Selenium Frameworks Overview

1. Data-Driven Framework

Definition: Test data is stored separately (Excel, CSV, DB) and scripts run with multiple data sets.

Purpose: Reuse same test with different data.

Example: Login tests using usernames and passwords from Excel.

Tools: Apache POI, CSV Reader.

2. Keyword-Driven Framework

Definition: Uses keywords like Click, EnterText, VerifyText stored in files; mapped to code actions.

Purpose: Allows non-programmers to create tests.

Example: Excel with steps like OpenBrowser, NavigateURL, EnterText.

Tools: Apache POI + custom keywords.

3. Hybrid Framework (TestNG)

Definition: Combines Data-Driven + Keyword-Driven with TestNG for execution and reporting.

Purpose: Leverage strengths of multiple frameworks.

Example: Excel data + keywords + TestNG annotations.

Features: @BeforeTest, @AfterTest, DataProviders, Parallel execution, HTML reports.

4. BDD (Behavior-Driven Development)

Definition: Uses Given-When-Then in natural language for collaboration.

Purpose: Bridge gap between business and QA.

Example:

Given user is on login page

When user enters valid username and password

Then user sees the dashboard.

Tools: Cucumber, JBehave.

5. Modular Testing Framework

Definition: Application divided into modules; each module tested separately.

Purpose: Easier maintenance.

Example: Separate modules for Login, Search, Checkout.

6. POM (Page Object Model)

Definition: Each page is a class with locators and actions.

Purpose: Reduce duplication, improve readability.

Example: LoginPage class with login() method.

7. Hybrid + BDD + POM

Definition: Combines Hybrid, BDD, and POM for high maintainability.

Purpose: Reusable, readable, collaborative framework.

Workflow:

- Feature files in Gherkin

- Step definitions map to Selenium actions
- POM classes for UI elements
- Data from Excel/CSV
- Keywords for reusable actions