1.what is .feature files

.feature file is used in **Behavior-Driven Development (BDD)** frameworks like **Cucumber**, **SpecFlow**, or **Behave** (Python). It defines application behavior **from the user's perspective** in **plain English**, using a language called **Gherkin**.

**Key Concepts of .feature Files:**

* **File Extension:** .feature
* **Language:** Gherkin (a structured, readable format)
* **Purpose:** Describe **scenarios** that illustrate software requirements (like tests, but readable by non-programmers)
* **Basic Structure:**
* gherkin
* Feature: User login
* Scenario: Successful login
* Given the user is on the login page
* When the user enters valid credentials
* Then they should be redirected to the dashboard
* **Gherkin Keywords:**

| **Keyword** | **Description** |
| --- | --- |
| Feature | A high-level description of a feature |
| Scenario | A specific example of how it works |
| Given | Initial context or precondition |
| When | The action the user takes |
| Then | The expected outcome |
| And, But | Help chain multiple steps |

* **🔹 Example Use Case:**
* Let’s say you're building a login system. You could write:
* gherkin
* CopyEdit
* Feature: Login functionality
* Scenario: Invalid login
* Given the user is on the login page
* When the user enters an incorrect password
* Then an error message should be displayed
* This file can then be **connected to actual test code** (step definitions) in Java, Python, JavaScript, etc.
* **Components of a .feature File:**
* Here's a breakdown of the **main components**:

| * **Component** | * **Purpose** |
| --- | --- |
| * Feature: | * Describes the functionality being tested |
| * Scenario: | * A single test case/example |
| * Scenario Outline: | * A scenario template to run with multiple data sets |
| * Background: | * Reusable setup that runs before each scenario |
| * Examples: | * Table of test data (used with Scenario Outline) |
| * Steps: | * Given, When, Then, And, But — define test flow |

* **✅ Example .feature File:**
* gherkin
* CopyEdit
* Feature: Login functionality
* Background:
* Given the user is on the login page
* Scenario: Successful login
* When the user enters username "john" and password "12345"
* Then the user should see the dashboard
* Scenario: Failed login
* When the user enters username "john" and password "wrong"
* Then an error message should be displayed
* Scenario Outline: Login with multiple users
* When the user enters username "<username>" and password "<password>"
* Then the user should see "<result>"
* Examples:
* | username | password | result |
* | alice | pass123 | dashboard |
* | bob | wrong | error message |
* **🔑 Gherkin Step Keywords:**

| * **Keyword** | * **Meaning** |
| --- | --- |
| * Given | * The starting state (precondition) |
| * When | * The action or event |
| * Then | * The expected outcome |
| * And/But | * Additional steps |

* **🧪 How It's Used:**
* .feature files are **not code** — they describe what should happen.
* Each line in the .feature file is connected to a **step definition** in code (e.g., Python, Java, JS).
* The automation tool runs the .feature file like a test script.