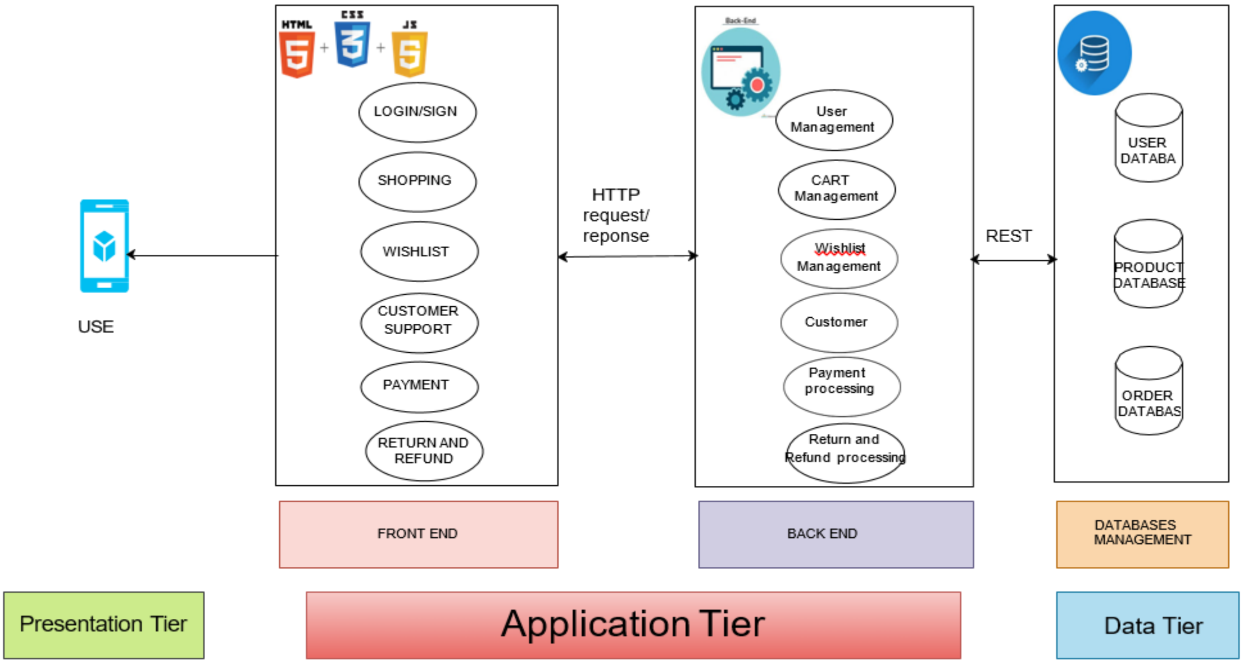
eCommerce Shopping Website

Tech Ninjas

Structure of System:

SYSTEM ARCHITECTURE FOR ECOMMERCE APPLICATION:



TOP-LEVEL DESCRIPTION FOR THE SYSTEM:

1. Presentation Tier (User Interaction Layer):

- **Login/Sign Up**: Provides user authentication authorization functionality. It allows users to create accounts or log in to access their shopping information.
- **Shopping Cart**: Enables users to add, view, modify products they intend to purchase. This interface typically shows product details, prices, total amounts.
- **Wishlist**: Allows users to save products for future consideration, which they might not want to purchase immediately.

- **Customer Support**: Provides way for users to get help regarding products, orders, or other inquiries, usually through chat interface or help ticket system.

2. Application Tier (Business Logic Layer):

This was core of application, handling requests from presentation tier processing them using underlying data. Key components include:

- **REST API**: Acts as communication bridge between front-end (presentation tier) back-end (application tier). It handles HTTP requests (e.g., GET, POST) returns appropriate responses, ensuring smooth interaction between users system.
- **Cart Management**: Manages user's shopping cart operations, including adding, removing, updating product quantities.
- **Wishlist Management**: Handles operations related to saving, retrieving, removing products from user's wishlist.
- **User Management**: Oversees user information, including personal details, authentication (login/logout), account settings.
- **Customer Support**: Manages customer queries support tickets, typically integrating with third-party customer service systems.
- **Payment Processing**: Handles secure transactions, ensuring that payment information was processed verified, possibly integrating with external payment gateways (e.g., PayPal, Stripe).
- **Return Refund Processing**: Manages order returns refunds based on user requests, ensuring compliance with return policies facilitating refunds via payment processors.

3. Data Tier (Data Storage Management):

This tier manages all application's data, ensuring proper storage, retrieval, management of system's records:

- **User Database**: Stores user-related data, including login credentials, contact information, order history.
- **Product Database**: Contains product catalog, with details like product descriptions, prices, stock levels, categories.
- **Order Database**: Tracks user orders, including purchase details, order status, payment confirmation, shipment tracking.

System Communication:

- The system follows typical client-server model where users send **HTTP requests** (such as product queries, order placements) to server via front-end, back-end processes these requests. back-end retrieves or stores data in appropriate databases returns processed results (e.g., product information, order confirmations) to front-end, which was displayed to user.

This three-tier architecture ensures clear separation of concerns:

- **Presentation Tier** handles user interaction,
- **Application Tier** deals with logic processing,
- **Data Tier** manages storage retrieval.

This design allows for scalability, maintainability, security by isolating each function of system into its own layer

Descriptions of the system's components:

- **LOGIN/SIGNUP:**
  The Login/Sign-Up component allows users either to register on platform or authenticate existing account for them to access customized services from site.

  Users sign up through Sign up process where they were required to submit their email password which were well encrypted.

  The Login process validates information provided by users provides Pass or Access if information matches that stored in program.

  Furthermore, session management keeps users logged in throughout browsing session with password reset two factor login to improve security.

- **SHOPPING CART:**
  Shopping Cart component where users can add view products they were planning to buy manage them before going to check out.

  It dynamically makes changes of quantities deletes item calculates total cost including total taxes total shipping fees.

  The cart synchronized with back-end to update price check availability it saves selection in one of storages: session storage for guests or database for user logged in.

  It improves site use by enabling smooth transition to purchasing phase where consumers complete their consumptions.

- **WISHLIST:**
  The Wishlist component lets users add products user interested in but does not wish to buy at moment; they can keep track of selected items without purchasing them.

  Thus users can view added products their details such as price availability add or delete from wish list or transfer to shopping cart in case of purchase.

The wish list data mainly saved in database for registered users while for guest users wish lists were saved temporarily using cookies.

They increase user interactions by making user come back to app next time then when notification of sale or stock-in of saved items included.

- **CUSTOMER:**

  The Customer Support segment helps people looking for any help regarding order product or account handling.

  Some of sub modules that make up this sub module include live chat help center which houses frequently asked questions ticketing for queries.

  The customers were able to submit support tickets to relevant teams get real time status of tickets.

  This component improves loyalty of customers since clients will always get their issues solved efficiently on time.

- **PAYMENT PROCESS:**

  This Payment part also includes secure monetary processing area which handy during payments or checking out processes therefore adding features such as credit cards or wallets.

  It links with other online payment systems such as Stripe or PayPal to confirm complete payments instantly.

  The system also performs some functions such as encrypting of data confirmation of payment changing of order status among others.

  Moreover, payment component for application includes options like refund payment advisories that make use dependency of buying process on application reliable among users.

- **RETURN REFUND PROCESS:**

  The Return Refund process enable consumer to apply for returns in regard to certain products that he/she has purchased get refund in accordance with laid policies.

  It controls workflow through return approval based on product conditions; delivery deadlines refund via initial payment method.

  Upon acceptance status of order changed customer provided with information regarding refund.

  The process finally improves customer satisfaction since it avails clear efficient method of handling returns recovering charges on qualifying products.

- **USER DATABASE:**

During User Database creation all necessary information required for authorization user's identification order history etc. has to be saved.

In this way it deals with user's delicate information such as password responsibly in order to protect user's privacy.

The database supports such operations like authentication of users control of personal profiles providing selected services based on users' interests actions.

It very important for user accounts including login registration also to do wishlist & cart recovery to subsequent session.

- **PRODUCT DATABAS :**

The Product Database contains all necessary details about products that were sold on e-commerce platform include product's name description materials used price quantity photo display.

And it enables platform to obtain provide correct product information for browsing or search purchase.

It has ever improving database for frequently changing stock price products on market.

Such aspects include: product search recommendation which facilitate optimization of product details for users; processing of orders so as to ensure that details provided were most current.

- **ORDER DATABASE:**

The Order Database has tracking details for all transactions of every user including order number products ordered quantity amount charged order status.

It follows complete order life cycle from order placement order shipping history payment details records returns refunds if any.

This database allows individual to reach information about past orders shipments they made returns.

It has significant position in order management where system guarantees that it users have updated correct order information.

REQUIREMENTS SPECIFICATIONS:

Functional Requirements:

Functional requirements define the actions, features, or jobs that a system needs to execute properly. They explain the actions the system will take to fulfill its main objective; they define what exactly does system does. These specifications would list the necessary functions and features that the system must offer to users, administrators, and other people in an e-commerce project.

eCommerce functional requirements:

1. **User Registration & Authentication**:

- **Here it allows users to register and login with their details like (username, phone number, e-mail id and password).**
- It also allows the user to reset the password.

2. Product Management:

- Users can browse the products of their wish, using names or filters.
- They can check the product specifications (color, usage, volume, type), price details, images, reviews.
- The administrator can add, edit and delete the products that are available. And organize the products for easy access.

3. Wishlist and shopping cart:

- Here users can add their interested products to their Wishlist for future purchase, can remove the product if no longer needed or can update.
- User can add to cart for the immediate checkout, there it shows the total price of the items added to the cart.

4. Checkout and Payment:

- Here users will be updating shipping and billing details, to check out the product to their destination.
- **Users can also update their account details.**
- Then the system provides the payment gateways, here we are going to provide a mock payment method just to act as the payment is done.
- Once the payment is made the order confirmation will be sent to the email with the product details.

5. Order Tracking:

- Once the order is placed successfully the system gives access to the user to check the product details ordered for the delivery, all past orders will be shown.

- The system provides tracking methodology to check the status of the product.
- Return and cancellation flexibility are provided to the user.

6. Reviews and Ratings:

- Once the product is delivered the user can post their reviews and rate the product.
- The system displays the reviews provided for the products.
- Administrators can edit or delete reviews.

7. Discounts and Coupons:

- The administrator will give the standard coupons and discounts sometimes to the user once the account is created.
- The system will update the price after applying the discount to the product at the time of checkout.

Non-Functional Requirements

Non-functional requirements describe the qualities the system should have, focusing on performance, security, and scalability. They focus on how the system performs. These are required for effective operations and meet the users' expectations. And to make the throughout process easy and secure.

1.Performance:

- The system loads the pages without delay.
- Shows accurate search result.

2. Security:

- The system secures the user details such as password, address, personal details.
- The system provides security for the user payment information.

3. Scalability:

- When there is traffic, the system diverts to the other websites.
- The system should display the product quickly.

4. Usability:

- Should be able to access all the devices at all networks.
- Easy to use and access.

5. Reliability:

- The website will be available all the time without server breakdown.
- The system will be recovering from server failures.

6. Maintenance:

- Flexibility will allow the system to change, modify, and resolve easily at any level.

7. Availability:

- The system is available all the time checking traffic.

8. Data Integrity:

- Maintains consistency throughout the lifecycle.

Interfaces:

The various interfaces that the system will communicate with are described in this section, including hardware, software, user, and communication interfaces.

1. User Interfaces:

- Easy navigation to product categories will be provided by the system's responsive homepage.
- An order summary must be clearly displayed on the checkout interface prior to the order being submitted.

2. Hardware Interfaces:

- Hardware for the system must be standard web servers that meet basic requirements.
- A minimum of 500 GB of storage for transaction and product data must be supported by the database server.

3. Software Interfaces:

- The Linux, Apache, MySQL, and PHP (LAMP) stack will be used in the construction of the eCommerce platform.
- To maintain customer relationships, the system must interface with third-party CRM software.

4. Communication Interfaces:

- When communicating between front-end and back-end services, the system must make use of RESTful APIs.
- For order updates and confirmations, email notifications must be provided using SMTP protocols.

DEVELOPMENT PHASE PLANE:

1. Planning and Design Phase (Duration: 2 weeks)

- **Objective:** Create a proper vision and enlist the requirements, design the system architecture and enlist the proper functionalities. This would include:
    - Create and finalize the scope of the project and requirements on the basis of system specifications
    - Design the architecture for the project.
    - Create the UI and the wireframes
    - Then define database schema
    - Identify and define the external APIs.
    - Plan the security of the application.

2. Phase 1: Core feature Development. (Duration: 2 weeks)

- **Objective:** To develop the most critical features of the project.
    - User Authentication and profile.
    - Product Listing and filtering
    - User Interface for login, signup and products
    - Implementing search and filter functionality
    - Start developing the recommendation engine.

3. Phase 2: Core feature development (Duration: 2 weeks)

- **Objective:** To develop more critical features of the project.
    - Shopping cart
    - Payment portal
    - Order tracking
    - Address Management
    - Testing the critical features and redeveloping if anything goes wrong
    - Continuing the recommendation engine.

4. Phase 3: Secondary feature development (Duration: 2 weeks)

- **Objective:** To develop secondary features as well as working on any bugs from critical core feature.
    - Implementing notification and email functionalities
    - Discount and coupons functionality
    - Order history functionality
    - Data validation and security measurements
    - Finalize the recommendation engine.

5. Phase 4: Final touches and testing (Duration: 2 weeks)

- **Objective:** Application polishing and debugging
    - UI and UX improvements

  - Implement user session timeouts
  - Fix final bugs
  - Full system testing including all use cases

6. Phase 5: Deployment and testing remotely (Duration: 2 weeks)

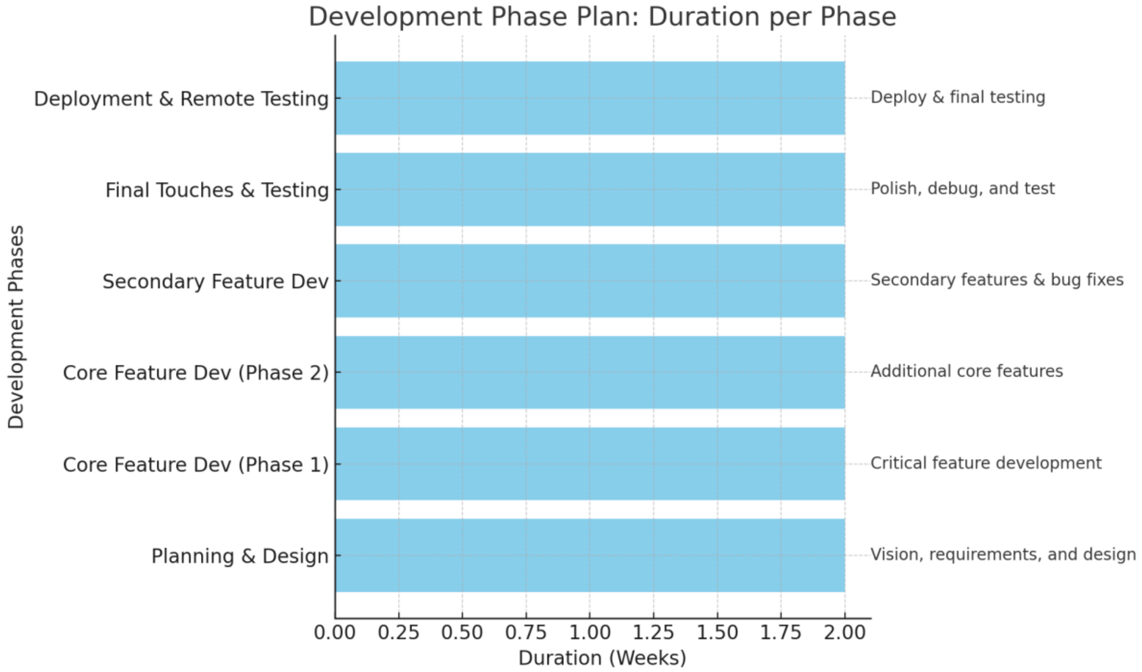
- **Objective:** Deploying the application
  - Deploy the frontend and backend to cloud
  - Gather feedback from professors and TAs
  - Conduct final round of acceptance testing

The entire process will be using agile methodology, which will include constant testing and feedback incorporation.

Phase	Duration	Objective	Key Activities
Planning and Design Phase	2 weeks	Create a proper vision, enlist requirements, design architecture	- Finalize project scope and system specifications - Design system architecture - Create UI and wireframes - Define database schema - Identify external APIs - Plan security of the application
Phase 1: Core Feature Development	2 weeks	Develop the most critical features	- User authentication and profile - Product listing and filtering - Create UI for login, signup, products - Implement search and filter functionality - Begin developing the recommendation engine

Phase 2: Core Feature Development	2 weeks	Continue core feature development	- Shopping cart - Payment portal - Order tracking - Address management - Test critical features - Continue developing the recommendation engine
Phase 3: Secondary Feature Development	2 weeks	Develop secondary features and address bugs	- Implement notification and email functionality - Discount and coupons functionality - Order history - Data validation and security measures - Finalize recommendation engine
Phase 4: Final Touches and Testing	2 weeks	Polishing and debugging	- UI/UX improvements - Implement user session timeouts - Fix final bugs - Full system testing, including all use cases
Phase 5: Deployment and Remote Testing	2 weeks	Deploy the application and gather feedback	- Deploy frontend and backend to cloud - Gather feedback from professors and TAs - Conduct final round of acceptance testing
Agile Methodology	2 weeks	Incorporate constant testing and feedback	Agile process includes ongoing testing, feedback, and adjustments throughout all phases.

Graph:



Member Contribution Table:

Member Name	**Contribution Description**	**Overall Contribution (%)**	**Note (if applicable)**
Harini	Requirement specification, Meeting minutes, member contribution table	12.5%	
Osama	Development phase plan	12.5%	
Nirupama	System structure, Note deliverable 2 file submissions	12.5%	
Niharika	System structure, Meeting minutes, member contribution table	12.5%	
Jaswanth	Requirement specification, Meeting minutes	12.5%	
Rajsekhar	Development phase plan	12.5%	
Triveni	Requirement specification	12.5%	
Sharanya	System structure	12.5%	

Meeting Minutes:

Meeting 1 Minutes:

Date: September 11th

Time: 10:30 pm - 11:30 pm

Location: Teams Meet

Attendees: All the team members

1. Objective:

   Had a discussion on deliverable 2 document topics and assigned the task to everyone and spoke on the deadline to finish up the assigned tasks.

2.Next Meeting:

- Date: September 17th, 2024
- Time: 11:00PM to 11:30PM
- Location: Teams Meet

3. Adjournment:

- The meeting was adjourned at 11:30 PM.

Meeting 2 Minutes:

Date: September 17th

Time: 11:00PM to 11:30PM

Location: Teams Meet

Attendees: All the team members

1. Objective:
   Updated the progress on the assigned tasks from respective members, discussed on the changes to be done and worked on the suggestions by team members, and finalized the deadline to finish assigned task.
2. Next Meeting:
   - Date: September 18th, 2024
   - Time: 4:30 PM to 6:30PM
   - Location: Teams Meet
3. Adjournment:
   The meeting was adjourned at 11:30PM.

Meeting 3 Minutes:

Date: September 18th

Time: 4:30PM to 6:30PM

Location: Teams Meet

Attendees: Harini, Jaswant, Triveni

1. Objective:

Worked on functional and non- functional requirements for deliverable 2.

2. Next Meeting:

- Date: September 18th, 2024
- Time: 9:30 PM to 11:00PM
- Location: Teams Meet

3. Adjournment:

The meeting was adjourned at 6:30PM.

Meeting 4 Minutes:

Date: September 18th

Time: 9:30PM to 11:00PM

Location: Teams Meet

Attendees: Osama, Rajsekhar

1. Objective:

Worked on Development phase plan for deliverable 2.

2. Next Meeting:

- Date: September 21st, 2024
- Time: 11:00 AM to 1:00PM
- Location: Teams Meet

3. Adjournment:

The meeting was adjourned at 11:00PM.

Meeting 5 Minutes:

Date: September 21st

Time: 11:00 AM to 1:00PM

Location: Teams Meet

Attendees: Nirupama, Niharika, Sharanya

1. Objective:

Worked on System architecture for deliverable 2.

2. Next Meeting:

- Date: September 27th, 2024
- Time: 10:30 PM to 11:30PM
- Location: Teams Meet

3. Adjournment:

The meeting was adjourned at 1:00PM.

Meeting 6 Minutes:

Date: September 27th

Time: 10:30 PM to 11:30PM

Location: Teams Meet

Attendees: All the team members

1. Objective:

Shared the prepared documents on deliverable-2 with the rest of the team members and made the final editing's and finalized the documents. And discussed about the project implementation and made updated Trello and github. And discussed on the final submissions of deliverable 2.

2. Next Meeting:
   - Date: September 28th, 2024
   - Time: 5:00 PM to 6:20PM
   - Location: Teams Meet
3. Adjournment:
   - The meeting was adjourned at 11:30 PM.

Meeting 7 Minutes:

Date: September 28th

Time: 5:00 PM to 6:20PM

Location: Teams Meet

Attendees: Niharika, Harini, Jaswanth

1. Objective:

Worked on Meeting minutes and member table for deliverable 2.

2. Next Meeting:
   - Date: September 29th, 2024
   - Time: 10:30 PM to 11:00PM
   - Location: Teams Meet

3.Adjournment:

- The meeting was adjourned at 6:20 PM.

Meeting 8 Minutes:

Date: September 29th

Time: 10:30 PM to 11:00PM

Location: Teams Meet

Attendees: All the team members

1. Objective:

Final submission for the deliverable 2.

2. Adjournment:

    - The meeting was adjourned at 11:00 PM.