Automation Framework Overview

The **Automation Testing Framework** for the Entrata website is designed to facilitate efficient testing of web applications through automation. This framework leverages the **Page Object Model** design pattern to enhance maintainability and readability, ensuring a robust testing process.

Key Features

1. Page Object Model (POM)

- **Modular Design**: Each page of the Entrata website is represented by a dedicated class, encapsulating the elements and actions specific to that page. This approach simplifies test case maintenance and enhances code reuse.

2. Test Cases

- **Navigation Tests**: Simple test cases have been created to automate the navigation to various pages on the Entrata website. These tests verify that users can navigate seamlessly and that the correct pages are displayed.
- **Web Form Automation**: A demo web form filling automation test has been implemented to validate form functionality using data.

3. Data Management

- **Properties File**: Configuration data, such as URLs and test credentials, are stored in a properties file. This eliminates hard-coded values in the test scripts, promoting better practice and ease of maintenance.
- **Constants Usage**: Data from the properties file is read and stored as constants, ensuring consistency across test cases, and reducing the risk of errors.

4. Test Management with TestNG

- **Test Suite Execution**: The framework utilizes **TestNG** to organize and execute a suite of test cases. This allows for easy management of tests, including grouping, prioritization, and parallel execution.

Technologies Used:

- **Selenium WebDriver**: Version 4.33.0
- **Java**: Version 17
- **TestNG**: Version 7.10.2
- **Maven**: Version 3.6.3

-The document and framework designed by Pavan Nanaware.

2. Framework Architecture

1. Components:

- **Test Scripts**: Contains the actual test cases organized by features.
- **Page Object Model**: Encapsulates web elements and actions for better maintainability.
- **Test Data**: External data files (properties file) for parameterized testing.
- **Reporting**: Utilizes **allure report** for detailed HTML reports.

2. Setup Instructions

Prerequisites:

1. **Java Development Kit (JDK) 11**
2. **Apache Maven**
3. **IDE**: Eclipse

Clone the Repository:

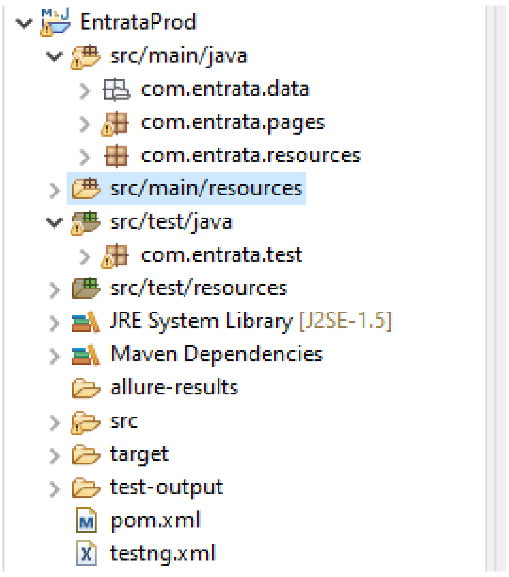
```
git clone https: https://github.com/Pavan7T/Entrata_Automation.git

cd automation-framework
```

Build the Project:

- mvn compile or mvn verify
- mvn test

Project File Structure:



```
EntrataProd
  src/main/java
    com.entrata.data
    com.entrata.pages
    com.entrata.resources
  src/main/resources
  src/test/java
    com.entrata.test
  src/test/resources
  JRE System Library [J2SE-1.5]
  Maven Dependencies
  allure-results
  src
  target
  test-output
  pom.xml
  testng.xml
```

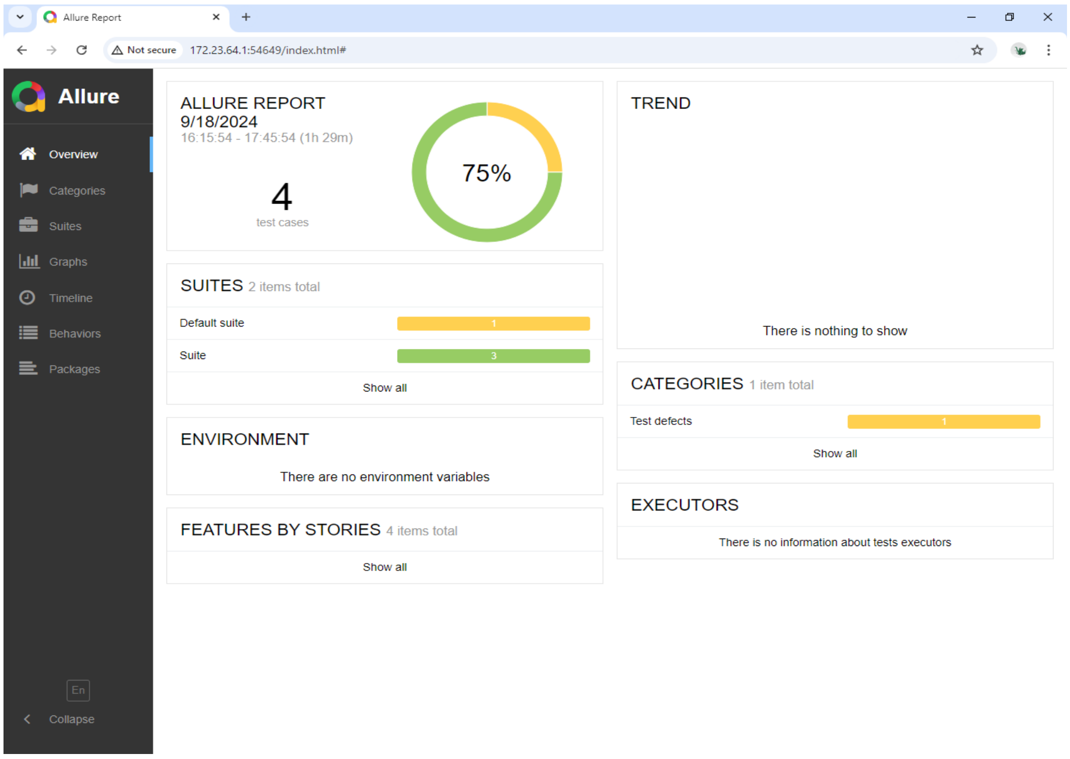
-The document and framework designed by Pavan Nanaware.

3. Reporting

We use **Allure Reports** for generating reports. Reports are generated in the allure report's directory after test execution. To use alure reports we need to install allure windows application. And the use following commands to view reports

Viewing Reports:

- Navigate to automation-framework/allure-reports in directory.
- Use command: allure serve D:\JS\Task\EntrataProd\allure-results
- These reports will be integrated with bug tracing tools like JIRA TFS or devOps.



ALLURE REPORT
9/18/2024
16:15:54 - 17:45:54 (1h 29m)

4
test cases

75%

TREND

There is nothing to show

SUITES 2 items total

Default suite	1
Suite	3

Show all

CATEGORIES 1 item total

Test defects	1

Show all

ENVIRONMENT

There are no environment variables

EXECUTORS

There is no information about tests executors

FEATURES BY STORIES 4 items total

Show all

4. Future Enhancements

- **CI/CD Integration**: We will Setup Jenkins for automated test execution on each commit.

Conclusion

This automation framework not only streamlines the testing process for the Entrata website but also ensures scalability and maintainability. By implementing best practices such as the Page Object Model and externalizing test data, the framework is equipped to adapt to future enhancements and changes in application requirements.

-

-The document and framework designed by Pavan Nanaware.