

amazon-bdd-framework/

- |— pom.xml # Maven dependencies and build config
- |— testng.xml # TestNG suite configuration
- |— src/
 - | |— main/
 - | | |— java/
 - | | | |— com.amazon.framework/
 - | | | | |— base/
 - | | | | | |— DriverFactory.java # Manages WebDriver setup/teardown
 - | | | | | |— BaseTest.java # Base test class for common setup
 - | | | | |— pages/
 - | | | | | |— LoginPage.java # Page Object Model (POM) for login page
 - | | | | | |— HomePage.java # POM for home/search page (product selection)
 - | | | | | |— ProductPage.java # POM for product details & add to cart
 - | | | | | |— CartPage.java # POM for cart page
 - | | | | | |— PaymentPage.java # POM for payment page
 - | | | | | |— LogoutPage.java # POM for logout functionality
 - | | | | |— utils/
 - | | | | | |— ConfigReader.java # Reads config properties (browser, url)
 - | | | | | |— WaitUtils.java # Helper methods for waits (explicit wait etc.)
 - | | | | |— config/
 - | | | | | |— config.properties # Configuration file (URLs, credentials)
 - | |— test/
 - | | |— java/
 - | | | |— com.amazon.tests/
 - | | | | |— stepdefinitions/
 - | | | | | |— LoginSteps.java # Step definitions for login.feature
 - | | | | | |— ProductSteps.java # Step definitions for product selection
 - | | | | | |— CartSteps.java # Step definitions for cart actions
 - | | | | | |— PaymentSteps.java # Step definitions for payment page
 - | | | | | |— LogoutSteps.java # Step definitions for logout steps
 - | | | | |— runners/
 - | | | | | |— TestRunner.java # Cucumber runner integrating with TestNG
 - | | |— resources/
 - | | | |— features/
 - | | | | |— login.feature # Gherkin feature file for login
 - | | | | |— product.feature # Feature file for product selection
 - | | | | |— cart.feature # Feature file for cart
 - | | | | |— payment.feature # Feature file for payment
 - | | | | |— logout.feature # Feature file for logout
 - | | | |— config/
 - | | | | |— testdata.properties # Test data or environment configs
- |— target/
 - | |— cucumber-reports/ # Generated test reports

1. pom.xml

- This file manages your project's dependencies, like Selenium, Cucumber, TestNG, etc.
 - It also controls how your project is built and run with Maven (a build tool).
-

2. testng.xml

- Configuration file for TestNG.
 - It tells TestNG which tests to run, and how (parallel execution, groups, etc.).
 - Used to run Cucumber tests with TestNG integration.
-

3.1 src/main/java/com.amazon.framework/

Contains your **main application code** (automation framework classes).

3.1.1 base/

- Contains setup classes that help initialize and manage browser drivers.
 - Example files:
 - **DriverFactory.java** — Creates and configures browser drivers like ChromeDriver or FirefoxDriver.
 - **BaseTest.java** — Parent test class that opens and closes browsers before/after tests.
-

3.1.2 pages/

- Contains Page Object Model (POM) classes.
 - Each class represents one page on Amazon's website and contains methods to interact with page elements.
 - Example:
 - **LoginPage.java** — Handles login page elements and actions (enter username, password, click login).
 - **HomePage.java** — Handles search and product selection features.
 - **ProductPage.java** — Handles adding products to cart.
 - **CartPage.java** — Manages cart operations.
 - **PaymentPage.java** — Manages payment actions.
 - **LogoutPage.java** — Handles logout operation.
-

3.1.3 utils/

- Utility/helper classes to support your framework.
 - Example:
 - **ConfigReader.java** — Reads configuration like browser name, URLs from properties files.
 - **WaitUtils.java** — Contains methods for waits (wait for element to be visible, clickable, etc.) to handle page loading delays.
-

3.1.4 config/

- Contains configuration files like **config.properties** which store values such as:
 - User credentials (can be stored here or in separate files)
-

3.2 src/test/java/com.amazon.tests/

Contains your **test-related code**.

3.2.1 stepdefinitions/

- Classes that “glue” your Gherkin feature steps to actual Java code.
 - Each method corresponds to a step in your .feature files.
 - Example files:
 - **LoginSteps.java** — Implements login steps like entering username, password.
 - **ProductSteps.java** — Implements steps for searching/selecting products.
 - **CartSteps.java** — Implements steps for adding/removing products in cart.
 - **PaymentSteps.java** — Implements payment processing steps.
 - **LogoutSteps.java** — Implements logout steps.
-

3.2.2 runners/

- Contains test runner classes to start your Cucumber tests.
 - **TestRunner.java** integrates Cucumber with TestNG and specifies where feature files and step definitions are located.
-

3.3 src/test/resources/

Contains resource files needed for tests.

3.3.1 features/

- Contains your **Gherkin .feature files** which describe test scenarios in simple English.
 - Example:
 - **login.feature** — Contains login scenarios.
 - **product.feature** — Contains product selection scenarios.
 - **cart.feature, payment.feature, logout.feature** — Similarly for other scenarios.
-

3.3.2 config/

- Stores test-specific config or test data files.
 - For example:
 - **testdata.properties** — Stores test data like user details, product names.
-

4. target/

- Automatically created by Maven after tests run.
- Contains generated reports, compiled code, and logs.
- **cucumber-reports/** folder inside contains reports of test executions with detailed pass/fail info.

Summary for Beginners

Folder/File	What it Does
pom.xml	Manage dependencies like Selenium, Cucumber.
testng.xml	Configure and run tests with TestNG.
base/	Browser driver setup and test initialization.
pages/	Represent website pages (POM classes).
utils/	Helper classes for config and waits.
config/	Config files for URLs, browser, credentials.
stepdefinitions/	Maps Gherkin steps to Java code.
runners/	Starts the Cucumber tests with TestNG.
features/	Test scenarios written in plain English.
testdata.properties	Test data like usernames, passwords, etc.
target/	Contains reports and compiled files after run.