CHAPTER 1: INTRODUCTION

PURPOSE:

This document outlines the testing procedures and strategies for MARU DRY FRUIT **(ONLINE DRY FRUIT STORE).**

SCOPE OF TESTING:

In this report we are going to test **Login, Logout, Sign Up Page,add dry fruits,delete dryfruits,add to cart** by unit testing.

Definitions/Abbreviations:

HTML: A markup language used for structuring content on the web.

CSS: Cascading Style Sheets, a style sheet language for describing the presentation of a document written in HTML.

JavaScript: A programming language that enables interactive web pages and dynamic content.

Bootstrap 5: A popular front-end framework for building responsive and visually appealing web pages.

PHP: A server-side scripting language for web development, used to create dynamic web pages.

MySQL: A relational database management system that stores and manages data for web applications.

Star UML: A UML (Unified Modeling Language) tool used for modeling, designing, and visualizing software systems.

References:

https://www.geeksforgeeks.org/software-testing-techniques/

https://www.zoho.com/qengine/cockpit/software-testing-techniques.html

Overview: This document shows testing procedures for MARU DRY FRUIT STORE, ensuring robustness. No additional tools are required for testing, as many features are tested during development and by users manually.

CHAPTER 2:

TEST PLAN:

OBJECTIVES:

The goal of the test plan is to make sure that Sovereign soirees works as intended, meeting quality standards and what users need.

Testing Types:

1. Unit Testing:

Goal: Check individual parts (like functions and methods) of the application.

2. Integration Testing:

Goal: Confirm that different parts work well together.

3. System Testing:

Goal: Make sure the whole system works as it should, following specific requirements.

4. Acceptance Testing:

Goal: Make sure the system meets what users expect and need.

Test Environment:

Hardware:

Standard laptop/system or server specifications (no specific hardware requirements, but latest generation for optimal results)

Software:

Development Environment:

HTML, CSS, JS ,PHP: Latest versions

Database: My SQL

Testing Frameworks:

Unit testing

Browser: any (Microsoft egde,opera)

Tools:

No additional tools are required for testing, as many features are tested during development and by users manually.

TEST SCENARIOS:

Test Scenario ID	Test Scenario	Expected Result	Acceptance Criteria	Date Of Implementation
TS-001	User login.	Successful login	User can log in with valid credentials	4th Jan 2024
TS-002	User sign up.	Successful registration	New user is added to the database	4th Jan 2024
TS-003	logout	Successful logout	Logout successful	4th Jan 2024
TS-004	Add Dryfruits	Successful addition	New product added to the table	5th Jan 2024
TS-005	Delete Dryfruit	Successful deletion	Product should be deleted from the table	5th Jan 2024
TS-006	Add to cart	Successful addition of product in cart	Products should be added in cart	5th Jan 2024

TEST CASES:

TEST SCENARIO ID: TS-001 **TEST SCENARIO:** User login

TEST CASE ID	TEST STEPS	TEST DATA	EXPECETED RESULT	STATUS
TC-001	Enter email	Valid Username	Successful Login	PASS
	Enter password			
	Click 'login'			
TC-002	Enter email	Invalid Username	Login failed	PASS
	Enter password			
	Click 'login'			
TC-003	Enter email	Correct Password	Successful Login	PASS

	Enter password			
	Click 'login'			
TC-004	Enter email	Incorrect Password	Login failed	PASS
	Enter password			
	Click 'login'			

TEST SCENARIO ID: TS-002 **TEST SCENARIO:** Sign up

TEST CASE ID	TEST STEPS	TEST DATA	EXPECETED RESULT	STATUS
TC-001	Enter email	Valid email	Successful registration	pass
	Enter password etc	Valid password		
	Click 'Sign up'			
TC-002	Enter email	Duplicate email	Registration failed	pass
	Enter password etc			
	Click 'Sign up'			
TC-003	Enter email	Invalid email format	Registration failed	pass
	Enter password etc			
	Click 'Sign up'			

TEST SCENARIO ID: TS-003 **TEST SCENARIO:** log out

TEST CASE ID	TEST STEPS	TEST DATA	EXPECETED RESULT	STATUS
TC-001	Click logout	-	Successful(session destroy)	pass

TEST SCENARIO ID: TS-004 **TEST SCENARIO:** Add Fruits

TEST CASE ID	TEST STEPS	TEST DATA	EXPECETED RESULT	STATUS
TS-001	Provide all information	Provide all information	Successful insertion	pass
TS-002	Provide missing information	Provide missing information	Insertion unsuccessful	pass

TEST SCENARIO ID: TS-005 **TEST SCENARIO:** delete Fruits

TEST CASE ID	TEST STEPS	TEST DATA	EXPECETED RESULT	STATUS
TS-001	Provide existing product id	Provide id	Successful deletion	pass
TS-002	Provide non-existing product id	Provide id	Message “no data to delete”	pass

TEST SCENARIO ID: TS-006 **TEST SCENARIO:** add to cart

TEST CASE ID	TEST STEPS	TEST DATA	EXPECETED RESULT	STATUS
TS-001	Navigate to the product page.	Product details for a specific item**.**		pass
	Click on the 'Add to Cart' button.			pass
	View the cart to confirm the added item.		The selected item should be successfully added to the cart.	pass

CHAPTER 3:

TESTING PROCEDURE

Login

1. Verify Successful Login:

- Set up valid credentials (username and password).
- Attempt to log in using the valid credentials.
- Confirm the login success, such as by checking if the user(customer) is redirected to the home page.
- Confirm the login success, such as by checking if the user(admin) is directed to the Admin page.

2. Validate Failed Login - Incorrect Password:

- Use a valid username and an incorrect password.
- Try logging in with the provided credentials.
- Verify that the login fails and check for appropriate error messaging.

3. Validate Failed Login - Incorrect Username:

- Utilize an invalid username and a valid password.
- Log in using the provided credentials.
- Confirm the login failure and check for relevant error messaging.

4. Validate Failed Login - Empty Credentials:

- Leave both username and password fields empty.
- Attempt to log in with the empty credentials.
- Ensure that the login fails and the system prompts the user to input both username and password.

Testing Successful Logout:

- Set up a login using valid credentials.
- Initiate the logout action.

- Confirm that the user has been successfully logged out and redirected to the login page.

Sign Up Page

1. Validate Successful Registration:

- Set up valid user details, including a username, password, and email.
- Submit the signup form.
- Verify that the user is registered successfully and redirected to the appropriate page.

2. Confirm Duplicate Email Detection:

- Provide an existing email address in the signup form.
- Submit the form and confirm that the system recognizes the duplicate email, informing the user accordingly.

3. Validate Email Format:

- Input an invalid email address format in the signup form.
- Submit the form and confirm that the system validates the email format, prompting the user for a valid email.

4. Check Database Integration:

- Examine the database for the user's information following a successful signup.
- Confirm the correct storage of the user's data in the database.
- Assert data integrity and consistency in the database.

Add Dryfruit

1.Basic Details:

- Arrange: prepare to add a new dryfruit.
- Act: Enter fundamental details like name, price and image for the new dryfruit.
- Assert: Confirm correct storage and display some message.

2.Missing Information:

If we left any field empty, it should take some action to show product cannot be stored with the missing information.

3.Verify in Database:

- Arrange: Ensure all details are entered for the new dryfruit.
- Act: Save the information.
- Assert: Access the database, to confirm correct storage of the new dryfruit details.

Delete Dryfruit:

1.Provide existing Id:

- Arrange: Choose an existing dryfruit in the admin panel.
- Act: Delete the selected dryfruit.
- Assert: Confirm the removal of the dryfruit from the admin panel , database and on other pages

2.Provide any non-existing Id:

It should give a message “no data to delete”.

Cart Functionality

1. Add to Cart:
   - Visit a product page.
   - Click 'Add to Cart' for a product.
   - Verify the product is added to the cart.
2. View Cart:
   - Navigate to the cart page.
   - Confirm all added items are displayed.
3. Manage Cart - Update Quantity:
   - Navigate to the cart page.
   - Update the quantity of an item.
   - Verify the cart reflects the updated quantity.
4. Manage Cart - Remove Item:
   - Navigate to the cart page.
   - Remove an item from the cart.
   - Verify the cart no longer contains the removed item.

View Customer Information

1. View Customer Information:

- Visit the customer information page.
- Verify the page displays customer details from the 'order_manager' table.

2. View Empty Customer Information:

- Simulate an empty 'order_manager' table.
- Visit the customer information page.
- Verify the page displays a message indicating no customer data.

Blog Page

1. View Blog Content:

- Visit the blog page.
- Verify nutritional information about pistachios and expert recommendations are displayed.

2. Verify Copyright and Sitemap:

- Confirm the presence of the copyright notice and sitemap link at the end of the blog page.

DEFECT TRACKING:

Defect Tracking for "Full Product Name Not Showing in Cart" Issue:

1. Description:

- When attempting to add a product to the cart, the full name of the product is not displayed as expected.

2. Steps to Reproduce:

- Visit a product page.
- Click 'Add to Cart' for a specific product.
- Navigate to the cart page.

3. Expected Result:

- The full name of the added product should be visible in the cart.

4. Actual Result:

- The cart does not display the full name of the product; it might show a truncated or incomplete name.

5. Severity:

- Moderate, as it impacts user experience and clarity in cart management.

6. Resolution Steps:

- Investigate the code related to adding products to the cart.
- Identify the source of the issue causing the product name not to display correctly.
- Implement a fix to ensure the full product name is accurately shown in the cart.

7. Status:

- Open/In Progress

Defect Tracking for "Grand Total Not Added to Database" Issue:

1. Description:

- The grand total, representing the cumulative cost of items in the cart, is not being added to the database after a purchase.

2. Steps to Reproduce:

- Add multiple items to the cart.
- Proceed to the checkout and complete the purchase.
- Check the database for the recorded grand total.

3. Expected Result:

- The grand total of the purchase should be accurately added to the database.

4. Actual Result:

- The database does not reflect the correct grand total value after completing a purchase.

5. Severity:

- High, as it affects accurate recording of financial transactions and order management.

6. Resolution Steps:

- Examine the code related to the checkout and purchase process.
- Identify the section responsible for calculating and storing the grand total.
- Implement a fix to ensure the correct grand total is recorded in the database.

7. Status:

- Open/In Progress

Chapter 4: Test Reports

Summary:

METRIC

Metric	Target/Threshold	Focus Areas
Test Coverage (> 80%)	Aim for > 80%	Critical functionalities: Cart management, purchase handling, user authentication
Pass/Fail Rates (High Pass Rates)	Strive for high rates	Key user interactions: Login, sign up, cart management, customer information
Defect Density (< 0.1)	Maintain < 0.1 defects	Focus on critical functionalities
Response Time (< 500 ms)	Ensure < 500 ms	Actions: Adding items to the cart, logging in
Usability (> 4/5)	Target > 4/5	Features: Shopping cart, login/logout, blog
Security Vulnerabilities	Address within 1 week	Areas: Authentication, purchase handling, user data storage

Recommendations:

After evaluating the Maru Dry Fruit Store , there is room for improvement. Expanding the scope of our testing to encompass more facets of the system is essential to ensure comprehensive functionality. Addressing any performance bottlenecks and enhancing the system's responsiveness will contribute to a more satisfying user experience.

Actively seeking user feedback is crucial for refining the system's appearance and functionality. Regularly identifying and resolving issues will fortify the system's security, ensuring a seamless and

problem-free experience for users. Employing streamlined and efficient methods to assess the system's integrity will facilitate prompt detection of potential issues.

To enhance user proficiency, focus on simplifying problem-solving processes and implementing quicker diagnostics to ensure ongoing system robustness. Additionally, providing user support and promoting ease of use will contribute to an improved overall user experience. Monitoring system capacity to accommodate concurrent users is vital to meet the evolving needs of Maru Dry Fruit Store's user base. Implementing these enhancements will elevate Maru Dry Fruit Store to meet and exceed user expectations.

Chapter 5: Conclusion

Lessons Learned

- Commencing testing at an early stage proves beneficial in identifying and resolving issues promptly during the development process.
- Meticulous test planning is essential to ensure comprehensive coverage and validation of all critical components.
- Actively engaging with user feedback is a cornerstone for continuous system improvement, aligning the system with user expectations.
- Maintaining a consistent focus on security, with regular checks, is an ongoing necessity for robust system protection.
- Timely resolution of identified issues is crucial for maintaining a smooth and efficient development workflow.
- Regular testing of key components is fundamental to early detection and mitigation of potential challenges.
- Prioritizing usability in the testing process ensures that the system remains intuitive and enjoyable for users.
- Incorporating continuous monitoring tools is vital for proactively identifying and addressing potential issues.

FUTURE CONSIDERATIONS

In considering the future, several aspects could enhance the development and testing processes:

1. **Automation and AI Integration**: Explore further integration of automation tools and artificial intelligence to streamline repetitive tasks, improve efficiency, and enhance accuracy in testing processes. This could include advanced testing frameworks and intelligent test case generation.

2. **Scalability Planning**: Anticipate future growth and plan for scalability. Ensure that the system and testing processes can handle increased loads and expanded functionalities. This involves evaluating current infrastructure, adopting scalable technologies, and designing flexible architectures.

3. Continuous Learning and Training: Invest in ongoing training and skill development for the development and testing teams. Staying abreast of emerging technologies, industry best practices, and evolving security standards is essential for long-term success.

4. User-Centric Development: Place a continued emphasis on understanding user needs and preferences. Regular user testing, feedback sessions, and usability studies can contribute to a more user-friendly system that evolves with changing user expectations.