

**PROJECT OBJECT MODEL WITH PAGE FACTORY**



**By :** Samruddhi Date

**Batch code:**2023-112655

**Enrollment number:** EBEON0224881521

**INDEX:**

|  |
| --- |
| * Acknowledgement * Introduction * Advantages of POM * POM with PageFactory * Advantages of PageFactory * Implementation and Code * Conclusion |

**Acknowledgement:**

I would like to express my sincere gratitude to everyone who contributed to the successful completion of this project on **Design and implement a POM-based framework to improve code maintenance, readability and reusability in your selenium project**.

First and foremost, I would like to thank my supervisor, Mr. Santhosh Kumar, for his invaluable guidance, support, and encouragement throughout this project. His expertise and insights were instrumental in shaping the direction and quality of my work.

I am also deeply grateful to my colleagues and peers who provided helpful feedback and assistance during various stages of the project. Their collaboration and shared knowledge greatly enhanced the overall outcome.

A special thanks to Edu bridge for providing the necessary resources and environment to carry out this project. The access to tools and documentation was crucial for the research and development involved.

In this project I have implemented POM based framework to improve code maintenance, readability and reusability

This project includes implementation of automation of “[Swag Labs (saucedemo.com)](https://www.saucedemo.com/)” website in three ways:

1. Without using POM
2. With using POM
3. Using POM with Page factory

The purpose of the project is to demonstrate the difference between above three scenarios and to understand the practical implementation of benefits that are achieved by using POM framework and POM with Page factory

Lastly, I would like to thank my family and friends for their unwavering support and understanding during the course of this project. Their encouragement and belief in my abilities kept me motivated throughout.

Thank you all for your contributions and support.

Samruddhi Date

**INTRODUCTION:**

The **Project Object Model (POM)** in Selenium is a framework that allows you to organize your code and manage dependencies efficiently.

There are several ways to add a border to all pages of a document, depending on the software you are using. Here are a few common methods for popular programs:

In POM, the automation project is divided into logical modules, where each module represents a particular functionality of the application. Each module contains its own set of classes, methods, and dependencies.

The main components of POM in Selenium are:

1. Test cases: These are individual test scenarios that need to be executed.

2. Test data: This includes the input data required for executing the test cases.

3. Object repository: It contains the locators and properties of GUI elements of the application.

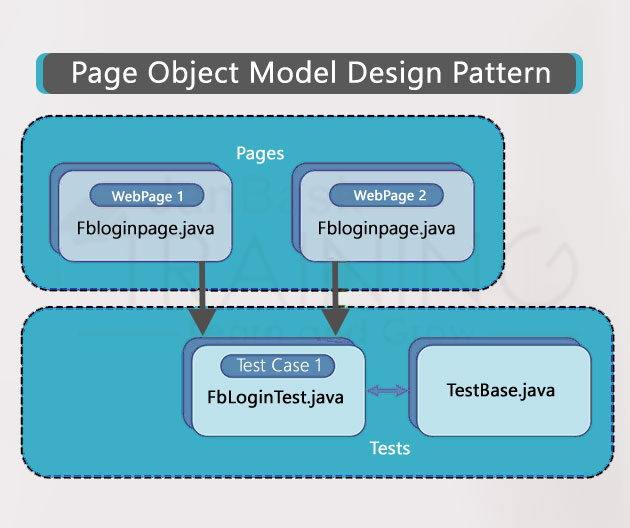
4. Utility files: These files contain common functions, such as reading data from an Excel sheet, capturing screenshots, etc.

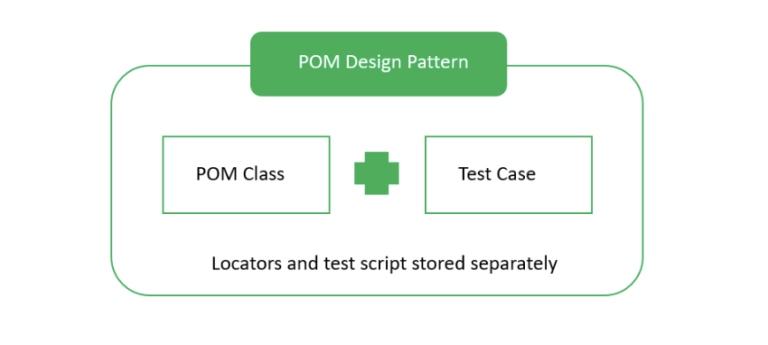
By using POM, you can easily maintain and update your automation code. It provides reusability and modularity, as the code is organized into different modules. Additionally, it simplifies code maintenance and debugging by separating the test logic from the test data and object repository.

Overall, the Project Object Model in Selenium helps in creating a structured and scalable automation framework, improving code readability, and enhancing efficiency and productivity in test automation projects.

The Page Factory design pattern is used in conjunction with the Page Object Model (POM) to implement a more maintainable and scalable automation framework. It helps in initializing the Web Elements in a Page Object Model class and improves the performance of the tests.

**Structure of POM:**





**Advantages Of POM:**

1. **Modular development**: The Project Object Model (POM) allows for the modular development of projects, where different components or modules can be independently developed, tested, and deployed. This promotes reusability and maintainability as each module can be focused on specific functionality or business logic.

2**. Dependency management**: POM enables efficient management of project dependencies. It provides a way to define and resolve dependencies on external libraries, frameworks, or modules. This ensures that all required dependencies are automatically downloaded and included in the project, resulting in reduced development effort and dependency conflicts.

3. **Standardization**: POM provides a standardized structure and format for project configuration and settings. This makes it easier for developers to understand and work with projects, as they can rely on a consistent structure and easily navigate through the project hierarchy.

4. **Build automation**: POM allows for the automation of various build tasks, such as compilation, packaging, testing, and deployment. This reduces human error, improves efficiency, and promotes continuous integration and delivery practices.

5. **Environment portability**: POM allows for the easy configuration and management of project settings across different environments. Developers can define specific configurations for development, testing, and production environments, ensuring consistent behavior and reducing the chances of environment-related issues.

6**. Integration with build tools**: POM seamlessly integrates with popular build tools like Apache Maven and Gradle. This enables developers to leverage the powerful features and functionalities provided by these tools, such as dependency management, project reporting, and release automation.

7. **Project reporting and documentation**: POM supports the generation of various reports and documentation, such as code coverage reports, test execution reports, and project documentation. This helps in monitoring the project's health, analyzing performance, and providing comprehensive project documentation for future reference.

8. **Collaboration and version control**: POM promotes collaboration among developers by providing a single source of truth for project configuration and settings. It allows multiple developers to work on the same project simultaneously, with version control systems ensuring proper synchronization and version management.

Overall, the Project Object Model offers numerous advantages, including modular development, dependency management, standardization, build automation, environment portability, integration with build tools, project reporting and documentation, and collaboration with version control. These advantages enhance productivity, maintainability, and the overall development experience.

**POM with Page factory:**

PageFactory is a way of implementing the “Page Object Model”. Here, we follow the principle of separation of Page Object Repository and Test Methods. It is an inbuilt concept of Page Object Model which is very optimized.

**To implement the Page Factory pattern in POM, you can follow these steps:**

**Step 1**: Create a Page Object Model class representing a web page

**Step 2**: Import the necessary packages:

**Step 3**: Initialize WebElements within the constructor using the `@FindBy` annotation:

**Step 4**: Use the initialized WebElements in your test methods:

**Advantages of using POM with page factory:**

1. Code reusability: With Page Factory, you can create reusable elements and methods. This helps in reducing code duplication and makes the code more maintainable.

2. Encapsulation: Page Factory allows you to encapsulate the elements and methods within a page object. This makes it easier to organize and manage the code.

3. Easy maintenance: Since the elements and methods are encapsulated within a page object, any changes in the UI can be easily updated in the respective page object class. This reduces the effort required for maintenance.

4. Improved readability: Page Factory makes use of annotations such as @FindBy to identify the elements on a page. This makes the code more readable and understandable.

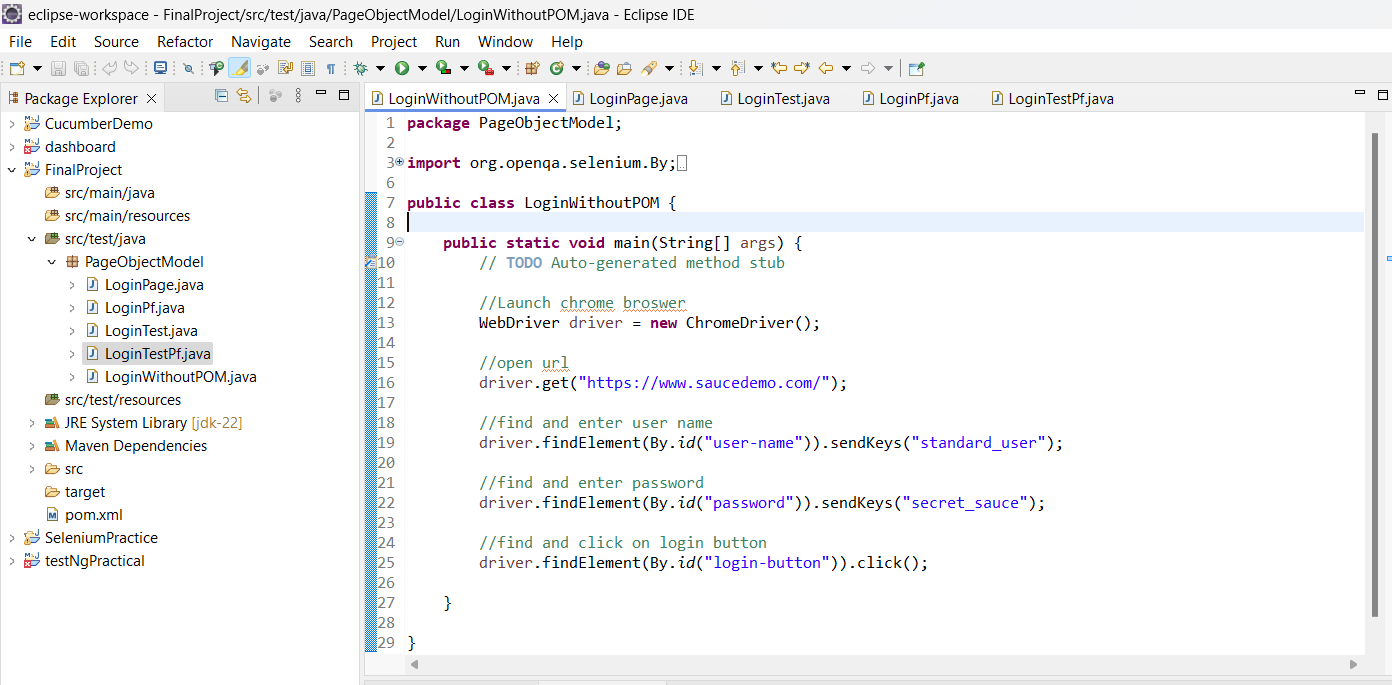
5. Efficient test execution: With Page Factory, the elements are initialized lazily i.e. they are initialized only when they are accessed. This helps in improving the performance and efficiency of test execution.

6. Enhanced debugging: Page Factory provides better debugging capabilities as it provides clear errors and exceptions when a page element is not found or accessed incorrectly.

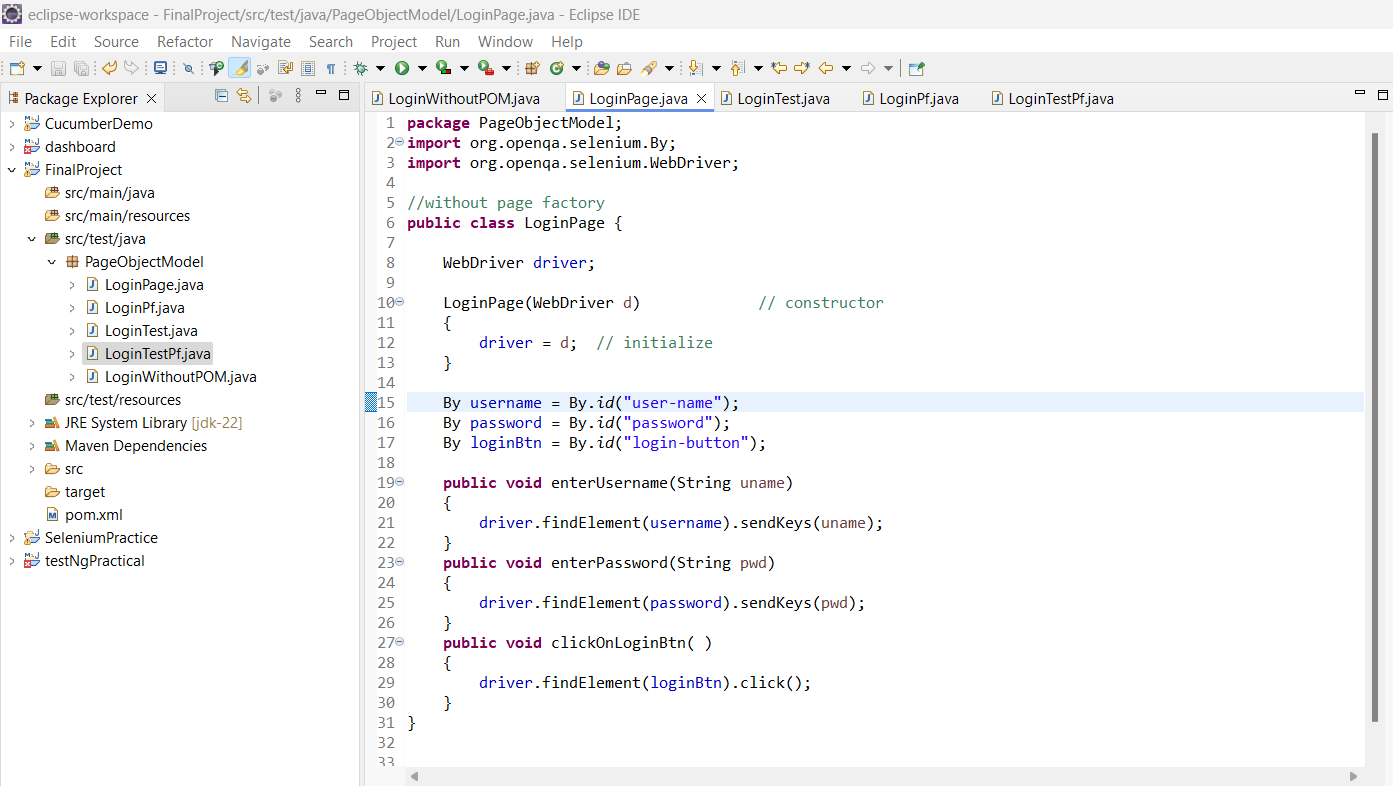
7. Cross-browser compatibility: Page Factory makes it easier to handle cross-browser compatibility by providing a single location to manage the page elements.

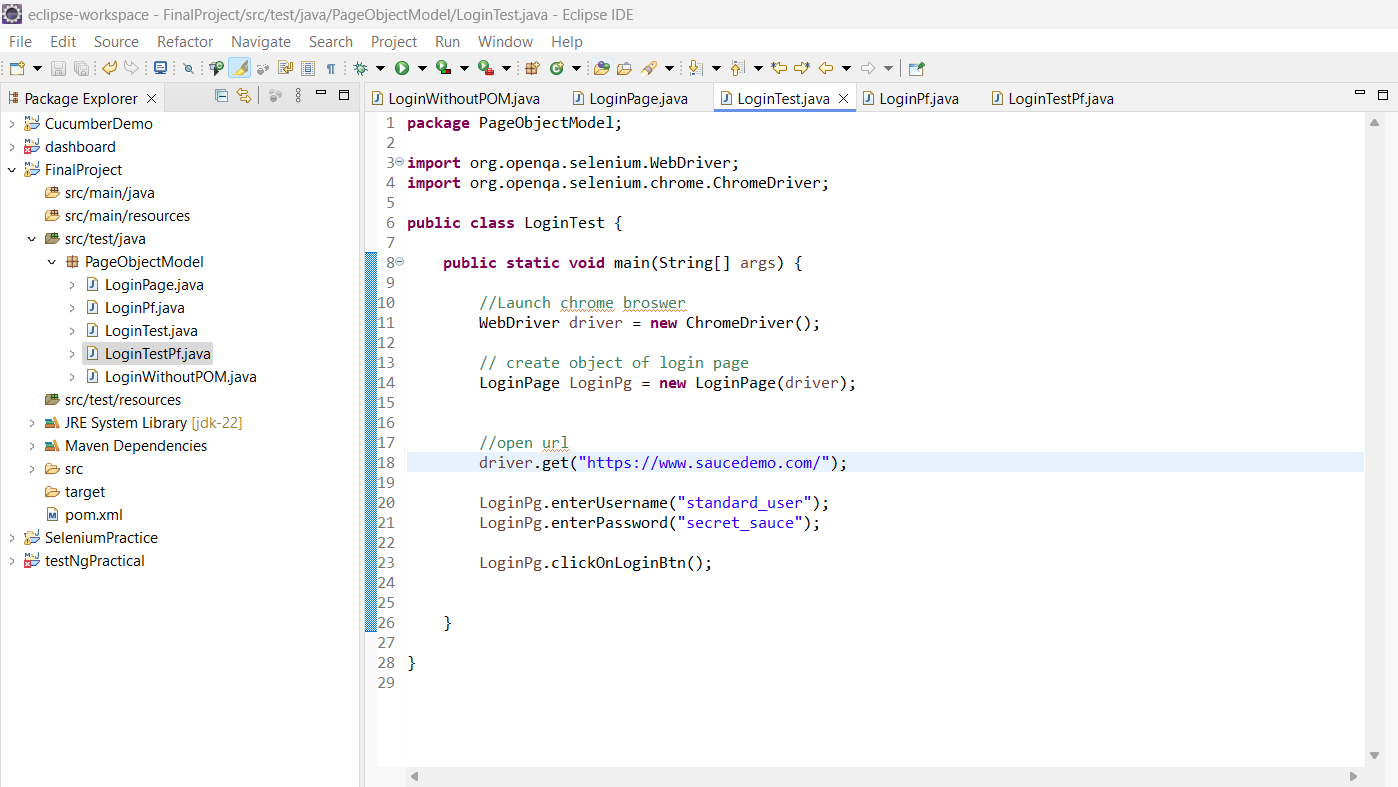
**Code and Implementation:**

* **Without using POM:**

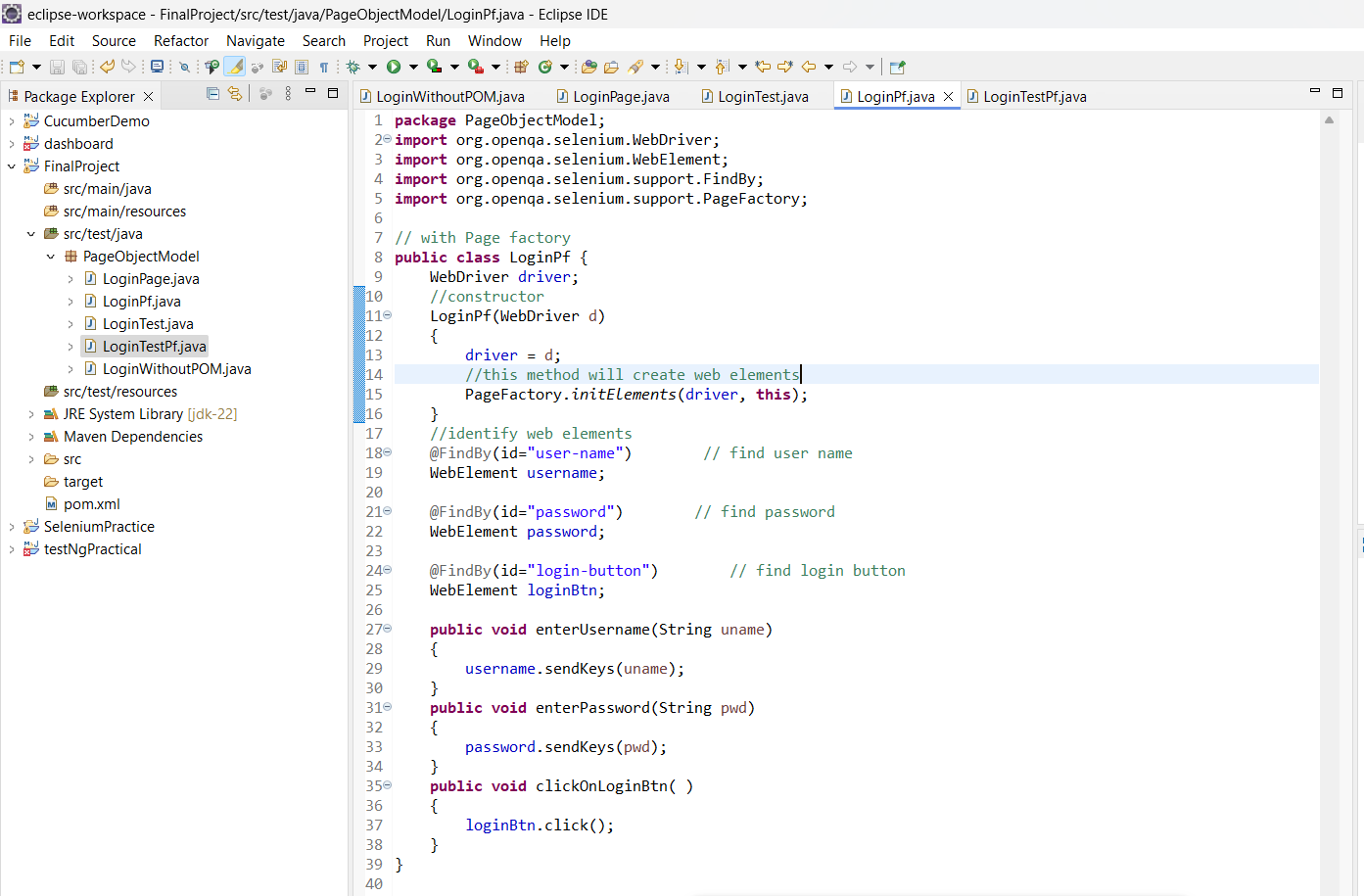
****

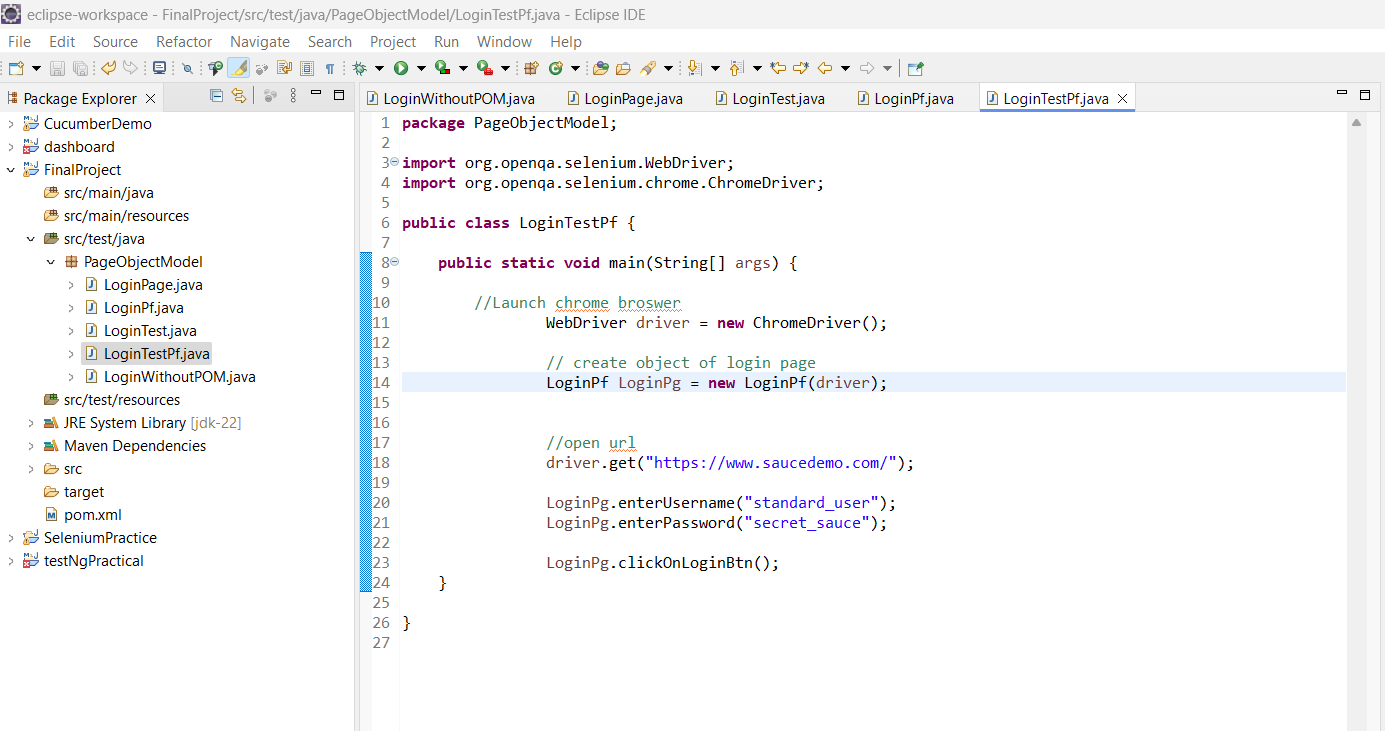
* **Using POM:**

****

****

* **Using POM with Page Factory:**

****

****

**Conclusion:**

By using POM, you can easily maintain and update your automation code. It provides reusability and modularity, as the code is organized into different modules. Additionally, it simplifies code maintenance and debugging by separating the test logic from the test data and object repository.

Overall, the Project Object Model in Selenium helps in creating a structured and scalable automation framework, improving code readability, and enhancing efficiency and productivity in test automation projects.

The Page Factory design pattern is used in conjunction with the Page Object Model (POM) to implement a more maintainable and scalable automation framework. It helps in initializing the Web Elements in a Page Object Model class and improves the performance of the tests.