

Task 1: Write a Simple Cucumber Feature for a Login Functionality

Topic: Cucumber with Selenium - Crafting Feature Files

Description: Create a basic Cucumber feature file for a login functionality, including scenarios for a successful login and a failed login attempt due to incorrect credentials.

Steps:

Open your IDE and create a new feature file named login.feature in your Cucumber project.

Define a Feature: User Login at the top of the file.

Write a Scenario: Successful Login with Gherkin steps that include Given, When, Then, such as:

Given the user is on the login page

When the user enters valid credentials

Then the user is redirected to the homepage

Write a Scenario: Unsuccessful Login with steps for attempting to log in with invalid credentials and asserting the error message.

```
package Day6;
```

```
import io.cucumber.java.en.*;
```

```
import org.openqa.selenium.By;
```

```
import org.openqa.selenium.WebDriver;
```

```
import org.openqa.selenium.WebElement;
```

```
import org.openqa.selenium.chrome.ChromeDriver;
```

```
import static org.junit.Assert.*;
```

```
public class Login {
```

```
    WebDriver driver;
```

```
    @Given("the user is on the login page")
```

```
    public void the_user_is_on_the_login_page() {
```

```
        System.setProperty("webdriver.chrome.driver", "D:/Wipro/Project/driver/chromedriver-win64/chromedriver.exe ");
```

```
        driver = new ChromeDriver();
```

```
        driver.get("https://jpetstore.aspectran.com/account/signonForm ");
```

```
    }
```

```
    @When("the user enters a valid username and password")
```

```

public void the_user_enters_a_valid_username_and_password() {

    // Enter valid username and password

    WebElement usernameField = driver.findElement(By.id("name"));

    WebElement passwordField = driver.findElement(By.id("p"));

    usernameField.sendKeys("jpetstore");

    passwordField.sendKeys("jpet@123");

    driver.findElement(By.xpath("//*[@id="Signon"]/form/div/div/button ")).click();

}

```

```

@Then("the user is redirected to the homepage")

public void the_user_is_redirected_to_the_homepage() {

    // Assert that the user is redirected to the homepage

    String expectedUrl = " https://jpetstore.aspectran.com/ ";

    assertEquals(expectedUrl, driver.getCurrentUrl());

    driver.quit();

}

```

```

@When("the user enters an invalid username and password")

public void the_user_enters_an_invalid_username_and_password() {

    // Enter invalid username and password

    WebElement usernameField = driver.findElement(By.id("username"));

    WebElement passwordField = driver.findElement(By.id("password"));

    usernameField.sendKeys("invalid");

    passwordField.sendKeys("Password");

    driver.findElement(By.xpath("//*[@id="Signon"]/form/div/div/button ")).click();  }

```

```

@Then("an error message is displayed")

public void an_error_message_is_displayed() {

    // Assert that an error message is displayed

    WebElement errorMessage = driver.findElement(By.id("Invalid username or password. Signon failed."));

    assertTrue(errorMessage.isDisplayed());

    driver.quit();

}

}

```

Login.feature File:

Feature: User Login

Scenario: Successful login with valid credentials

Given the user is on the login page

When the user enters a valid username and password

Then the user is redirected to the homepage

Scenario: Unsuccessful login with invalid credentials

Given the user is on the login page

When the user enters an invalid username and password

Then an error message is displayed

Task 2: Perform a Simple GET Request Using Postman

Topic: Postman for API Testing - Making API Calls

Description: Use Postman to send a GET request to a public API and validate the response.

Steps:

Open Postman.

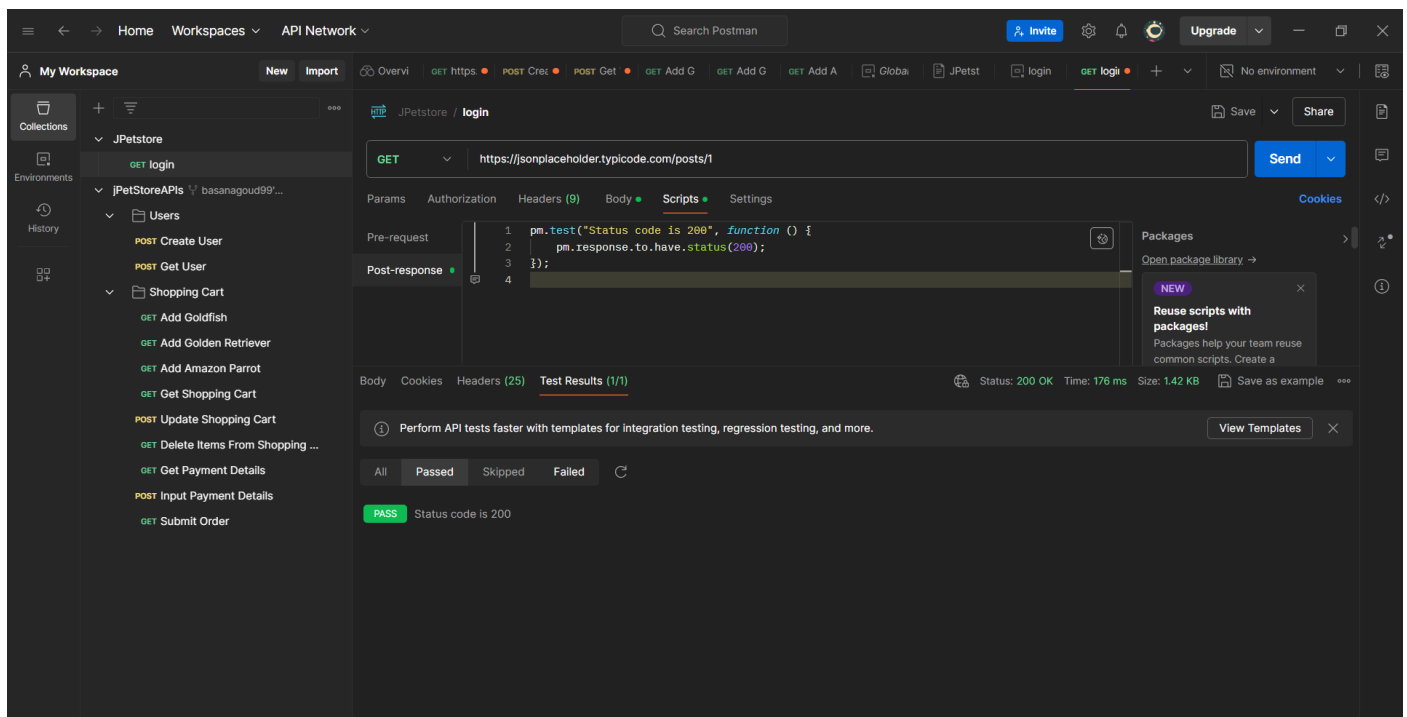
Create a new request by selecting the HTTP method GET.

Enter the URL of a public API endpoint that returns JSON data (e.g., <https://jsonplaceholder.typicode.com/posts/1>).

Send the request and observe the response.

Check the status code is 200 OK and use the Tests tab in Postman to write a simple test confirming the status code, such as:

```
pm.test("Status code is 200", function () { pm.response.to.have.status(200); });
```



Status pass