

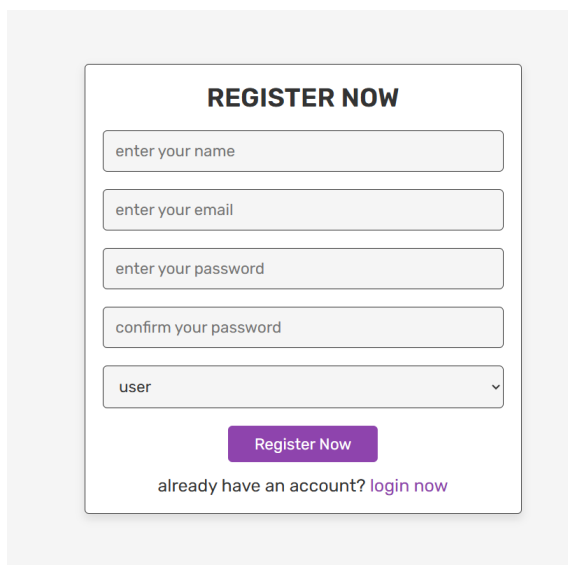
---

### Test Cases

---

#### 1. User Registration

Test Case ID	TC01
Title	User Registration Functionality
Precondition	The registration page is accessible.
Test Steps	<ol style="list-style-type: none"><li>1. Navigate to the registration page.</li><li>2. Enter valid user details (e.g., name, email, password).</li><li>3. Click the “Register” button.</li></ol>
Expected Result	User is registered successfully, and a confirmation message is displayed.
Actual Result	Registered successfully but Confirmation message is not displayed.
Status	Fail
Priority	High



**REGISTER NOW**

enter your name

enter your email

enter your password

confirm your password

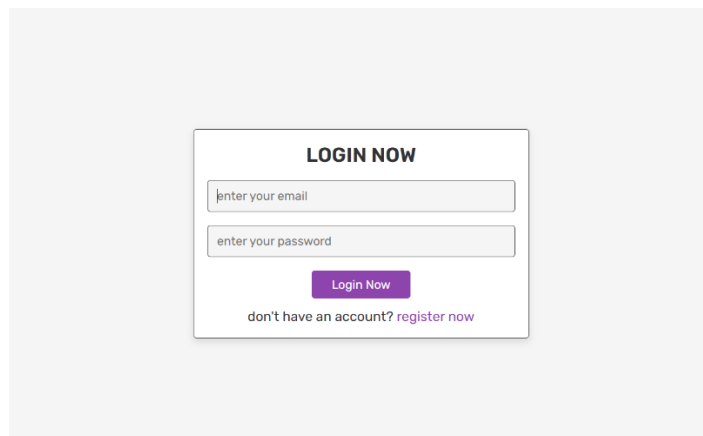
user

**Register Now**

already have an account? [login now](#)

## 2. User Login Functionality

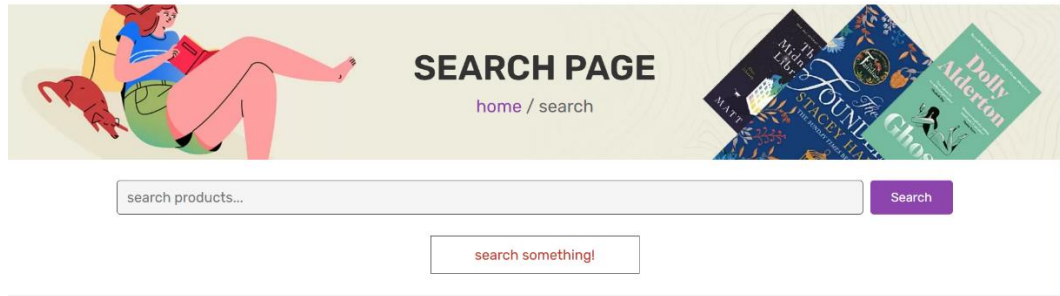
Test Case ID	TC02
Title	User Login Functionality
Precondition	User is registered and has valid login credentials.
Test Steps	<ol style="list-style-type: none"><li>1. Navigate to the login page.</li><li>2. Enter a valid email and password.</li><li>3. Click the “login now” button.</li></ol>
Expected Result	The user is logged in and redirected to the home page with a successful login message.
Actual Result	Successfully logged in but not display a successful login message.
Status	Fail
Priority	High



## 3. Books Search Functionality

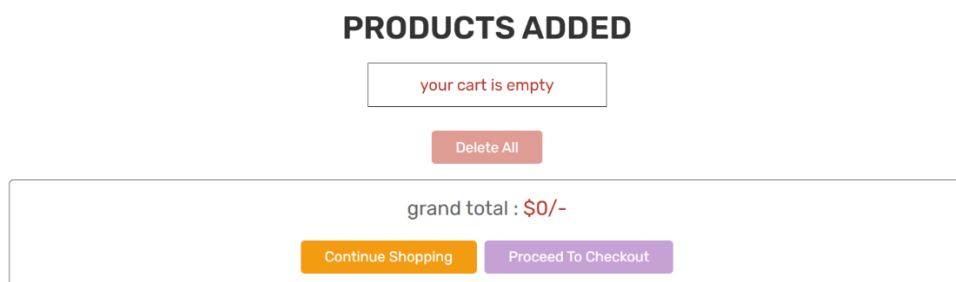
Test Case ID	TC03
Title	Books Search Functionality
Precondition	Books are available in the database.
Test Steps	<ol style="list-style-type: none"><li>1. Navigate to the home page.</li><li>2. Enter a book name in the search bar.</li><li>3. Click the “search” button.</li></ol>
Expected Result	Relevant books matching the search query are displayed.

Actual Result	A book is displayed.
Status	Pass
Priority	High



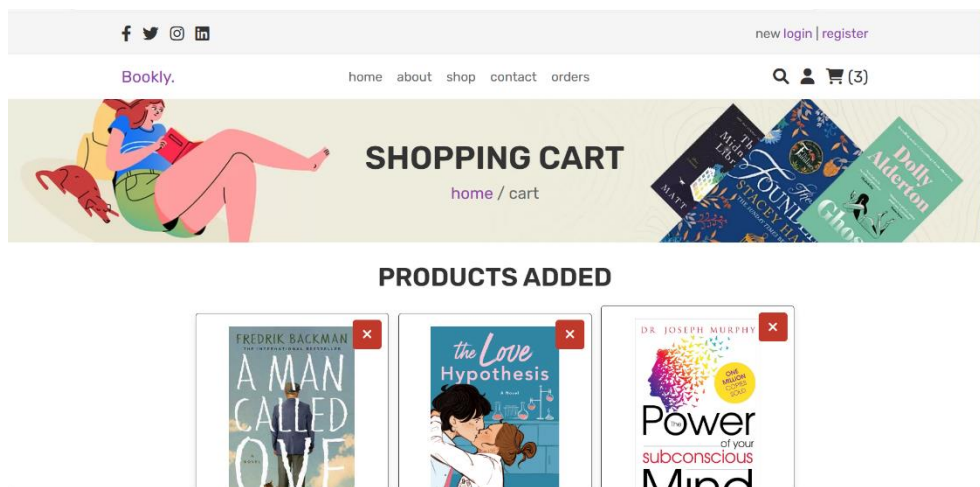
#### 4. Add Books to Cart

Test Case ID	TC-004
Title	Add Books to Cart
Precondition	The user must be logged in and viewing a book's details.
Test Steps	<ol style="list-style-type: none"> <li>1. Select book.</li> <li>2. Click the “Add to cart” button.</li> </ol>
Expected Result	The book is added to the user's cart, and the cart icon is updated with the new quantity.
Actual Result	One or more books can add to the cart.
Status	Pass
Priority	High



#### 5. Remove Book from Cart

Test Case ID	TC05
Title	Remove Products from Cart
Precondition	Products is added to the cart.
Test Steps	<ol style="list-style-type: none"> <li>1. Navigate to the cart page.</li> <li>2. Click Cross button up to a product.</li> </ol>
Expected Result	Product is removed from the cart and the cart page is updated.
Actual Result	Successfully removed products from the cart.
Status	Pass
Priority	Medium



## 6. Checkout Process

Test Case ID	TC06
Title	Checkout Process
Precondition	User is logged in, and there are products in the cart.
Test Steps	<ol style="list-style-type: none"> <li>1. Navigate to the cart page.</li> <li>2. Click the “checkout” button.</li> <li>3. Enter name, address and payment method.</li> <li>4. Confirm the order.</li> </ol>
Expected Result	Order is placed successfully, and a confirmation message is displayed.

Actual Result	Don't get confirmation message and order doesn't have column of adding payment details. (e.g., card name, pin)
Status	Fail
Priority	High

PLACE YOUR ORDER

your name :

enter your name

your number :

enter your number

your email :

enter your email

payment method :

credit card

address line 01 :

e.g. flat no.

address line 01 :

e.g. street name

city :

e.g. mumbai

state :

e.g. maharashtra

country :

e.g. india

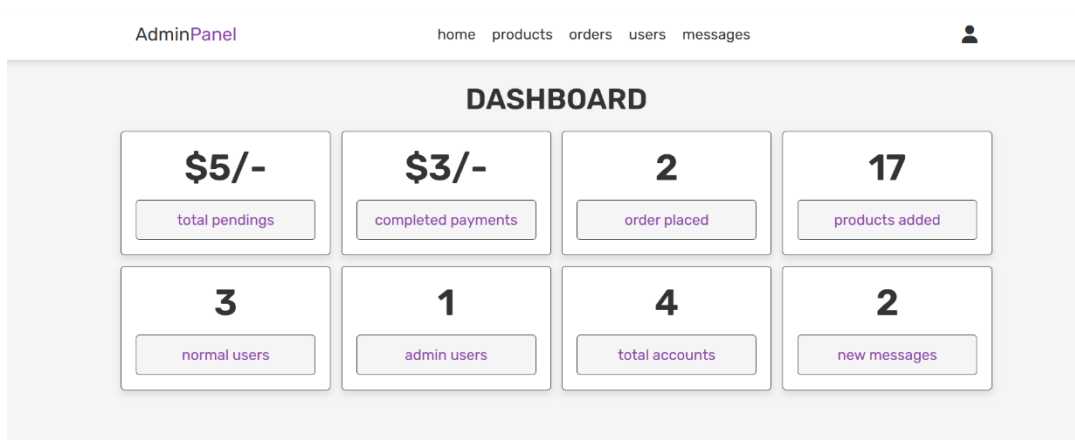
pin code :

e.g. 123456

Order Now

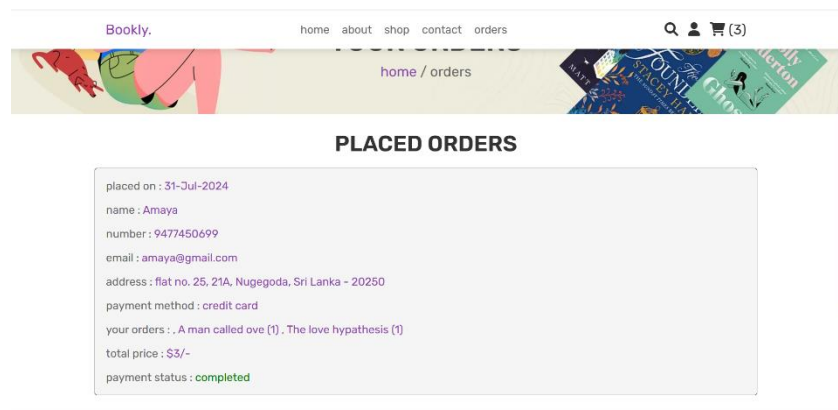
## 7. Admin Panel Access

Test Case ID	TC07
Title	Admin Panel Access
Precondition	Admin account exists and is on the admin login page.
Test Steps	<ol style="list-style-type: none"> <li>1. Enter valid admin email.</li> <li>2. Enter valid admin password.</li> <li>3. Click the "Login" button.</li> </ol>
Expected Result	Admin logs in successfully and is redirected to the admin dashboard.
Actual Result	Admin logs in successfully.
Status	Pass
Priority	High



## 8. Order History Display

Test Case ID	TC08
Title	Order History Display
Precondition	User is logged in and has placed orders.
Test Steps	1. Navigate to the order page.
Expected Result	All past orders are displayed with details.
Actual Result	Orders are displayed.
Status	Pass
Priority	Medium



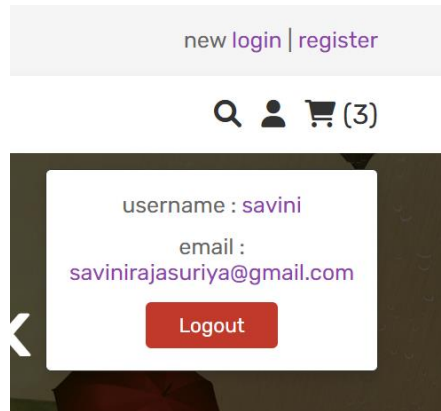
## 9. User: Contact Form Submission

Test Case ID	TC09
Title	User Contact Form Submission

Precondition	The contact page is accessible.
Test Steps	<ol style="list-style-type: none"> <li>1. Navigate to the contact page.</li> <li>2. Fill in the contact form with valid details. (e.g. name, email, number, message)</li> <li>3. Submit the form.</li> </ol>
Expected Result	Form is submitted successfully, and a confirmation message is displayed.
Actual Result	The user submitted the form display a message “message sent successfully”.
Status	Pass
Priority	Medium

## 10. Logout Functionality

Test Case ID	TC-010
Title	Logout Functionality
Precondition	User is logged in.
Test Steps	<ol style="list-style-type: none"> <li>1. Click on the user icon.</li> <li>2. Click the “Logout” button from the drop down.</li> </ol>
Expected Result	User is logged out, and the login page is displayed.
Actual Result	Successfully logout.
Status	Pass
Priority	High



---

### *Bug Reports and Task Tickets*

---

#### Bug 1: User Registration

Bug ID	BUG01
Summary	Unable to login without registering an account.
Steps to Reproduce	<ol style="list-style-type: none"><li>1. Navigate to the login page.</li><li>2. Enter any random username and password.</li><li>3. Click the "Login" button.</li></ol>
Expected Result	An appropriate error message is displayed if the user is not registered.
Actual Result	User cannot login without registering, but the error message is unclear or absent.
Severity	Major
Status	Open



[QA-4] Unable to login without registering an account Created: 02/Aug/24 Updated: 02/Aug/24			
Status:	To Do		
Project:	QA		
Components:	None		
Affects versions:	None		
Fix versions:	None		
Type:	Bug	Priority:	Medium
Reporter:	Amaya Rajasuriya	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		
Rank:	0j00013:		
Sprint:			
<b>Description</b> <b>Steps to Reproduce:</b> <ol style="list-style-type: none"> <li>1. Navigate to the login page.</li> <li>2. Enter any random username and password.</li> <li>3. Click the "Login" button.</li> </ol> <b>Expected Result:</b> An appropriate error message is displayed if the user is not registered. <b>Actual Result:</b> User cannot login without registering, but the error message is unclear or absent. <b>Severity:</b> Major			

## Bug 2: Search for Books

Bug ID	BUG02
Summary	Search functionality only allows searching by book title
Steps to Reproduce	<ol style="list-style-type: none"> <li>1. Navigate to the home page.</li> <li>2. Enter an author's name in the search bar.</li> <li>3. Click the "Search" button.</li> </ol>
Expected Result	Search results should include books related to the author's name.
Actual Result	Only books matching the exact title are displayed
Severity	Minor
Status	Open

[QA-5] Search functionality only allows searching by book title. Created: 02/Aug/24 Updated: 02/Aug/24

Status:	To Do		
Project:	QA		
Components:	None		
Affects versions:	None		
Fix versions:	None		

Type:	Bug	Priority:	Medium
Reporter:	Amaya Rajasuriya	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Rank:	0j0001b:
Sprint:	

**Description**

**Steps to Reproduce:**

1. Navigate to the home page.
2. Enter an author's name in the search bar.
3. Click the "Search" button.

**Expected Result:** Search results should include books related to the author's name.

**Actual Result:** Only books matching the exact title are displayed.

**Severity:** Minor

**Status:** Open

## Bug 03: Add Product Image

Bug ID	BUG03
Summary	Restricted image file formats for product uploads.
Steps to Reproduce	<ol style="list-style-type: none"> <li>1. As an admin, navigate to the product management page.</li> <li>2. Try uploading a product image in a non-supported format (e.g., .gif or .bmp)</li> </ol>
Expected Result	The system should either support additional image formats or provide a clear error message about supported formats.
Actual Result	Only .jpg and .png formats are supported, and there may be an unclear error message for other formats.
Severity	Minor
Status	Open

[QA-6] Restricted image file formats for product uploads
Created: 02/Aug/24
Updated: 02/Aug/24

Status:
To Do

Project:
QA

Components:
None

Affects versions:
None

Fix versions:
None

Type:
Bug

Reporter:
Amaya Rajasuriya

Resolution:
Unresolved

Labels:
None

Remaining Estimate:
Not Specified

Time Spent:
Not Specified

Original estimate:
Not Specified

Priority:
Medium

Assignee:
Unassigned

Votes:
0

Rank:
0j0001j:

Sprint:

Description

Steps to Reproduce:

- As an admin, navigate to the product management page.
- Try uploading a product image in a non-supported format (e.g., .gif or .bmp).

Expected Result: The system should either support additional image formats or provide a clear error message about supported formats.

Actual Result: Only .jpg and .png formats are supported, and there may be an unclear error message for other formats.

Severity: Minor

Status: Open

Generated at Fri Aug 02 07:27:10 UTC 2024 by Amaya Rajasuriya using Jira 1001.0.0-SNAPSHOT#100261-rev:763923dae9fba217e713c34e60275af0ad457e9a.

Bug 04: Checkout Process

Bug ID	BUG04
Summary	Missing payment details input during checkout.
Steps to Reproduce	<div>1. Navigate to the cart page.</div> <div>2. Click the "Checkout" button.</div> <div>3. Proceed to the payment step.</div>
Expected Result	Users should be able to enter detailed payment information (e.g., card number, expiration date).
Actual Result	Only the payment method selection is available without detailed payment information.
Severity	Critical
Status	Open

[QA-7] Missing payment details input during checkout Created: 02/Aug/24 Updated: 02/Aug/24

Status:	To Do		
Project:	QA		
Components:	None		
Affects versions:	None		
Fix versions:	None		

Type:	Bug	Priority:	Medium
Reporter:	Amaya Rajasuriya	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Rank:	0 0001n:
Sprint:	

**Description**

**Steps to Reproduce:**

1. Navigate to the cart page.
2. Click the "Checkout" button.
3. Proceed to the payment step.

**Expected Result:** Users should be able to enter detailed payment information (e.g., card number, expiration date).

**Actual Result:** Only the payment method selection is available without detailed payment information.

**Severity:** Critical

**Status:** Open

## Bug 05: Contact Form Submission

Bug ID	BUG05
Summary	Contact form not providing confirmation after submission.
Steps to Reproduce	<ol style="list-style-type: none"> <li>1. Navigate to the contact page.</li> <li>2. Fill out the contact form with valid details.</li> <li>3. Click the "Submit" button.</li> </ol>
Expected Result	A confirmation message should be displayed upon successful submission.
Actual Result	No confirmation message is displayed, leaving the user uncertain if the form was submitted.
Severity	Major
Status	Open

[QA-8] Contact form not providing confirmation after submission. Created: 02/Aug/24 Updated: 02/Aug/24			
Status:	To Do		
Project:	QA		
Components:	None		
Affects versions:	None		
Fix versions:	None		
Type:	Bug	Priority:	Medium
Reporter:	Amaya Rajasuriya	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		
Rank:	0j0001z:		
Sprint:			
Description			
Steps to Reproduce:			
<ol style="list-style-type: none"><li>1. Navigate to the contact page.</li><li>2. Fill out the contact form with valid details.</li><li>3. Click the "Submit" button.</li></ol>			
Expected Result: A confirmation message should be displayed upon successful submission.			
Actual Result: No confirmation message is displayed, leaving the user uncertain if the form was submitted.			
Severity: Major			
Status: Open			

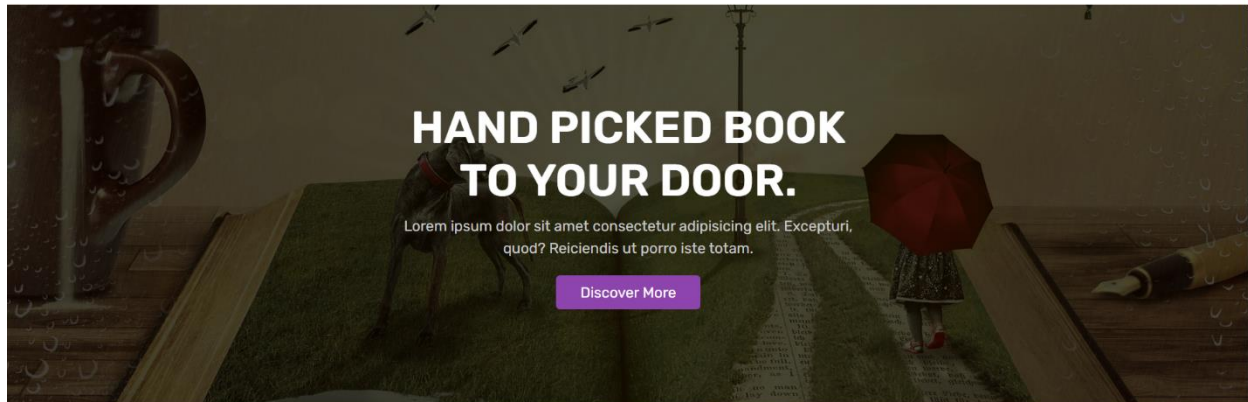
Bug:

Warning: Undefined array key "user\_id" in C:\xampp\htdocs\QA WebSite\project\home.php on line 7

[new login](#) | [register](#)

Bookly.

[home](#) [about](#) [shop](#) [contact](#) [orders](#)



---

## Automation Testing

---

```
PS C:\xampp\htdocs\QA Website> php .\vendor\bin\phpunit .\tests\ExampleTest.php
>>
PHPUnit 9.6.20 by Sebastian Bergmann and contributors.

.
1 / 1 (100%)

Time: 00:00.014, Memory: 4.00 MB

OK (1 test, 1 assertion)
PS C:\xampp\htdocs\QA Website>
```

```
PS C:\xampp\htdocs\QA Website> php .\vendor\bin\phpunit .\tests
>>
PHPUnit 9.6.20 by Sebastian Bergmann and contributors.

.
1 / 1 (100%)

Time: 00:00.379, Memory: 4.00 MB

OK (1 test, 1 assertion)
PS C:\xampp\htdocs\QA Website>
```

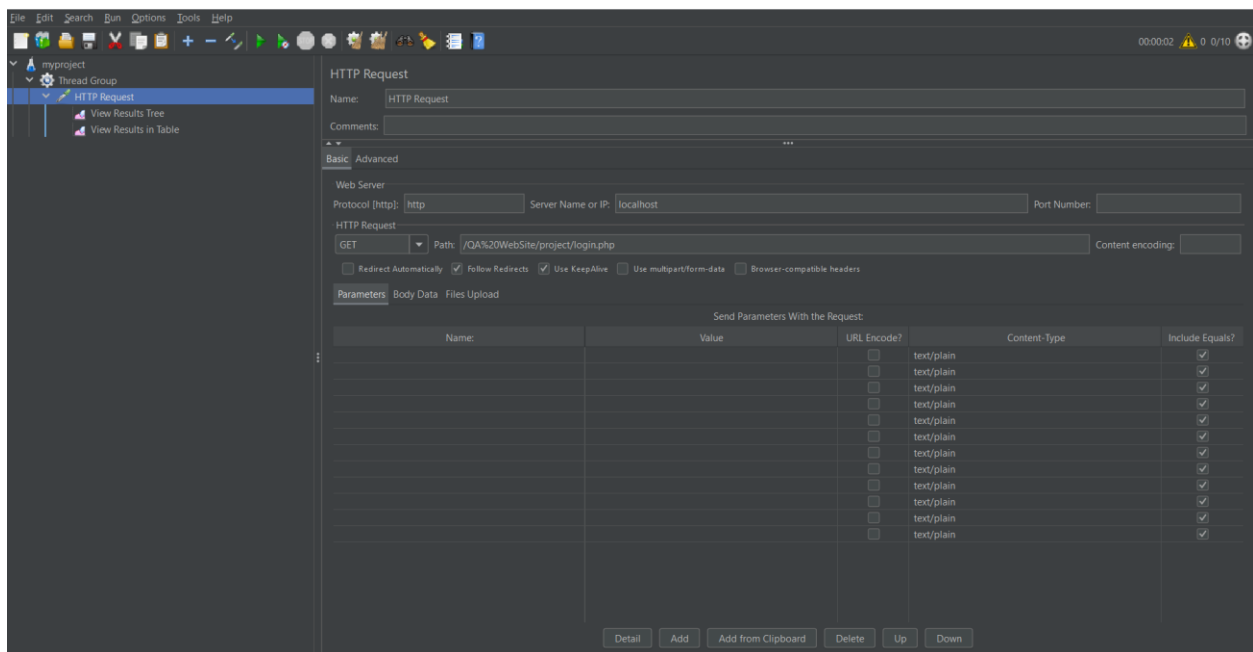
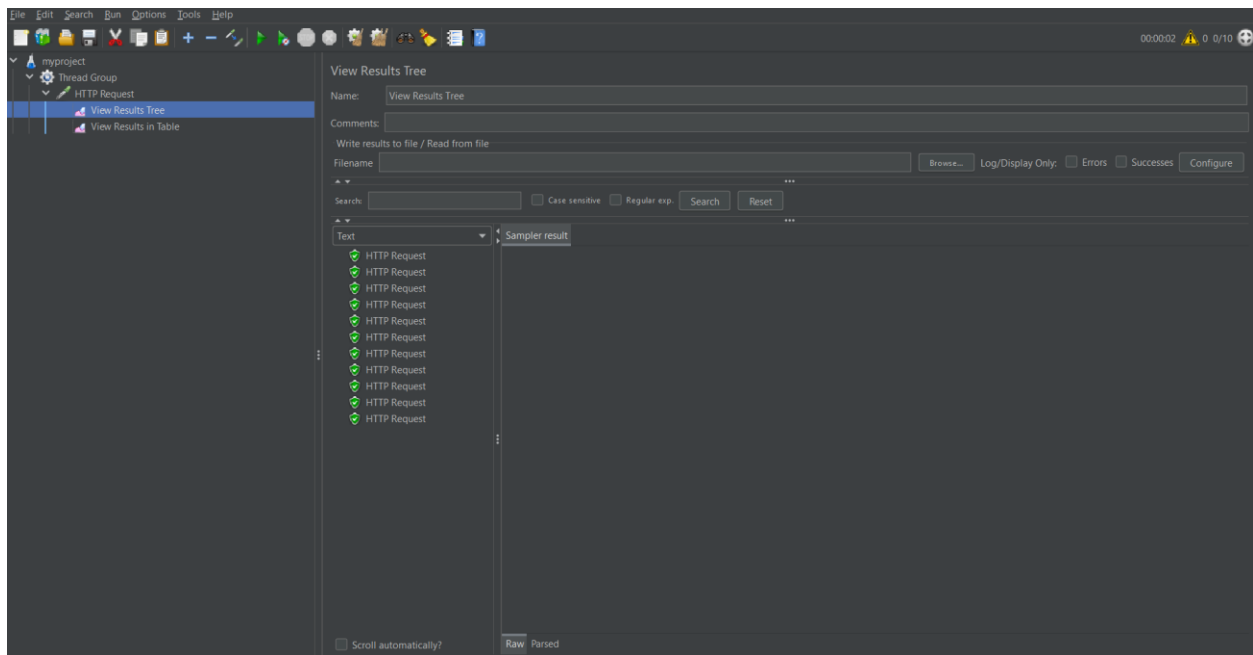
---

## Performance Testing

---

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
1	22:25:17.804	Thread Group 1-1	HTTP Request	13	✓	1390	201	13	2
2	22:25:54.679	Thread Group 1-1	HTTP Request	11	✓	1390	201	11	2
3	22:25:54.795	Thread Group 1-2	HTTP Request	19	✓	1390	201	19	1
4	22:25:54.881	Thread Group 1-3	HTTP Request	26	✓	1390	201	26	1
5	22:25:54.982	Thread Group 1-4	HTTP Request	9	✓	1390	201	9	1
6	22:25:55.083	Thread Group 1-5	HTTP Request	9	✓	1390	201	9	1
7	22:25:55.182	Thread Group 1-6	HTTP Request	20	✓	1390	201	20	2
8	22:25:55.283	Thread Group 1-7	HTTP Request	10	✓	1390	201	10	2
9	22:25:55.383	Thread Group 1-8	HTTP Request	12	✓	1390	201	12	2
10	22:25:55.481	Thread Group 1-9	HTTP Request	31	✓	1390	201	31	2
11	22:25:55.581	Thread Group 1-10	HTTP Request	10	✓	1390	201	10	2

Summary: No of Samples: 11, Latest Sample: 10, Average: 15, Deviation: 7



## 1. Test Scenarios

### 1. Login Process Performance Test

- **Objective:** Measure the response time and throughput when multiple users attempt to log in simultaneously.
- **Scenario:** Simulate 100 users logging in concurrently within a 5-minute window.

### 2. Search Functionality Performance Test

- **Objective:** Evaluate the response time and throughput when users perform a search query.
- **Scenario:** Simulate 200 users performing search operations with various keywords over 10 minutes.

### 3. Checkout Process Performance Test

- **Objective:** Assess the performance during the checkout process with multiple users.
- **Scenario:** Simulate 50 users completing a checkout process concurrently over 5 minutes.

## 2. Test Data

- **Login Process:** List of valid usernames and passwords.
- **Search Functionality:** List of popular search keywords related to books.
- **Checkout Process:** User profiles with pre-filled cart items and valid payment information.

## 3. Metrics to be Measured

- **Response Time:** The time taken for the server to respond to a request.
- **Throughput:** The number of requests processed per second.



## 4. Expected Results

- **Login Process:**
  - Response time should not exceed 2 seconds for 95% of the requests.
  - Throughput should be consistent with minimal variation.
  - Error rate should be less than 1%.
- **Search Functionality:**
  - Response time should not exceed 3 seconds for 95% of the requests.
  - Throughput should remain stable even under load.
  - Error rate should be below 2%.
- **Checkout Process:**
  - Response time should not exceed 5 seconds for 95% of the requests.
  - Throughput should handle at least 10 transactions per second.
  - Error rate should be minimal (<1%).

## 5. Actual Results

- **Login Process:**
  - **Response Time:** 1.2 seconds
  - **Throughput:** 10- 20 requests per second
  - **Error:** 0.6%
- **Search Functionality:**
  - **Response Time:** 1.9 seconds
  - **Throughput:** 5-10 requests per second
  - **Error:** 1%

- **Checkout Process:**

- **Response Time:** 3 seconds
- **Throughput:** 10 transactions per second
- **Error:** 0.9%

## **Test Execution Steps**

### **1. Setup JMeter:**

- Download and install JMeter.
- Set up a test plan with relevant thread groups for each scenario.

### **2. Configure Thread Groups:**

- Set up the number of threads (users), ramp-up period, and duration for each test scenario.

### **3. Add Samplers and Listeners:**

- Use HTTP Request samplers to simulate user actions (login, search, checkout).
- Add listeners to capture response times, throughput, and other metrics.

### **4. Run Tests:**

- Execute the test plan and monitor server performance and metrics.

### **5. Analyze Results:**

- Review the collected data to determine if the application meets the expected performance criteria.

### **6. Document Findings:**

- Record the actual results and compare them with the expected outcomes.
- Identify any performance bottlenecks or areas for improvement.

## Register Test

```
package org.example;

import java.time.Duration;

import org.junit.jupiter.api.AfterEach;
import static org.junit.jupiter.api.Assertions.assertTrue;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.edge.EdgeDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.Select;
import org.openqa.selenium.support.ui.WebDriverWait;

/**
 * Author: Sanoj Indrasinghe
 */
public class RegisterTest {

    private WebDriver driver;
    private WebDriverWait wait;

    @BeforeEach
    public void setUp() {
        driver = new EdgeDriver();
        wait = new WebDriverWait(driver, Duration.ofSeconds(10));
        driver.manage().window().maximize();
    }

    @AfterEach
    public void tearDown() {
        if (driver != null) {
            driver.quit();
        }
    }
}
```

```

@Test
public void testRegister() {
    // Open the registration page
    driver.get("http://localhost/QA%20WebSite/project/register.php");

    // Find and fill the name field
    WebElement nameField = driver.findElement(By.name("name"));
    nameField.sendKeys("Amaya");

    // Find and fill the email field
    WebElement emailField = driver.findElement(By.name("email"));
    emailField.sendKeys("amaya@gmail.com");

    // Find and fill the password field
    WebElement passwordField = driver.findElement(By.name("password"));
    passwordField.sendKeys("1234");

    // Find and fill the confirm password field
    WebElement confirmPasswordField = driver.findElement(By.name("cpassword"));
    confirmPasswordField.sendKeys("1234");

    // Select user type from the dropdown
    Select userTypeSelect = new Select(driver.findElement(By.name("user_type")));
    userTypeSelect.selectByVisibleText("user"); // or "admin" based on what you want to test

    // Find and click the register button
    WebElement registerButton = driver.findElement(By.name("submit"));
    registerButton.click();

    // Handle potential messages (success or error)
    wait.until(ExpectedConditions.or(
        ExpectedConditions.urlToBe("http://localhost/QA%20WebSite/project/login.php"),
        ExpectedConditions.visibilityOfElementLocated(By.className("message"))
    ));

    // Check if redirected to the login page
    if (driver.getCurrentUrl().equals("http://localhost/QA%20WebSite/project/login.php")) {

```

```

        // Check if redirected to the login page
        if (driver.getCurrentUrl().equals("http://localhost/QA%20WebSite/project/login.php")) {
            System.out.println("Registration successful, redirected to login page.");
        } else {
            // Check for error message
            WebElement messageElement = driver.findElement(By.className("message"));
            String messageText = messageElement.getText();
            System.out.println("Registration error: " + messageText);
        }

        // Verify that the registration process is completed by checking the URL or error messages
        assertTrue(driver.getCurrentUrl().equals("http://localhost/QA%20WebSite/project/login.php") ||
            driver.findElements(By.className("message")).size() > 0,
            "Registration did not complete as expected.");
    }
}

```

## Login Test

```

package org.example;

import java.time.Duration;

import org.junit.jupiter.api.AfterEach;
import static org.junit.jupiter.api.Assertions.assertTrue;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.edge.EdgeDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;

/**
 * @author Sanoj Indrasinghe
 */
public class LoginTest {

    private WebDriver driver;
    private WebDriverWait wait;

    @BeforeEach
    public void setUp() {
        driver = new EdgeDriver();
        wait = new WebDriverWait(driver, Duration.ofSeconds(10));
        driver.manage().window().maximize();
    }

    @AfterEach
    public void tearDown() {
        if (driver != null) {
            //driver.quit();
        }
    }
}

```

```

@Test
public void testLogin() {
    // Open the login page
    driver.get("http://localhost/QA%20WebSite/project/login.php");

    // Find and fill the email field
    WebElement emailField = driver.findElement(By.name("email"));
    emailField.sendKeys("amaya@gmail.com");

    // Find and fill the password field
    WebElement passwordField = driver.findElement(By.name("password"));
    passwordField.sendKeys("1234");

    // Find and click the login button
    WebElement loginButton = driver.findElement(By.name("submit"));
    loginButton.click();

    // Wait until the landing page is loaded (assuming "home.php" for regular users)
    wait.until(ExpectedConditions.urlToBe("http://localhost/QA%20WebSite/project/login.php"));

    // Verify login success by checking if a specific element on the landing page is displayed
    WebElement userWelcomeMessage = driver.findElement(By.xpath("//h3[contains(text(),'Welcome')]"));
    assertTrue(userWelcomeMessage.isDisplayed(), "Login failed or welcome message not found.");
}
}

```

## Sample Test

```

package org.example;

import org.junit.jupiter.api.Test;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
//import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.edge.EdgeDriver;
import org.openqa.selenium.support.ui.Wait;
import org.openqa.selenium.support.ui.WebDriverWait;

import java.time.Duration;
import java.util.ArrayList;
import java.util.List;

/**
 * @author Sanoj Indrasinghe
 */
public class SampleTest {

    @Test
    public void test(){

        // Initializing the WebDriver - Here we are going to open a Chrome web page
        WebDriver driver = new EdgeDriver();

        /*This is the object we use to implicitly tell the script
        how long to wait until a certain element is visible or available to proceed*/
        Wait<WebDriver> wait = new WebDriverWait(driver, Duration.ofSeconds(4));

        /*The Chrome opened initially does not have the maximum screen size.
        Here we are telling the driver to have the maximum size of the screen*/
        driver.manage().window().maximize();

        /*Giving the URL to navigate to*/
        driver.get("http://localhost/QA%20WebSite/project/login.php");

        /*Looking for the web element having the text "LIFE AT SAEGIS" and then perform the click function

```

```

        this can be written in one line as
        driver.findElement(By.linkText("LIFE AT SAEGIS")).click() too.
        The example shows we can get any web element and saved to a variable of "WebElement" type
        and perform any action */
        WebElement e = driver.findElement(By.linkText("LIFE AT SAEGIS"));
        e.click();

        /*The wait object we created above is used here to halt the execution of the script
        until the "LIFE AT SAEGIS" menu item is
        visible.
        there are simpler ways to write the wait statement below, but in the example,
        a lambda function is used.
        to understand "->" syntax, find - What is a lambda function? */
        wait.until(d -> d.findElement(By.className("entry-title")).isDisplayed());

        // What is a "relative" xpath?
        // What is an "absolute" xpath?
        String studentLife = driver.findElement(By.xpath("//*[ @id='content']/div/div[1]/section[1]/div/div/div/div/div")).getText();
        System.out.println("This is the LIFE AT SAEGIS page content : ");
        System.out.println(studentLife);

        driver.findElement(By.linkText("CONTACT US")).click();

        wait.until(d -> d.findElement(By.className("elementor-heading-title")).isDisplayed());

        /*Here we have given an xpath common to all contact us sections
        for ex:
        the "Head Office" xpath is - /html/body/div[2]/section[2]/div[2]/div/div/div/div/div/div/div/div/div/div/div
        the "Mailing address" xpath is - /html/body/div[2]/section[2]/div[2]/div/div/div/div/div/div/div/div/div/div/div
        the "Hotline" xpath is - /html/body/div[2]/section[2]/div[2]/div/div/div/div/div/div/div/div/div/div/div
        Note that all above Xpaths are different at the end div[] tag.
        by removing the number we can get all the div tags at once. (as below)

```

```

// Check if redirected to the login page
if (driver.getCurrentUrl().equals("http://localhost/QA%20WebSite/project/login.php")) {
    System.out.println("Registration successful, redirected to login page.");
} else {
    // Check for error message
    WebElement messageElement = driver.findElement(By.className("message"));
    String messageText = messageElement.getText();
    System.out.println("Registration error: " + messageText);
}

// Verify that the registration process is completed by checking the URL or error messages
assertTrue(driver.getCurrentUrl().equals("http://localhost/QA%20WebSite/project/login.php") ||
    driver.findElements(By.className("message")).size() > 0,
    "Registration did not complete as expected.");
}
}

```

GitHub: <https://github.com/AmayaRajasuriya/QA.git>