An Acute Eyewear

A PROJECT REPORT

Submitted By

Akshep Yadav
(University Roll No: 2100290140017)
Sudhanshu Mishra
(University Roll No: 2100290140138)

Submitted in partial fulfilment of the
Requirements for the Degree of

MASTER OF COMPUTER APPLICATION

Under the Supervision of
Dr. Amit Kumar Gupta
(Associate Professor)

Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206
(JUNE 2023)

CERTIFICATE

Certified that **Akshep Yadav (2100290140017), Sudhanshu Mishra (2100290140138)** has carried out the project work having "**An Acute Eyewear**" for Master of Computer Applications from Dr. A.P. J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Technical University, Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Date:

Akshep Yadav (2100290140017)
Sudhanshu Mishra (2100290140138)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

Dr. Amit Kumar Gupta
(Associate Professor)
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Signature of Internal Examiner **Signature of External Examiner**

Dr. Arun Tripathi
Head, Department of Computer Applications
KIET Group of Institutions, Ghaziabad

ABSTRACT

Welcome to our An Acute Eyewear Website, where we offer a seamless and convenient shopping experience for all your eyewear needs. We understand that finding the perfect pair of glasses or sunglasses can be a challenging task, so we have curated a vast collection of trendy and high-quality eyewear to cater to every style and preference.

Our website is designed to provide a user-friendly interface, ensuring easy navigation and efficient browsing. With just a few clicks, you can explore our extensive range of eyeglasses, sunglasses, and contact lenses, featuring the latest designs from renowned brands. Whether you're looking for a sophisticated and professional look or a bold and fashionable statement, we have something to suit every taste.

We prioritize your visual health and offer prescription lenses tailored to your unique needs. Our knowledgeable opticians are available to assist you in selecting the right lenses, guiding you through the process and ensuring optimal vision correction.

At our online optical shop, customer satisfaction is our utmost priority. We strive to provide excellent service and a seamless purchasing experience. Our secure payment gateway guarantees the safety of your personal information, and we offer flexible and hassle-free return and exchange policies.

Embrace the convenience of online shopping with our quick and reliable delivery service. We ship worldwide, ensuring that your desired eyewear reaches you in a timely manner, no matter where you are.

Experience the joy of finding your perfect pair of glasses or sunglasses from the comfort of your own home. Shop with confidence at our online optical shop and enhance your visual style today.

ACKNOWLEDGEMENTS

Success in life is never attained single handedly. My deepest gratitude goes to my thesis supervisor, **Dr. Amit Kumar Gupta (Associate Professor)** for his guidance, help and encouragement throughout my research work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to **Dr. Arun Tripathi**, **Professor and Head, Department of Computer Applications**, for his insightful comments and administrative help at various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Akshep Yadav (2100290140017)
Sudhanshu Mishra (2100290140138)

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Welcome to our An Acute Eyewear, where convenience meets style! We are thrilled to provide you with a seamless and enjoyable shopping experience for all your eyewear needs. Whether you're looking for prescription glasses, sunglasses, or contact lenses, our website is your one-stop destination for fashionable eyewear that enhances your vision and complements your unique style.

At our online optical store, we understand the importance of both functionality and fashion when it comes to eyewear. That's why we have curated a wide range of high-quality products from renowned brands, ensuring that you have access to the latest trends and timeless classics. With our extensive collection, you'll find the perfect pair of glasses or sunglasses that not only correct your vision but also express your personal style.

Shopping with us is easy and convenient. Our user-friendly website allows you to browse through various categories, filter options, and compare products, making it simple to find exactly what you're looking for. Additionally, our detailed product descriptions and images provide you with all the necessary information to make an informed decision.

When it comes to affordability, we believe that everyone deserves access to quality eyewear without breaking the bank. That's why we offer competitive prices and regular discounts, making our online optical store an attractive option for budget-conscious shoppers. We also provide secure payment options, ensuring that your transactions are protected.

With our fast and reliable shipping services, your chosen eyewear will be delivered right to your doorstep, saving you time and effort. We understand that waiting for your new glasses can be exciting, so we make it a priority to dispatch orders promptly, ensuring a swift delivery.

Thank you for choosing our online optical store. We look forward to serving you and helping you find the perfect eyewear that not only enhances your vision but also reflects your individuality. Start browsing our extensive collection now and embark on a journey to find your ideal pair of glasses or sunglasses!

1.1 PROJECT DESCRIPTION

The online optical store website is a comprehensive platform that aims to provide customers with a seamless and convenient way to browse, purchase, and receive eyewear products. The website serves as a virtual store where customers can explore a wide range of eyeglasses, sunglasses, contact lenses, and other vision-related products. It aims to deliver a user-friendly and visually appealing interface to enhance the overall shopping experience.

1.2 PROJECT SCOPE

The scope of the An Acute Eyewear project includes:

- Objective: The objective of an Acute Eyewear website is to provide customers with a convenient and efficient platform to purchase eyewear and related products. It aims to offer a wide range of high-quality eyeglasses, sunglasses, contact lenses, and eye care accessories to cater to different needs and preferences. The website strives to provide a seamless user experience, allowing customers to easily browse, select, and customize their eyewear choices. It aims to ensure customer satisfaction by offering accurate product descriptions, clear images, and reliable customer reviews. Additionally, the online optical store website may provide virtual try-on features, expert guidance, and responsive customer support to enhance the overall shopping experience.

- Features: An Acute Eyewear website offers a range of features to provide a convenient and efficient shopping experience for customers. These features include a user-friendly interface with easy navigation, allowing customers to browse and search for various eyewear products. The website should provide detailed product descriptions, including specifications and sizing information, along with high-quality product images to aid in decision-making. Additionally, it should offer a virtual try-on feature that allows customers to virtually try on glasses using their own photos or through augmented reality technology. Secure online payment options, reliable shipping, and a hassle-free return policy are also essential features. Finally, personalized recommendations based on customer preferences can enhance the overall shopping experience.

- Target audience: The target audience of an Acute Eyewear website includes individuals seeking to purchase prescription eyeglasses, sunglasses, and contact lenses conveniently from home. They may be tech-savvy, value convenience and variety, and prioritize affordability without compromising on quality. The website caters to customers of diverse ages, styles, and vision correction needs

- Project constraints: The project constraints of an online optical store website may include tight timelines for development and deployment, budget limitations for design and marketing, integration challenges with existing systems, ensuring compliance with data protection regulations, and the need to provide a seamless user experience across different devices and platforms.

In conclusion, an Acute Eyewear website caters to a diverse target audience seeking convenient access to prescription eyeglasses, sunglasses, and contact lenses. While facing project constraints like tight timelines, budget limitations, and technical integration challenges, it aims to provide a seamless user experience and comply with data protection regulations.

1.3 HARDWARE / SOFTWARE USED IN PROJECT

Hardware Requirement

- System: Pentium i5 Processor
- Hard Disk :500 GB minimum
- Monitor: 15’’ LED
- Input Devices: Keyboard, Mouse
- Ram: 2 GB

Software Requirement

- Windows 7/8/10 (32- bit or 64-bit)
- 8GB RAM Minimum Processori3 or i5

Technology Used

- HTML, CSS, Bootstrap, Java Script, ReactJS, Node JS
- MongoDB

CHAPTER 2

FEASIBILITY STUDY

2.1 TECHNICAL FEASIBILITY

The technical feasibility of an online optical store website involves evaluating the capability and viability of implementing the necessary technology infrastructure. Factors to consider include:

- Website Development: Assessing the expertise and resources required to develop a user-friendly and visually appealing website with features like product catalogs, shopping carts, and secure payment gateways.
- Scalability: Determining whether the website architecture can handle increasing traffic, product inventory, and customer data as the business grows.
- Integration: Ensuring seamless integration with existing systems such as inventory management, order processing, and customer relationship management (CRM) software.
- Security: Implementing robust security measures to protect customer information, payment transactions, and sensitive data from cyber threats and breaches.
- Mobile Responsiveness: Designing a mobile-friendly website that provides a consistent and optimized experience across different devices and screen sizes.
- Performance and Speed: Optimizing website loading times and performance to ensure quick and efficient browsing, product search, and checkout processes.
- Hosting and Infrastructure: Evaluating hosting options and infrastructure requirements to support the website's expected traffic, bandwidth, and storage needs.

By assessing these technical factors, an online optical store can determine the feasibility of implementing the required technology infrastructure to support its operations and provide a seamless online shopping experience to customers.

2.2 OPERATIONAL FEASIBILITY

The operational feasibility of an online optical store website examines whether the business operations can be effectively supported by the website. Key considerations include:

- Product Management: Assessing the ability to efficiently manage and update product inventory, including adding new products, updating prices, and removing discontinued items.

- Order Processing: Ensuring that the website can handle and process orders accurately and promptly, including generating order confirmations, tracking shipments, and handling returns or exchanges.

- Customer Support: Evaluating the mechanisms in place to provide timely and effective customer support, such as live chat, email support, or a dedicated customer service helpline.

- Logistics and Fulfillment: Determining the feasibility of coordinating with shipping partners and logistics providers to ensure timely delivery of orders, tracking shipments, and managing inventory replenishment.

- Payment Processing: Assessing the integration of secure payment gateways to facilitate smooth and secure online transactions, including accepting various payment methods and ensuring compliance with PCI-DSS standards.

- Returns and Refunds: Establishing protocols and systems to handle returns, refunds, and exchanges efficiently, including clear policies, automated processes, and seamless communication with customers.

- Website Maintenance: Evaluating the ability to perform regular maintenance and updates to ensure optimal website performance, security, and compatibility with evolving technologies.

By considering these operational aspects, an online optical store can determine the feasibility of effectively managing day-to-day operations, delivering quality customer service, and maintaining a seamless online shopping experience through the website.

2.3 BEHAVIOURAL FEASIBILITY

Behavioral feasibility assesses the acceptance and adoption of an online optical store website by its target audience and key stakeholders. It focuses on understanding how users will interact with the website and whether they will embrace the new platform. Key considerations for behavioral feasibility include:

- User Experience: Ensuring that the website provides a user-friendly and intuitive interface, easy navigation, and clear product information. Conducting user testing and gathering feedback to refine the website's design and functionality.

- User Acceptance: Assessing the willingness of customers to embrace online shopping for optical products and determining if the website meets their needs and expectations. Conducting surveys or interviews to understand customer preferences and concerns.

- Customer Education: Evaluating the feasibility of providing educational resources and guides to help customers understand product options, lens prescriptions, and how to measure their PD (pupillary distance) accurately.

- Trust and Security: Building trust with customers by implementing robust security measures, displaying privacy policies, and offering secure payment options. Communicating transparently about data protection and handling customer concerns regarding online transactions.

- Digital Literacy: Assessing the digital skills and comfort level of the target audience to ensure that the website is accessible and usable for users with varying levels of technical expertise.

- Feedback and Reviews: Incorporating mechanisms for customers to leave feedback, reviews, and ratings, allowing potential customers to make informed decisions and improving the overall user experience.

- Stakeholder Engagement: Considering the support and engagement of key stakeholders, such as optometrists, eyewear manufacturers, and industry partners, to ensure their willingness to collaborate and promote the online optical store.

By addressing these behavioral factors, an online optical store can increase the likelihood of user acceptance and adoption, fostering positive customer experiences, and driving the success of the website.

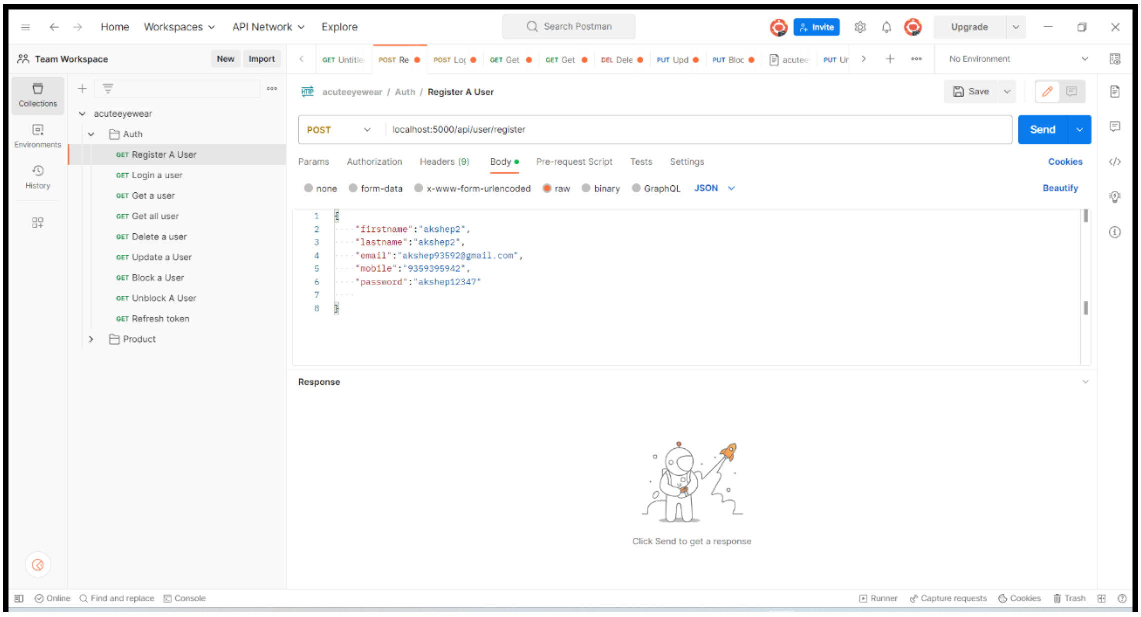
CHAPTER 3

DATABASE DESIGN

MongoDB is a NoSQL document-oriented database that stores data in flexible, JSON-like documents. It provides high scalability, automatic shading, and replication for managing large amounts of data across distributed systems. MongoDB supports dynamic schema, making it suitable for handling diverse data types and agile development environments.

It supports dynamic schema, allowing easy storage of diverse and evolving data types. MongoDB is widely used in various applications, including web and mobile applications, content management systems, real-time analytics, and IoT platforms, providing developers with a powerful and flexible database solution.

3.1 DATABASE TABLE



POST localhost:5000/api/user/register

Body: ○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
    "firstname":"akshep2",
    "lastname":"akshep2",
    "email":"akshep93592@gmail.com",
    "mobile":"9359395942",
    "password":"akshep12347"
    
}
```

Figure 3.1: Registration page

3.2 FLOW CHART

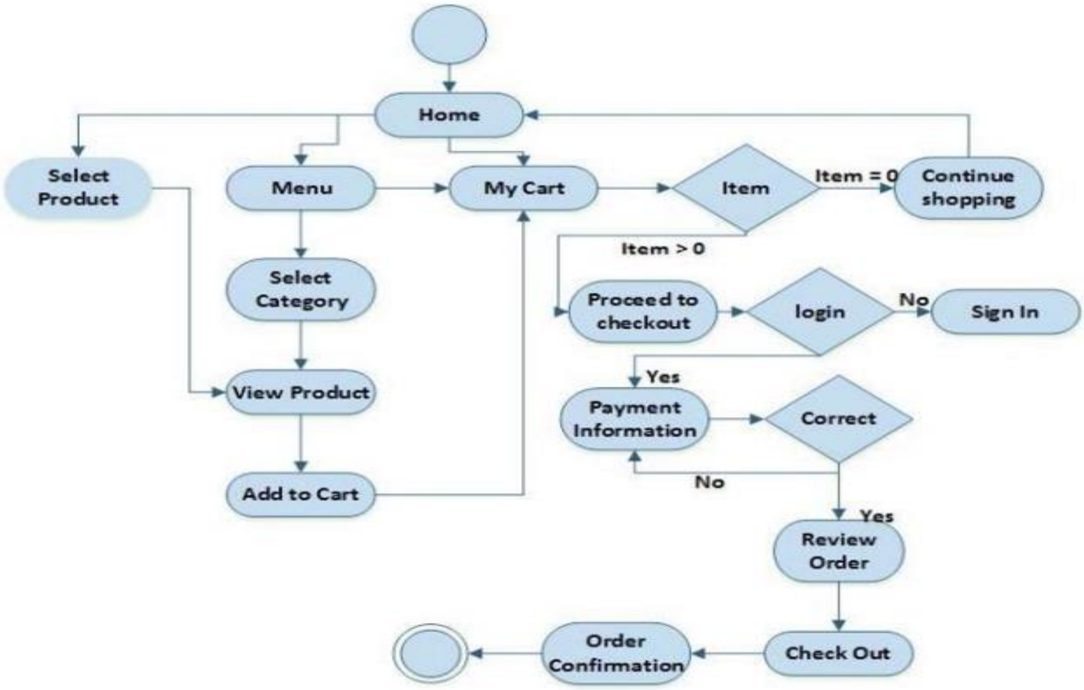


Figure 3.2: Flow Chart

3.3 USE CASE DIAGRAM

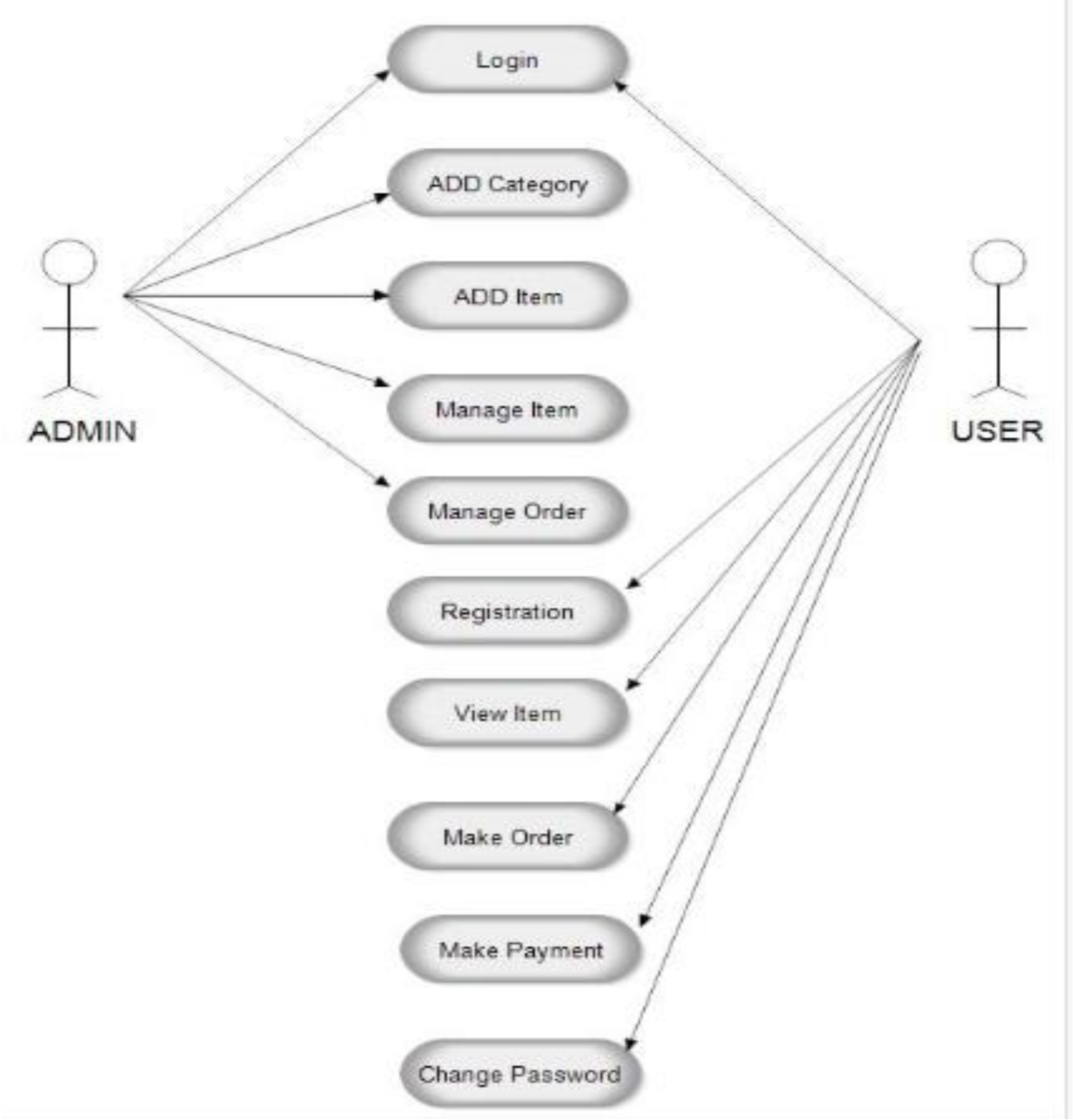


Figure 3.3 Use Case Diagram

3.4 SEQUENCE DIAGRAM

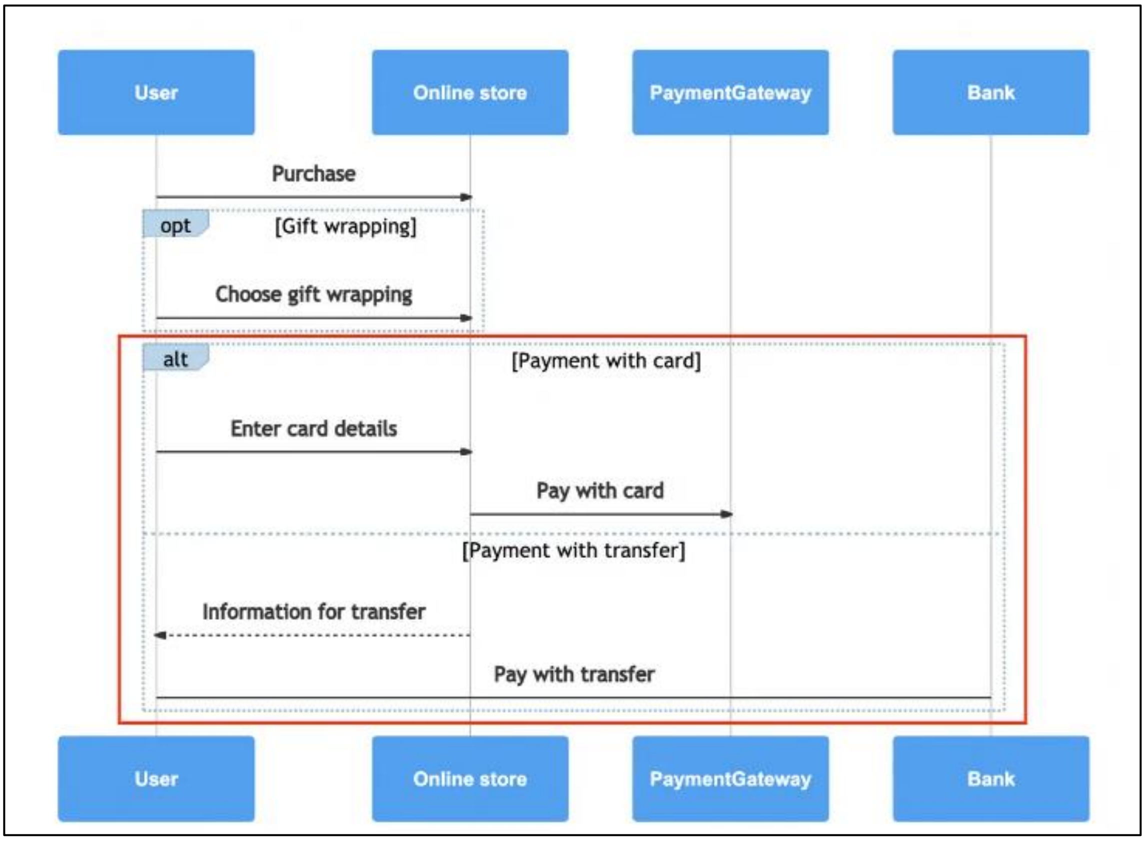


Figure 3.4 Sequence Diagram

CHAPTER 4

FORM DESIGN

4.1 HOME PAGE

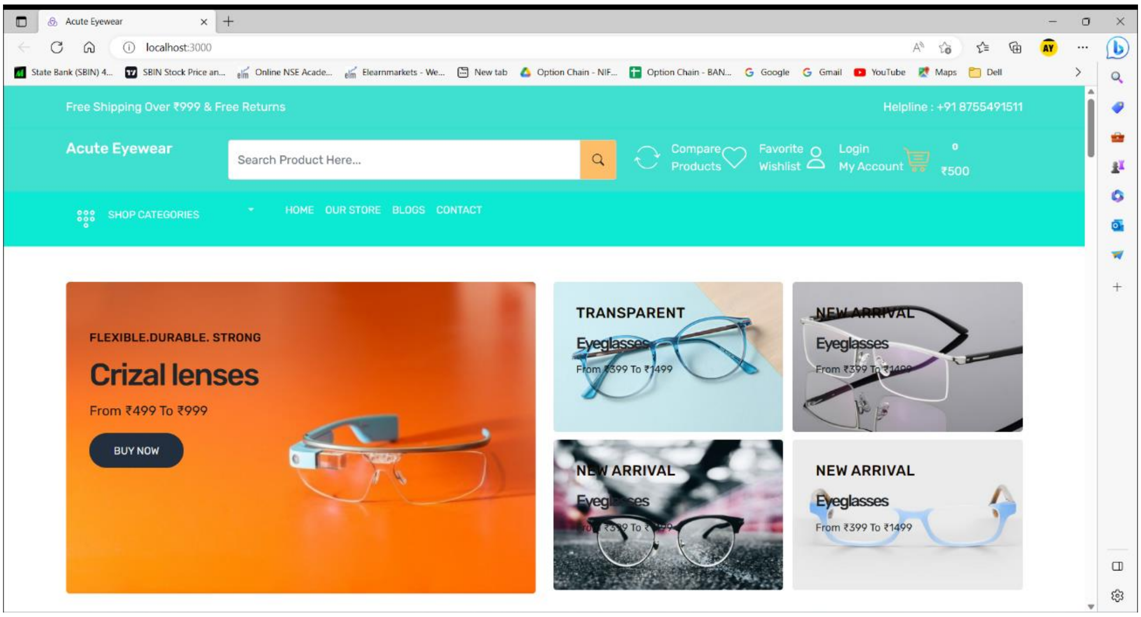


Figure 4.1 Home Page

4.2 OUR STORE

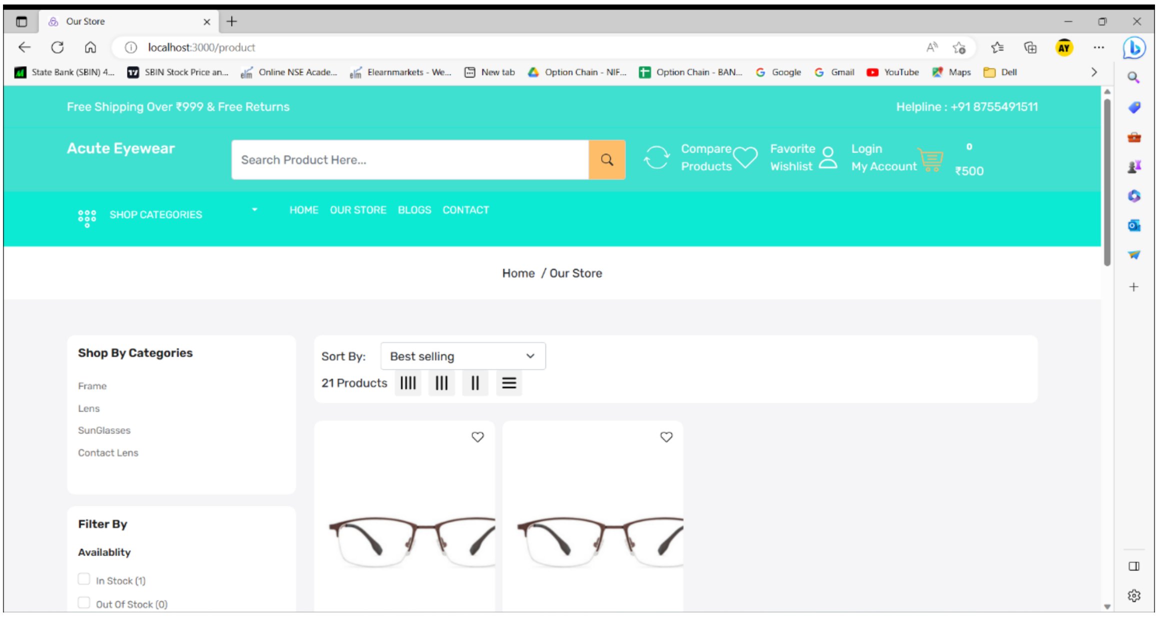
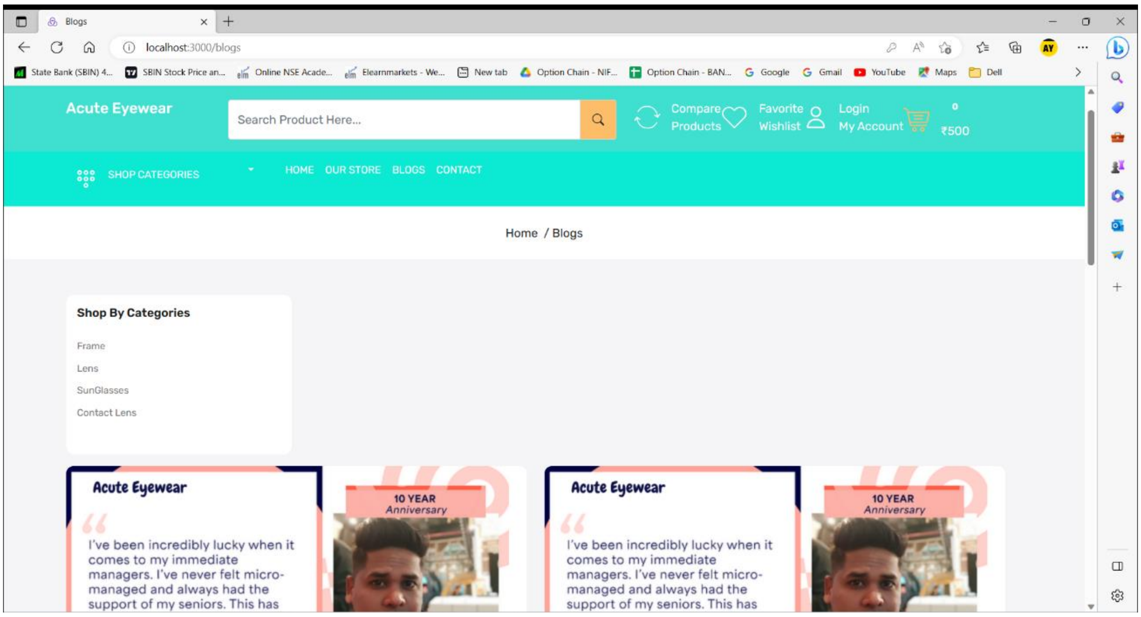


Figure 4.1 Our Store

4.3 BLOGS



I've been incredibly lucky when it comes to my immediate managers. I've never felt micro-managed and always had the support of my seniors. This has

I've been incredibly lucky when it comes to my immediate managers. I've never felt micro-managed and always had the support of my seniors. This has

Figure 4.3 Blogs

4.4 CONTACT US

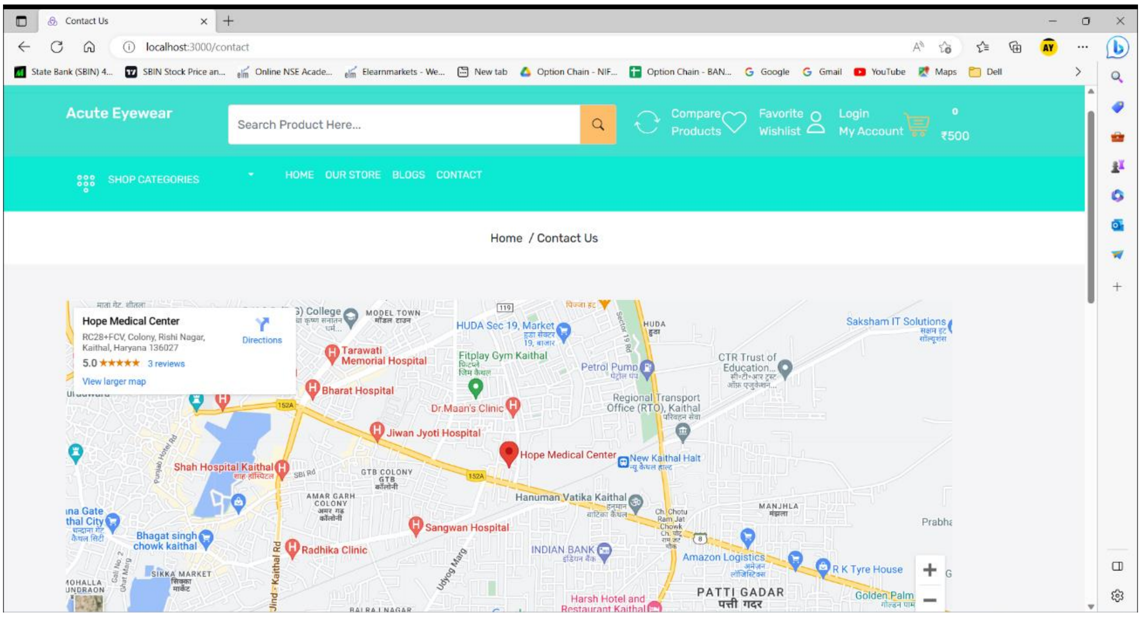
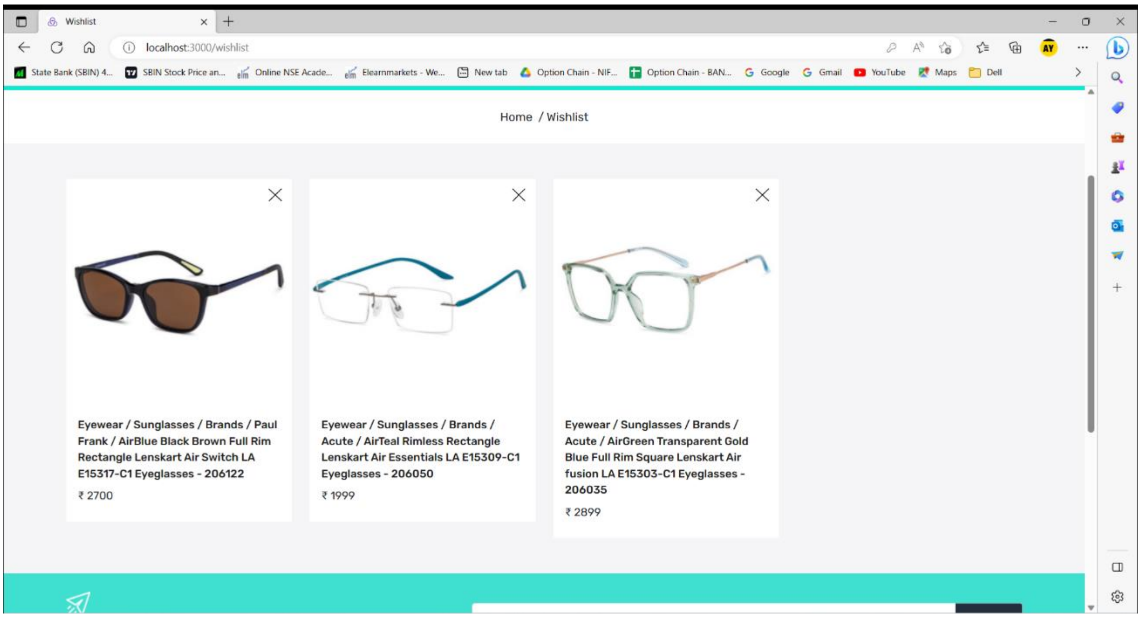


Figure 4.4 Contact Us

4.5 WISHLIST



Eyewear / Sunglasses / Brands / Paul Frank / AirBlue Black Brown Full Rim Rectangle Lenskart Air Switch LA E15317-C1 Eyeglasses - 206122

₹ 2700

Eyewear / Sunglasses / Brands / Acute / AirTeal Rimless Rectangle Lenskart Air Essentials LA E15309-C1 Eyeglasses - 206050

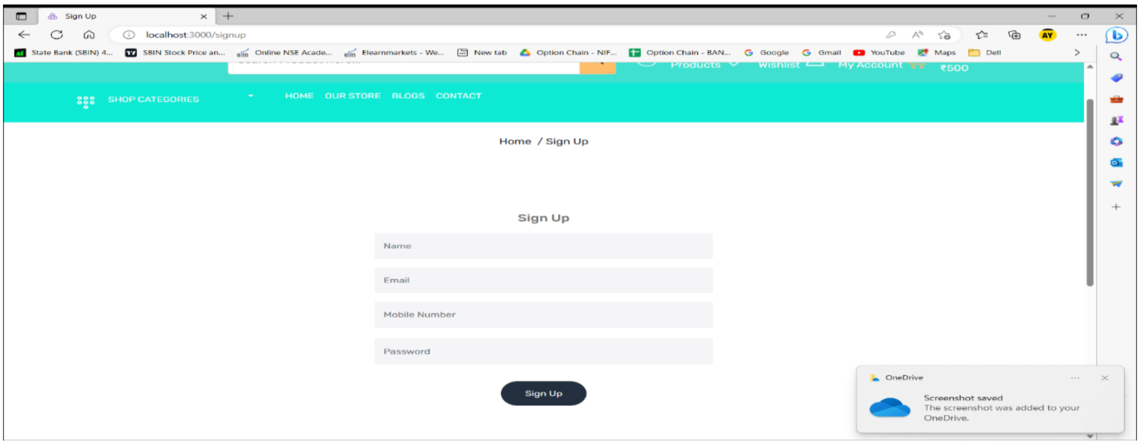
₹ 1999

Eyewear / Sunglasses / Brands / Acute / AirGreen Transparent Gold Blue Full Rim Square Lenskart Air fusion LA E15303-C1 Eyeglasses - 206035

₹ 2899

Figure 4.5 Wishlist

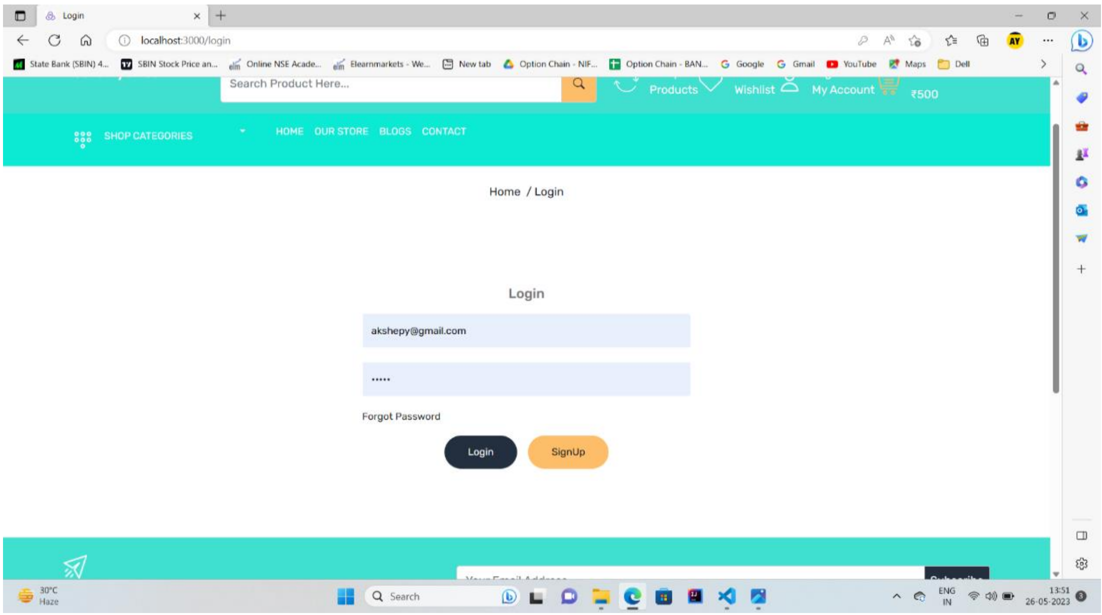
4.6 SIGN UP



Screenshot saved
The screenshot was added to your OneDrive.

Figure 4.6 Sign Up

4.7 Login

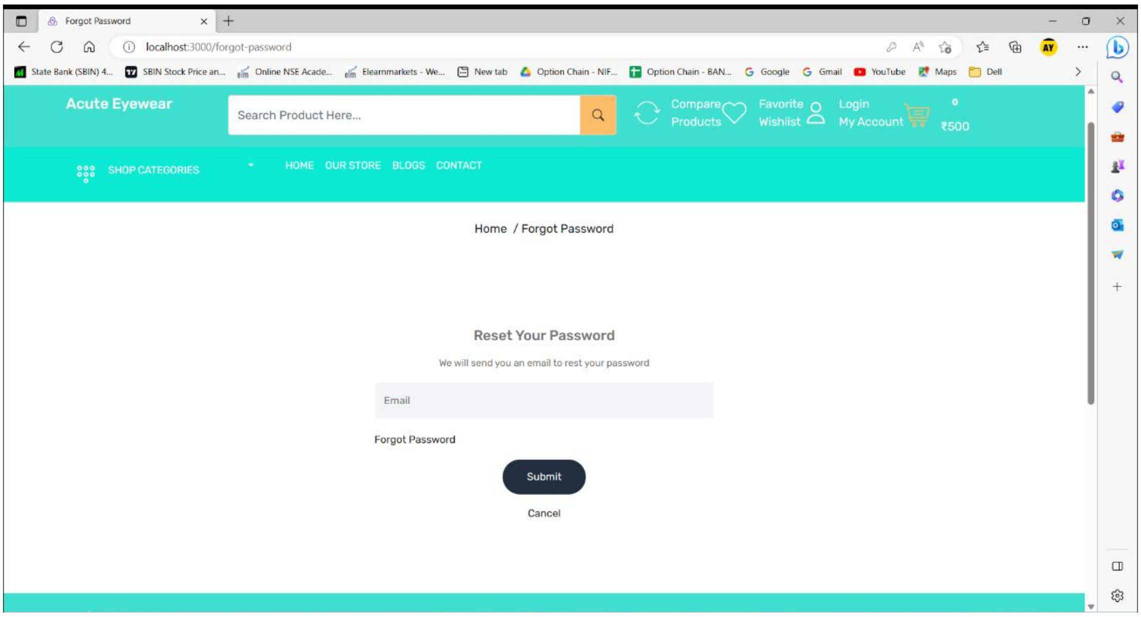


akshepy@gmail.com

•••••

Figure 4.7 Login

4.8 Forgot Password



Reset Your Password

We will send you an email to rest your password

Figure 4.8 Forgot Password

CHAPTER 5

CODING

5.1 HOME

```
import React from "react"
import {Link} from "react-router-dom";
import Marquee from "react-fast-marquee";
import BlogCard from "../components/BlogCard";
import ProductCard from "../components/ProductCard";
import SpecialProduct from "../components/SpecialProduct";
const Home = () => {
  return (
   <>
  <section className="home-wrapper-1 py-5">
    <div className="container-xxl">
      <div className="row">
        <div className="col-6">
          <div className="main-banner position-relative ">
            <img
            src="images/main-banner-1.jpg"
            className="img-fluid rounded-3"
            alt="main banner"
            />
            <div className="main-banner-content position-absolute">
              <h4>Flexible.Durable.  Strong</h4>
              <h5>Crizal lenses</h5>
              <p>From ₹499 To ₹999</p>
              <Link  className="button"> BUY NOW</Link>
            </div>
          </div>
        </div>
        <div className="col-6">
          <div className="d-flex flex-wrap gap-10 justify-content-between align-items-
center">
          <div className="small-banner position-relative ">
            <img
            src="images/catbanner-01.png"
            className="img-fluid rounded-3"
            alt="main banner"
            />
            <div className="small-banner-content position-absolute">
              <h4>Transparent</h4>
              <h5>Eyeglasses</h5>
              <p>From ₹399 To ₹1499</p>

            </div>
          </div>
          <div className="small-banner position-relative ">
            <img
            src="images/catbanner-02.png"
            className="img-fluid rounded-3"
            alt="main banner"
            />
            <div className="small-banner-content position-absolute">
              <h4>NEW ARRIVAL</h4>
              <h5>Eyeglasses</h5>
              <p>From ₹399 To ₹1499</p>
              </div>
          </div>
          <div className="small-banner position-relative ">
            <img
            src="images/catbanner-03.png"
            className="img-fluid rounded-3"
            alt="main banner"
            />
            <div className="small-banner-content position-absolute">
              <h4>NEW ARRIVAL</h4>
              <h5>Eyeglasses</h5>
              <p>From ₹399 To ₹1499</p>
              </div>
          </div>
          <div className="small-banner position-relative ">
            <img
            src="images/catbanner-04.png"
            className="img-fluid rounded-3"
            alt="main banner"
            />
            <div className="small-banner-content position-absolute">
```

```
            <h4>NEW ARRIVAL</h4>
            <h5>Eyeglasses</h5>
            <p>From ₹399 To ₹1499</p>
            </div>
        </div>
        </div>
      </div>
    </div>
  </div>
</section>
<section className="home-wrapper-2 py-5">
  <div className="container-xxl">
    <div className="row">
      <div className="col-12">
        <div className="service d-flex align-items-center justify-content-between">
          <div className="d-flex align-items-center gap-15">
            <img src="images/service.png" alt="services" />
            <div>
              <h6>Free Shipping</h6>
              <p className="mb-0">From all orders over ₹999
              </p>
            </div>
          </div>
          <div className="d-flex align-items-center gap-15">
            <img src="images/service-02.png" alt="services" />
            <div>
              <h6>Daily Suprise Offer</h6>
              <p className="mb-0">Save up to 25% off
              </p>
            </div>
          </div>
          <div className="d-flex align-items-center gap-15">
            <img src="images/service-03.png" alt="services" />
            <div>
              <h6>Support 24/7</h6>
              <p className="mb-0">Shop with an expert
              </p>
            </div>
          </div>
          <div className="d-flex align-items-center gap-15">
            <img src="images/service-04.png" alt="services" />
            <div>
              <h6>Affordable Prices</h6>
```

```
                <p className="mb-0">Get Factory direct price
                </p>
              </div>
            </div>
            <div className="d-flex align-items-center gap-15">
              <img src="images/service-05.png" alt="services" />
              <div>
                <h6>Secure Payments</h6>
                <p>100% Protected Payments
                </p>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </section>
  <section className="home-wrapper-2 py-5">
    <div className="container-xxl">
      <div className="row">
        <div className="col-12">
          <div className="categories d-flex justify-content-between flex-wrap align-items-center">
            <div className="d-flex gap align-items-center">
              <div>
                <h6>Eyeglasses</h6>
                <p>10</p>
              </div>
              <img src="images/camera.png" alt="camera" />
            </div>
            <div className="d-flex gap align-items-center">
              <div>
                <h6>Sunglasses</h6>
                <p>10</p>
              </div>
              <img src="images/camera.png" alt="camera" />
            </div>
            <div className="d-flex gap align-items-center">
              <div>
                <h6>Computer <br /> Glasses</h6>
                <p>10</p>
              </div>
              <img src="images/camera.png" alt="camera" />
            </div>
            <div className="d-flex gap align-items-center">
              <div>
                <h6>Eyeglasses</h6>
                <p>10</p>
              </div>
              <img src="images/camera.png" alt="camera" />
            </div>
            <div className="d-flex gap align-items-center">
              <div>
                <h6>Sunglasses</h6>
                <p>10</p>
              </div>
              <img src="images/camera.png" alt="camera" />
            </div>
            <div className="d-flex gap align-items-center">
              <div>
                <h6>Computer <br /> Glasses</h6>
                <p>10</p>
              </div>
              <img src="images/camera.png" alt="camera" />
            </div>
            <div className="d-flex gap align-items-center">
              <div>
                <h6>Contacts <br /> Lenses</h6>
                <p>10</p>
              </div>
              <img src="images/camera.png" alt="camera" />
            </div>
            <div className="d-flex gap align-items-center">
              <div>
                <h6>Power <br /> Sunglasses</h6>
                <p>10</p>
              </div>
              <img src="images/camera.png" alt="camera" />
            </div>
          </div>

        </div>
      </div>
    </div>
  </section>
  <section className="featured-wrapper py-5 home-wrapper">
```

```
  <div className="container-xxl">
    <div className="row">
      <div className="col-12">
        <h3 className="section-heading">Featured Collection</h3>
      </div>
      <ProductCard />
      <ProductCard />
      <ProductCard />
      <ProductCard />
    </div>
  </div>
</section>
<section className="famous-wrapper py-5 home-wrapper-2">
       <div className="container-xxl">
        <div className="row">
          <div className="col-3">
            <div className="famous-card position-relative">
              <img src="images/famous1.png"
              className="img-fluid"
               alt="famous" />
              <div className="famous-content position-absolute">
                <h5>Big Screen</h5>
                <h6>Smart Watch Series 7</h6>
                <p>From 500 for 24 month</p>
              </div>
            </div>
          </div>
          <div className="col-3">
            <div className="famous-card position-relative">
              <img src="images/famous1.png"
              className="img-fluid"
               alt="famous" />
              <div className="famous-content position-absolute">
                <h5>Big Screen</h5>
                <h6>Smart Watch Series 7</h6>
                <p>From 500 for 24 month</p>
              </div>
            </div>
          </div>
          <div className="col-3">
            <div className="famous-card position-relative">
              <img src="images/famous1.png"
              className="img-fluid"
```

```
               alt="famous" />
              <div className="famous-content position-absolute">
                <h5>Big Screen</h5>
                <h6>Smart Watch Series 7</h6>
                <p>From 500 for 24 month</p>
              </div>
            </div>
          </div>
          <div className="col-3">
            <div className="famous-card position-relative">
              <img src="images/famous1.png"
              className="img-fluid"
               alt="famous" />
              <div className="famous-content position-absolute">
                <h5>Now available in the India</h5>
                <h6>Smart Watch Series 7</h6>
                <p>From  ₹999 to  ₹4999</p>
              </div>
            </div>
          </div>
        </div>
       </div>
</section>
<section className="special-wrapper py-5 home-wrapper-2">
  <div className="container-xxl">
    <div className="row">
    <div className="col-12">
        <h3 className="section-heading">Special Product</h3>
        </div>
    </div>
    <div className="row">
      <SpecialProduct />
      <SpecialProduct />
      <SpecialProduct />
    </div>
  </div>
</section>
<section className="popular-wrapper py-5 home-wrapper-2">
  <div className="container-xxl">
    <div className="row">
      <div className="col-12">
        <h3 className="section-heading">Our Popular Products</h3>
      </div>
```

```
       </div>
       <div className="row">
       <ProductCard />
         <ProductCard />
         <ProductCard />
         <ProductCard />
       </div>
     </div>
  </section>
  <section className="marquee-wrapper home-wrapper-2 py-5">
     <div className="container-xxl">
       <div className="row" >
       <div className="col-12">
           <div className="marquee-inner-wrapper card-wrapper ">
         <Marquee className="d-flex">
        <div className="mx-4 w-25">
         <img src="images/f1.png" alt="brand" />
        </div>
        <div className="mx-4 w-25">
         <img src="images/brand-02.png" alt="brand" />
        </div>
        <div className="mx-4 w-25">
         <img src="images/brand-03.png" alt="brand" />
        </div>
        <div className="mx-4 w-25">
         <img src="images/brand-04.png" alt="brand" />
        </div>
        <div className="mx-4 w-25">
         <img src="images/brand-05.png" alt="brand" />
        </div>
        <div className="mx-4 w-25">
         <img src="images/brand-06.png" alt="brand" />
        </div>
        <div className="mx-4 w-25">
         <img src="images/brand-07.png" alt="brand" />
        </div>
        <div className="mx-4 w-25">
         <img src="images/brand-08.png" alt="brand" />
        </div>
</Marquee>
         </div>
       </div>
     </div>
```

```
    </div>
  </section>

  <section className="blog-wrapper py-5 home-wrapper">
    <div className="container-xxl">
      <div className="row">
        <div className="col-12">
          <h3 className="section-heading">Our Latest Blogs</h3>
        </div>
        </div>
       <div className="row">
        <div className="col-3">
        <BlogCard />
       </div>
       <div className="col-3">
        <BlogCard />
       </div>
       <div className="col-3">
        <BlogCard />
       </div>
       <div className="col-3">
        <BlogCard />
       </div>
      </div>
    </div>
  </section>

  </>

  );
};

export default Home;
```

5.2 LOGIN

```
import React from 'react'
import BreadCrumb from "../components/BreadCrumb";
import Meta from "../components/Meta";
import { Link } from 'react-router-dom';

export const Login = () => {
  return (
  <>
  <Meta title={"Login"} />
   <BreadCrumb title="Login" />

   <div className="login-wrapper py-5 home wrapper-2">
       <div className="container-xxl">
       <div className="row">
           <div className="col-12">
               <div className="auth-card">
                   <h3 className="text-center mb-3">Login</h3>
                   <form action="" className="d-flex flex-column gap-15">
                       <div>
                           <input type="email" name="email" placeholder="Email" className="form-control" />
                       </div>
                       <div className="mt-1">
                           <input type="password" name="password" placeholder="Password" className="form-control" />
                       </div>
                       <div>
                           <Link to='/forgot-password'>Forgot Password</Link>
                           <div className="mt-3 d-flex justify-content-center gap-15 align-items-center">
                               <button className="button border-0" type="submit">
                                   Login
                                   </button>
                               <Link to="/signup"className="button signup">
                                   SignUp
                                   </Link>

                           </div>
                       </div>
                   </form>UU
               </div>
           </div>
       </div>
       </div>
   </div>
  </>
 )
 }
```

5.3 OUR STORE

```
import React, { useState } from "react";
import BreadCrumb from "../components/BreadCrumb";
import Meta from "../components/Meta";
import ReactStars from "react-rating-stars-component";
import ProductCard from "../components/ProductCard";
import Color from "../components/Color";

const OurStore = () => {
  const [grid, setGrid] = useState(4);


  return  (
  <>
    <Meta title={"Our Store"} />
    <BreadCrumb title="Our Store" />
    <div className="store-wrapper home-wrapper-2 py-5">
      <div className="container-xxl">
        <div className="row">
          <div className="col-3">
            <div className="filter-card mb-3">
              <h3 className="filter-title">
                Shop By Categories
              </h3>
              <ul className="ps-0">
                <li>Frame</li>
                <li>Lens</li>
                <li>SunGlasses</li>
                <li>Contact Lens</li>
              </ul>
            </div>
            <div className="filter-card mb-3">
              <h3 className="filter-title">
                Filter By
              </h3>
              <div>
                <h5 className="sub-title">Availablity</h5>
                <div>
                <div className="form-check">
                  <input
                  className="form-check-input"
                  type="checkbox"
                  value=""
```

```
             id=""/>
               <label className="form-check-label" htmlFor="">
                 In Stock (1)
               </label>
           </div>
           <div className="form-check">
           <input
           className="form-check-input"
           type="checkbox"
           value=""
           id=""

           />
               <label class="form-check-label" htmlFor="">
                 Out Of Stock (0)
               </label>
           </div>
           </div>
           <h5 className="sub-title">Price</h5>

           <div className="d-flex align-items-center gap-10">
           <div className="form-floating ">
          <input
          type="email"
          className="form-control "
          id="floatingInput"
          placeholder="From"
/>
<label htmlFor="floatingInput">From</label>
             </div>
           <div className="form-floating ">
          <input
          type="email"
          className="form-control "
          id="floatingInput1"
          placeholder="To"
           />
            <label htmlFor="floatingInput1">To</label>
              </div>
           </div>
           <h5 className="sub-title">Colors</h5>
           <div>
               <color/>
           </div>
```

```
    <h5 className="sub-title">Size</h5>
    <div>
    <div className="form-check">
      <input
      className="form-check-input"
      type="checkbox"
      value=""
      id="color-1"/>
        <label className="form-check-label" htmlFor="color-1">
          S (2)
        </label>
    </div>
    <div className="form-check">
      <input
      className="form-check-input"
      type="checkbox"
      value=""
      id="color-2"/>
        <label className="form-check-label" htmlFor="color-2">
          M (2)
        </label>
    </div>
    </div>
  </div>
</div>
<div className="filter-card mb-3">
  <h3 className="filter-title">  Product Tags </h3>
  <div>
    <div className="product-tags d-flex flex-wrap align-items-center gap-10">
      <span className="badge bg-light text-secondary rounded-3 py-2 px-3">
        Frame
      </span>
      <span className="badge bg-light text-secondary rounded-3 py-2 px-3">
       SunGlasses
      </span>
      <span className="badge bg-light text-secondary rounded-3 py-2 px-3">
        Lens
      </span>
      <span className="badge bg-light text-secondary rounded-3 py-2 px-3">
        contact Lens
      </span>
    </div>
  </div>
</div>
```

```
<div className="filter-card mb-3">
  <h3 className="filter-title">  Random Product   </h3>
  <div>
    <div className="random-products mb-3 d-flex ">
      <div className="w-50">
        <img
        src="images/f1/f1.png"
        className="img-fluid"
        alt="watche"
        />
      </div>
      <div className="w-50">
        <h5>
          fastrack Frame
        </h5>
        <ReactStars
    count={5}
     size={24}
     value={4}
     edit={false}
      activeColor="#ffd700"
      />
      <b>₹ 200</b>
      </div>
    </div>
    <div className="random-products d-flex">
      <div className="w-50">
        <img
        src="images/f1/f1.1.png"
        className="img-fluid"
        alt="watche"
        />
      </div>
      <div className="w-50">
        <h5>
          Paul Frank Frame
        </h5>
        <ReactStars
    count={5}
     size={24}
     value={4}
     edit={false}
      activeColor="#ffd700"
      />
```

```
        <b>₹ 200</b>
        </div>
      </div>
    </div>
  </div>
</div>
<div className="col-9">
  <div className="filter-sort-grid mb-4">
    <div className="d-flex justify-content-between align-items-center">
    <div className="d-flex align-items-center gap-10">
      <p className="mb-0 d-block" style={{width: "100px"}}>
        Sort By:</p>
      <select name="" className="form-control form-select" id="">
     <option value="manual">Featured</option>
          <option value="best-selling" selected="selected">
            Best selling
          </option>
          <option value="title-ascending">
            Alphabetically, A-Z
          </option>
          <option value="title-descending">
            Alphabetically, Z-A
          </option>
          <option value="price-ascending">
            Price, low to hingh
          </option>
          <option value="price-descending">
            Price, high to low
          </option>
          <option value="created-descending">
            Date, new to old
          </option>
          </select>
          </div>
          </div>
    <div>
      <div className="d-flex align-items-center gap-10">
        <p className="totalproducts mb-0">21 Products</p>
        <div className="d-flex gap-10 align-items-center grid ">
           <img onClick={() => {
            setGrid(3);
           }} src="images/gr4.svg"
           className="d-block img-fluid"
            alt=" grid"
```

```
                        />
                        <img onClick={() => {
                         setGrid(4);
                        }} src="images/gr3.svg"
                     className="d-block img-fluid"
                      alt=" grid"
                      />
                        <img onClick={() => {
                         setGrid(6);
                        }} src="images/gr2.svg"
                     className="d-block img-fluid"
                      alt=" grid"
                        />
                         <img onClick={() => {
                           setGrid(12);
                         }} src="images/gr.svg"
                     className="d-block img-fluid"
                      alt=" grid"
                      />
                  </div>
                </div>
              </div>
            </div>
            <div className="products-list pb-5">
              <div className="d-flex gap-10 flex-wrap">
              <ProductCard grid={grid} />
              </div>

            </div>
          </div>
        </div>
      </div>
    </div>
  </>
 );

};

export default OurStore;
```

5.4. SINGUP

```
import React from "react";
import BreadCrumb from "../components/BreadCrumb";
import Meta from "../components/Meta";
import { Link } from "react-router-dom";
const Signup = () => {
  return (
    <>
    <Meta title={"Sign Up"} />
    <BreadCrumb title="Sign Up" />
    <div className="login-wrapper py-5 home wrapper-2">
        <div className="container-xxl">
        <div className="row">
            <div className="col-12">
                <div className="auth-card">
                    <h3 className="text-center mb-3">Sign Up</h3>
                    <form action="" className="d-flex flex-column gap-15">
                    <div>
                            <input
                            type="text"
                            name="name"
                            placeholder="Name"
                            className="form-control"
                             />
                        </div>
                        <div>
                            <input
                            type="email"
                            name="email"
                            placeholder="Email"
                            className="form-control" />
                        </div>
                        <div>
                            <input
                            type="tel"
                            name="mobile"
                            placeholder="Mobile Number"
                            className="form-control" />
                        </div>
                        <div className="mt-1">
                            <input
                            type="password"
                            name="password"
                            placeholder="Password"
                            className="form-control" />
            </div>
            <div>
              <div className="mt-3 d-flex justify-content-center gap-15 align-items-center">
                <button className="button border-0">Sign Up</button>


              </div>
            </div>
          </form>
        </div>
      </div>
    </div>
    </div>
  </div>
  </>
 );
};

export default Signup;
```

5.5 BLOGS

```
import React from "react";
import BreadCrumb from "../components/BreadCrumb";
import Meta from "../components/Meta";
import BlogCard from "../components/BlogCard";


const Blog = () => {
 return (

 <>

 <Meta title={"Blogs"} />
  <BreadCrumb title="Blogs" />

  <div className="blog-wrapper home-wrapper-2 py-5">
    <div className="container-xxl">
      <div className="row">
        <div className="col-3">
        <div className="filter-card mb-3">
```

```
              <h3 className="filter-title">
                Shop By Categories
              </h3>
              <ul className="ps-0">
                <li>Frame</li>
                <li>Lens</li>
                <li>SunGlasses</li>
                <li>Contact Lens</li>
              </ul>
            </div>
                </div>

                <div className="col-9"></div>
                <div className="row">
                    <div className="col-6 mb-3">
                    <BlogCard/>
                    </div>
                    <div className="col-6  mb-3">
                    <BlogCard/>
                    </div>
                    <div className="col-6 mb-3">
                    <BlogCard/>
                    </div>
                    <div className="col-6  mb-3">
                    <BlogCard/>
                    </div>
                </div>


            </div>
        </div>
    </div>
    </>

  );
};

export default Blog;
```

5.6 CSS

```
*{
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}
:root {
  --color-131921: #131921;
  --color-3b4149:  #3b4149;
  --color-febd69: #febd69;
  --color-232f3e: #232f3e;
  --color-bf4800: #100701;
  --color-f5f5f7: #f5f5f7;
  --color-ededed: #ededed;
  --color-777777: #777777;
  --color-bf4800:#100701;
  --color-lclclb: #1c1c1b;
  --color-0dead4: #0dead4;
  --color-40E0D0: #40E0D0;




}
.section-heading {
  font-size: 26px;
  line-height: 32px;
  font-weight: 500;
  margin-bottom: 30px;
}
.card-wrapper {
  background-color: white;
  box-shadow: 0 0 10px #0000001a;
  padding: 15px;
}
body {
  font-family: 'Rubik', sans-serif;
}
a {
  text-decoration: none;
  display: inline-block;
}
.form-control:focus{
  box-shadow: none;
  border-color: inherit;
}
```

```
.gap-10 {
  gap: 5px;
  }
.gap-10 {
gap: 10px;
}
.gap-15 {
  gap: 15px;
}
.gap-30 {
  gap: 30px;
}
.button {
  background-color: var(--color-232f3e);
  color: white;
  font-size: 14px;
  padding: 13px 33px;
  border-radius: 25px;
}
.button:hover {
  background-color: var(--color-febd69);
  color: #40E0D0;
}
/* Header Part */

.header-top-strip  {
  border-bottom: 1px solid var(--color-0dead4);
  background-color: var(--color-40E0D0);
}
.header-upper{
  background-color: var(--color-40E0D0);
}
.header-bottom{
  background-color: var(--color-0dead4);
}
 .header-upper .input-group-text{
  background-color: var(--color-febd69);
 }

 .header-upper .header-upper-links img{
  width: 40px;
  height: 40px;
 }
 .header-bottom .menu-links a,
  .header-bottom .dropdown button{
  color: white;
  font-size: 14px;
  line-height: 18px;
  font-weight: 400px;
  letter-spacing: .3s;
  text-transform: uppercase;
 }

 .header-bottom .dropdown button:focus{
  box-shadow: none;

 }
 .header-bottom .dropdown-menu {
  background-color: var(--color-131921);
  width: 100%;
  transform: translate3d(0px, 57px, 0px) !important;
 }
 .header-bottom .dropdown-item {
  padding: 20px;
  margin-bottom: 3px;
  border-bottom: 1px solid var(--color-3b4149);
 }
 .header-bottom .dropdown-item:hover {
  background-color: transparent;
  color: var(--color-febd69) !important;
 }

.col-5{
  display: flex;
}


/* Header Part End Here  */

/* Footer Part */

footer {
  background-color: var(--color-40E0D0);
}
footer:not(:first-child) {
  border-top: 1px solid rgba(255, 255, 255, 0.1);
```

```
}
footer .input-group-text {
  background-color: var(--color-232f3e);
  color: white;
  padding: 8px 17px;
}
/* Footer Part End Here */

/* Home Page Start */

.main-banner-content {
  top: 15%;
  left: 5%;
}
.main-banner-content h4 {
  font-size: 16px;
  font-weight: 450;
  line-height: 24px;
  color: var(--color-bf4800);
  margin: 0 0 12px;
  letter-spacing: 0.3px;
  text-transform: uppercase;
}

.main-banner-content h5 {
  font-size: 42px;
  line-height: 54px;
  letter-spacing: -2px;
  font-weight: 500;
  text-transform: none;
}

.main-banner-content p {
  font-size: 18px;
  line-height: 28px;
  letter-spacing: 0.4;
  color: var(--color-131921);
}
.small-banner-content {
  top: 15%;
  left: 10%;
}
.small-banner-content h4 {
  font-size: 20px;
  font-weight: 450;
  line-height: 24px;
  color: var(--color-bf4800);
  margin: 0 0 12px;
  letter-spacing: 0.3px;
  text-transform: uppercase;
}

.small-banner-content h5 {
  font-size: 22px;
  line-height: 34px;
  letter-spacing: -2px;
  font-weight: 500;
  text-transform: none;
}

.small-banner-content p {
  font-size: 14px;
  line-height: 22px;
  letter-spacing: 0.4;
  color: var(--color-131921);
}
.small-banner {
  width: 49%;
}


/* home wrapper 2 */
.home-wrapper-2 {
  background-color: var(--color-f5f5f7);

}

/* home wrapper 2 end here */

/* categories section */

.categories {
  background-color: white;
  box-shadow: 0 0 10px #0000001a;
  padding: 15px;
}
```

```
.categories > div {
  width: 25%;
  padding: 10px, 10px, 10px, 20px;
}
.categories > div >div {
  width: 60%;
}
.categories > div:nth-child(1),
.categories > div:nth-child(2),
.categories > div:nth-child(3),
.categories > div:nth-child(4){
  border-bottom: 1px solid var(--color-ededed);
}
.categories > div:nth-child(1),
.categories > div:nth-child(2),
.categories > div:nth-child(3){
  border-right: 1px solid var(--color-ededed);
}
.categories > div:nth-child(5),
.categories > div:nth-child(6),
.categories > div:nth-child(7){
  border-right: 1px solid var(--color-ededed);
}
/* categories section part end here */

/* Home Page Part End Here */

/* Blog Part */

.blog-card {
  background-color: white;

  border-radius: 10px;
}

.blog-card img {
  border-top-left-radius: 10px;
  border-top-right-radius: 10px;
}

.blog-content {
  padding: 15px;
}
```

```
.blog-contentp.date {
  font-size: 13px;
  line-height: 24px;
  text-transform: uppercase;
  letter-spacing: 0;
  font-weight: 400;
  color: var(--color-777777);
  padding: 0;
}
.blog-content h5 {
  font-size: 10px;
  color: var(--color-131921);
}
.blog-content p.desc {
  font-size: 13px;
  line-height: 22px;
  color: var(--color-777777);
}
.single-blog-card a{
  font-size: 18px;
  color: var(--color-777777);
  margin-bottom: 20px;
}

.single-blog-card h3{
  font-size: 24px;
  font-weight: 600;
  color: var(--color-lclclb);
}
.single-blog-card p {
  color: var(--color-777777);
  font-size: 14px;
}

/* Blog Part ends here */

/* product card */

.product-card {
  padding: 15px;
  background-color: white;
  border-radius: 10px;
  overflow: hidden;
  width: 100%;
}
.product-card .product-image{
  height: 270px;
}
.product-card .product-image img {
  object-fit: contain;
}

.product-card .product-image>img:last-child{
display: none;

}
.product-card:hover .product-image > img:nth-child(1) {
  display: none;

}
.product-card:hover .product-image > img:last-child {
  display: block;

}
.product-card:hover .action-bar{
  right: 15px;
}
.product-card .product-details h6{
  color: var(--color-bf4800);
  font-size: 13px;

}
.product-card .product-details h5{
  font-size: 15px;
  color: var(--color-lclclb);
}
.product-card .product-details p.price{
  font-size: 16px;
  color: var(--color-lclclb);
}
.product-card p.description {
  font-size: 13px;
  color: var(--color-777777);
  margin-right: 20px;
}
.action-bar{
  top: 10%;
  right: -23px;
  transition: 0.3s;
}
.wishlist-icon{
  top: 2%;
  right: 15px;

}


/* product card ends here */

 /* Special Product */

 .special-product-card{
  padding: 20px 10px;
  border-radius: 10px;
  background-color: white;
 }


 /* Special Product ends here */

 /* famous card */
.famous-card .famous-content{
  top: 10%;
  left: 10%;
}
.famous-card img{
  border-radius: 10px;
}
.famous-content h5{
  font-size: 13px;
  line-height: 20px;
  font-weight: 400px;
  color: white;
  margin-bottom: 7px;
}

.famous-content h6{
  font-size: 24px;
  line-height: 38px;
  font-weight: 500;
  color: white;

}

.famous-content p{
  font-size: 16px;
  line-height: 24px;
  font-weight: 100;
  color: white;
}


 /* famous card ends here*/


 /* Our Store */

.filter-card {
  background-color: white;
  border-radius: 10px;
  padding: 15px 15px 25px;
}
.filter-title {
  font-size: 16px;
  line-height: 20px;
  font-weight: 600;
  color: var(--color-lclclb);
  margin-bottom: 20px;
}
.filter-card ul,
.filter-card .form-check-label {
  list-style-type: none;
  font-size: 13px;
  line-height: 30px;
  color: var(--color-777777);
  cursor: pointer;
}
.filter-card .form-check-input:focus {
  box-shadow: none;
  border-color: rgba(0, 0, 0, 0, 25);
}
.filter-card .form-check-input:checked{
  background-color: var(--color-febd69);
  border-color: var(--color-febd69);
}
.sub-title  {
  font-size: 14px;
  font-size: 600;
  color: var(--color-lclclb);
  margin-bottom: 15px;
  margin-top: 20px;
}

.colors {
  list-style-type: none;
  display: flex;
  flex-wrap: wrap;
  gap: 10px;
}
.colors li {
  width: 20px;
  height: 20px;
  background-color: red;
  border-radius: 50%;
}
.random-products h5{
  font-size: 14px;
  margin-bottom: 8px;

}
.random-products:first-child{
  border-bottom: 1px solid var(--color-ededed);
}
.filter-sort-grid {
  padding: 10px;
  background-color: white;
  border-radius: 10px;
}
.grid img{
  width: 35px;
  height: 35px;
  background-color: #f2f2f2;
  padding: 8px;
  border-radius: 3px;
  cursor: pointer;
```

```
}
.gr-12{
  width: 100%;
}
.gr-6{
  width: 49%;
}
.gr-4{
  width: 32%;
}
.gr-3{
  width: 24%;
}
.gr-12 .product-card {
  display: flex;
  gap: 20px;
}
.gr-66 .product-image {
  width: 100%;
}
.gr-6 .product-image img{
  margin: auto;
  display: block;

}
.gr-12 .product-card .product-image {
  width: 270px;
}
.gr-12 .product-card .product-image {
  height: auto;
}

.gr-12 .action-bar {
  top: 17%;
}
 /* Our Store ends here */

 /* Contact Page */
.contact-inner-wrapper {
  padding: 30px 20px;
  border-radius: 15px;
  background-color: white;
  gap: 15px;
}
.contact-inner-wrapper > div {
  width: 48;
}
.contact-title {
  font-size: 26px;
  font-weight: 500;
  text-align: left;

}
.contact-inner-wrapper from .form-control{
  background-color: var(--color-f5f5f7);
  border-radius: 10px;
  border-color: transparent;
  padding: 10px;
}
.contact-inner-wrapper ul {
  list-style: none;
}
.contact-inner-wrapper address,.contact-inner-wrapper a,.contact-inner-wrapper p{
  color: var(--color-777777)
}

 /* Contact Page ends*/

/* Compare Product Page */

.compare-product-cart{
  padding: 20px 15px;
  background-color: white;
  border-radius: 10px;
}
.compare-product-cart .cross,
.wishlist-card .cross {
  top: 10px;
  right: 10px;
  width: 25px;
  padding: 3px;
  cursor: point 3px;
}
.compare-product-details h5.title,
.wishlist-card h5.title {
  font-size: 15px;
  line-height: 22px;
  font-weight: 500;
  color: var(--color-lclclb);
}
.compare-product-details h6.price,
.wishlist-card h6.price {
  font-size: 15px;
  line-height: 22px;
  font-weight: 400;
}
.compare-product-cart .product-detail {
  display: flex;
  justify-content: space-between;
  align-items: center;
  padding: 10px 0px;
  border-top: 1px solid rgba(0, 0, 0, 0.1);
}

/* Compare Product Page  end here*/

/* auth-card */

.auth-card {
  padding: 20px;
  background-color: white;
  width: 500px;
  margin: 30px auto;
  border-radius: 10px;
}

.auth-card h3 {
  font-size: 19px;
  font-weight: 500;
  color: var(--color-777777);
}
.auth-card p {
  font-size: 13px;
  color: var(--color-777777);
}

.auth-card .form-control{
  height: 50px;
  background-color: var(--color-f5f5f7);
  color: var(--color-777777);
  font-size: 14px;
  border-color: transparent;
}
.auth-card a {
  font-size: 14px;
  font-weight: 400;
  color: var(--color-lclclb);
}
.auth-card .signup {
  background-color: var(--color-febd69);
  color: var(--color-131921);
}
.auth-card .signup.singup:hover {
  background-color: var(--color-131921);
  color: white;
}
/* auth-card ends */

/* Policy Pages */
.policy {
  background-color: white;
  padding: 30px;
}

/* Policy Pages */

/* Main Peroduct Page */

.description-wrapper h4  .reviews-wrapper h3{
  font-size: 26px;
  color: #1c1c1b;
  margin-bottom: 10px;
}
.description-wrapper p {
  font-size: 14px;
  color: var(--color-777777);
}

.review-ineer-wrapper {
  background-color: white;
  padding: 16px;
```

```
}

.review-head {
  border-bottom: 1px solid rgba(0, 0, 0, 0.1);
  padding-bottom: 20px;
}
.review-head h4 {
  font-size: 18px;
  color: var(--color-lclclb);

}
.review-head p {
  font-size: 14px;
  color: var(--color-777777);

}
.review-form h4 {
  font-size: 16px;
  color: var(--color-777777);
}
.review-form {
  border-bottom: 1px solid rgba(0, 0, 0, 0.1);
}
.review:not(:last-child)
{
  border-bottom: 1px solid rgba(0, 0, 0, 0.1);
}.review p {
  font-size: 14px;
  color: var(--color-777777);
}

.main-product-image {
  padding: 20px;
  background-color: white;
  border-radius: 10px;
}
.main-product-image>div {
  border: 1px solid rgba(0, 0, 0, 0.25);
}
.main-product-image img {
  width: 100% !important;
  height: 500px !important;
  object-fit: cover;
```

```
}
.other-product-images {
  background-color: white;
  padding: 20px;

}
.other-product-images>div{
  padding: 20px;
  width: 48%;
  border: 1px solid rgba(0, 0, 0, 0.25);
}
.main-product-image .js-images-zoom__zommed-image {
  left: 230px !important;
  padding: 10px;
  background-color: white;
  border-radius: 10px;
}
.main-product-details {
  padding: 20px;
  background-color: white;
  border-radius: 10px;
}
.border-bottom {
  border-bottom: 1px solid rgba(0, 0, 0, 0, 0.25);
}
.main-product-details h3.title {
  font-size: 18px;
  font-weight: 600;
  color: var(--color-lclclb);
}
```

5.7 COMPARE PRODUCT

```
import React from "react";
import BreadCrumb from "../components/BreadCrumb";
import Meta from "../components/Meta";

const CompareProduct = () => {
    return <>
     <Meta title={"Compare Products"} />
    <BreadCrumb title="Compare Products" />
```

```
  <div className="compare-product-wrapper py-5 home-wrapper-2">
    <div className="container-xxl">
      <div className="row">
        <div className="col-3">
          <div className="compare-product-cart position-relative">
            <img src="images/cross.svg"
            alt="cross"
            className="position-absolute cross img-fluid"/>
            <div className="product-cart-image">
              <img src="images/compare1.png" alt="watch" />
              <div>
                <div className="compare-product-details">
                  <h5 className="title">
                  White Marble Full Rim Square Acute STUDIO Indian Accent
LK E15172-C1 Eyeglasses - 205365
                  </h5>
                  <h6 className="price mb-3">₹500</h6>

                  <div>
                    <div className="product-detail">
                      <h5>Brand:</h5>
                      <p>Paul Frank</p>
                    </div>
                    <div className="product-detail">
                      <h5>Type:</h5>
                      <p>Frame</p>
                    </div>
                    <div className="product-detail">
                      <h5>Availability:</h5>
                      <p>In Stock</p>
                    </div>
                    <div className="product-detail">
                      <h5>Color:</h5>

                    </div>
                    <div className="product-detail">
                      <h5>Size:</h5>
                      <div className="d-flex gap-10">
                        <p>S</p>
                        <p>M</p>
                      </div>
                    </div>
                  </div>
```

```
                              </div>
                          </div>
                      </div>
                  </div>
              </div>
          </div>
      </div>
    </div>
    </>
};

export default CompareProduct;
```

5.7 SINGLE BLOG

```
import React from "react";
import { Link } from 'react-router-dom';
import BreadCrumb from "../components/BreadCrumb";
import { HiOutlineArrowLeft } from "react-icons/hi";
import Meta from "../components/Meta";
const SingleBlog = () => {
  return (
    <>
     <Meta title={"Dynamic Blog Name"} />
    <BreadCrumb title="Dynamic Blog Name" />
    <div className="blog-wrapper home-wrapper-2 py-5">
        <div className="container-xxl">
            <div className="row">

                <div className="col-12">
                  <div className="single-blog-card">
                    <Link to="/blogs" className="d-flex align-items-center gap-10">
                    <HiOutlineArrowLeft className="fs-4"/>
                      Go back to Blogs</Link>
                    <h3 className="title">
                    A Beautiful Sunday Morning Renaissance
                    </h3>
                    <img
                    src="images/blog.3.webg"
                    className="img-fluid w-100 my-4"
                    alt="blog" />
                    <p>Himalaya brings to you a range of accredited eye wear that must be used by labour that handles such threat to eyes while working.

                    </p>
                  </div>
                </div>




            </div>
        </div>
    </div>
    </>
  );
};

export default SingleBlog;
```

5.9 FORGOT PASSWORD

```
import React from "react";
import BreadCrumb from "../components/BreadCrumb";
import Meta from "../components/Meta";
import { Link } from 'react-router-dom';
const Forgotpassword = () => {
  return (
    <>
    <Meta title={"Forgot Password"} />
    <BreadCrumb title="Forgot Password" />
    <div className="login-wrapper py-5 home wrapper-2">
    <div className="container-xxl">
    <div className="row">
            <div className="col-12">
                <div className="auth-card">
                    <h3 className="text-center mb-3">Reset Your Password</h3>
                    <p className="text-center mt-2 mb-3">
                        We will send you an email to rest your password</p>
                    <form action="" className="d-flex flex-column gap-15">
                        <div>
                            <input type="email"
                            name="email"
                            placeholder="Email"
```

```
                      className="form-control" />
                    </div>

                    <div>
                      <Link to='/forgot-password'>Forgot Password</Link>
                      <div className="mt-3 d-flex justify-content-center flex-column gap-
15 align-items-center">
                        <button className="button border-0" type="submit">
                          Submit
                          </button>
                        <Link to='/login'>Cancel</Link>

                      </div>
                    </div>
                  </form>
                </div>
              </div>
            </div>
          </div>

          </div>
          </>

  );
};

export default Forgotpassword;
```

5.10 CONTACT

```
import React from 'react'
import BreadCrumb from "../components/BreadCrumb";
import Meta from "../components/Meta";
import { AiOutlineHome, AiOutlineMail} from "react-icons/ai";
import { BiPhoneCall, BiInfoCircle } from "react-icons/bi";

const Contact = () => {
  return (
    <>
      <Meta title={"Contact Us"} />
    <BreadCrumb title="Contact Us" />
    <div className="contact-wrapper py-5 home-wrapper-2">
      <div className="container-xxl">
    <div className="row">
     <div className="col-12">
     <iframe
src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d6924.2725625364
79!2d76.40529039301202!3d29.80260695018468!2m3!1f0!2f0!3f0!3m2!1i1024!2i768
!4f13.1!3m3!1m2!1s0x3911e198baaa226d%3A0x455ecf41f2aec02!2sHope%20Medica
l%20Center!5e0!3m2!1sen!2sin!4v1682328574516!5m2!1sen!2sin"
     width="1200"
     height="450"
     classNmaee="border-0 w-100;"
     allowFullScreen=""
     loading="lazy"
     referrerpolicy="no-referrer-when-downgrade">
      </iframe>
     </div>
     <div className="col-12 mt-5">
     <div className="contact-inner-wrapper d-flex justify-contact-between ">
      <div>
      <h3 className="contact-title mb-4">Contact</h3>
      <form action="" className="d-flex flex-column gap-15">
       <div>
        <input type="text" className="form-control"
        placeholder="Name"
         />
       </div>
       <div>
        <input type="email" className="form-control"
        placeholder="Email"
        />
       </div>
       <div>
        <input type="tel" className="form-control"
        placeholder="Mobile Number"
         />
       </div>
       <div>
        <textarea
        name=""
        id=""
        className="w-100 form-control"
        cols="30"
        rows="4"
        placeholder="Comments"
        ></textarea>
       </div>
       <div>
        <button className="button border-0">Submit</button>
       </div>
      </form>

      </div>
      <div>
      <h3 className="contact-title mb-4">Get in touch with Us</h3>
      <div>
       <ul className="ps-0">
        <li className="mb-3 d-flex gap-15 align-items-center">
         <AiOutlineHome className="fs-5" />
         <address className="mb-0">Acute Eyewear, Hope Medical Centre, Gali
No 1,
MITC Colony, Karnal Road, Kaithal
136027 (Near Padma City Mall)
Haryana</address>
         </li>
        <li className="mb-3 d-flex gap-15 align-items-center">
         <BiPhoneCall className="fs-5" />
         <a href="tel:+91 00000000">+91 8755859391</a>
        </li>
        <li className="mb-3 d-flex gap-15 align-items-center">
         <AiOutlineMail className="fs-5" />
         <a
href="mailto:akshepbhosdibaleko@gmail.com">acuteeyewearsolution@gmail.com</a>
         </li>
         <li className="mb-3 d-flex gap-15 align-items-center">
         <BiInfoCircle className="fs-5" />
         <p className="mb-0">Monday - Friday 10 AM - 8 PM</p>
         </li>


       </ul>
      </div>
      </div>




     </div>
     </div>
    </div>
   </div>
```

```
    </div>


    </>

  )
};

export default Contact;
```

5.11 BACKEND(INDEX)

```
const bodyParser = require("body-parser");
const express = require("express");
const dbConnect = require("./config/dbConnect");
const { notFound, errorHandler } = require("./middlewares/errorHandler");
const app = express();
const dotenv = require("dotenv").config();
const PORT = 5000;
const authRouter = require("./routes/authRoute");
const productRouter = require("./routes/productRoute");
const blogRouter = require("./routes/blogRoute");
const categoryRouter = require("./routes/prodcategoryRoute");
const blogcategoryRouter = require("./routes/blogCatRoute");
const brandRouter = require("./routes/brandRoute");
const colorRouter = require("./routes/colorRoute");
const enqRouter = require("./routes/enqRoute");
const couponRouter = require("./routes/couponRoute");
const uploadRouter = require("./routes/uploadRoute");
const cookieParser = require("cookie-parser");
const morgan = require("morgan");
const cors = require("cors");

dbConnect();
app.use(morgan("dev"));
app.use(cors());
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended: false }));
app.use(cookieParser());
app.use("/api/user", authRouter);
app.use("/api/product", productRouter);
app.use("/api/blog", blogRouter);
app.use("/api/category", categoryRouter);
app.use("/api/blogcategory", blogcategoryRouter);
app.use("/api/brand", brandRouter);
app.use("/api/coupon", couponRouter);
app.use("/api/color", colorRouter);
app.use("/api/enquiry", enqRouter);
app.use("/api/upload", uploadRouter);

app.use(notFound);
app.use(errorHandler);
app.listen(PORT, () => {
  console.log(`Server is running  at PORT ${PORT}`);
});
```

5.12 AUTHROUTE

```
const express = require("express");
const {
  createUser,
  loginUserCtrl,
  getallUser,
  getaUser,
  deleteaUser,
  updatedUser,
  blockUser,
  unblockUser,
  handleRefreshToken,
  logout,
  updatePassword,
  forgotPasswordToken,
  resetPassword,
  loginAdmin,
  getWishlist,
  saveAddress,
  userCart,
  getUserCart,
  emptyCart,
  applyCoupon,
  createOrder,
  getOrders,
  updateOrderStatus,
  getAllOrders,
} = require("../controller/userCtrl");
```

```
const { authMiddleware, isAdmin } = require("../middlewares/authMiddleware");
const router = express.Router();
router.post("/register", createUser);
router.post("/forgot-password-token", forgotPasswordToken);

router.put("/reset-password/:token", resetPassword);

router.put("/password", authMiddleware, updatePassword);
router.post("/login", loginUserCtrl);
router.post("/admin-login", loginAdmin);
router.post("/cart", authMiddleware, userCart);
router.post("/cart/applycoupon", authMiddleware, applyCoupon);
router.post("/cart/cash-order", authMiddleware, createOrder);
router.get("/all-users", getallUser);
router.get("/get-orders", authMiddleware, getOrders);
router.get("/getallorders", authMiddleware, isAdmin, getAllOrders);
router.post("/getorderbyuser/:id", authMiddleware, isAdmin, getAllOrders);
router.get("/refresh", handleRefreshToken);
router.get("/logout", logout);
router.get("/wishlist", authMiddleware, getWishlist);
router.get("/cart", authMiddleware, getUserCart);

router.get("/:id", authMiddleware, isAdmin, getaUser);
router.delete("/empty-cart", authMiddleware, emptyCart);
router.delete("/:id", deleteaUser);
router.put(
  "/order/update-order/:id",
  authMiddleware,
  isAdmin,
  updateOrderStatus
);
router.put("/edit-user", authMiddleware, updatedUser);
router.put("/save-address", authMiddleware, saveAddress);
router.put("/block-user/:id", authMiddleware, isAdmin, blockUser);
router.put("/unblock-user/:id", authMiddleware, isAdmin, unblockUser);

module.exports = router;
```

5.13 BLOGCATROUTE

```
const express = require("express");
const {
  createCategory,
  updateCategory,
  deleteCategory,
  getCategory,
  getallCategory,
} = require("../controller/blogCatCtrl");
const { authMiddleware, isAdmin } = require("../middlewares/authMiddleware");
const router = express.Router();

router.post("/", authMiddleware, isAdmin, createCategory);
router.put("/:id", authMiddleware, isAdmin, updateCategory);
router.delete("/:id", authMiddleware, isAdmin, deleteCategory);
router.get("/:id", getCategory);
router.get("/", getallCategory);

module.exports = router;
```

5.14 COLORROUTE

```
const express = require("express");
const {
  createColor,
  updateColor,
  deleteColor,
  getColor,
  getallColor,
} = require("../controller/colorCtrl");
const { authMiddleware, isAdmin } = require("../middlewares/authMiddleware");
const router = express.Router();

router.post("/", authMiddleware, isAdmin, createColor);
router.put("/:id", authMiddleware, isAdmin, updateColor);
router.delete("/:id", authMiddleware, isAdmin, deleteColor);
router.get("/:id", getColor);
router.get("/", getallColor);

module.exports = router;
```

5.15 USERCTRL

```
const User = require("../models/userModel");
const Product = require("../models/productModel");
const Cart = require("../models/cartModel");
const Coupon = require("../models/couponModel");
const Order = require("../models/orderModel");
const uniqid = require("uniqid");

const asyncHandler = require("express-async-handler");
const { generateToken } = require("../config/jwtToken");
const validateMongoDbId = require("../utils/validateMongodbId");
const { generateRefreshToken } = require("../config/refreshtoken");
const crypto = require("crypto");
const jwt = require("jsonwebtoken");



// Create a User ---------------------------------------------

const createUser = asyncHandler(async (req, res) => {
  /**
   * TODO:Get the email from req.body
   */
  const email = req.body.email;
  /**
   * TODO:With the help of email find the user exists or not
   */
  const findUser = await User.findOne({ email: email });

  if (!findUser) {
    /**
     * TODO:if user not found user create a new user
     */
    const newUser = await User.create(req.body);
    res.json(newUser);
  } else {
    /**
     * TODO:if user found then thow an error: User already exists
     */
    throw new Error("User Already Exists");
  }
});

// Login a user
const loginUserCtrl = asyncHandler(async (req, res) => {
  const { email, password } = req.body;
  // check if user exists or not
  const findUser = await User.findOne({ email });
  if (findUser && (await findUser.isPasswordMatched(password))) {
    const refreshToken = await generateRefreshToken(findUser?._id);
    const updateuser = await User.findByIdAndUpdate(
      findUser.id,
      {
        refreshToken: refreshToken,
      },
      { new: true }
    );
    res.cookie("refreshToken", refreshToken, {
      httpOnly: true,
      maxAge: 72 * 60 * 60 * 1000,
    });
    res.json({
      _id: findUser?._id,
      firstname: findUser?.firstname,
      lastname: findUser?.lastname,
      email: findUser?.email,
      mobile: findUser?.mobile,
      token: generateToken(findUser?._id),
    });
  } else {
    throw new Error("Invalid Credentials");
  }
});

// admin login

const loginAdmin = asyncHandler(async (req, res) => {
  const { email, password } = req.body;
  // check if user exists or not
  const findAdmin = await User.findOne({ email });
  if (findAdmin.role !== "admin") throw new Error("Not Authorised");
  if (findAdmin && (await findAdmin.isPasswordMatched(password))) {
    const refreshToken = await generateRefreshToken(findAdmin?._id);
    const updateuser = await User.findByIdAndUpdate(
      findAdmin.id,
      {
        refreshToken: refreshToken,
      },
      { new: true }
    );
    res.cookie("refreshToken", refreshToken, {
      httpOnly: true,
      maxAge: 72 * 60 * 60 * 1000,
    });
    res.json({
      _id: findAdmin?._id,
      firstname: findAdmin?.firstname,
      lastname: findAdmin?.lastname,
      email: findAdmin?.email,
      mobile: findAdmin?.mobile,
      token: generateToken(findAdmin?._id),
    });
  } else {
    throw new Error("Invalid Credentials");
  }
});

// handle refresh token

const handleRefreshToken = asyncHandler(async (req, res) => {
  const cookie = req.cookies;
  if (!cookie?.refreshToken) throw new Error("No Refresh Token in Cookies");
  const refreshToken = cookie.refreshToken;
  const user = await User.findOne({ refreshToken });
  if (!user) throw new Error(" No Refresh token present in db or not matched");
  jwt.verify(refreshToken, process.env.JWT_SECRET, (err, decoded) => {
    if (err || user.id !== decoded.id) {
      throw new Error("There is something wrong with refresh token");
    }
    const accessToken = generateToken(user?._id);
    res.json({ accessToken });
  });
});

// logout functionality

const logout = asyncHandler(async (req, res) => {
  const cookie = req.cookies;
  if (!cookie?.refreshToken) throw new Error("No Refresh Token in Cookies");
  const refreshToken = cookie.refreshToken;
  const user = await User.findOne({ refreshToken });
  if (!user) {
    res.clearCookie("refreshToken", {
      httpOnly: true,
      secure: true,
    });
    return res.sendStatus(204); // forbidden
  }
  await User.findOneAndUpdate(refreshToken, {
    refreshToken: "",
  });
  res.clearCookie("refreshToken", {
    httpOnly: true,
    secure: true,
  });
  res.sendStatus(204); // forbidden
});

// Update a user

const updatedUser = asyncHandler(async (req, res) => {
  const { _id } = req.user;
  validateMongoDbId(_id);

  try {
    const updatedUser = await User.findByIdAndUpdate(
      _id,
      {
        firstname: req?.body?.firstname,
        lastname: req?.body?.lastname,
        email: req?.body?.email,
        mobile: req?.body?.mobile,
      },
      {
        new: true,
      }
    );
    res.json(updatedUser);
  } catch (error) {
    throw new Error(error);
  }
});

// save user Address

const saveAddress = asyncHandler(async (req, res, next) => {
  const { _id } = req.user;
  validateMongoDbId(_id);

  try {
    const updatedUser = await User.findByIdAndUpdate(
      _id,
      {
        address: req?.body?.address,
      },
      {
        new: true,
      }
    );
    res.json(updatedUser);
  } catch (error) {
    throw new Error(error);
  }
});

// Get all users

const getallUser = asyncHandler(async (req, res) => {
  try {
    const getUsers = await User.find().populate("wishlist");
    res.json(getUsers);
  } catch (error) {
    throw new Error(error);
  }
});

// Get a single user

const getaUser = asyncHandler(async (req, res) => {
  const { id } = req.params;
  validateMongoDbId(id);

  try {
    const getaUser = await User.findById(id);
    res.json({
      getaUser,
    });
  } catch (error) {
    throw new Error(error);
  }
});

// Get a single user
```

```
const deleteaUser = asyncHandler(async (req, res) => {
  const { id } = req.params;
  validateMongoDbId(id);

  try {
    const deleteaUser = await User.findByIdAndDelete(id);
    res.json({
      deleteaUser,
    });
  } catch (error) {
    throw new Error(error);
  }
});

const blockUser = asyncHandler(async (req, res) => {
  const { id } = req.params;
  validateMongoDbId(id);

  try {
    const blockusr = await User.findByIdAndUpdate(
      id,
      {
        isBlocked: true,
      },
      {
        new: true,
      }
    );
    res.json(blockusr);
  } catch (error) {
    throw new Error(error);
  }
});

const unblockUser = asyncHandler(async (req, res) => {
  const { id } = req.params;
  validateMongoDbId(id);

  try {
    const unblock = await User.findByIdAndUpdate(
      id,
      {
        isBlocked: false,
      },
      {
        new: true,
      }
    );
    res.json({
      message: "User UnBlocked",
    });
  } catch (error) {
    throw new Error(error);
  }
});

const updatePassword = asyncHandler(async (req, res) => {
  const { _id } = req.user;
  const { password } = req.body;
  validateMongoDbId(_id);
  const user = await User.findById(_id);
  if (password) {
    user.password = password;
    const updatedPassword = await user.save();
    res.json(updatedPassword);
  } else {
    res.json(user);
  }
});

const forgotPasswordToken = asyncHandler(async (req, res) => {
  const { email } = req.body;
  const user = await User.findOne({ email });
  if (!user) throw new Error("User not found with this email");
  try {
    const token = await user.createPasswordResetToken();
    await user.save();
    const resetURL = `Hi, Please follow this link to reset Your Password. This link is valid till 10 minutes from now. <a href='http://localhost:5000/api/user/reset-password/${token}'>Click Here</>`;
    const data = {
      to: email,
      text: "Hey User",
      subject: "Forgot Password Link",
      htm: resetURL,
    };
    sendEmail(data);
    res.json(token);
  } catch (error) {
    throw new Error(error);
  }
});

const resetPassword = asyncHandler(async (req, res) => {
  const { password } = req.body;
  const { token } = req.params;
  const hashedToken = crypto.createHash("sha256").update(token).digest("hex");
  const user = await User.findOne({
    passwordResetToken: hashedToken,
    passwordResetExpires: { $gt: Date.now() },
  });
  if (!user) throw new Error(" Token Expired, Please try again later");
  user.password = password;
  user.passwordResetToken = undefined;
  user.passwordResetExpires = undefined;
  await user.save();
  res.json(user);
});

const getWishlist = asyncHandler(async (req, res) => {
  const { _id } = req.user;
  try {
    const findUser = await User.findById(_id).populate("wishlist");
    res.json(findUser);
  } catch (error) {
    throw new Error(error);
  }
});

const userCart = asyncHandler(async (req, res) => {
  const { cart } = req.body;
  const { _id } = req.user;
  validateMongoDbId(_id);
  try {
    let products = [];
    const user = await User.findById(_id);
    // check if user already have product in cart
    const alreadyExistCart = await Cart.findOne({ orderby: user._id });
    if (alreadyExistCart) {
      alreadyExistCart.remove();
    }
    for (let i = 0; i < cart.length; i++) {
      let object = {};
      object.product = cart[i]._id;
      object.count = cart[i].count;
      object.color = cart[i].color;
      let getPrice = await Product.findById(cart[i]._id).select("price").exec();
      object.price = getPrice.price;
      products.push(object);
    }
    let cartTotal = 0;
    for (let i = 0; i < products.length; i++) {
      cartTotal = cartTotal + products[i].price * products[i].count;
    }
    let newCart = await new Cart({
      products,
      cartTotal,
      orderby: user?._id,
    }).save();
    res.json(newCart);
  } catch (error) {
    throw new Error(error);
  }
});

const getUserCart = asyncHandler(async (req, res) => {
  const { _id } = req.user;
  validateMongoDbId(_id);
  try {
    const cart = await Cart.findOne({ orderby: _id }).populate(
      "products.product"
    );
    res.json(cart);
  } catch (error) {
    throw new Error(error);
  }
});

const emptyCart = asyncHandler(async (req, res) => {
  const { _id } = req.user;
  validateMongoDbId(_id);
  try {
    const user = await User.findOne({ _id });
    const cart = await Cart.findOneAndRemove({ orderby: user._id });
    res.json(cart);
  } catch (error) {
    throw new Error(error);
  }
});

const applyCoupon = asyncHandler(async (req, res) => {
  const { coupon } = req.body;
  const { _id } = req.user;
  validateMongoDbId(_id);
  const validCoupon = await Coupon.findOne({ name: coupon });
  if (validCoupon === null) {
    throw new Error("Invalid Coupon");
  }
  const user = await User.findOne({ _id });
  let { cartTotal } = await Cart.findOne({
    orderby: user._id,
  }).populate("products.product");
  let totalAfterDiscount = (
    cartTotal -
    (cartTotal * validCoupon.discount) / 100
  ).toFixed(2);
  await Cart.findOneAndUpdate(
    { orderby: user._id },
    { totalAfterDiscount },
    { new: true }
  );
  res.json(totalAfterDiscount);
});

const createOrder = asyncHandler(async (req, res) => {
  const { COD, couponApplied } = req.body;
  const { _id } = req.user;
  validateMongoDbId(_id);
  try {
    if (!COD) throw new Error("Create cash order failed");
    const user = await User.findById(_id);
    let userCart = await Cart.findOne({ orderby: user._id });
    let finalAmout = 0;
    if (couponApplied && userCart.totalAfterDiscount) {
      finalAmout = userCart.totalAfterDiscount;
    } else {
      finalAmout = userCart.cartTotal;
    }

    let newOrder = await new Order({
      products: userCart.products,
      paymentIntent: {
        id: uniqid(),
        method: "COD",
        amount: finalAmout,
        status: "Cash on Delivery",
        created: Date.now(),
        currency: "usd",
      },
      orderby: user._id,
      orderStatus: "Cash on Delivery",
    }).save();
    let update = userCart.products.map((item) => {
      return {
        updateOne: {
          filter: { _id: item.product._id },
          update: { $inc: { quantity: -item.count, sold: +item.count } },
        },
      };
    });
    const updated = await Product.bulkWrite(update, {});
    res.json({ message: "success" });
  } catch (error) {
    throw new Error(error);
  }
});

const getOrders = asyncHandler(async (req, res) => {
  const { _id } = req.user;
  validateMongoDbId(_id);
  try {
    const userorders = await Order.findOne({ orderby: _id })
      .populate("products.product")
      .populate("orderby")
      .exec();
    res.json(userorders);
  } catch (error) {
    throw new Error(error);
  }
});

const getAllOrders = asyncHandler(async (req, res) => {
  try {
    const alluserorders = await Order.find()
```

```
      .populate("products.product")
      .populate("orderby")
      .exec();
    res.json(alluserorders);
  } catch (error) {
    throw new Error(error);
  }
});
const getOrderByUserId = asyncHandler(async (req, res) => {
  const { id } = req.params;
  validateMongoDbId(id);
  try {
    const userorders = await Order.findOne({ orderby: id })
      .populate("products.product")
      .populate("orderby")
      .exec();
    res.json(userorders);
  } catch (error) {
    throw new Error(error);
  }
});
const updateOrderStatus = asyncHandler(async (req, res) => {
  const { status } = req.body;
  const { id } = req.params;
  validateMongoDbId(id);
  try {
    const updateOrderStatus = await Order.findByIdAndUpdate(
      id,
      {
        orderStatus: status,
        paymentIntent: {
          status: status,
        },
      },
      { new: true }
    );
    res.json(updateOrderStatus);
  } catch (error) {
    throw new Error(error);
  }
});

module.exports = {
  createUser,
  loginUserCtrl,
  getallUser,
  getaUser,
  deleteaUser,
  updatedUser,
  blockUser,
  unblockUser,
  handleRefreshToken,
  logout,
  updatePassword,
  forgotPasswordToken,
  resetPassword,
  loginAdmin,
  getWishlist,
  saveAddress,
  userCart,
  getUserCart,
  emptyCart,
  applyCoupon,
  createOrder,
  getOrders,
  updateOrderStatus,
  getAllOrders,
  getOrderByUserId,
};
```

5.16 PRODUCTCTRL

```
const Product = require("../models/productModel");
const User = require("../models/userModel");
const asyncHandler = require("express-async-handler");
const slugify = require("slugify");
const validateMongoDbId = require("../utils/validateMongodbId");

const createProduct = asyncHandler(async (req, res) => {
  try {
    if (req.body.title) {
      req.body.slug = slugify(req.body.title);
    }
    const newProduct = await Product.create(req.body);
    res.json(newProduct);
  } catch (error) {
    throw new Error(error);
  }
});

const updateProduct = asyncHandler(async (req, res) => {
  const id = req.params;
  validateMongoDbId(id);
  try {
    if (req.body.title) {
      req.body.slug = slugify(req.body.title);
    }
    const updateProduct = await Product.findOneAndUpdate({ id }, req.body, {
      new: true,
    });
    res.json(updateProduct);
  } catch (error) {
    throw new Error(error);
  }
});

const deleteProduct = asyncHandler(async (req, res) => {
  const id = req.params;
  validateMongoDbId(id);
  try {
    const deleteProduct = await Product.findOneAndDelete(id);
    res.json(deleteProduct);
  } catch (error) {
    throw new Error(error);
  }
});

const getaProduct = asyncHandler(async (req, res) => {
  const { id } = req.params;
  validateMongoDbId(id);
  try {
    const findProduct = await Product.findById(id);
    res.json(findProduct);
  } catch (error) {
    throw new Error(error);
  }
});

const getAllProduct = asyncHandler(async (req, res) => {
  try {
    // Filtering
    const queryObj = { ...req.query };
    const excludeFields = ["page", "sort", "limit", "fields"];
    excludeFields.forEach((el) => delete queryObj[el]);
    let queryStr = JSON.stringify(queryObj);
    queryStr = queryStr.replace(/\b(gte|gt|lte|lt)\b/g, (match) => `$${match}`);

    let query = Product.find(JSON.parse(queryStr));

    // Sorting

    if (req.query.sort) {
      const sortBy = req.query.sort.split(",").join(" ");
      query = query.sort(sortBy);
    } else {
      query = query.sort("-createdAt");
    }

    // limiting the fields

    if (req.query.fields) {
      const fields = req.query.fields.split(",").join(" ");
      query = query.select(fields);
    } else {
      query = query.select("-__v");
    }

    // pagination

    const page = req.query.page;
    const limit = req.query.limit;
    const skip = (page - 1) * limit;
    query = query.skip(skip).limit(limit);
    if (req.query.page) {
      const productCount = await Product.countDocuments();
      if (skip >= productCount) throw new Error("This Page does not exists");
    }
    const product = await query;
    res.json(product);
  } catch (error) {
    throw new Error(error);
  }
});
const addToWishlist = asyncHandler(async (req, res) => {
  const { _id } = req.user;
  const { prodId } = req.body;
  try {
    const user = await User.findById(_id);
    const alreadyadded = user.wishlist.find((id) => id.toString() === prodId);
    if (alreadyadded) {
      let user = await User.findByIdAndUpdate(
        _id,
        {
          $pull: { wishlist: prodId },
        },
        {
          new: true,
        }
      );
      res.json(user);
    } else {
      let user = await User.findByIdAndUpdate(
        _id,
        {
          $push: { wishlist: prodId },
        },
        {
          new: true,
        }
      );
      res.json(user);
    }
  } catch (error) {
    throw new Error(error);
  }
});

const rating = asyncHandler(async (req, res) => {
  const { _id } = req.user;
  const { star, prodId, comment } = req.body;
  try {
    const product = await Product.findById(prodId);
    let alreadyRated = product.ratings.find(
      (userId) => userId.postedby.toString() === _id.toString()
    );
    if (alreadyRated) {
      const updateRating = await Product.updateOne(
        {
          ratings: { $elemMatch: alreadyRated },
        },
        {
          $set: { "ratings.$.star": star, "ratings.$.comment": comment },
        },
        {
          new: true,
        }
      );
    } else {
      const rateProduct = await Product.findByIdAndUpdate(
        prodId,
        {
          $push: {
            ratings: {
              star: star,
              comment: comment,
              postedby: _id,
            },
          },
        },
        {
          new: true,
        }
      );
    }
    const getallratings = await Product.findById(prodId);
    let totalRating = getallratings.ratings.length;
    let ratingsum = getallratings.ratings
      .map((item) => item.star)
      .reduce((prev, curr) => prev + curr, 0);
    let actualRating = Math.round(ratingsum / totalRating);
    let finalproduct = await Product.findByIdAndUpdate(
      prodId,
      {
        totalrating: actualRating,
      },
      { new: true }
    );
    res.json(finalproduct);
  } catch (error) {
    throw new Error(error);
  }
});
```

```
module.exports = {
  createProduct,
  getaProduct,
  getAllProduct,
  updateProduct,
  deleteProduct,
  addToWishlist,
  rating,
};
```

CHAPTER 6

TESTING

6.1 TEST CASE-1

Test Case: User Registration

- Verify that users can successfully register on the website.
- Navigate to the website.
- Click on the "Sign Up" or "Register" button.
- Fill in the required fields with valid information.

Test Case: Product Browsing

- Log in to the website.
- Navigate to the "Products" or "Shop" section.
- Verify that the products are displayed correctly with relevant information (name, price, image, etc.).
- Perform a search for a specific product.
- Verify that the search results are accurate and relevant.
- Click on a product to view its details.
- Verify that the product details (description, specifications, etc.) are displayed correctly.

Test Case: Add to Cart

- Log in to the website.
- Browse or search for a product.
- Click on the product to view its details.
- Click on the "Add to Cart" button.
- Verify that the product is added to the cart.
- Repeat steps 2-5 for multiple products.
- Verify that all the selected products are correctly added to the cart.

6.2 TEST CASE-2

Test Case: User Account Management

- Log in to the website.
- Navigate to the user account or profile section.
- Verify that the user's information is displayed correctly.
- Edit the user's information (name, email, address, etc.).

- Save the changes and verify that the information is updated.
- Change the user's password.
- Verify that the password is changed successfully.

Test Case: Checkout and Payment

- Log in to the website.
- Add products to the cart.
- Navigate to the shopping cart or checkout page.
- Verify that the order summary is accurate.
- Enter the shipping address and select a payment method.
- Submit the order.
- Verify that the order is placed successfully.
- Complete the payment process using a valid payment method.
- Verify that the payment is processed correctly and

CHAPTER 7

MAINTENANCE

Maintaining an Acute Eyewear website is crucial for its smooth functioning and providing a seamless shopping experience to customers. Effective maintenance ensures that the website remains secure, up-to-date, and user-friendly. In this article, we will discuss the importance of website maintenance and highlight key areas that require attention.

First and foremost, security is paramount for any online store. Regular maintenance helps protect customer data, prevents hacking attempts, and ensures a secure environment for online transactions. Implementing security measures like SSL certificates, encryption protocols, and firewalls is essential. Regular monitoring of the website for vulnerabilities and promptly applying security patches and updates is crucial to mitigate potential risks.

Another important aspect of website maintenance is keeping the software and plugins up-to-date. Content management systems (CMS) and e-commerce platforms release regular updates to improve functionality, fix bugs, and enhance security. By applying these updates, you ensure that your website remains compatible with the latest technologies and features. Outdated software can pose security risks and lead to compatibility issues, hampering the overall user experience.

Optimizing website performance is also a significant part of maintenance. Slow-loading pages can frustrate visitors and drive them away. Regularly monitoring and optimizing website speed, image compression, and caching mechanisms can significantly improve the loading time. Compressing images, minifying CSS and JavaScript files, and utilizing a content delivery network (CDN) are common techniques to enhance performance.

The website's design and user experience should be regularly reviewed and updated. Aesthetically pleasing visuals, intuitive navigation, and a responsive design that adapts to different devices are vital for user engagement. Conducting periodic user testing and collecting feedback can help identify areas for improvement and enhance the overall user experience.

CHAPTER 8

LITERATURE REVIEW

A literature review on Acute Eyewear websites reveals several key insights and trends in the field. Scholars and researchers have explored various aspects, including consumer behavior, website design, customer satisfaction, and the impact of technology on the optical retail industry. Here is a summary of the literature findings:

- Consumer Behavior: Several studies have examined consumer behavior in the context of online optical stores. These studies have highlighted factors influencing online purchase decisions, such as price, product variety, convenience, trust, and customer reviews. Consumers tend to value a seamless browsing experience, easy navigation, and detailed product information. Additionally, personalized recommendations and virtual try-on features have shown to positively impact purchase intention.

- Website Design: The design and usability of online optical store websites have been the subject of extensive research. Studies have emphasized the importance of clear and intuitive interfaces, visually appealing layouts, and efficient search functionalities. Responsive web design, ensuring compatibility across different devices, has been highlighted as essential to accommodate the increasing use of mobile devices for online shopping.

- Customer Satisfaction: Research has focused on identifying the determinants of customer satisfaction in the context of online optical stores. Factors such as website performance, order processing efficiency, responsive customer service, accurate product descriptions, and secure payment processes have been identified as critical factors impacting customer satisfaction. Positive customer experiences and satisfaction contribute to customer loyalty and repeat purchases.

- Technology Impact: The literature emphasizes the impact of technology on the optical retail industry and the emergence of innovative solutions. Virtual reality (VR) and augmented reality (AR) technologies have gained attention for enhancing the online eyewear shopping experience by allowing customers to virtually try on glasses and see how they look on their faces. Artificial intelligence (AI) and machine learning algorithms have been utilized for personalized product recommendations, enhancing customer engagement, and improving conversion rates.

- Competitive Advantage: Online optical store websites offer a competitive advantage to traditional brick-and-mortar stores. The ability to offer a wider range of products, competitive pricing, and convenience of shopping from home contributes to attracting customers. The literature highlights the importance of strategic differentiation, customer-centric approaches, and effective marketing strategies to gain a competitive edge in the online optical retail market.

- Challenges and Opportunities: The literature acknowledges several challenges faced by online optical stores. These include ensuring data security and privacy, managing inventory and logistics efficiently, maintaining high website performance, and addressing customer concerns regarding accurate prescription requirements. However, the studies also highlight the opportunities that arise from embracing technology, such as expanding market reach, targeting niche customer segments, and offering personalized shopping experiences.

Overall, the literature review reveals a growing interest in online optical store websites. It highlights the importance of understanding consumer behavior, optimizing website design, ensuring customer satisfaction, and leveraging technology to gain a competitive advantage. The findings provide valuable insights for practitioners and researchers alike, guiding the development and implementation of successful online optical store websites.

CONCLUSION

In conclusion, the establishment of an Acute Eyewear website presents a compelling opportunity to serve a diverse target audience seeking convenient access to a wide range of eyewear products. By considering various factors such as technical feasibility, operational feasibility, and behavioral feasibility, the website can be developed and implemented successfully to meet customer needs and drive business growth.

Technical feasibility is crucial in ensuring that the website is developed with the necessary infrastructure and features. This includes creating a user-friendly interface with intuitive navigation, incorporating product catalogs and shopping carts, and implementing secure payment gateways to facilitate seamless transactions. Scalability and mobile responsiveness are also key considerations to accommodate increasing traffic and provide a consistent user experience across different devices.

Operational feasibility focuses on the practicality of managing the online store's day-to-day operations. Efficient product management systems should be in place to update inventory, prices, and product information. Seamless order processing mechanisms must be implemented, including order confirmations, shipment tracking, and returns handling. Additionally, customer support services need to be available to address inquiries and provide assistance promptly. Integration with logistics partners and secure payment gateways is essential to ensure smooth and reliable fulfillment of orders.

Behavioral feasibility is crucial for the successful adoption and acceptance of the online optical store website by the target audience. User experience is a significant aspect, and the website should provide a visually appealing and user-friendly interface. Conducting user testing and gathering feedback can help refine the website's design and functionality. Customer education resources, such as guides on product options and lens prescriptions, can enhance the user experience and ensure customer satisfaction. Building trust and ensuring security measures are in place are vital to gain customer confidence in making online transactions. Reviews and ratings from customers contribute to the credibility of the website and help potential customers make informed decisions.

Engaging key stakeholders, such as optometrists, eyewear manufacturers, and industry partners, is important to establish credibility and foster collaboration. Their support can contribute to promoting the online optical store and expanding its reach. Regular communication and collaboration can lead to valuable partnerships and potential opportunities for growth.

In conclusion, an Acute Eyewear website has the potential to meet the needs and preferences of customers seeking a convenient and accessible platform to purchase eyewear products. By addressing technical, operational, and behavioral feasibility, the website can provide a seamless user experience, efficient operations, and customer satisfaction. Continuous monitoring and improvement are crucial to adapt to evolving customer expectations and technological advancements. With effective planning, execution, and continuous enhancement, an online optical store website can establish a strong online presence, attract a loyal customer base, and drive business success in the competitive eyewear market

.

BIBLIOGRAPHY

References for the Project Development were taken from the following Web Sites.

[1] https://www.w3schools.com

[2] http://developeíscoíneí.in/

[3] www.arno.unimass.nl/show.cgi?fid:2720

[4] Bootstrap · The most popular HTML, CSS, and JS library in the world. (getbootstrap.com)