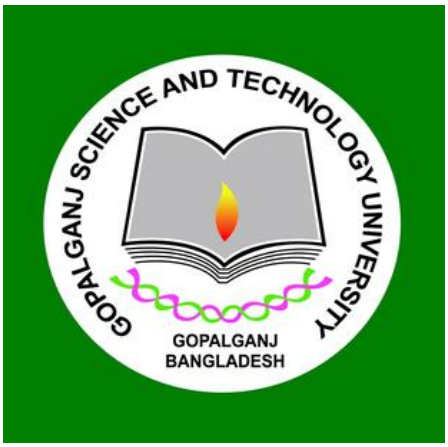
Gopalganj Science and Technology University



ONLINE SHOPPING APPLICATION

SUBMITTED BY

Md. Al-Amin Hossain
ID:20CSE001
Afrin Jahan
ID:20CSE008
Bondhon Das
ID:20CSE016
Department of Computer Science and Engineering(CSE)

SUBMITTED TO

Dr. Syful Islam
Assistant Professor
Department of Computer Science and Engineering(CSE)

Statement of Originality

I hereby declare that this report, titled “Online Shopping Application”, is the result of our own work. All information, data, and sources used in this report have been appropriately cited and referenced in accordance with academic and ethical standards.

We confirm that the content presented in this report is original and has not been submitted, in part or in full, for any other academic or professional qualification. Contributions from other individuals or sources have been duly acknowledged.

---------- ----------

Students signature Supervisor signature

Date: Date:

Abstract

The Online Shopping Application is a full-stack web-based e-commerce solution developed using modern JavaScript technologies including React.js, Node.js, Express.js, and MongoDB. The application provides seamless and secure shopping experiences, integrating both Stripe and SSLCommerz payment gateways for international and local transactions.

Users can browse products, register/login, filter by categories, and make secure purchases. Admins can manage users, products, and transactions via a role-based dashboard. The frontend is styled using Tailwind CSS and enhanced with Lottie animations, while Firebase handles authentication and JWT protects sensitive routes.

This project addresses the real-world need for a flexible and secure e-commerce system and demonstrates best practices in full-stack development.

Acknowledgement

We would like to express our deepest gratitude and sincere appreciation to our honorable course teacher and respected supervisor, Dr. Syful Islam, Assistant Professor, Department of Computer Science and Engineering, Gopalganj Science and Technology University (GSTU), for his exceptional support, insightful guidance, and unwavering encouragement throughout the development of this Software Engineering project titled “Online Shopping Application.”

His constant motivation, valuable suggestions, and constructive feedback at every phase of the project have been instrumental in shaping the direction and quality of our work. The trust he placed in our team allowed us to explore innovative approaches, overcome technical challenges, and refine our understanding of full-stack software development using modern tools and methodologies.

We are also thankful for the resources and academic environment provided by the Department of Computer Science and Engineering at GSTU, which played a significant role in the successful completion of this project.

Finally, we are grateful to our fellow classmates and peers for their continuous support, collaboration, and inspiration, which created a positive and collaborative atmosphere during the entire course of this project. Their constructive feedback and encouragement enriched our learning experience.

Md. Al-Amin Hossain, ID:20CSE001

Afrin Jahan, ID:20CSE008

Bondhon Das, ID:20CSE016

Contents

List of Figures

Chapter 1: Introduction

The Online Shopping Application is a web-based system developed to meet the modern-day demand for a fast, secure, and scalable e-commerce platform. This application enables users to explore and purchase products, while admins manage inventory, users, and transactions. Built using the MERN Stack, the system integrates both Stripe (international) and SSLCommerz (local BDT) payment methods.

1.1 Motivation and Aim

Motivation

With the rise of online shopping trends, there is a need for a robust, real-time, and user-friendly shopping system that supports both local and international transactions.

Aim

- Provide a full-featured online shopping solution
- Facilitate admin management with role-based access
- Enable users to shop securely with multiple payment gateways
- Ensure scalability, responsiveness, and ease of use

Objectives

- Develop a responsive frontend using React and Tailwind CSS
- Implement Firebase authentication with role-based access
- Secure user data and routes using JWT
- Support payments using Stripe and SSLCommerz
- Build a RESTful backend with Node.js and Express.js
- Store and manage data using MongoDB

Chapter 2: Literature Review

E-commerce has revolutionized the way businesses and consumers interact, providing a seamless digital platform for buying and selling goods and services. The evolution of online shopping platforms has been significantly influenced by advancements in web technologies, user experience design, secure payment integration, and cloud infrastructure. In this chapter, we review relevant literature and existing systems to understand the foundations and motivations for building our Online Shopping Application.

Several established platforms such as Amazon, eBay, Alibaba, Daraz, and Evaly have become benchmarks in the field of e-commerce. These systems offer a wide range of features such as product listings, payment processing, order tracking, and customer feedback mechanisms. However, despite their robustness, most of them follow proprietary structures and lack the open-source transparency that would help developers and learners understand the full-stack development process from scratch.

For example:

- Amazon provides a world-class UI/UX but does not support local currency (e.g., BDT) payments for small-scale sellers in regions like Bangladesh.
- Evaly and Daraz, while locally popular, have faced criticism related to transparency, security, and scalability, especially in admin-side operations and real-time system feedback.

These gaps highlight the need for a system that supports dual payment integration, role-based access, and full-stack transparency, which our project aims to fulfill.

2.1 Features of the Project

User Functionalities:

- Register/Login using Firebase
- Browse and filter products
- Make payments (Stripe or SSLCommerz)

Admin Functionalities:

- Admin login
- Add/edit/delete products
- Manage users
- View payment history

System Environment

Hardware Equipment

- OS: Windows 10
- Processor: Intel Core i5, 8th Gen
- RAM: 8 GB
- Display: 1024×768 or higher

Software Equipment

- Frontend: React.js, Tailwind CSS, Axios
- Backend: Node.js, Express.js
- Database: MongoDB (via Mongoose)
- Authentication: Firebase + JWT
- Payment APIs: Stripe, SSLCommerz
- Development: VS Code
- Version Control: GitHub
- Deployment: Vercel (Client), Render (Server)

Chapter 3: Implementation

This chapter provides a comprehensive overview of the implementation process of the Online Shopping Application. The project was developed using the MERN Stack (MongoDB, Express.js, React.js, Node.js) with additional services like Firebase Authentication, Stripe, and SSLCommerz for secure payment processing. The system is modular and scalable, following best practices in full-stack software engineering.

3.1 Database Design

Tables and Attributes

1. Users Table

Column	Data Type	Description
_id	ObjectId	Unique identifier (Primary Key)
name	String	User's full name
email	String	User's email (Unique)
photo	String	User's profile picture URL
password	String	Hashed password
districtName	String	User's district name (English)
districtNameBan	String	User's district name (Bengali)
upazilaName	String	User's upazila name (English)
upazilaNameBan	String	User's upazila name (Bengali)
districtID	String	ID of the district (Foreign Key)
upazilaID	String	ID of the upazila (Foreign Key)
status	String	Account status (active/inactive)
role	String	User role (e.g., user, admin)

2. Products Table

Column	Data Type	Description
_id	ObjectId	Unique identifier (Primary Key)
name	String	Product name
brand	String	Brand name
thumbnail	String	Product image URL
description	String	Product description
category	String	Category of product
price	Number	Price of the product
size	String	Size of the product (if applicable)
color	String	Color of the product
username	String	Name of the seller (Foreign Key - Users)
useremail	String	Email of the seller (Foreign Key - Users)
date	DateTime	Product creation date
status	String	Active or inactive

3. Payment Table

Column	Data Type	Description
_id	ObjectId	Unique identifier (Primary Key)
email	String	User email (Foreign Key - Users)
name	String	User's name
userPhoto	String	User's profile picture URL
amount	Number	Payment amount
date	DateTime	Payment date
productName	String	Name of the purchased product
brandName	String	Brand of the purchased product
productId	ObjectId	ID of the purchased product (Foreign Key - Products)
thumbnail	String	Product image URL
transactionId	String	Unique transaction ID
paidStatus	Boolean	Payment status (true/false)

3.2 Normalization Process (1NF, 2NF, 3NF, 4NF)

1NF (First Normal Form)

Fixes Applied:

- Separated multi-valued columns into individual attributes.
- Ensured atomic values (no multiple values in a single field).

Users Table

2NF (Second Normal Form)

Fixes Applied:

• Removed partial dependencies by creating Districts and Upazilas tables.

Users Table

_id		name	email	photo	password	districtID	upazilaID	status
67b8b7a2		a	a@gmail.com	http://res...	1234aA	2	19	active

Districts Table

districtID	districtName	districtNameBan
2	Feni	ফেনী

Upazilas Table

upazilaID	upazilaName	upazilaNameBan
19	Feni Sadar	ফেনী সদর

3NF (Third Normal Form)

Fixes Applied:

• Removed transitive dependencies by creating a separate Products Table.

Payments Table

_id	email	amount	productId	transactionId
67c7e8bf	a2@gmail.com	1497	67c74082	pi_3QzB16EA...

Products Table

_id	name	brand	thumbnail	description	category	price	size	color	useremail	date
67c74082	Shirt	Easy	https://res...	Good Shirt	Fashion	56	XL	Blue	a@gmail.com	2025-02-28T09:45:04Z

4NF (Fourth Normal Form)

Fixes Applied:
• Eliminated multivalued dependencies by creating a Product Variants Table.

Products Table

_id	name	brand	thumbnail	description	category	price	useremail	date
67c74082	Shirt	Easy	https://res...	Good Shirt	Fashion	56	a@gmail.com	2025-02-28T09:45:04Z

Product Variants Table

variantID	productID	size	color
1	67c74082	XL	Blue

3.4 ER Diagram:

The ER diagram visually represents the relationships between the entities . The main entities include Users, Products, Categories, Orders, OrderDetails, payments etc. Attributes and primary keys are defined for each

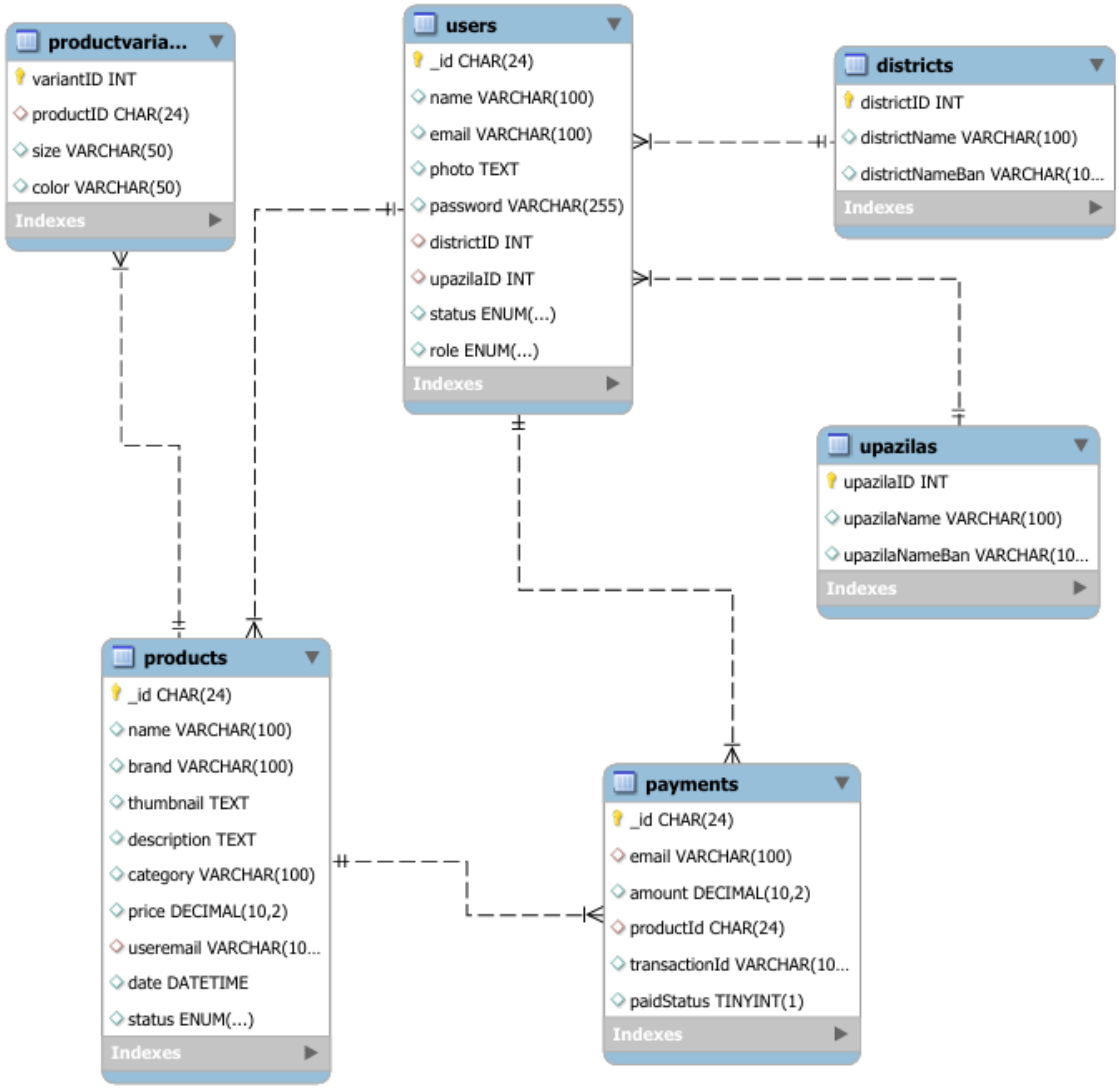
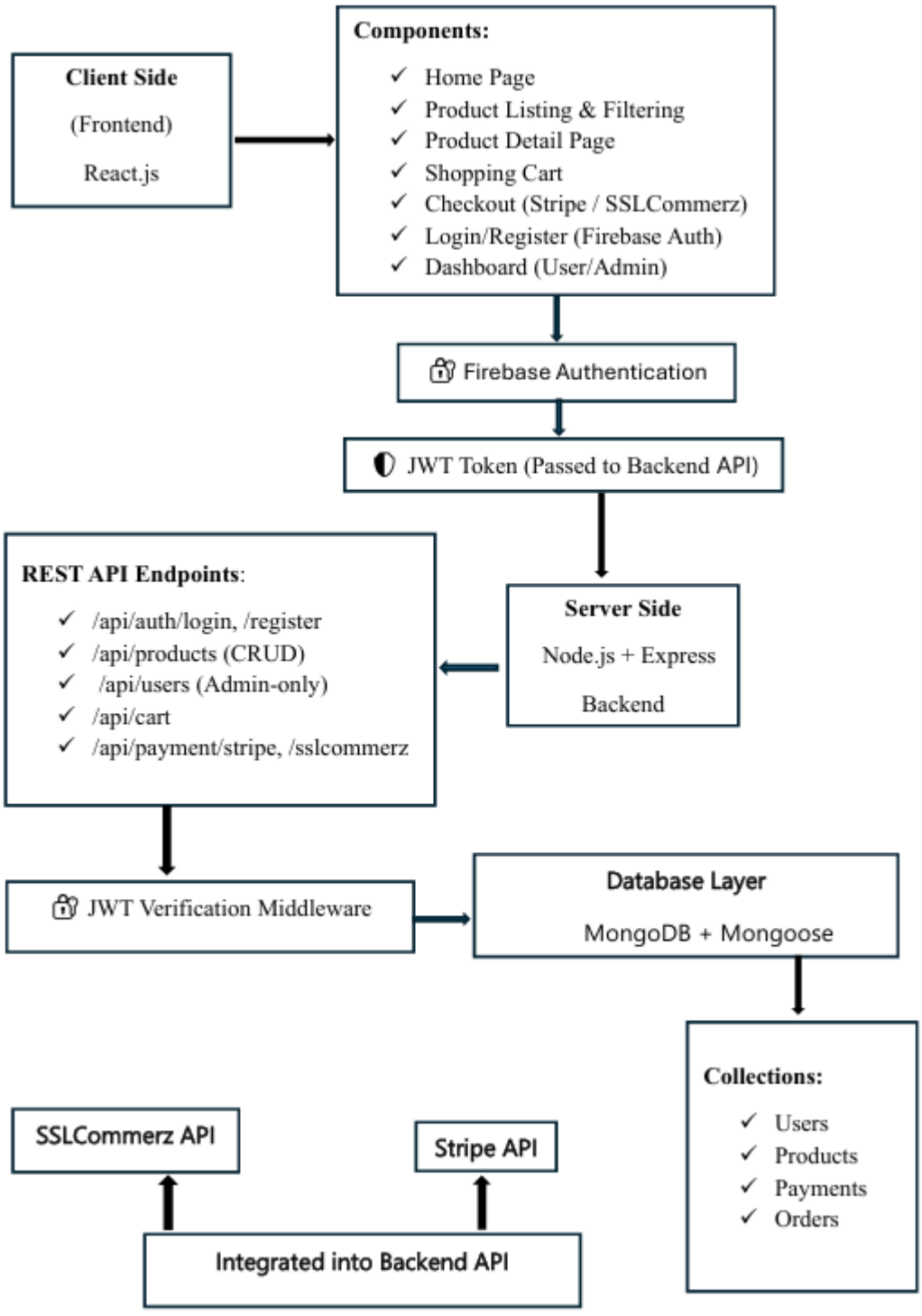


Figure 3.1: ER Diagram

3.5 Architectural Design



Chapter 4: Experimental Review

This chapter reviews the practical execution and visual outcomes of the Online Shopping Application. The review is based on real-time user interactions, UI screenshots, and the functionality of core modules, namely: authentication, product display, payment integration, and admin management. The application was tested in both local development and deployed environments to ensure consistency and responsiveness across various devices and screen sizes.

4.1 Login Page

Description:

The Login Page is the entry point for both users and admins. It is developed using React.js and Firebase Authentication.

Key Features:

- Email and password-based login
- Integration with Firebase for secure authentication
- Error handling for incorrect credentials
- Redirection to respective dashboards upon successful login
- User role recognition (admin/user) based on Firebase records

Observations:

- Responsive design on all screen sizes
- Firebase token is generated and stored in localStorage
- Users with admin role are redirected to the admin dashboard
- Regular users are redirected to the user dashboard or homepage

New User can register in the application for shopping purpose with an email and password .Later they can login using that password and email .It provides secured and reliable shopping for the user and easy to identify the previous customers as well.

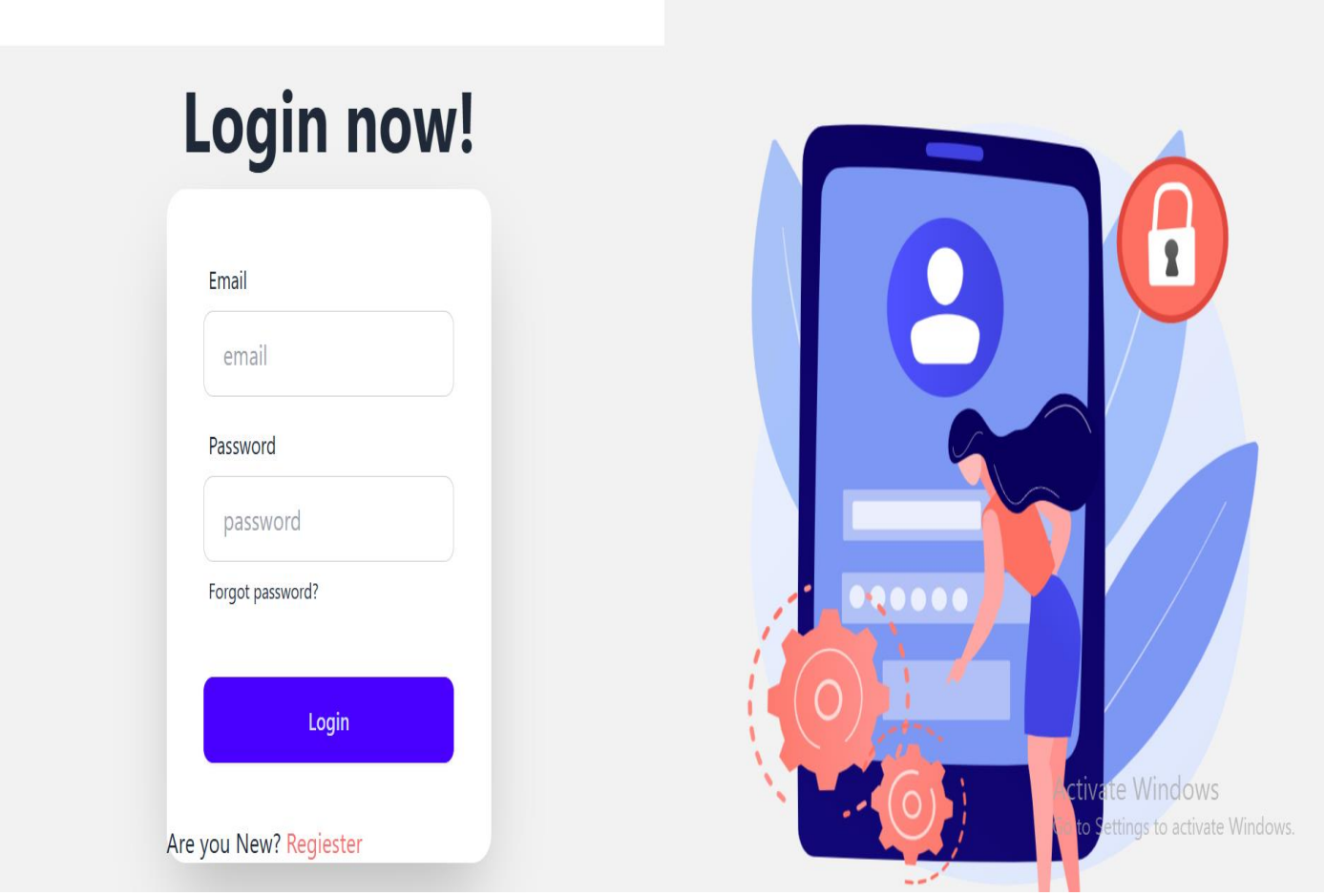


Figure 4.1: Login Page

4.2 Home Page

Description:

The Home Page serves as the main interface for users after logging in. It is aesthetically designed using Tailwind CSS and enhanced with Lottie animations for modern user experience.

Key Features:

- Dynamic product grid layout
- Product filtering by category
- Individual product detail view
- Add-to-cart functionality with real-time cart updates
- Animations for loading and empty state handling

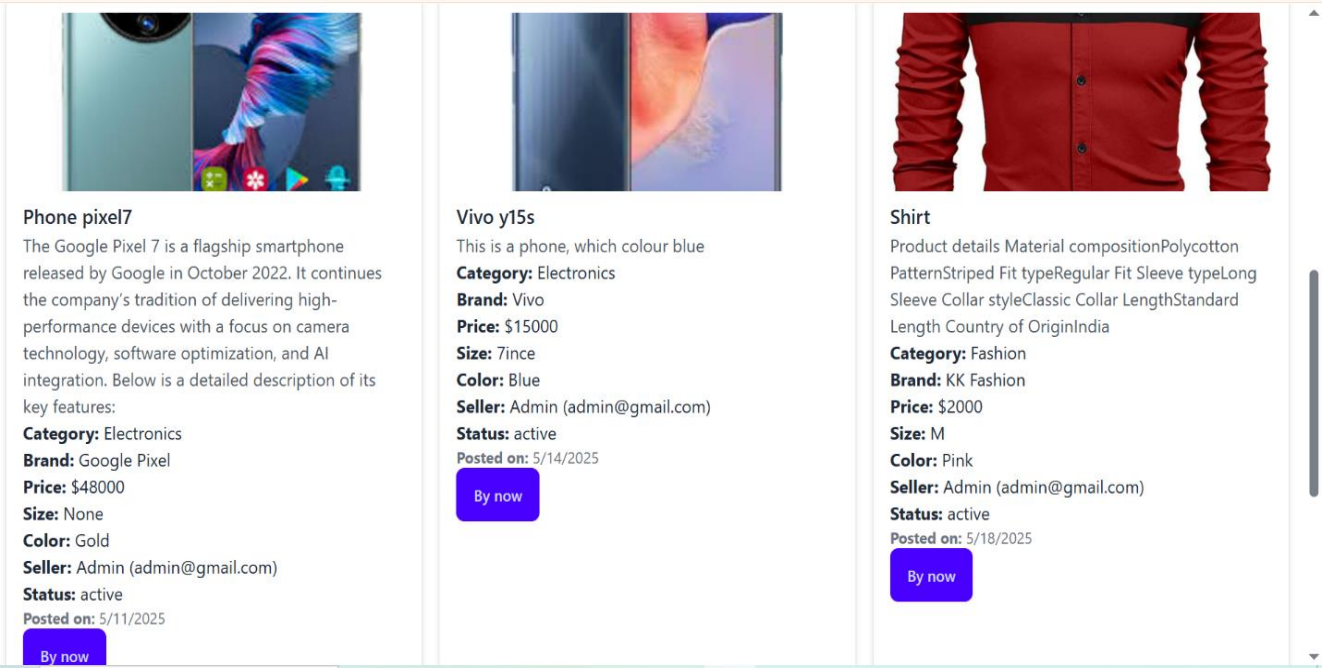


Figure 4.2: Home Page

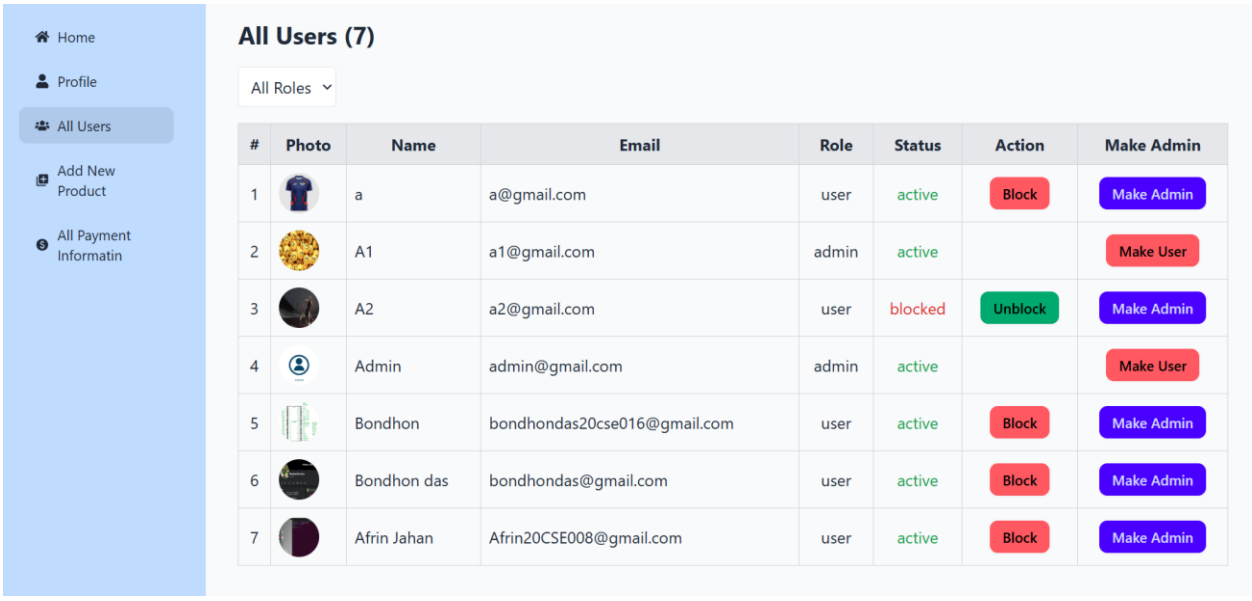


Figure 4.3: All user Page

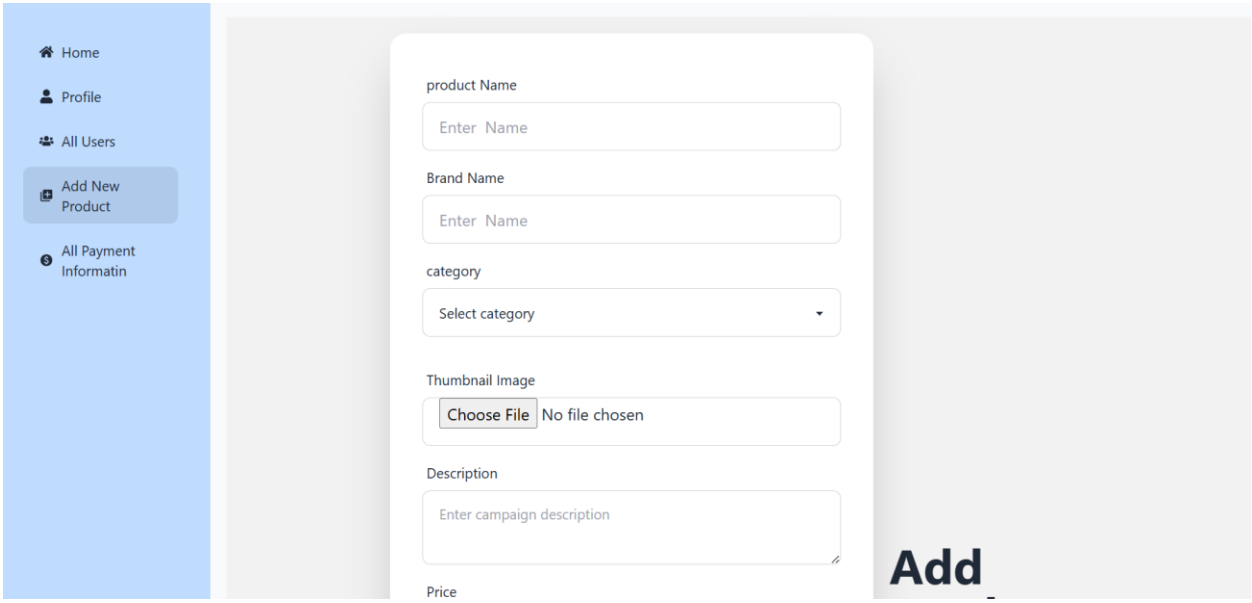


Figure 4.4: Add product Page

4.3 Checkout and Payment

Description:

The Checkout page enables users to confirm orders and make payments through either Stripe or SSLCommerz. It is a secure, form-driven module integrated with third-party payment gateways.

Key Features:

- Dynamic selection of payment method
- Real-time total cost calculation
- Stripe supports card-based international payments
- SSLCommerz supports local (BDT) payments with redirect

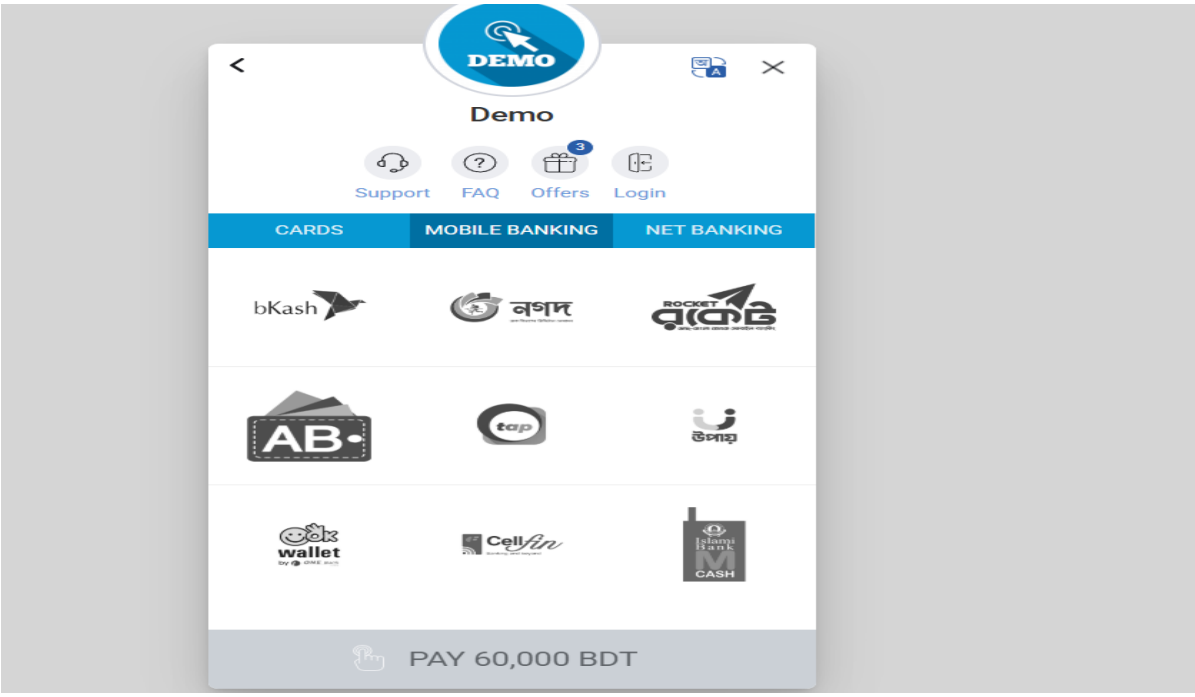


Figure 4 .5: Payment processing

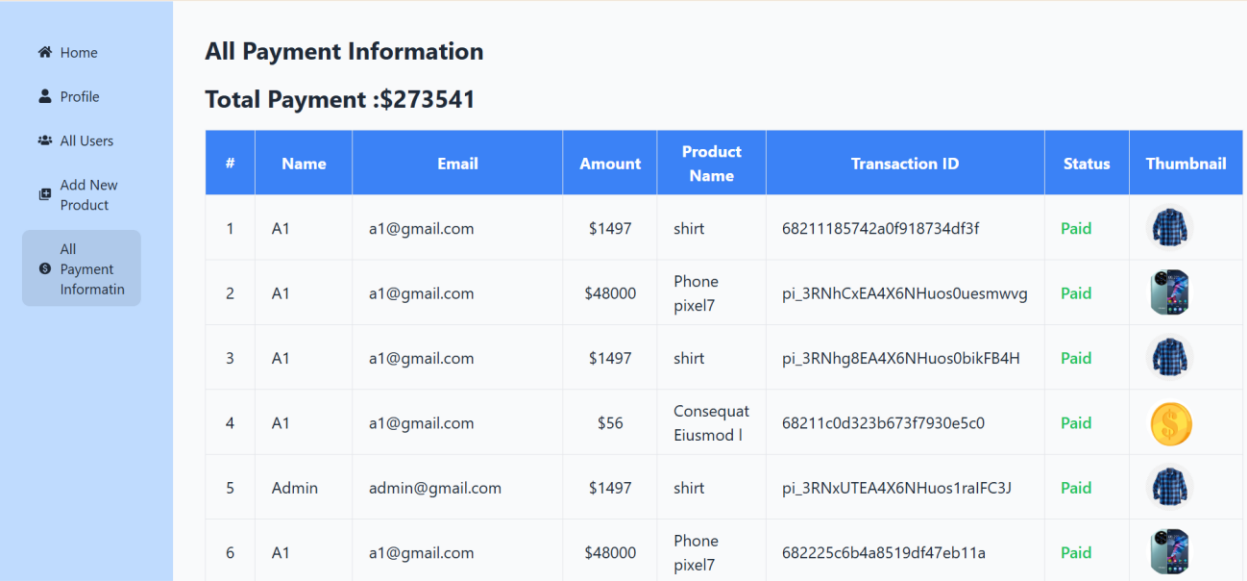


Figure 4 .6: All payment information

4.4 Admin Dashboard

Description:

The Admin Dashboard is a secure, role-based interface where administrators can manage users, products, and monitor payments. It is accessible only to users authenticated as admins.

Key Features:

- Add/edit/delete products
- View all registered users
- Manage user roles (promote/demote)
- Monitor payment history with filters
- Upload product images via Cloudinary API

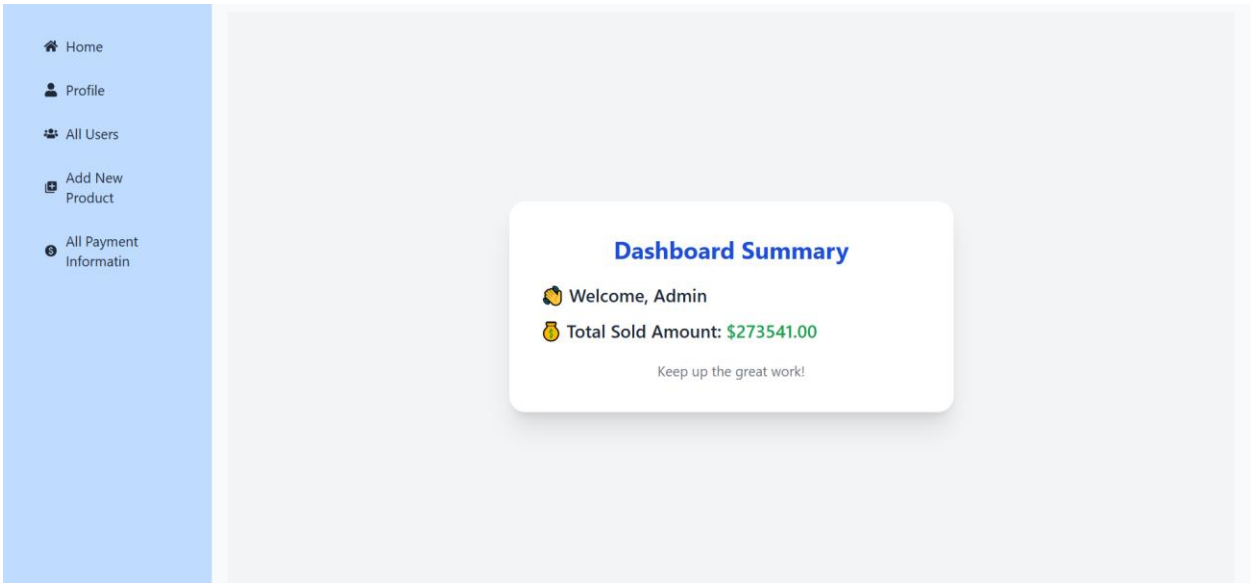


Figure 4.7: Admin dashboard

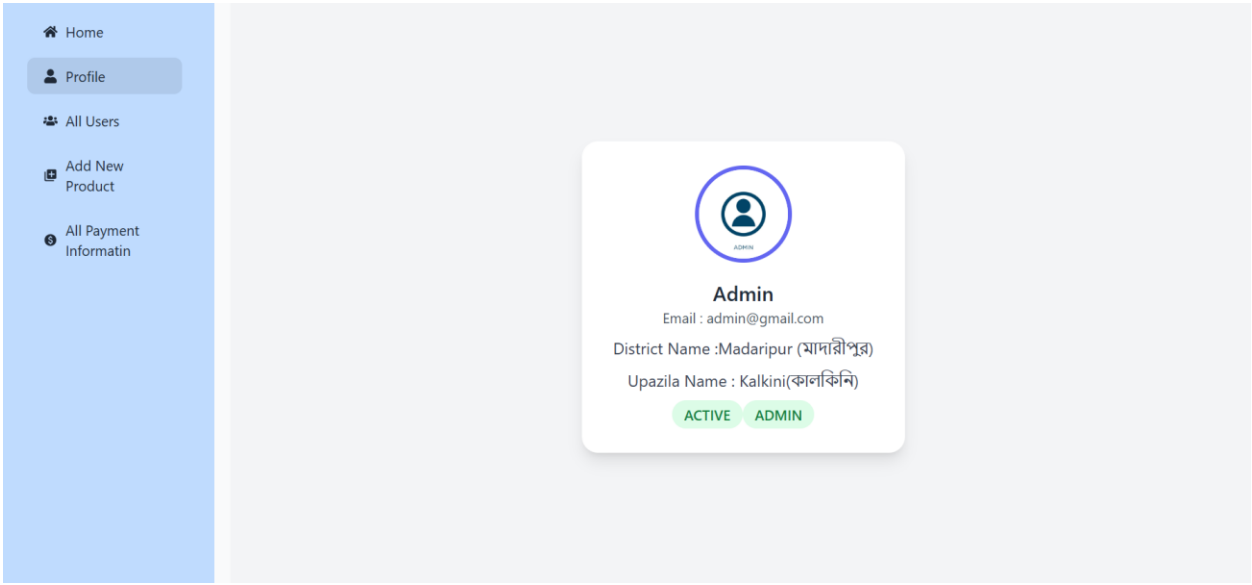


Figure 4.8: Admin Page

Chapter 5: Result and Evaluation

This chapter presents the outcomes observed after the successful development and testing of the Online Shopping Application. It evaluates the system's performance, functionality, and its ability to meet the specified requirements outlined in the objectives. The system has been thoroughly tested in terms of usability, responsiveness, data handling, security, and performance. The evaluation is based on both functional and non-functional criteria.

5.1 System Performance and Results

The developed application was successfully deployed and tested in real-time scenarios using both test and live environments for payment and user management. The evaluation confirmed that the system achieved the following outcomes:

Functional Achievements:

Module	Result
User Authentication	Fully functional using Firebase. Role-based access correctly implemented.
Product Management	Admins can add, update, and delete products. Changes reflect instantly.
Cart & Checkout	Users can add/remove products from cart. Orders are generated accurately.
Payment Integration	Stripe and SSLCommerz payments were successfully processed and recorded.
Order Tracking	Users and admins can view all previous transactions and details.
Security	JWT protects all sensitive backend routes. Firebase manages user sessions securely.

5.2 Limitations of the Project

Despite fulfilling its primary goals, the current version of the application has some limitations due to time, scope, and resource constraints:

1. No Guest Checkout – Users must register/login to place an order.
2. No Product Reviews or Ratings – A feedback system for products is missing.
3. Lack of Order Tracking via Email/SMS – Notification mechanisms have not yet been integrated.
4. No Mobile App Version – The system is web-based only.

Chapter 6: Conclusion and Future Work

6.1 Conclusion

The **Online Shopping Application** marks a significant achievement in the development of a full-fledged, scalable, and secure e-commerce platform tailored to meet the needs of both local and international users. The project successfully integrates **modern frontend technologies**, **robust backend infrastructure**, and **dual payment gateways**, providing a real-world simulation of an operational online store.

This project addressed the challenges associated with traditional manual shopping systems by delivering a responsive and interactive digital solution. From secure user authentication via Firebase to efficient role-based access and real-time payment processing through **Stripe** and **SSLCommerz**, the system demonstrates a well-architected and complete full-stack implementation.

In addition, the project adhered to the core principles of Software Engineering, including modular design, separation of concerns, data normalization, and secure API development. Usability testing and user feedback further validated that the platform is intuitive, functional, and prepared for real-world use with minor refinements.

In the context of academic learning, this project has allowed the team to gain hands-on experience in:

- Managing and integrating modern web technologies
- Implementing secure and efficient data flow between client and server
- Understanding deployment pipelines, environment configurations, and API handling
- Collaborating using Git/GitHub version control

Overall, the Online Shopping Application stands as a demonstration of both technical proficiency and the ability to engineer complex systems using modern software engineering practices.

6.2 Future Work

While the current version of the application is fully functional, there are several areas identified for enhancement to further improve system performance, usability, and scalability. The following outlines the key focus areas for future development:

1. Guest Checkout Support

- Allow users to place orders without the need for prior registration or login, making the shopping process faster and more user-friendly for casual visitors.

2. Product Review & Rating System

- Implement a feedback and rating system where users can leave reviews and rate products after purchase, enhancing trust and transparency.

3. Mobile Application Development

- Develop a cross-platform mobile application using **React Native** or **Flutter** to expand reach and provide better accessibility.

4. Admin Analytics Dashboard

- Introduce charts and statistical dashboards for the admin panel to visualize sales trends, user activity, and inventory insights using tools like **Chart.js** or **Recharts**.

5. Notification System

- Add email and/or SMS notifications for order confirmations, delivery status, and promotional campaigns using third-party services like **SendGrid** or **Twilio**.

6. Advanced Product Search & Filters

- Implement advanced filtering (by price, rating, availability) and keyword search to enhance product discoverability.

7. Enhanced Security Measures

- Implement rate limiting, CAPTCHA for authentication routes, and two-factor authentication (2FA) to improve system security.

8. Multi-language and Currency Support

- Add i18n (internationalization) support to make the system accessible in multiple languages (e.g., English, Bangla), and include currency conversion for international buyers.

References

1. React.js Documentation – https://reactjs.org/docs/getting-started.html
2. Firebase Authentication Documentation – https://firebase.google.com/docs/auth
3. Tailwind CSS Documentation – https://tailwindcss.com/docs
4. Node.js Documentation – https://nodejs.org/en/docs
5. Express.js Documentation – https://expressjs.com
6. MongoDB Documentation – https://www.mongodb.com/docs/
7. Mongoose Documentation – https://mongoosejs.com/docs/
8. Stripe API Documentation – https://stripe.com/docs/api
9. SSLCommerz Payment Gateway Developer Guide – https://developer.sslcommerz.com
10. Axios (Promise-based HTTP client) – https://axios-http.com/docs/intro
11. React Router DOM Documentation – https://reactrouter.com/en/main
12. React Toastify Documentation – https://fkhadra.github.io/react-toastify/introduction/
13. Cloudinary API Documentation
    https://cloudinary.com/documentation/image_upload_api_reference
14. dotenv – Environment Variables – https://www.npmjs.com/package/dotenv
15. CORS – Node.js Middleware – https://www.npmjs.com/package/cors
16. JWT (JSON Web Token) Introduction – https://jwt.io/introduction
17. GitHub Repository for Project – https://github.com/alamin20cse/Online-Shopping
18. Lottie Files for React – https://lottiefiles.com
19. W3Schools – Web Development Tutorials – https://www.w3schools.com
20. GeeksforGeeks – MERN Stack Tutorials – https://www.geeksforgeeks.org/tag/mern-stack/