

UCS503- Software Engineering Lab

E-COMMERCE WEBSITE

UCS503 Software Engineering Project Report

Mid-Semester Evaluation

Submitted by:

(102103508) Dheeraj Arora

BE (COE) Summer Course

Team Id: G

Submitted to:

Dr. Harkiran Kaur

(Assistant Professor)



**THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)**

Computer Science and Engineering Department

TIET, Patiala June-July 2024

TABLE OF CONTENTS

S.No.	Assignment	Page No.
1.	Project Write Up	3
2	Planning Phase	5
2.1	i. Scope Estimation - Function Point Count Calculation	5
2.2	ii. Effort Estimation – COCOMO	6
2.3	iii. Schedule Estimation - Gantt Chart	7
2.4	iv. Critical Path Method	9
3.	Analysis Phase	10
3.1	i. Use Case Template	10
3.2	ii. Use-Case Diagram	22
3.3	iii. Swimlane Diagrams	23
3.4	iv. Data Flow Diagrams - Level 0, Level 1	23
3.5	v. Data Dictionary Entries for Data Stores, Processes and Entities	24
3.6	vi. Software Requirement Specification (SRS) in IEEE Format	29
4.	Design Phase	41
4.1	i. Module Structure Chart	41
4.2	ii. Class Diagram	41
4.3	iii. Database Design - ER Diagram	42
4.4	iv. Sequence Diagram	42
5.	Coding Phase	43
5.1	Working Project Screenshots	43
6	Testing Phase	44
6.1	i. Cyclomatic Complexity Calculation	44
6.2	ii. Test Case Template	46

1. Project Write-Up:

Need

The need for this project arises from the increasing demand for e-commerce platforms that offer seamless, intuitive, and responsive user experiences. This project serves as a learning tool for developers to understand the fundamental functionalities of an e-commerce website, utilizing the MERN stack. By simulating the entire e-commerce journey, it provides practical hands-on experience in building and managing an online marketplace, preparing developers for real-world projects.

Functional Requirements

1. User Authentication

- Sign up, log in, log out, and manage account details.
- Admin login for product and order management.

2. Product Management

- Admins add, update, and delete products with details (name, description, price, category, stock, images).

3. Product Browsing and Searching

- View, filter by categories, and search for products.

4. Product Details

- View detailed product information.

5. Shopping Cart

- Add, view, modify, and remove products in the cart.

6. Checkout Process

- Enter shipping information, review, and confirm orders.

7. Order Management

- Admins view and update order status.
- Users view order history and details.

Non-Functional Requirements

1. Performance

- Load time within 3 seconds, support for 100 concurrent users.

2. Scalability

- Accommodate future growth in products and users.

3. Usability

- Intuitive, responsive design for various devices.

4. Security

- Encrypted passwords, protection against SQL injection, XSS, CSRF.

5. Reliability

- 99.9% uptime, regular data backups.

2. Planning Phase

2.1 Scope Estimation - Function Point Count Calculation

Function point = (Unadjusted function point count) * (Complexity adjustment factor)

— Unadjusted function point count calculation:

1) No. Of external inputs = 9 (6 average, 3 low):

- Average: $6 * 4 = 24$
- Low: $3 * 3 = 9$

2) No. Of external outputs = 12 (8 average, 4 low):

- Average: $8 * 5 = 40$
- Low: $4 * 4 = 16$

3) No. Of external inquiries = 2 (2 low):

- Low: $2 * 3 = 6$

4) No. Of external logical files = 3 (1 average, 2 low):

- Average: $1 * 10 = 10$
- Low: $2 * 7 = 14$

5) No. Of external interface files = 6 (4 average, 2 low):

- Average: $4 * 7 = 28$
- Low: $2 * 5 = 10$

$$\text{UFPC} = 24 + 9 + 40 + 16 + 6 + 10 + 14 + 28 + 10 = 157$$

— Complexity adjustment factor calculation:

Table 1: CAF calculating factors

S.no.	Factors	Output
1	Does the system require reliable backup and recovery?	5
2	Is data communication required?	5
3	Are there distributed processing functions?	4
4	Is performance critical?	5
5	Will the system run in an existing heavily utilized operational environment?	3
6	Does the system require online data entry?	5
7	Does the online data entry require the input transaction to be built over multiple screens or operations?	4
8	Are the master files updated online?	5
9	Are the inputs, outputs, files, or inquiries complex?	4
10	Is the internal processing complex?	4
11	Is the code designed to be reusable?	3
12	Are conversion and installation included in the design?	3
13	Is the system designed for multiple installations in different organizations?	2
14	Is the application designed to facilitate change and ease of use by the user?	5
	TOTAL	57

$$CAF = 0.65 + (0.01 * 57) = 1.22$$

$$\text{Function point} = 157 * 1.22 = \mathbf{191.54}$$

2.2 Effort Estimation – COCOMO

Language factor for Java Script = 47

$$KLOC = (191.54 * 47) / 1000 = 9 \text{ KLOC (Organic: } a=2.4; b=1.05; c=2.5; d=0.38)$$

$$\text{Effort} = a(KLOC^b) = 2.4 (9 ^ 1.05) = 24.10 \text{ PM}$$

$$\text{Duration} = c(\text{Effort}^d) = 2.5 (24.10 ^ 0.38) = 8.37 \text{ Months}$$

Average Staff Size = Effort/Duration = 2.87 Persons

$$\text{Productivity} = \text{KLOC/Effort} = 9/24.10 = 0.37 \text{ KLOC/PM}$$

2.3 Schedule Estimation - Gantt Chart

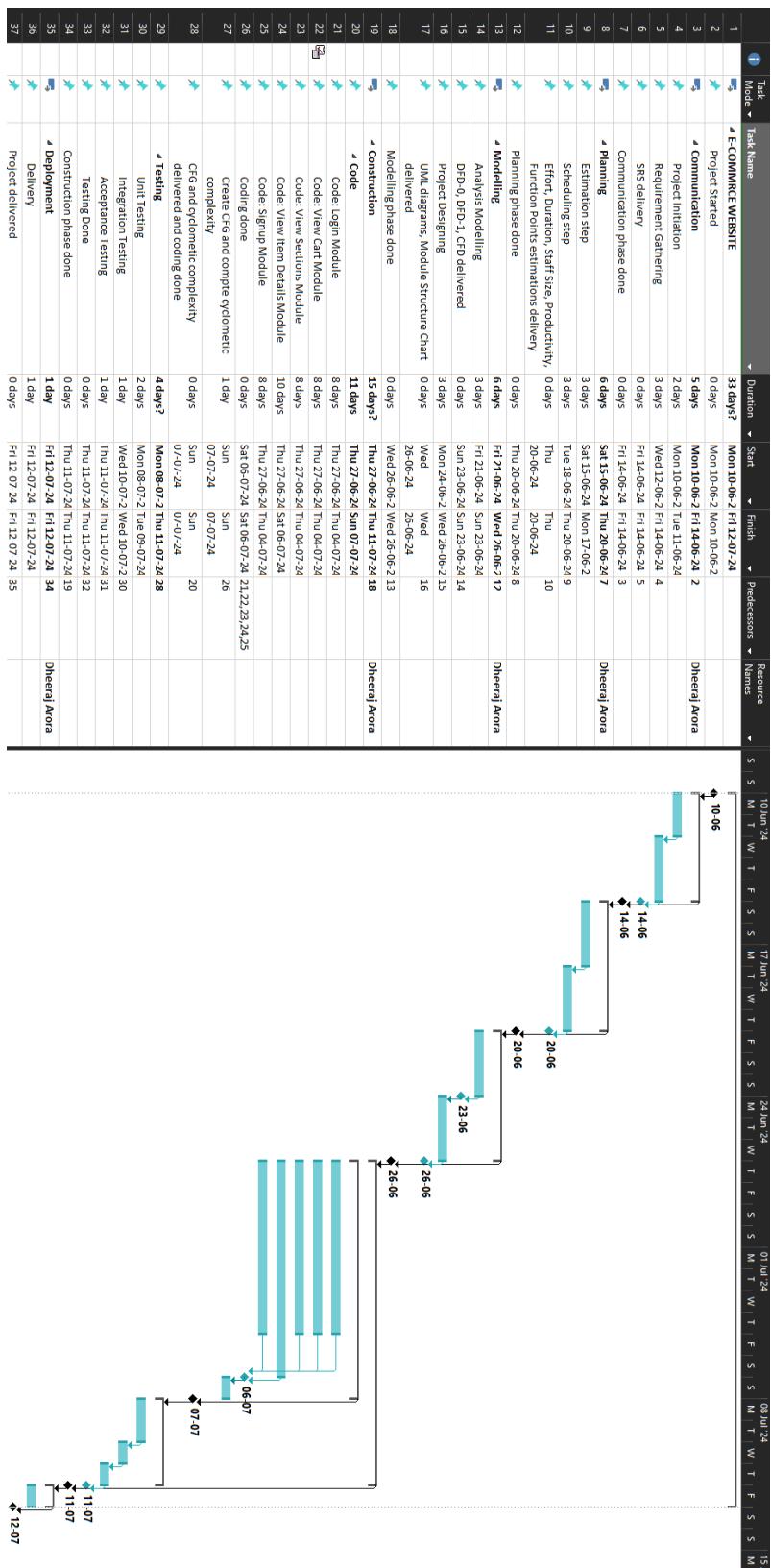


Fig 1: Gantt chart for E-commerce Website Project

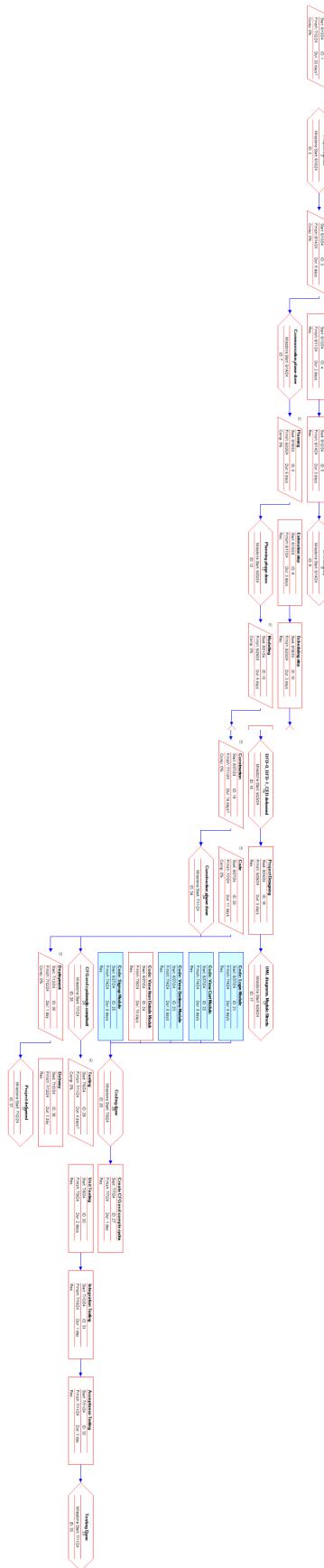


Fig 2: Network Diagram for E-commerce Website Project

2.4 Critical Path

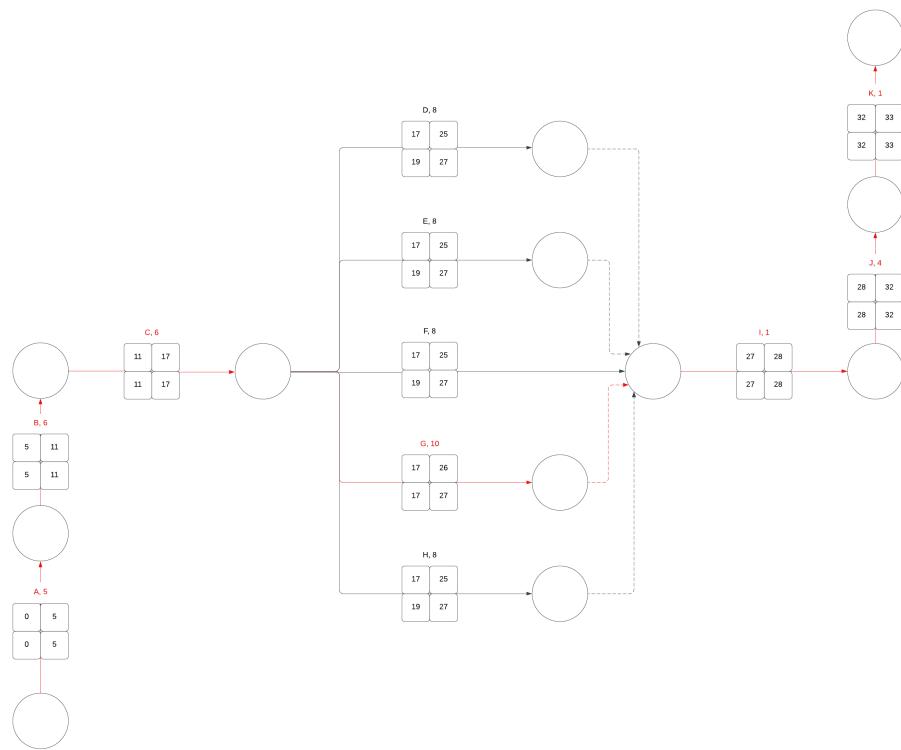


Fig 3: CPM Diagram for E-commerce Website Project

Table:

Symbol	Activity Name
A	Communication
B	Planning
C	Modelling
D	Code Login Module
E	Code View Cart Module
F	Code Sections Module
G	Code Item Details Module
H	Code Signup Module
I	Create CFG and Compute Cyclometric Complexity
J	Testing
K	Deployment

Critical Path : A-B-C-G-I-J-K

Project Duration : 33 days

3. Analysis Phase

3.1 Use Case Template

Table 2: Use-case Template for 'View Cart' Function

Use Case Title:	View Cart
Abbreviated Title:	View Cart
Use Case Id:	1
Actors:	Customer

Description: The "View Cart" function allows users to see a detailed summary of all the items they have selected for purchase. It displays product names, quantities, prices, and the total cost, enabling users to review and modify their cart contents before proceeding to checkout.

Pre-Condition: None

Task Sequence:

1. (SA) The system displays the "View Cart" option on the navigation menu.
2. (AA) The user selects the "View Cart" option.
3. (SR) The system retrieves the current contents of the user's cart.
4. (SR) The system displays the cart contents, including product names, quantities, prices, and total cost.
5. (SA) The system provides options to modify quantities, remove items, or proceed to checkout.
6. (AA) The user reviews the cart and makes any necessary modifications.
7. (SR) The system updates the cart contents and total cost based on any modifications made by the user.
8. (SR) The system displays the updated cart with the revised total cost.

Post Conditions:

1. The cart contents are accurately displayed, reflecting all current items, quantities, prices, and the total cost.
2. Any modifications made by the user, such as quantity changes or item removals, are correctly updated and saved in the cart.
3. The user can proceed to checkout with an up-to-date view of their cart, ensuring all information is correct for the final purchase.

Modification History: Date 03-july-2024

Author: Dheeraj Arora

Alternate Flow:

3. The system displays an error message reading network issue.
7. The system displays an error message reading out of stock
7. The system displays gift unlocked

Table 3: Use-case Template for 'Add to Cart' Function

Use Case Title:	Add to Cart
Abbreviated Title:	Add to Cart
Use Case Id:	2
Actors:	Customer

Description: The "Add to Cart" function allows users to select items they wish to purchase and add them to their shopping cart. It ensures the selected items are available and confirms the addition of the items to the cart.

Pre-Condition: User is on view item details page.

Task Sequence:

1. (SA) The system displays the product details and "Add to Cart" button.
2. (AA) The user selects the desired product and clicks the "Add to Cart" button.
3. (SR) The system checks the availability of the selected item.
4. (SR) The system adds the item to the user's cart.
5. (SR) The system updates the cart with the new item and adjusts the total cost.
6. (SR) The system displays a confirmation message that the item has been added to the cart.
7. (SA) The system provides options to continue shopping or proceed to checkout.

Post Conditions:

1. The selected item is accurately added to the user's cart.
2. The cart contents and total cost are updated to reflect the new item.
3. The user receives confirmation of the addition and can continue shopping or proceed to checkout.

Modification History: Date 03-july-2024

Author: Dheeraj Arora

Alternate Flow:

3. The system displays an error message indicating the item is out of stock and suggests similar products.
6. The system displays gift unlocked.

Table 4: Use-case Template for ‘View Item Details’ Function

Use Case Title:	View Item Details
Abbreviated Title:	View Item Details
Use Case Id:	3
Actors:	Customer

Description: The "View Item Details" function allows users to see comprehensive information about a specific product. It includes product images, descriptions, specifications, prices, customer reviews, and availability, enabling users to make informed purchasing decisions.

Pre-Condition: User is browsing product listings.

Task Sequence:

1. (SA) The system displays a list of products with brief information.
2. (AA) The user selects a specific product to view in detail.
3. (SR) The system retrieves detailed information about the selected product.
4. (SR) The system displays the product details, including images, descriptions, specifications, prices, customer reviews, and availability.
5. (SA) The system provides options to add the item to the cart, add to wishlist, or return to product listings.

Post Conditions:

1. The user is presented with detailed and accurate information about the selected product.
2. The user can decide to add the item to their cart, add to wishlist, or continue browsing other products.

Modification History: Date 03-july-2024

Author: Dheeraj Arora

Alternate Flow:

3. The system displays an error message indicating that the product details could not be retrieved and suggests the user try again later.

Table 5: Use-case Template for ‘View Similar Items’ Function

Use Case Title:	View Similar Item
Abbreviated Title:	View Similar Item
Use Case Id:	4
Actors:	Customer
Description: The "View Similar Items" function allows users to see a list of products similar to the one they are currently viewing. It helps users explore alternative options with similar features, prices, or brands, facilitating informed purchasing decisions.	
Pre-Condition: User is viewing the details of a specific product.	
Task Sequence:	
1. (SA) The system displays detailed information about a specific product. 2. (AA) The user selects the "View Similar Items" option. 3. (SR) The system retrieves a list of products similar to the currently viewed item. 4. (SR) The system displays the list of similar items, including brief information such as images, names, prices, and key features. 5. (SA) The system provides options to view details of a similar item, add to cart, or return to the previous product.	
Post Conditions:	
1. The user is presented with a list of products similar to the one they are currently viewing. 2. The user can choose to view more details about any of the similar items, add one to the cart, or return to the previously viewed product.	
Modification History: Date 03-july-2024	
Author: Dheeraj Arora	
Alternate Flow:	
3. The system displays an error message indicating that the product details could not be retrieved and suggests the user try again later. 4. The system displays a message indicating that no similar items are available and suggests viewing related categories.	

Table 6: Use-case Template for 'View Sections' Function

Use Case Title:	View Sections
Abbreviated Title:	View Sections
Use Case Id:	5
Actors:	Customer

Description: The "View Sections" function allows users to navigate through different sections of products categorized by gender and age group, including Men's, Women's, and Kids' sections. It provides users with easy access to browse and shop products tailored to their preferences.

Pre-Condition: User is viewing the main navigation menu.

Task Sequence:

1. (SA) The system displays options for different product sections: Men's, Women's, and Kids'.
2. (AA) The user selects a specific section option (Men's, Women's, or Kids').
3. (SR) The system retrieves and displays products categorized under the selected section.
4. (SR) The system presents the products with brief information such as images, names, prices, and key features within the chosen section.
5. (SA) The system provides options to view details of a product, add to cart, or return to the section menu.

Post Conditions:

1. The user is presented with products categorized under the selected section (Men's, Women's, or Kids').
2. The user can explore and shop products within the chosen section based on their preferences.

Modification History: Date 03-july-2024

Author: Dheeraj Arora

Alternate Flow:

3. The system displays an error message indicating that the product details could not be retrieved and suggests the user try again later.
4. The system displays a message indicating that no products are available and suggests exploring other sections.

Table 7: Use-case Template for ‘Log-in’ Function

Use Case Title:	Log in
Abbreviated Title:	Log in
Use Case Id:	6
Actors:	Registered Customer
<p>Description: The "Login" function allows users to securely access their account by providing valid credentials. It verifies user identity and grants access to personalized features such as viewing order history, managing preferences, and completing purchases.</p>	
<p>Pre-Condition: User is not logged into the system.</p>	
<p>Task Sequence:</p> <ol style="list-style-type: none"> 1. (SA) The system displays the login screen with fields for username/email and password. 2. (AA) The user enters their username/email and password. 3. (SR) The system validates the entered credentials against stored user data. 4. (SR) The system verifies the user's identity and authenticates access. 5. (SR) The system redirects the user to their account dashboard or the previously visited page. 6. (SA) The system displays personalized account options such as order history, profile settings, and shopping cart. 	
<p>Post Conditions:</p> <ol style="list-style-type: none"> 1. The user is successfully logged into their account and can access personalized features. 2. The system securely manages user authentication and protects user information. 	
<p>Modification History: Date 03-july-2024</p>	
<p>Author: Dheeraj Arora</p>	
<p>Alternate Flow:</p> <ol style="list-style-type: none"> 3. The system displays an error message indicating invalid username/email or password. 3. The system displays an error message indicating that login cannot be processed at this time and suggests trying again later. 	

Table 8: Use-case Template for ‘Sign-up’ Function

Use Case Title:	Sign up
Abbreviated Title:	Sign up
Use Case Id:	7
Actors:	Non-Registered Customer

Description: The "Sign Up" function allows new users to create an account by providing necessary information such as username, email, and password. It registers the user in the system and grants access to personalized features and services.

Pre-Condition: User is not registered in the system.

Task Sequence:

1. (SA) The system displays the sign-up screen with fields for username, email, and password.
2. (AA) The user enters the required information (username, email, password) and submits the form.
3. (SR) The system validates the entered information (e.g., checks for duplicate email, password strength).
4. (SA) The system creates a new user account with the provided information.
5. (SR) The system displays a message indicating successful registration
6. (SR) The system redirects the user to the login screen or the user dashboard.

Post Conditions:

1. A new user account is created and stored in the system.
2. The user can log in and access personalized features and services.

Modification History: Date 03-july-2024

Author: Dheeraj Arora

Alternate Flow:

3. The system displays an error message indicating that the password does not meet the security requirements and suggests creating a stronger password.
3. The system displays an error message indicating account already exists.
3. The system displays an error message indicating details added in wrong format.
3. The system displays an error message indicating that login cannot be processed at this time and suggests trying again later.

Table 9: Use-case Template for 'Update Account Details' Function

Use Case Title:	Update Account Details
Abbreviated Title:	Update Account Details
Use Case Id:	8
Actors:	Registered Customer

Description: The "Update Account Details" function allows users to modify their account information, such as name, email, password, address, and other personal details. It ensures that users can keep their account information up-to-date and accurate.

Pre-Condition: User is logged into the system.

Task Sequence:

1. (SA) The system displays the user account dashboard with an option to update account details.
2. (AA) The user selects the "Update Account Details" option.
3. (SR) The system retrieves and displays the current account details.
4. (AA) The user modifies the desired account details (e.g., name, email, password, address).
5. (AA) The user submits the updated information.
6. (SR) The system validates the entered information (e.g., checks for duplicate email, password strength).
7. (SA) The system updates the account details in the database.
8. (SR) The system displays a confirmation message indicating that the account details have been successfully updated.

Post Conditions:

1. The user's account details are updated and stored in the system.
2. The user can see the updated information reflected in their account.

Modification History: Date 03-july-2024

Author: Dheeraj Arora

Alternate Flow:

6. The system displays an error message indicating invalid updation.
6. The system displays an error message indicating details added in wrong format.
6. The system displays an error message indicating that login cannot be processed at this time and suggests trying again later.

Table 10: Use-case Template for ‘View Total Orders Placed Details’ Function

Use Case Title:	View Total Orders Placed Details
Abbreviated Title:	View Total Orders Placed Details
Use Case Id:	9
Actors:	Manager

Description: The “View Total Orders Placed Details” function allows a manager to see comprehensive information about all orders placed. It includes order numbers, customer information, product details, quantities, prices, order status, and delivery information, enabling effective order management and tracking.

Pre-Condition: Manager is logged into the system with appropriate permissions to view order details..

Task Sequence:

1. (SA) The system displays the manager dashboard with an option to view total orders placed details.
2. (AA) The manager selects the "View Total Orders Placed Details" option.
3. (SR) The system retrieves and displays a list of all orders with brief information such as order numbers, customer names, and order dates.
4. (AA) The manager selects a specific order to view detailed information.
5. (SR) The system retrieves and displays detailed information about the selected order, including order number, customer information, product details, quantities, prices, order status, and delivery information.
6. (SA) The system provides options to update order status, print order details, or return to the order list.

Post Conditions:

1. The manager is presented with detailed and accurate information about all orders placed.
2. The manager can view, manage, and track orders effectively.

Modification History: Date 03-july-2024

Author: Dheeraj Arora

Alternate Flow:

3. The system displays an error message indicating that the orders cannot be retrieved at this time and suggests trying again later.
3. The system displays an error message indicating no orders placed.
5. The system displays an error message indicating that the order details are unavailable and suggests contacting support.

Table 11: Use-case Template for ‘Place Order’ Function

Use Case Title:	Place Order
Abbreviated Title:	Place Order
Use Case Id:	10
Actors:	Registered Customer

Description: The “View Total Orders Placed Details” function allows a manager to see comprehensive information about all orders placed. It includes order numbers, customer information, product details, quantities, prices, order status, and delivery information, enabling effective order management and tracking.

Pre-Condition: User has items in cart and is logged in.

Task Sequence:

1. (SA) The system displays the user's shopping cart
2. (AA) The user selects the "Proceed to Checkout" option.
3. (SR) The system prompts the user to enter or confirm shipping information.
4. (AA) The user enters or confirms the shipping information.
5. (SR) The system prompts the user to enter or confirm payment information.
6. (AA) The user enters or confirms the payment information.
7. (SA) The system validates the shipping and payment information.
8. (SR) The system displays a summary of the order details, including items, shipping address, and total cost, and prompts the user to confirm the order.
9. (AA) The user reviews the order summary and confirms the order.
10. (SA) The system processes the payment and updates the order status to "Processing".
11. (SR) The system generates an order confirmation and displays a confirmation message with the order number.

Post Conditions:

1. The order is successfully placed and recorded in the system.

Modification History: Date 03-july-2024

Author: Dheeraj Arora

Alternate Flow:

2. The system displays an error message indicating no item in cart.
7. The system displays an error message indicating the issue and prompts the user to re-enter the payment information.
7. The system displays an error message indicating the issue and prompts the user to re-enter the shipping information.
7. The system displays an error message indicating that the order cannot be processed at this time and suggests trying again later.
8. The user cancels the order.

Table 12: Use-case Template for 'Dispatch Orders' Function

Use Case Title:	Dispatch Order
Abbreviated Title:	Dispatch Order
Use Case Id:	11
Actors:	Staff

Description: The "Dispatch Order" function allows staff to manage the dispatch process of orders that have been placed. It ensures that orders are correctly packed, labeled, and sent out for delivery.

Pre-Condition: Order has been placed by the customer

Task Sequence:

1. (SA) The system displays a list of orders that are ready for dispatch.
2. (AA) The staff selects an order to dispatch.
3. (SR) The system retrieves and displays the details of the selected order, including customer information, shipping address, items, and quantities.
4. (AA) The staff confirms the items and quantities to be dispatched.
5. (SR) The system generates a packing slip and shipping label for the order.
6. (AA) The staff prints the packing slip and shipping label.
7. (AA) The staff packs the items, attaches the shipping label, and includes the packing slip.
8. (SA) The system updates the order status to "Dispatched".
9. (SR) The system updates the inventory to reflect the dispatched items.

Post Conditions:

1. The order status is updated to "Dispatched".
2. The system records the dispatch information and updates the inventory.

Modification History: Date 03-july-2024

Author: Dheeraj Arora

Alternate Flow:

3. The system detects an issue with the order details and displays the issue.
3. The system displays an error message indicating that the order cannot be processed for dispatch at this time and suggests trying again later.
6. Paper out of stock warning generated
8. The system displays an error message indicating that the order cannot be processed for dispatch at this time and suggests trying again later.

Table 13: Use-case Template for 'Add item' Function

Use Case Title:	Add Item
Abbreviated Title:	Add Item
Use Case Id:	12
Actors:	Staff

Description: The "Add Item" function allows staff to upload new products onto the website. This includes providing detailed information about the item such as name, description, price, category, stock quantity, and images.

Pre-Condition: Staff is logged into the system with appropriate permissions to add items.

Task Sequence:

1. (SA) The system displays the staff dashboard with an option to add a new item.
2. (AA) The staff selects the "Add Item" option.
3. (SR) The system displays a form for entering item details.
4. (AA) The staff enters the item details, including name, description, price, category, stock quantity, and uploads images.
5. (AA) The staff submits the item details form.
6. (SR) The system validates the entered information (e.g., checks for required fields, valid price format).
7. (SA) The system saves the new item details to the database.
8. (SR) The system displays a confirmation message indicating the item has been successfully added.
9. (SR) The system updates the website to reflect the new item in the appropriate category.

Post Conditions:

1. The new item is successfully added to the system and displayed on the website.
2. The inventory is updated to reflect the new item.

Modification History: Date 03-july-2024

Author: Dheeraj Arora

Alternate Flow:

6. The system detects missing or invalid information and displays an error.
6. The system displays an error message indicating that the item cannot be added at this time and suggests trying again later.

3.2 Use-Case Diagram

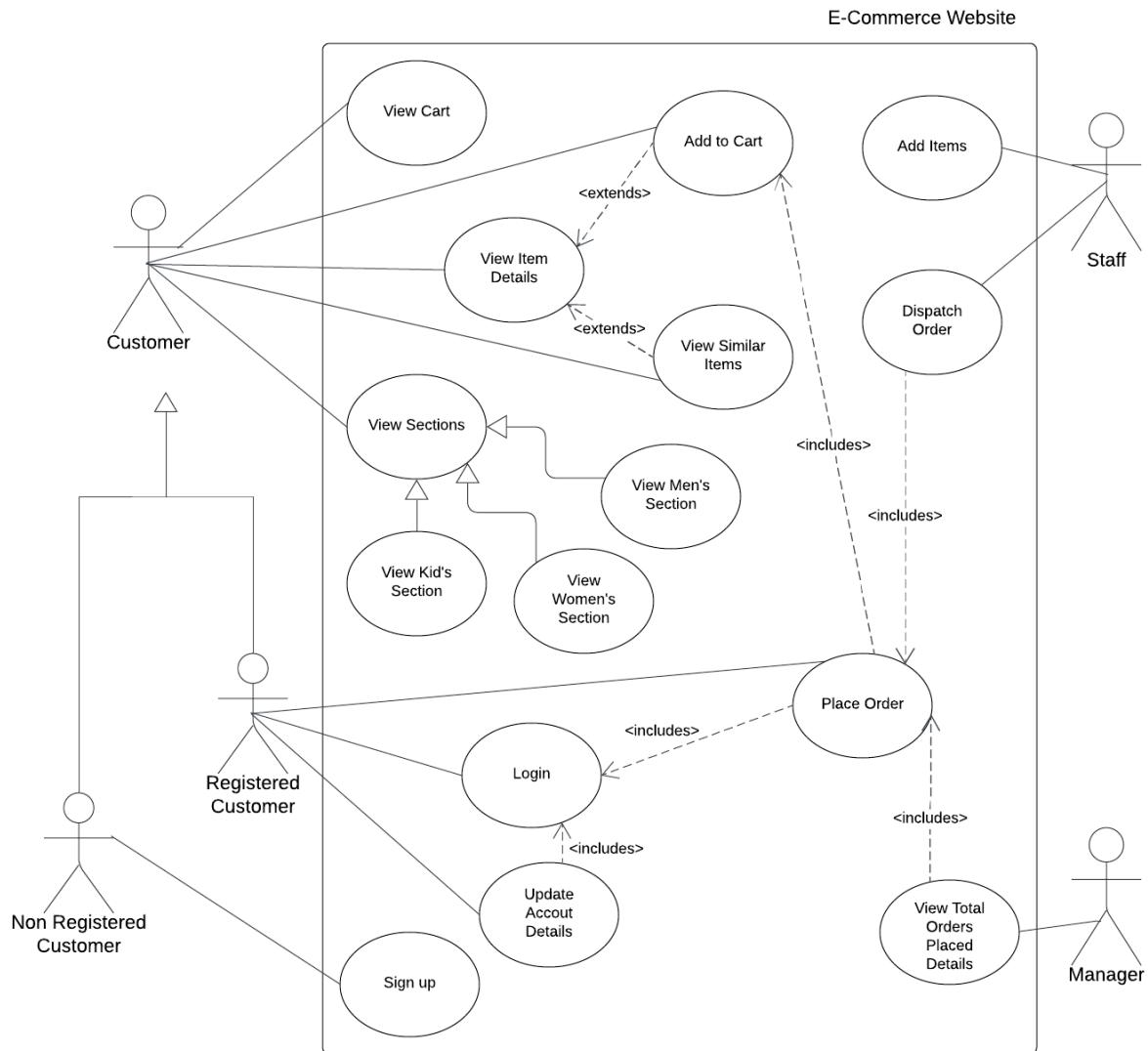


Fig 4: Use-Case Diagram for E-commerce Website Project

3.3 Swimlane Diagram

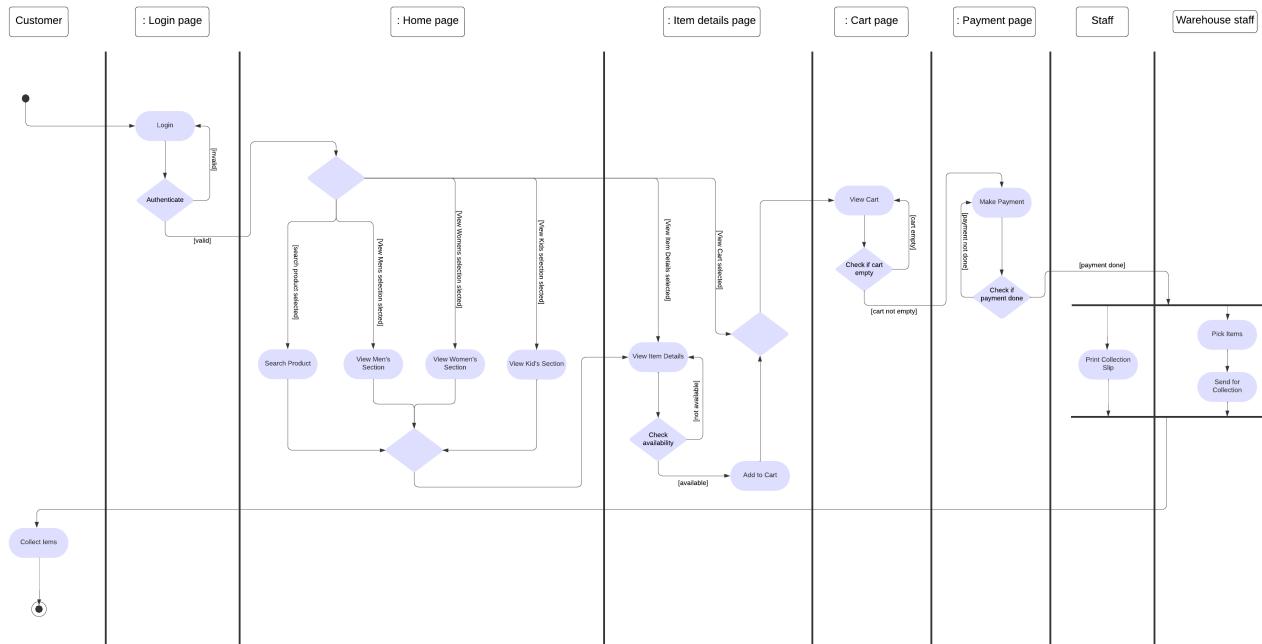


Fig 5: Swimlane Diagram for E-commerce Website Project

3.4 DFD Level 0

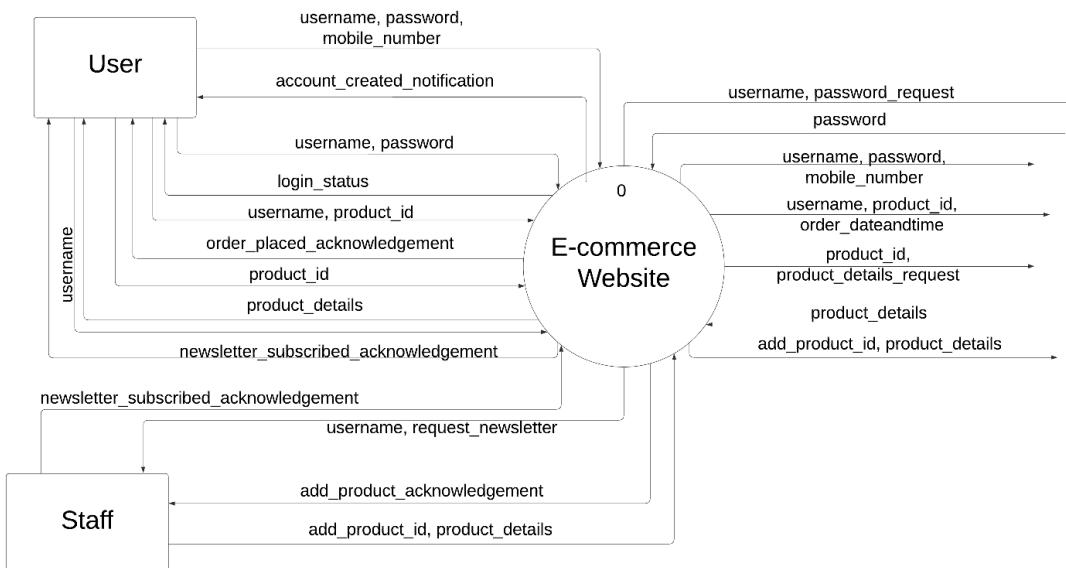


Fig 6: DFD Level 0 Diagram for E-commerce Website Project

DFD Level 1

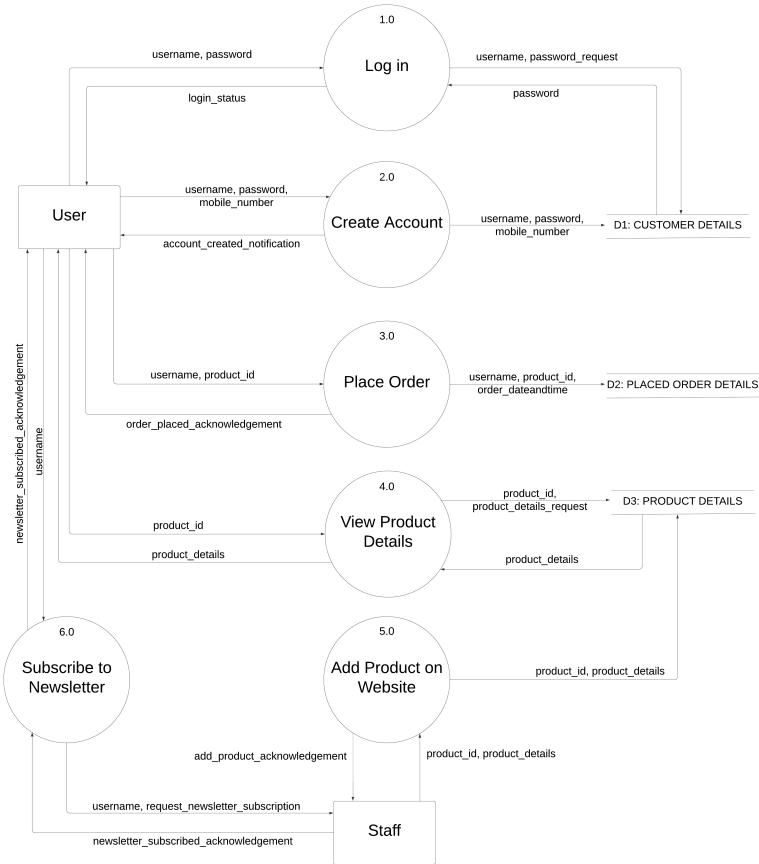


Fig 7: DFD Level 1 Diagram for E-commerce Website Project

3.5 Data Dictionary Entries for Data Stores, Processes and Entities

Data Stores:

1)

Name:	Customer Details
Description:	Contains personal information and login credentials of the customer.
Inward Flow:	username, mobile_number, password_request
Outward Flow:	password

2)

Name:	Placed Order Details
Description:	Includes information related to a customer's order, such as product IDs, quantities, and delivery address.
Inward Flow:	username, product_id, order_dateandtime
Outward Flow:	nil

3)

Name:	Product Details
Description:	Consists of detailed information about products, including ID, name, price, and stock levels.
Inward Flow:	product_id, product_details, product_details_request
Outward Flow:	product_details

Entities:

1)

Name:	User
Description:	Represents customers who browse and purchase products on the website.
Inward Flow:	account_created_notification, login_status, order_placed_acknowledgement, product_details, newsletter_subscribed_acknowledgment
Outward Flow:	username, password, mobile_number, product_id

2)

Name:	Staff
Description:	Represents employees managing the e-commerce website, including administrators and customer service representatives.
Inward Flow:	username, request_newsletter, add_product_acknowledgement
Outward Flow:	add_product_id, product_details, newsletter_subscribed_acknowledgment

Processes:

1)

Name:	Login
Description:	Authenticates users and staff by verifying their credentials.
Inward Flow:	username, password
Outward Flow:	username, password_request, login_status
Pspec	<p>Process narrative: The Login process authenticates users and staff by verifying their credentials against stored data. If the credentials match, access to the system's features is granted.</p> <p>Pseudo code:</p> <ol style="list-style-type: none"> 1. Retrieve stored_password for username from User database 2. If stored_password is NULL <ul style="list-style-type: none"> a. Return error message "Username not found" 3. Compare input password with stored_password 4. If passwords match <ul style="list-style-type: none"> a. Generate access_token b. Return success message and access_token 5. Else a. Return error message "Incorrect password"

2)

Name:	Create Account
Description:	Registers new users by collecting personal information and creating login credentials.
Inward Flow:	username, password, mobile_number
Outward Flow:	account_created_notification, username, password, mobile_no.
Pspec	<p>Process narrative: The Create Account process registers new users by collecting personal information, creating login credentials, and storing them in the user database.</p> <p>Pseudo code:</p> <ol style="list-style-type: none"> 1. Check if email already exists in User database 2. If email exists <ul style="list-style-type: none"> a. Return error message "Email already registered" 3. Else <ul style="list-style-type: none"> a. Encrypt password b. Store user_details in User database c. Return success message "Account created successfully"

3)

Name:	Place Order
Description:	Handles product selection, cart management, and order confirmation.
Inward Flow:	username, product_id
Outward Flow:	order_placed_acknowledgement, username, product_id, order_dateandtime
Pspec	<p>Process narrative: The Place Order process handles product selection, cart management, payment processing, and order confirmation, resulting in an order record in the system.</p> <p>Pseudo code:</p> <ol style="list-style-type: none"> 1. Validate cart_details (check stock availability) 2. If any product is out of stock <ul style="list-style-type: none"> a. Return error message "Some items are out of stock" 3. Process payment with payment_info 4. If payment is successful <ul style="list-style-type: none"> a. Create order record in Order database with cart_details, payment_info, and shipping_address b. Update product stock in Product database c. Return order_confirmation 5. Else <ul style="list-style-type: none"> a. Return error message "Payment failed"

4)

Name:	View Product Details
Description:	Displays detailed information about a specific product selected by the user.
Inward Flow:	product_id, product_details
Outward Flow:	product_id, product_details, product_details_request
Pspec	<p>Process narrative: The View Product Details process displays detailed information about a specific product selected by the user.</p> <p>Pseudo code:</p> <ol style="list-style-type: none"> 1. Retrieve product_details from Product database using product_id 2. If product_details are found <ul style="list-style-type: none"> a. Return product_details 3. Else <ul style="list-style-type: none"> a. Return error message "Product not found"

5)

Name:	Add Product on Website
Description:	Allows staff to add new products to the website's catalog by entering product details.
Inward Flow:	product_id, product_details
Outward Flow:	product_id, product_details, add_product_acknowledgement
Pspec	<p>Process narrative: The Add Product on Website process allows staff to add new products to the website's catalog by entering product details and storing them in the product database.</p> <p>Pseudo code:</p> <ol style="list-style-type: none">1. Validate product_details (check for required fields)2. If validation fails<ol style="list-style-type: none">a. Return error message "Invalid product details"3. Store product_details in Product database4. Return success message "Product added successfully"

A CASE STUDY (IEEE Format)

Software Requirements Specification Document

Version 1.0

E-Commerce Website

TABLE OF CONTENTS

Chapter No.	Topic	Page No.
1.	<u>Introduction</u>	3
1.1	<u>Purpose of this Document</u>	3
1.2	<u>Scope of the Development</u>	3
1.3	<u>Abbreviations and Acronyms</u>	4
1.4	<u>References</u>	4
1.5	<u>Overview</u>	5
2.	<u>Overall Description</u>	5
2.1	<u>Product Perspective</u>	5
2.2	<u>Product functions</u>	6
2.3	<u>User Characteristics</u>	7
2.4	<u>General Constraints, Assumptions and Dependencies</u>	7
2.5	<u>Apportioning of the requirements</u>	8
3.	<u>Specific Requirements</u>	9
3.1	<u>External Interface Requirements</u>	9
3.2	<u>Detailed Description of Functional Requirements</u>	9
3.2.1	<u>Functional Requirements for Customer</u>	10
3.2.2	<u>Functional Requirements for Staff admin</u>	10
3.2.3	<u>Functional Requirements for Manager admin</u>	10
3.3	<u>Performance requirements</u>	10
3.4	<u>Logical database requirements</u>	11
3.5	<u>Quality attributes</u>	11
3.6	<u>Other requirements</u>	11
4.	<u>Change History</u>	12
5.	<u>Document Approvers</u>	12

1. Introduction

1.1 Purpose of this Document

The purpose of this SRS document is to provide a detailed overview of our software product, its parameters, and its goals. This document describes the project's target audience and user interface, hardware, and software requirements. It defines how our client, team, and audience will see the product and use it. We further discuss the future scope of the project.

1.2 Scope of Development

The goal of this project is to design an e-commerce website interface so that users can shop for clothing items in an easier and more efficient way.

In this project, the user will log in using their email address, and then continue to the main page to shop for clothes in three different sections, men's, women's, and kid's. The user can then further select the items of their choice add them to the cart and proceed to checkout and track their packet. Users can also subscribe to the website's newsletter for further notifications and details.

The software should perform the following operations:

- **Login:** The user should be able to log in using only their Email address.
- **Create a new account:** For new users.
- **Newsletter:** Subscribe to the newsletter (optional)
- **Browse through sections:** Users should be able to browse through the three different sections of our clothing website.
- **Add to cart:** Select and add items to the cart and proceed to place an order.
- **Track packet:** Users can easily track the movement of their package.

1.3 Abbreviations and Acronyms

Table 1 gives the full form of most commonly used mnemonics in this SRS document.

Table 1: Full form of the most commonly used mnemonics.

SNo	Mnemonic	Full Form
1.	MERN	MongoDB, Express.js, React, Node.js
2.	UFI	User-friendly interface
3.	SFF	Small form factor
4.	MongoDB	Mongo DataBase
5.	URL	Uniform Resource Locator
6.	Ad	Advertisement

1.4 References

- [1] GeeksforGeeks, “E-commerce Website using MERN Stack,” 20 March 2024. [Online]. Available: <https://www.geeksforgeeks.org/e-commerce-website-using-mern-stack/>.
- [2] GeeksforGeeks, “What is MERN Stack?,” GeeksforGeeks, 5 April 2024. [Online]. Available: <https://www.geeksforgeeks.org/what-is-mern-stack/>.
- [3] MongoDB, “ How To Use MERN Stack: A Complete Guide,” MongoDB, 2024. [Online]. Available: <https://www.mongodb.com/resources/languages/mern-stack-tutorial>.
- [4] O. C. Jude, “How To Get Started with the MERN Stack,” DigitalOcean, 31 August 2021. [Online]. Available: <https://www.digitalocean.com/community/tutorials/getting-started-with-the-mern-stack>.
- [5] Justlookingforthemoment, “MEAN (solution stack),” Wikipedia, 22 December 2023. [Online]. Available: [https://en.wikipedia.org/wiki/MEAN_\(solution_stack\)](https://en.wikipedia.org/wiki/MEAN_(solution_stack)).
- [6] N. Ranganath, “The Ultimate Guide to MERN Stack,” DEV Community, 15 January 2021. [Online]. Available: <https://dev.to/itsnitinr/the-ultimate-guide-to-mern-stack-1051>.
- [7] N. Abramowski, “What is the MERN Stack? A Complete 2024 Beginner's Guide,” CareerFoundry, 10 January 2023. [Online]. Available: <https://careerfoundry.com/en/blog/web-development/what-is-the-mern-stack-a-complete-beginners-guide>.

development/what-is-the-mern-stack/.

- [8] M. M. & H. A., “25 Best eCommerce Websites With Fantastic Designs (2024),” Hostinger, 20 May 2024. [Online]. Available: <https://www.hostinger.in/tutorials/e-commerce-website-examples>.
- [9] K. Shubham, “Build an E-Commerce website with MERN Stack - Part 1 (Setting Up the Project),” DEV Community., 22 January 2022. [Online]. Available: <https://dev.to/shubham1710/build-an-e-commerce-website-with-mern-stack-part-1-setting-up-the-project-112d>.
- [10] G. Roy, “Ultimate guide to build an E-commerce website using MERN Stack: Comprehensive tutorial,” Medium, 20 April 2024. [Online]. Available: <https://medium.com/@royg4250/ultimate-guide-to-build-an-e-commerce-website-using-mern-stack-comprehensive-tutorial-5fdb45e1dfd6>.

1.5 Overview

The remaining sections of this document provide a general description, including the **characteristics** of the users of this project, the **product's hardware**, and the **functional and data** requirements of the product. A general description of the project is discussed in section 2 of this document. Section 2 gives the functional requirements, data requirements, **constraints, and assumptions** made while designing the **multi-utility system**. It also gives the user a viewpoint of product use. Section 3 gives the **specific requirements of the product**. Section 3.0 also discusses the **external interface requirements** and gives a detailed description of functional requirements.

2. Overall Description

2.1 Product Perspective

The product will run as a MERN stack e-commerce website called ‘SHOPPER’. It will be compatible with laptops and PCs and capable of running on any web browser. This application will enable users to shop for clothes online.

The SHOPPER website will have the following modules:

1. Home Page
2. Home Page-Newsletter
3. Signup/Login
4. Men's Section
5. Women's Section
6. Kids's Section
7. Shopping Cart

The user will **log in** using their email address and password, and then continue to the **main page** to shop for clothes in **three different sections**, men's, women's, and kid's. The user can then further **select the items** of their choice **add them to the cart** and proceed to **checkout and track their packet**. Users can also subscribe to the website's **newsletter** for further notifications and details. User and Product information is maintained using MongoDB. The admin users can update this information.

2.2 Product Functions

The product should perform the following operations:

1. It must be able to successfully sign up users during login and view the home page.
2. It must be able to view different clothing sections for men, women, and children.
3. It must be able to show detailed information about each product, including images, descriptions, prices, and specifications.
4. It must allow users to add products to a shopping cart, display the contents of the cart with quantities and prices, and implement functionalities like updating quantities and removing items from the cart.
5. It must allow users to place orders and provide users with order confirmation and order tracking capabilities.
6. It must create an admin panel to manage products, user accounts, and order details.
7. It must allow users to subscribe to the newsletter and send email notifications to the subscribers.

2.3 User Characteristics

The goal is to design an e-commerce website using the MERN (MongoDB, Express.js, React, Node.js) stack.

The basic user types that can access the website are listed below:

1. Registered Customer
2. Unregistered Customer
3. Staff Admin
4. Manager Admin

Understanding your target audience is crucial when developing an e-commerce website. As one can see from the list, each type of user will have different access to the functionality of using the system. We aim to develop software that should be easy to use for all types of users. Here are some key user characteristics to consider:

- The user is computer-literate and has little or no difficulty in using online websites to access information such as login, clothing details, etc.
- In order to use the website it is not required that a user beware of the internal working of the site, but he/she is expected to know how to interact with the website.

2.4 General Constraints, Assumptions, and Dependencies

The following list presents the constraints, assumptions, dependencies, or guidelines that are imposed upon implementation of the website SHOPPER:

- The website must have a UFI, **user-friendly interface** that is simple enough for all types of users to understand.
- There are **no memory requirements**.
- **Response time** for loading the website should be no longer than five seconds.
- A general knowledge of **basic computer skills** is required to use the product.
- An accurate and up-to-date **product information** should be available for integration.
- **Security measures**, such as encryption and secure connections, should be implemented

effectively.

- The website should comply with **legal and regulatory requirements** related to e-commerce, including data protection laws, privacy regulations, and online transaction regulations.

2.5 Apportioning of Requirements

The MERN-based e-commerce website is to be implemented in the following phases:

1. **Pilot Phase:** In pilot phase we will we will deploy the site on a staging server, conduct extensive functionality and performance tests, and gather user feedback to assess the system's usability and stability. This phase will involve testing features such as user registration, product browsing, cart management, and simulated checkout, ensuring all components work seamlessly. By identifying and resolving any issues, optimizing performance, and documenting our findings, we aim to refine the platform, ensuring a robust and user-friendly online marketplace.
2. **Advertisement:** We will leverage targeted ads on Facebook, Instagram, and Twitter to attract potential users, highlighting key features such as amazing products stock, easy product navigation, and seamless browsing experience. Collaborating with influencers and running promotional campaigns, including limited-time offers and giveaways, will further boost visibility and user participation. Additionally, we will use analytics tools to track engagement and adjust our strategy in real-time, ensuring we effectively reach and resonate with our target audience.
3. **Future Deployment:** This can also be made into a mobile application so that user can shop for the items from their phones too. Functionalities such as user display activity, and payment methods can also be implemented to make the interface, even more, user-friendly. ML technology can be introduced such as in areas of personalized recommendations, dynamic pricing, fraud detection, virtual assistants, Ad targeting, and trend analysis.

3. Specific Requirements

3.1 External Interface Requirements

The following list presents the external interface requirements:

- The primary user interface will be a responsive web application accessible through standard web browsers.
- The product requires very limited graphics usage with just a simple keypad for taking user input.
- The product does not require the usage of sound. It is not an essential feature but it can be considered for future variants of the system.
- The hardware and operating system require a screen resolution of not more than 320 x 240 pixels (owing to the small form factor SFF).
- The product ensures real-time synchronization of product details, including images, descriptions, and prices. It interacts with the MongoDB database for storing and retrieving product information, user data, and order details.

3.2 Detailed Description of Functional Requirements

3.2.1 Functional Requirements for Customer.

Table 2 gives the Functional Requirements for Customer.

Table 2: Functional Requirements for Customer.

Purpose	This screen thus provides information specific to each user upon the successful identification of the Login.
Inputs	A user can view a page of information by choosing from one of the options given on the welcome screen. Selection is performed with a simple keypad.
Processing	The menu responds to selections by displaying a page containing the pre-defined text requested information.
Outputs	Output consists of a screen of information specific to a user. For example, upon choosing a section eg. Men's, the user can then further view items and products of that section.

3.2.2 Functional Requirements for Staff Admin

Table 3 gives the Functional Requirements for Staff Admin.

Table 3: Functional Requirements for Staff Admin.

Purpose	This screen provides information specific to the Staff Admin.
Inputs	The staff can view a page of information by choosing from one of the options given on the welcome screen. Selection is performed with a simple keypad.
Processing	The menu responds to selections by displaying a page containing the pre-defined text requested information.
Outputs	Output consists of a screen of information with options. For example, upon choosing the option named- ‘Deleting/update User profiles’ in the menu displayed on the welcome screen, the Admin may be able to disbar and ban suspicious users.

3.2.3 Functional Requirements for Manager Admin

Table 4 gives the Functional Requirements for Manager Admin

Table 4: Functional Requirements for Manager Admin

Purpose	This screen provides information specific to Manager Admin.
Inputs	Manager Admin can view a page of information by choosing from one of the options given on the welcome screen. Selection is performed with a simple keypad.
Processing	The menu responds to selections by displaying a page containing the pre-defined text requested information.
Outputs	Output consists of a screen of information. For example, upon choosing Option - ‘update specific Item’ in the menu displayed on the welcome screen, the Manager may be able to update items.

3.3 Performance Requirements

- The website is primarily designed for laptops/desktops to view and order clothing products from the SHOPPER e-commerce website.
- The website ensures compatibility with popular web browsers and optimizes performance for each.
- The system will support concurrent users without significant degradation in performance.
- The architecture is scalable to handle a growing number of products, users, and transactions.

- Optimized and compressed images to minimize their impact on page load times.

3.4 Logical Database Requirements

Figure 1 shows the E-R diagram for the entire system.

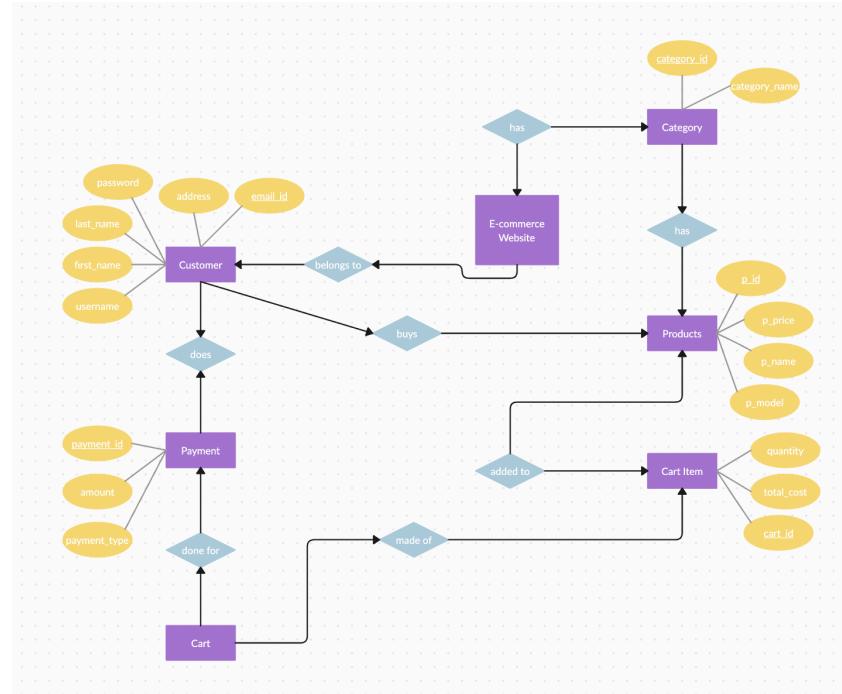


Figure 1: E-R Diagram for E-commerce Website Project

3.5 Quality Attributes

Quality attributes are essential aspects that define the overall quality, reliability, and effectiveness of an e-commerce website developed using the MERN stack. The project should prioritize attributes such as performance, ensuring quick page load times, and responsive design to enhance user experience. The product must load quickly and work well on a variety of terminals. It must also tolerate a wide variety of input possibilities from a user.

3.6 Other Requirements

- Business Rules
- Good network bandwidth

4. Change History

200209	Version 1.0 – Initial Release

5. Document Approvers

SRS for *E-Commerce Website* approved by:

Name: Dr. Harkiran Kaur

Designation: Assistant Professor

Date: 04/07/24

4. Design Phase

4.1 Modular Structure Chart

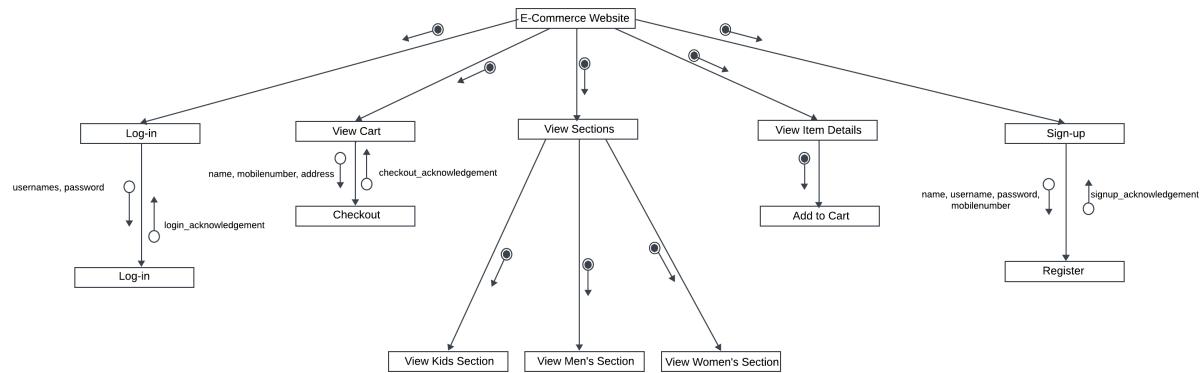


Fig 8: Modular Structure Chart for E-commerce Website Project

4.2 Class Diagram

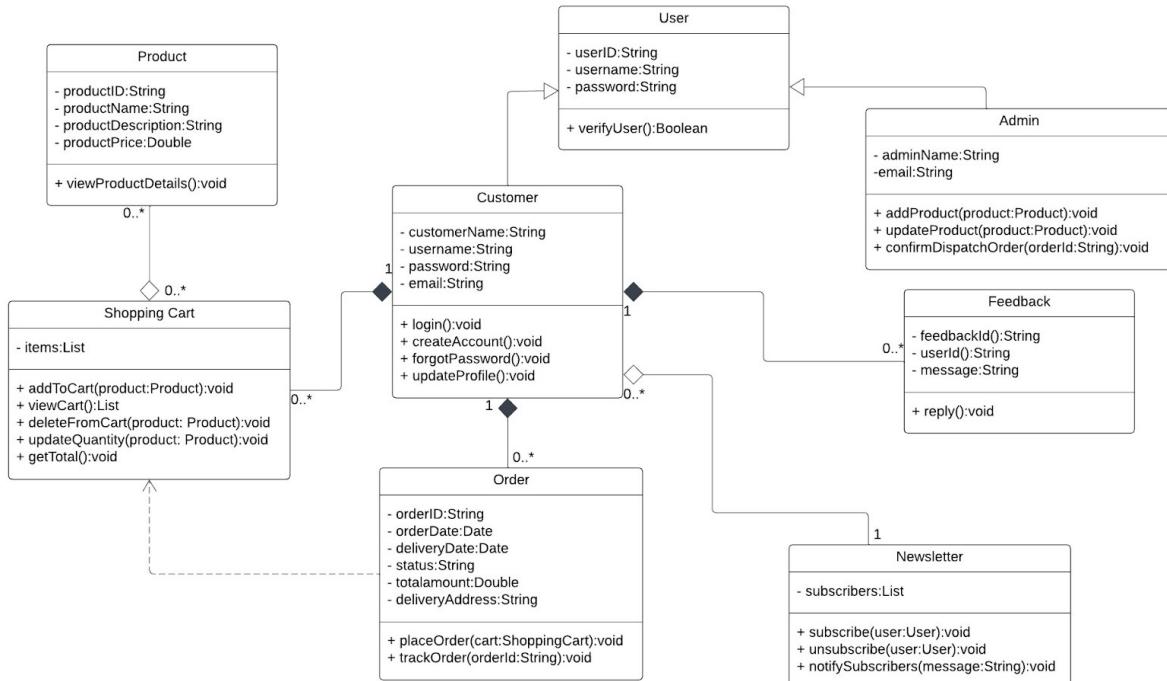


Fig 9: Class Diagram for E-commerce Website Project

4.3 Database Design - ER Diagram

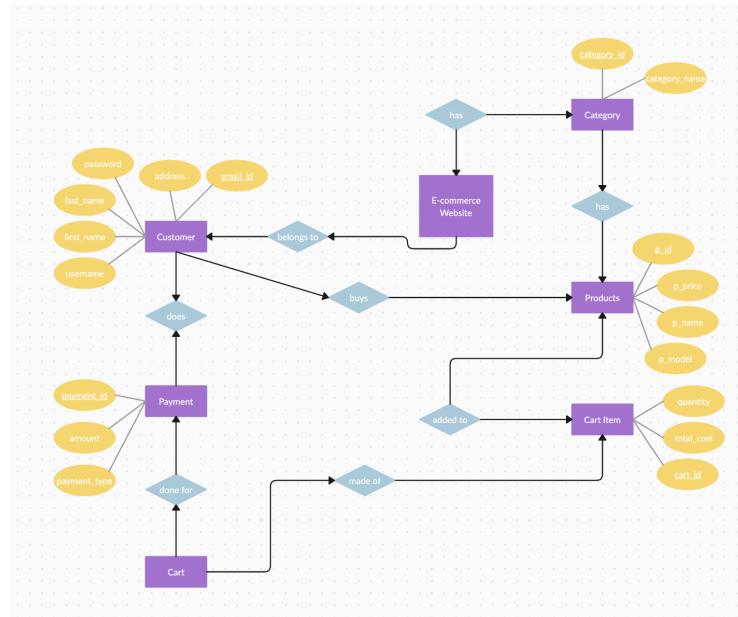


Fig 10: ER Diagram for E-commerce Website Project

4.4 Sequence Diagram

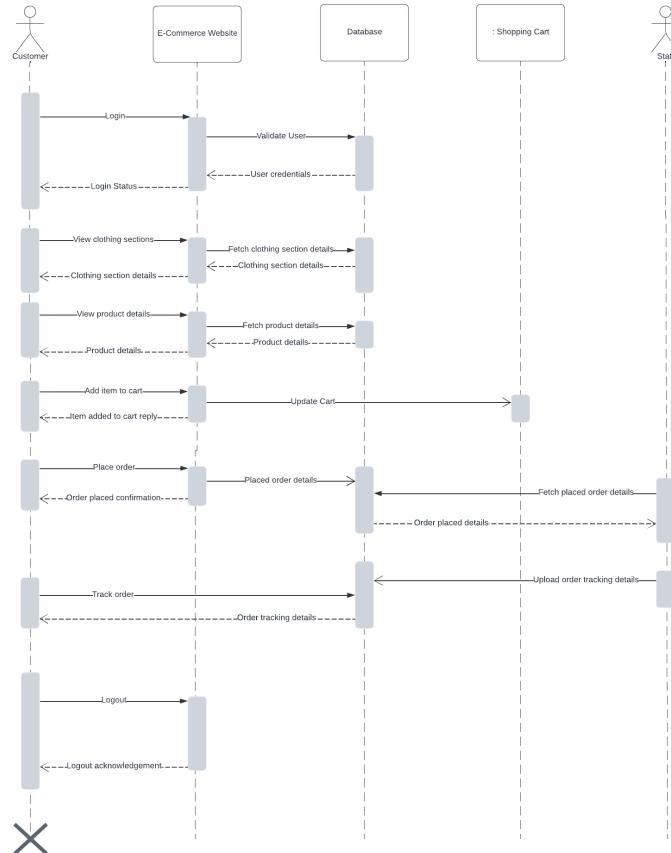
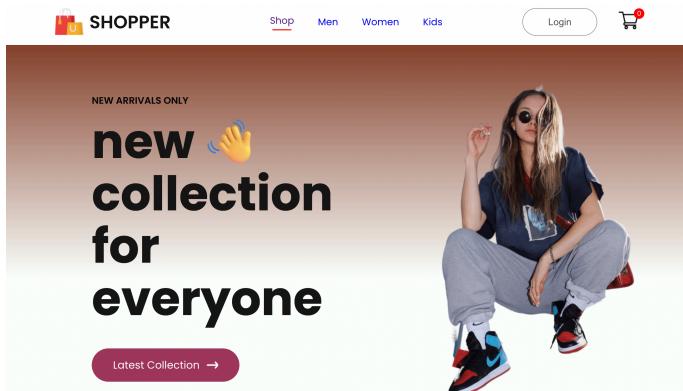
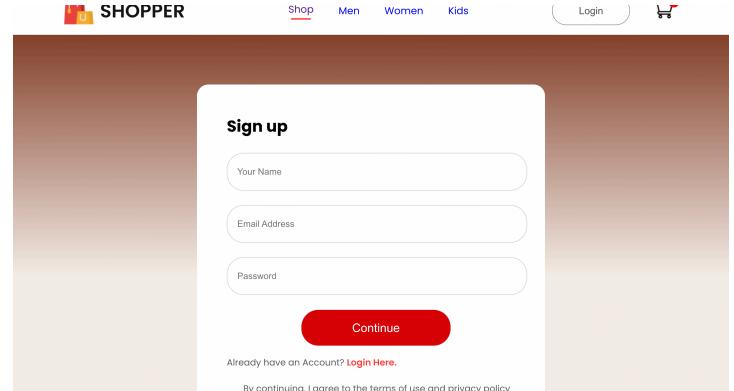


Fig 11: Sequence Diagram for E-commerce Website Project

5. Coding Phase

5.1 Working Project Screenshots

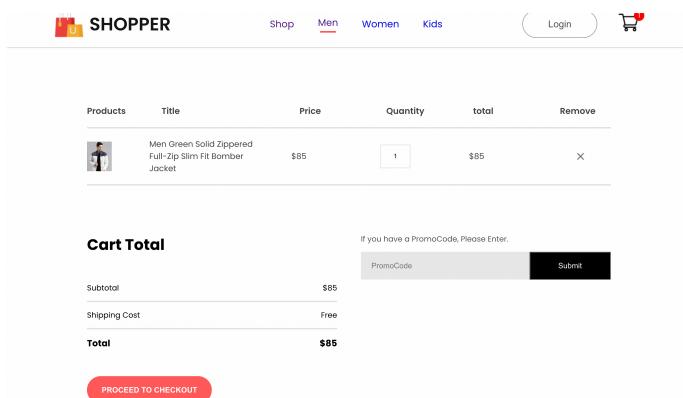
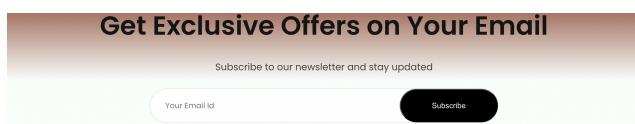




Fig 12: Working Screenshots of E-commerce Website Project

6. Testing Phase

6.1 Cyclomatic Complexity Calculation

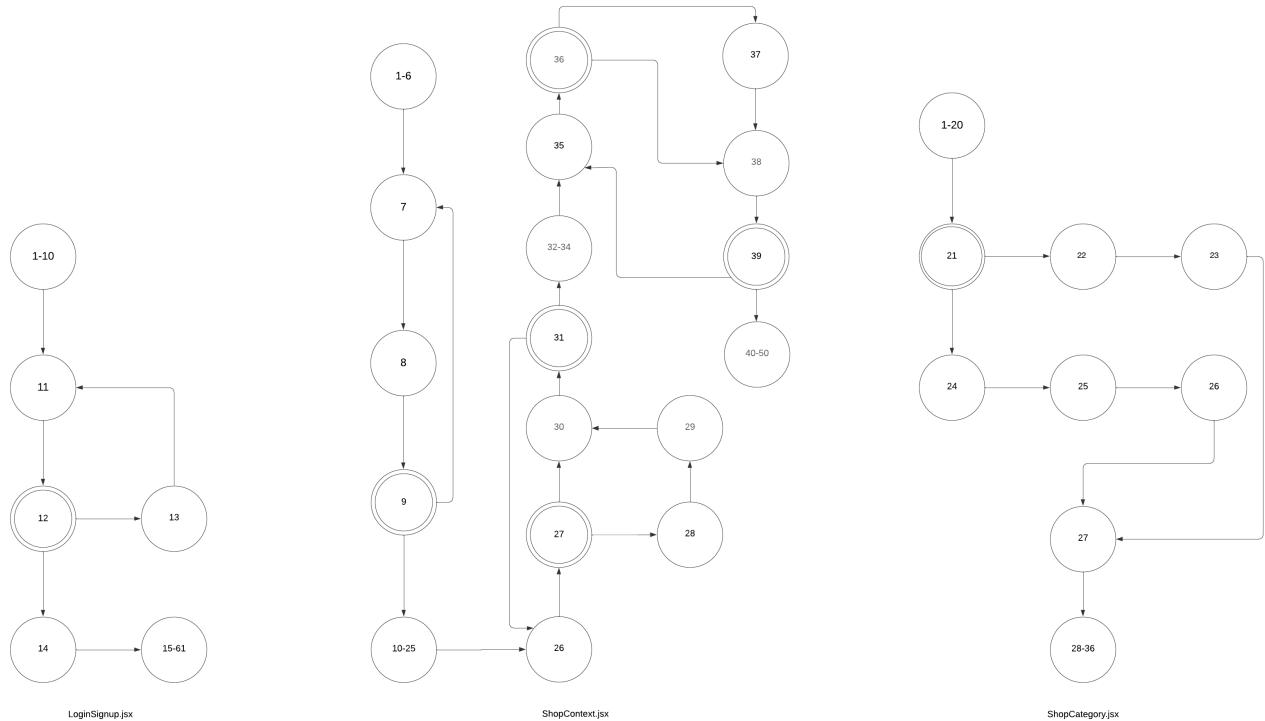


Fig 13: CPM Diagram for E-commerce Website Project

$$\begin{aligned}\text{Cyclomatic Complexity} &= \text{No. Of independent paths} \\ &= \text{No. Of decision nodes} + 1\end{aligned}$$

- LoginSignup.jsx

Independent Paths:

- 1) 1..10 - 11 - 12 - 14 - 15..61
- 2) 1..10 - 11 - 12 - 13 - 11 - 12 - 14 - 15..61

$$\text{Cyclomatic Complexity} = 2$$

$$\text{No. Of decision nodes: } 1$$

$$\text{Cyclomatic Complexity} = 1+1 = 2$$

- ShopContext.jsx

Independent Paths:

- 1) 1..6 - 7 - 8 - 9 - 10..25 - 26 - 27 - 30 - 31 - 32..34 - 35 - 36 - 37..50
- 2) 1..6 - 7 - 8 - 9 - 7 - 8 - 9 - 10..25 - 26 - 27 - 30 - 31 - 32..34 - 35 - 36 - 37..50
- 3) 1..6 - 7 - 8 - 9 - 10..25 - 26 - 27 - 30 - 31 - 32..34 - 35 - 36 - 37 - 38 - 39 - 40..50

- 4) 1..6 - 7 - 8 - 9 - 10..25 - 26 - 27 - 30 - 31 - 26 - 27 - 30 - 31 - 32..34 - 35 - 36 - 37 - 38 - 39 - 40..50
- 5) 1..6 - 7 - 8 - 9 - 10..25 - 26 - 27 - 28 - 29 - 30 - 31 - 32..34 - 35 - 36 - 37 - 38 - 39 - 40..50
- 6) 1..6 - 7 - 8 - 9 - 10..25 - 26 - 27 - 30 - 31 - 32..34 - 35 - 36 - 38..50

Cyclomatic Complexity = 6

No. Of decision nodes: 5

Cyclomatic Complexity = $5+1 = 6$

- ShopCategory.jsx

Independent Paths:

- 1) 1..20 - 21 - 22 - 23 - 27 - 28..36
- 2) 1..20 - 21 - 24 - 25 - 26 - 27 - 28..36

Cyclomatic Complexity = 2

No. Of decision nodes: 1

Cyclomatic Complexity = $1+1 = 2$

6.2 Test Case Template

Test Case 1: Items Added and Removed from cart

Test Case#: 1 System: E-Commerce Website Designed By: Dheeraj Arora (102103508) Executed By: Dheeraj Arora (102103508) Short description: Testing for the working of cart	Test Case Name: Functional Cart Subsystem: Cart Validation Design Date: 27/6/2024 Execution Date: 8/7/2024	Page 1 of 3
Pre-Conditions: <ul style="list-style-type: none"> ● User must be logged in ● Item selected should be in cart 		

Step	Action	Expected System Response	Pass/Fail	Comments
1.	Add to cart	Item added to cart	Pass	
2.	View cart	Cart opened	Pass	
3.	Another Item Added	Item added to cart	Pass	
4.	Total amount in cart added	Amount updated	Pass	

Post-Conditions:

- The claimed item should be in the cart and the cart can be viewed successfully.

Conclusion: The working of the cart is executing successfully

Test Case 2: Related items for every item selected

Test Case#: 2 System: E-Commerce Website Designed By: Dheeraj Arora (102103508) Executed By: Dheeraj Arora (102103508) Short description: Items and related items validity tested.	Test Case Name: Related items Subsystem: Item Validation Design Date: 28/6/2024 Execution Date: 9/7/2024	Page 2 of 3
Pre-Conditions: <ul style="list-style-type: none"> ● User must be logged in. 		

Step	Action	Expected System Response	Pass/Fail	Comments
1.	Select Product	Related Products featured along with details.	Pass for women section products Fail for men section products	Did not show related items for men.

Post-Conditions:

- The selected section was found to have related products to the section chosen by the user.

Conclusion: The related items feature is partially executing.

Test Case 3: Login

Test Case#: 3 System: E-Commerce Website Designed By: Dheeraj Arora (102103508) Executed By: Dheeraj Arora (102103508) Short description: Test Login page functionality	Test Case Name: Login Subsystem: Login Design Date: 27/6/2024 Execution Date: 9/7/2024	Page 3 of 3
Pre-Conditions: <ul style="list-style-type: none">User must be sign up interface		

Step	Action	Expected System Response	Pass/Fail	Comments
1.	Fill details	If block left empty, error should be displayed	Pass	
2.	Clicked on login, after filling details	Information verified and user logged in	Fail	Did not process input information

Post-Conditions:

- The user logged in to the website

Conclusion: The login page is running partially