

Chapter: Writing Your First Feature Files (1.5 hours)

Objective

By the end of this session, you will be able to:

- Understand how to write **.feature** files using Gherkin syntax
 - Translate real-world use-cases into BDD scenarios
 - Draft clear and testable BDD scenarios for:
 - User Login
 - Task Creation
 - Error Scenarios
 - Collaborate with peers or product stakeholders for review
-

Theoretical Concepts

1. What is a Feature File?

A **Feature file** is a plain-text file written in **Gherkin syntax**, containing one or more **Scenarios** that describe the behavior of a system.

It bridges the gap between business stakeholders and developers by using **natural language**.

- File extension: **.feature**
- Location: Typically placed in a **features/** directory

```
Feature: User Login
  Scenario: Successful login
    Given the user is on the login page
    When the user enters valid credentials
    Then the user should be redirected to the dashboard
```

Real-World Examples

Let's walk through three types of scenarios using real-world examples.

A. User Login

Successful Login

```
Feature: User Authentication
  Scenario: Successful login
```

Given the user is on the login page
When the user enters username "john" and password "password123"
Then the user should be redirected to the dashboard

✖ Failed Login

Scenario: Invalid password
Given the user is on the login page
When the user enters username "john" and password "wrongpass"
Then an error message "Invalid credentials" should be displayed

🔗 B. Task Creation

Feature: Task Management

Scenario: Creating a new task
Given the user is logged in
When the user adds a task with title "Buy groceries"
Then the task should appear in the task list

🔗 C. Error Scenario (Empty Input)

Scenario: Task creation fails with empty title
Given the user is logged in
When the user adds a task with an empty title
Then an error message "Task title cannot be empty" should be displayed

🤝 Peer Review Checklist

- ☒ Are the scenarios readable by non-developers?
- ☒ Do they cover both positive and negative paths?
- ☒ Is each scenario **independent** and **clear**?
- ☒ Do the steps avoid implementation detail?
- ☒ Are reusable steps considered (e.g., "Given the user is logged in")?

💬 Common Interview Questions

1. What is a **.feature** file in BDD?
2. How do you ensure your Gherkin scenarios are testable and maintainable?

3. Can you describe the difference between **Scenario** and **Scenario Outline**?
 4. What's the advantage of using Given/When/Then format?
 5. How do you handle shared steps across multiple features?
-

Curated YouTube Video Links

1. [Gherkin and Cucumber Tutorial for Beginners \(Automation Step by Step\)](#)
 2. [Writing Feature Files in Gherkin - ToolsQA](#)
 3. [Cucumber BDD for Beginners \(from scratch\)](#)
 4. [What is Gherkin? \(Cucumber Series\)](#)
-

Assignment

Write **.feature** files for the following:

1. **User registration**
2. **Task deletion**
3. **Password reset with invalid token**

Peer-review with:

- A colleague
 - A developer on your team
 - Or ChatGPT! Paste your **.feature** file for feedback
-

☒ Summary

- Feature files serve as living documentation and automation instructions
 - Gherkin uses Given/When/Then syntax to describe expected system behavior
 - Writing clear, real-world BDD scenarios helps bridge communication between devs and stakeholders
 - Practicing both happy and error paths improves test coverage and reliability
-