WEBSITE FOR NIRANJAN TEXTILES

PROJECT WORK I

Submitted by

KAVIYA P

21CSR089

KAVYA P

21CSR090

LOKESH A

21CSR099

in partial fulfilment of the requirements for

the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

KONGU ENGINEERING COLLEGE

(**Autonomous)**

PERUNDURAI ERODE – 638 060

MAY 2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

KONGU ENGINEERING COLLEGE

(Autonomous)

PERUNDURAI ERODE – 638 060

MAY 2024

BONAFIDE CERTIFICATE

This is to certify that the Project report entitled **WEBSITE FOR NIRANJAN TEXTILES** is the bonafide record of the project work done by **KAVIYA P (Register No: 21CSR089), KAVYA P (Register No: 21CSR090), LOKESH A (Register No:21CSR099),** in the partial fulfilment of the requirements for the award of the Degree of Bachelor of Engineering in **Computer Science and Engineering** of Anna University Chennai during the academic year 2023 - 2024.

SUPERVISOR **HEAD OF THE DEPARTMENT**

Date: **(Signature with seal)**

ABSTRACT

This website development project is specifically designed to improve a company's digital footprint and enhance customer interaction through their e-commerce platform for online purchases. The objective is to create a visually appealing website that showcases their extensive range of products while also providing a seamless shopping experience for users.

The development process will use the MERN stack—MongoDB, Express.js, React.js, and Node.js—selected for its ability to create dynamic, high-performance web applications. React.js will handle the front-end to ensure the site is responsive and engaging, while Node.js and Express.js will manage server-side operations. MongoDB will be used for managing extensive product and user data.

The website will feature a structured layout with several key sections: a Home page that introduces the brand and features prominent products, an About page that conveys the company's history and values, a Products page with a comprehensive listing of items complete with filters for easy navigation, a Contact page for customer inquiries and additional contact information, and an Admin panel where administrators can manage product listings and monitor user activities and transactions.

Throughout the development process, the team will maintain regular communication with the company, providing updates and integrating feedback to ensure the website aligns with their needs and expectations. Quality assurance and thorough testing will be conducted to ensure that the final product is polished, functional, and meets all client specifications. This focused approach is designed to deliver a high-quality e-commerce platform that effectively supports business growth and enhances brand visibility.

KONGU ENGINEERING COLLEGE

(**Autonomous**)

PERUNDURAI ERODE – 638 060

MAY 2024

DECLARATION

We affirm that the Project Report titled **WEBSITE FOR NIRANJAN TEXTILES** being submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering is the original work carried out by us. It has not formed the part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion this or any other candidate.

Date:

KAVIYA P
21CSR089

KAVYA P
21CSR090

LOKESH A
21CSR099

I certify that the declaration made by the above candidate is true to the best of my knowledge.

Date: Name and Signature of the Supervisor

ACKNOWLEDGEMENT

We express our sincere thanks and gratitude to **Thiru. A. K. ILANGO B.Com., M.B.A., LLB.,** our beloved Correspondent, and all other philanthropic trust members of Kongu Vellalar Institute of Technology Trust who have always encouraged us in academic and co-curricular activities.

We are extremely thankful with no words of a formal nature to the dynamic Principal **Dr. V. BALUSAMY, M.Tech., Ph.D**., for providing the necessary facilities to complete our work.

We would like to express our sincere gratitude to our respected Head of the Department **Dr. S. MALLIGA, M.E., Ph.D.**, for providing the necessary facilities.

We extend our thanks to **Ms. P. KALAIVANI, M.Tech ,** Assistant Professor, Computer Science Engineering, Project Coordinator for her encouragement and valuable advice that made us carry out the project work successfully.

We extend our gratitude to our Supervisor **Dr. S. SHANTHI, M.E., Ph.D.**, Associate Professor of Computer Science Engineering, for her valuable ideas and suggestions, which have been very helpful in the project. We are grateful to all the faculty members of the Computer Science and Engineering Department, for their support.

TABLE OF CONTENTS

CHAPTER No.	TITLE	PAGE No.
	ABSTRACT	**iii**
	LIST OF FIGURES	**viii**
1.	**INTRODUCTION**	**1**
	1.1 EXISTING SYSTEM	1
	1.2 SYSTEM STUDY	2
	1.3 OBJECTIVE	3
	1.4 SCOPE	3
2.	**GENERAL DESCRIPTION**	**4**
	2.1 PROJECT PERSPECTIVE	4
	2.2 USER CHARACTERISTICS	4
	2.3 DESIGN AND IMPLEMENTATION CONSTRAINTS	5
3.	**REQUIREMENTS**	**6**
	3.1 FUNCTIONAL REQUIREMENTS	6
	3.2 NON-FUNCTIONAL REQIREMENTS	7
	3.3 USER INTERFACES	8
4.	**DETAILED DESIGN**	**13**
	4.1 ARCHITECTURAL DESIGN	13
	4.1.1 MODULE DESCRIPTION	13
	4.1.1.1 USER MODULE AND ADMIN MODULE	13
	4.1.2 USE CASE DIAGRAM	14
	4.1.3 SEQUENCE DIAGRAM	15

	4.2 INTERFACE DESIGN	15
	4.3 DATABASE DESIGN	16
	4.4 OUTPUT DESIGN	16
5.	**TESTING**	17
	5.1 UNIT TESTING	17
	5.2 REGRESSION TESTING	17
	5.3 VALIDATION TESTING	18
	5.4 VERIFICATION TESTING	18
	5.5 INTEGRATION TESTING	18
6.	**CONCLUSION AND FUTURE SCOPE**	19
	APPENDIX 1	20
	APPENDIX 2	41
	REFERENCES	45

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE NO.
3.1	HOME PAGE	8
3.2	ABOUT US	8
3.3	PRODUCT PAGE	9
3.4	CART PAGE	9
3.5	PLACE ORDER PAGE	10
3.6	CONTACT US PAGE	10
3.7	MANAGE PRODUCTS PAGE	11
3.8	MANAGE USERS PAGE	11
3.9	INVOICE	12
4.1	FLOWCHART	13
4.2	USECASE DIAGRAM	14
4.3	SEQUENCE DIAGRAM	15
4.4	DATABASE DESIGN	16
A2.1	HOME PAGE	41
A2.2	ABOUT US	41
A2.3	PRODUCT PAGE	42
A2.4	CART PAGE	42
A2.5	PLACE ORDER PAGE	43
A2.6	CONTACT US PAGE	43
A2.7	MANAGE PRODUCTS PAGE	44
A2.7	MANAGE USERS PAGE	44

CHAPTER 1

INTRODUCTION

Niranjan Textiles, a distinguished name in the textile industry, is celebrated for its dedication to preserving the rich heritage of traditional saree-making while incorporating modern designs. This family-owned business combines generations of artisanal expertise with contemporary fashion sensibilities to offer a unique collection of sarees that cater to both classic and modern tastes. Committed to quality and innovation, Niranjan Textiles crafts each saree with a blend of meticulous attention to detail and the finest materials, ensuring that each piece is not only beautiful but also a testament to the art of saree-making. Their offerings are designed to enhance the elegance of every wearer, making Niranjan Textiles a trusted partner in the celebration of Indian culture and craftsmanship.

1.1 EXISTING SYSTEM

In the existing system, Niranjan Textiles lacks an online platform, which is a substantial drawback in today's digital-first economy. As a start-up, they have no website to establish their online identity, leaving potential customers unaware of their brand. This absence represents a significant gap in their market reach, preventing them from showcasing their unique textile offerings and from engaging with a broader audience.

The lack of a website also restricts their ability to communicate their offerings effectively, display their diverse range of products, and attract potential customers. Moreover, without an online platform, Niranjan Textiles misses out on establishing credibility and building trust with potential buyers who increasingly rely on the internet to research and make purchasing decisions. A dedicated e-commerce website would enable

them to highlight their unique value propositions, differentiate themselves in the competitive market, and position themselves as a top choice for customers seeking quality textiles.

1.2 SYSTEM STUDY

1.2.1 Understanding the Business Requirements

The first step in the system study is to understand the business requirements of the project. This involves identifying the business objectives, target audience, and the features and functionalities required in the website.

1.2.2 Analyzing the Existing Systems and Processes

The next step is to analyze the existing systems and processes in place, including the current website and any related software systems. This will help identify any limitations or areas for improvement in the new website.

1.2.3 Identifying User Needs

The system study should also involve identifying the needs of the users who will be accessing the website. This can be done through surveys, focus groups, or other user research methods.

1.2.4 Developing Use Cases

Use cases can be developed to illustrate how the website will function and how users will interact with it. This can help identify any potential issues or areas for improvement in the design and functionality.

1.3 OBJECTIVE

The key objective of this project is to elevate the digital footprint of Niranjan Textiles by developing a captivating and professional e-commerce website that displays their diverse selection of textile products.

1.4 SCOPE

The scope of the project being developed for Niranjan Textiles is to create a highly functional and user-friendly e-commerce website that effectively showcases their extensive range of sarees. The site will not only display their products but also highlight the quality and variety of textiles they offer.

To ensure the success of the project, we will work closely with Niranjan Textiles to fully understand their requirements and expectations, and we will provide regular progress updates throughout the development process. Our ultimate goal is to deliver a website that exceeds Niranjan Textiles' expectations and effectively enhances its online establishment, making it easy for customers to explore and purchase their products.

CHAPTER 2

GENERAL DESCRIPTION

2.1 PROJECT PERSPECTIVE

- The website will serve as Niranjan Textiles' primary online platform and a critical investment for growth and success in the competitive textile market.
- The website will function as a marketing tool to showcase Niranjan Textiles' diverse range of products and facilitate online sales, enhancing customer engagement and revenue.
- The development team will need to possess expertise in modern web development technologies and integration with e-commerce and CMS platforms to ensure the website's functionality, security, and performance are optimal.

2.2 USER CHARACTERISTICS

The admin can enter, view, and update the details of the sarees that are available for sale on the platform. Using the login form, customers can enter their details like name and email ID to register and interact with the company. On the contact page, customers can also communicate with the company, providing inquiries or requesting support. They need to enter their personal details such as name and email ID to approach the company through this page. The admin can view these communications to better understand customer needs and enhance customer service.

2.3 DESIGN AND IMPLEMENTATION CONSTRAINTS

2.3.1 Time

There may be a deadline or time constraint for the development of the website, which can impact the design and implementation process.

2.3.2 Security

The website may need to meet certain security requirements and secure login features, which can impact the design and implementation process.

2.3.3 Scalability

The website may need to be designed and developed with scalability in mind, to accommodate future growth or changes in the company's needs.

2.3.4 User Experience

The website may need to meet certain user experience standards, such as accessibility or ease of use, which can impact the design and implementation process.

CHAPTER 3
REQUIREMENTS

3.1 FUNCTIONAL REQUIREMENTS

Functional requirements define the specific behaviors, tasks, and functionalities a system must perform, including data processing and user interactions.

3.1.1 User Registration and Login

Users should be able to register for an account and log in to access the website's features and content.

3.1.2 About Us

The website should have an "About Us" page that provides a company description.

3.1.3 Contact Forms

The website should include contact forms that allow users to get in touch with the company or its representatives.

3.1.4 Products Page

The 'Products' page on company’s website showcases a diverse portfolio of sarees, highlighting the company’s expertise in crafting traditional and contemporary designs to attract and inform potential customers.

3.1.5 Invoice generation

Upon order placement, the system will automatically generate and send invoices to customers via email. This functionality ensures efficient processing of transactions and provides customers with timely documentation of their purchases.

3.2 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements define how a system performs specified functions, focusing on usability, reliability, performance, and security, ensuring effective, efficient, and secure operation under diverse conditions.

3.2.1 Performance

The website should be designed to load quickly and respond to user interactions promptly, to ensure a positive user experience.

3.2.2 Usability

The website should be easy to use and navigate, with a clear and intuitive user interface, to ensure a positive user experience.

3.2.3 Reliability

The website should be reliable and available for use at all times, with minimal downtime or interruptions.

3.2.4 Maintenance

The website should be easy to maintain and update, with a well-organized codebase and documentation, to minimize maintenance and support costs.

3.3 USER INTERFACES

HOME PAGE:

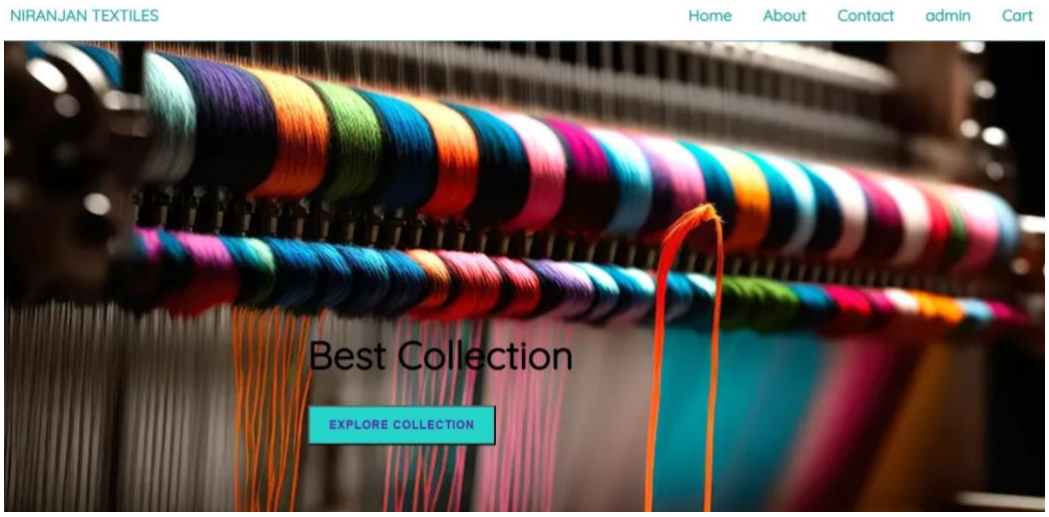

FIGURE 3.1 HOME PAGE

ABOUT US PAGE:

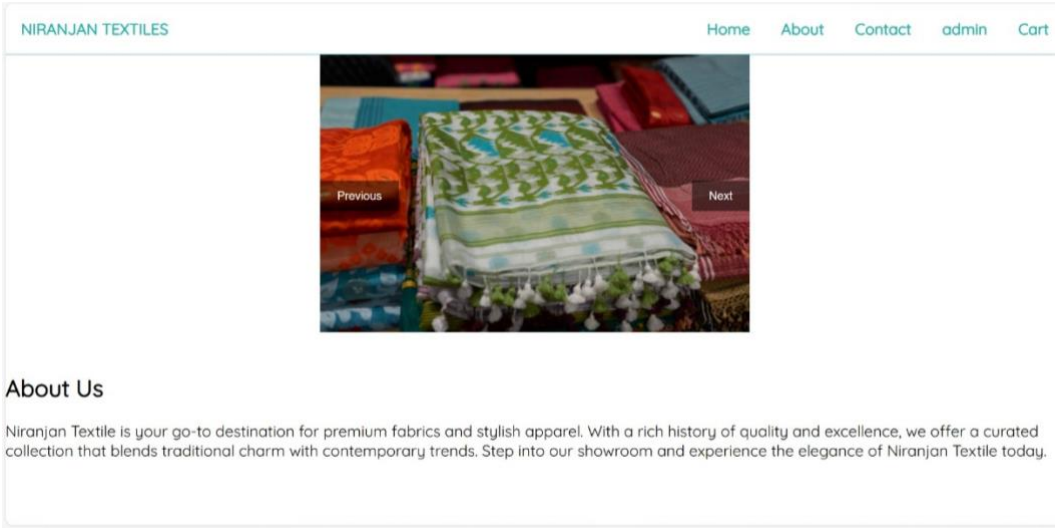

FIGURE 3.2 ABOUT PAGE

PRODUCTS PAGE:

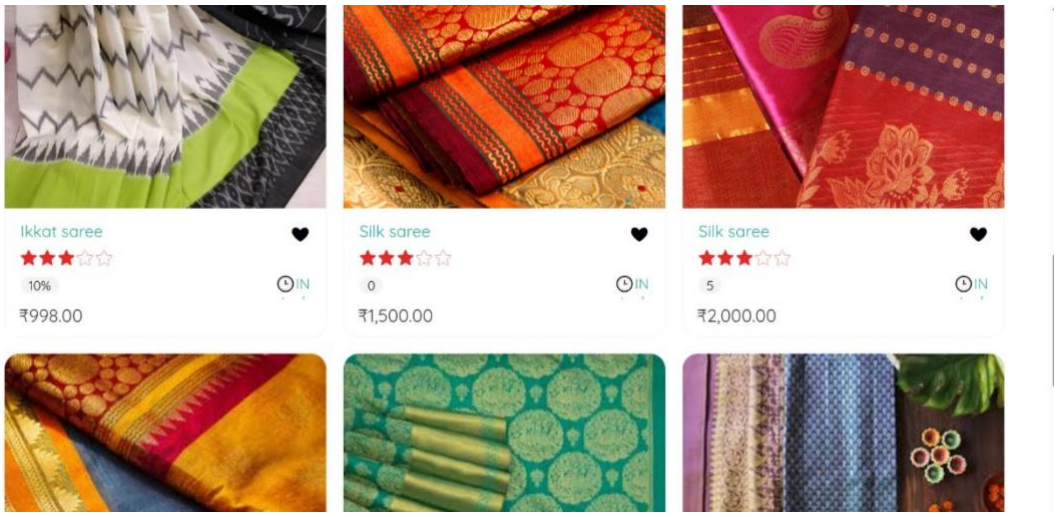

FIGURE 3.3 PRODUCTS PAGE

CART PAGE:

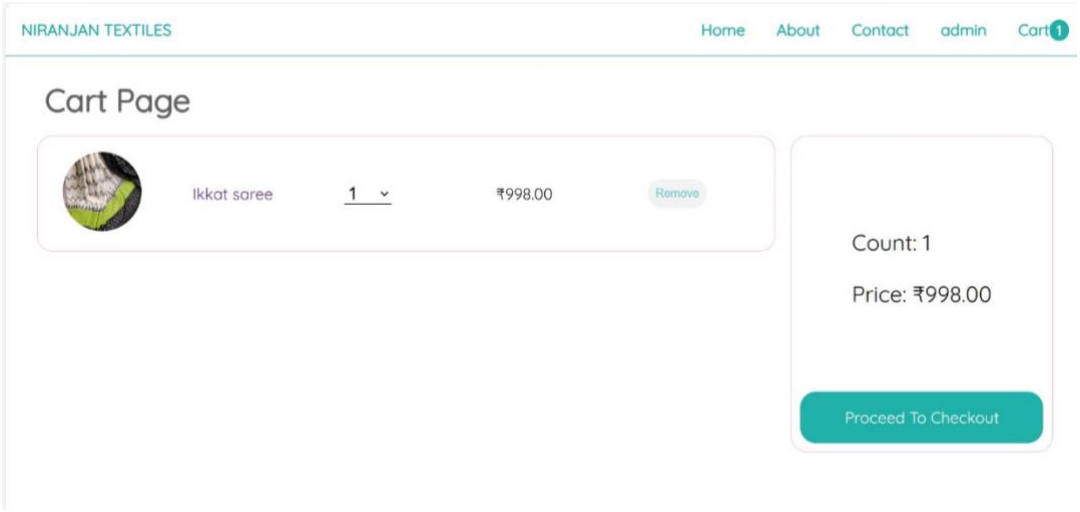

FIGURE 3.4 CART PAGE

PLACE ORDER PAGE:

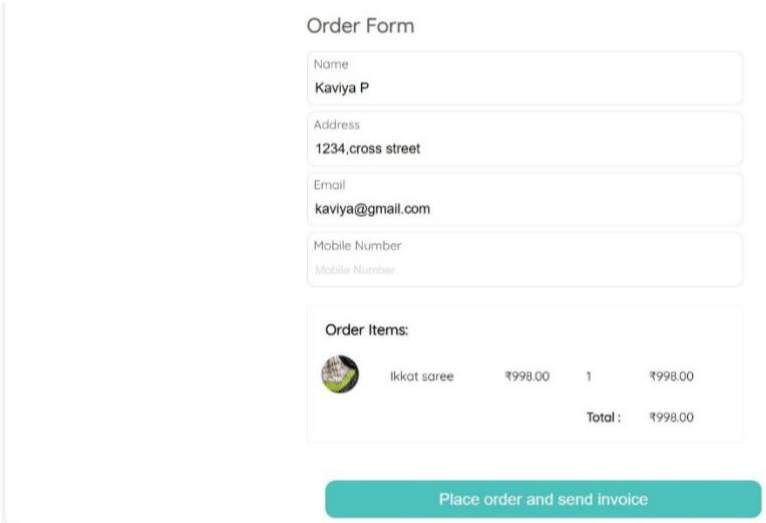

FIGURE 3.5 PLACE ORDER PAGE

CONTACT US PAGE:

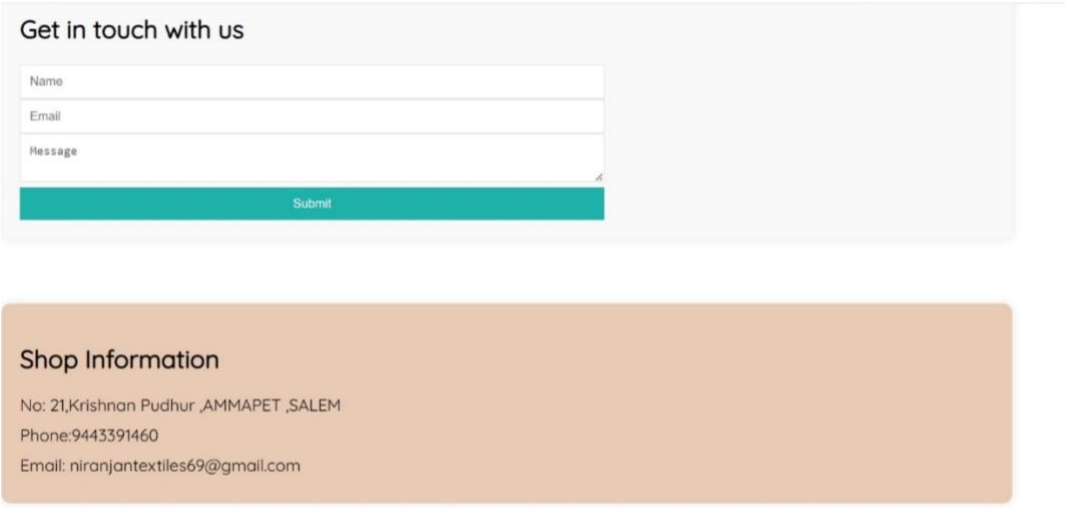

FIGURE 3.6 CONTACT PAGE

ADMIN PANEL

MANAGE PRODUCTS PAGE:

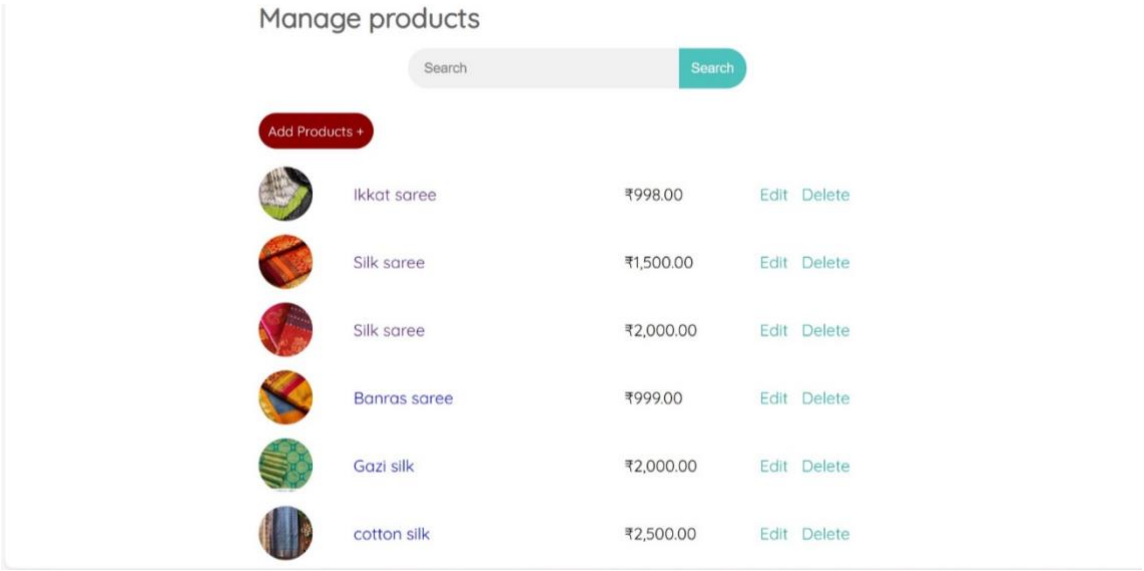

FIGURE 3.7 MANAGE PRODUCTS PAGE

MANAGE USERS PAGE:

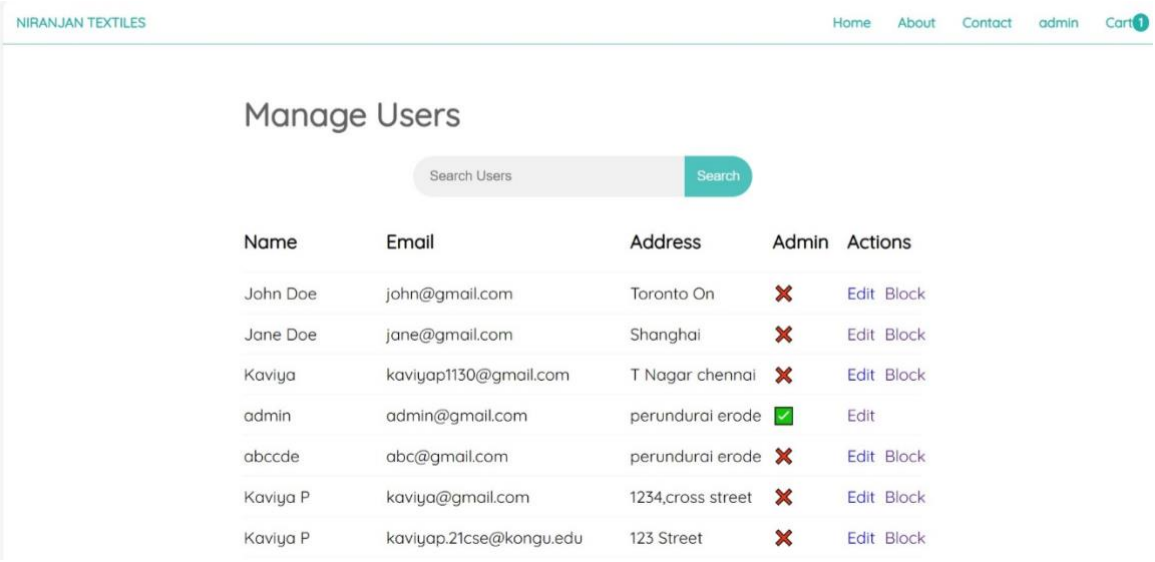

FIGURE 3.8 MANAGE USERS PAGE

INVOICE :

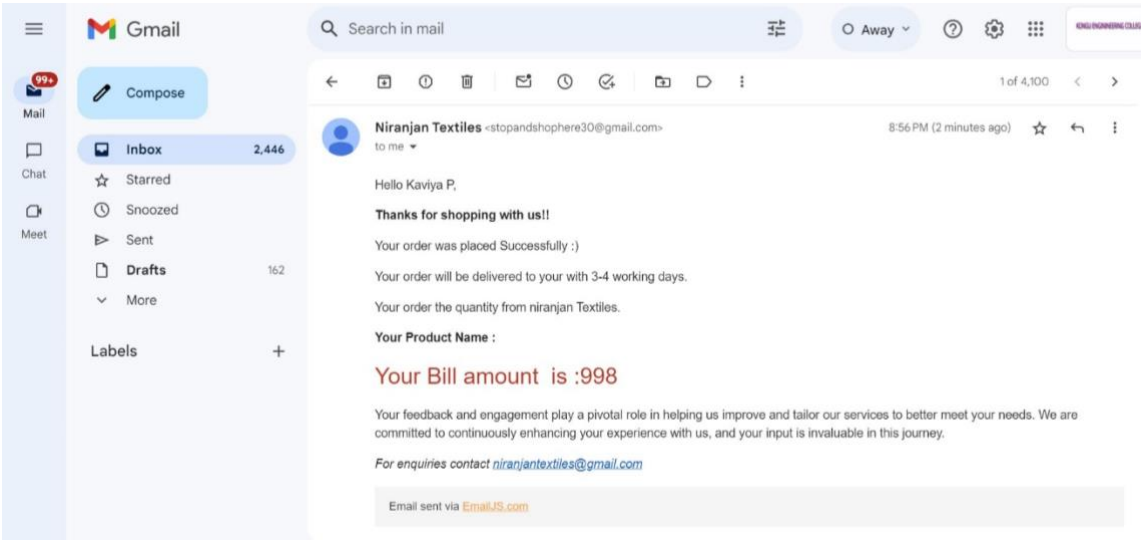

FIGURE 3.9 INVOICE TO MAIL

CHAPTER 4

DETAILED DESIGN

4.1 ARCHITECTURAL DESIGN

The architectural design includes various diagrams such as use case diagram, sequence diagram and activity diagram for both user and admin modules.

4.1.1 MODULE CHARACTERISTICS

The project contains two modules as follows and their description is given below.

- User module
- Admin module

4.1.1.1 USER MODULE AND ADMIN MODULE

The user module contains a home page, products page, about us page, and contact us page. The user can view all the pages as mentioned in the user module flow.

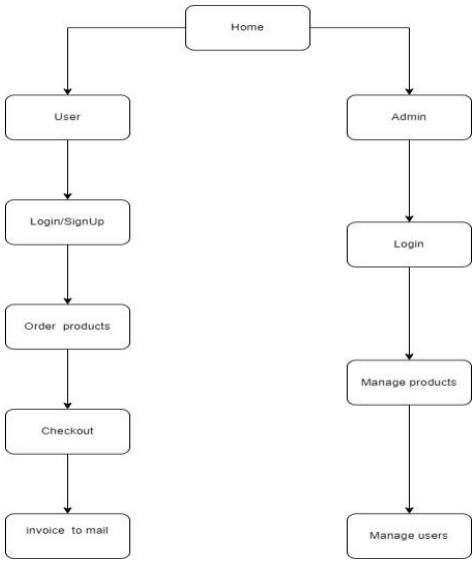

FIGURE 4.1 FLOWCHART

4.1.2 USE CASE DIAGRAM

A use case diagram is a dynamic or behaviour diagram in UML. Use case diagrams model the functionality of a system using actors and use cases. Use cases are set of actions, services, and functions that the system needs to perform. A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. The use cases are represented by either circles or ellipses. Due to their simplistic nature, use case diagrams can be a good communication tool for stakeholders

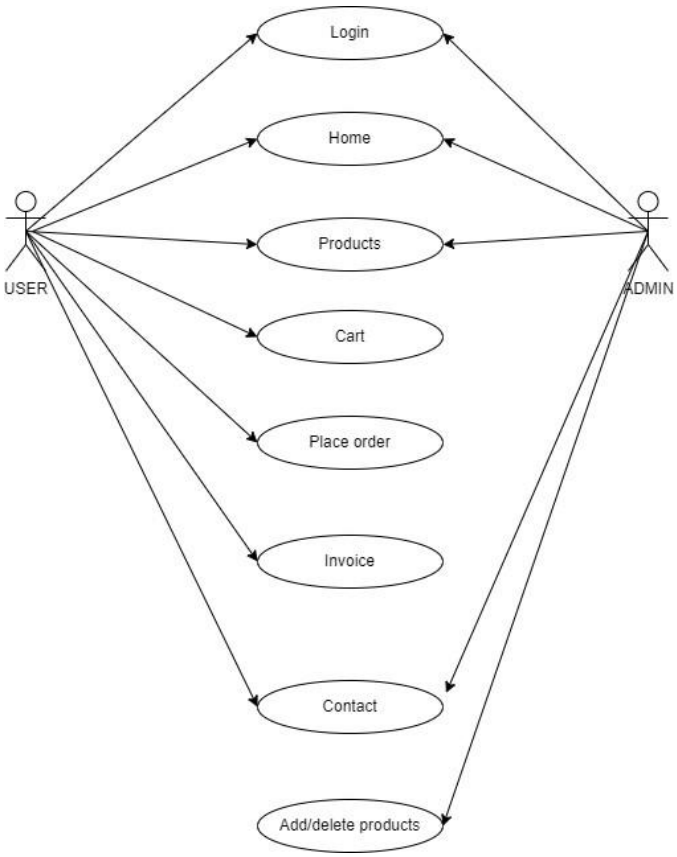

FIGURE 4.2 USE CASE DIAGRAM

4.1.3 SEQUENCE DIAGRAM

Sequence chart diagram is one of the five UML diagrams used to model the dynamic nature of a system. The admin defines the sequence of an object during its lifetime.

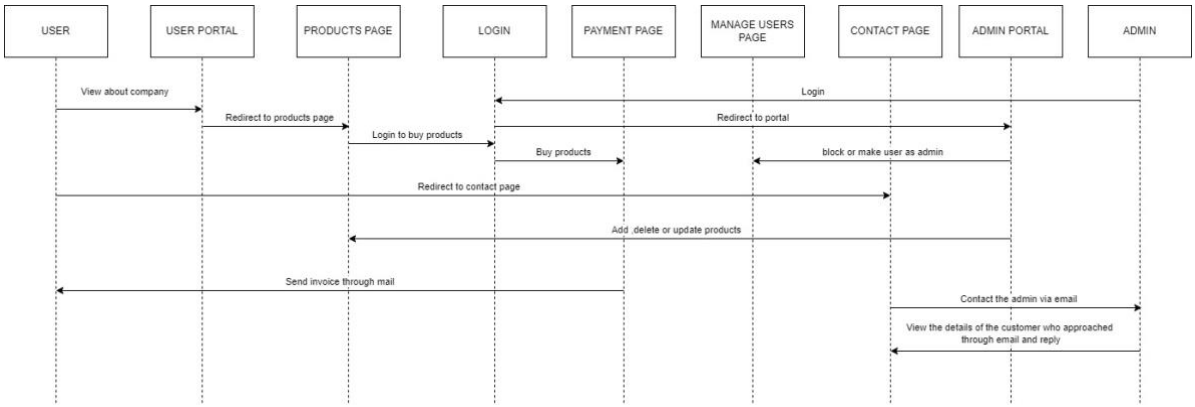

FIGURE 4.3 SEQUENCE DIAGRAM

4.2 INTERFACE DESIGN

The application provides various interfaces to the users that make them more convenient with the application. The user interface contains Home, About, and Products showcasing the diverse sarees offered by Niranjan Textiles. Additionally, users can contact the company by providing their details through the Contact page. This layout is simple to use and clear to understand. Users can approach the company with inquiries or support requests. The user interface is attractive and consistent across all interface screens, enhancing the shopping experience.

4.3 DATABASE DESIGN

The Database used in this system is MongoDB. In Mongo, the data is stored in the form of collections and documents. Key collections include Services, Projects, Admins, and Feedback. These collections facilitate operations such as storing, updating, and deleting company projects, managing admin roles, and capturing customer feedback. Additionally, customer details like name, email ID, and phone number are stored for those who contact the company via email.

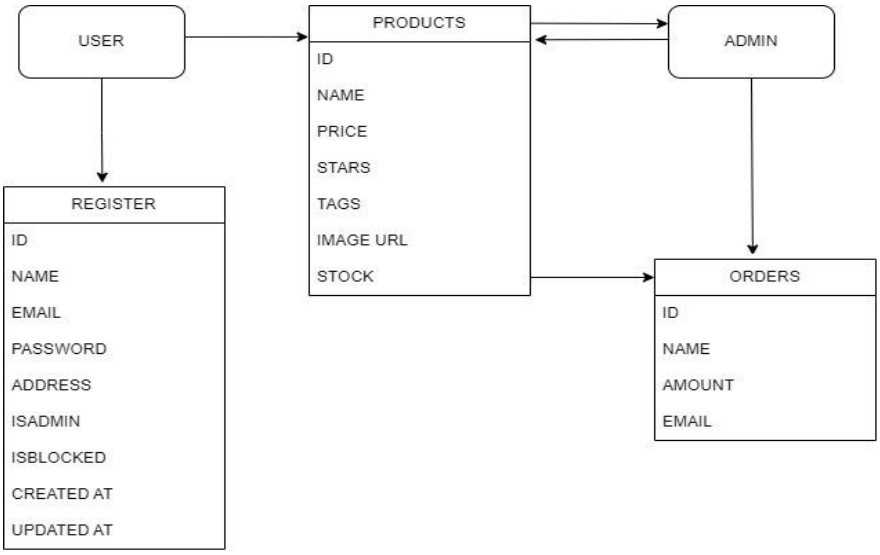

FIGURE 4.4 DATABASE DESIGN

4.4 OUTPUT DESIGN

Output design generally refers to the results and information that are generated by the system for many end-users; it should be understandable with the enhanced format. Previewing the output reports by the user is extremely important because the user is the ultimate judge of the quality of the output and, in turn, the success of the system. The output is designed in such a way that it is attractive, convenient and informative.

CHAPTER 5
TESTING

5.1 UNIT TESTING

Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation. Unit testing can be done manually, but is often automated. In this process, each module is considered as individual units and are tested for proper operation. If each module meets up with the user's requirement, then it is subjected to integration testing where more than one module is integrated and tested.

5.2 REGRESSION TESTING

Regression testing helps in identifying any unintended side effects or issues that may arise due to code changes, updates to dependencies, or modifications in the underlying infrastructure. By rerunning test cases, the development team can ensure that previously implemented features and functionality have not been negatively impacted by recent updates. The primary objective of regression testing is to ensure that the website remains fully functional, reliable, and user-friendly, even after modifications or enhancements have been made.

Overall, regression testing serves as an essential quality assurance measure to maintain the integrity and performance of the website throughout its lifecycle. It helps minimize the risk of functional regressions, ensuring that the website continues to meet the expectations and requirements of the end users and stakeholders.

5.3 VALIDATION TESTING

The process of evaluating software during the development process or at the end of the development process to determine whether it satisfies specified business requirements. Validation Testing ensures that the service actually meets the client's needs. It can also be defined as to demonstrate that the service fulfils its intended use when deployed on appropriate environment.

5.4 VERIFICATION TESTING

Verification is the process of evaluating work-service of a development phase to determine whether they meet the specified requirements. When developing each module, the individuality of the service is checked at its development stage. Thus, the modules must be verified at the development stage.

5.5 INTEGRATION TESTING

Integration tests are designed to test the integrated software components to determine if they actually run as a program. It is specifically aimed at exposing problems that arise from the combination of components. Integration testing is done after integration of the model with the core service. The integration testing can be done for our project by integrating the user module.

CHAPTER 6
CONCLUSION AND FUTURE WORK

Niranjan Textiles started as a startup company without a dedicated platform to showcase their sarees to potential customers. Even now, they rely on manually showcasing their saree collections, resulting in lengthy explanations of their products each time a customer approaches them for information. To simplify the process, we have designed a user interface that facilitates communication with clients, streamlining the process of understanding our offerings.

Our web application ensures compatibility across various browsers, making it convenient for users to access the platform. We are constantly looking for ways to improve our services, and we plan to integrate payment options into our platform shortly. In addition, we are also exploring the development of a mobile application to make it easier for customers to browse and purchase our sarees.

Extensive testing has confirmed high levels of end-user satisfaction with the platform's functionality and usability. Looking ahead, we are excited about the potential for significant growth and expansion of our project, aligning with our evolving requirements and ensuring continued customer satisfaction.

APPENDIX 1

GITHUB LINK: https://github.com/KAVIYAP30/TextileApp

CODING:

App.js:

```
function App() {
  const { showLoading, hideLoading } = useLoading();

  useEffect(() => {
    setLoadingInterceptor({ showLoading, hideLoading });
  }, []);

  return (
    <>
      {/* <Loading /> */}
      <Header />
      <AppRoutes />
      {/* <Footer/> */}
    </>
  );
}

export default App;
```

About Us:

```
const About = () => {
  const [currentImageIndex, setCurrentImageIndex] = useState(0);
  const images = [
    'https://images.unsplash.com/photo-1616756141603-6d37d5cde2a2?w=600&auto=format&fit=crop&q=60&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxzZWFyY2h8MjB8fHNhcmVlfGVufDB8fDB8fHww',
    'https://media.istockphoto.com/id/1355125086/photo/indian-handloom-sarees-with-vibrant-colours-and-beautiful-designs.jpg?s=612x612&w=0&k=20&c=zoF_CAMr-eijNWjVvDp2nzTZ4m13yP9t3YDgn5rbH6w=',
    'https://media.istockphoto.com/id/93355119/photo/indian-saris.jpg?s=612x612&w=0&k=20&c=afmfiTJg0VAmIY6P_TJ_JYsTfGhUdevv18WXQRUZ8NQ=',
    'https://media.istockphoto.com/id/153515439/photo/pink-indian-fabric.jpg?s=612x612&w=0&k=20&c=9XSa_ZRdM8B1M_jKgTMNFS2xuer7ayiDBjzZ-ZqMUJ0='
  ];

  useEffect(() => {
    const interval = setInterval(() => {
      nextSlide();
    }, 1000); // Change interval time (in milliseconds) as needed

    return () => clearInterval(interval);
  }, [currentImageIndex]); // Restart interval on slide change

  const nextSlide = () => {
```

```
    setCurrentImageIndex((prevIndex) => (prevIndex === images.length - 1 ? 0 : prevIndex + 1));
  };

  const prevSlide = () => {
    setCurrentImageIndex((prevIndex) => (prevIndex === 0 ? images.length - 1 : prevIndex - 1));
  };

  return (
    <>
      <div className="about">
        <div className="slider">
          <button className="prev" onClick={prevSlide}>Previous</button>
          <img src={images[currentImageIndex]} alt="Gift box" />
          <button className="next" onClick={nextSlide}>Next</button>
        </div>
        <div className="text">
          <h2>About Us</h2>
          <p> Niranjan Textile is your go-to destination for premium fabrics and stylish apparel. With a rich history of quality and excellence, we offer a curated collection that blends traditional charm with contemporary trends. Step into our showroom and experience the elegance of Niranjan Textile today.</p>
        </div>
      </div>

      <style>{`
        .about {
          display: flex;
          align-items: center;
```

```
  justify-content: center;
  flex-direction: column;
}

.slider {
  position: relative;
  width: 500px; /* Adjust width as needed */
  margin-bottom: 20px;
}

.slider img {
  width: 100%;
  height: auto;
}

.prev,
.next {
  position: absolute;
  top: 50%;
  transform: translateY(-50%);
  background-color: rgba(0, 0, 0, 0.5);
  color: #fff;
  border: none;
  padding: 10px 20px;
  cursor: pointer;
}

.prev {
```

```
        left: 0;
      }

      .next {
        right: 0;
      }

      .vision-mission-container {
        margin-top: 20px;
        padding: 20px;
        background-color: lightseagreen
        ;
        border-radius: 8px;
        text-align: center;
      }

      .vision-mission-container h2 {
        margin-bottom: 10px;
      }

      .vision-mission-container p {
        margin-bottom: 5px;
      }
    `}</style>
  </>
 );
};
```

```
export default About;
```

Cart:

```
export default function CartPage() {
  const { cart, removeFromCart, changeQuantity } = useCart();

  const handleQuantityChange = (item, quantity) => {
    changeQuantity(item.food.id, quantity);
  };

  return (
    <>
      <Title title="Cart Page" margin="1.5rem 0 0 2.5rem" />

      {cart.items.length === 0 ? (
        <NotFound message="Cart Page Is Empty!" />
      ) : (
        <div className={classes.container}>
          <ul className={classes.list}>
            {cart.items.map(item => (
              <li key={item.food.id}>
                <div>
                  <img src={`${item.food.imageUrl}`} alt={item.food.name} />
                </div>
                <div>
```

```
      <Link to={`/food/${item.food.id}`}>{item.food.name}</Link>
    </div>

    <div>
      <select
        value={item.quantity}
        onChange={e => handleQuantityChange(item, Number(e.target.value))}
      >
        {[...Array(10)].map((_, index) => (
          <option key={index + 1} value={index + 1}>{index + 1}</option>
        ))}
      </select>
    </div>

    <div>
      <Price price={item.price * item.quantity} />
    </div>

    <div>
      <button
        className={classes.remove_button}
        onClick={() => removeFromCart(item.food.id)}
      >
        Remove
      </button>
    </div>
  </li>
))}
```

```
        </ul>

        <div className={classes.checkout}>
          <div>
            <div className={classes.foods_count}>{cart.totalCount}</div>
            <div className={classes.total_price}>
              <Price price={cart.totalPrice} />
            </div>
          </div>

          <Link to="/checkout">Proceed To Checkout</Link>
        </div>
      </div>
    )}
  </>
 );
}
```

Checkout:

```
import OrderItemsList from '../../components/OrderItemsList/OrderItemsList';

export default function CheckoutPage() {
  const { cart } = useCart();
  const { user } = useAuth();
  const navigate = useNavigate();
  const [order, setOrder] = useState({ ...cart });

  const {
```

```
    register,
    formState: { errors },
    handleSubmit,
  } = useForm();


  const submit = async data => {
    await createOrder({ ...order, name: data.name, address: data.address, mobile:
data.mobile });



    const templateParams = {
      to_name: data.name,
      address: data.address,
      to_email: data.email,
      to_quantity: order.quantity,
      to_price: order.totalPrice,


    };


    emailjs.send("service_w7jjaq1", "template_h9l1lpz", templateParams,
"swLGgX2b4zzuUfGvA")
      .then((response) => {
        console.log("Email sent successfully", response);
        toast.success("Order placed Successfully and invoice sent to your mail");
      })
      .catch((error) => {
        console.log("Failed to send email", error);
        toast.error("Oops error occurred");
      });
```

```
  toast.success('Order placed successfully!');
};

return (
  <>
    <form onSubmit={handleSubmit(submit)} className={classes.container}>
      <div className={classes.content}>
        <Title title="Order Form" fontSize="1.6rem" />
        <div className={classes.inputs}>
          <Input
            defaultValue={user.name}
            label="Name"
            {...register('name')}
            error={errors.name}
          />
          <Input
            defaultValue={user.address}
            label="Address"
            {...register('address')}
            error={errors.address}
          />
          <Input
            defaultValue={user.email}
            label="Email"
            {...register('email')}
            error={errors.email}
          />
```

```
            <Input
              label="Mobile Number"
              {...register('mobile', { required: true })}
              error={errors.mobile}
            />
          </div>
          <OrderItemsList order={order} />
        </div>

        <div className={classes.buttons_container}>
          <div className={classes.buttons}>
            <Button
              type="submit"
              text="Place order and send invoice"
              width="100%"
              height="3rem"
            />
          </div>
        </div>
      </form>
    </>
  );
}
```

Contact:

```
  const Contact = () => {
  const [submitted, setSubmitted] = useState(false);
```

```
const handleSubmit = (event) => {
  event.preventDefault();
  window.location.reload();
};

return (
  <div className="contact-page">
    <div className="contact-form">
      <h2>Get in touch with us</h2>
      <form onSubmit={handleSubmit}>
        <input type="text" placeholder="Name" />
        <input type="email" placeholder="Email" />
        <textarea placeholder="Message"></textarea>
        <button type="submit">Submit</button>
      </form>
      {submitted && <p>Thank you for your suggestions!</p>}
    </div>

    <div className="company-info">
      <h2>Shop Information</h2>

      <p>No: 21,Krishnan Pudhur ,AMMAPET ,SALEM </p>
      <p>Phone:9443391460  </p>
      <p>Email: niranjantextiles69@gmail.com</p>
    </div>

    <div className="customer-support-hours">
      <h2>Customer Support Hours</h2>
```

```
        <p>Monday - Saturday: 9:00 AM to 9:00 PM</p>
        <p>Sunday: Closed</p>
      </div>




      {/* <div className="map-directions"> */}
        {/* <h2>Map & Directions</h2> */}
         {/* <ContactMap/> */}

        {/* <p>PHOENIX GIFTS CUDDALORE-607 002</p> */}
      </div>
    // </div>
  );
};

export default Contact;
```

Payment:

```
export default function PaymentPage() {
  const [order, setOrder] = useState();

  const navigate = useNavigate();
  const [{ isPending }] = [false];
  const { showLoading, hideLoading } = useLoading();
  useEffect(() => {
    isPending ? showLoading() : hideLoading();
```

```
});
useEffect(() => {
  getNewOrderForCurrentUser().then(data => setOrder(data));
}, []);


const { clearCart } = useCart();

if (!order) return;

const initPayment = (data) => {
  const options = {
    key: "rzp_test_2SuRloEn2Xmn8F",
    amount: data.amount,
    currency: data.currency,
    name: order.items[0].food.name,
    // description: data.items[0].food.description,
    image: order.items[0].food.imageUrl,
    order_id: data.id,
    handler: async (response) => {
      try {
        const verifyUrl = "/api/orders/verify";
        const { razorpay_order_id, razorpay_payment_id, razorpay_signature } = response;
        const verificationData = {
          razorpay_order_id,
          razorpay_payment_id,
          razorpay_signature
        };
```

```
        const { data } = await Axios.post(verifyUrl, verificationData);
        console.log(data);
        if(data.success===true){
          const { data } = await Axios.put("/api/orders/pay",{razorpay_payment_id});
          console.log(data);
          clearCart();
          toast.success('Payment Saved Successfully', 'Success');
          navigate('/track/' + data.order_id);
        }
      } catch (err) {
        console.log("Verification error:", err);
      }
    }
  };
  const razorpay1 = new window.Razorpay(options);
  razorpay1.open();
};

async function placeOrderHandler(){
  try{
    const {data}=await Axios.post('https://textileapp-2.onrender.com/api/orders/orders',
      {
        amount: order.totalPrice,
      });
      console.log(data);
      initPayment(data.data);
    }
    catch(error){
```

```
      console.log(error);
    }
}

return (
  <>
    <div className={classes.container}>
      <div className={classes.content}>
        <Title title="Order Form" fontSize="1.6rem" />
        <div className={classes.summary}>
          <div>
            <h3>Name:</h3>
            <span>{order.name}</span>
          </div>
          <div>
            <h3>Address:</h3>
            <span>{order.address}</span>
          </div>
        </div>
        <OrderItemsList order={order} />
      </div>

      {/* <div className={classes.map}>
        <Title title="Your Location" fontSize="1.6rem" />
        <Map readonly={true} location={order.addressLatLng} />
      </div> */}

      <div className={classes.buttons_container}>
```

```
        <div className="d-grid">
                <Button
                  type="button"
                  onClick={placeOrderHandler}
                >
                  Place Order
                </Button>
              </div>
        </div>
      </div>
    </>
  );
}
module.exports = mongoose.model("Email", emailSchema);
```

Database Configuration:

```
import { connect, set } from 'mongoose';
import { UserModel } from '../models/user.model.js';
import { FoodModel } from '../models/food.model.js';
import { sample_users } from '../data.js';
import { sample_foods } from '../data.js';
import bcrypt from 'bcryptjs';
const PASSWORD_HASH_SALT_ROUNDS = 10;
set('strictQuery', true);

export const dbconnect = async () => {
  try {
    connect(process.env.MONGO_URI, {
```

```
      useNewUrlParser: true,
      useUnifiedTopology: true,
    });
    await seedUsers();
    await seedFoods();
    console.log('connect successfully---');
  } catch (error) {
    console.log(error);
  }
};

async function seedUsers() {
  const usersCount = await UserModel.countDocuments();
  if (usersCount > 0) {
    console.log('Users seed is already done!');
    return;
  }

  for (let user of sample_users) {
    user.password = await bcrypt.hash(user.password,
PASSWORD_HASH_SALT_ROUNDS);
    await UserModel.create(user);
  }

  console.log('Users seed is done!');
}

async function seedFoods() {
```

```
  const foods = await FoodModel.countDocuments();
  if (foods > 0) {
    console.log('Foods seed is already done!');
    return;
  }

  for (const food of sample_foods) {
    food.imageUrl = `/foods/${food.imageUrl}`;
    await FoodModel.create(food);
  }

  console.log('Foods seed Is Done!');
}
```

Mail configuration:

```
import FormData from 'form-data';
import Mailgun from 'mailgun.js';

export function getClient() {
  const mailgun = new Mailgun(FormData);
  const client = mailgun.client({
    username: 'api',
    key: process.env.MAILGUN_API_KEY,
  });

  return client;
}
```

Main Server code:

```
import dotenv from 'dotenv';
dotenv.config();
import { fileURLToPath } from 'url';
import express from 'express';
import cors from 'cors';
import foodRouter from './routers/food.router.js';
import userRouter from './routers/user.router.js';
import orderRouter from './routers/order.router.js';
import uploadRouter from './routers/upload.router.js';
import { dbconnect } from './config/database.config.js';
import path, { dirname } from 'path';
dbconnect();

const __filename = fileURLToPath(import.meta.url);
const __dirname = dirname(__filename);

const app = express();
app.use(express.json());
app.use(
  cors({
    credentials: true,
    origin: ['https://textileapp-2.onrender.com/'],
  })
);

app.use('/api/foods', foodRouter);
app.use('/api/users', userRouter);
```

```
app.use('/api/orders', orderRouter);
app.use('/api/upload', uploadRouter);

// Serve static files from the client build directory
app.use(express.static(path.join(__dirname, 'client', 'build')));

// Route all other requests to the client's index.html
app.get('*', function(req, res) {
  res.sendFile(path.join(__dirname, 'client', 'build', 'index.html'));
});

const PORT = process.env.PORT || 5000;
app.listen(PORT, () => {
  console.log('listening on port ' + PORT);
  console.log('JWT_SECRET:', process.env.JWT_SECRET);
});
```

APPENDIX 2

SNAPSHOTS

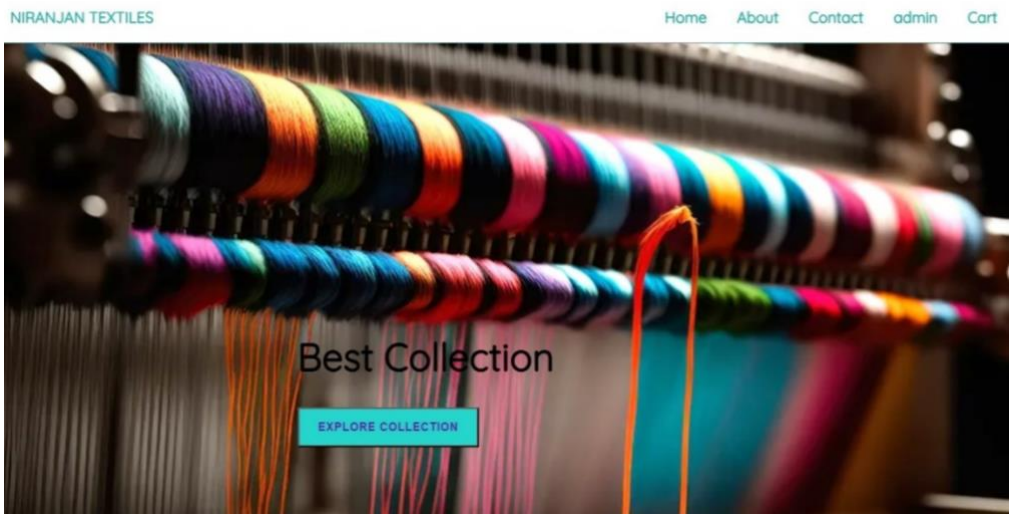

FIGURE A2.1 HOME PAGE

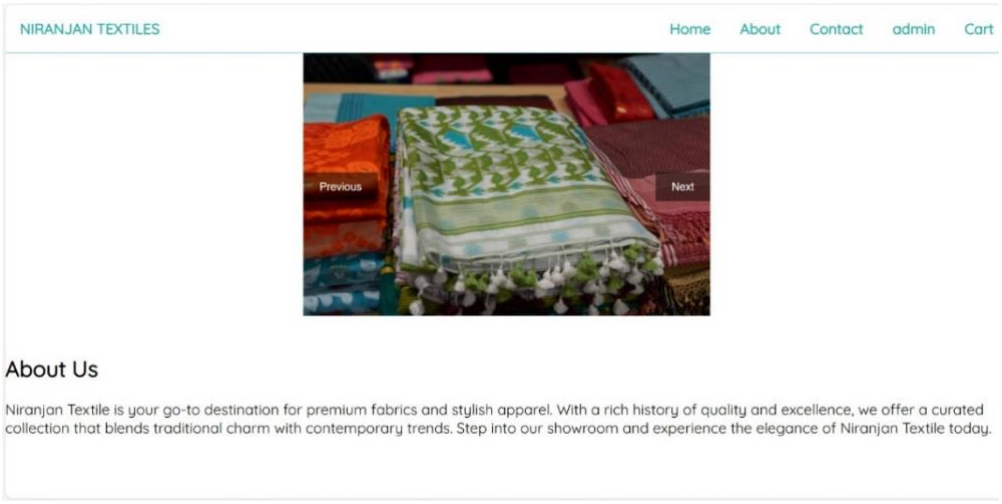

FIGURE A2.2 ABOUT US PAGE

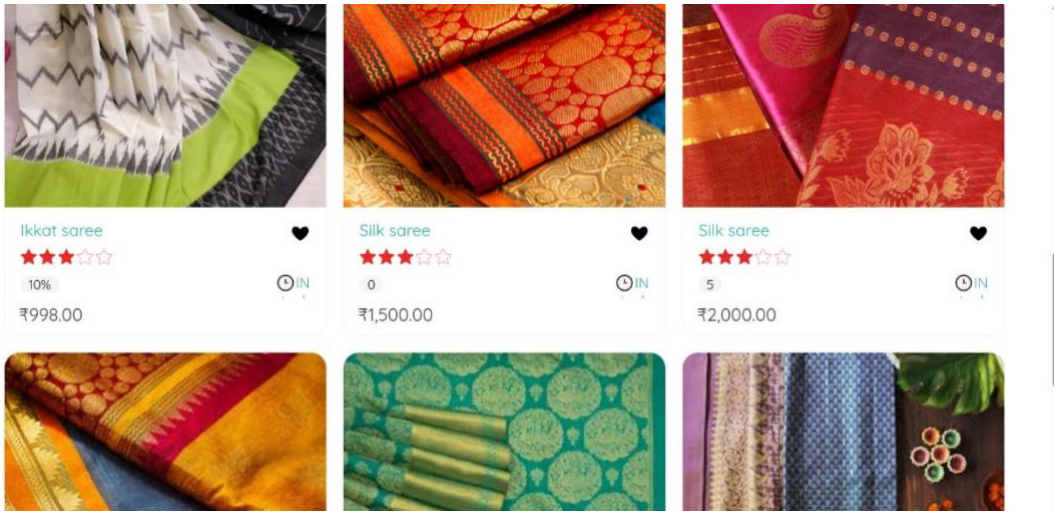

FIGURE A2.3 PRODUCTS PAGE

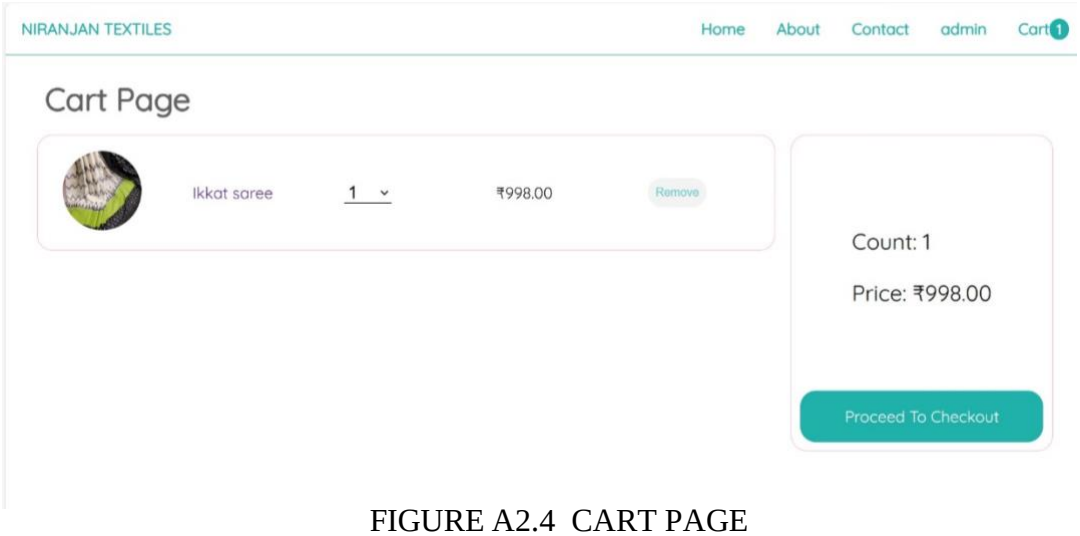

FIGURE A2.4 CART PAGE

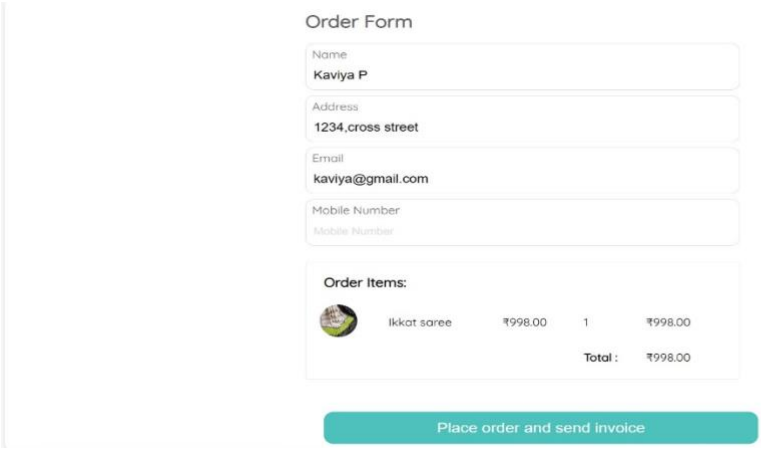

FIGURE A2.5 PLACE ORDER PAGE

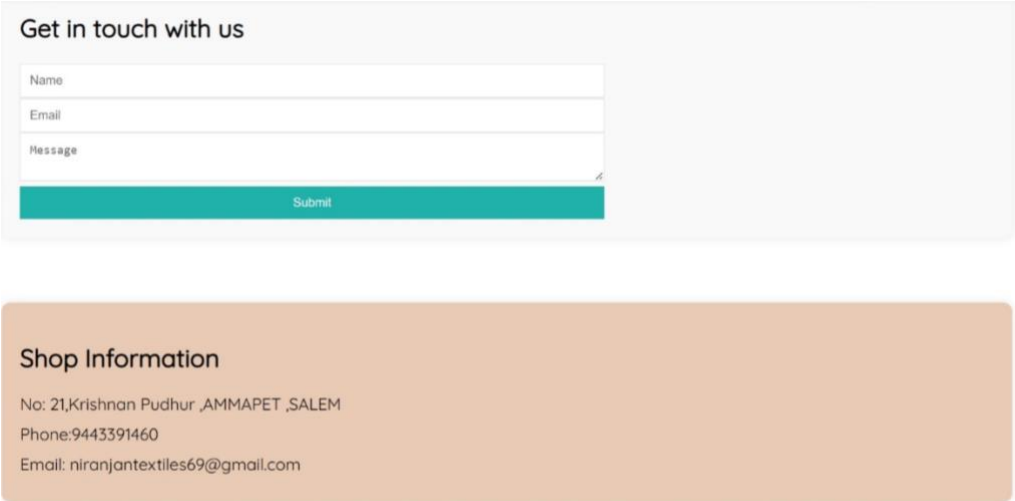

FIGURE A2.6 CONTACT US PAGE

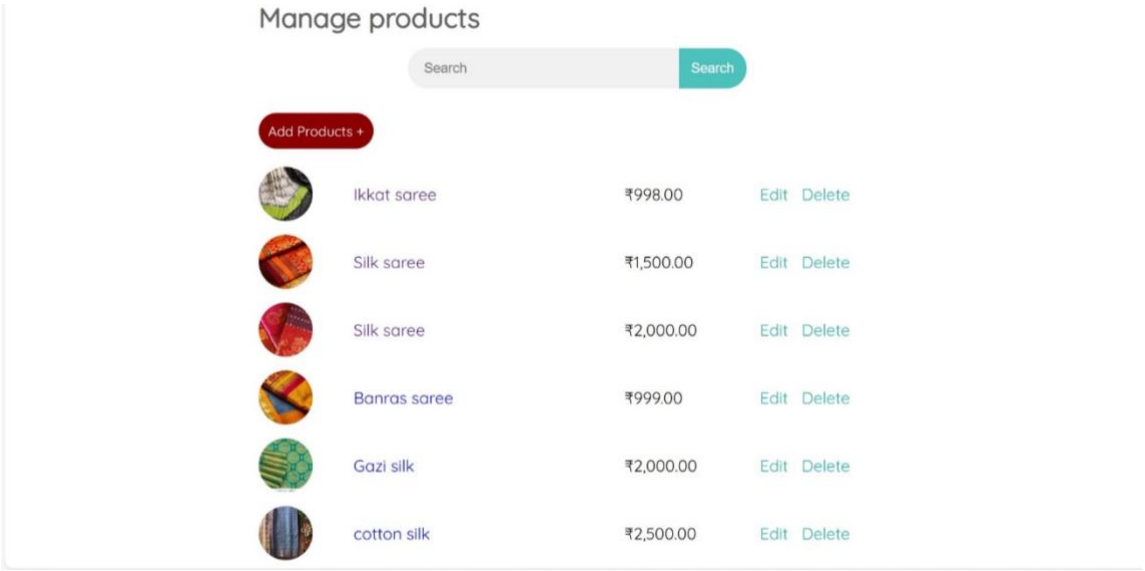

FIGURE A2.7 MANAGE PRODUCTS PAGE

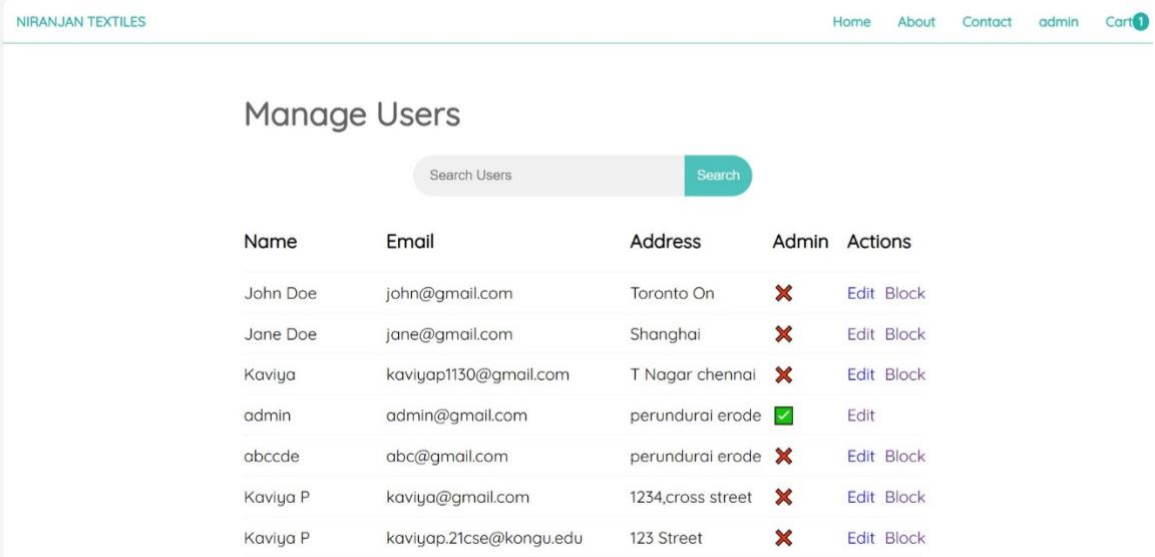

FIGURE A2.8 MANAGE USERS PAGE

REFERENCES

[1]	https://reactjs.org/docs/getting-started.html
[2]	https://docs.mongodb.com/manual/
[3]	https://expressjs.com/en/starter/installing.html
[4]	https://nodejs.org/en/docs/