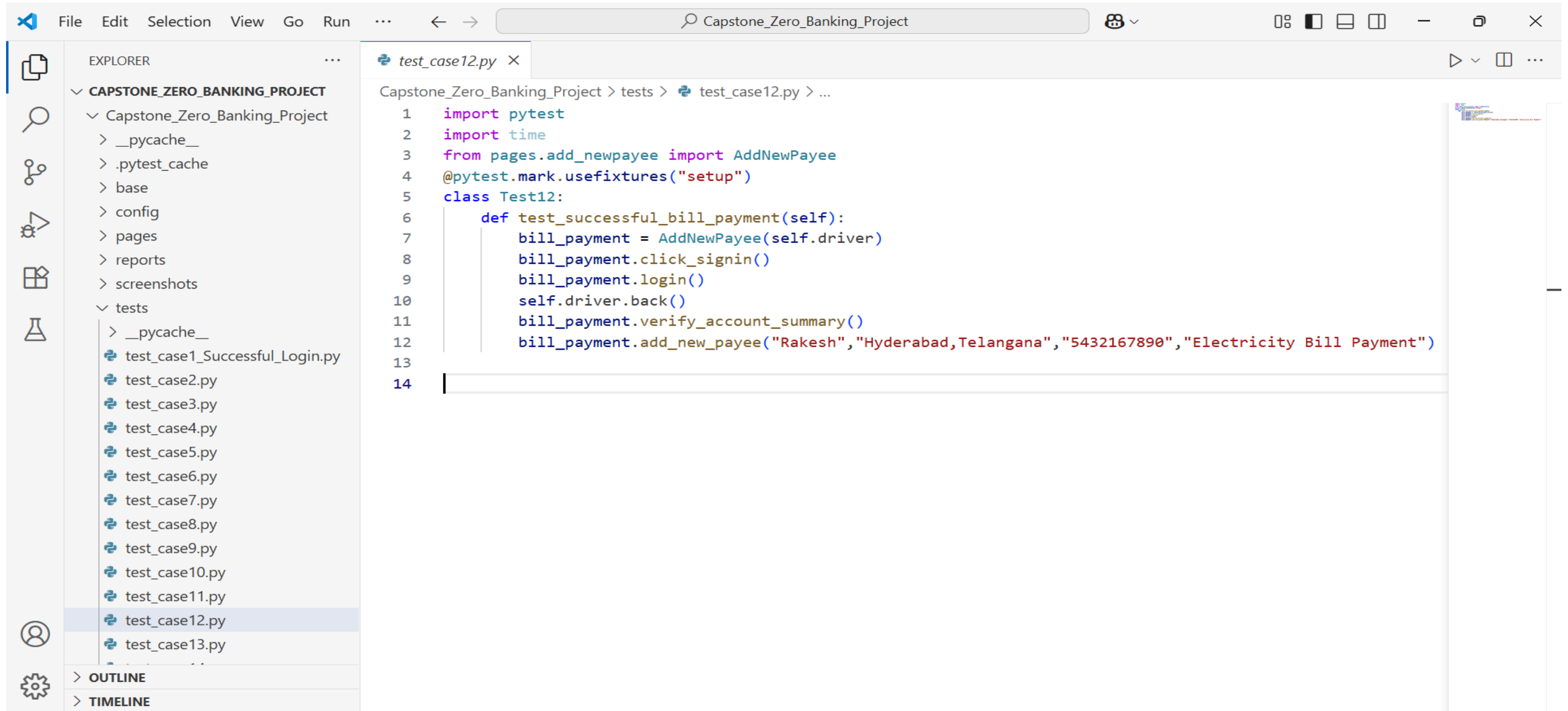
Tools and Technologies

- **Python** – Scripting language for automation.
- **Selenium WebDriver** – Browser automation.
- **PyTest** – Test framework.
- **Allure** – Test reports generation.
- **VS Code** – IDE for development.
- **Website URL:** <http://zero.webappsecurity.com/>

Project Architecture

- **Page Object Model (POM)** is implemented for maintainability.
- **BasePage class** handles common operations like click, enter text, hover, etc.
- **Separate page classes** for Login, Account Verification, Fund Transfer, and Error Handling.
- **PyTest** used for test execution.
- **Allure Reports** for visualization of test results.

Project Structure



The screenshot displays the Visual Studio Code interface for a project named 'Capstone_Zero_Banking_Project'. The Explorer sidebar on the left shows the project's directory structure, including folders like '__pycache__', '.pytest_cache', 'base', 'config', 'pages', 'reports', 'screenshots', and 'tests'. The 'tests' folder is expanded, showing a list of test files: '__pycache__', 'test_case1_Successful_Login.py', 'test_case2.py', 'test_case3.py', 'test_case4.py', 'test_case5.py', 'test_case6.py', 'test_case7.py', 'test_case8.py', 'test_case9.py', 'test_case10.py', 'test_case11.py', 'test_case12.py' (which is selected), and 'test_case13.py'. Below the Explorer, there are sections for 'OUTLINE' and 'TIMELINE'. The main editor window shows the code for 'test_case12.py', which is a pytest test case. The code includes imports for 'pytest' and 'time', a fixture 'AddNewPayee' from 'pages.add_newpayee', and a test class 'Test12' with a method 'test_successful_bill_payment' that performs a series of actions on a web application, including logging in, adding a new payee, and verifying the account summary.

```
1 import pytest
2 import time
3 from pages.add_newpayee import AddNewPayee
4 @pytest.mark.usefixtures("setup")
5 class Test12:
6     def test_successful_bill_payment(self):
7         bill_payment = AddNewPayee(self.driver)
8         bill_payment.click_signin()
9         bill_payment.login()
10        self.driver.back()
11        bill_payment.verify_account_summary()
12        bill_payment.add_new_payee("Rakesh", "Hyderabad,Telangana", "5432167890", "Electricity Bill Payment")
13
14
```

Handling Webpage

Zero Bank

Search

Signin


HOME

ONLINE BANKING


FEEDBACK

Online Banking


Pay bills easily




Our Bank is trusted by over 1,000,000 customers world wide. Sign in now!

 Account Summary

See all of your account balances at a glance.

 Account Activity

View the most up-to-date listings of your deposits, withdrawals, interest payments, and other transactions.

 Transfer Funds

Safely and securely transfer funds between accounts.

Implemented Features

- ✓ **Login Automation** – Validates user login.
- ✓ **Account Type Verification** – Checks all account types are visible.
- ✓ **Fund Transfer** – Automates transfer between accounts.
- ✓ **Insufficient Funds Handling** – Validates error messages when balance is low.
- ✓ **Reporting** – Generates Allure reports for each test run.

Test Cases

➤ Test Case 1: Successful Login

Steps: Open the website, enter valid username and password, and click Sign In.

Expected Result: User should be redirected to the account summary page, confirming successful login.

➤ Test Case 2: Validate Account Types

Steps: Navigate to Account Summary, check all account types displayed.

Expected Result: “Checking,” “Savings,” “Credit Card,” and “Loan” account types should be visible.

Test Cases

➤ Test Case 3: Fund Transfer with Sufficient Balance

Steps: Select a valid “From Account” and “To Account,” enter amount less than balance, and submit.

Expected Result: Transaction should succeed and confirmation message should appear.

➤ Test Case 4: Fund Transfer with Insufficient Balance

Steps: Select a “From Account” with insufficient funds, enter an amount greater than balance, and submit.

Expected Result: Error message “Insufficient funds.” should be displayed, and transaction should not be processed.

Test Cases

☐ **Test Case 5: Validate Account Types**

Login and navigate to the **Account Summary** page.

Verify that the following account types are displayed:

Checking

Savings

Credit Card

Loan

Verify that funds can be transferred between accounts.

Test Cases

Test Case 6: Successful Fund Transfer

Login and navigate to the **Fund Transfer** page.
Select "**From Account**" and "**To Account**".
Enter a valid amount.
Click **Submit**.
Confirm the transfer.

Test Case 7: Fund Transfer with Insufficient Balance

Select an account with **insufficient funds** as the "From Account".
Enter an amount greater than the available balance.
Click **Submit**.

Test Cases

❑ Test Case 8: Fund Transfer with Negative Amount

Enter a **negative value** in the amount field.
Click **Submit**.

Test Case 9: Successful Bill Payment

Login and go to **Pay Bills**.
Select a payee.
Enter a valid amount and date.
Click **Pay**.

Test Cases

Test Case 10: Schedule a Bill Payment for a Future Date

Enter a future date in the date field.
Click **Pay**.

☐ **Test Case 11: Pay Bills Without Entering Amount**

Leave the amount field empty.
Click **Pay**.

Test Cases

Test Case 12: Successfully Add a New Payee

Navigate to **Add New Payee**.

Enter payee details (Name, Address, Account Number).

Click **Add**.

Test Case 13: Add Payee with Missing Details

Leave any field empty.

Click **Add**.

Objective: Verify that account statements can be downloaded.

Test Cases

Test Case 14: Download Account Statement

Navigate to **Statements & Documents**.

Select an account and a date range.

Click **Download PDF**.

Objective: Ensure users can log out securely.

☐ Test Case 15: Logout Successfully

Login:

The script opens the login page and enters valid credentials (username and password) to log in to the Zero Bank platform.

After clicking the login button, it waits for the account dashboard to load and verifies that the login was successful by checking for the presence of the dashboard.

Logout:

After a successful login, the script locates the **Logout button** (or link) by its ID (replace logout-btn with the actual ID of the logout element on the Zero Bank platform).

The script clicks the logout button to initiate the logout process.

Test Cases

Test Case 16: Verify Session Expiry After Logout

Logout of the application.
Click the **Back button** in the browser.

Screenshots

Zero Bank

Search

Settings

username

Account Summary

Account Activity

Transfer Funds

Pay Bills

My Money Map

Online Statements

Pay Saved Payee

Add New Payee

Purchase Foreign Currency

Make payments to your saved payees

Payee

Sprint

?

Account

Savings

Amount

\$

Date

2025-09-03

Please fill out this field.

Description

fund transfer

Pay

Screenshots

Zero Bank

Settings

username

Account Summary

Account Activity

Transfer Funds

Pay Bills

My Money Map

Online Statements

Pay Saved Payee

Add New Payee

Purchase Foreign Currency

Who are you paying?

Payee Name

Payee Address

Account

Payee Details

Please fill out this field.

Add

Screenshots

Zero Bank

Search

Settings

username

Account Summary

Account Activity

Transfer Funds

Pay Bills

My Money Map

Online Statements

Transfer Money & Make Payments

From Account

Checking(Avail. balance = \$ -500.2)

To Account

Loan(Avail. balance = \$ 780)

Amount

\$ -5000

Description

transaction complete

Descriptions appear for checking, savings, money market or market rate accounts only.

Continue

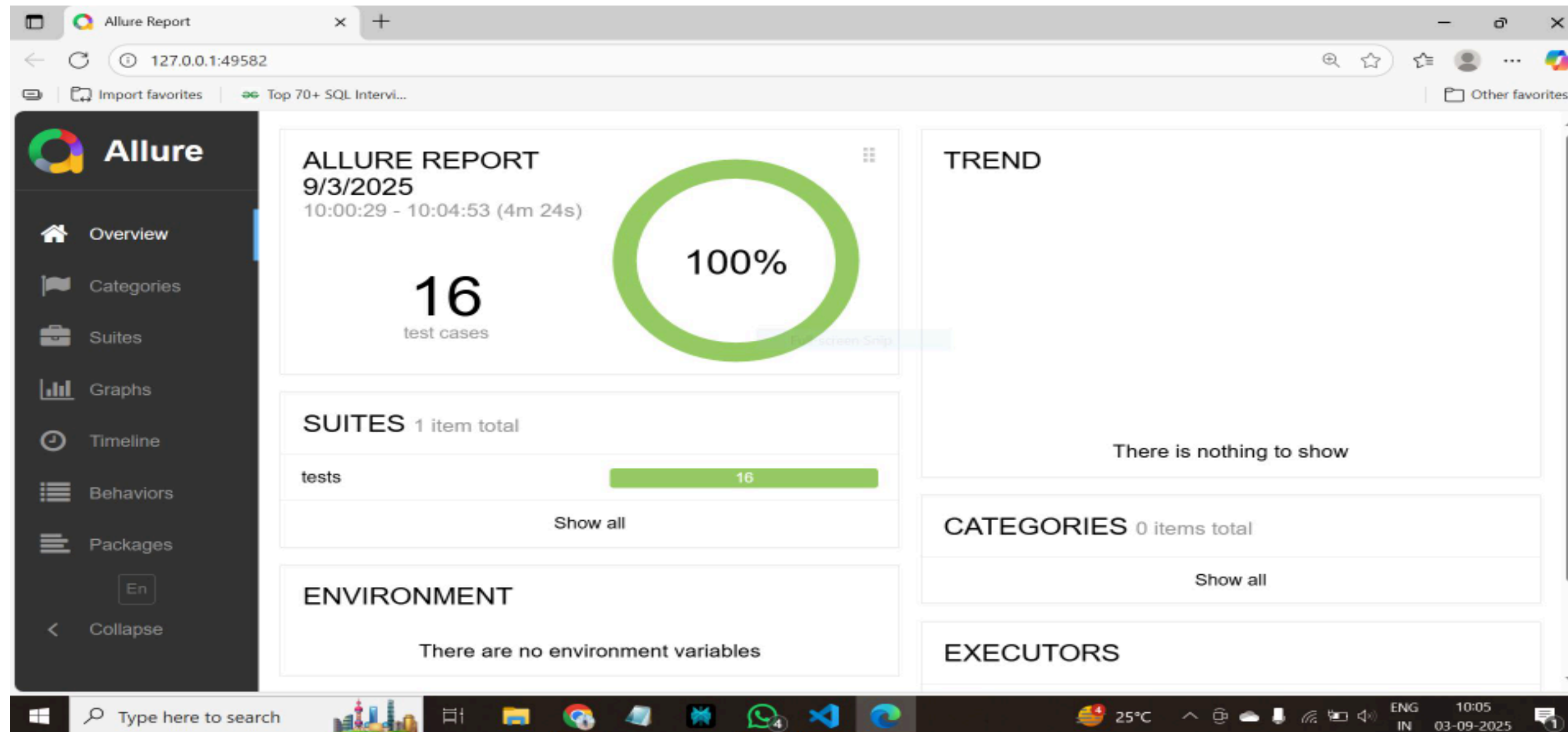
Download WebInspect

Terms of Use

Contact Micro Focus

ALLURE WITH SELENIUM

Allure Report



Conclusion

- Demonstrates the use of **Selenium and Python** for automating banking web application functionalities.
- Framework is **robust, reusable, and scalable**.
- Provides **reliable results** and **detailed reports**.
- Ensures **faster regression testing**.
- Improves **overall software quality**.



Rakesh.