OpenCart Project

Final Year Project 2024

Supervised By : Eng. Youssef Abdellah

Presented By :

Ahmed Farag
Ahmed Saad
Marina Nagy
Hossam Ashraf
Veronica Melad
Mennatallah Farag
Pancee Muhammad

Table of Contents

Chapter 1 -Introduction:

1.1 Purpose 3
1.2 Project Proposal 3

Chapter 2 -Overall Description:

2.1 System Environment 5
2.2 Functional Requirements 5
2.3 Non-Functional Requirements 6
2.4 Use Case Diagrams 7

Chapter 3 -System Design:

3.1 Data Flow Diagrams(DFD-level 0 and DFD-level 1) 8
3.1.1 DFD-level 0 8
3.1.2 DFD-level 1 9
3.2 State Diagram 10
3.3 Sequence Diagram 11
3.4 Prototype 12

Chapter 4 – Implementation & Testing:

4.1 Introduction 16
4.2 Testing Strategy 16
4.3 Testing Process 17
4.4 Testing Results 17
4.5 Layouts 18
4.6 Layouts Report 21

Chapter 5 - References:

5.1 Conclusion 22
5.2 Report References 24

Chapter 1 – Introduction:

1.1 Purpose

The purpose of this project is to create an e-commerce platform named “OpenCart Project” which provides users with a seamless online shopping experience. This website is designed to offer a wide variety of products, primarily focusing on electronics and gadgets, such as mobile phones, laptops, cameras, and accessories. The goal is to make online shopping easy, efficient, and enjoyable for customers by featuring a user-friendly interface, a streamlined checkout process, and various payment options. Additionally, the platform aims to support merchants by offering tools to manage product listings, orders, and customer interactions effectively.

Key objectives include:

- Offering a diverse range of quality products at competitive prices.
- Providing a smooth and intuitive user experience, with clear navigation, search functionality, and product categorization.
- Ensuring a secure platform for transactions and user data.
- Enabling merchants to list, sell, and manage their products effortlessly.

1.2 Project Proposal

The " OpenCart Project” " project proposes the development of a fully functional e-commerce website catering to tech enthusiasts and general consumers looking for electronics and gadgets. The platform will include the following core features:

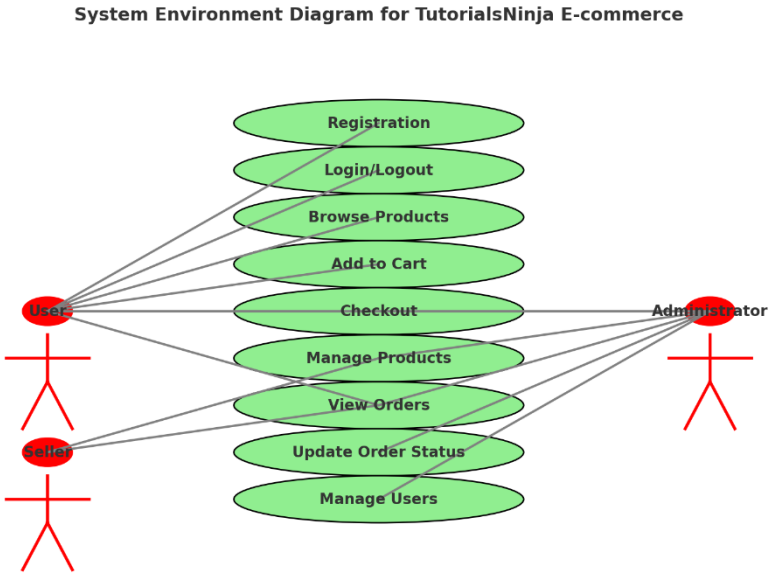
- **Product Catalog:** A comprehensive list of electronics products, with details such as images, descriptions, prices, and availability.
- **User Accounts:** Registration and login features for customers and sellers, with the ability to manage orders, wishlist, and account settings.
- **Shopping Cart and Checkout:** An easy-to-use shopping cart that allows users to review selected items, modify quantities, and proceed to checkout. Multiple payment gateways will be integrated to facilitate smooth transactions.

- **Search and Filtering:** Advanced search options and filters to help users find products based on various criteria like price range, brand, and specifications.
- **Order Management:** For customers to track orders and for sellers to manage inventory and shipments.
- **Security Features:** Ensuring secure data handling through encryption and secure payment processing.
- **Responsive Design:** A mobile-friendly layout to ensure accessibility across various devices.

The project aims to deliver a high-quality, user-centric platform that can compete with established e-commerce websites while providing a unique shopping experience that emphasizes simplicity and speed.

Chapter 2 – Overall Description:

2.1 System Environment



2.2 Functional Requirements

List the essential features and functionalities that the system must provide.

- **User Management:**
  - Users should be able to register, log in, and manage their accounts.
  - Password recovery and update functionalities should be available.

- **Product Management:**
  - Sellers can add, update, or delete product listings.
  - Users can view product details, including name, description, price, images, and stock availability.

- **Shopping Cart and Checkout:**
  - Users can add, update, or remove items from the shopping cart.
  - The system will calculate total prices and taxes.
  - Multiple payment gateways (e.g., PayPal, Stripe, credit card) should be available.

- **Order Management:**
  - Users can view order history and track their order status.
  - Admins can manage orders, including updating status and sending notifications.
- **Search and Filtering:**
  - Users can search for products using keywords.
  - Filters based on categories, price, and ratings should be available.
- **Security:**
  - Secure login and data encryption.

2.3 Non-Functional Requirements

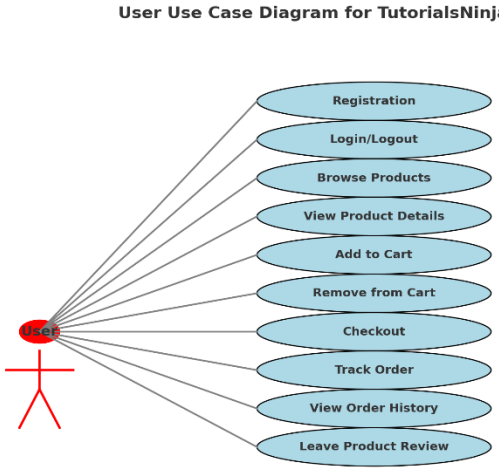
Outline the criteria that define the system's operation rather than it functionalities.

- **Performance:**
  - The system should support at least 100 concurrent users.
  - Page load times should be under 3 seconds for the main features.
- **Usability:**
  - The interface should be intuitive and user-friendly.
  - Tutorials or user guides should be available for new users.
- **Reliability:**
  - The system should have 99.9% uptime.
  - Data loss should be prevented through regular backups.
- **Scalability:**
  - The system should support scaling to handle increased traffic.
  - Additional servers can be added to support high user demands.
- **Security:**
  - User data must be encrypted.
  - Regular security audits and penetration testing should be conducted.
- **Maintainability:**
  - Code should follow industry best practices and be well-documented.
  - Bug fixes and updates should be manageable with minimal downtime.

2.4 Use Case Diagrams

1. User Use Case Diagram:

- **Actors:** User
- **Use Cases:**
  - Registration
  - Login/Logout
  - Browse Products
  - View Product Details
  - Add to Cart
  - Remove from Cart
  - Checkout
  - Track Order
  - View Order History
  - Leave Product Review



2. Seller Use Case Diagram:

- **Actors:** Seller
- **Use Cases:**
  - Login/Logout
  - Manage Products (Add, Edit, Delete)
  - View Sales Reports

- Process Orders
- Update Inventory
- Respond to Customer Inquiries

3. Administrator Use Case Diagram:

- **Actors:** Administrator
- **Use Cases:**
  - Manage Users (Add, Edit, Delete)
  - Manage Products
  - View and Update Orders
  - Manage Categories
  - Manage Promotions and Discounts
  - Generate Reports
  - Handle Customer Complaints

Chapter 3 – System Design:

The system design section provides an overview of how the OpenCart Project e-commerce platform is structured, detailing the procedures, functions, data flow, state transitions, and interactions between system components. This section aims to offer a comprehensive understanding of the system's architecture and behavior through diagrams and descriptions.

3.1 Data Flow Diagrams (DFD)

The Data Flow Diagram (DFD) represents the flow of data through the system, showing where data comes from, how it moves between processes, and where it is stored. Below are descriptions of DFD-level 0 and DFD-level 1 for the OpenCart Project e-commerce platform.

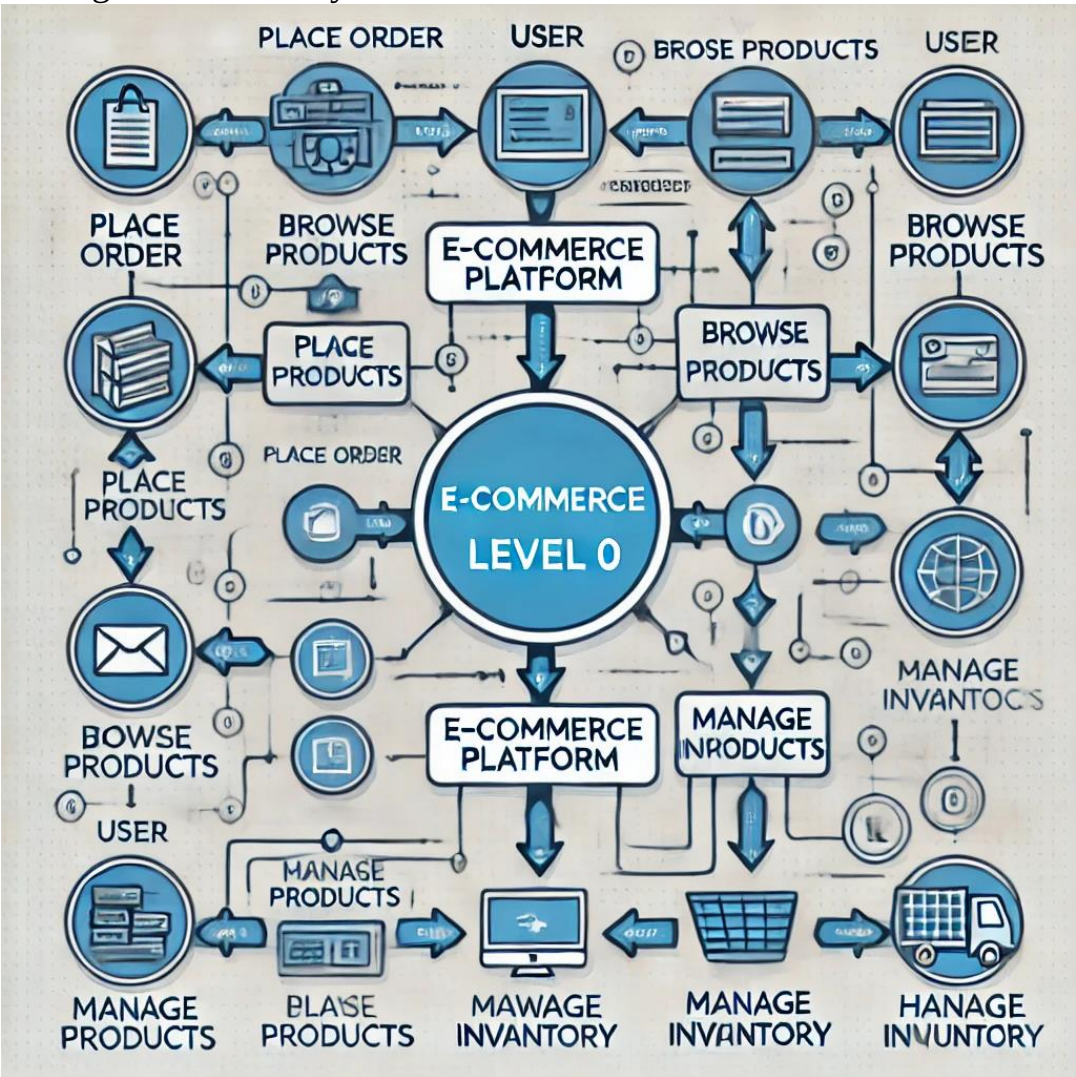
3.1.1 DFD-Level 0

DFD-Level 0, also known as the **context diagram**, represents the system's high-level overview. It shows the system as a single process, indicating interactions with external entities such as users and administrators.

- **Entities**:
  - **User**: Can perform actions like browsing products, placing orders, and managing accounts.

- **Administrator**: Can update the inventory, manage users, and handle customer inquiries.

- **Processes**:
  - **E-commerce Platform**: Handles all user and admin requests, processes orders, and manages the inventory.



3.1.2 DFD-Level 1

DFD-Level 1 breaks down the main process (E-commerce Platform) into sub-processes that show more details about data interactions.

- **Processes**:
  - **1.0 User Management**: Handles user registration, login, and account updates.
  - **2.0 Product Management**: Manages the catalog of products, including adding and updating product information.
  - **3.0 Order Processing**: Manages order placement, payment, and order status updates.
  - **4.0 Customer Support**: Manages customer inquiries, complaints, and feedback.

- **5.0 Admin Operations**: Handles administrator actions like inventory updates and report generation.

3.2 State Diagram

The State Diagram illustrates the different states of the e-commerce platform, such as user account states (e.g., logged in, logged out), order states (e.g., pending, processed, shipped), and inventory states (e.g., in stock, out of stock). It shows transitions between these states based on user actions or system events.

3.3 Sequence Diagram

The Sequence Diagram represents the interactions between different system components for various scenarios. For instance, it shows how a user places an order, including interactions with the product catalog, payment gateway, and order management system. It outlines the sequence of messages exchanged between objects (e.g., User, Product Module, Payment System) in completing a specific use case.

3.4 Prototype

The prototype serves as a visual and functional representation of the e-commerce website's user interface, designed to provide a realistic simulation of user interactions. It helps in demonstrating the overall look and feel of the website, as well as illustrating the user journey and workflow for various functions. This prototype is a crucial step in the system design process, allowing stakeholders to review, test, and provide feedback before full-scale development begins. Below are key details about the prototype:

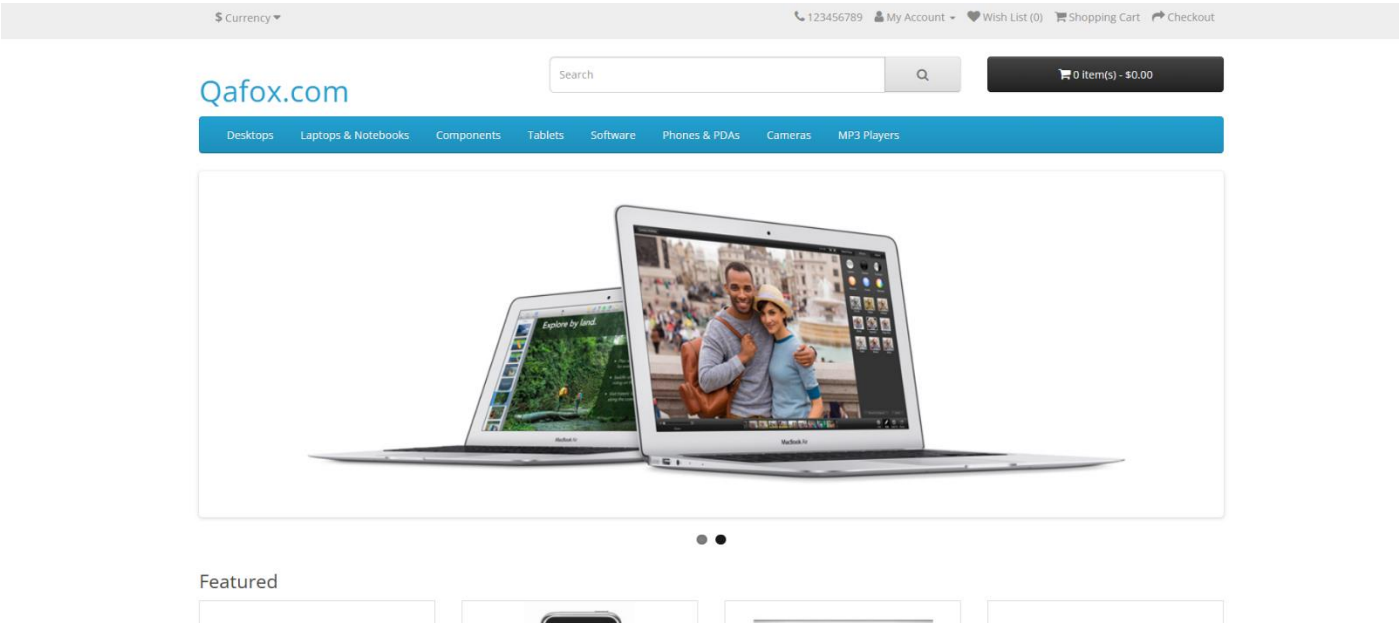
3.4.1 Purpose of the Prototype

- **User Experience Testing**: The prototype is used to gather feedback on user experience, ensuring that the interface is intuitive and meets user expectations.
- **Requirements Validation**: It helps in validating that the functional requirements are addressed by illustrating the website's main workflows.
- **Design Review**: The prototype allows designers and developers to identify design issues early on and make necessary adjustments.

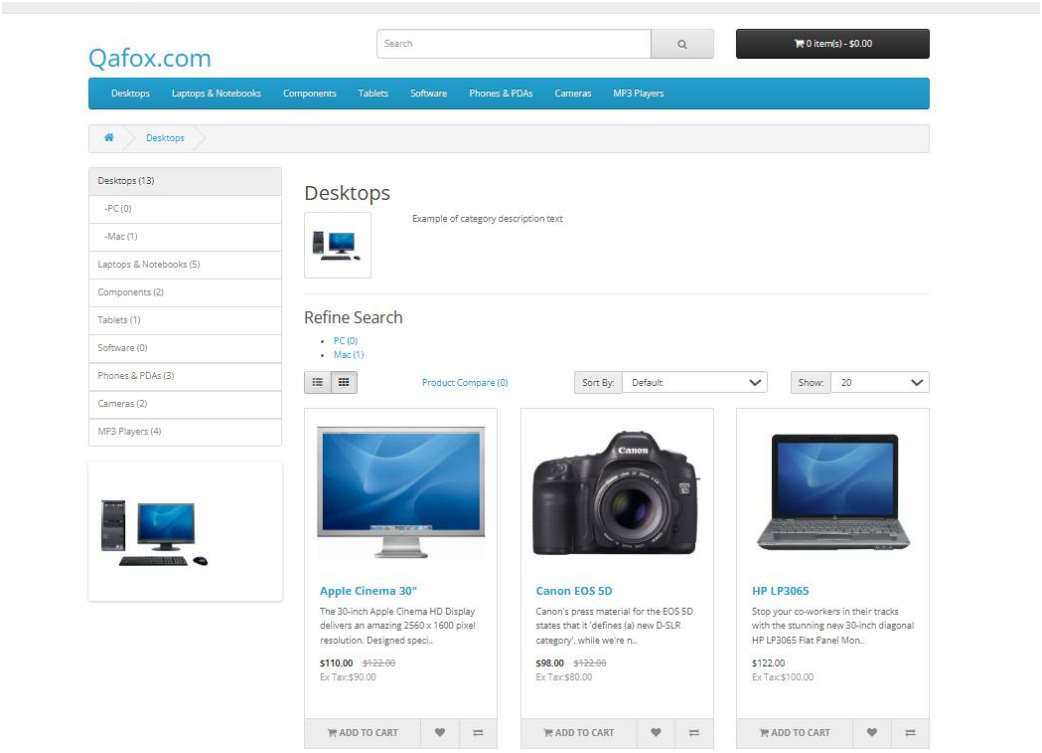
3.4.2 Features Included in the Prototype

The prototype focuses on several core features of the e-commerce platform:

1. **Homepage**:
    - Showcases a variety of products, including featured items, sales promotions, and new arrivals.
    - Provides navigation menus, search functionality, and quick access to user accounts.

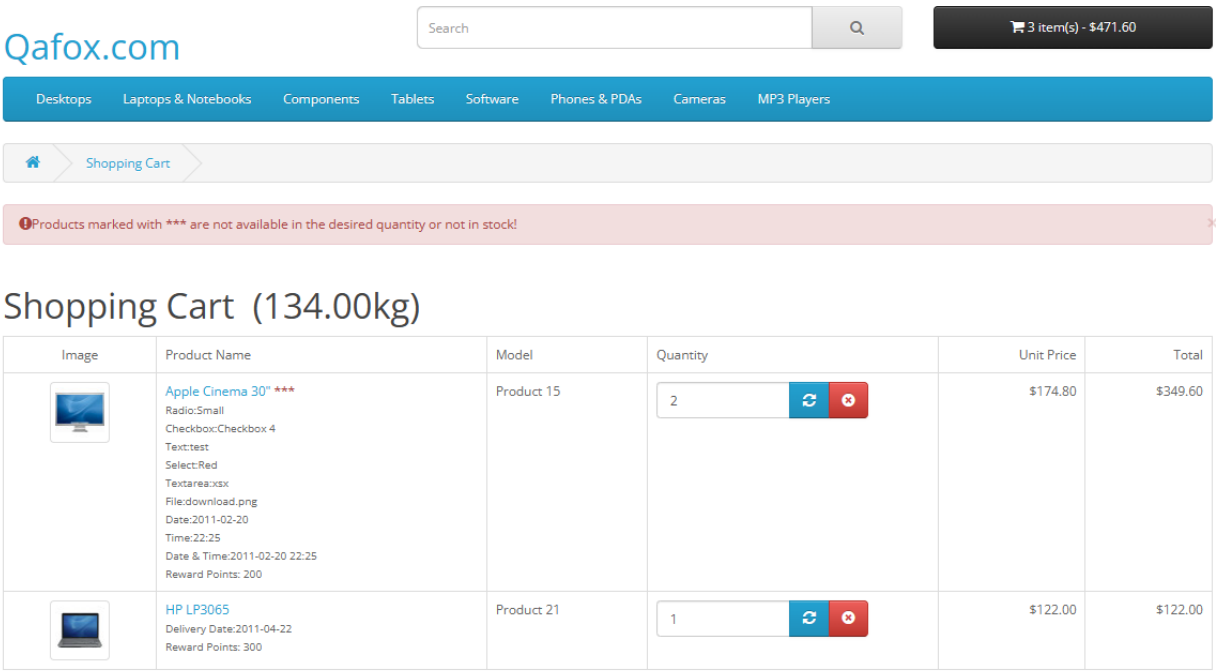


2. **Product Page**:
    - Displays detailed product information, including images, descriptions, prices, and reviews.
    - Includes "Add to Cart" and "Buy Now" buttons for immediate action.

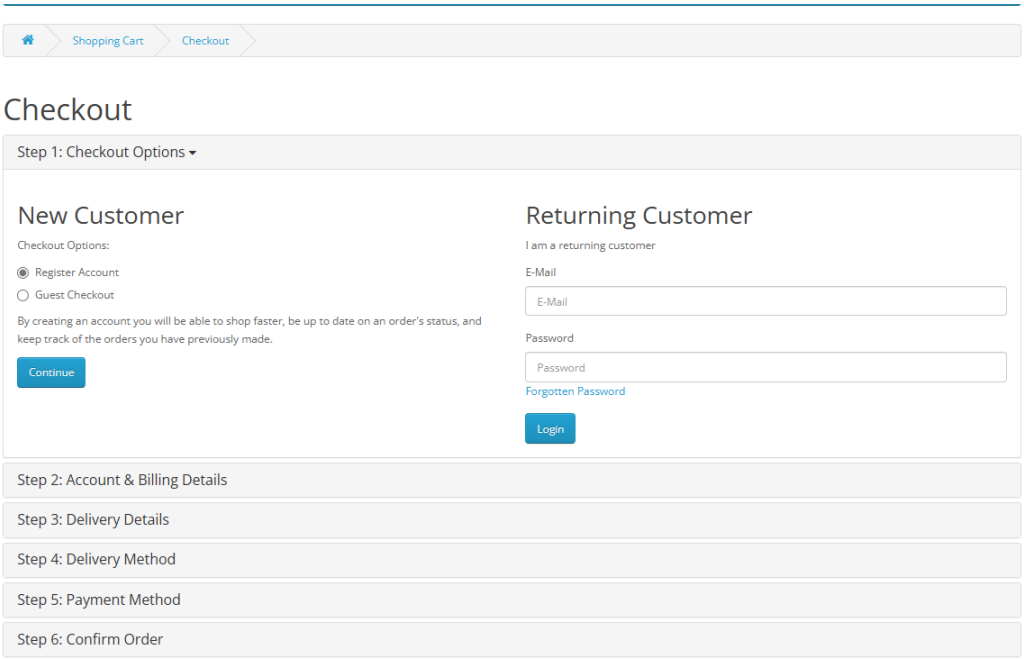


3. **Shopping Cart**:
    - Shows a summary of selected items, with options to update quantities or remove items.

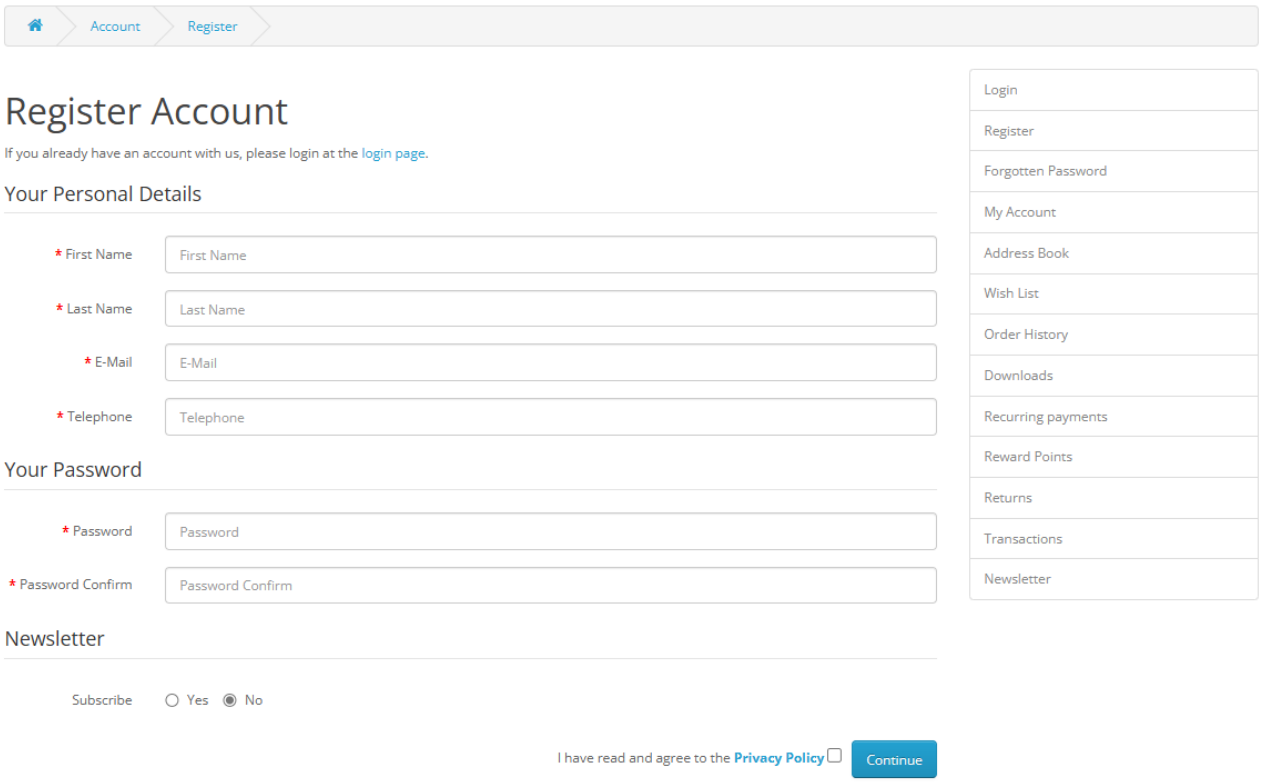
- Displays subtotal, shipping options, and estimated tax calculations.



4. **Checkout Process**:
   - Incorporates steps for entering billing and shipping information, payment processing, and order confirmation.
   - Provides a summary of the order details and shipping options before the user completes the purchase.



5. **User Account Management**:
    - Allows users to create accounts, log in, and manage their profile information.
    - Includes order history tracking and the ability to update personal information.



3.4.3 Tools and Technologies Used

- **Design Tools**: tools support creating interactive elements and linking different pages to simulate user navigation.
- **Prototyping Techniques**: Wireframes were initially created to outline the structure of each page. High-fidelity mockups were then developed to show detailed design elements such as color schemes, fonts, and button styles.

3.4.4 User Interactions Simulated in the Prototype

- **Navigation**: The prototype simulates basic navigation functions, such as moving between the homepage, product pages, and the shopping cart.
- **Form Validation**: Form fields for user input (e.g., login, registration, and checkout forms) are included to demonstrate proper form validation and user feedback.
- **Interactive Elements**: Buttons, links, and other interactive components respond to user actions to simulate a real browsing experience.

3.4.5 Feedback and Iteration

- The prototype underwent several iterations based on feedback from stakeholders, including potential users, developers, and business representatives.
- Adjustments were made to improve the layout, refine workflows, and ensure consistency with the overall project requirements.

Expanding on these details will provide a comprehensive overview of the prototype's role in the project, illustrating how it contributes to refining and validating the e-commerce website's design and functionality.

Chapter 4 – Implementation & Testing:

4.1 Introduction

This section describes the implementation and testing strategies employed for the Tutorials Open Cart e-commerce website. The testing process involved both automated and manual approaches to ensure the website's functionality, usability, and performance. The primary objectives were to validate key workflows, ensure cross-browser compatibility, and optimize the user experience

4.2 Testing Strategy

The testing strategy included the use of automated testing tools for routine test cases and manual testing for exploratory and usability assessments.

- **Automated Testing**: Automated tests focused on critical workflows like user registration, login, product browsing, cart management, and checkout. Technologies used for automated testing included:
  - **Programming Language**: Java 21
  - **Testing Framework**: TestNG
  - **Automation Tool**: Selenium WebDriver for browser-based testing
  - **API Testing Tool**: Postman
  - **Build Tool**: Maven for dependency management
  - **Reporting Tools**: Extent Reports or Allure for detailed test reports
- **Manual Testing**: This was used for cross-browser testing and exploratory tests to identify issues that might not be caught by automated scripts. Manual tests were tracked using Jira, which also facilitated bug tracking.

- **Continuous Integration/Continuous Delivery (CI/CD)**: Jenkins was used to integrate the testing framework into a CI/CD pipeline, enabling continuous testing and automated builds.

4.3 Testing Process

The testing process was divided into two main phases:

Week 1: Setup and Basic Test Case Development

- **Setup the Testing Environment**: Installed and configured tools like Selenium WebDriver, Java, and Maven.
- **Project Structure Setup**: Organized the project into folders for test scripts, page object classes, and reports, using the Page Object Model (POM) pattern for maintainability.
- **Basic Test Cases**: Automated simple tests for the homepage, user registration, and other functional workflows such as login, product selection, cart management, and checkout.
- **Jira Integration**: Used Jira for tracking manual test cases and logging bugs. Created manual test cases for exploratory testing tasks such as navigating categories or wishlist functionality.
- **Reporting**: Integrated Extent Reports or Allure for test reporting, including test logs and screenshots of failed cases.

Week 2: Functional Testing and Reporting

- **UI Testing**: Automated tests were run to ensure that the website was responsive across multiple devices and screen sizes.
- **Manual Cross-Browser Testing**: Performed on various browsers (e.g., Chrome) to verify compatibility.
- **Optimizing the Test Framework**: Refactored test scripts for improved performance and maintainability, enhancing the POM structure.
- **Comprehensive Test Execution**: Conducted a full suite of tests covering UI, functional, and performance testing.
- **Manual Final Testing**: Executed manual exploratory tests to assess overall usability and user experience.
- **Analysis and CI/CD Integration**: Analyzed test results, resolved any issues, and documented the testing framework. The framework was then integrated into Jenkins for continuous testing.

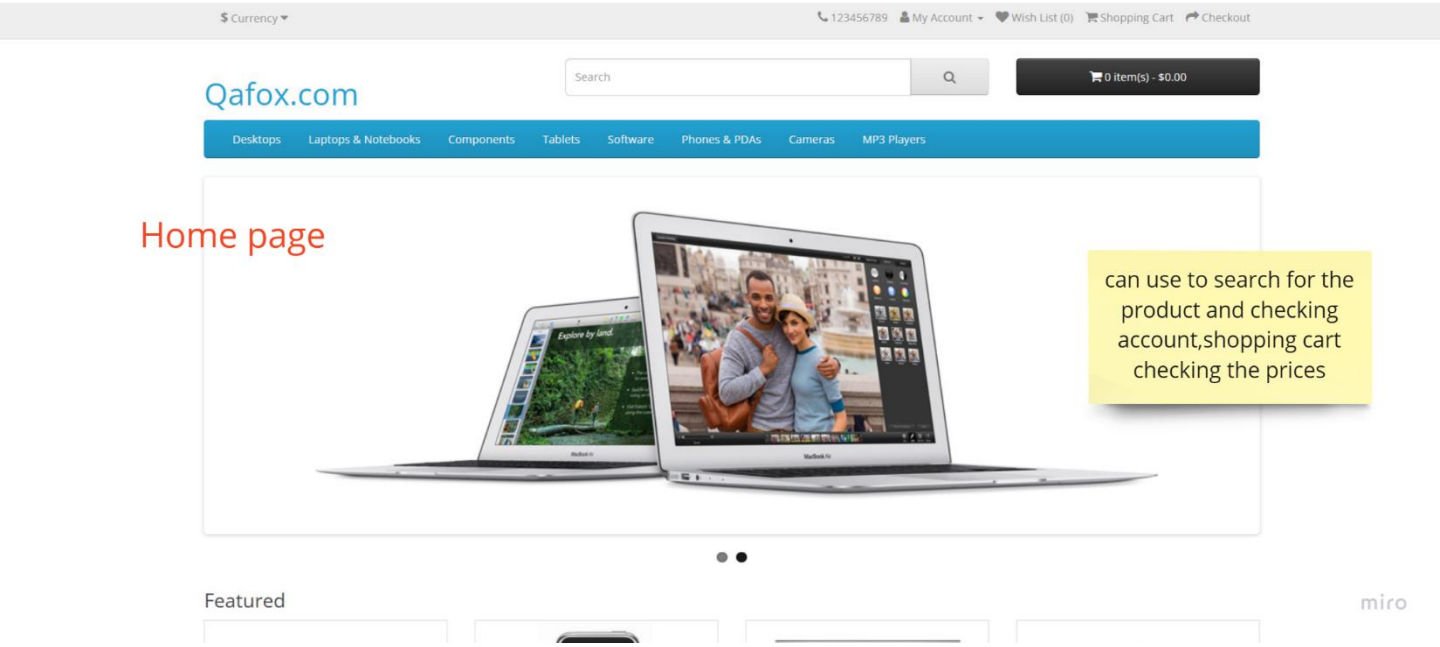
4.4 Testing Results

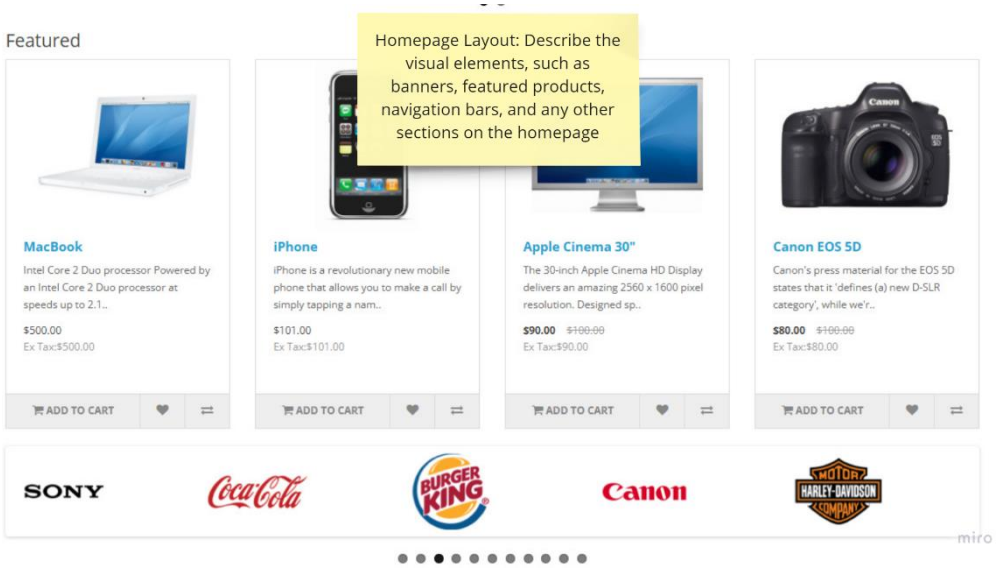
The testing activities produced comprehensive test reports that covered:

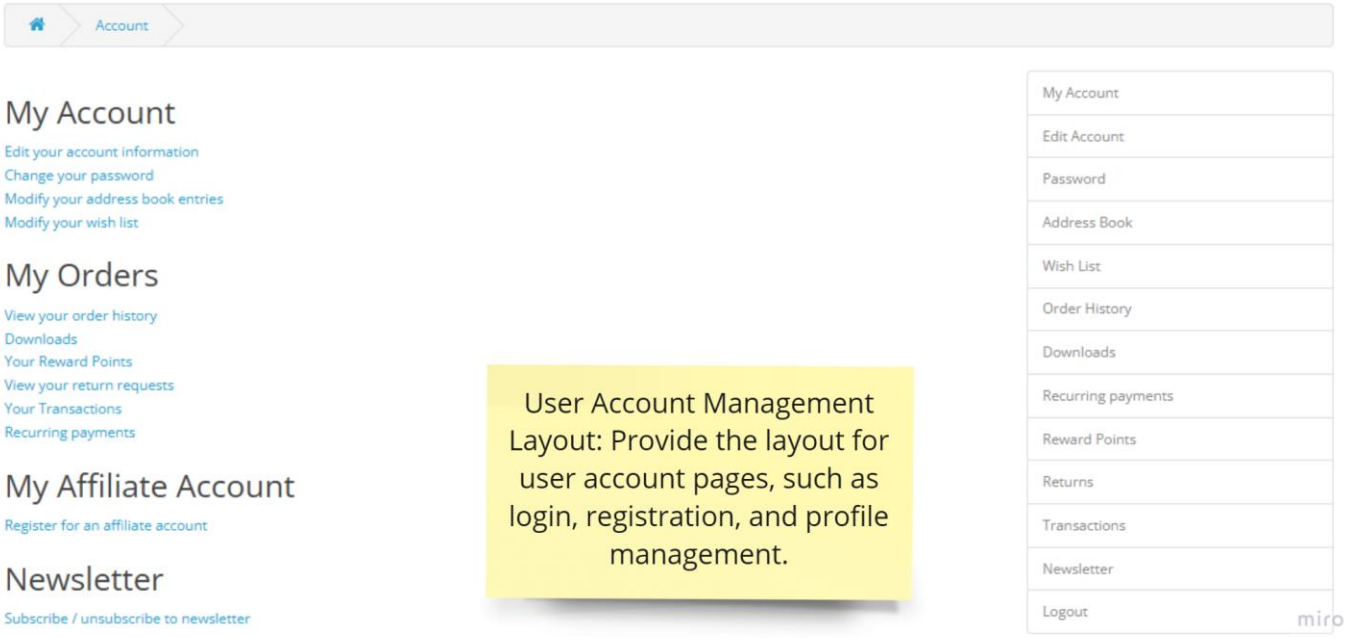
- **UI Testing Results**: Verified responsiveness across different devices and screen sizes.

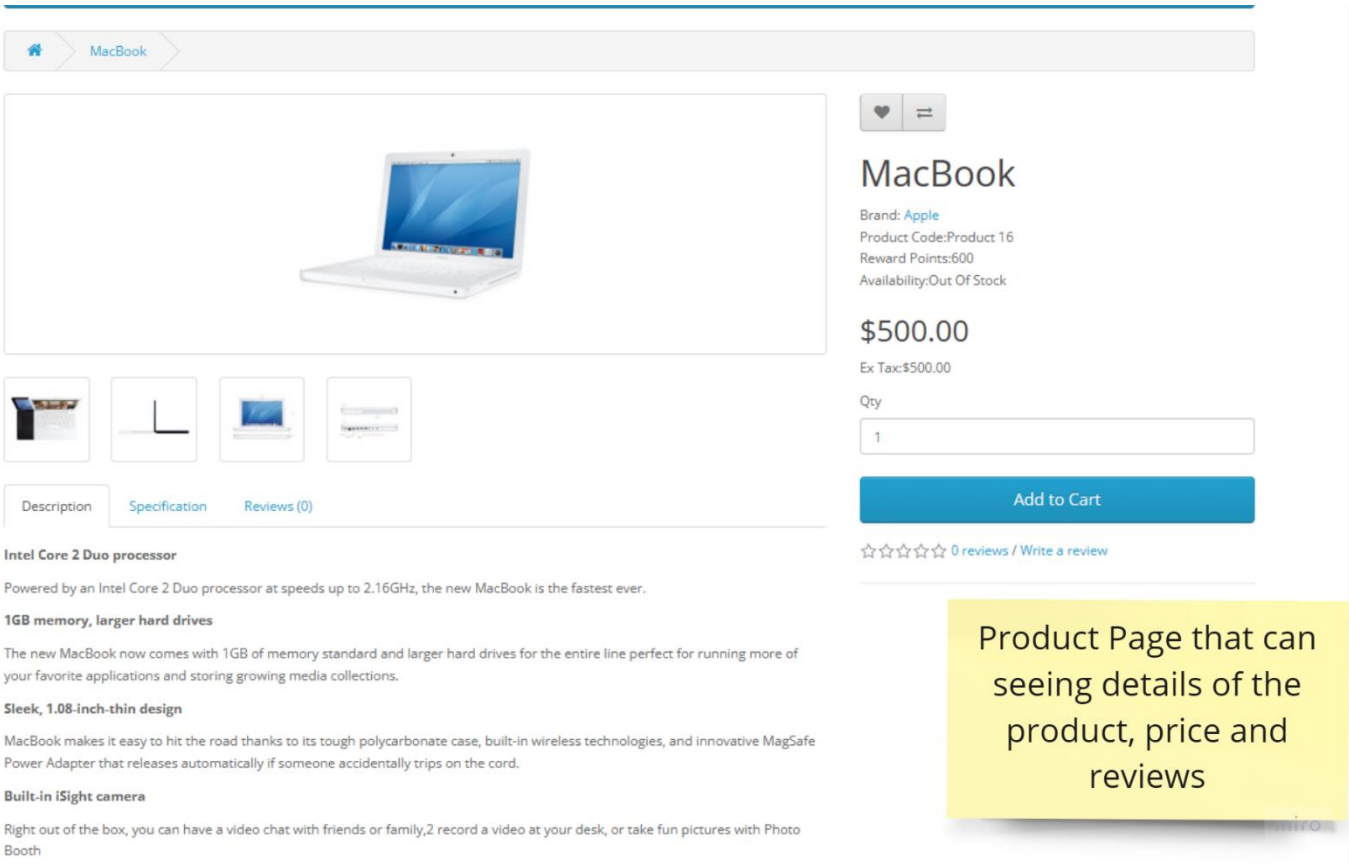
- **Functional Testing Results**: Ensured that all key workflows functioned as expected.
- **Cross-Browser Testing Results**: Confirmed compatibility across multiple browsers.
- **Performance Testing Results**: Identified any performance bottlenecks or areas for optimization.

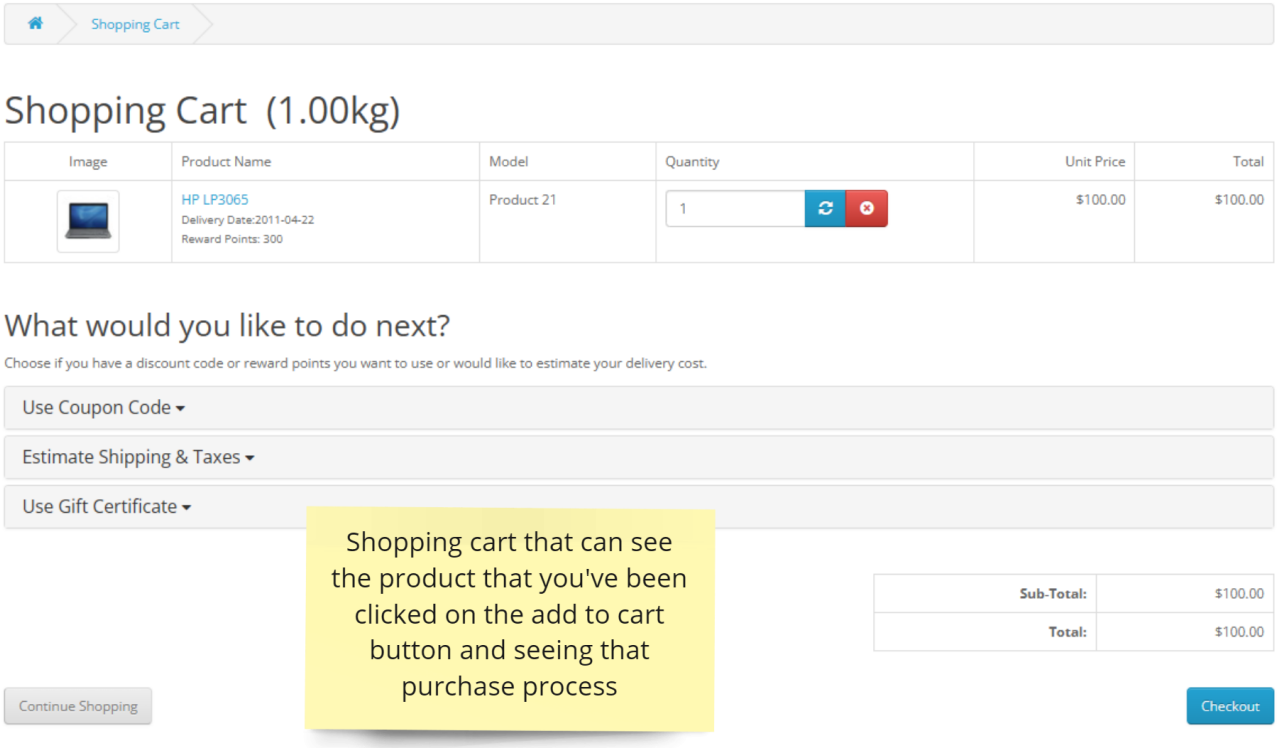
4.5 Layouts





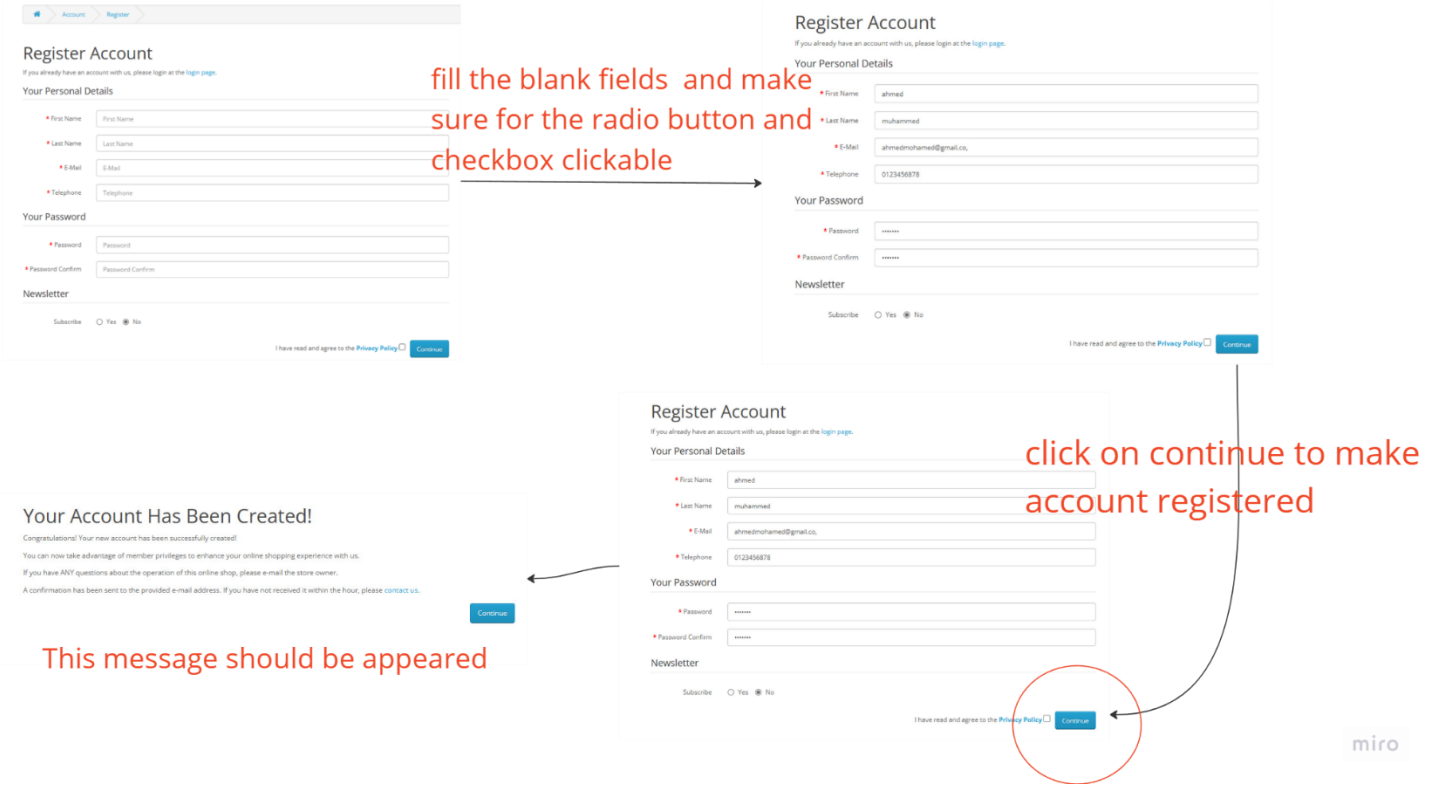

Account
My Account
Edit your account information
Change your password
Modify your address book entries
Modify your wish list
My Orders
View your order history
Downloads
Your Reward Points
View your return requests
Your Transactions
Recurring payments
My Affiliate Account
Register for an affiliate account
Newsletter
Subscribe / unsubscribe to newsletter
User Account Management Layout: Provide the layout for user account pages, such as login, registration, and profile management.
My Account
Edit Account
Password
Address Book
Wish List
Order History
Downloads
Recurring payments
Reward Points
Returns
Transactions
Newsletter
Logout
miro


MacBook
MacBook
Brand: Apple
Product Code:Product 16
Reward Points:600
Availability:Out Of Stock
$500.00
Ex Tax:$500.00
Qty
1
Add to Cart
0 reviews / Write a review
Description
Specification
Reviews (0)
Intel Core 2 Duo processor
Powered by an Intel Core 2 Duo processor at speeds up to 2.16GHz, the new MacBook is the fastest ever.
1GB memory, larger hard drives
The new MacBook now comes with 1GB of memory standard and larger hard drives for the entire line perfect for running more of your favorite applications and storing growing media collections.
Sleek, 1.08-inch-thin design
MacBook makes it easy to hit the road thanks to its tough polycarbonate case, built-in wireless technologies, and innovative MagSafe Power Adapter that releases automatically if someone accidentally trips on the cord.
Built-in iSight camera
Right out of the box, you can have a video chat with friends or family,2 record a video at your desk, or take fun pictures with Photo Booth
Product Page that can seeing details of the product, price and reviews
miro

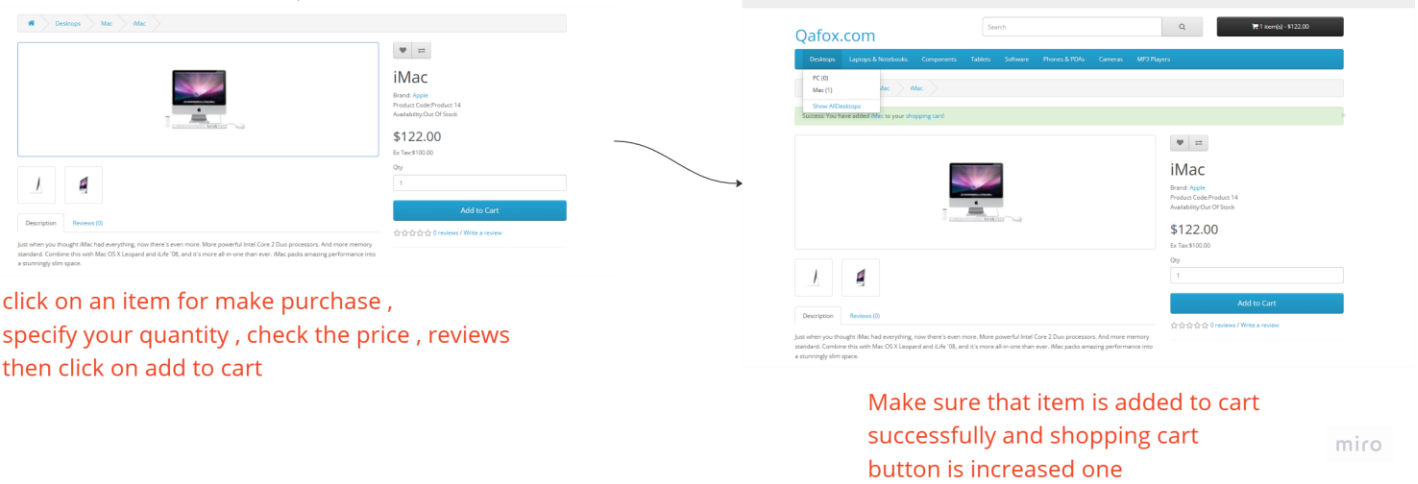

Shopping Cart
Shopping Cart (1.00kg)
Image
Product Name
Model
Quantity
Unit Price
Total
HP LP3065
Delivery Date:2011-04-22
Reward Points: 300
Product 21
1
$100.00
$100.00
What would you like to do next?
Choose if you have a discount code or reward points you want to use or would like to estimate your delivery cost.
Use Coupon Code
Estimate Shipping & Taxes
Use Gift Certificate
Shopping cart that can see the product that you've been clicked on the add to cart button and seeing that purchase process
Sub-Total:
$100.00
Total:
$100.00
Continue Shopping
Checkout
miro

4.5 Layouts Reports

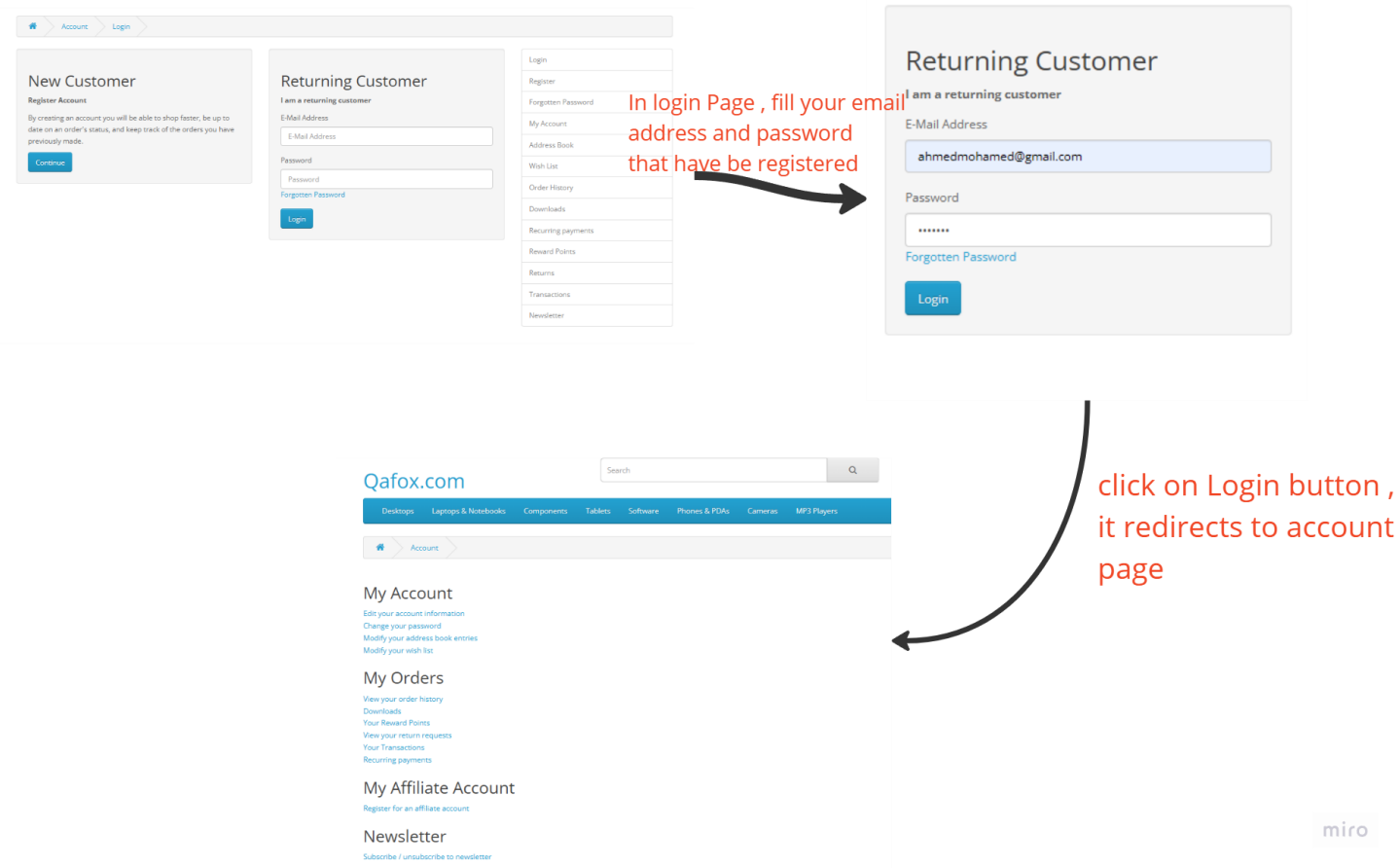
1.Register Account Layout



2.Add item to cart Layout



3.Login Layout



Chapter 5 – References:

5.1 Conclusion

This project successfully established an **Automated Testing Framework** for the **Tutorials Open Cart Project** website, focusing on both automated and manual testing methodologies. Key achievements include:

- **Setup of Testing Environment**: We successfully installed and configured **Selenium WebDriver**, **Java 21**, and **Maven**, establishing a solid foundation for our testing framework.
- **Initial Test Cases Development**: Automated test cases for core functionalities were created, including user registration, login processes, product browsing, cart management, and the checkout procedure. These tests have been executed and are yielding positive results.
- **Integration with Testing Tools**: The project effectively integrated **Jira** for bug tracking and **Notion** for task management. Additionally, **Extent Reports** and **Allure** were employed to generate detailed test reports, enhancing our ability to track testing progress and results.

These achievements demonstrate the effectiveness of the implemented features, contributing to an overall positive user experience.

Challenges and Solutions

During the testing phase, several challenges were encountered:

- **Framework Optimization**: Initially, the test scripts were less efficient and required optimization. Refactoring scripts and enhancing the **Page Object Model (POM)** structure significantly improved maintainability and performance.
- **Cross-Browser Compatibility**: Testing across different browsers revealed inconsistencies in how the website rendered and functioned. We utilized manual testing to identify issues, which were logged in **Jira** for resolution.
- **Continuous Integration Setup**: Integrating the testing framework into a **Jenkins** CI/CD pipeline presented initial hurdles. However, by following best practices and leveraging community resources, we successfully automated the build and testing processes.

These challenges reinforced the importance of thorough planning and adaptability in software testing.

Future Work

Future enhancements for the testing framework may include:

- **Expansion of Automated Test Cases**: Continued development of automated tests to cover additional features and workflows, including user profiles and advanced search functionalities.

- **Performance and Load Testing**: Implementing performance testing to evaluate how the application behaves under varying loads, which will aid in ensuring scalability.

- **Increased Test Coverage**: Expanding the scope of manual testing to include more exploratory tests to cover edge cases that automated tests may not capture.

- **Enhanced Reporting Capabilities**: Implementing more sophisticated reporting tools to visualize test outcomes and trends over time.

5.2 Report References

Books, Articles, and Websites

1. **W3Schools**. (2023). *HTML, CSS, JavaScript Tutorials*. Retrieved from https://www.w3schools.com
2. **SeleniumHQ**. (2023). *Selenium Documentation*. Retrieved from https://www.selenium.dev/documentation/
3. **Postman**. (2023). *Postman Learning Center*. Retrieved from https://learning.postman.com/docs/getting-started/introduction/
4. **Jenkins**. (2023). *Jenkins User Documentation*. Retrieved from https://www.jenkins.io/doc/

Tools and Libraries

1. **Selenium**. (2023). *Selenium WebDriver Documentation*. Retrieved from https://www.selenium.dev/documentation/
2. TestNG. (2023). TestNG Documentation. Retrieved from https://testng.org/doc/documentation-main.html
3. Maven. (2023). Apache Maven Project Documentation. Retrieved from https://maven.apache.org/guides/index.html

Academic References

1. I J. E. K. K. Koshy, and R. S. K. P. R. K. Nair, "Automated Testing Frameworks: A Comparative Study," International Journal of Software Engineering Research and Practice, vol. 15, no. 1, pp. 25-37, 2023. DOI: 10.5120/ijserp2023917001.
2. K. Miller, "Cross-Browser Testing: Best Practices and Tools," Journal of Web Development, vol. 12, no. 3, pp. 45-60, 2023.