**What is BDD Framework?**

**BDD (Behavior Driven Development)** is a **software development approach** that enhances collaboration between developers, testers, and business stakeholders.

It extends **Test-Driven Development (TDD)** by focusing on the **behavior of an application** for an end user, instead of the technical implementation.

BDD ensures:

* Everyone (technical & non-technical) understands what the system should do.
* Requirements are written in a **common language** (like English), so even non-programmers can understand them.

**What is behave?**

* behave is behaviour-driven development, Python style.
* behave operates on directories containing:
  + **feature files** written by your Business Analyst / Sponsor / whoever with your behaviour scenarios in it, and
  + a **steps** directory with Python step implementations for the scenarios.

**Example: OrangeHRM Login Feature**

**Feature:** OrangeHRM Login

1. **Scenario: Logo presence on OrangeHRM Home Page**
   * **Given** I launch the Chrome browser
   * **When** I open the OrangeHRM homepage
   * **Then** I verify that the logo is present on the page

👉 In short:

* **Feature file** = describes what the system should do.
* **Scenario** = one example of system behavior.
* **Given/When/Then** = steps that explain conditions, actions, and outcomes.

bdd\_framework/

│

├── features/ # Gherkin scenarios

│ ├── login.feature # Example UI feature

│ ├── api.feature # Example API feature

│ ├── environment.py # Hooks (setup/teardown)

│ └── steps/ # Step definitions

│ ├── login\_steps.py

│ ├── api\_steps.py

│ └── common\_steps.py

│

├── pages/ # Page Object Model (for Selenium UI tests)

│ ├── base\_page.py

│ └── login\_page.py

│

├── utils/ # Utilities & helpers

│ ├── config.py # Configuration (URLs, credentials, etc.)

│ ├── logger.py # Logging setup

│ └── api\_client.py # API wrapper using requests

│

├── reports/ # Test reports (JUnit/Allure/HTML)

│

├── requirements.txt # Dependencies

└── behave.ini # Behave configuration

Commit and run the this command

git commit -m "Updated feature files"

git push -u origin main

python -m behave Features/LoginApp.feature

python -m behave -f allure\_behave.formatter:AllureFormatter -o reports/ Features