

**VIETNAM NATIONAL UNIVERSITY OF HOCHIMINH CITY**  
**THE INTERNATIONAL UNIVERSITY**  
**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**



**WEB APPLICATION DEVELOPMENT**

**IT093IU**

**FINAL REPORT**

**Topic: Furniture E-commerce Website**

**By Group 05 – Member List**

No.	Full Name	Student ID	Role
1	Huỳnh Thanh Thuỷ	ITITIU21325	Project Manager
2	Đoàn Hữu Nguyên	ITITIU20260	Team Member
3	Nguyễn Vạn Huy	ITITIU20215	Team Member
4	Nguyễn Phạm Kỳ Phương	ITITIU21287	Team Member

Instructor: Assoc. Prof. Nguyen Van Sinh

## **ACKNOWLEDGEMENTS**

We express our sincere gratitude and appreciation to Assoc. Prof. Nguyen Van Sinh for his professional guidance. His unwavering encouragement and support were instrumental in helping our team achieve its goals. We also extend our thanks to MSc. Nguyen Trung Nghia, a laboratory lecturer, whose technical assistance and good humor greatly enriched our learning experience this semester.

We would also like to express our sincere gratitude to the irreplaceable members of our group. Their technical expertise and collaborative spirit were essential to our progress. Beyond their willingness to share their knowledge and troubleshoot challenges, their good humor and positive attitudes made this project an enriching and enjoyable learning experience. We are grateful to have had the opportunity to work alongside such a talented and supportive team.

# TABLE OF CONTENTS

<b>ACKNOWLEDGEMENTS.....</b>	<b>2</b>
<b>TABLE OF CONTENTS.....</b>	<b>3</b>
<b>LIST OF FIGURES.....</b>	<b>5</b>
<b>LIST OF TABLES.....</b>	<b>5</b>
<b>ABSTRACT.....</b>	<b>6</b>
<b>CHAPTER 1: INTRODUCTION.....</b>	<b>7</b>
1. Project Description.....	7
a. Background.....	7
b. Goals and Objectives.....	7
c. Report Structure.....	8
<b>CHAPTER 2: PROJECT PLANNING.....</b>	<b>9</b>
1. Project Timeline and Milestone.....	9
2. People In-charge.....	10
3. Development Process & Environment.....	10
a. Development Process.....	10
b. Development Environment.....	11
<b>CHAPTER 3: REQUIREMENT ANALYSIS AND DESIGN.....</b>	<b>13</b>
1. Functional Requirements.....	13
a. User Management.....	13
b. Product Management.....	13
c. Shopping Cart.....	13
d. Checkout and Payment.....	13
e. Order Management.....	13
2. Non - functional Requirements.....	13
a. Performance.....	13
b. Scalability.....	13
c. Usability.....	14
d. Scalability.....	14
3. Use Cases.....	14
4. UI/UX Design.....	24
a. Site Map and Main User Flow.....	24
b. Wireframes.....	27
c. Interface Modules.....	29
5. Database Design.....	33
a. Entity Relationship Diagram (ERD).....	33
b. Database Schema.....	36
<b>CHAPTER 4: IMPLEMENTATION.....</b>	<b>39</b>
1. Login and Authentication.....	39
2. Landing Page.....	41

3. Product Listing.....	42
4. Product Cart.....	43
5. Checkout.....	44
6. Order Management.....	46
<b>CHAPTER 5: TESTING.....</b>	<b>48</b>
1. Test case 1.....	48
2. Test case 2.....	49
<b>CHAPTER 6: DISCUSSION AND EVALUATION.....</b>	<b>50</b>
1. Discussion.....	50
a. Challenges Faced.....	50
b. Solutions and Current Implementation.....	50
2. Evaluation.....	50
a. Strengths.....	51
b. Weaknesses.....	51
c. Opportunities.....	51
d. Threats.....	52
<b>CHAPTER 7: CONCLUSION AND FUTURE WORK.....</b>	<b>53</b>
1. Conclusion.....	53
2. Future work.....	53
<b>REFERENCES.....</b>	<b>55</b>

## LIST OF FIGURES

Figure 2.1: Gantt chart for illustrating project timeline.....	9
Figure 3.1: Use Case Diagram.....	15
Figure 3.2: The Site Map.....	24
Figure 3.3: The Main User Flow for Purchasing Products.....	26
Figure 3.4: Wireframe for the main screens.....	28
Figure 3.5: Landing Page UI Design.....	29
Figure 3.6: Login - Register UI Design.....	30
Figure 3.7: Product Listing Page UI Design.....	30
Figure 3.7: Cart and Checkout UI Design.....	31
Figure 3.8: Admin Panel UI Design.....	32
Figure 3.9: Entity Relationship Diagram.....	33
Figure 3.10: Database schema.....	36
Figure 4.1: Login and Sign Up Front-end.....	40
Figure 4.2: Landing Page Front-end.....	41
Figure 4.3: Product Listing Front-end.....	43
Figure 4.4: Product Cart Front-end.....	44
Figure 4.5: Shipping Address Confirmation Front-end.....	45
Figure 4.5: Order Successful Front-end.....	46
Figure 4.6: API for Order Management.....	47

## LIST OF TABLES

Table 2.1: Individual responsibility and contribution.....	10
Table 5.1: Test case 1.....	48
Table 5.2: Test case 2.....	49

# ABSTRACT

This report details the development of a furniture e-commerce website built using React.js, Node.js, and MongoDB, focusing on essential functionalities for an efficient online shopping experience. The website provides secure customer authentication, a shopping cart for managing selected items, an order system for tracking purchases, and robust product and category management for seamless navigation. React.js ensures a responsive and user-friendly interface, Node.js facilitates efficient server-side operations, and MongoDB offers a scalable database solution for storing customer, product, and order data. The project highlights the integration of these technologies and the challenges addressed, showcasing the effectiveness of modern web development tools in delivering streamlined e-commerce solutions.

# CHAPTER 1: INTRODUCTION

This section provides a brief introduction to our project, Furnitech, outlining its core aspects and fundamental details. It also concludes with a summary of the key constraints we faced during the project.

## 1. Project Description

### a. Background

- This project is developed for the Final Project of the Web Application Course, aimed at providing students with hands-on experience in web development. This project simulates a real-world e-commerce platform, where users can browse, purchase, and review furniture products while administrators manage inventory and orders.

### b. Goals and Objectives

#### i. Goals:

- Create a user-friendly website where customers can browse and purchase furniture online.
- Provide hands-on experience with web development, user interface design, and project management.
- Develop skills in implementing practical solutions for real-world problems through a simulated e-commerce platform.

#### ii. Objective:

- Build a website with approximately 10 pages using HTML and CSS that adapts to different screen sizes (responsive design).
- Implement essential functionalities such as user registration, login, product browsing, shopping cart, and checkout.
- Ensure pages load within acceptable time limits and the interface is intuitive for all users.
- Prepare detailed documentation and a demonstration to showcase the system's functionality and implementation process.

### c. Report Structure

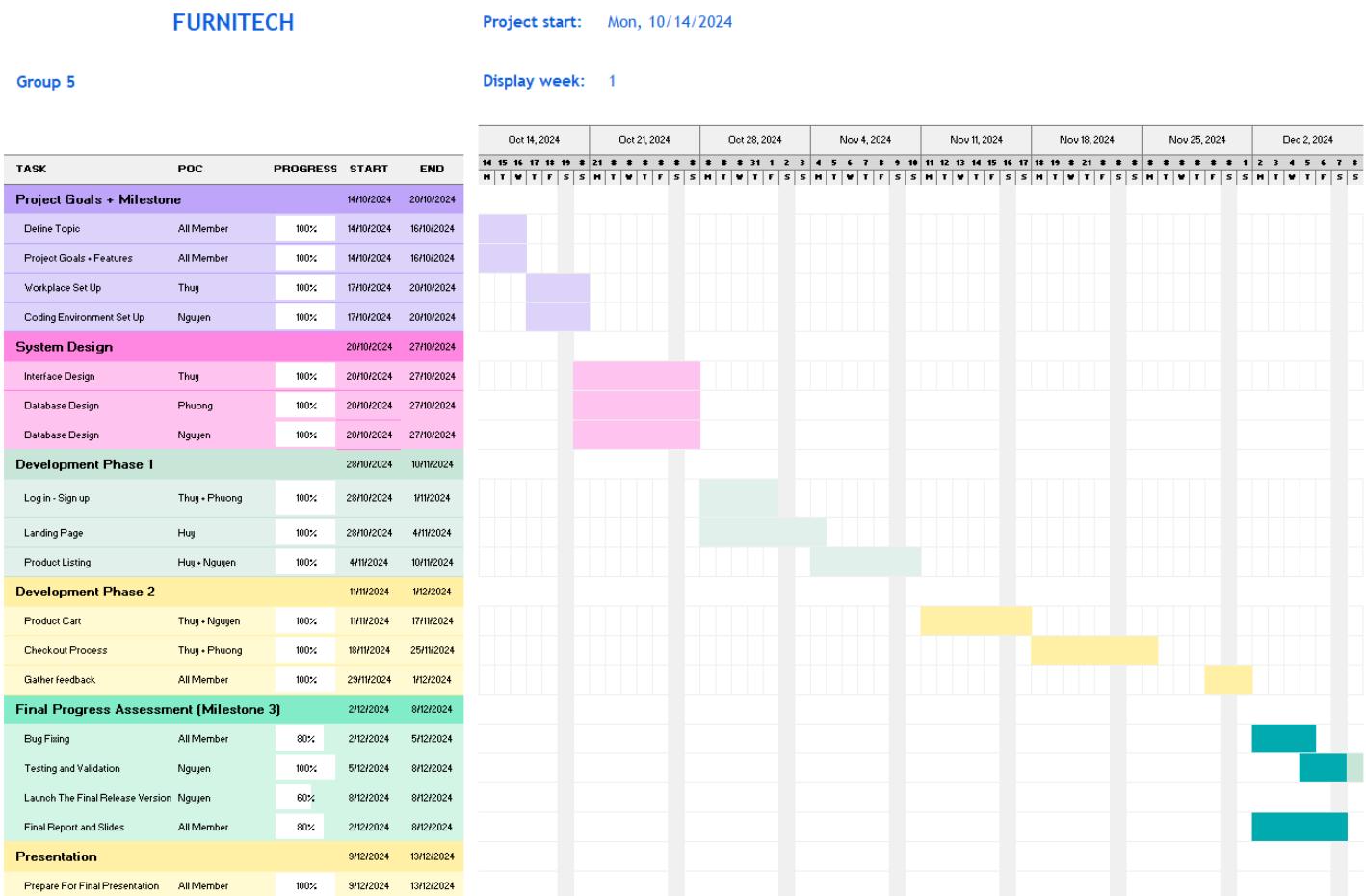
- **Chapter 1 Introduction:** Provide an overview of the project.
- **Chapter 2 Project Planning:** Present the project plan which considers all the factors that may affect the development process.
- **Chapter 3 Requirement Analysis and Design:** Describe the project requirements and system design.
- **Chapter 4 Implementation and Results:** Describe the project's implementation and present the outcomes or results achieved.
- **Chapter 5 Testing:** Focus on validating core features, ensuring stability, security, and checking for potential bugs or performance issues.
- **Chapter 6 Discussion and Evaluation:** Interpret and analyze the results obtained in the previous chapter.
- **Chapter 7 Conclusion and Future Work:** Summarize the essential findings and conclusions of the project.

# CHAPTER 2: PROJECT PLANNING

## 1. Project Timeline and Milestone

- The project timeline is illustrated using a Gantt chart, which provides a visual overview of the project schedule, including the start and end dates of each task, their durations, and dependencies between tasks.

Figure 2.1: Gantt chart for illustrating project timeline



### + Key Notes

- Dependencies: The development phase heavily relied on the completion of the design and approval of wireframes to proceed.

### + Milestones:

- Project Topic Selection and Environment Set Up (End of Week 1).

- Completion of Interface and Database Design (End of Week 2).
  - Core Modules Implement (End of Week 4)
  - Fully Functional Website (End of Week 8)
  - Final Delivery and Document (End of Week 10)
- + **Adjustments:** Feedback iterations during the design phase caused minor timeline adjustments to ensure an optimal user interface.

## 2. People In-charge

*Table 2.1: Individual responsibility and contribution*

Full Name	Responsibility	Contribution
Huỳnh Thanh Thủy	Front-end Developer, UI/UX Developer	25%
Đoàn Hữu Nguyên	Back-end Developer, Tester	25%
Nguyễn Vạn Huy	Front-end Developer	25%
Nguyễn Phạm Kỳ Phương	Back-end Developer, Database Developer	25%

## 3. Development Process & Environment

### a. Development Process

- The development process for the furniture e-commerce website was carried out using a structured, step-by-step approach to ensure efficiency and clarity throughout the entire project lifecycle. The first step involved requirement analysis, which focused on identifying the core features necessary for the website, such as user authentication, cart management, order processing, product cataloging, and category browsing. This initial step established the foundation by understanding user needs and mapping out technical functionalities.
- Following the analysis, the design phase was undertaken. This included designing the database schema using the Entity-Relationship Diagram (ERD) and creating intuitive wireframes for the user interface. The front-end design was conceptualized to prioritize user experience, while the back-end architecture was planned to handle the platform's logic and API development efficiently.

- Next was the technology selection phase, where the tools and frameworks were chosen for their reliability, scalability, and compatibility with the project needs. React.js was selected for building a responsive and dynamic front-end interface, while Node.js with Express.js was chosen for handling server-side operations. MongoDB was adopted as the database due to its flexibility and ease of integration.
- During the development phase, the front-end and back-end were developed concurrently. The front-end team focused on creating reusable React.js components for product displays, user interaction, and authentication flows. Meanwhile, the back-end team developed RESTful APIs using Node.js and Express.js to handle database operations like user authentication, order processing, and product management. MongoDB was used to store customer data, product details, cart transactions, and order history.
- After the development of core features, a series of testing and debugging activities were conducted. Functional testing ensured that all features performed as intended, while debugging tools like Chrome DevTools and Postman were employed to identify and resolve any issues. Testing was vital to ensure that API calls, database interactions, and user workflows were working seamlessly.
- Finally, the platform was deployed for user access. Hosting services such as Heroku and Netlify were used to host the application and ensure that it was accessible to users. The deployment phase included preparing the website for real-time use and scaling it for concurrent users while monitoring server performance.

## b. Development Environment

- **Front-end:**
  - + React.js: Framework for building the user interface.
  - + JavaScript: Core programming language for front-end scripting.
  - + HTML & CSS (Tailwind CSS): For structuring and styling web pages.

- + Axios: Used for handling HTTP requests to interact with the back-end APIs.
- **Back-end:**
  - + Node.js: JavaScript runtime for back-end logic.
  - + Express.js: Framework for creating RESTful APIs.
  - + JSON Web Tokens (JWT): For user authentication and security.
- **Database:**
  - + MongoDB: NoSQL database for data storage and retrieval.
  - + Mongoose: ODM library to simplify interaction with MongoDB.
- **Tools & Platforms:**
  - + Visual Studio Code (VS Code) + IntelliJ IDEA: Code editor for front-end and back-end development.
  - + Postman: For testing API endpoints and ensuring proper request-response functionality.
  - + Git and GitHub: Version control system to track changes and collaborate efficiently.
  - + Chrome DevTools: For debugging and optimizing front-end performance.
- **Operating System:**
  - + The project was developed on MacOS/Windows environments, ensuring compatibility across platforms.

# CHAPTER 3: REQUIREMENT ANALYSIS AND DESIGN

## 1. Functional Requirements

### a. User Management

- Users can register, log in, and log out.
- Basic user profiles with a name and contact details.

### b. Product Management

- Admins can add, edit, and delete furniture products.
- Products include basic details: name, price, description, and image.
- Users can view and search for furniture products by category.

### c. Shopping Cart

- Users can add and remove items from a shopping cart.
- A summary of the cart displays item names, quantities, and total cost.

### d. Checkout and Payment

- A simple checkout process that allows users to enter their shipping details.
- Payment can be simulated (e.g., no real payment gateway integration).

### e. Order Management

- Users can view their past orders.
- Orders have statuses (e.g., "Pending," "Completed").

## 2. Non - functional Requirements

### a. Performance

- Pages should load within 5 seconds on local hosting.
- The system should handle up to 20 concurrent users.

### b. Scalability

- The architecture should allow for adding new features or increasing capacity without major restructuring.

### c. Usability

- A clean and simple interface suitable for a university project demonstration.

### d. Scalability

- Basic scalability to add new categories or extend features without significant code changes.

## 3. Use Cases

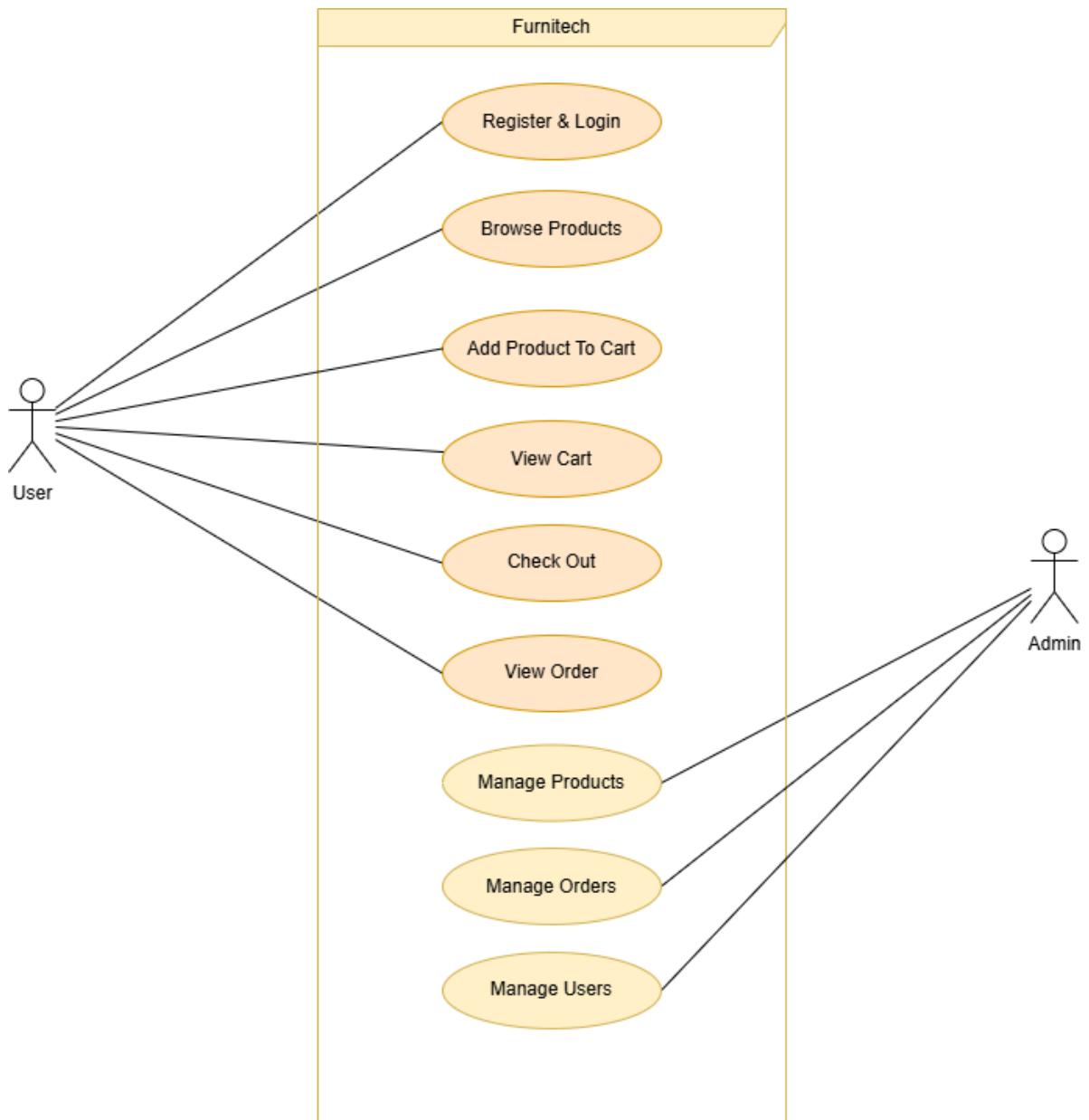
- The use case diagram is drawn to visually represent the interactions between different types of users (actors) and the various functionalities (use cases) provided by the system. This summary use case diagram helps stakeholders understand the system's functionality, identify the user requirements, and facilitate communication among the development team and stakeholders. Below is the description.

- **Actors**

- + **User:** A customer who visits the website to browse, purchase furniture, and leave reviews.
- + **Admin:** A person managing the website's backend to maintain products and monitor activity.

## - Use Case Diagram

Figure 3.1: Use Case Diagram



## Use Cases for User

### 1. Register & Login

- + **Description:** Allows new users to create or log in to an account.
- + **Inputs:**

- For Registration: User-provided details such as name, email, password, phone number and address.
- For Login: User email and password.

+ **Outputs:**

- Success message upon successful registration or login.
- Error message if any input validation fails or login credentials are incorrect.

+ **Basic Course:**

Step	User Action	System Response
1	Access the login page.	Displays the login form with an option to navigate to the registration page.
2	Navigate to the registration page if creating an account.	Displays the registration form with the required fields.
3	Enter information to log in or register.	Captures the input and validates it.
4	Submit the registration or login form.	<p style="text-align: center;"><b>- For Registration:</b></p> <p>If valid, save user data and display a success message. If invalid, show error messages.</p> <p style="text-align: center;"><b>- For Login:</b></p> <p>If credentials are correct, grant access and display a success message. If incorrect, show an error message.</p>

+ **Preconditions:**

- Users have access to the system's login or registration page.
- Registration requires an internet connection and all mandatory fields to be filled.

+ **Postconditions:**

- Upon successful registration, the user's details are stored in the database, and they can log in using their credentials.

- Upon successful login, the user is granted access to the application/system.
- In case of failure, appropriate error messages guide the user to resolve the issue.

## 2. Browse Products

- + **Description:** Enables users to search, filter, and view available furniture products.
- + **Inputs:**
  - Filters: Category, price range, or other relevant attributes.
  - Search Query: Keyword entered by the user.
- + **Outputs:**
  - A list of products that match the search query or applied filters.
  - Relevant error messages if no products are found.

- + **Basic Course:**

Step	User Action	System Response
1	Navigate to the products page.	Displays the product page with a list of available furniture and filter/search options.
2	Apply filters (e.g., category, price) or enter a search keyword.	Captures the selected filters or search query.
3	Submit filters or search queries.	Displays products that match the criteria, or a message if no products are found.

- + **Preconditions:**

- The system has an inventory of furniture products.
- Filters and search functionality are accessible on the products page.

- + **Postconditions:**

- Products matching the applied filters or search query are displayed.
- If no matches are found, an appropriate message is displayed to the user.

## 3. Add Product to Cart

- + **Description:** Allows users to select items, view detailed product information, and add them to their shopping cart.

+ **Inputs:**

- **Product Selection:** The user selects a product from the product list.
- **Action:** Clicks the "Add to Cart" button.

+ **Outputs:**

- A pop-up displaying product information with an "Add to Cart" button.
- Updated shopping cart content.

+ **Basic Course:**

Step	User Action	System Response
1	Selects a product from the list or details page.	Displays a pop-up with detailed product information, including specifications, price, and an "Add to Cart" button.
2	Clicks the "Add to Cart" button in the pop-up.	Validates the action and updates the shopping cart.

**Preconditions:**

- User is logged in or shopping as a guest.
- The system has available inventory of the selected product.

+ **Postconditions:**

- The selected product is added to the shopping cart.
- The shopping cart reflects the updated product quantity and total cost.

#### 4. View Cart

+ **Description:** Allows users to view items in their shopping cart, edit item quantities, or remove items.

+ **Inputs:**

- **Action:** Click on the cart icon.
- **Optional Actions:** Edit item quantities or remove items.

+ **Outputs:**

- Displays cart details, including items, quantities, individual prices, and the total cost.
- Updates the cart and total cost when quantities are edited or items are removed.

+ **Basic Course:**

Step	User Action	System Response
1	Clicks on the cart icon.	Displays the shopping cart page with a list of items, quantities, individual prices, and total cost.
2	Edits the quantity of an item.	Updates the cart with the new quantity and recalculates the total cost.
3	Removes an item from the cart.	Deletes the item from the cart and updates the total cost.
4	Reviews updated cart details.	Displays the modified cart contents and recalculates total cost.

+ **Preconditions:**

- User has added at least one item to the cart.

+ **Postconditions:**

- Updated cart reflects changes in item quantities or removed items.
- Total cost is recalculated based on the changes.

## 5. Checkout

+ **Description:** Allows users to provide shipping details and place an order successfully.

+ **Inputs:**

- **Action:** Click "Checkout" from the cart.
- **Shipping Details:** User-provided information such as name, address, phone number, and email (if not logged in). If logged in, details are prefilled but can be adjusted.

+ **Outputs:**

- Confirmation screen displaying the success message and order details.

+ **Basic Course:**

Step	User Action	System Response

1	Click "Checkout" from the cart.	Navigate to the shipping detail page, where shipping details are either prefilled (if logged in) or empty.
2	If not logged in, enter shipping details manually. If logged in, review and adjust the prefilled details.	The system validates the entered details.
3	Confirms the order.	Processes the order, saves the details, and displays a success confirmation screen with order details.

+ **Preconditions:**

- Users have items in their cart.
- The shipping form is accessible and functional.
- User is either logged in or not logged in.

+ **Postconditions:**

- The order is placed and saved in the system.
- A confirmation screen with order details is displayed, and the cart is cleared.

## Use Cases for Admin

### 1. Manage Products

+ **Description:** Admins can add, edit, or delete furniture products from the system.

+ **Inputs:**

- **Add Product:** Product name, description, price, category, stock quantity, image, etc.
- **Edit Product:** Modified product details such as price, stock quantity, description, etc.
- **Delete Product:** Select product to be removed from the database.

+ **Outputs:**

- A list of updated products in the admin panel.

+ **Basic Course:**

<b>Step</b>	<b>Admin Action</b>	<b>System Response</b>
1	Admin logs into the admin panel.	Displays the admin dashboard with management options.
2	Admin navigates to the product management section.	Displays a list of existing products with options to add, edit, or delete a product.
3	Admin selects an option to add, edit, or delete a product.	Displays a form to enter or modify product details, or prompts to confirm deletion.
4	Admin provides or modifies product details.	Validates and saves the product information to the product database.
5	System updates the product database.	Confirms the update, and displays the modified or newly added product on the admin panel.

+ **Preconditions:**

- Admin is logged into the admin panel.
- Admin has necessary permissions to manage products.

+ **Postconditions:**

- The product database is updated with the new, edited, or deleted product.
- The product list in the admin panel is refreshed to reflect the changes.

## 2. Monitor Orders

+ **Description:** Admins can view and update the status of customer orders.

+ **Inputs:**

- **Order Status Update:** Order status such as "Pending," "Shipped," "Delivered," etc.

+ **Outputs:**

- List of orders with updated statuses.

+ **Basic Course:**

Step	Admin Action	System Response
1	Admin logs into the admin panel.	Displays the admin dashboard with order management options.
2	Admin navigates to the order management section.	Displays a list of all customer orders with their current status.
3	Admin views a list of orders.	Displays order details, including order number, customer information, and order status.
4	Admin updates the status of a specific order.	Validates the new status (e.g., "Pending," "Shipped") and updates the order status in the system.
5	System saves the updated order status.	Confirms the update and displays the revised status in the order list.

+ **Preconditions:**

- Admin is logged into the admin panel.
- Admin has necessary permissions to view and manage orders.

+ **Postconditions:**

- Order status is updated in the system.
- The order list is updated to reflect the latest statuses.

### 3. Manage Orders

+ **Description:** Allows admins to update, track, delete orders, and modify user shipping information for customer orders.

+ **Inputs:**

- **Shipping Information Update:** User's name, address, contact number, and email.
- **Delete Order:** Admin selects an order to delete from the list.

+ **Basic Course:**

<b>Step</b>	<b>Admin Action</b>	<b>System Response</b>
1	Admin logs into the admin panel.	Displays the admin dashboard with options for managing orders.
2	Admin navigates to the order management section.	Displays a list of all customer orders with their current details and status.
3	Admin selects a specific order from the list.	Displays the order details, including customer info, items ordered, current order status, and shipping information.
4	Admin updates the user's shipping information (e.g., address, contact info).	Validates and saves the updated shipping information in the system.
5	Admin selects the option to delete the order (optional).	Prompts confirmation for deletion of the selected order.
6	Admin confirms deletion (optional).	Deletes the selected order from the system.
7	System removes the order and updates the order list.	Confirms the deletion or modified shipping information and updates the order list accordingly.

+ **Preconditions:**

- Admin is logged into the admin panel.
- Admin has necessary permissions to manage, update, or delete orders and modify user shipping information.
- The order exists in the system.

+ **Postconditions:**

- The order is updated with the modified shipping information, or the order is deleted.
- The order list is updated to reflect the changes.

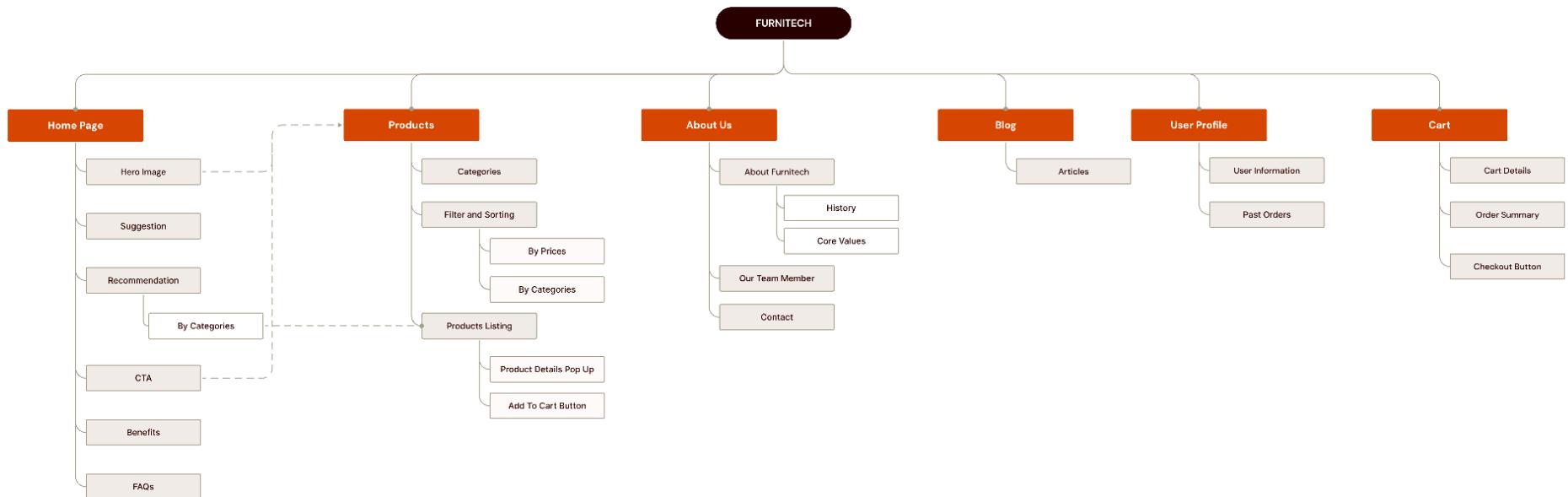
## 4. UI/UX Design

### a. Site Map and Main User Flow

#### i. Site Map

- The sitemap outlines the structure of the Furnitech website, detailing its main sections and sub-sections. Each section is designed to improve user navigation and interaction with the site. Below is an explanation of the key sections.

*Figure 3.2: The Site Map*



- + **Home Page (Landing Page):** The landing page serves as the main entry point and introduces the website's content.
  - Hero Image: Features a prominent banner to catch the user's attention.
  - Suggestions and Recommendations: Highlights furniture items by categories and provides recommendations for users.
  - Call to Action (CTA): Encourages users to take specific actions, such as exploring products or learning more about the store.
  - Benefits: Lists the advantages of purchasing from the store, enhancing customer trust.
  - FAQs: Addresses common queries to assist users and improve their shopping experience.
- + **Products Page:** This section allows users to explore the store's offerings in detail.
  - Categories: Group products into easily navigable categories.
  - Filter and Sorting: Provides tools to refine search results by price or category.
  - Products Listing: Displays available products, with the Product Details Pop-Up to provide detailed information about a product when selected.
  - Add to Cart Button: Allows users to add items to their shopping cart directly.
- + **About Us Page:** This section provides information about the project background and human.
  - About Furnitech: Explains the store's history and its mission.
  - Core Values: Showcases the principles guiding the business.
  - Our Team Members: Introduces the team responsible for operations.
  - Contact: Offers contact information for inquiries or support.
- + **Blog Page:** This section includes informative articles for users, such as furniture care tips, interior design inspiration, and more.
- + **User Profile Page:** This area is tailored for registered users to manage their accounts and view order history. If the user is not logged in, this page will be the login and sign up page.
  - User Information: Displays account details and allows updates.
  - Past Orders: Lists previously purchased items and their statuses.

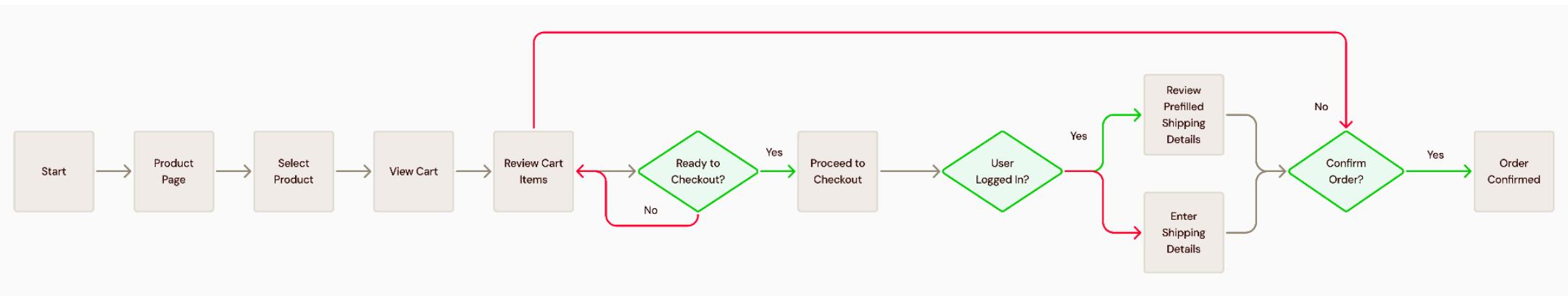
- + **Cart Section:** The cart section streamlines the checkout process for users.

- Cart Details: Displays all selected items, including quantities and prices.
- Order Summary: Provides a detailed breakdown of the total cost.
- Checkout Button: Redirects users to the checkout page to finalize their purchase.

## ii. Main User Flow for Purchasing Products

- The user flow diagram illustrates the step-by-step process for a customer to purchase a product on the Furniture Store Website. The flow ensures a seamless and intuitive experience, guiding users through key decision points. Below is an explanation of the flow.

*Figure 3.3: The Main User Flow for Purchasing Products*



- + **Start and Select Product in Product Page:** The user navigates to the Product Page and selects a specific product they are interested in. This action typically leads to a product details page or a pop-up with additional information. Users can then choose to add the product to their cart.
- + **View Cart:** The user accesses their shopping cart to review items added. This page displays:
  - A list of selected products.
  - Item quantities and total cost.

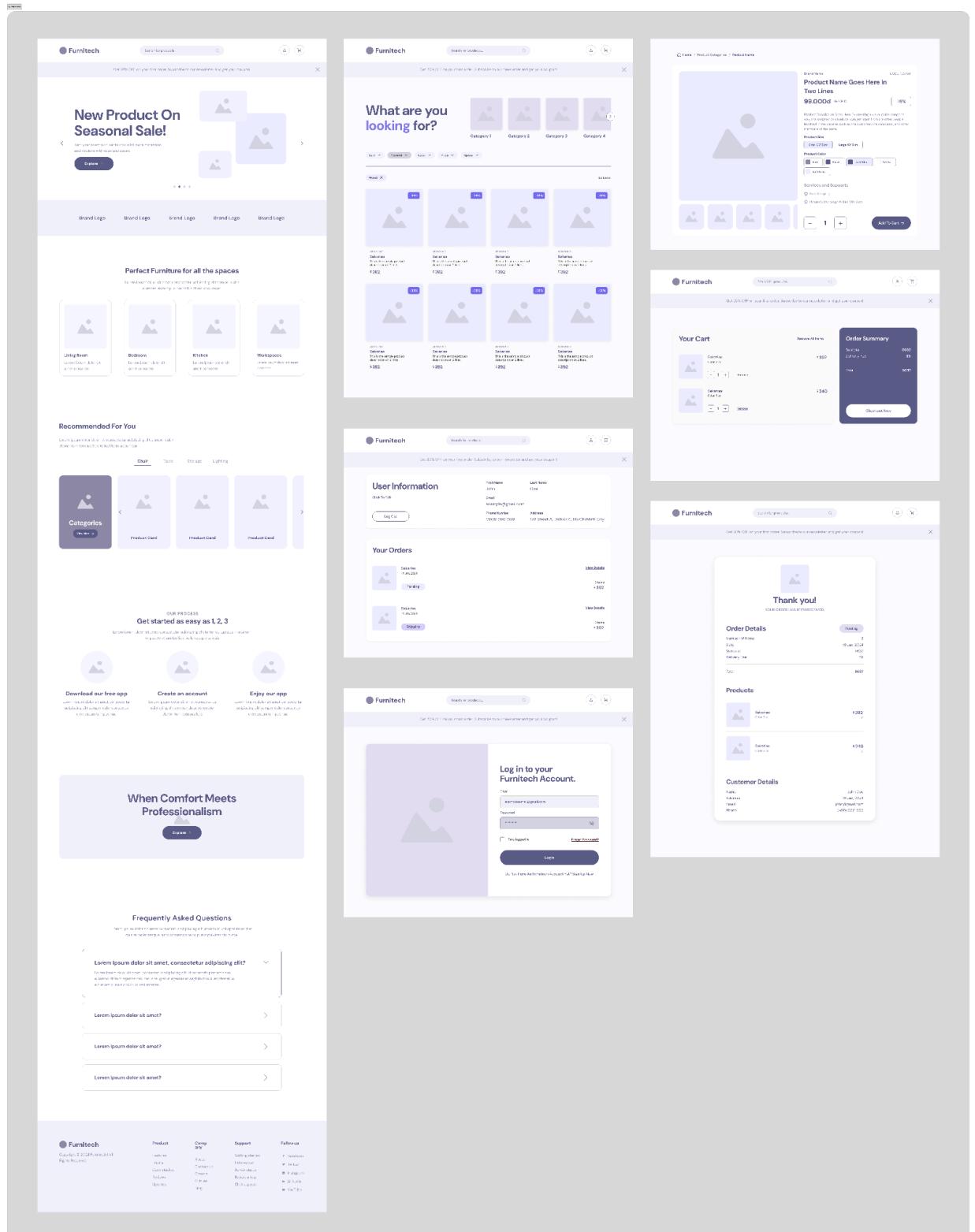
- Options to edit the quantity or remove products.
- + **Review Cart Items:** The user carefully reviews the cart items before deciding whether they are ready to proceed to checkout.
- + **Decision Point 1: Ready to Checkout?**
  - **Yes:** The user moves to the checkout process.
  - **No:** The user is redirected back to the product page to make changes or add more items.
- + **Proceed to Checkout:** Once the user decides to proceed, they move to the checkout page to finalize their order.
- + **User Logged In?**
  - **Yes:** If the user is logged in, their shipping information is pre-filled and can be reviewed or adjusted.
  - **No:** The user is prompted to enter their shipping details manually.
- + **Decision Point 2: Confirm Order?**
  - **Yes:** The order is finalized, and a confirmation screen is displayed.
  - **No:** The user is redirected back to make adjustments.
- + **Order Confirmed:** The system confirms the order, and the user receives a success screen with the order details.

## b. Wireframes

- The wireframes section provides a detailed visual representation of the Furniture Store website's structure and layout. These wireframes serve as the blueprint for the website, ensuring that both the user experience (UX) and user interface (UI) are intuitive and aligned with the project goals. They guide the development process by highlighting critical features such as content hierarchy and interactive components.
- In this section, each wireframe is accompanied by descriptions that explain its purpose and functionality, providing insight into how users will interact with the website. The wireframes are structured around key pages, such as the homepage,

product page, shopping cart, checkout, and user profile, offering a comprehensive overview of the site's architecture and intended user journey.

Figure 3.4: Wireframe for the main screens

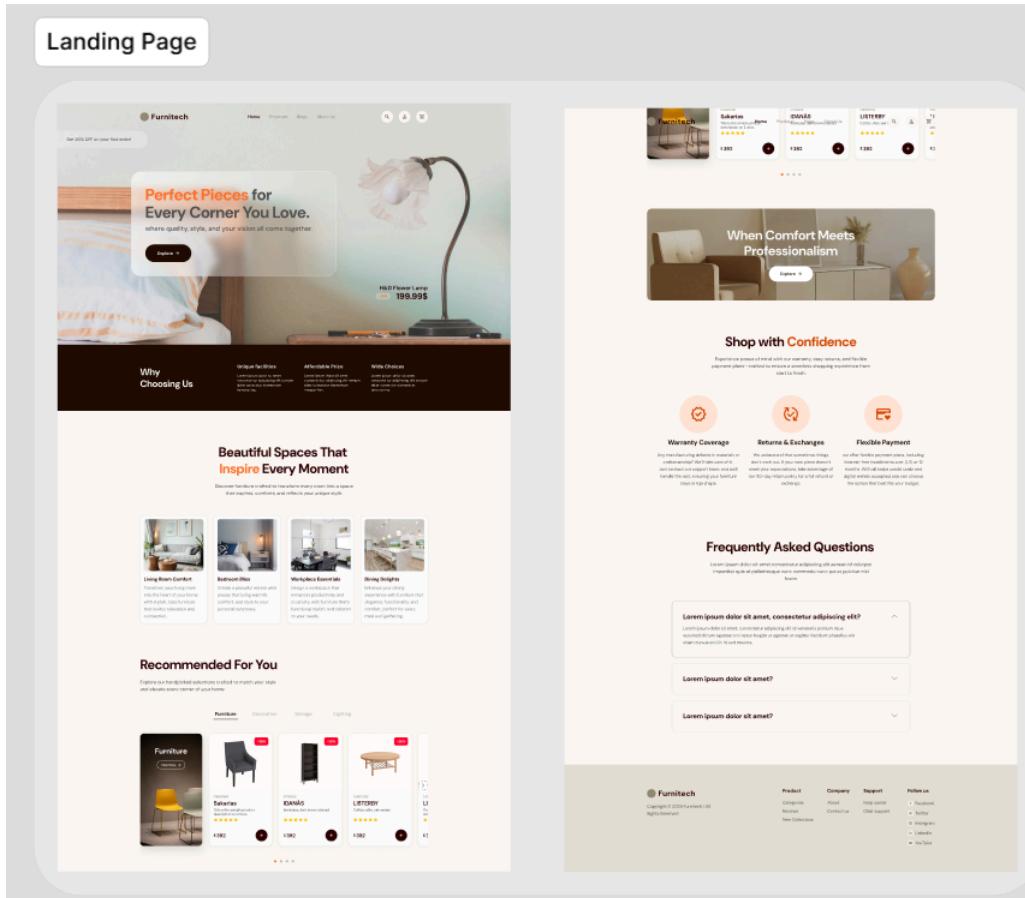


## c. Interface Modules

- The Interface Modules section provides a breakdown of the key components that form the user interface (UI) of the Furniture Store website. Each module is designed to ensure usability, functionality, and an engaging experience for users. The modules represent the core building blocks of the website, covering all primary pages and features that guide users through their journey, from browsing products to completing a purchase. Below is an overview of the main interface modules.

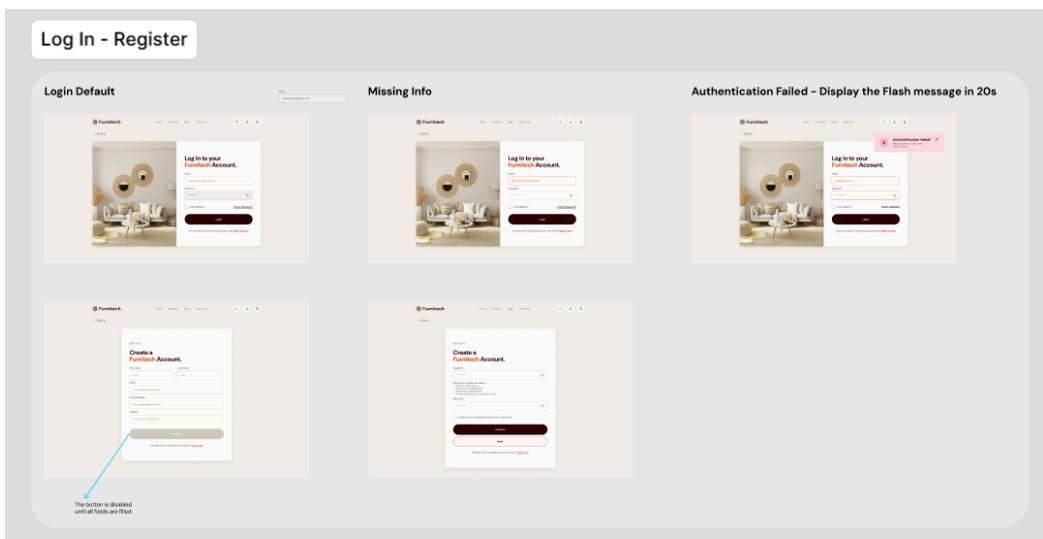
### + Landing Page:

Figure 3.5: Landing Page UI Design



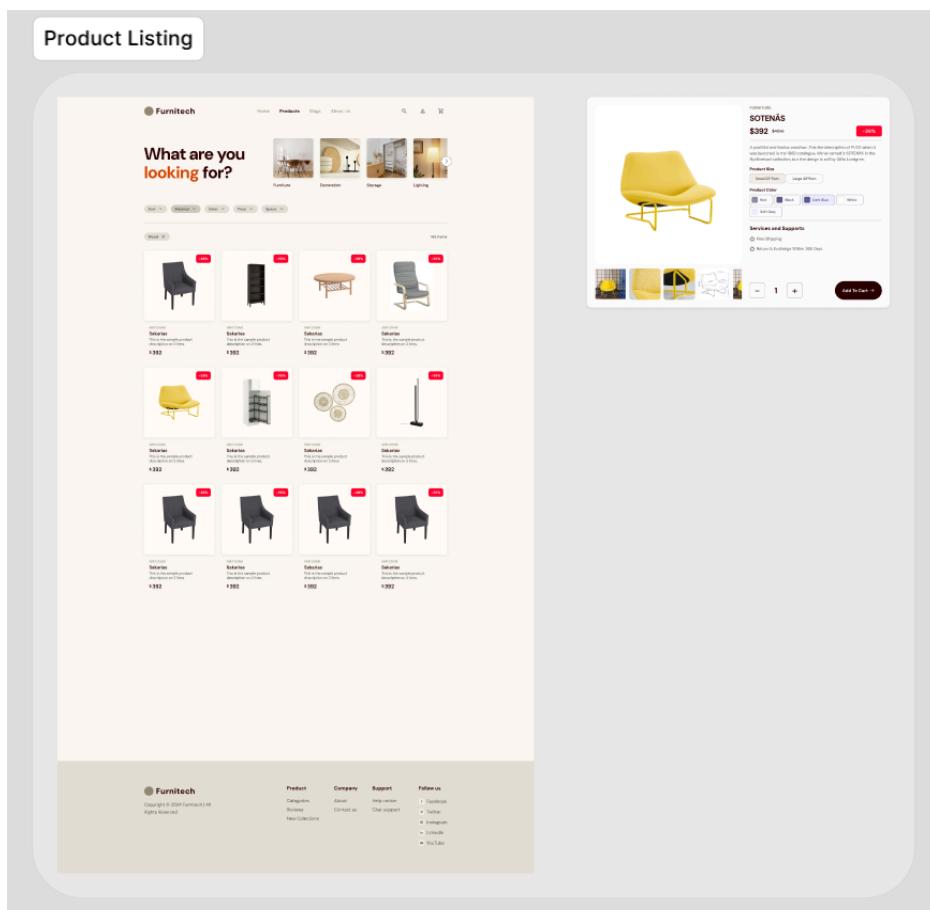
### + Login - Register:

*Figure 3.6: Login - Register UI Design*



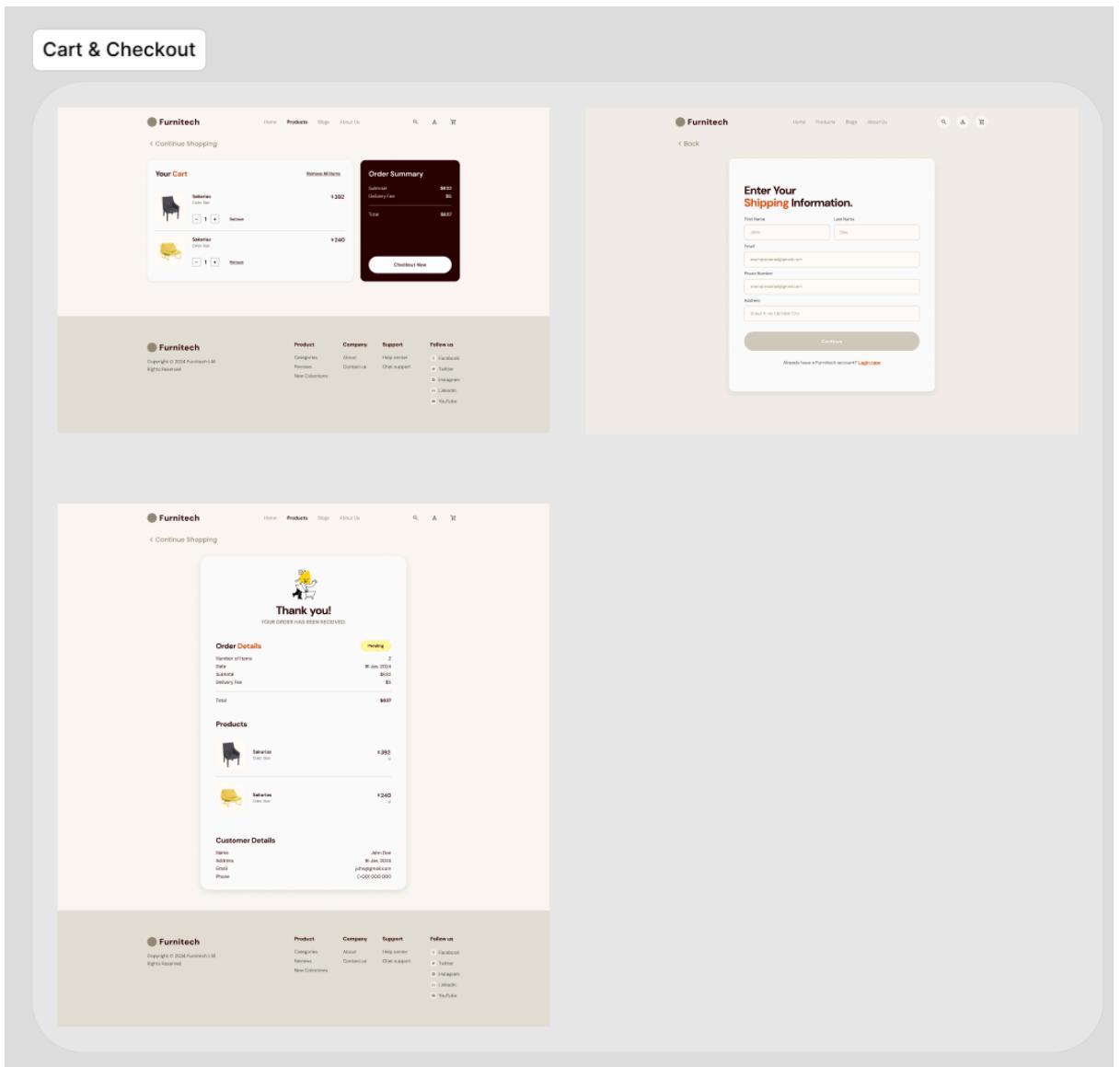
+ **Product Listing and Product Details Popup:**

*Figure 3.7: Product Listing Page UI Design*



+ **User Cart and Checkout Process:**

Figure 3.7: Cart and Checkout UI Design



### + Admin Panel:

Figure 3.8: Admin Panel UI Design

The image displays the Admin Panel UI Design, featuring three main sections:

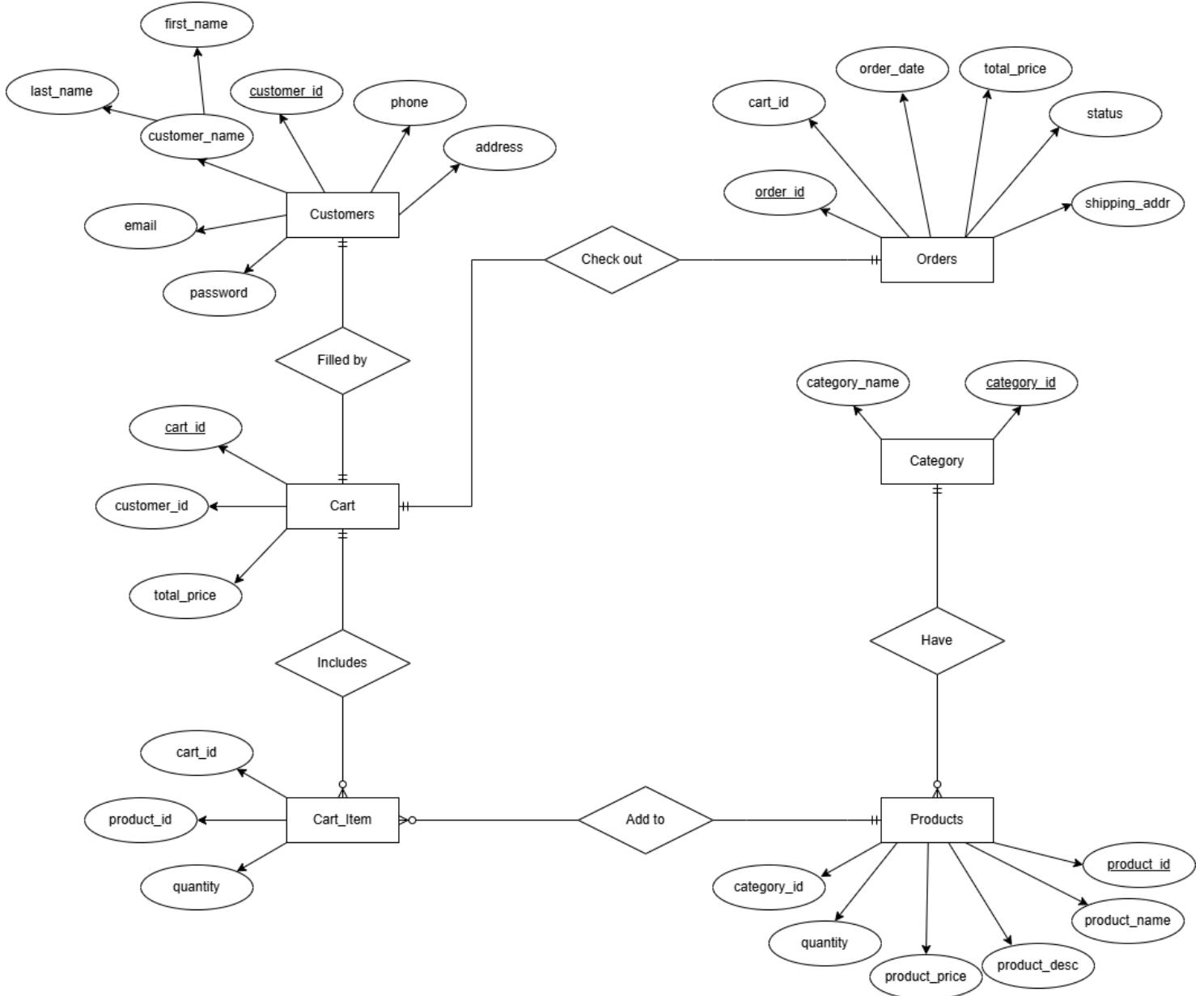
- Overview:** Shows a summary of total orders (12), shipped orders (5), and pending orders (7). It also includes a bar chart for "Most Selling Products" and a donut chart for "Product Sold By Categories".
- Manage Orders:** A list of pending orders. Each order item includes a thumbnail, product name, quantity, price, discount, and status (Pending). Total pending orders are listed as \$392.
- Manage Products:** A list of products under the "Sakuras" category. Each product item includes a thumbnail, name, price (\$392), quantity (40), discount (10%), and status (Pending). Total products are listed as \$392.

The interface includes a sidebar with navigation links (Overview, Manage Products, Manage Orders, Manage Users) and a header with a logo, user name, and log out button.

## 5. Database Design

### a. Entity Relationship Diagram (ERD)

Figure 3.9: Entity Relationship Diagram



- **Customers:** Fill a Cart and check out Orders.

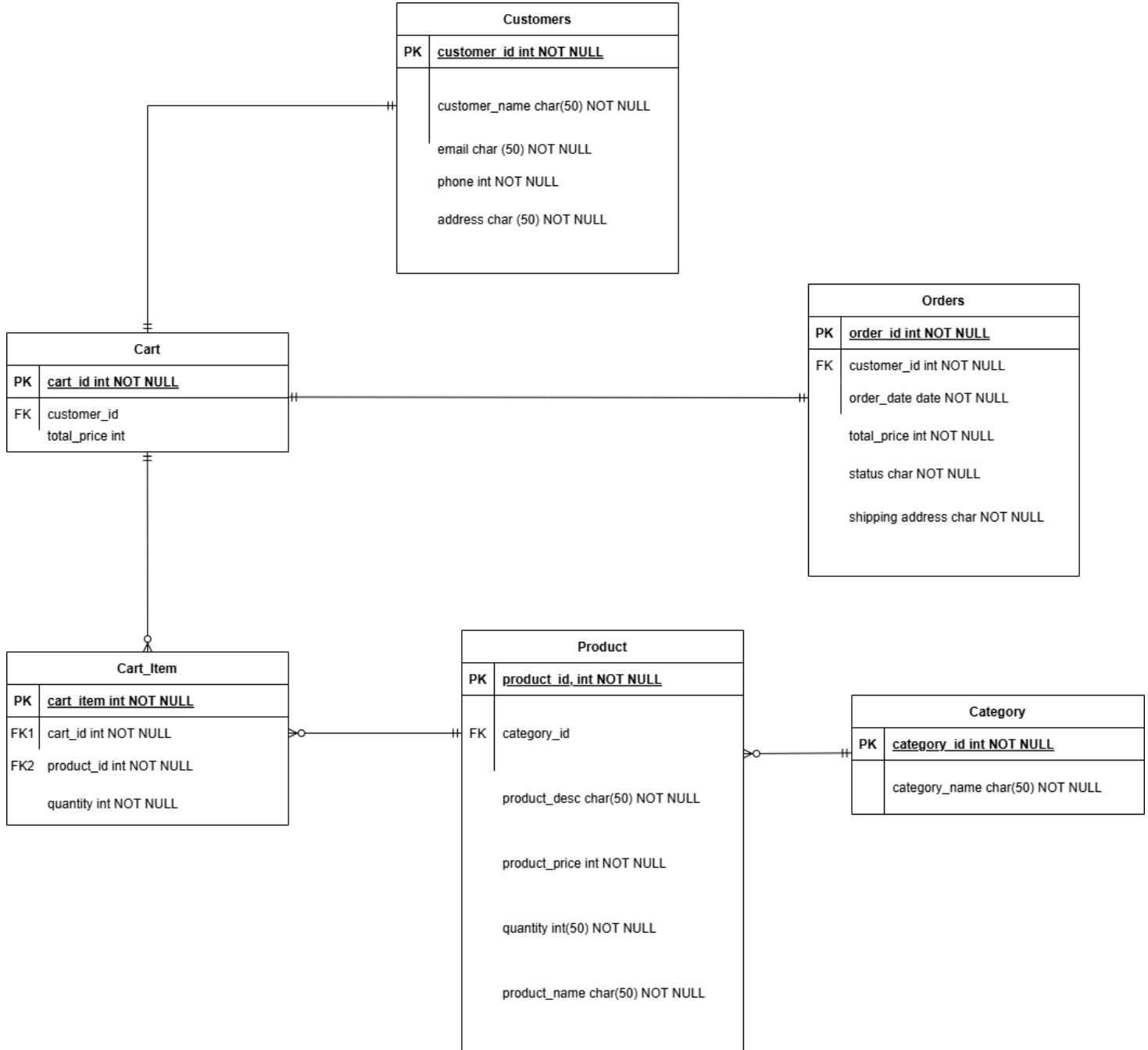
- + customer\_id: Unique identifier for each customer (primary key).
- + customer\_name, first\_name, last\_name: Name details of the customer.
- + email, password: Authentication details.

- + phone, address: Contact information.
  
- **Cart:** Filled by a customer, includes multiple Cart\_Item entries, can lead to an Order upon checkout.
  - + cart\_id: Unique identifier for each cart (primary key).
  - + customer\_id: References the customer who owns the cart.
  - + total\_price: Total value of items in the cart.
  
- **Cart\_item:** Represents items that a Cart includes, connects to Products (items being added to the cart).
  - + cart\_id: Foreign key referencing the cart.
  - + product\_id: Foreign key referencing the product added to the cart.
  - + quantity: Number of items of the product.
  
- **Orders:** Connects to a Cart through cart\_id, represents the Check out process for the cart.
  - + order\_id: Unique identifier for each order (primary key).
  - + cart\_id: References the cart associated with the order.
  - + total\_price: Total cost of the order.
  - + order\_date: Date when the order was placed.
  - + status: Status of the order (e.g., pending, shipped, delivered).
  - + shipping\_addr: Shipping address for the order.
  
- **Category:** Has multiple Products.
  - + category\_id: Unique identifier for each product category (primary key).
  - + category\_name: Name of the category (e.g., chairs, tables).
  
- **Products:** Belong to a Category, can be added to a Cart\_Item when a customer selects them.
  - + product\_id: Unique identifier for each product (primary key).
  - + category\_id: Foreign key referencing the product's category.

- + product\_name: Name of the product.
  - + product\_desc: Description of the product.
  - + product\_price: Price of the product.
  - + quantity: Available stock or inventory.
- **Relationships Between Entities:**
- + **Customers and Cart:** A customer can have one or more carts.
  - + **Cart and Cart\_Item:** A cart can include multiple cart items.  
Each Cart\_Item references a product.
  - + **Cart\_Item and Products:** Each cart item points to a product being added to the cart.
  - + **Cart and Orders:** Once a customer checks out, the cart details are converted into an order.
  - + **Products and Category:** Each product belongs to a specific category.
- This ERD represents a typical workflow for an e-commerce system where customers can browse products, add them to their cart, and place orders. The relationships are well-structured to handle core functionalities like product categorization, order management, and shopping cart features.

## b. Database Schema

Figure 3.10: Database schema



### - Customers:

- + customer\_id: Unique identifier for each customer (Primary Key).
- + customer\_name: Name of the customer (not null).
- + email: Customer's email address (not null).

- + phone: Customer's phone number (not null).
- + address: Shipping address for the customer (not null).

- **Cart:**

- + cart\_id: Unique identifier for each shopping cart (Primary Key).
- + customer\_id: Reference to the customer owning the cart (Foreign Key).
- + total\_price: Total price of items in the cart.

- **Cart\_item:**

- + cart\_item\_id: Unique identifier for each item in a cart (Primary Key).
- + cart\_id: Reference to the associated shopping cart (Foreign Key).
- + product\_id: Reference to the product in the cart (Foreign Key).
- + quantity: Number of units of the product.

- **Orders:**

- + order\_id: Unique identifier for each order (Primary Key).
- + customer\_id: Reference to the customer placing the order (Foreign Key).
- + order\_date: Date the order was placed.
- + total\_price: Total price of the order.
- + status: Current status of the order (e.g., pending, shipped).
- + shipping\_address: Address where the order will be shipped.

- **Category:**

- + category\_id: Unique identifier for each product category (Primary Key).
- + category\_name: Name of the category.

- **Products:**

- + product\_id: Unique identifier for each product (Primary Key).
- + product\_name: Name of the product.
- + product\_desc: Description of the product.
- + product\_price: Price of the product.
- + quantity: Available quantity of the product.
- + category\_id: Reference to the product category (Foreign Key).

# CHAPTER 4: IMPLEMENTATION

## 1. Login and Authentication

- The **Login and Register** module is a critical component of the website's back-end functionality. It ensures secure user authentication and account management, enabling both new user registration and login for existing users. This section focuses on the development aspects of these features, including logic and workflows.

### + Login Functionality

#### - Back-End Process:

- + When a user submits their login email and password, the system verifies the provided email/username and password combination against the encrypted data stored in the database.
- + If the credentials match, the system generates an authentication token and initiates a user session.

#### - Core Development Features:

- + Session Management: The system establishes a secure session using tokens to keep users logged in while browsing. Sessions automatically expire after a set duration for security.
- + Error Handling: The module handles incorrect login attempts by displaying appropriate error messages.

### + Register Functionality

#### - Back-End Process:

- + When a user submits registration details, the system validates the input (e.g., checking email format and ensuring password strength).
- + The password is hashed and stored securely in the database using encryption algorithms.

#### - Core Development Features:

- + Validation: Enforces client-side and server-side validation to ensure data integrity (e.g., verifying required fields, password confirmation).

- + Database Integration: Stores user information (name, email, hashed password) in the database, maintaining referential integrity for future transactions.
- + Registration Success Workflow: After successful registration, the system redirects the user to the landing page automatically.

*Figure 4.1: Login and Sign Up Front-end*

The image consists of two screenshots of the Furnitech website. The top screenshot shows the login page. It features a background image of a living room with a sofa, two circular mirrors on the wall, and a small round coffee table. Overlaid on this is a white login form with the heading "Login to your Furnitech account." It contains fields for "Email" (with placeholder "Enter your email") and "Password" (with placeholder "Enter password" and a visibility icon). Below these is a "Login" button. At the bottom of the form, there is a link "Don't have an account? [Register here](#)". The bottom screenshot shows the sign-up page. It has a similar layout with a background image of a living room. The heading is "Create a Furnitech Account.". It includes fields for "First Name" and "Last Name" (each in its own box), "Email" (placeholder "Enter your email"), "Phone Number" (placeholder "Enter your phone number"), and "Address" (placeholder "Enter your address"). Below these is a "Next" button. At the bottom, there is a link "Already have a Furnitech account? [Login now](#)". Both screenshots show a navigation bar at the top with links for Home, Products, Blogs, About Us, and icons for user profile and shopping cart.

#### + Integration with Other Modules

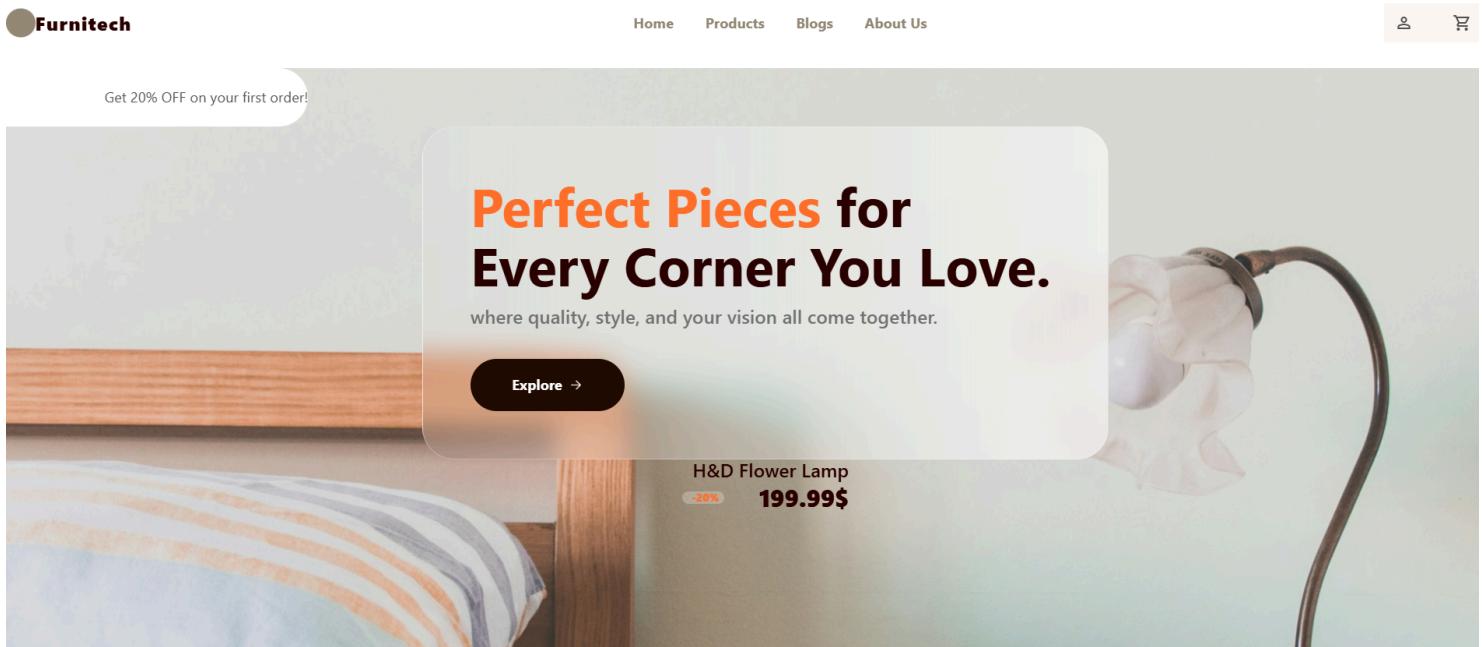
- **User Profile:** Data from the Login and Register module is used to initialize and update user profiles.

- **Checkout Process:** Logged-in users benefit from prefilled shipping details, streamlining the checkout process.

## 2. Landing Page

- The Landing Page is the entry point of the website, designed to provide users with an overview of the platform's offerings and guide them seamlessly to their desired sections. This section details the development process, including features, workflows, and functionality implementation.
- + **Back-End Integration**
  - Database Integration: Product recommendations and featured categories are fetched from the product database.
- + **Optimization:**
  - Lazy Loading: Images and assets for off-screen elements load only when the user scrolls to them, reducing initial page load time.

*Figure 4.2: Landing Page Front-end*



The screenshot shows the homepage of the Furnitech website. At the top, there is a navigation bar with icons for search, refresh, and user account, followed by the URL 'localhost:3000'. Below the navigation bar is the Furnitech logo and a main title 'Beautiful Spaces That Inspire Every Moment' in large, bold letters. A subtitle below it reads 'Discover furniture crafted to transform every room into a space that inspires, comforts, and reflects your unique style.' There are four sections with images and descriptions: 'Living Room Comfort', 'Bedroom Bliss', 'Workplace Essentials', and 'Dining Delights'.

**Living Room Comfort**  
Transform your living room into the heart of your home with stylish, cozy furniture that invites relaxation and connection.

**Bedroom Bliss**  
Create a peaceful retreat with pieces that bring warmth, comfort, and style to your personal sanctuary.

**Workplace Essentials**  
Design a workspace that enhances productivity and creativity, with furniture that's functional, stylish, and tailored to your needs.

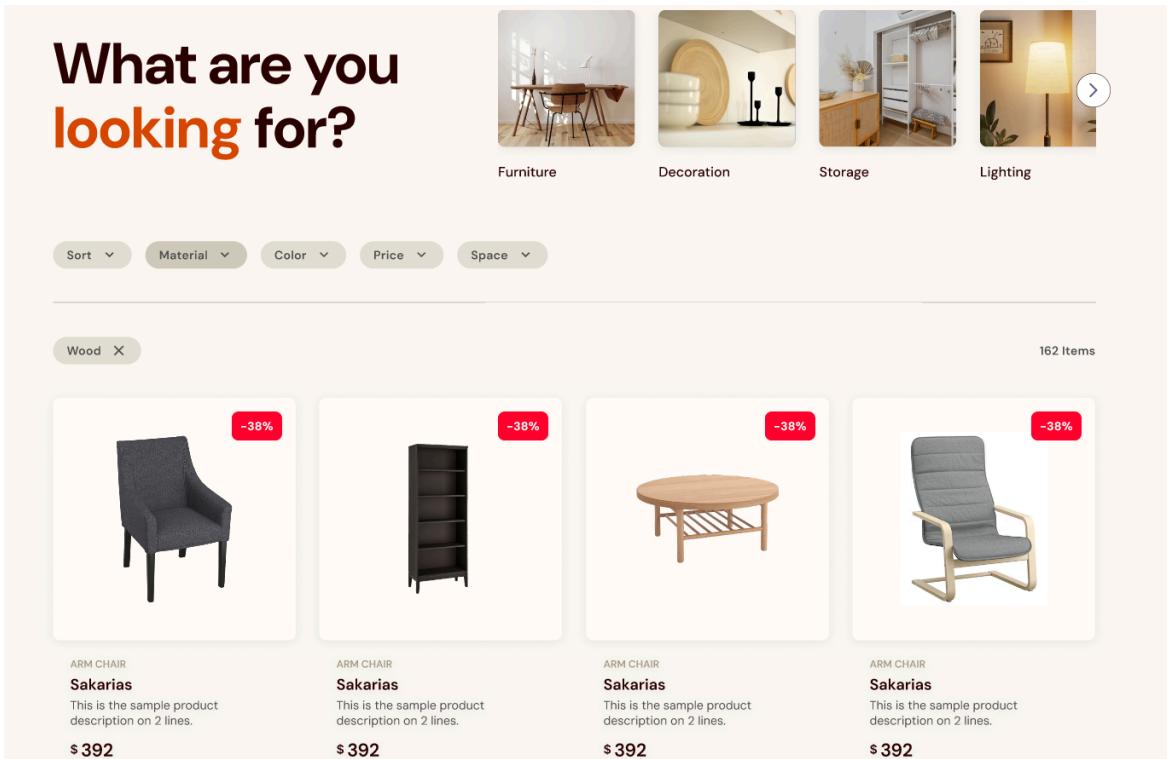
**Dining Delights**  
Enhance your dining experience with furniture that's elegance, functionality, and comfort, perfect for every meal and gathering.

### 3. Product Listing

- The Product Page is one of the most critical elements of the website, as it is where users will browse and explore the various furniture products offered. The development of this page focuses on creating an intuitive, responsive, and fast-loading interface that allows users to filter, and select products with ease.
- + **Back-End Integration**
  - Database Queries: The back-end processes user input from search and filter options, querying the product database to return matching results.
  - APIs: The front end side sends requests to the back end via API to retrieve filtered product data without needing to reload the page.
  - RESTful APIs handle queries related to product filtering, search, and display, ensuring smooth communication between the front end and back end.

- Product Detail Pop-Up: When a user clicks on a product, a pop-up window displays more detailed information, such as product specifications, images, and the option to add it to the cart. This feature enhances the user experience by providing more details without navigating to a different page, keeping users engaged on the product page.

*Figure 4.3: Product Listing Front-end*

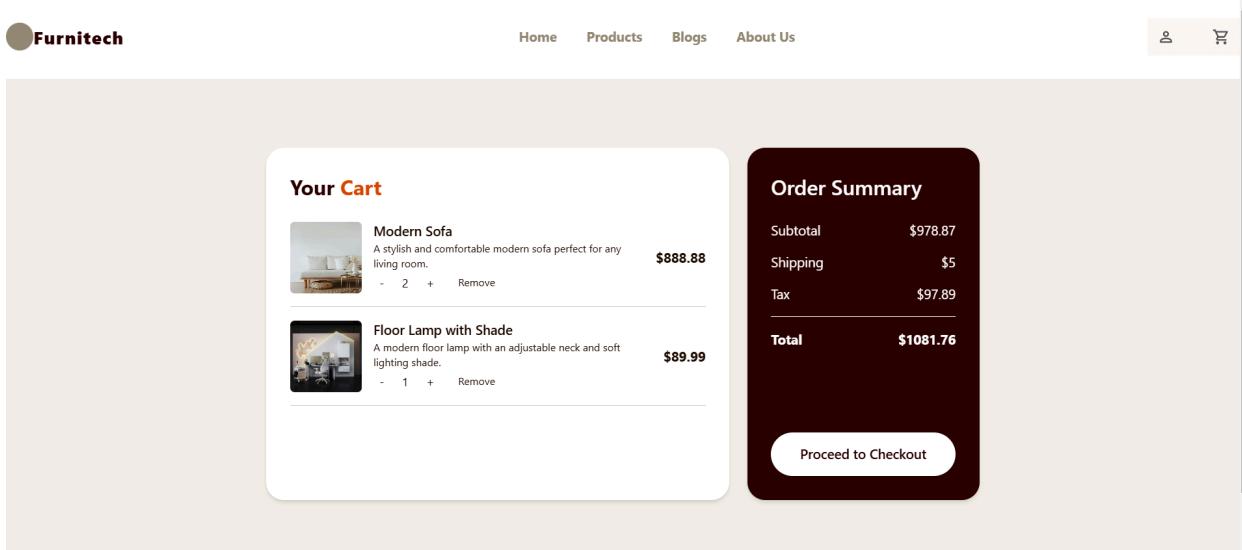


## 4. Product Cart

- The Product Cart module is a critical feature of the website, enabling users to manage the products they wish to purchase. The development focuses on ensuring a smooth and intuitive cart experience by allowing users to view, edit, and finalize the items in their cart.
- + Back-End Integration
  - Database Interaction: The cart data (items, quantities, and prices) is stored in the database for logged-in users, ensuring their cart is persistent across sessions. For guest users, cart data is stored temporarily in browser cookies or local storage.
  - API Calls are used for cart operations, such as:

- Adding products to the cart.
- Updating product quantities.
- Removing items from the cart.
- These API endpoints interact with the server to fetch and save updated cart information without a full-page reload.
- The cart summary displays:
  - Subtotal of all items.
  - Applicable taxes or discounts.
  - Final total cost.
  - It dynamically updates based on user actions like quantity changes or item removal.

*Figure 4.4: Product Cart Front-end*



## 5. Checkout

- The Checkout Module is the final step in the purchasing process, where users confirm their order and provide shipping details. The development of this module prioritizes efficiency and user convenience to ensure a smooth transition from cart to purchase completion.

### + Back-End Integration

- Order Processing: Once the user submits the form, the back-end validates the input data and checks stock availability for the items in the order.

- Database Updates: The system saves the order information, including user details, shipping address, items purchased, and payment status, into the database.

+ **Shipping Information**

- Logged-In Users: The system pre-fills the form with saved shipping information from the user's profile, which they can update if needed.
- Guest Users: Guest users must provide all required details manually before proceeding to order placement.

*Figure 4.5: Shipping Address Confirmation Front-end*

Please confirm your  
Shipping Information.

First Name

Ky

Last Name

Phuong

Email

kyphuong0414@gmail.com

Phone Number

0972423427

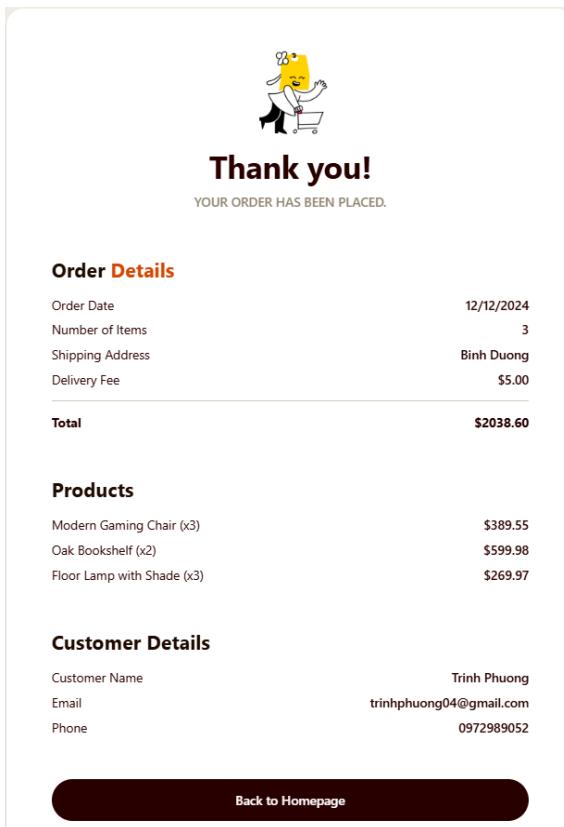
Address

Enter your address

Confirm Order

- + **Order Confirmation:** After the order is successfully placed, the confirmed order screen will be displayed, this section includes a detailed list of items being purchased, subtotal, taxes, final total cost and a summary of the selected shipping information.

Figure 4.5: Order Successful Front-end



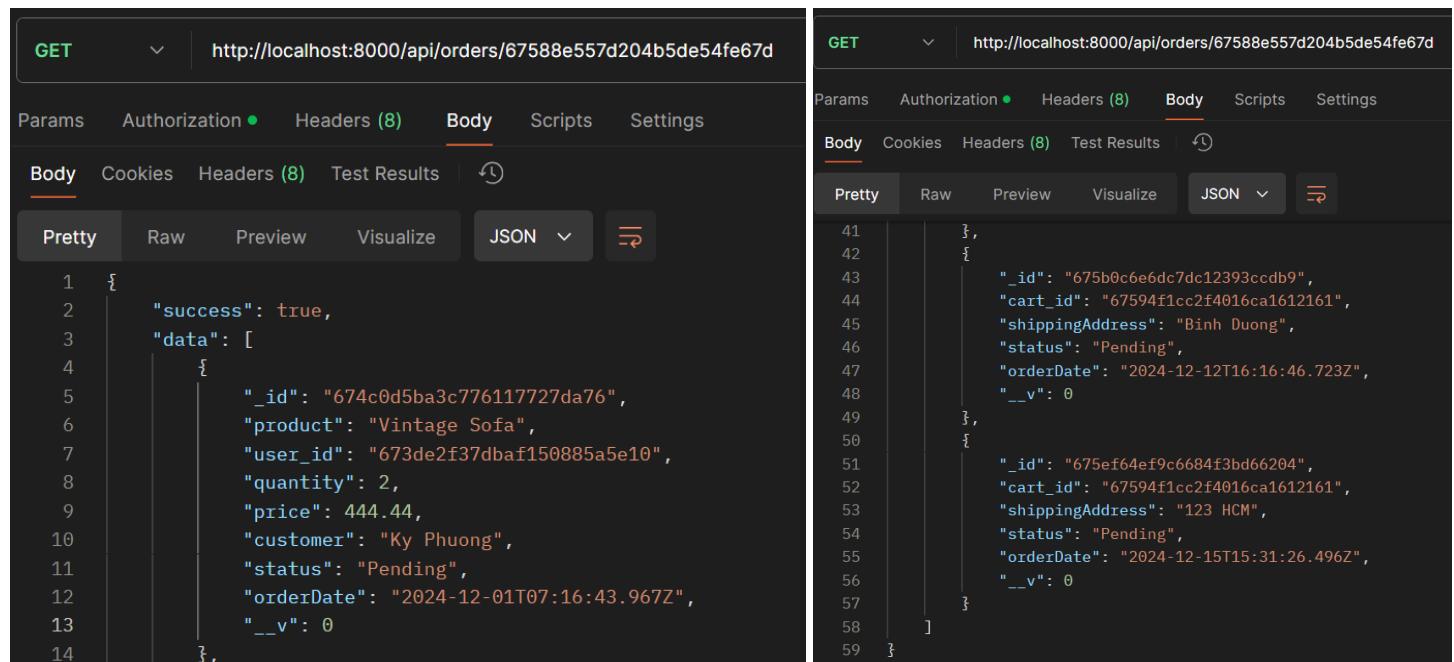
## 6. Order Management

- The order management system is designed to handle all aspects of customer orders, ensuring smooth interaction between the user, cart, and product entities. The system is built on a structured data model where Orders serve as the central component, linking information such as product details, user IDs, quantities, and prices.
- When a user places an order, the system processes the input data, such as product name, quantity, price, and shipping address, and creates a new entry in the database with the default status as "Pending". Each order is associated with a unique cart ID and linked back to the user. The database, implemented using MongoDB, efficiently stores and organizes the data, ensuring quick retrieval and updates.
- For order retrieval, the system allows fetching all orders belonging to a specific user through their user ID or querying by order ID. This feature provides the flexibility to display comprehensive order details, including the order date, status, shipping address, and total price.

## + Back-End Integration:

- Order Creation: When a user places an order from the front-end, the request is sent via a POST API to the back end. The server processes input such as product details, quantity, price, customer name, and shipping address, then stores the data in the database. A unique order ID is automatically generated for tracking.
- Order Retrieval: To fetch orders, the front end sends a GET API request. The server queries the MongoDB database using filters like user ID or order ID to return matching orders. The data is then sent back in JSON format, including all necessary order information such as product name, quantity, price, and status.

*Figure 4.6: API for Order Management*



The image shows two side-by-side Postman API requests for order management. Both requests are GET methods with the URL `http://localhost:8000/api/orders/67588e557d204b5de54fe67d`.

**Request 1 (Left):**

- Params: None
- Authorization: Bearer [REDACTED]
- Headers (8): None
- Body** (selected):
  - Pretty: `1 {  
2 "success": true,  
3 "data": [  
4 {  
5 "_id": "674c0d5ba3c776117727da76",  
6 "product": "Vintage Sofa",  
7 "user_id": "673de2f37dbaf150885a5e10",  
8 "quantity": 2,  
9 "price": 444.44,  
10 "customer": "Ky Phuong",  
11 "status": "Pending",  
12 "orderDate": "2024-12-01T07:16:43.967Z",  
13 " __v": 0  
14 },  
15 ]`
  - Raw: None
  - Preview: None
  - Visualize: None
  - JSON: Selected
- Scripts: None
- Settings: None

**Request 2 (Right):**

- Params: None
- Authorization: Bearer [REDACTED]
- Headers (8): None
- Body** (selected):
  - Pretty: `41 },  
42 {  
43 "_id": "675b0c6e6dc7dc12393ccdb9",  
44 "cart_id": "67594f1cc2f4016ca1612161",  
45 "shippingAddress": "Binh Duong",  
46 "status": "Pending",  
47 "orderDate": "2024-12-12T16:16:46.723Z",  
48 " __v": 0  
49 },  
50 {  
51 "_id": "675ef64ef9c6684f3bd66204",  
52 "cart_id": "67594f1cc2f4016ca1612161",  
53 "shippingAddress": "123 HCM",  
54 "status": "Pending",  
55 "orderDate": "2024-12-15T15:31:26.496Z",  
56 " __v": 0  
57 ]  
58 }  
59 }`
  - Raw: None
  - Preview: None
  - Visualize: None
  - JSON: Selected
- Scripts: None
- Settings: None

# CHAPTER 5: TESTING

## 1. Test case 1

Table 5.1: Test case 1

Test Case ID: 1
Project Name: Furniture E-commerce Website
Module Name: Login
Created By: Furnitech
Created Date: 28/10/2024
Reviewed By: Furnitech
Reviewed Date: 12/12/2024

Steps	Test steps	Test Case Description	Pre-Conditions	Test Data	Post-Conditions	Expected Results	Status
1	Enter valid email	Verify user login with valid and invalid credentials	- The user is on the login page. - The user has already registered an account by testing valid login credentials.	email:kyphuong0414@gmail.com password: phuong04142003	- If the user logged in successfully, they are on the homepage/dashboard. - If the credentials were invalid, the user remains on the login page with an error message visible.	- If valid credentials: Redirected to homepage/dashboard - If invalid credentials: Error message "Invalid email/username or password." displayed.	Pass
2	Enter valid password.		Pass				
3	Click the Login button.		Pass				

## 2. Test case 2

*Table 5.2: Test case 2*

Test Case ID: 2
Project Name: Furniture E-commerce Website
Module Name: Cart Management
Created By: Furnitech
Created Date: 11/11/2024
Reviewed By: Furnitech
Reviewed Date: 12/12/2024

Steps	Test steps	Test Case Description	Pre-Conditions	Test Data	Post-Conditions	Expected Results	Status
1	Navigate to the product page.	Verify adding items to cart works correctly	<ul style="list-style-type: none"> <li>- The user is logged into the system.</li> <li>The user is on the product page of the desired item to add to the cart.</li> </ul>	product_id :6756d8a3 0dc9a3ca1 a8815dc quantity: 2 price: 399.99	<ul style="list-style-type: none"> <li>- The selected product(s) with the specified quantity is added to the shopping cart.</li> <li>The total price is updated accordingly in the shopping cart view.</li> </ul>	<ul style="list-style-type: none"> <li>- Product is visible in the cart with the correct quantity.</li> <li>- Total price is accurate (e.g., price * 2).</li> </ul>	Pass
2	Select quantity (e.g., 2) and click Add to Cart..						Pass
3	Go to the shopping cart page..						Pass
4	Verify items and total price in the cart.						Pass

# CHAPTER 6: DISCUSSION AND EVALUATION

## 1. Discussion

- The development of the Furnitech furniture e-commerce website highlights several key aspects of web development and project management. This section discusses the challenges encountered, the solutions implemented, and the overall evaluation of the project.

### a. Challenges Faced

- **Scope and Features:** Determining the right balance between simplicity and functionality was a challenge. The project aimed to include essential features like user authentication, product browsing, and order management without overextending its scope.
- **Time Management:** Given the academic timeline, delivering a fully functional website required efficient time allocation for the design, implementation, and testing phases.
- **Security and Data Management:** The project did not incorporate methods to ensure data security, such as password encryption, input validation, or secure communication protocols, which are essential for a real-world e-commerce platform.

### b. Solutions and Current Implementation

- **Simplified Development:** The project focused on building core features with basic interactivity.
- **User and Admin Role Separation:** Functionality was divided to allow users to shop and admins to manage the platform, enabling clear workflows and modular design.

## 2. Evaluation

- The platform has a solid foundation with a modern tech stack and core e-commerce features, ensuring usability and scalability. However, to remain competitive, it must address weaknesses such as the absence of advanced features and limited user engagement tools. By seizing opportunities for feature expansion, global market reach, and operational improvements, the platform can position itself as a robust solution. At the same time, overcoming

external threats like competition, security risks, and scalability challenges will be essential for long-term success.

#### a. Strengths

- The furniture e-commerce website is built on a modern technology stack using React.js for the frontend, Node.js for the backend, and MongoDB for the database. This stack provides a highly responsive user interface, ensures efficient server-client communication, and offers a scalable data structure. Combined, these technologies enable smooth operations for key functionalities like user authentication, product management, shopping cart handling, and order placement.
- Another major strength is the simplicity and usability of the platform. The website offers a clean, intuitive interface that ensures users can easily navigate through products, manage their carts, and place orders. This focus on usability lays a strong foundation for building customer trust and satisfaction.

#### b. Weaknesses

- While the core functionalities are implemented, the platform still lacks advanced features commonly found in modern e-commerce systems. There is no personalized recommendation system, advanced product filtering, or real-time order tracking, which limits the user experience. Additionally, the absence of an admin dashboard reduces operational efficiency, making it harder for administrators to monitor orders, sales, and inventory effectively.
- From a performance standpoint, the current system may face challenges in handling a large number of concurrent users or an extensive product catalog. This could lead to slowdowns or inconsistencies under heavy load, highlighting a need for optimization.

### c. Opportunities

- The platform has significant opportunities to enhance its functionality and market reach. By introducing AI-driven features like product recommendations and integrating multiple payment gateways, the system can provide a more personalized and seamless shopping experience. Additionally, features such as real-time order tracking and mobile optimization can further increase convenience and accessibility, catering to a wider audience.
- The growing demand for online furniture shopping provides a favorable market opportunity. Enhancing the platform with advanced analytics tools, SEO integration, and marketing campaigns will help attract new customers and retain existing ones. Partnerships with logistics providers and third-party tools will streamline operations and improve customer satisfaction.

### d. Threats

- The platform operates in a highly competitive e-commerce market, where established players offer advanced features and loyalty programs. Meeting user expectations for features like faster delivery, personalized recommendations, and multiple payment options becomes challenging in such an environment.
- Security risks are another concern, as e-commerce platforms are prime targets for cyberattacks. Ensuring user data protection requires continuous security updates and monitoring.
- Technological advancements pose additional threats. The platform must stay updated with the latest tools and frameworks to remain relevant, which may require significant time and resources. Lastly, economic factors and potential disruptions in the supply chain could affect customer purchasing power, product availability, and overall platform success.

# CHAPTER 7: CONCLUSION AND FUTURE WORK

## 1. Conclusion

- The furniture e-commerce website developed using React.js, Node.js, and MongoDB has successfully achieved its core objectives of creating a secure, efficient, and user-friendly online shopping platform. The implemented functionalities, including customer authentication, shopping cart management, order processing, and product and category organization, provide a solid foundation for e-commerce operations. The use of React.js ensures a responsive and dynamic user interface, while Node.js facilitates seamless server-side communication, and MongoDB offers a flexible and scalable database for managing product, user, and transaction data. Despite its achievements, the project faced challenges such as optimizing database queries, ensuring smooth integration between front-end and back-end, and maintaining security during transactions. These challenges were addressed through modular code architecture, secure API development, and careful testing. The current version of the platform is a reliable solution for basic e-commerce needs, but there is room for improvement to enhance functionality, scalability, and user engagement.

## 2. Future work

- Future work on the platform will focus on expanding its features and improving user engagement. Key areas for enhancement include implementing a recommendation system to provide personalized product suggestions, integrating multiple payment gateways for a smoother checkout experience, and adding real-time order tracking. Additionally, performance optimization, advanced search capabilities, and an analytics dashboard for administrators can further improve usability and operational efficiency.
  - + **Personalization Features:** Integrating a recommendation system powered by machine learning to provide personalized product suggestions based on user preferences and browsing history.

- + **Payment Integration:** Adding support for multiple secure payment gateways, including digital wallets, credit cards, and cryptocurrency, to cater to diverse customer preferences.
- + **Order Management Enhancements:** Incorporating real-time order tracking to provide customers with visibility into their shipment status and delivery timelines.
- + **Performance Optimization:** Optimizing server and database queries for faster response times and better scalability, especially during high-traffic periods.
- + **Advanced Search and Filtering:** Implementing intelligent search and filtering options, such as voice search, auto-complete, and multi-criteria filtering, to improve product discovery.
- + **Analytics Dashboard:** Developing an analytics dashboard for administrators to monitor key metrics such as sales performance, user activity, and inventory levels, enabling data-driven decisions.
- + **Security Enhancements:** Strengthening security measures by incorporating advanced authentication mechanisms like two-factor authentication and regular vulnerability assessments.

## REFERENCES

1. YouTube. (n.d.). Full stack e-commerce website using react JS, mongodb, express JS & Node JS 2024. YouTube. [Full Stack E-Commerce Website Using React JS, MongoDB, Express Js & Node JS 2024 - YouTube](#). Accessed 12 October 2024.
2. W3schools.com. W3Schools Online Web Tutorials. (n.d.). [Node.js Tutorial](#). Accessed 10 October 2024.
3. YouTube. (n.d.-a). Completed: Modern ecommerce website with react, nodejs, Express, Mongoose DB & Tailwindcss, paypal payment. YouTube. <https://www.youtube.com/playlist?list=PLRQ1njZomKvF9Xewm0BRR1M-DLE-joGeQ>. Accessed 22 October 2024.
4. Website Wireframes UI Kit | BRIX Templates. Figma. [Website Wireframes UI Kit | BRIX Templates](#). Accessed 20 October 2024.