Automated E-commerce Testing Framework

This project provides an automated testing framework for an e-commerce website (Amazon India) using Selenium WebDriver, Java, Maven, and TestNG. It implements the Page Object Model (POM) design pattern for maintainability and reusability, and includes features for data-driven testing using Excel, comprehensive reporting with ExtentReports, and logging with Lombok.

Project Structure

```
 selenium-ecommerce-test/
├── src/
│   ├── main/
│   │   ├── java/
│   │   │   └── com/
│   │   │       └── ecommerce/
│   │   │           └── test/
│   │   │               ├── pages/             # Page Object Model classes
│   │   │               └── utilities/         # Helper classes
(WebDriverFactory, ExcelUtility, CsvUtility, ExtentReporterNG, Listeners,
ConfigReader)
│   │   └── resources/       # Configuration files (config.properties)
│   └── test/
│       ├── java/
│       │   └── com/
│       │       └── ecommerce/
│       │           └── test/
│       │               └── tests/           # Test classes (BaseTest,
ProductSearchTest, ProductPageValidationTest, SearchFunctionalityTest)
│       └── resources/     # TestNG XML and test data (testng.xml,
testdata.xlsx)
├── reports/                # Generated test reports and CSV files
├── pom.xml                 # Maven Project Object Model file
└── todo.md                 # Project To-Do list
```

Prerequisites

- Java Development Kit (JDK) 8 or higher
- Maven 3.6.0 or higher

- Web browser (Chrome or Firefox)
- Internet connection

Setup and Installation

1. **Clone the repository (if applicable):**

   `bash git clone <repository_url> cd selenium-ecommerce-test`

2. **Build the project using Maven:**

   `bash mvn clean install`

   This command will download all necessary dependencies and compile the project.

Running Tests

Tests can be executed using TestNG via Maven.

Running all tests

To run all tests defined in `testng.xml`:

```
mvn test
```

Running tests with specific browser (e.g., Firefox)

You can override the browser specified in `config.properties` by passing it as a Maven parameter:

```
mvn test -Dbrowser=firefox
```

Test Data Management

Test data is managed using an Excel file (`src/test/resources/testdata.xlsx`). The `ExcelUtility` class reads data from this file, allowing for easy modification of test inputs without changing the code.

Reporting

Test results are generated using ExtentReports. After test execution, an HTML report will be available at `reports/index.html`. This report provides a detailed overview of test passes, failures, and skips.

Extracted product information from crawling is saved to `reports/product_details.csv`.

Logging

Lombok's `@Slf4j` annotation is used for logging. Logs are integrated with ExtentReports listeners.

Features Implemented

- **Basic Crawling:**
  - Automated opening of the homepage.
  - Search for a product (e.g., "laptop").
  - Extraction of Product Name, Price, Ratings, and URL from search results.
- **Functional Testing:**
  - Validation of "Add to Cart" button presence.
  - Validation of Product details section.
  - Validation of Image gallery functionality.
  - Testing search functionality with valid and invalid inputs.
- **Reporting:**

    - Extracted product information stored in a CSV file.
    - Test results logged and reported using ExtentReports.
- **Bonus Tasks:**
    - **Multi-page Crawling:** The `SearchResultsPage` class is enhanced to crawl multiple pages of search results.

Customization

- **Browser:** Default browser is Chrome. You can change it in `config.properties` or via Maven command line.
- **Base URL:** The base URL for the e-commerce website can be configured in `config.properties`.
- **Test Data:** Modify `src/test/resources/testdata.xlsx` to add or change test products and search types.
- **Test Scenarios:** Add new test methods in existing test classes or create new test classes under `src/test/java/com/ecommerce/test/tests/`.
- **Page Objects:** Update existing page object classes or create new ones under `src/main/java/com/ecommerce/test/pages/` as the website UI evolves.