Project Title: **ShopNest: An E-Commerce Application**

Team Members:

1. Name: MUFEED

CAN ID: CAN_33682299

2. Name: OBALESHA B

CAN ID: CAN_33465346

3. Name: RIZA RIYAS ABDUL KHADER

CAN ID: CAN_33903966

4. Name: VARSHA M

CAN ID: CAN_33663583

Institution Name: P A COLLEGE OF ENGINEERING, MANGALORE

Phase 3: Implementation of Project

Objective

The goal of Phase 3 is to implement the core components of the ShopNest platform based on the planned architecture. This includes developing the front-end and back-end systems, implementing secure payment processing, integrating user authentication, and conducting initial testing to ensure a seamless e-commerce experience.

1. Front-End Development

Overview

The front-end serves as the user interface for browsing products, managing the shopping cart, and completing orders. React.js is used to ensure responsiveness and interactivity, with dynamic rendering for product categories and cart updates.

Implementation

UI Design:

- Developed reusable React components for homepage, product listing, cart, checkout, and user profile.
- Ensured responsive design using Bootstrap and Tailwind CSS.

State Management:

    - Implemented Redux for managing global application state, including user sessions and cart updates.

API Integration:
    - Used Axios for fetching product metadata, user details, and order history from the backend.

Interactive Features:
    - Enabled dynamic cart updates, search functionality, and category filtering.

Outcome

- By the end of Phase 3, users can browse products, add items to the cart, and proceed to checkout seamlessly with an engaging UI.

2. Back-End Development

Overview

The back-end handles user authentication, product management, and order processing using Node.js with Express.js.

Implementation

- **RESTful APIs:**
    - Developed endpoints for user authentication, product retrieval, cart management, and order processing.
- **Authentication:**
    - Implemented JWT-based authentication for secure user login and session management.
- **Order Processing:**
    - Integrated a payment gateway (e.g., Razorpay/Stripe) for secure transactions.
- **User Activity Tracking:**
    - Implemented APIs to track order history and cart interactions.
- **Security Measures:**
    - Applied HTTPS, CORS, and rate-limiting for secure communication and data protection.

Outcome

The back-end ensures smooth product retrieval, secure authentication, and reliable order processing.

3. Database Design

Overview

- MongoDB is used to store user profiles, product data, and order details efficiently.

Implementation

- **Schemas:**
  - Used Mongoose to define schemas for users, products, orders, and carts with validation rules.
- **Data Relationships:**
  - Linked user profiles to orders and cart data for a personalized shopping experience.
- **Product Metadata:**
  - Stored metadata like names, descriptions, prices, and stock availability.

Outcome

- A structured MongoDB database effectively manages all required data, ensuring fast retrieval and data integrity.

4. Payment Gateway Integration

Overview

- Secure payment processing is implemented to allow users to complete transactions smoothly.

Implementation

- Integrated Razorpay/Stripe to handle payments securely.
- Used webhook notifications to confirm successful payments.
- Implemented order tracking for successful transactions.

Outcome

- Users can make payments securely, and order details are stored for future reference.

5. Security Implementation

Overview

Implemented security measures to protect user data and prevent unauthorized access.

Implementation

- **Authentication:** Used JWT for token-based user authentication.
- **Encryption:** Encrypted sensitive user data using Bcrypt.js and SSL/TLS.
- **Access Control:** Restricted access to certain API endpoints.

Outcome

The application ensures data privacy and secure transactions.

6. Testing and Feedback

Overview

Conducted thorough testing to evaluate functionality, performance, and user experience.

Implementation

- **Unit Testing:**
  - Used Jest/Mocha for testing individual components and APIs.
- **Integration Testing:**
  - Verified seamless communication between front-end, back-end, and database.
- **User Testing:**
  - Gathered feedback from test users to refine UI and fix bugs.

Outcome

Testing ensured a stable and user-friendly e-commerce experience.

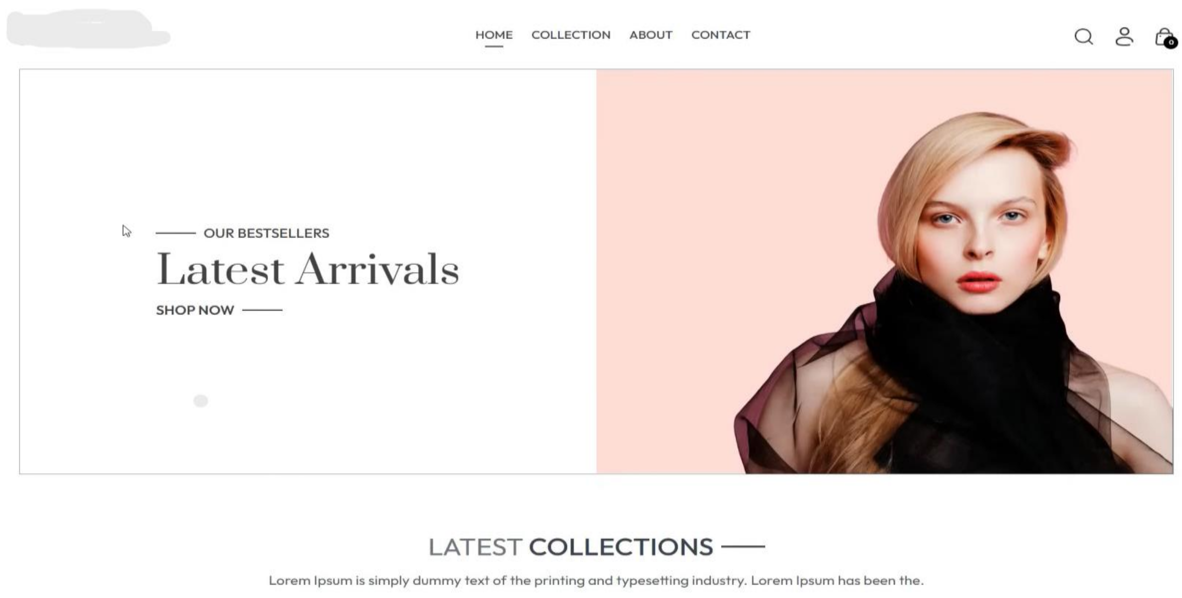
Outcomes of Phase 3

- **Front-End:** Functional UI for product browsing and cart management.
- **Back-End:** Secure and scalable APIs for user authentication and order processing.
- **Database:** Well-structured MongoDB database for efficient data management.
- **Payment:** Secure integration of a payment gateway.
- **Security:** Enforced data protection measures.
- **Testing:** Refined the platform based on feedback.

Next Steps for Phase 4

- Optimize product recommendations using AI-based algorithms.
- Improve order tracking and delivery status updates.
- Implement advanced security measures for user authentication.
- Enhance the mobile responsiveness of the platform.

Screenshots of Code and Progress



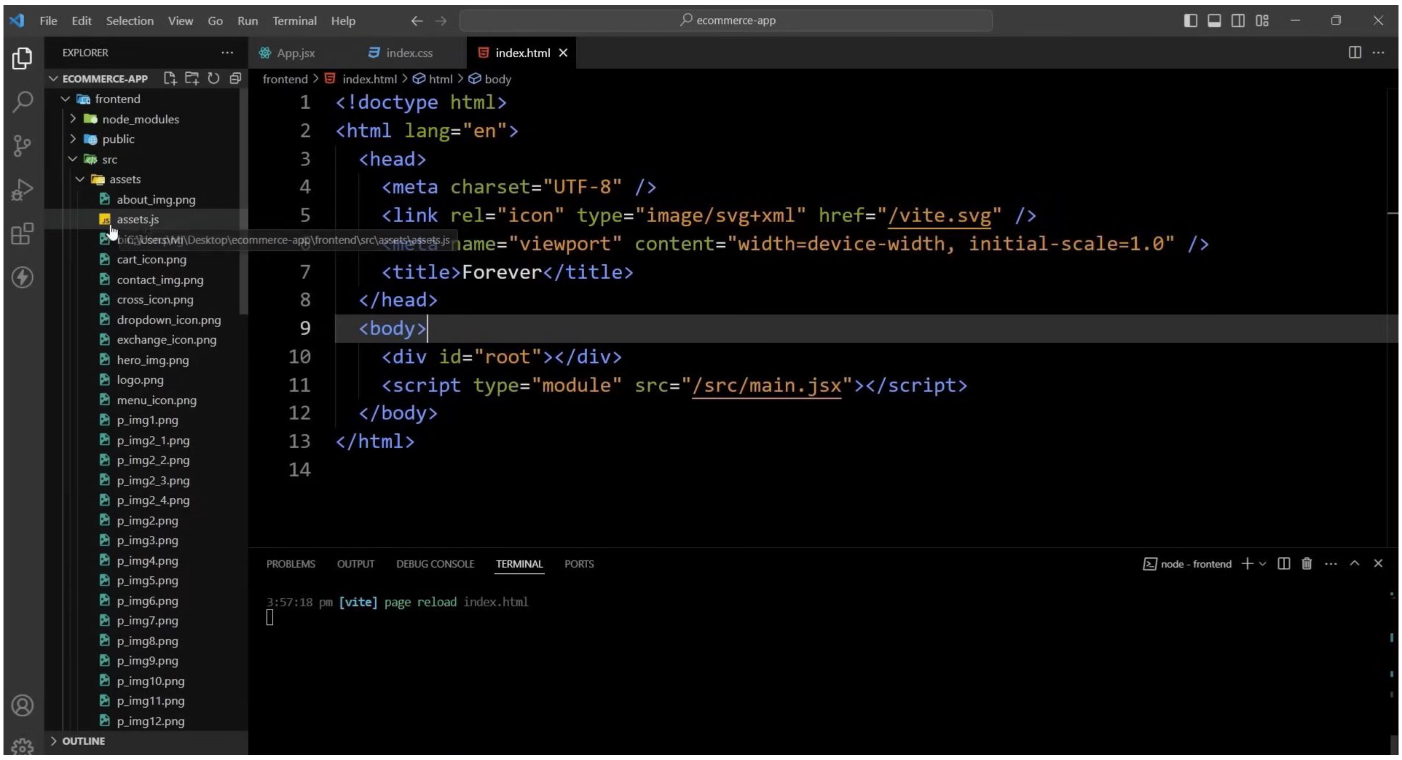
OUR BESTSELLERS

Latest Arrivals

SHOP NOW

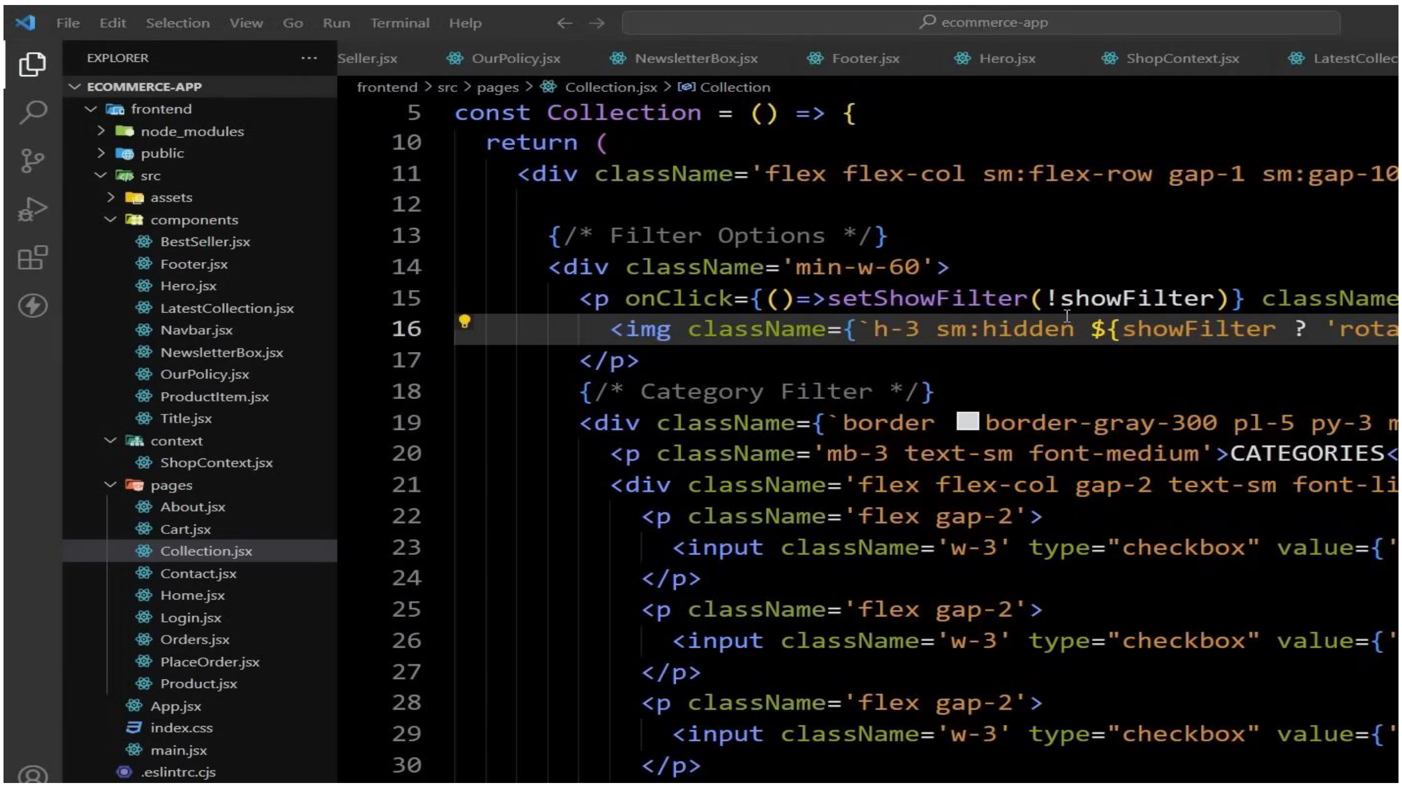
LATEST COLLECTIONS

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the.

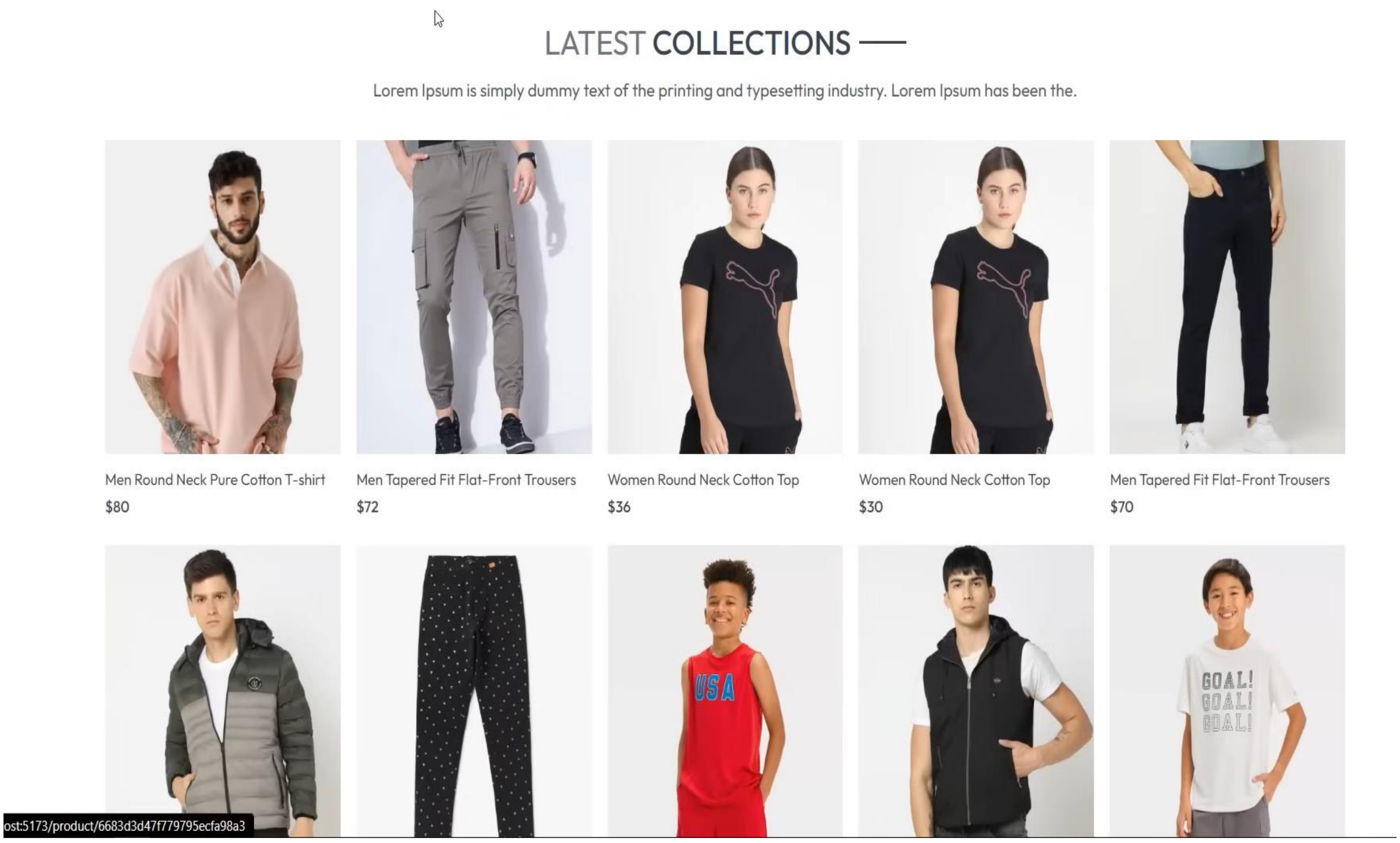


```html
<!doctype html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <link rel="icon" type="image/svg+xml" href="/vite.svg" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Forever</title>
  </head>
  <body>
    <div id="root"></div>
    <script type="module" src="/src/main.jsx"></script>
  </body>
</html>

```

```
3:57:18 pm [vite] page reload index.html
```



```
const Collection = () => {
  return (
    <div className='flex flex-col sm:flex-row gap-1 sm:gap-10

      {/* Filter Options */}
      <div className='min-w-60'>
        <p onClick={()=>setShowFilter(!showFilter)} className
          <img className={`h-3 sm:hidden ${showFilter ? 'rota
        </p>
        {/* Category Filter */}
        <div className={`border border-gray-300 pl-5 py-3 m
          <p className='mb-3 text-sm font-medium'>CATEGORIES<
          <div className='flex flex-col gap-2 text-sm font-li
            <p className='flex gap-2'>
              <input className='w-3' type="checkbox" value={'
            </p>
            <p className='flex gap-2'>
              <input className='w-3' type="checkbox" value={'
            </p>
            <p className='flex gap-2'>
              <input className='w-3' type="checkbox" value={'
            </p>
```



LATEST **COLLECTIONS** ——

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the.

Men Round Neck Pure Cotton T-shirt
$80

Men Tapered Fit Flat-Front Trousers
$72

Women Round Neck Cotton Top
$36

Women Round Neck Cotton Top
$30

Men Tapered Fit Flat-Front Trousers
$70