Software Engineering - CSPE41

Report

Project: E-commerce website – A simple representation of an online shopping website

Prepared by: **Aaditya Pandey(106123001)**

Purpose of the Document:

E-commerce, or electronic commerce, refers to the buying and selling of goods and services over the internet, facilitating transactions through digital platforms. It has revolutionized retail by offering convenience, a wider product selection, and seamless payment options. This documentation outlines the development and testing of a simple e-commerce website.

The purpose of this document is to provide a comprehensive overview of the project's structure, functionality, and testing methodologies. It details the implemented features—such as product listings, cart management, and user authentication—while also explaining the testing strategies used to ensure reliability and usability. Additionally, this documentation serves as a reference for future enhancements, ensuring scalability and maintainability of the codebase.

In essence, software testing plays a pivotal role in guaranteeing that the e-commerce website fulfills its primary objective by creating a stable and usable platform for customers, thereby enriching their overall shopping experience.

TABLE OF CONTENTS

1) Introduction:

1.1) Purpose:

This documentation serves to comprehensively outline the design, functionality, and testing methodology of our e-commerce website project developed for Software Engineering coursework. The primary objectives are to: (1) document the system architecture and key features including product catalog, shopping cart, and user authentication, (2) establish rigorous testing protocols to validate functionality and usability, (3) provide a reference framework for future maintenance and scalability, (4) serve as a reference for stakeholders. By detailing both development and quality assurance processes, this document ensures alignment with academic requirements while demonstrating professional software engineering practices.

1.2) Project Overview:

Project Summary:

The e-commerce platform is a fully client-side web application built with HTML5, CSS3, and vanilla JavaScript, leveraging browser localStorage for data persistence. Designed as a modular single-page application, it features responsive product displays, real-time cart management with quantity adjustment, and secure credential-based authentication. The solution emphasizes intuitive user workflows - from product selection through checkout - while maintaining session state across page navigations. This responsive web application enables users to:

- Browse products with images and pricing
- Add/remove items and adjust quantities in a cart
- Submit orders with delivery details
- Log in/log out with session management
- Give feedback

Though intentionally limited in scope to frontend implementation, the architecture supports potential integration with backend systems for payment processing and inventory management.

2) Scope:

The scope of this e-commerce website project encompasses comprehensive testing and validation of all core functionalities to ensure a robust, user-friendly online shopping experience. In-scope elements include:

(1) **Functional testing** of product catalog display, shopping cart operations (add/remove items, quantity adjustments), and user authentication flows (login/logout, session persistence)

(2) **Interface validation** covering the UI/UX design for responsiveness across devices, localStorage integration for data persistence, and form handling for checkout processes

(3) **Usability testing** to verify intuitive navigation, accessibility, and adherence to e-commerce best practices.

(4) **Shopping cart system** with local storage persistence.

(5) **Responsive** web design.

Out-of-scope components deliberately excluded from this phase are: (1) Payment gateway integration (e.g., Stripe, PayPal), (2) Backend server or database infrastructure (relying on localStorage as a temporary solution), (3) Advanced features like product search, filters, or recommendation engines, and (4) Multi-user roles (e.g., admin dashboards). Vendor integrations (e.g., third-party APIs for shipping or inventory) and scalability testing for enterprise-level traffic are also deferred to future iterations.

The testing strategy aligns with the project's academic objectives while mirroring industry standards, focusing on delivering a viable product with fully validated core workflows. This phased approach ensures foundational reliability before expanding to complex features, as demonstrated in the referenced documents' structured testing frameworks. Dependencies include stable browser environments and adherence to HTML5/CSS3 specifications, with assumptions that test hardware meets baseline performance requirements.

3)UML Diagrams:

3.1) CONTEXT DIAGRAM

A **context diagram** provides the highest-level overview of the system, illustrating its interactions with external entities (users, services, or devices). The context diagram (Fig 1) establishes the e-commerce system's ecosystem by defining its boundaries and interactions with external entities. At its core, the **Customer** actor initiates key flows: submitting orders, managing account details, and providing feedback.
The **System** interacts with these inputs by processing orders through localStorage (acting as a temporary data store) and generating confirmation messages. Notably absent are payment gateways or admin interfaces, reflecting the project's current frontend-only scope. The diagram uses simple, unidirectional arrows to represent data flows like "Order Details" (customer to system) and "Order Confirmation" (system to customer), avoiding implementation specifics. This high-level abstraction serves as an ideal stakeholder communication tool, clearly delineating what the system does (order processing, cart management) without exposing how these features are implemented internally. Future iterations could extend this diagram to include payment processors or inventory management APIs.

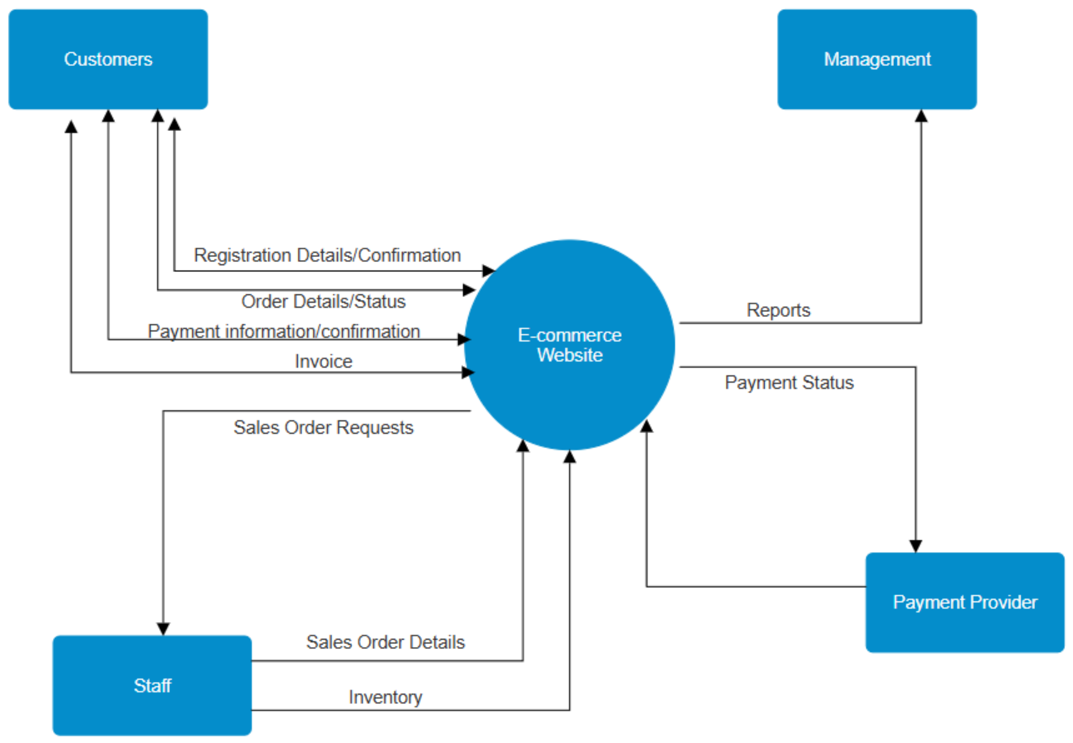


Fig 1: Context Diagram of an e-commerce website

3.2) DATA FLOW DIAGRAM (DFD)

A **data flow diagram (DFD)** breaks down how data moves through the system, detailing internal processes, data stores (like databases), and data pathways. Unlike the context diagram, a DFD reveals step-by-step workflows—such as order processing, cart updates, and payment validation—making it essential for developers to understand logic and for testers to identify critical validation pointsThe Level 1 DFD (Fig 2) decomposes the context diagram's "System" into four key processes:

1. **Product Browsing**: Shows how product data flows from localStorage to the UI, filtered by user queries.
2. **Cart Management**: Details the bidirectional flow between the "Update Cart" process and the "Cart Data" store, including quantity adjustments.
3. **Order Processing**: Illustrates how delivery details merge with cart data to generate orders, with validation checks against user login status.
4. **Authentication**: Depicts credential verification against hardcoded user data (in this prototype).

Data stores are represented by open rectangles:

- **Product Data**: Stores item details (name, price, image URLs).
- **Cart Data**: Persists between sessions via localStorage.
- **User Data**: Manages login credentials (email/password).

External entities (customers) interact only through defined interfaces, maintaining system boundaries. The DFD reveals potential scalability limitations—such as localStorage capacity constraints during high-volume cart operations—while providing developers a blueprint for future backend integration by clearly isolating data handling processes.

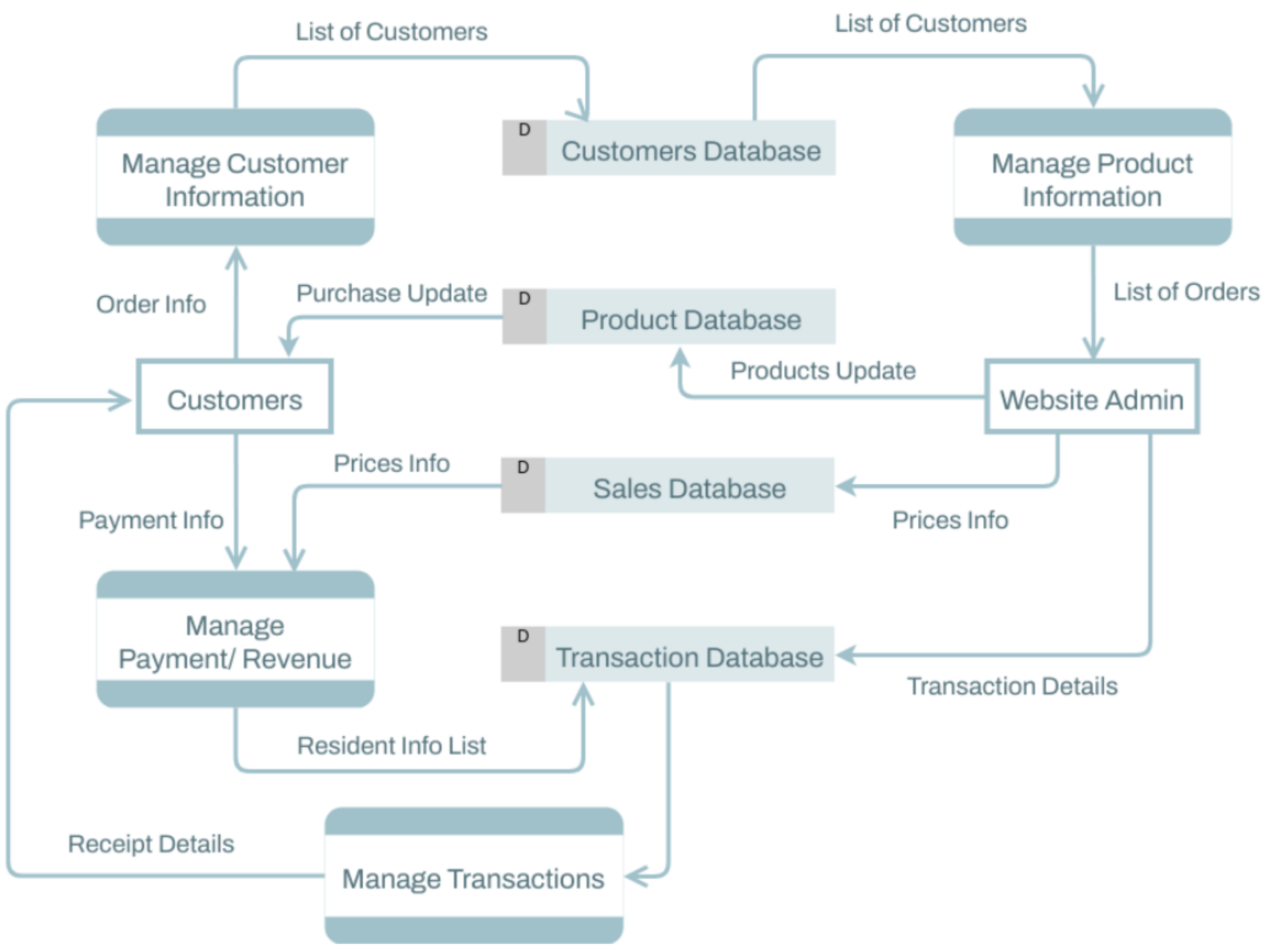
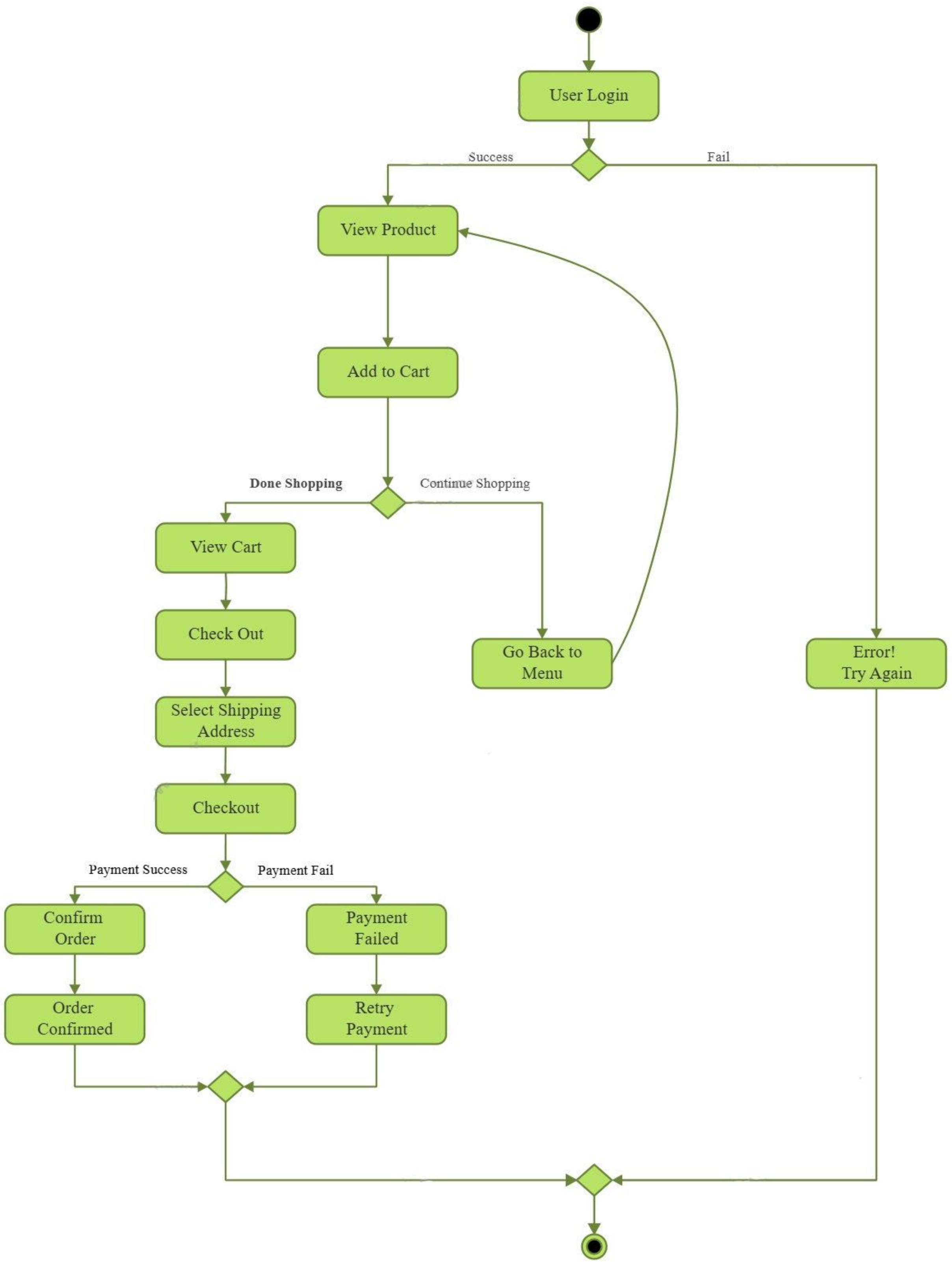


Fig 2: Data Flow Diagram(DFD) of an e-commerce website

3.3) ACTIVITY DIAGRAM

The activity diagram captures the complete customer journey through the e-commerce platform, from initial product discovery to order completion. Beginning with the customer accessing the website, the flow branches into two primary activities: **searching for specific products** using keywords or **browsing** the general catalog. If an item is found, the customer can view detailed product information before deciding to add it to their shopping cart. The diagram highlights iterative actions like adjusting item quantities or continuing to shop, represented by looping flows back to the browsing/searching phases. Critical decision points are clearly marked with diamond symbols, such as checking whether the cart is empty before proceeding to checkout. The process concludes with the checkout sequence, where the customer confirms delivery details and finalizes the purchase. This visualization emphasizes the system's flexibility, allowing users to navigate freely between shopping stages while maintaining a logical progression toward order completion.



4) Testing Strategy

4.1) TEST OBJECTIVES

The primary objectives of our testing strategy focus on validating the e-commerce platform's core functionality while ensuring optimal performance and security. We aim to thoroughly verify all user workflows including product browsing, shopping cart operations, and the checkout process. Special attention will be given to data integrity aspects such as cart persistence across sessions and accurate order processing. Through different testing strategies exploring various scenarios and edge cases we ensure project's requirements are met while maintaining academic rigor, with particular emphasis on validating critical user paths that form the foundation of the shopping experience.

4.2) TEST APPROACH

Our testing methodology employs a comprehensive hybrid approach combining multiple techniques. The black-box testing will examine system functionality from an end-user perspective without considering internal implementation details, focusing on user interface interactions and workflow validation. Simultaneously, white-box testing will analyze the internal code structure, including conditional logic and data flow within key modules like the shopping cart and authentication system. The execution strategy follows an iterative phased approach, beginning with unit testing of individual components, progressing through integration testing of connected modules, and culminating in complete system testing of end-to-end user scenarios. This layered approach ensures thorough validation at every level of the system architecture.

4.3) TEST TYPES

The testing regimen incorporates multiple specialized test types to ensure comprehensive quality assurance. **Functional testing** will validate all specified requirements and user stories, particularly focusing on critical e-commerce features like product catalog display and order processing. **Usability testing** will assess the intuitive nature of the user interface through heuristic evaluation and task-based assessments. Control Structure Testing, particularly **Condition Testing** checks if each if branch executes correctly or not. Each test type employs specific techniques and success criteria tailored to its particular focus area.

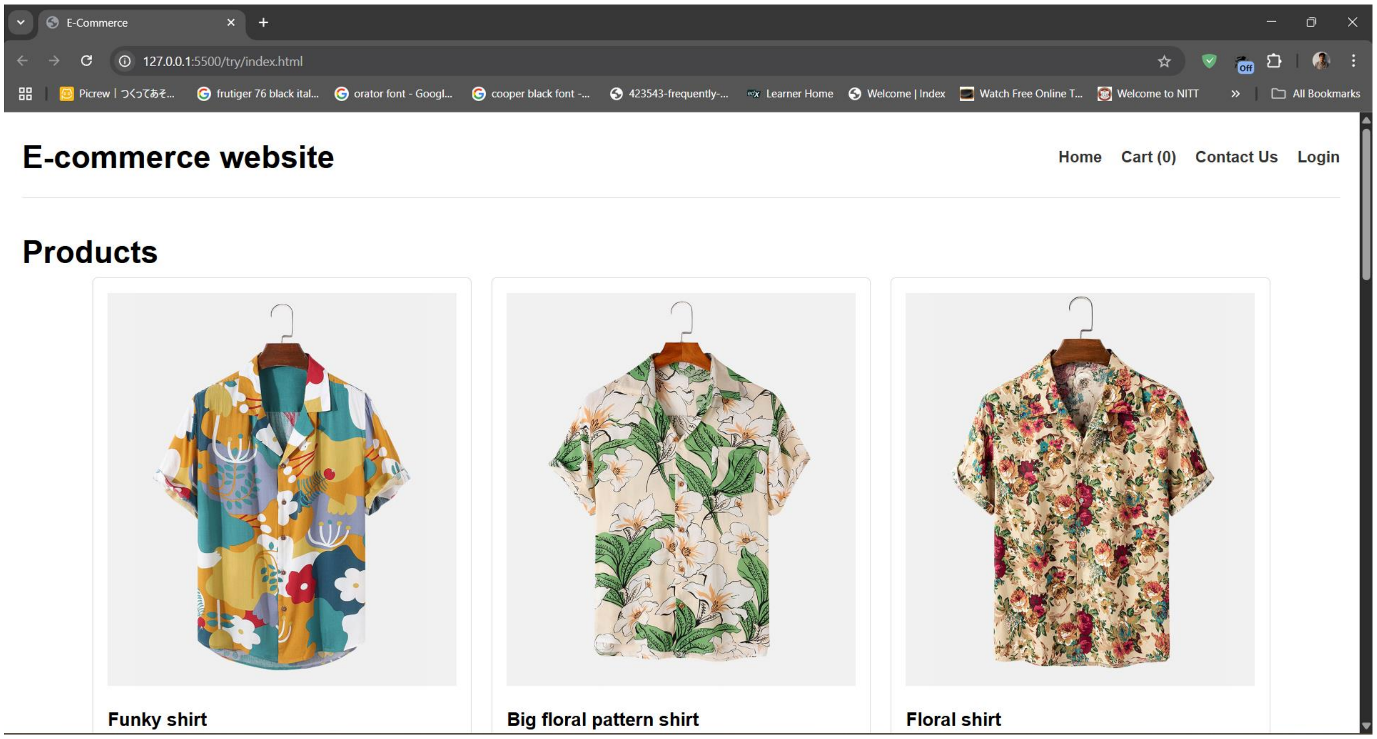
4.4) TEST EXECUTION

The plan is to execute the following testing strategies in the following manner:

1. Levels of testing
   a. Unit testing:
      Checking individual modules like just the add to cart feature, just the login page, etc.
   b. Integration testing:
      Checking how modules that are supposed to work together interact like checkout + authentication
   c. System testing:
      Checking if the whole system functions properly or not in a full scenario
2. Technique based testing
   a. Black box testing
      Testing functionality of modules without code knowledge.
      Validate the UI against requirements.
   b. White box testing
      Testing internal logic/structure of the code
      Verify all the conditionals and loops.
   c. Condition testing
      Validate all the logical paths of a conditional statement work properly(if/else if/else)
3. Feature specific testing
   a. Functional testing
      Verify specific features again their requirements like checkout form validation, adding to cart.
   b. Usability testing
      Evaluate user experience/intuitiveness of the interface for the users.

5) Testing:

5.1) TEST PREVIEW



E-commerce website

Home Cart (0) Contact Us Login

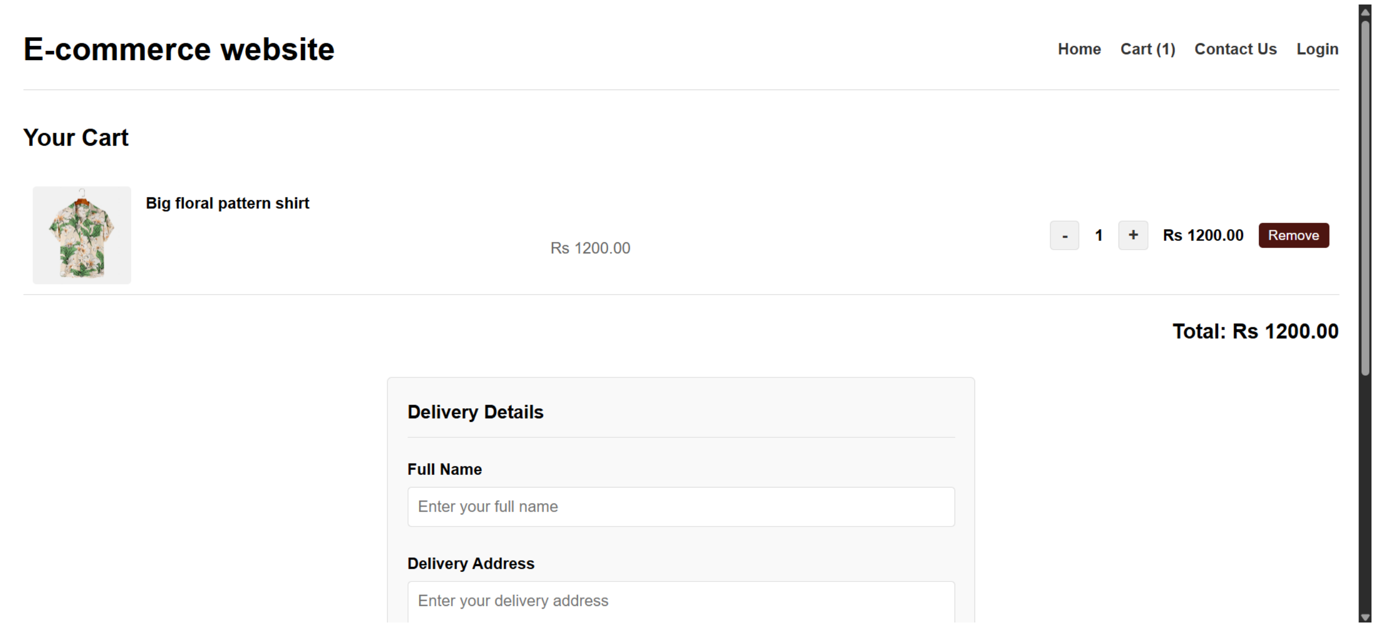
Products

Funky shirt

Big floral pattern shirt

Floral shirt

Fig 3: Home page



E-commerce website

Home Cart (1) Contact Us Login

Your Cart

Big floral pattern shirt

Rs 1200.00

- 1 + Rs 1200.00 Remove

Total: Rs 1200.00

Delivery Details

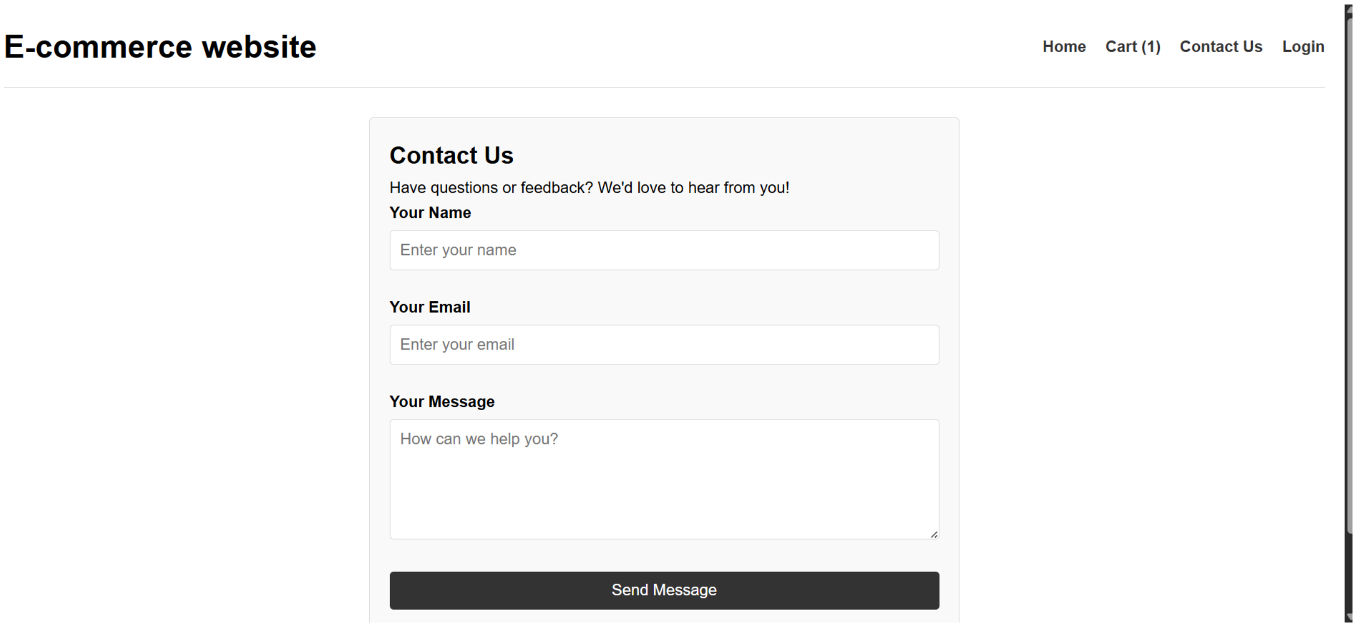
Full Name

Enter your full name

Delivery Address

Enter your delivery address

Fig 4: Cart + Delivery details page


E-commerce website

Home Cart (1) Contact Us Login

Contact Us

Have questions or feedback? We'd love to hear from you!

Your Name

Enter your name

Your Email

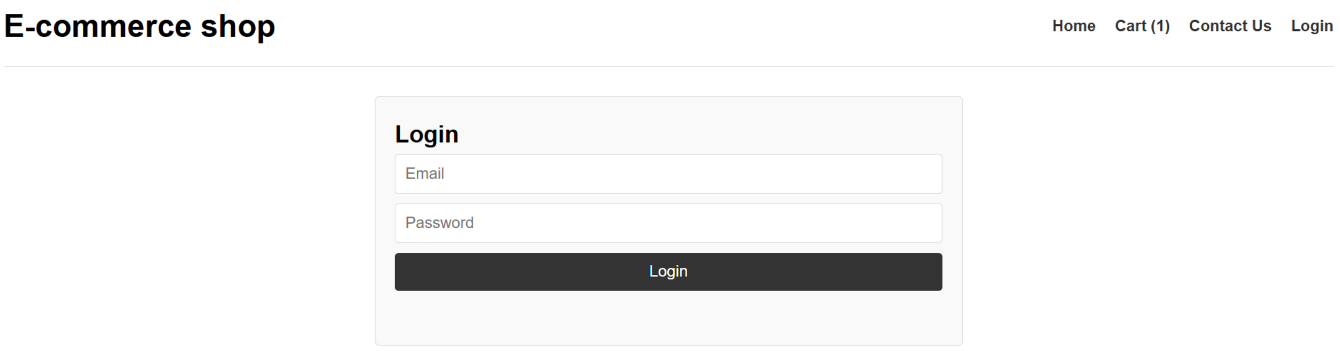
Enter your email

Your Message

How can we help you?

Send Message

Fig 5: Contact Us page


E-commerce shop

Home Cart (1) Contact Us Login

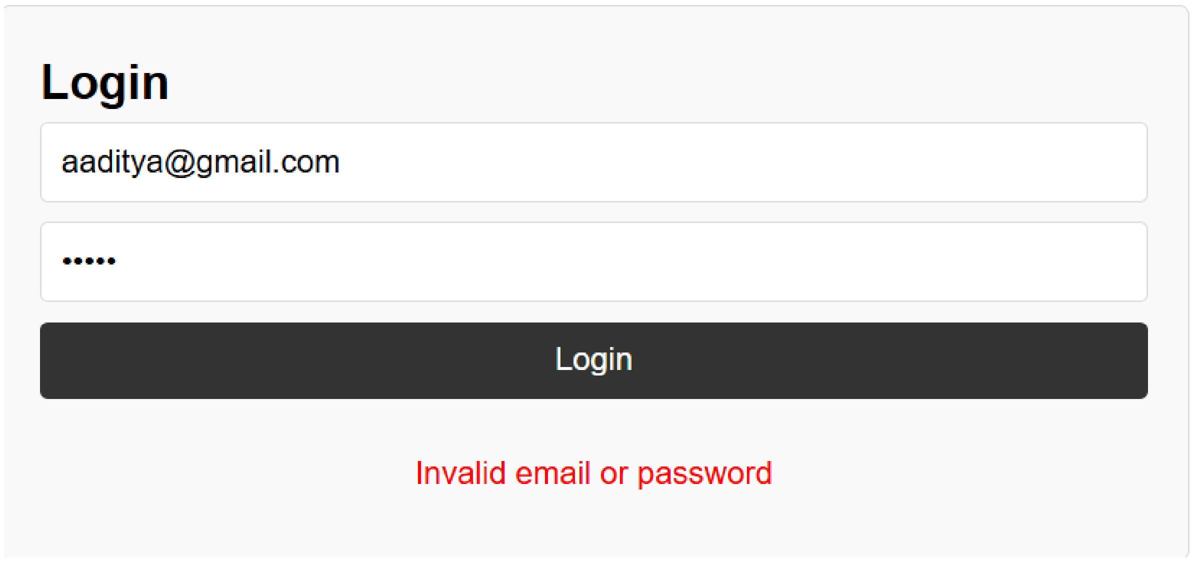
Login

Email

Password

Login

Fig 6: Login page view


Login

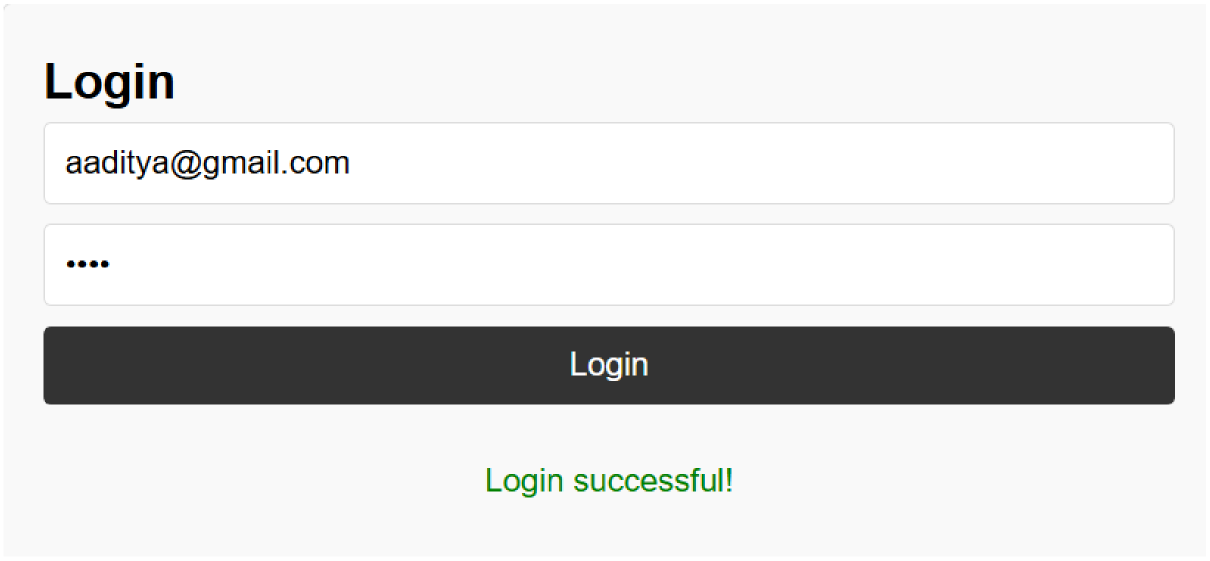
aaditya@gmail.com

•••••

Login

Invalid email or password

Fig 7: Login page if incorrect credentials entered



Login

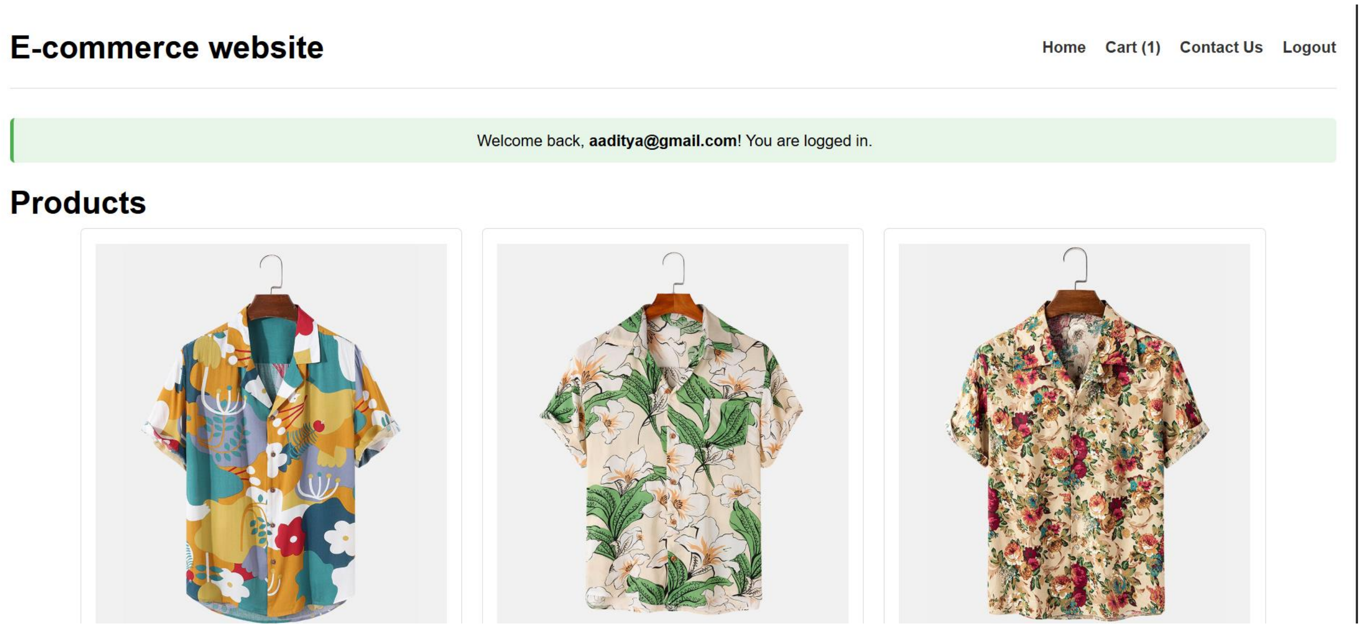
aaditya@gmail.com

••••

Login

Login successful!

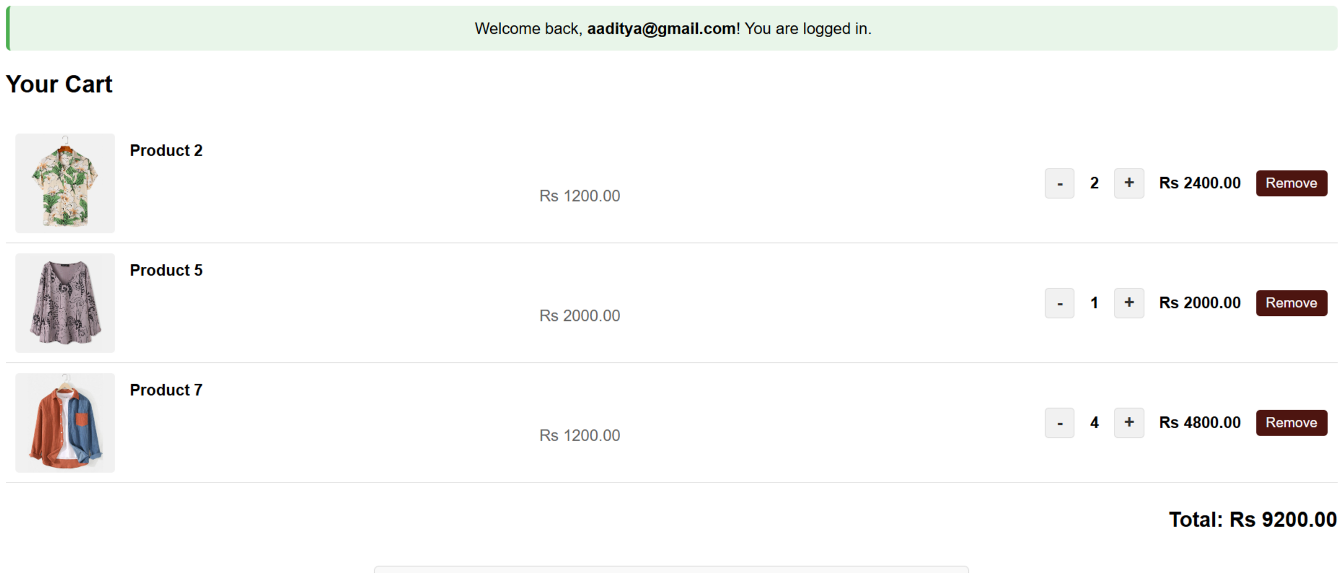
Fig 8: Login page if correct credentials entered



Welcome back, **aaditya@gmail.com**! You are logged in.

Products

Fig 9: Home page after login

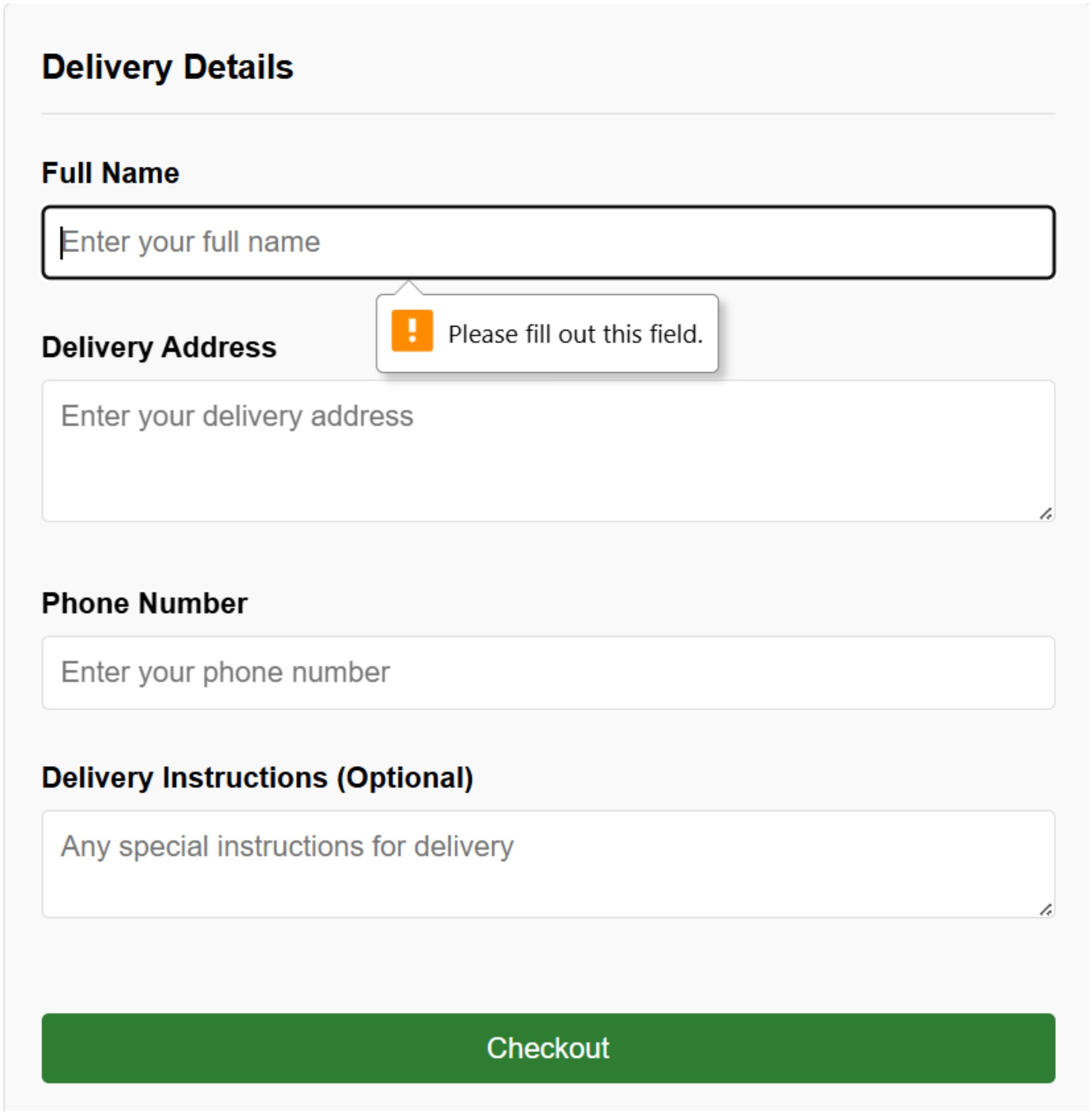

Welcome back, **aaditya@gmail.com**! You are logged in.

Your Cart

Product 2	Rs 1200.00	- 2 +	**Rs 2400.00**	Remove
Product 5	Rs 2000.00	- 1 +	**Rs 2000.00**	Remove
Product 7	Rs 1200.00	- 4 +	**Rs 4800.00**	Remove

Total: Rs 9200.00

Fig 10: Cart after adding multiple items in multiple quantities


Delivery Details

Full Name

Enter your full name

Please fill out this field.

Delivery Address

Enter your delivery address

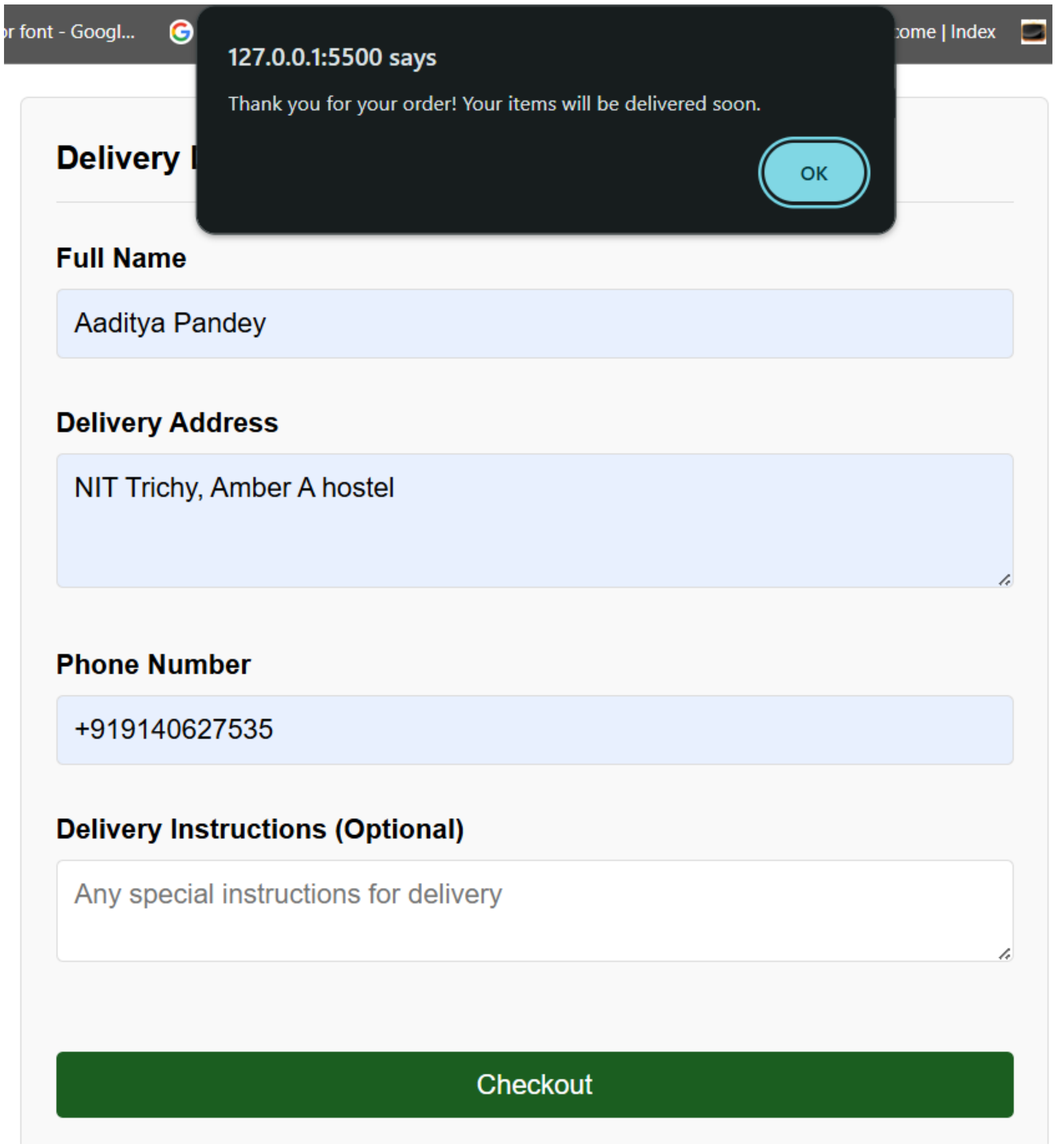
Phone Number

Enter your phone number

Delivery Instructions (Optional)

Any special instructions for delivery

Fig 11: Delivery Details entering



127.0.0.1:5500 says

Thank you for your order! Your items will be delivered soon.

Delivery

Full Name

Aaditya Pandey

Delivery Address

NIT Trichy, Amber A hostel

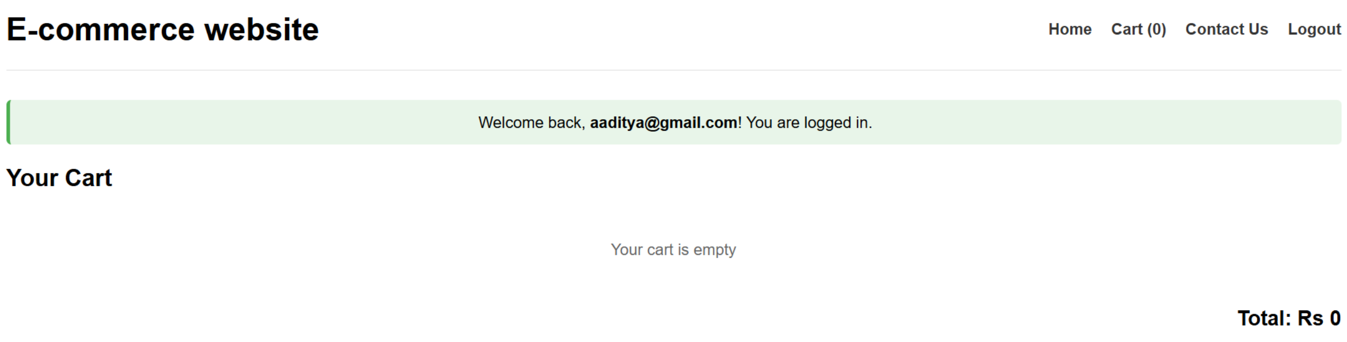
Phone Number

+919140627535

Delivery Instructions (Optional)

Any special instructions for delivery

Fig 12: After entering delivery details and checking out


E-commerce website

Home Cart (0) Contact Us Logout

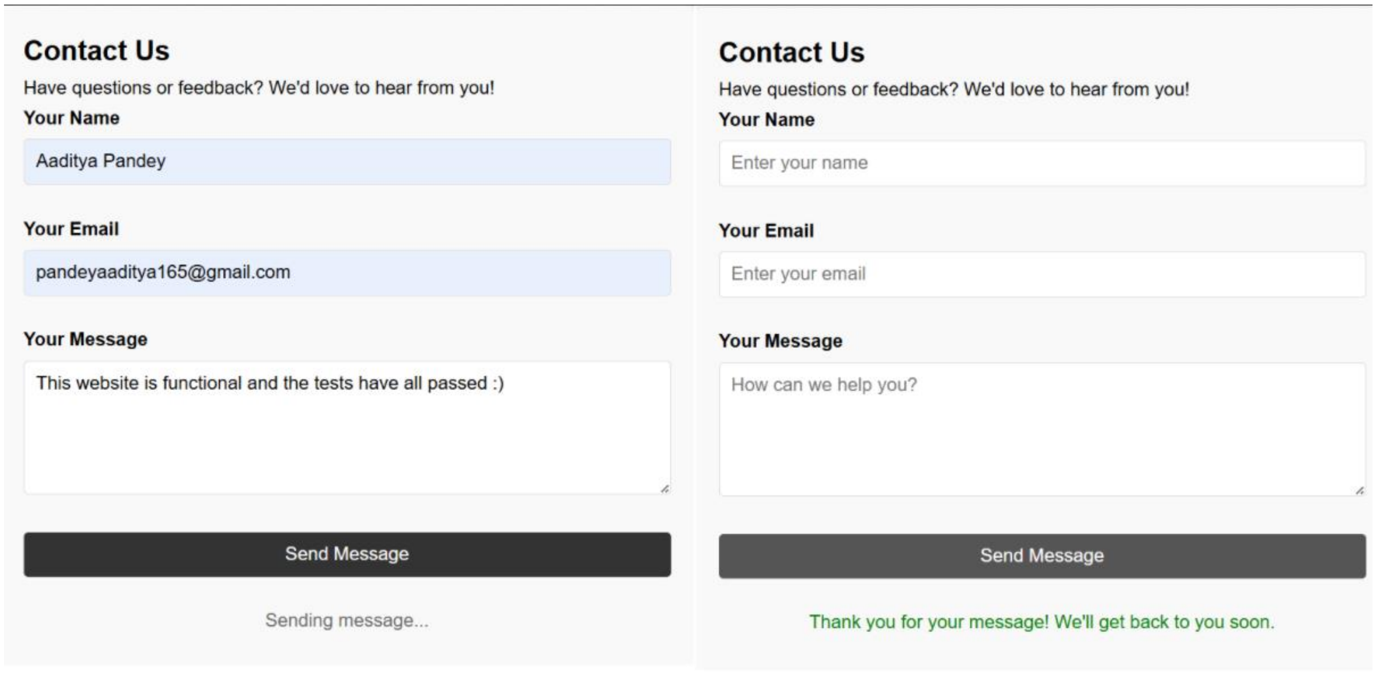
Welcome back, **aaditya@gmail.com**! You are logged in.

Your Cart

Your cart is empty

Total: Rs 0

Fig 13: Cart after ordering items


Contact Us

Have questions or feedback? We'd love to hear from you!

Your Name: Aaditya Pandey

Your Email: pandeyaaditya165@gmail.com

Your Message: This website is functional and the tests have all passed :)

Send Message

Sending message...

Contact Us

Have questions or feedback? We'd love to hear from you!

Your Name: Enter your name

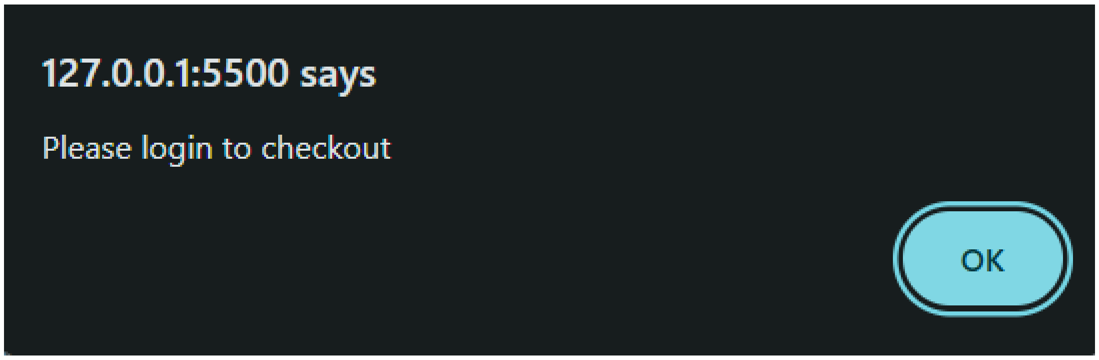
Your Email: Enter your email

Your Message: How can we help you?

Send Message

Thank you for your message! We'll get back to you soon.

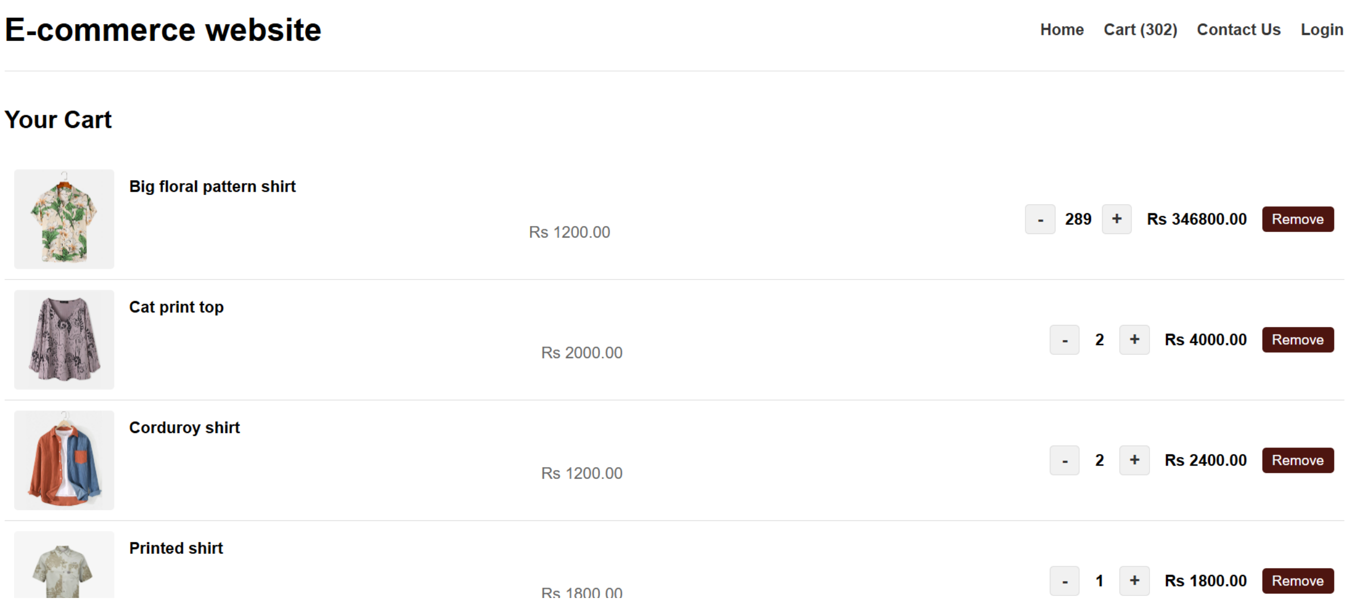
Fig 14: Sending a feedback


127.0.0.1:5500 says

Please login to checkout

OK

Fig 15: Prompt that appears when you try to checkout without logging in


E-commerce website

Home Cart (302) Contact Us Login

Your Cart

Big floral pattern shirt	Rs 1200.00	289	Rs 346800.00	Remove
Cat print top	Rs 2000.00	2	Rs 4000.00	Remove
Corduroy shirt	Rs 1200.00	2	Rs 2400.00	Remove
Printed shirt	Rs 1800.00	1	Rs 1800.00	Remove

Fig 16: Cart performance tested at high load

5.2) OBSERVATIONS:

- Here all the individual units/modules (fig 3-6) open properly and don't seem to have any visual errors or any errors while loading/rendering. Therefore, **Unit Testing** is passed.
- Modules that are supposed to work together (eg. Fig 4: adding an item to the cart and it appearing correctly along with correct price and default quantity i.e. 1 which is also displayed in the top right as Cart(1)) function smoothly and correctly. Therefore, I**ntegration Testing** has passed.
- Adding item to cart, adjusting quantity as per user's wish, entering their delivery address/details, logging in, and checking out is a sequence of functions (Fig 10->11->8->12->13) which executes properly and smoothly. Sending feedback is yet another feature with full functionality. **System Testing** has passed as the whole system is functional.
- The User Interface of the website is checked in all the scenarios against user's requirements without focusing too much on the code and it is observed that the added features match the standards. **Black Box** testing.
- The internal structure of the code is assessed for coherent logic that is readable/understandable to any developer working on it and can be verified by anyone with programming knowledge. The code works properly as intended so **White Box testing** is passed.

- Our code has conditional statements that have 2 (or more) paths that could be taken. For eg. in the login page there are two paths- correct credentials and incorrect credentials (fig 7 and 8). In the cart when we try to checkout without entering delivery details, an error message prompt appears (Fig 11) or if we fill delivery details but haven't logged in then another message appears asking us to login (Fig 15). If a product has been already added to the cart, then the + and – symbol increment or decrement the quantity. With this all the major conditionals and their correctness is verified. **Condition Testing** is verified.
- Through the multiple tests we have successfully verified that specific features are functional and don't seem to encounter any malfunctions. Cases like checking if the quantity of an item is reduced from 1 will the item get removed from cart is also checked. The cart is also put under high load (Fig 16) to check if it still performs well which it does. This concludes the **Functional Testing.**
- The user interface is very simple and easily understandable by any user with any level of technical/computer knowledge with buttons clearly labelled and all within eye's reach. No buttons/features are hidden making the interface very intuitive and easy to use. This concludes **Usability Testing.**

6) Conclusion

This documentation has comprehensively outlined the development and rigorous testing of our e-commerce website, demonstrating its adherence to software engineering principles and academic requirements. Through structured methodologies—including **unit**, **integration**, and **system testing**—we validated core functionalities such as product browsing, cart management, and user authentication. Techniques like **black-box** and **white-box testing** ensured both user-facing reliability and robust internal logic, while **condition testing** verified all critical decision paths. The platform's intuitive interface, confirmed via **usability testing**, delivers a seamless shopping experience. By leveraging **localStorage** for data persistence and maintaining a responsive design, the project achieves its goals while laying a foundation for future enhancements like payment integration or backend expansion. This end-to-end validation process underscores the project's success in merging academic rigor with practical e-commerce standards.

Project Link: https://github.com/Aaditya165/Software-Engineering-Project.git