## Front End Engineering-II

Project Report

Semester-IV (Batch-2022)

Ecommerce Website

A red and white sign

Description automatically generated with low confidence

**Supervised By: Submitted By:**

Mr. Suhaib Rehman Aditya Garg - 2210990064

Akash - 2210990081

Abhinav Rawat - 2210990038

Aditya Mittal - 2210990069

**Department of Computer Science and Engineering**

## Chitkara University Institute of Engineering & Technology,

## Chitkara University, Punjab

**Table Of Contents**

|  |  |  |
| --- | --- | --- |
| Sr.no | Section | Page No |
| 1. | Introduction | 3-5 |
| 2. | Problem Definition and Requirements | 6 |
| 3. | Proposed Design/Methodology | 7-9 |
| 4. | Result | 10-13 |
| 5. | References | 14 |

1. **Introduction**
   1. **Background**

The evolution of web technologies has significantly impacted online commerce, leading to the rise of eCommerce platforms catering to diverse consumer needs. The project aims to create a user-friendly and efficient eCommerce website using modern web development tools and frameworks. The website is built on the React JS library, which offers modularity, reusability, and maintainability of code. React JS is chosen for its advantages such as component reusability, virtual DOM, one-way data binding, and ecosystem support.

The project also incorporates essential web development practices to ensure code quality, maintainability, and scalability. ESLint is used to enforce coding standards and identify potential errors or inconsistencies in the codebase. Font Awesome is integrated for iconography and a favicon is included for a visually appealing user experience. The project aims to create a modern and efficient eCommerce platform for success in today's digital marketplace.

**1.2 Objectives**

• Implement secure user authentication using Firebase Authentication and OAuth providers.

• Develop a comprehensive product management system with features for product addition, update, and deletion.

• Implement a shopping cart for seamless shopping experience.

• Integrate a reliable and secure payment gateway using Razorpay.

• Manage and track customer orders efficiently using a system for order placement, tracking, and history.

• Create a responsive design using HTML, CSS, and React for all devices.

• Provide an intuitive and engaging user interface with features like product search, filters, and sorting.

• Ensure user data and transactions are secure with HTTP, data validation, and secure payment processing.

• Optimize the website for fast load times and smooth performance using React best practices.

• Build a scalable and maintainable codebase using modular and reusable components in React.

• Implement analytics to track user behavior and sales using tools like Google Analytics and Firebase Analytics.

• Provide real-time notifications to users using push notifications and email alerts for important events using Firebase Cloud Messaging.

* 1. **Significance**
* Secure user authentication: Protects user accounts and personal information.
* Comprehensive product management: Ensures easy handling of product listings and inventory.
* Seamless shopping cart: Enhances the shopping experience by allowing easy product selection and modification.
* Reliable payment integration: Facilitates smooth and secure transactions.
* Efficient order management: Enables effective tracking and fulfilment of orders.
* Responsive design: Provides accessibility and usability across various devices.
* Intuitive user interface: Improves user engagement and satisfaction.
* Strong data security: Safeguards user data and builds trust.
* Performance optimization: Ensures fast load times and smooth operation.
* Scalable codebase: Allows for future growth and easy maintenance.
* Analytics and reporting: Provides insights into user behavior and business performance.
* Real-time notifications: Keeps users informed about order updates and promotions.

1. **Problem Definition and Requirements**

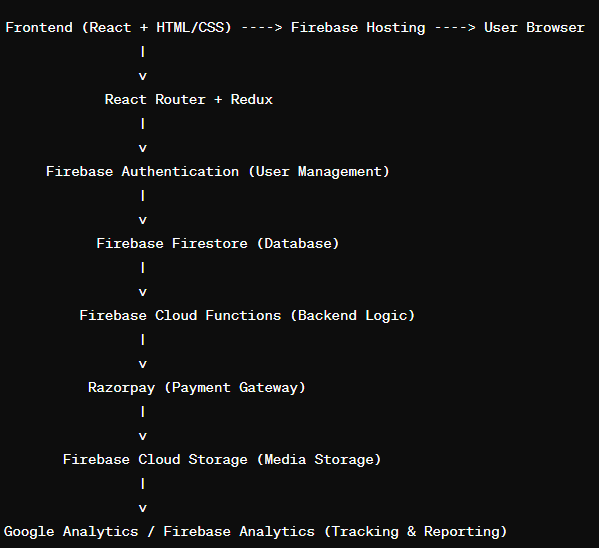
The current e-commerce market lacks a seamless, secure, and user-friendly platform that efficiently integrates modern technologies for product management, user authentication, secure payment processing, and real-time order tracking. The goal is to develop an e-commerce website that addresses these deficiencies by leveraging Firebase for backend services, Razorpay for payment processing, and React for a dynamic frontend, ensuring a robust, responsive, and scalable system.

* Secure user authentication: Protects user accounts and personal information.

* Comprehensive product management: Ensures easy handling of product listings and inventory
* Seamless shopping cart: Enhances the shopping experience by allowing easy product selection and modification.
* Reliable payment integration: Facilitates smooth and secure transactions.
* Efficient order management: Enables effective tracking and fulfillment of orders.
* Responsive design: Provides accessibility and usability across various devices.
* Intuitive user interface: Improves user engagement and satisfaction.
* Strong data security: Safeguards user data and builds trust.
* Performance optimization: Ensures fast load times and smooth operation.
* Scalable codebase: Allows for future growth and easy maintenance.
* Analytics and reporting: Provides insights into user behavior and business performance.
* Real-time notifications: Keeps users informed about order updates and promotions.

1. **Proposed Design / Methodology**
   1. **System Architecture:**

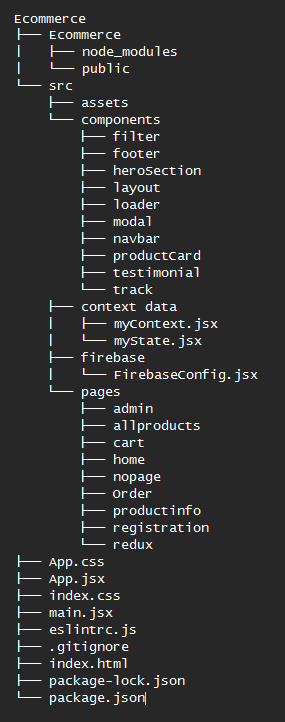
The system architecture for the e-commerce website will be designed to ensure seamless integration between the frontend, backend, and external services, providing a robust and scalable platform. The architecture will be component-based, ensuring modularity and maintainability.



The React application provides a dynamic shopping experience using HTML and CSS. Firebase services handle backend tasks like user authentication and data storage. Razorpay handles payment transactions securely. Firebase Hosting delivers web content, while Cloud Storage handles media assets. HTTPS ensures secure communication, Firebase Security Rules protect the database, and Google Analytics tracks user interactions for continuous improvement.

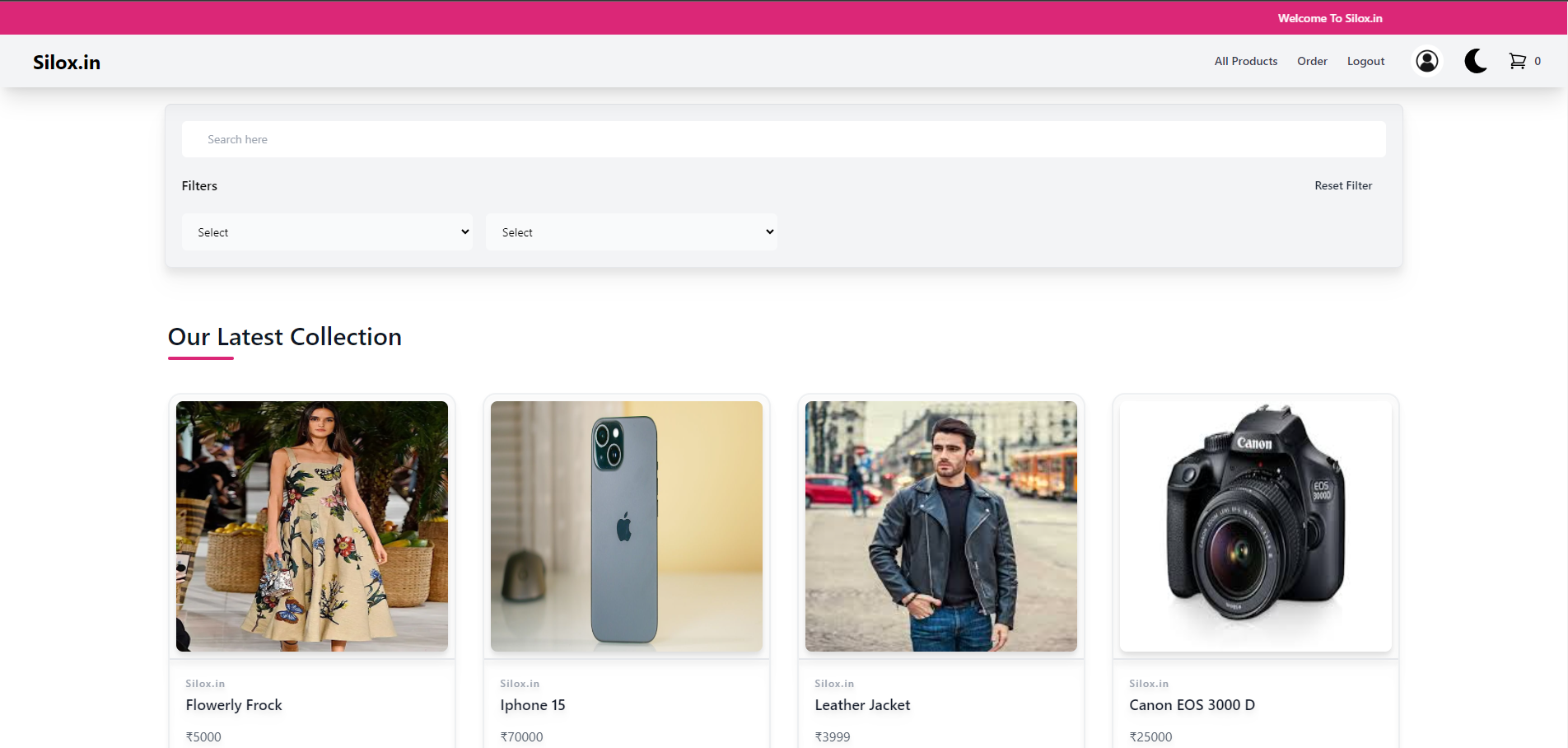
* 1. **File Structure:**

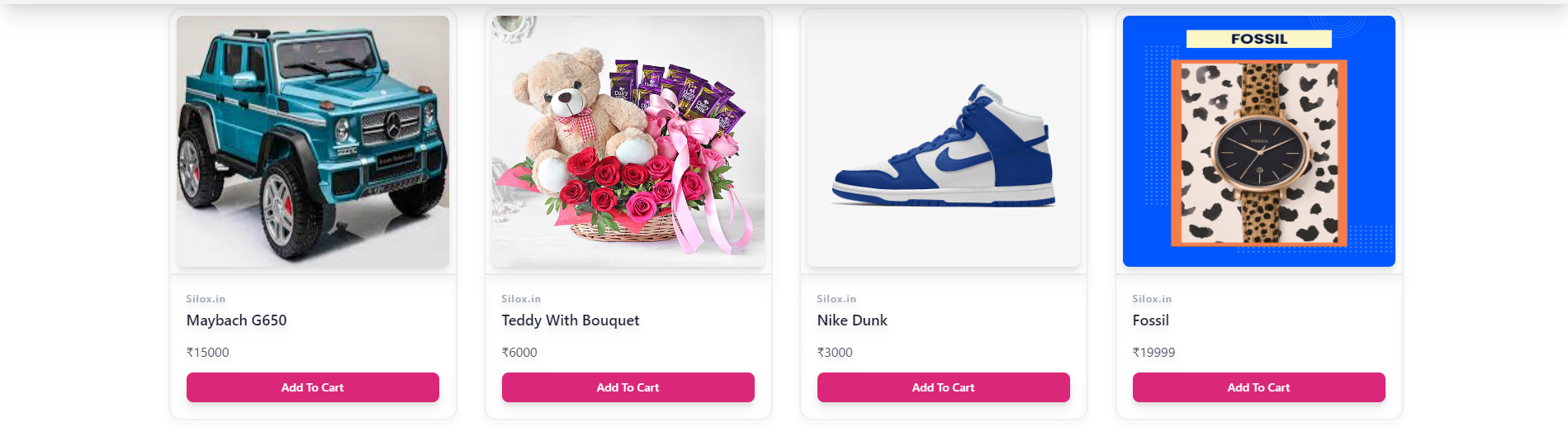
The image depicts a web application using React.js and Node.js, with various components like root directory, source code, assets, firebase, pages, and.gitignore, with React application's main entry points and Node.js managing dependencies.



* 1. **Algorithms Used**
     1. Search Algorithms: Identified binary search and trie data structures as useful algorithms for efficient product search and autocomplete suggestions, respectively.
     2. Recommendation Algorithms: Collaborative filtering and content-based filtering are indeed popular techniques for generating personalized product recommendations based on user behavior and preferences.
     3. Sorting Algorithms: QuickSort and MergeSort are efficient sorting algorithms that can be used to sort product lists based on various attributes, as you mentioned.
     4. Payment Processing Algorithms: Accurately mentioned encryption algorithms for securing payment transactions and tokenization for reducing the risk associated with storing payment information.
     5. Routing Algorithms: Shortest path algorithms and vehicle routing algorithms are essential for optimizing delivery routes and minimizing transportation costs.
     6. Caching Algorithms: The LRU (Least Recently Used) algorithm is a widely used caching technique for improving performance by caching frequently accessed data, such as product information and user sessions.
     7. Machine Learning Algorithms: Clustering algorithms, classification, and regression algorithms are powerful tools for personalizing the user experience by segmenting users, predicting behavior, and generating relevant recommendations.

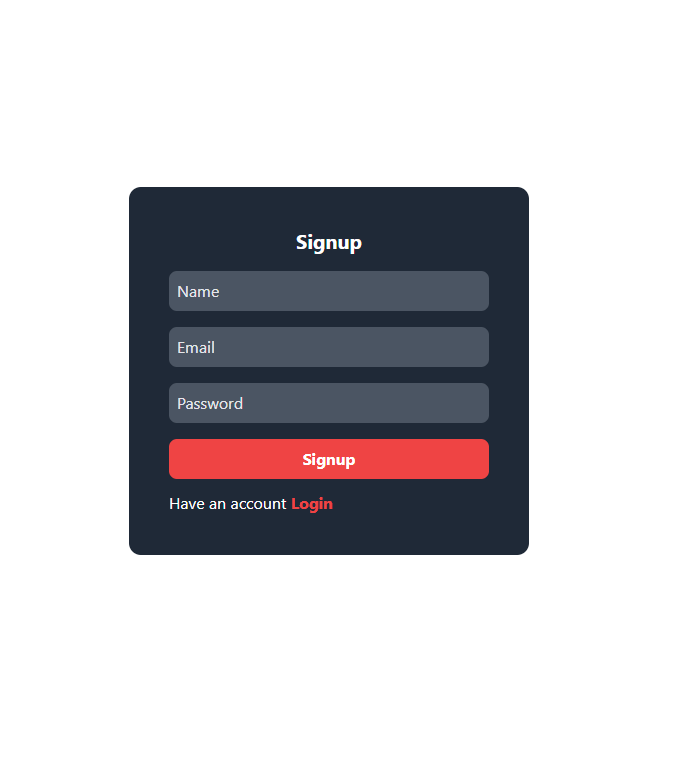
1. **Results**
   1. **WEBSITE**

****

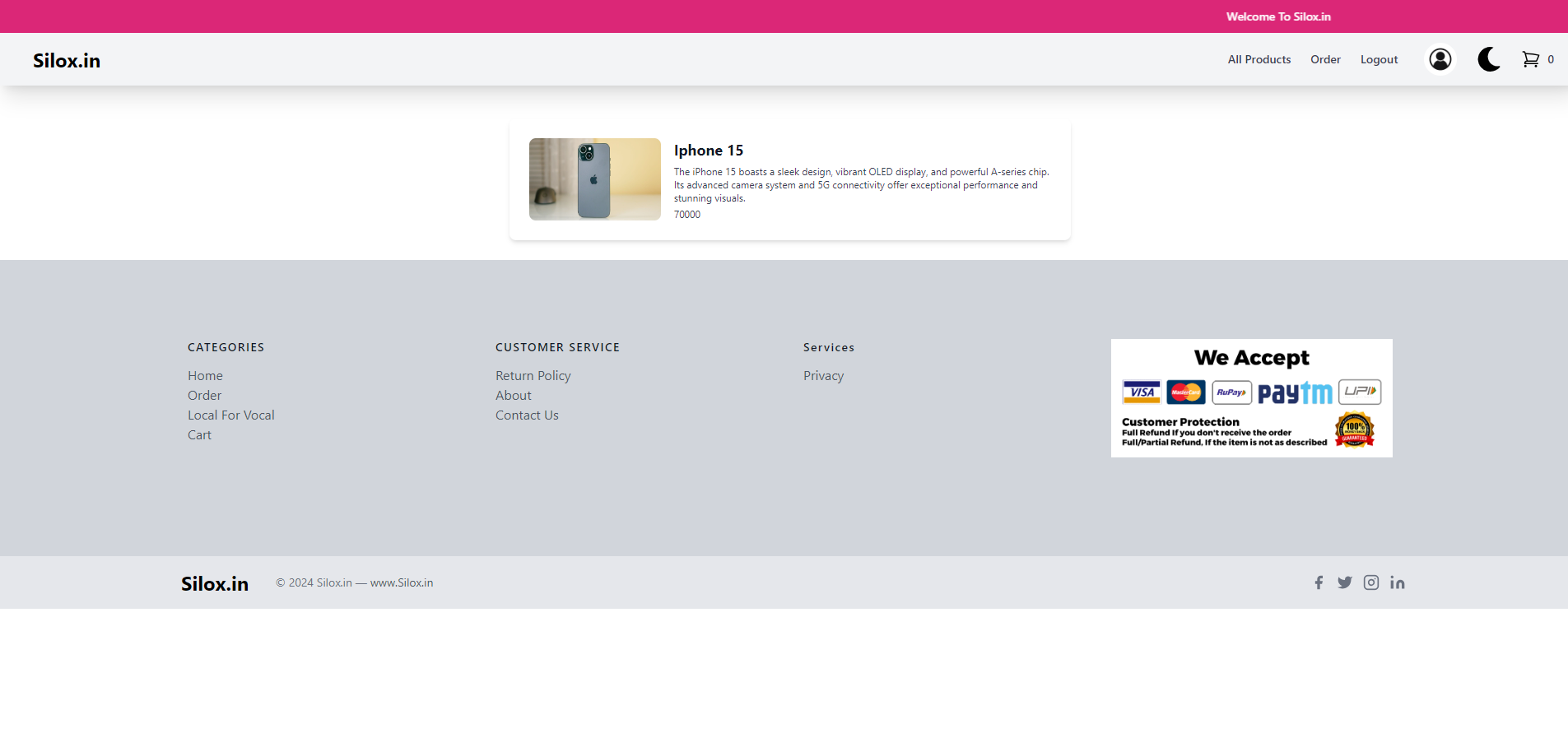
****

****

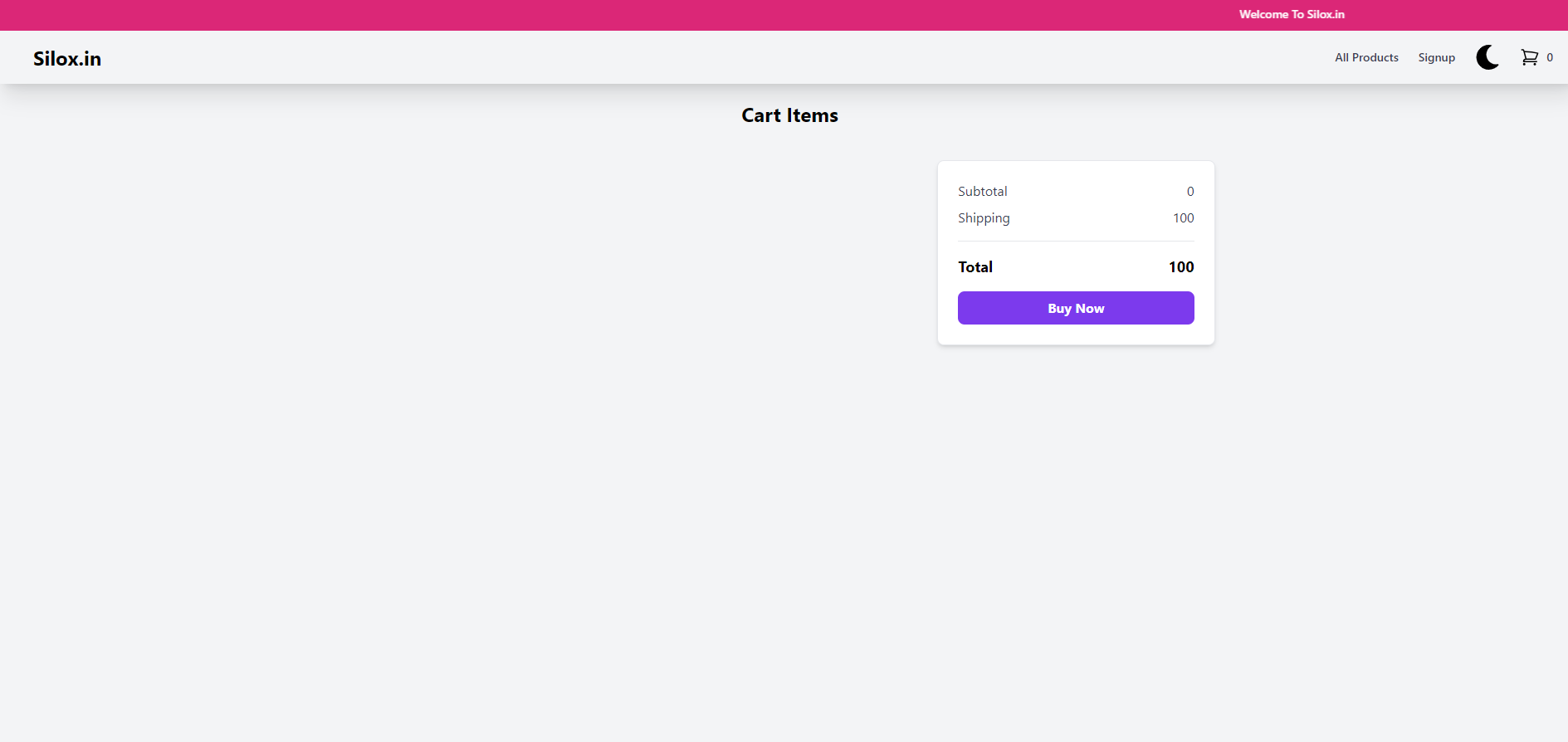
**4.2 SIGNUP**

****

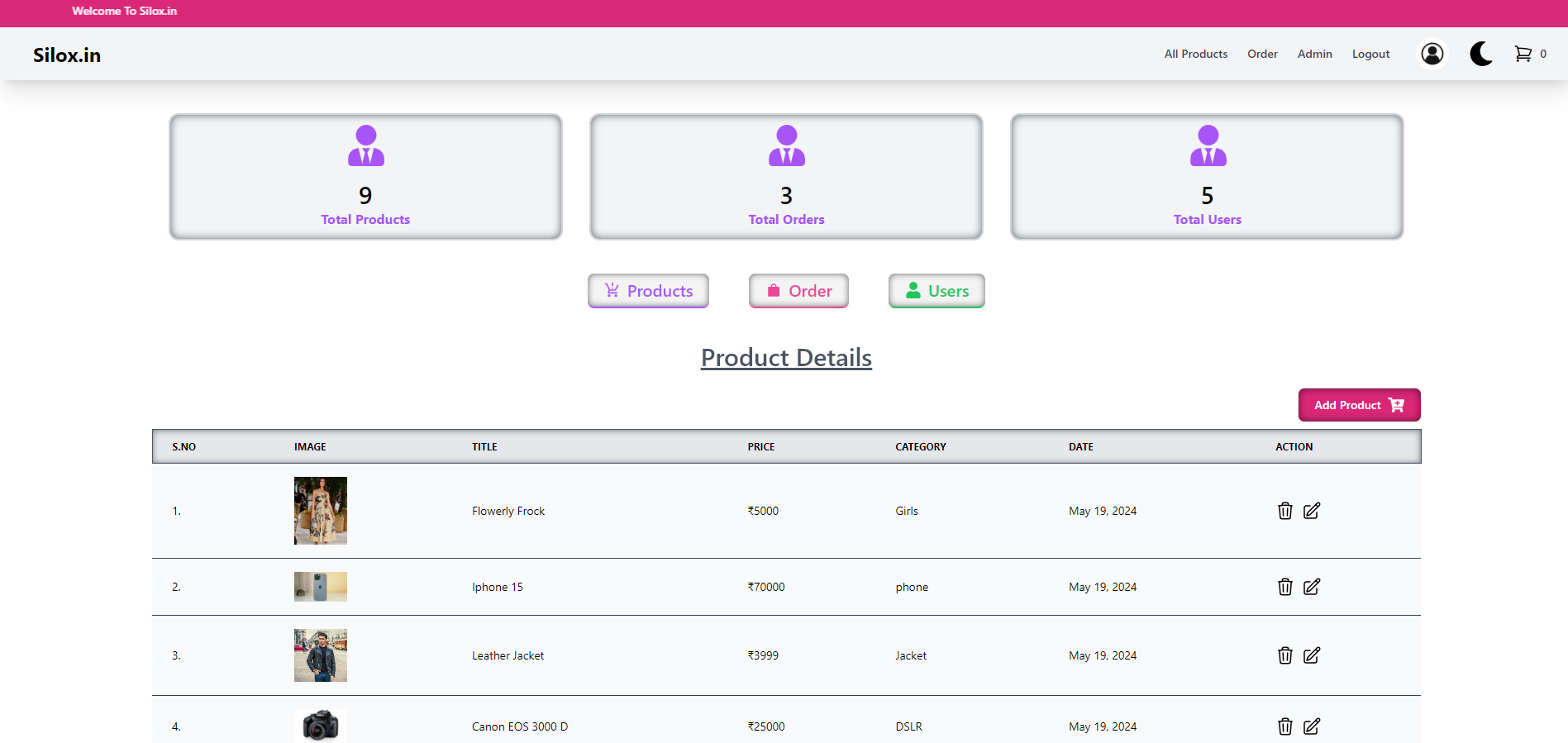
**4.3 ORDER**



**4.4 CART**



**4.5 ADMIN PORTAL**

****

## 

## 

**5. References**

|  |  |
| --- | --- |
| 1. Tailwind | https://tailwindcss.com/ |
| 2. React | https://react.dev/ |
| 3. Github | https://github.com/Aditya-1406/Ecommerce\_Final |
| 3. Razorpay | https://razorpay.com/ |
| 3. Firebase | https://firebase.google.com/ |