**Test Case Document for home.php**

**1. Introduction**

This document contains test cases for the home.php file, focusing on functional, UI, and security testing. The execution results will indicate the pass/fail status.

**2. Test Cases**

| **Test Case ID** | **Test Scenario** | **Steps to Execute** | **Expected Result** | **Actual Result** | **Status** |
| --- | --- | --- | --- | --- | --- |
| TC-01 | Page Load | Open home.php in a browser | The home page should load without errors |  |  |
| TC-02 | Navigation Links | Click on different navigation links (e.g., Categories, Cart) | Navigation should work as expected |  |  |
| TC-03 | Book Display | Check if books are displayed with correct titles, prices, and images | Books should be listed correctly with accurate details |  |  |
| TC-04 | Search Functionality | Enter a book title in the search bar and click search | Books related to the search query should be displayed |  |  |
| TC-05 | Add Book to Cart | Click on "Add to Cart" for a book | Book should be added to the shopping cart |  |  |
| TC-06 | UI Testing | Check alignment, font, and colors of the page | UI should be visually correct |  |  |
| TC-07 | Mobile Responsiveness | Open the homepage on mobile view | The homepage should be responsive |  |  |
| TC-08 | Login Prompt | Click on the "Login" button on the home page | User should be redirected to the login page |  |  |
| TC-09 | Security Testing | Try SQL Injection (' OR '1'='1) in the search bar | Input should be sanitized |  |  |
| TC-10 | Automation Using Selenium | Run an automated test script for book browsing | Books should display and cart should be functional |  |  |

**3. Automated Testing Approach**

* **Tool**: Selenium WebDriver
* **Script**: Automates book search and adds books to the cart
* **Verification**: Verifies the display of books and successful addition to the cart.

**Test Case Document for login.php**

**1. Introduction**

This document contains test cases for the login.php file, focusing on functional, UI, and security testing. The execution results will indicate the pass/fail status.

**2. Test Cases**

| **Test Case ID** | **Test Scenario** | **Steps to Execute** | **Expected Result** | **Actual Result** | **Status** |
| --- | --- | --- | --- | --- | --- |
| TC-01 | Page Load | Open login.php in a browser | The login page should load without errors |  |  |
| TC-02 | Required Fields | Leave the email and password fields empty and submit | Error message should appear for empty fields |  |  |
| TC-03 | Invalid Credentials | Enter incorrect email and password, then submit | Error message should be displayed |  |  |
| TC-04 | Valid Credentials | Enter valid email and password, then submit | User should be redirected to the homepage |  |  |
| TC-05 | Password Masking | Enter a password in the password field | The password should be masked with dots |  |  |
| TC-06 | UI Testing | Check alignment, font, and colors of the login page | UI should be visually correct |  |  |
| TC-07 | Forgot Password Link | Click on the "Forgot Password" link | User should be redirected to the reset password page |  |  |
| TC-08 | Security Testing | Try SQL Injection (' OR '1'='1) in the email or password field | Input should be sanitized |  |  |
| TC-09 | Mobile Responsiveness | Open the login page on mobile view | The login page should be responsive |  |  |
| TC-10 | Automation Using Selenium | Run an automated test script for successful login | User should be able to log in and be redirected to the homepage |  |  |

**3. Automated Testing Approach**

* **Tool**: Selenium WebDriver
* **Script**: Automates the login process for valid and invalid credentials
* **Verification**: Verifies successful login and error handling for invalid login attempts.

**Selenium Test Scripts**

**Test Case 1: Valid Login**

python

Copy

from selenium import webdriver

from selenium.webdriver.common.keys import Keys

# Set up WebDriver

driver = webdriver.Chrome()

driver.get("http://localhost/login.php")

# Fill in the login form

driver.find\_element("name", "email").send\_keys("user@example.com")

driver.find\_element("name", "password").send\_keys("validpassword")

driver.find\_element("name", "submit").click()

# Verify successful login (redirect to homepage)

assert "Welcome" in driver.page\_source

driver.quit()

**Test Case 2: Invalid Login Handling**

python

Copy

from selenium import webdriver

from selenium.webdriver.common.keys import Keys

# Set up WebDriver

driver = webdriver.Chrome()

driver.get("http://localhost/login.php")

# Enter invalid login details

driver.find\_element("name", "email").send\_keys("user@example.com")

driver.find\_element("name", "password").send\_keys("invalidpassword")

driver.find\_element("name", "submit").click()

# Verify error message

assert "Invalid credentials" in driver.page\_source

driver.quit()