Timeless Haven

A PROJECT REPORT
for
Mini Project (KCA353)
Session (2024-25)

Submitted by
Lakshay Goel
University Roll No 2300290140094
Rajat Sinha
University Roll No 2300290140133
Mohammad Aamir
University Roll No 2300290140103
Preet Gupta
University Roll No 2300290140127

Submitted in partial fulfilment of the
Requirements for the Degree of

MASTER OF COMPUTER APPLICATION

Under the Supervision of
Dr. Neelam Rawat
(Associate Professor)

Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206

December 2024

CERTIFICATE

Certified that **Lakshay Goel (2300290140094), Rajat Sinha (2300290140133), Mohammad Aamir (2300290140103), Preet Gupta (2300290140127)** have carried out the project work having "**Timeless Haven**" (**Mini Project KCA353**) for **Master of Computer Application** from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself 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:

Lakshay Goel (2300290140094)
Rajat Sinhsa (2300290140133)
Mohammad Aamir(2300290140094)
Preet Gupta (2300290140127)

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

Date:

Dr. Neelam Rawat	**Dr. Arun Tripathi**
Associate Professor	**Head**
Department of Computer Applications	**Department of Computer Applications**
KIET Group of Institutions, Ghaziabad	**KIET Group of Institutions, Ghaziabad**

ABSTRACT

The Timeless Haven website, built on the React framework, is a dynamic platform designed to immerse users in a world of E - commerce. This abstract encapsulates the essence of the website

Timeless Haven is a sophisticated online platform that brings luxury and comfort into your living spaces through an exquisite range of designer bedsheets, blankets, etc. Powered by the dynamic React framework, our website delivers an intuitive and elegant shopping experience tailored to meet your home decor needs. With a vast collection of premium designs, customers can effortlessly explore products that blend aesthetics with functionality, transforming houses into timeless havens.

Detailed product descriptions and high-resolution images ensure informed decisions, while real customer reviews provide trusted insights. Our seamless navigation and secure payment options make shopping a breeze, with dedicated customer support ready to assist every step of the way. Timeless Haven also encourages creativity and engagement through style guides and decor tips, inspiring customers to create personalized spaces. Join us in redefining home luxury, where every visit brings a touch of elegance and every purchase becomes part of your story.

ACKNOWLEDGEMENTS

Success in life is never attained single-handedly. My deepest gratitude goes to my project supervisor, **Dr. Neelam Rawat** for her guidance, help, and encouragement throughout my project work. Their enlightening ideas, comments, and suggestions.

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

Lakshay Goel

Rajat Sinha

Mohammad Aamir

Preet Gupta

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background

Timeless Haven serves as a comprehensive platform addressing various challenges faced by customers in creating comfortable and stylish living spaces. It simplifies home shopping with tools for product discovery and personalized recommendations, ensuring an effortless and enjoyable experience. Our website prioritizes quality and trust, offering reliable customer reviews and secure payment options to provide peace of mind. With dedicated support services and a curated selection of designer bedsheets, blankets, and decor essentials, Timeless Haven enhances the shopping journey, making it more convenient, inspiring, and satisfying for customers looking to elevate their homes.

1.2 Project Overview

Timeless Haven is a MERN-based web application transforming the way customers shop for home furnishings. It features seamless user authentication, dynamic product showcases, real-time inventory updates, secure payment integration, personalized product recommendations, and a responsive design for enhanced accessibility across devices. With its innovative technology and customer-centric approach, Timeless Haven simplifies online shopping while providing a premium platform for users to discover and purchase exquisite designer bedsheets, blankets, and decor essentials.

1.3 Objective

The primary objectives of a e commerce website include:

Efficiency: Streamlining the discovery, browsing, and purchasing processes for home decor essentials.

User Convenience: Providing customers with a seamless and intuitive interface to explore designer bedsheets, blankets, and other home furnishings, ensuring a hassle-free shopping experience.

4o

.

1.4 Key Features

Timeless Haven include a number of features

Shopping Interface: Offers customers an intuitive interface to browse and purchase designer bedsheets, blankets, and other home furnishings. Users can filter products by category, material, size, and style to streamline the shopping experience.

Dynamic Product Management: Admins can manage inventory effortlessly, adding or updating product details such as images, descriptions, and pricing to keep the catalog fresh and accurate.

Personalized Recommendations: Leverages user preferences and browsing history to suggest products tailored to individual tastes, enhancing customer satisfaction.

1.5 Scope of the project

The scope of Timeless Haven extends to the following:

UserInterface

Developing an intuitive and visually appealing interface that allows users to seamlessly browse and explore a wide range of designer bedsheets, blankets, and home decor items. The interface will prioritize ease of navigation and accessibility across all devices.

ProductDiscoveryandPersonalization:

Incorporating advanced filters and search functionality to help users find products based on material, size, color, and design. Personalized product recommendations will enhance the shopping experience by catering to individual preferences.

SecurePaymentandCheckout:

Implementing a secure and efficient payment system that supports various payment options, including credit/debit cards and QR code payments, ensuring a safe and user-friendly checkout process.

CHAPTER 2

PROBLEM IDENTIFICATION & FEASIBILITY STUDY

2.1 Problem Identification

Identifying potential problems in a e commerce website is crucial for ensuring a smooth user experience. Common issues include poor user interface and experience, limited search and filtering options, incomplete or inaccurate information, booking and payment issues, ineffective customer support, poor mobile responsiveness, limited personalization, and security concerns. Addressing these issues through regular user testing, feedback gathering, and continuous improvement efforts can enhance the overall user satisfaction and usability of the website.

2.2 Feasibility Study

Before initiating the development of the e commerce website, a thorough feasibility study was conducted to evaluate the feasibility and practicality of the proposed platform. This study examined technical, operational, and economic aspects to ensure the viability of the project.

2.2.1 Technical Feasibility

Assess the technical requirements and capabilities needed to develop and maintain the website. Consider factors such as web hosting, scalability, security measures, and integration with third-party services like booking systems and payment gateways.

2.2.2 Operational Feasibility

Assess the practicality of managing and operating the website on a day-to-day basis. Consider factors such as staffing requirements, content management processes, customer support mechanisms, and strategic partnerships with travel suppliers and service providers

2.2.3 Economic Feasibility

The economic study of a e commerce website involves assessing its financial viability and sustainability through various analyses. This includes evaluating costs associated with development, maintenance, and operations, as well as identifying potential revenue streams such as booking commissions, advertising, and subscription fees. Market research is conducted to understand demand, competition, and growth opportunities, while financial projections and ROI calculations provide insights into the website's potential performance. Risk assessment and sensitivity analysis help identify and mitigate potential challenges, ensuring the website's resilience in fluctuating market conditions. Overall, the economic study informs stakeholders about the feasibility and profitability of the traveling website, guiding strategic decisions for its successful implementation and operation.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 Introduction

Timeless Haven is a premier online destination for luxurious home furnishings, offering a curated selection of designer bedsheets, blankets, and decor essentials. Combining style and functionality, the platform provides an effortless shopping experience with personalized recommendations, secure payments, and detailed product insights. With a focus on quality and elegance, Timeless Haven empowers customers to transform their living spaces into timeless retreats.

3.2 Functional Requirements

3.2.1 User Module

Allow users to create accounts by providing necessary details such as name, email, and password. Implement authentication mechanisms to ensure secure access to user accounts.

3.2.2 Administrator Module

Creating a package for user administration from source to destination involves organizing the necessary components and functionalities to manage users' details. Here's a detailed breakdown of the process:

3.2.3 Package Module

The package module of a e commerce website refers to a structured organization of components, functionalities, and dependencies related to specific features or modules within the website. Here's an outline of how you might structure the package module for a traveling website:.

3.3 NON-FUNCTIONAL REQUIREMENTS

3.3.1 PERFORMANCE

Response Time: Ensure that pages load quickly to provide a seamless user experience, especially during peak traffic times.

Scalability: The website should be able to handle increasing user loads without significant degradation in performance.

Reliability: Minimize downtime and ensure high availability to prevent disruptions in service.

3.3.2 SECURITY

Data Encryption: Implement secure communication protocols (e.g., HTTPS) toencrypt data transmission between the user's browser and the server.

Authentication and Authorization: Ensure secure user authentication mechanisms and role-based access control to protect user accounts and sensitive information.

Data Protection: Adhere to data protection regulations such as GDPR (General Data Protection Regulation) to safeguard user data and privacy.

3.3.3 SCALABILITY

Horizontal Scalability: Design the website architecture to scale horizontally by adding more servers or resources to handle increased traffic.

Vertical Scalability: Ensure that the website can scale vertically by optimizing code, databases, and server configurations to handle increased load on existing resources.

3.3.4 USABILITY

Intuitive Interface: The website should be easy to navigate, with clear menus, intuitive search functionalities, and user-friendly design elements.

Accessibility: Ensure that the website is accessible to users with disabilities, complying with accessibility standards such as WCAG (Web Content Accessibility Guidelines).

Multilingual Support: Provide support for multiple languages to cater to a diverse user base.

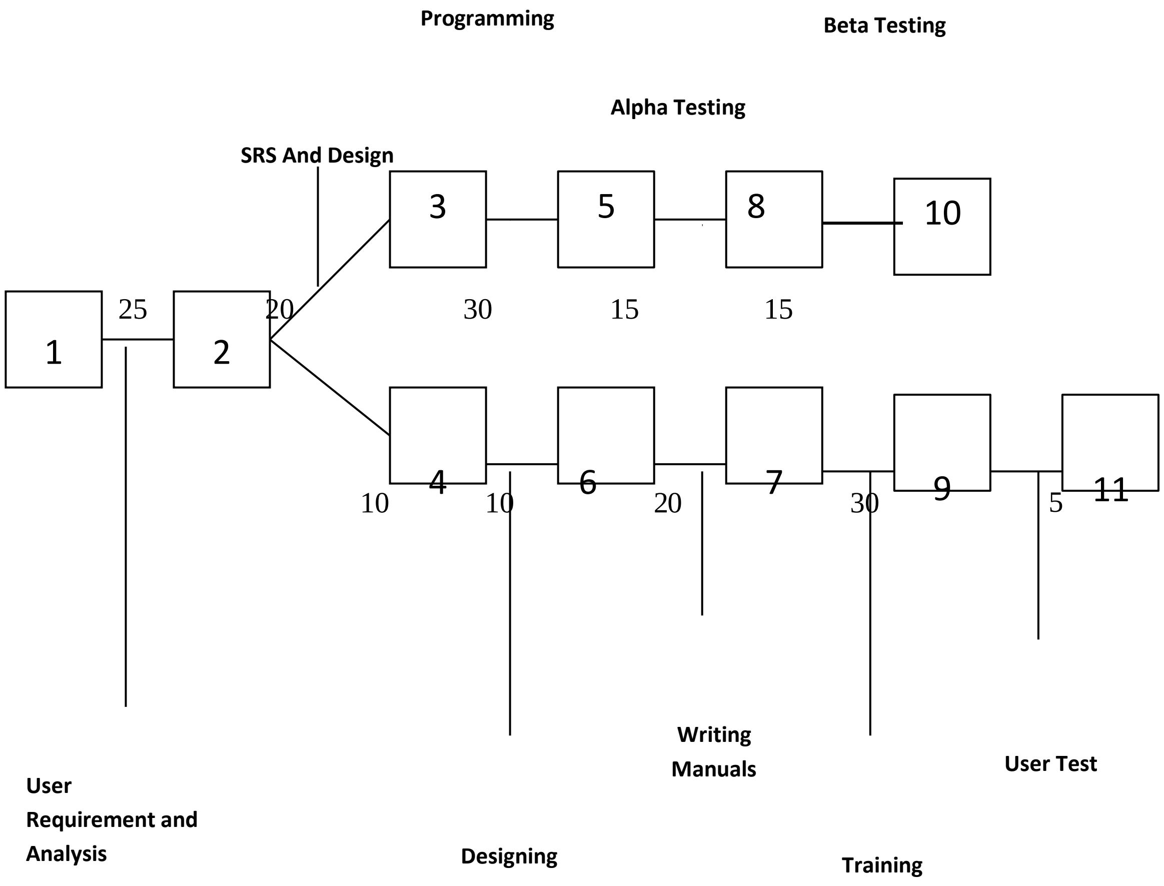
CHAPTER 4

PROJECT PLANNING AND SCHEDULING

4.1 Pert Chart:

A PERT chart is a project management tools used to schedule, organize, and coordinate tasks within a project. PERT stands for Program Evaluation Review Technique. A PERT chart presents a graphic illustration of a project as network diagram consisting of numbered nodes (either circles or rectangles) representing events, ormilestones in the project linked by labelled vectors (directional lines) representing tasks in the project.

The direction of the arrows on the lines indicates the sequence of tasks.



CHAPTER 5

HARDWARE & SOFTWARE SPECIFICATION

5.1 Hardware Specification

The Timeless Haven will be developed and deployed on a hardware infrastructure that ensures optimal performance and reliability. The recommended hardware specifications are as follows:

Server:

Processor: Intel Core i5 or equivalent

RAM: 8 GB or higher

Storage: 256 GB SSD or higher

Database Server:

Processor: Intel Core i5 or equivalent

RAM: 8 GB or higher

Storage: 256 GB SSD or higher

Network Interface: Gigabit Ethernet

Client Machines:

Processor: Intel Core i3 or equivalent

RAM: 4 GB or higher

Storage: 128 GB SSD or higher

Network Interface: 100 Mbps Ethernet or Wi-Fi

5.2 Software Specification

The E commerce website's Timeless Haven will be developed using a combination of server-side and client-side technologies within the MERN (MongoDB, Express.js, React.js, Node.js) stack. The software specifications for the E commerce website encompass:

Server-Side Technologies:

Operating System: Windows Server 2016 or later

Web Server: Node.js

Database Management System: MongoDB

Server-Side Scripting Language: JavaScript (Node.js)

Client-Side Technologies:

Web Browser: Latest versions of Chrome, Firefox, Safari, or Edge

Client-Side Scripting: JavaScript

Development Tool

Integrated Development Environment (IDE): Visual Studio Code

Version Control:

Git: Version control for collaborative development

Security:

SSL/TLS: Ensure secure data transmission over the network

Firewall: Implement firewall rules to restrict unauthorized access

Anti-malware Software: Regularly updated anti-malware software on server and client machines

CHAPTER 6

CHOICE OF TOOLS & TECHNOLOGY

6.1 React

React is a JavaScript library used for building user interfaces. React allows developers to describe the UI's appearance at any point in time, and it automatically updates and renders the right components when the data changes.

React organizes the UI into reusable components, making it easier to manageand maintain complex applications. React uses a virtual DOM to optimize and update the actual DOM efficiently, improving performance by minimizing unnecessary re-rendering.

React uses JSX, a syntax extension that looks similar to XML or HTML, to describe the structure of UI components in a more concise and readable way**.** React components can manage their internal state, and data can be passed to components through props,allowing for dynamic and interactive UIs

6.2 MongoDB

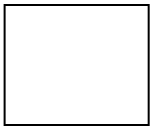
MongoDB is a NoSQL (non-relational) database management system that stores data in a flexible, JSON-like format known as BSON (Binary JSON). It is designed to handle large volumes of unstructured or semi-structured data, making it particularly suitable for applications with evolving and dynamic data requirements.

6.3 Data Flow Diagram

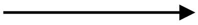
The data flow diagram shows the flow of data within any system. It is an importanttool for designing phase of software engineering. Larry Constantine first developed it. It represents graphical view of flow of data. It’s also known as BUBBLE CHART. The purpose of DFD is major transformation that will become in system design symbols used in DFD: -

In the DFD, four symbols are used and they are as follows.

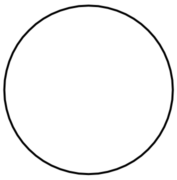
1. A square defines a source (originator) or destination of system data.



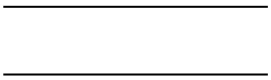
2. An arrow identifies data flow-data in motion. It is 2a pipeline through which information flows.



3. A circle or a “bubble “(Some people use an oval bubble) represents a process that transfers informing data flows into outgoing data flows.

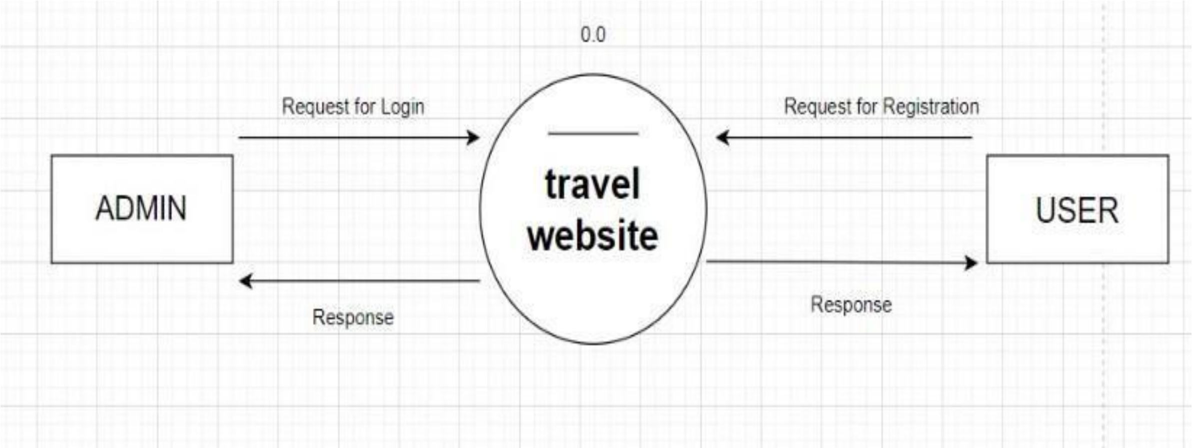


4.An open rectangle is a data store-data at rest, or a temporary repository of data.



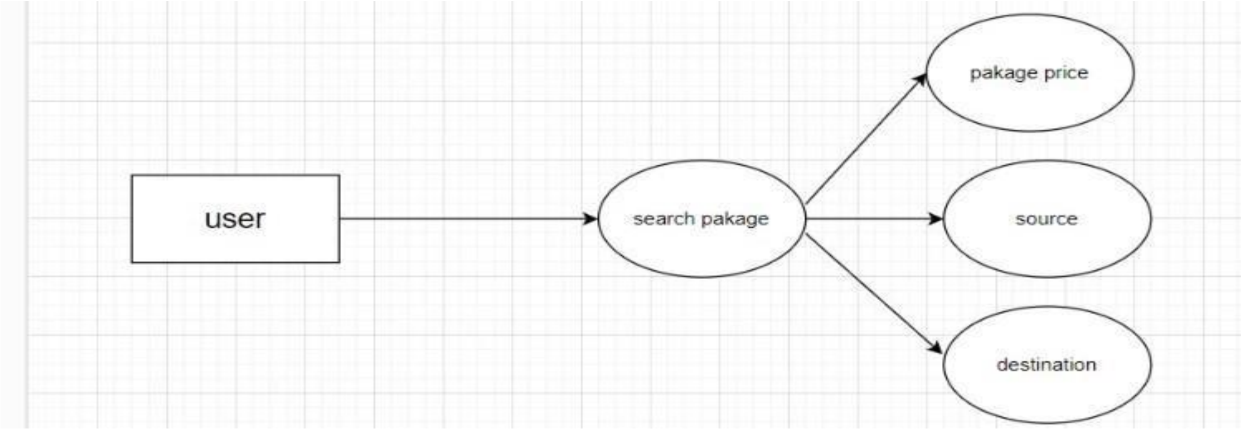
6.4 Context Level Diagram

This level shows the overall context of the system and its operating environment and shows the whole system as just one process. Timeless Haven isshown as one process in the context diagram; which is also known as zero level DFD, shown below. The context diagram plays important role in understandingthe system and determining the boundaries. The main process can be broken into sub-processes and system can be studied with more detail; this is where 1st level DFD comes into play.



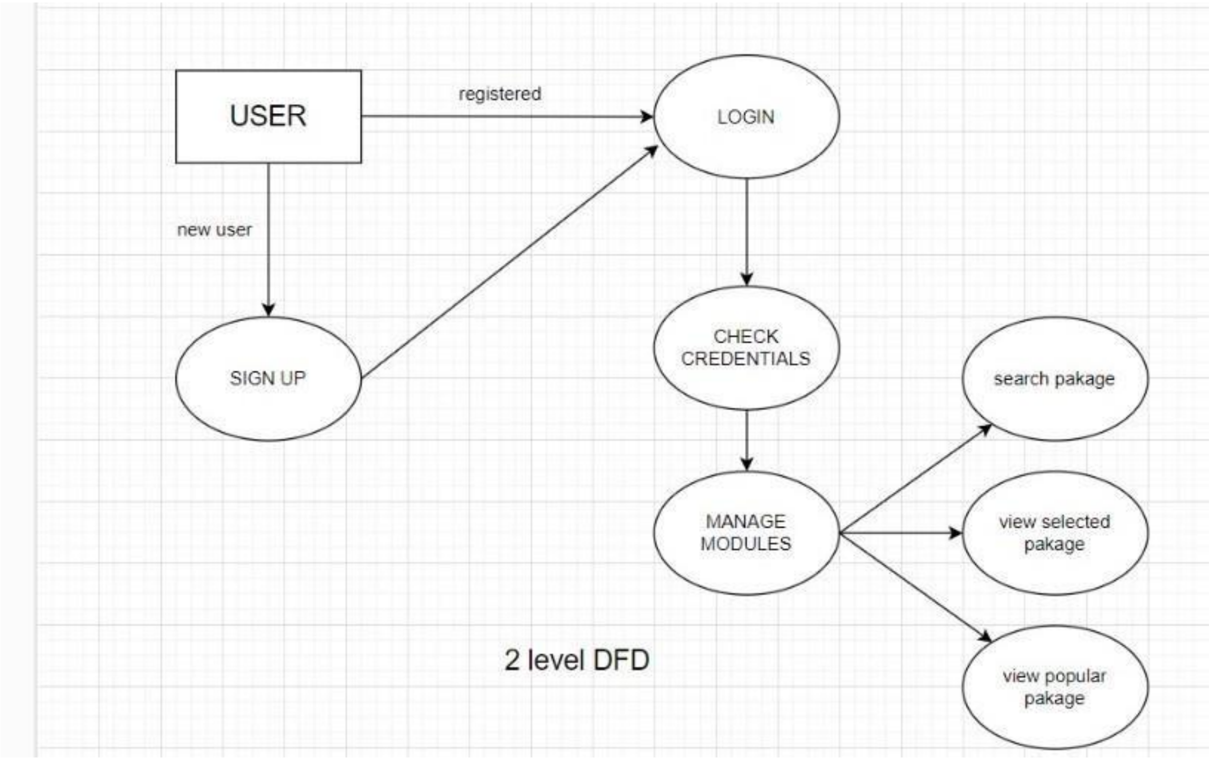
0 Level DFD

Fig 6.1



1st Level DFD

Fig 6.2



2nd Level DFD

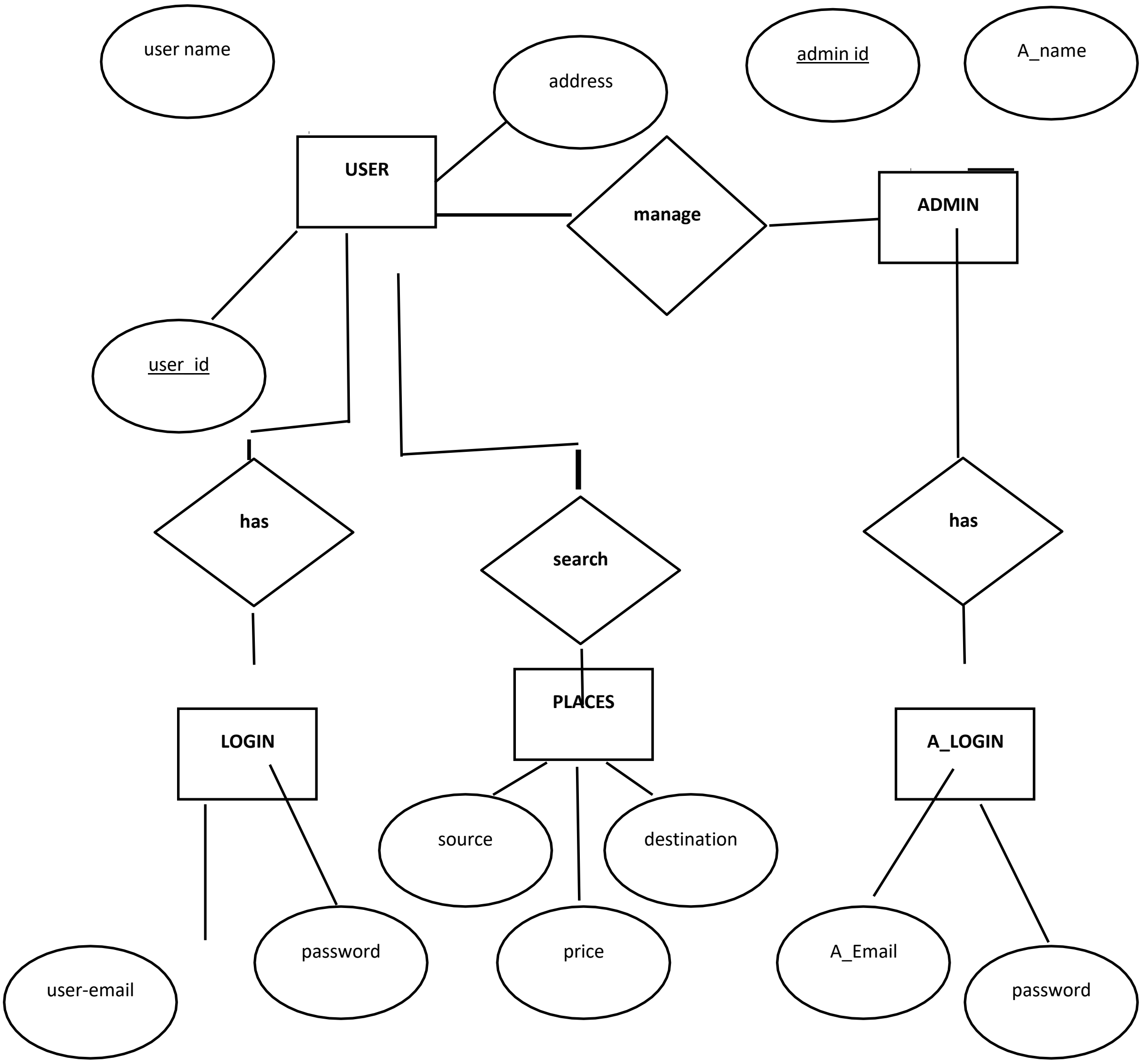
Fig 6.3

CHAPTER 7

ER-DIAGRAM

7.1 Entity-relationship model: -

The entity-relationship model or entity-relationship diagram (ERD) is a data model or diagram for high-level descriptions of conceptual data model, and it provides a graphical notation for representing such data models in the form of entity- relationship diagrams.



7.2 Class Diagram: -

Authentication:

- Classification: Weak Class
- Description: Represents user authentication details, including username and password. This class is responsible for user login functionality.

User:

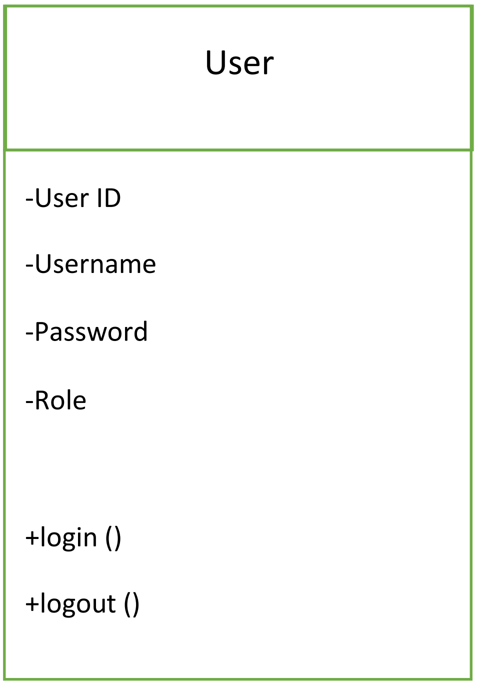
- Classification: Strong Class
- Description: Represents the users of the system, including administrators and employees.

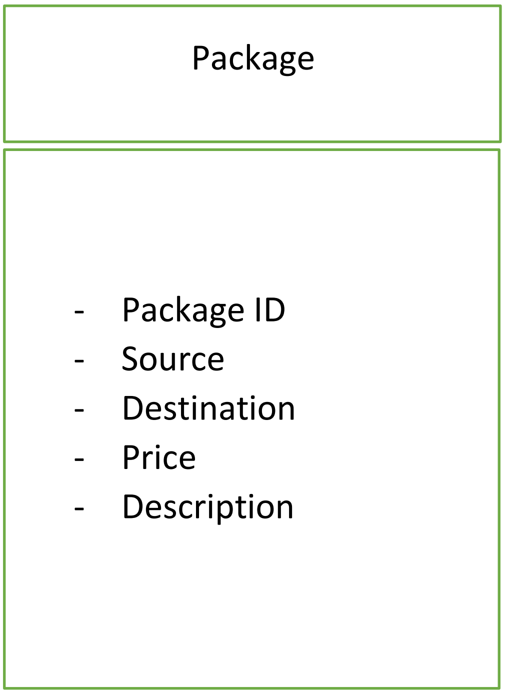
Package:

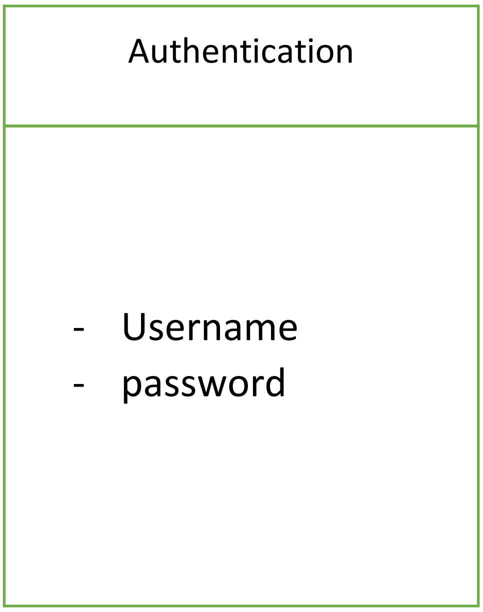
- Classification: Strong Class
- Description: Represents the package source and destinations including details such as Package ID ,source and destination price and description.

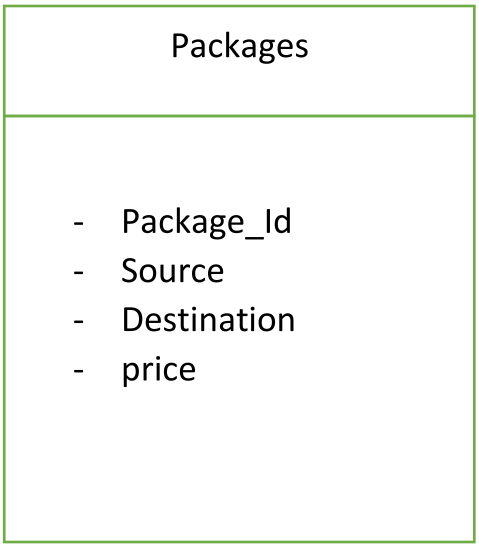
Packages:

- Classification: Strong Class
- Description: Represents the packages from source to destination in the package cart, including details such as package ID, Description and price.


User
-User ID
-Username
-Password
-Role
+login ()
+logout ()


Package
- Package ID
- Source
- Destination
- Price
- Description

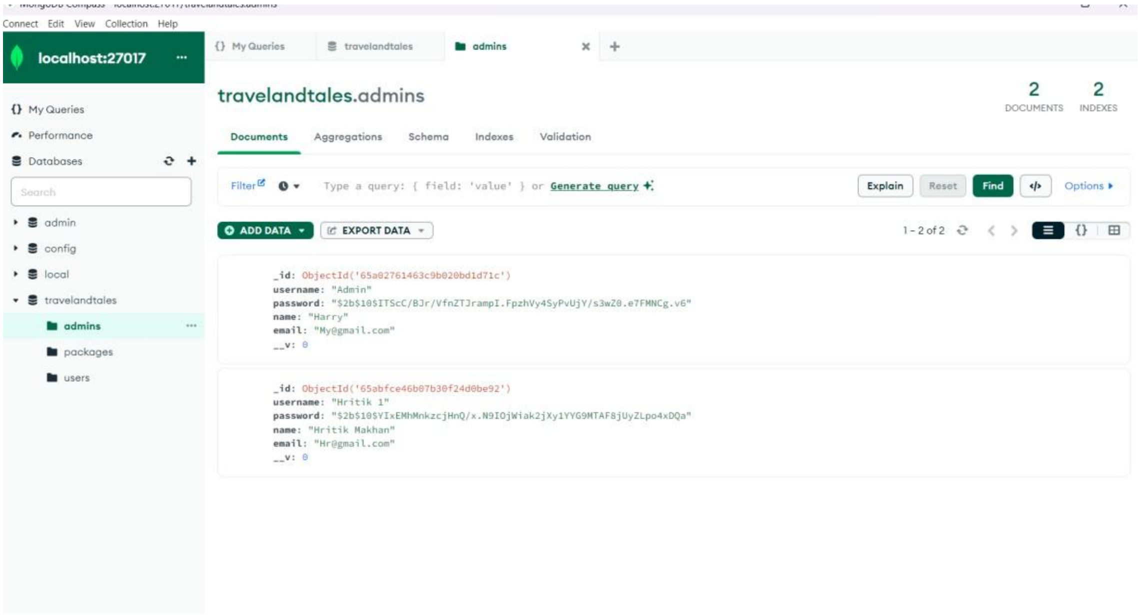

Authentication
- Username
- password


Packages
- Package_Id
- Source
- Destination
- price

CHAPTER 8

DATABASE

8.1 Admin



```
_id: ObjectId('65a02761463c9b020bd1d71c')
username: "Admin"
password: "$2b$10$ITScC/BJr/VfnZTJrampI.FpzhVy4SyPvUjY/s3wZ0.e7FMNCg.v6"
name: "Harry"
email: "My@gmail.com"
__v: 0
```

```
_id: ObjectId('65abfce46b07b30f24d0be92')
username: "Hritik 1"
password: "$2b$10$YIxEMhMnkzcjHnQ/x.N9IOjWiak2jXy1YYG9MTAF8jUyZLpo4xDQa"
name: "Hritik Makhan"
email: "Hr@gmail.com"
__v: 0
```

Email:

This attribute stores the email address of an individual. It is a unique identifier and is commonly used for user authentication and communication.

Name:

The "name" attribute typically stores the full name of an individual. It might be divided into first name, middle name, and last name for more detailed records.

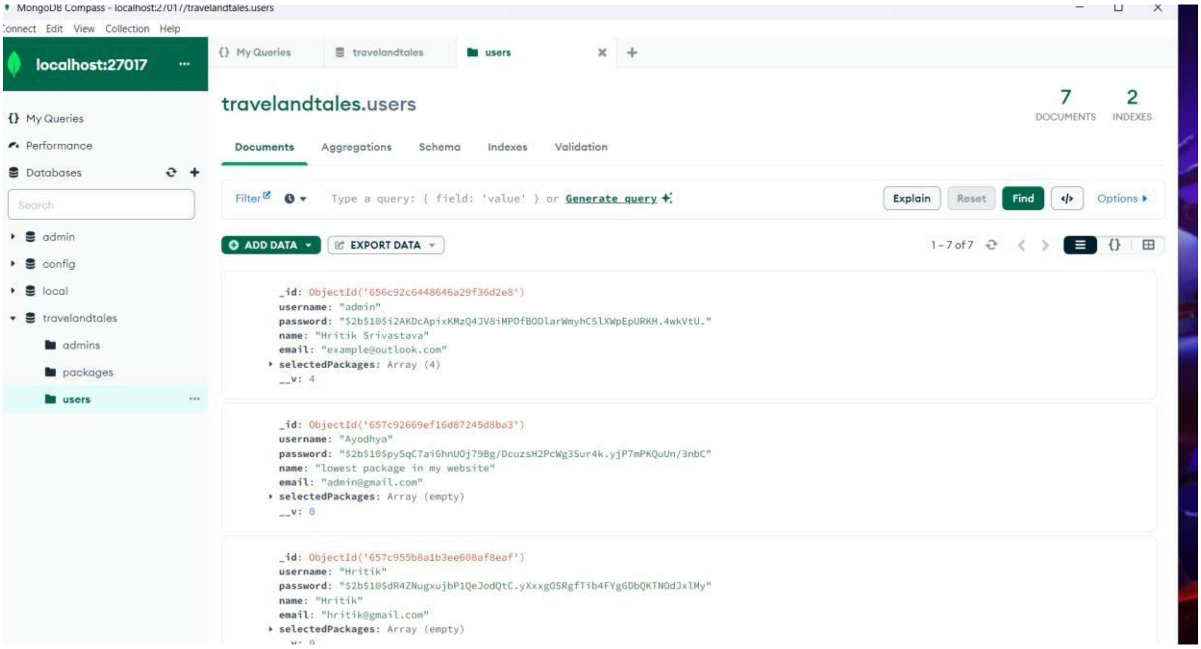
Address:

The "Address" attribute stores the physical address or location details of an individual. It could include components such as street address, city, state, and postal code..

Password:

The "password" attribute stores a securely hashed or encrypted version of the user's password. It is a critical attribute for user authentication, ensuring secure access tothe system.

8.2 User



```
_id: ObjectId('656c92c6448646a29f36d2e8')
username: "admin"
password: "$2b$10$i2AKDcApixKMzQ4JV8iMPOfBODlarWmyhC5lXWpEpURKH.4wkVtU."
name: "Hritik Srivastava"
email: "example@outlook.com"
selectedPackages: Array (4)
__v: 4

_id: ObjectId('657c92669ef16d87245d8ba3')
username: "Ayodhya"
password: "$2b$10$pySqC7aiGhnUOj79Bg/DcuzsH2PcWg3Sur4k.yjP7mPKQuUn/3nbC"
name: "lowest package in my website"
email: "admin@gmail.com"
selectedPackages: Array (empty)
__v: 0

_id: ObjectId('657c955b8a1b3ee608af8eaf')
username: "Hritik"
password: "$2b$10$dR4ZNugxujbP1QeJodQtC.yXxxgOSRgfTib4FYg6DbQKTNOdJxlMy"
name: "Hritik"
email: "hritik@gmail.com"
selectedPackages: Array (empty)
__v: 0
```

Users Credentials

Id:

The "Id" attribute is typically a unique identifier assigned to each individual in the database. It serves as a primary key, ensuring that each record can be uniquely identified and referenced.

Name:

The "Name" attribute stores the full name of an individual.

Email:

The "Email" attribute stores the email address of an individual. It serves as a unique identifier for user accounts and is commonly used for communication and login credentials.

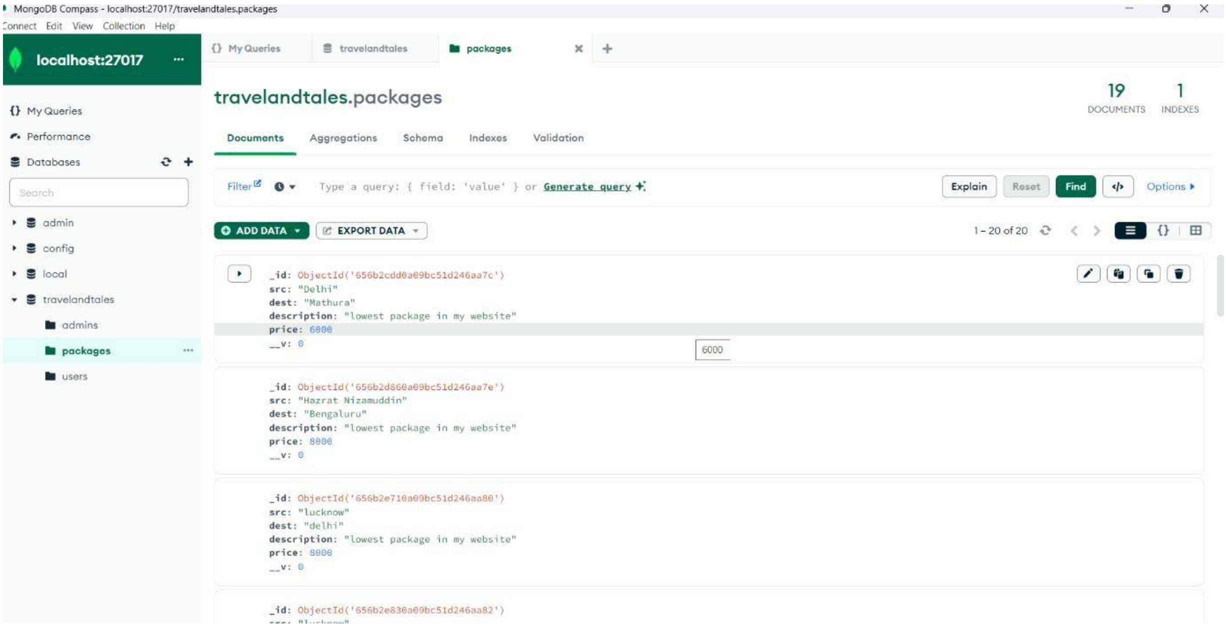
Address:

The "Address" attribute captures the physical address or location details of an individual. It might include components such as street address, city, and state, providing a comprehensive overview of an individual's residence.

Password:

The "Password" attribute stores a securely hashed or encrypted version of the user's password. It is a critical attribute for user authentication, ensuring the security of user accounts by protecting access to sensitive information.

8.3 Packages



```
_id: ObjectId('656b2cdd0a09bc51d246aa7c')
src: "Delhi"
dest: "Mathura"
description: "lowest package in my website"
price: 6000
__v: 0

_id: ObjectId('656b2d860a09bc51d246aa7e')
src: "Hazrat Nizamuddin"
dest: "Bengaluru"
description: "lowest package in my website"
price: 8000
__v: 0

_id: ObjectId('656b2e710a09bc51d246aa80')
src: "lucknow"
dest: "delhi"
description: "lowest package in my website"
price: 8000
__v: 0

_id: ObjectId('656b2e830a09bc51d246aa82')
```

Packages

.

Packages_Id:

The “Id” attribute typically serves as a unique identifier for each record in the database table. It is a primary key that ensures each entry can be uniquely identified andreferenced..

Description:

The “Description” attribute provides additional details or a brief description of the product. This field allows for a more comprehensive understanding of the product’s characteristics or features.

Src(Source):

The source refers to the origin or starting point of a project, including the initial problem statement, objectives, and the resources utilized for its implementation. Itencapsulates the background information, research findings, and any existing frameworks or methodologies that have contributed to the project's inception.

.

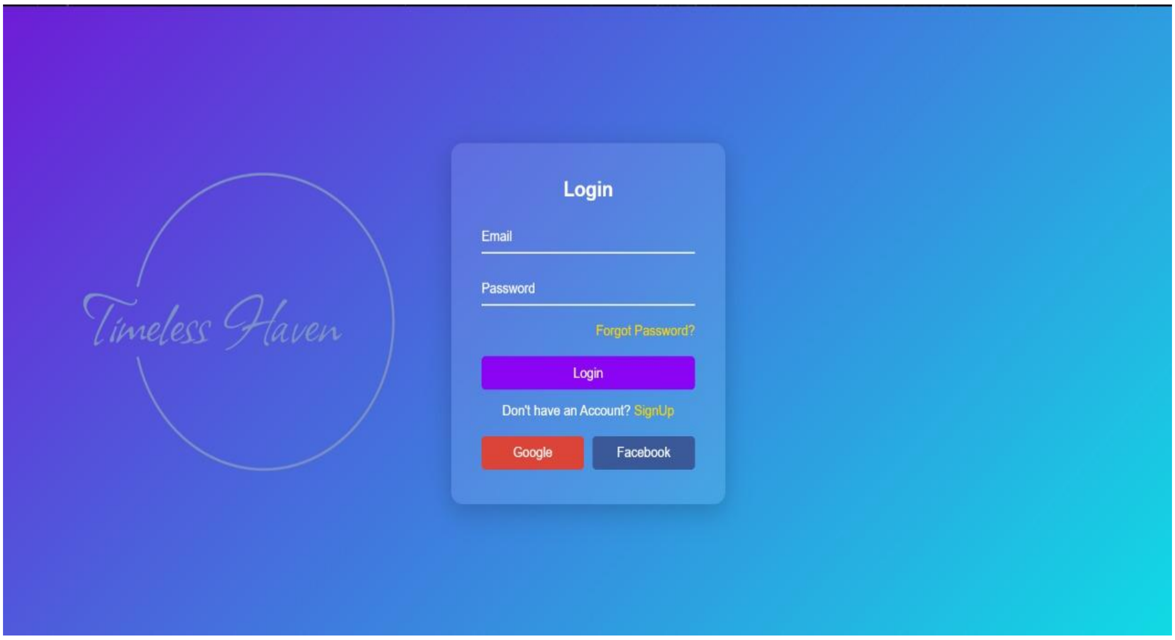
Desc (Destination):

The source refers to the origin or starting point of a project, including the initial problem statement, objectives, and the resources utilized for its implementation. Itencapsulates the background information, research findings, and any existing frameworks or methodologies that have contributed to the project's inception.

CHAPTER 9

FORM DESIGN

9.1 Login

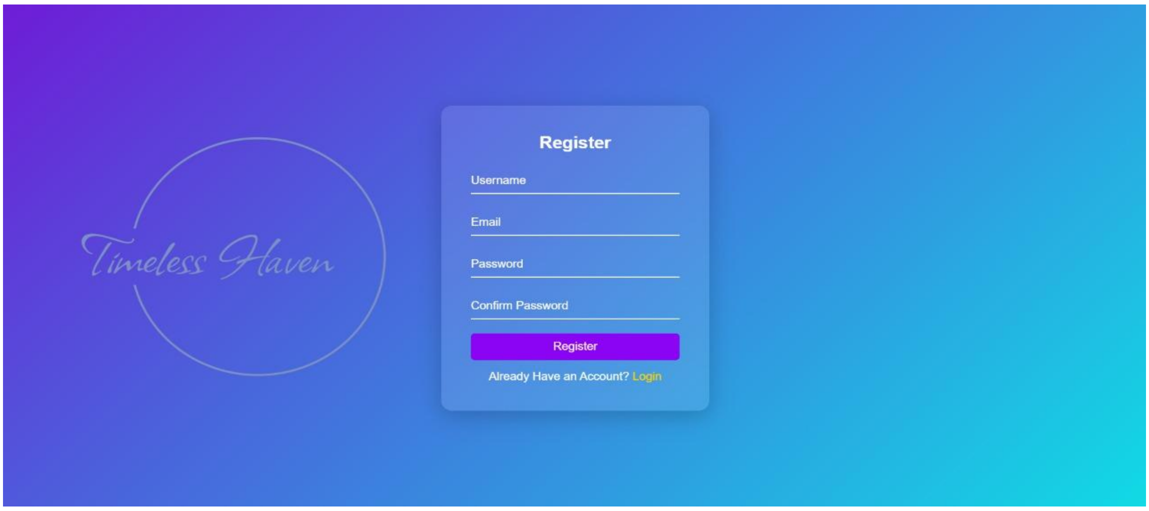


login page

The authentication module in the Timeless haven is a fundamental component tasked with securely verifying user identities and controlling access to the platform. This module validates user credentials, typically comprising a username and password, to authenticate users effectively. Upon successful authentication, the module assigns appropriate roles to users, distinguishing between administrators and regular users. Session management functionalities are employed to maintain user sessions throughout their interactions on the website.

The authentication module in the Timeless Haven website is a fundamental component tasked with securely verifying user identities and controlling access to the platform. This module validates user credentials, typically comprising a username and password, to authenticate users effectively. Upon successful authentication, the module assigns appropriate roles to users, distinguishing between administrators and regular users. Session management functionalities are employed to maintain user sessions throughout their interactions on the website

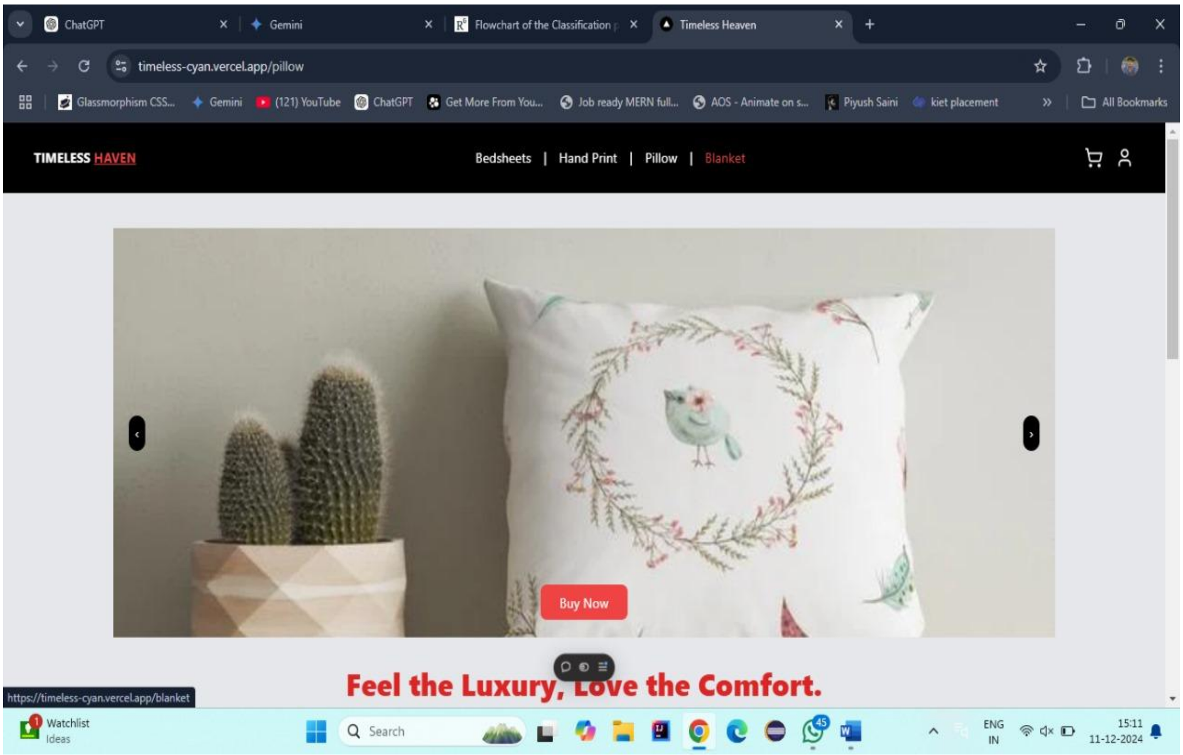
9.2 Signup



Signup Page

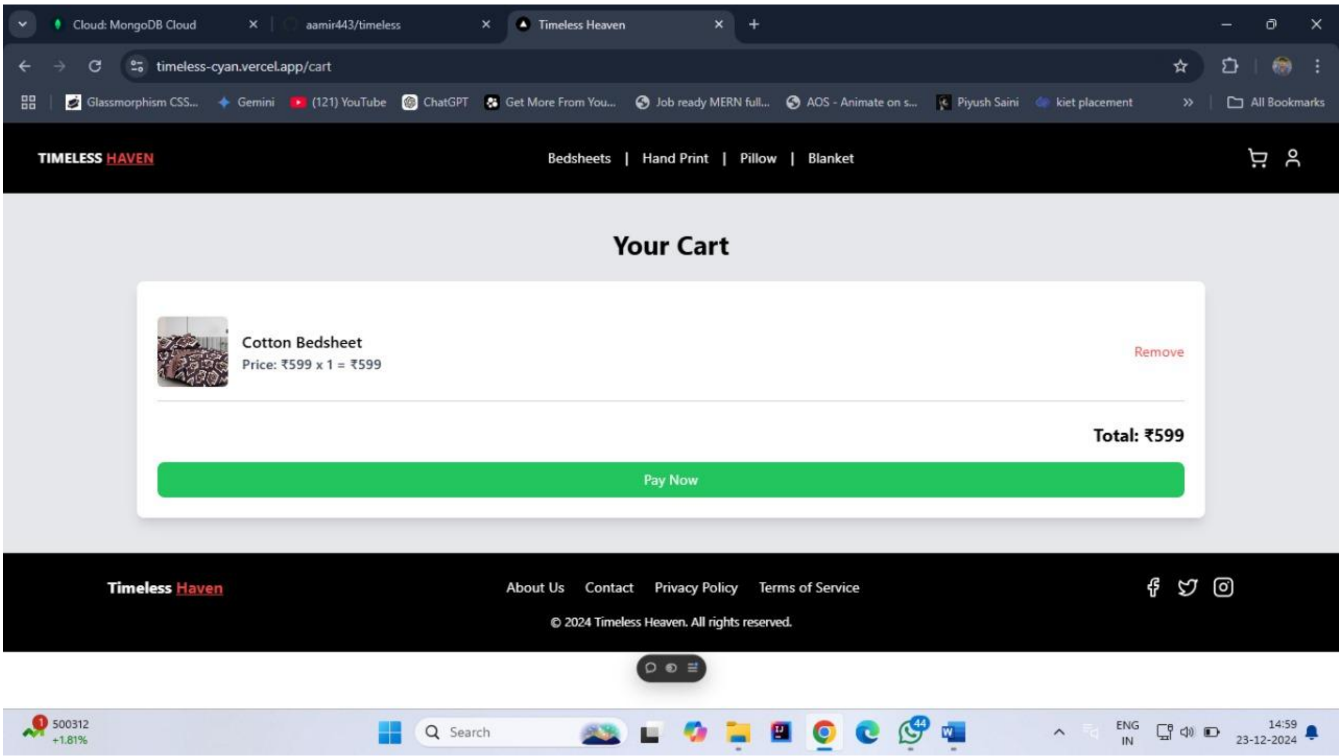
The signup page on Timeless haven facilitates easy registration for users eager to join our community. By providing basic details like name and email, visitors can swiftly create personalized accounts. Optional fields may capture interests, enriching user profiles. Our secure process includes email verification to ensure authenticity.

9.3 Users



The user module for "Timeless Haven" is designed to provide an intuitive and seamless experience for customers exploring the platform. It allows users to browse the collection of premium bedsheets effortlessly, filter products by material, size, color, and design, and view detailed descriptions with high-quality images. The module supports easy account creation and management, enabling users to save their preferences, view order history, and manage wish lists. It also includes a secure and straightforward checkout process, offering multiple payment options and order tracking. With a focus on user-friendliness, the module ensures customers can navigate the platform efficiently while enjoying a personalized and engaging shopping experience..

9.3.1 Items Add



Your Cart

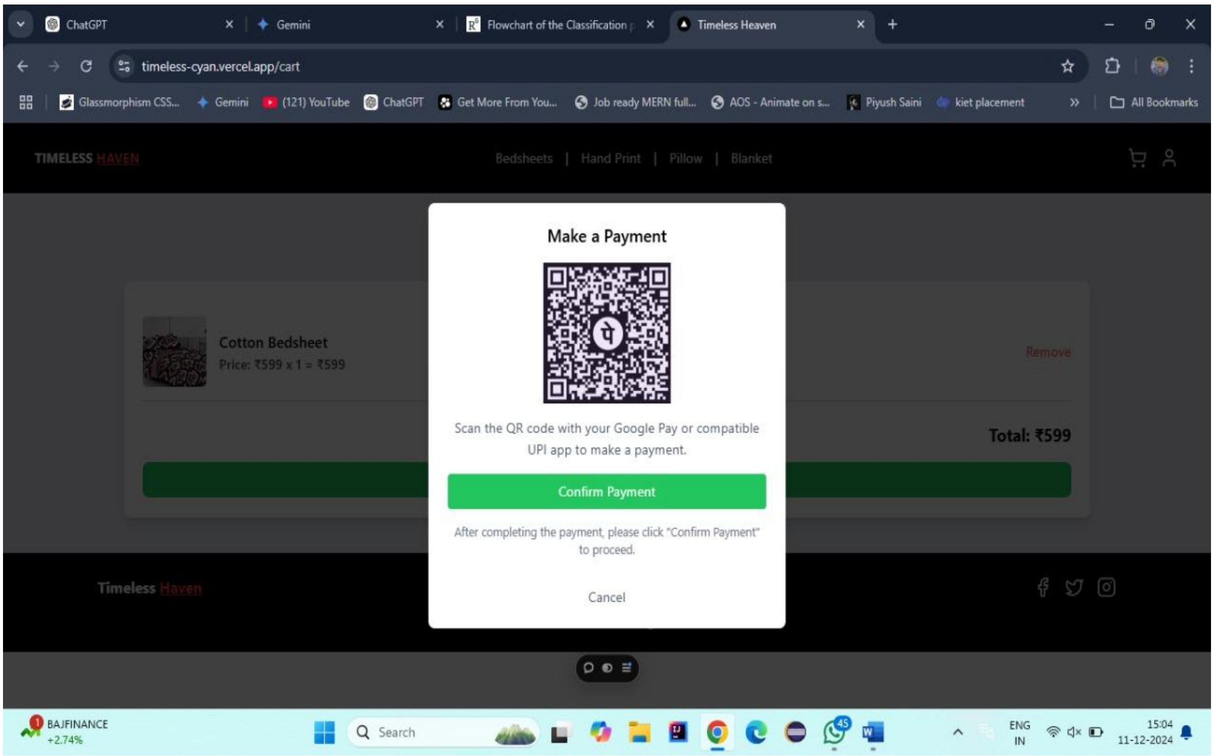
Cotton Bedsheet
Price: ₹599 x 1 = ₹599

Total: ₹599

Items Added

"Timeless Haven brings to you a collection of bedsheets that blend elegance, comfort, and durability, designed to elevate your living spaces. Each bedsheet is crafted with premium materials, offering a soft and luxurious feel that ensures a restful sleep experience. With a variety of stunning designs and vibrant colors, our collection caters to diverse tastes and complements any interior decor. Experience the perfect combination of style and practicality with bedsheets that are easy to maintain and long-lasting. Transform your home into a cozy haven with our thoughtfully curated range."

9.3.2 Payment



Cotton Bedsheet
Price: ₹599 x 1 = ₹599

Remove

Total: ₹599

Make a Payment

Scan the QR code with your Google Pay or compatible UPI app to make a payment.

Confirm Payment

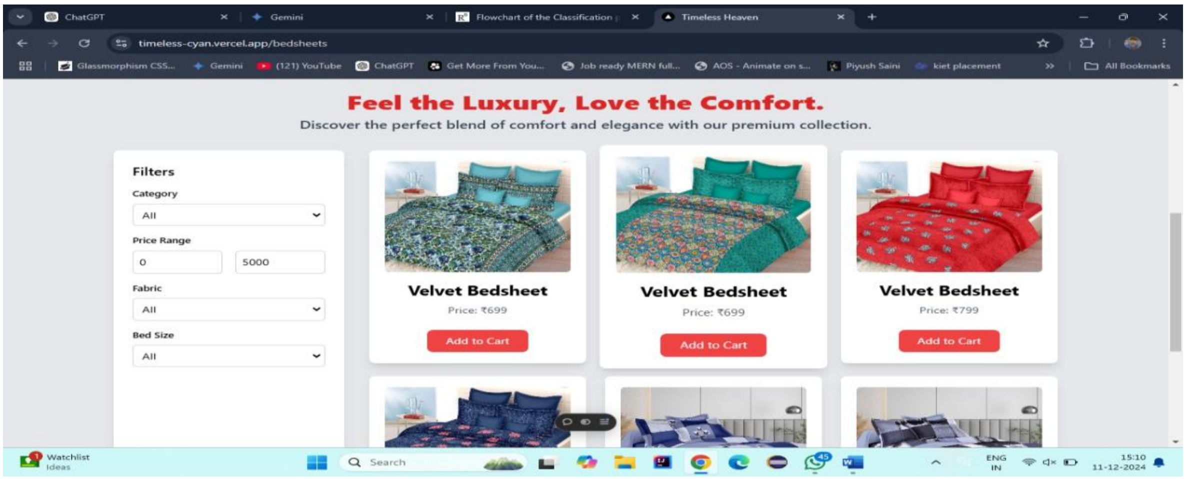
After completing the payment, please click "Confirm Payment" to proceed.

Cancel

Payment

At Timeless Haven Home Decor, we offer a seamless and secure payment process to ensure a hassle-free shopping experience for our customers. Whether you choose to pay through credit or debit cards, digital wallets, UPI, or net banking, our payment gateway is designed to handle transactions with utmost security and encryption. We are committed to providing flexible and convenient options to suit your needs, making it easy for you to bring elegance and comfort to your home with our exquisite bedsheet collection. Rest assured, your personal and payment information is always protected with the highest standards of confidentiality.

9.3.3 Search Items



Feel the Luxury, Love the Comfort.

Discover the perfect blend of comfort and elegance with our premium collection.

Filters

Category: All

Price Range: 0 5000

Fabric: All

Bed Size: All

Velvet Bedsheet
Price: ₹699
Add to Cart

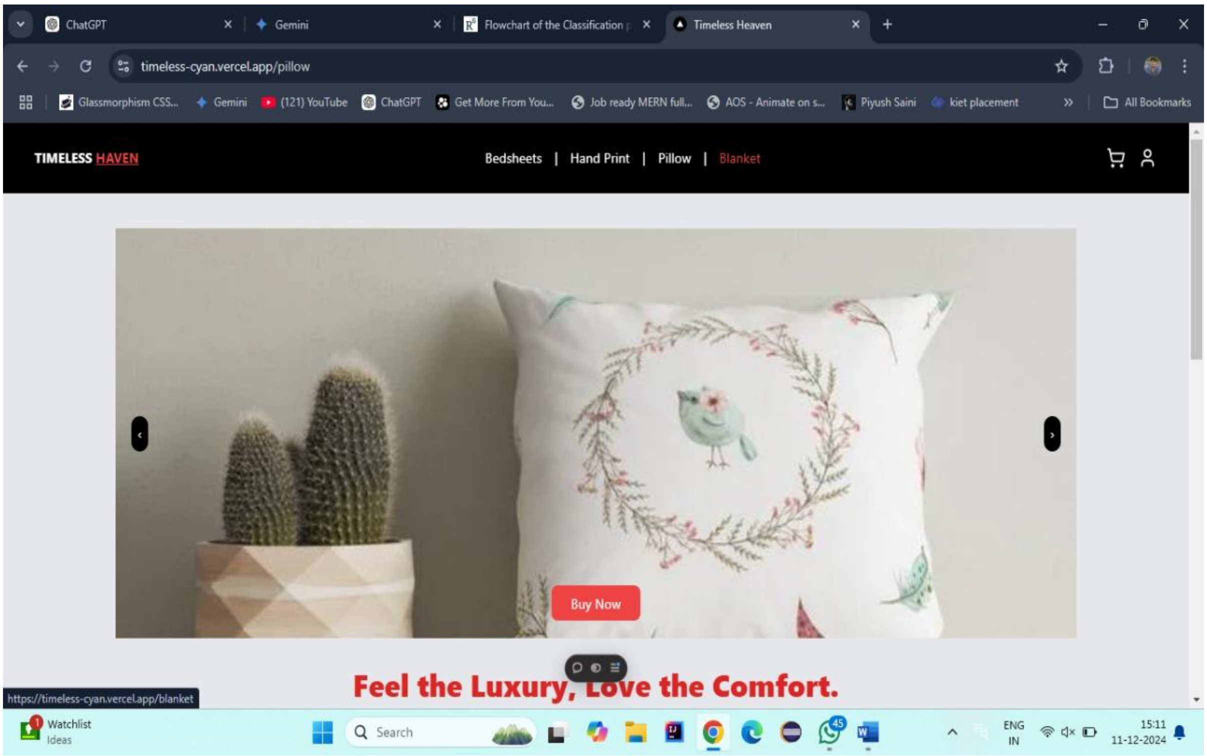
Velvet Bedsheet
Price: ₹699
Add to Cart

Velvet Bedsheet
Price: ₹799
Add to Cart

Search Item

To search for a package on our Timeless Haven website, simply use the search bar located at the top of the homepage. Timeless Haven Home Decor offers a curated selection of classic and elegant pieces designed to infuse your living space with enduring charm. Their collection features regal furnishings, such as oversized sofas and ornate coffee tables, that exude opulence and sophistication. Complementing these are vintage accents like antique vases and weathered picture frames, adding a touch of nostalgia and history to your decor. Luxurious drapery crafted from sumptuous fabrics, including velvet and silk, introduces an element of grandeur, while traditional patterns like damask and floral prints on upholstery and accessories contribute to a cohesive and timeless aesthetic. The inclusion of grand mirrors and elegant lighting fixtures, such as chandeliers and sconces, enhances the ambiance, creating a warm and inviting atmosphere. Each item in their collection is thoughtfully selected to blend seamlessly, ensuring your home reflects a sense of classic elegance that stands the test of time

9.3.4 Homepage



Homepage

Welcome to Timeless Haven Home Decor, where elegance meets comfort to transform your house into a sanctuary of style and warmth. Our carefully curated collection of decor pieces blends classic charm with contemporary trends, ensuring that every item adds a touch of sophistication to your living space. Whether you're seeking statement furniture, cozy textiles, or striking accents, we offer designs that celebrate both timeless beauty and everyday functionality.

At Timeless Haven, we believe your home should reflect your unique personality and provide a haven of relaxation. That's why we focus on quality craftsmanship and sustainable materials to create decor that lasts. Explore our range of exquisite products, thoughtfully designed to inspire creativity and elevate every corner of your home into a harmonious retreat.

CHAPTER 10

CODING

HEADER

```
import { ShoppingCart, User } from "lucide-react";
import Link from "next/link";
import React from "react";

const Header = () => {
  return (
    <div className="flex w-full bg-black h-20 items-center text-white justify-between mx-auto">
      <div className="font-bold mx-10 uppercase">
        <Link href="/">
          <div>
            Timeless <span className="text-red-500 underline">Haven</span>
          </div>
        </Link>
      </div>
      <div className="lg:flex mx-auto hidden">
        <ul className="flex flex-row gap-4">
          <li>
            <Link href="/bedsheets" className="hover:text-red-500">
              Bedsheets
            </Link>
          </li>
          <span>|</span>
          <li>
            <Link href="/handPrint" className="hover:text-red-500">
            Hand Print
            </Link>
          </li>
          <span>|</span>
          <li>
            <Link href="/pillow" className="hover:text-red-500">
              Pillow
            </Link>
          </li>
          <span>|</span>
          <li>
            <Link href="/blanket" className="hover:text-red-500">
              Blanket
            </Link>
          </li>
        </ul>
      </div>
      <div className="mx-10 flex flex-row gap-4">
```

```
        <Link href="/cart" className="hover:text-red-500">
          <ShoppingCart />
        </Link>
        <Link href="/profile" className="hover:text-red-500">
          <User />
        </Link>
      </div>
    </div>
  );
};

export default Header;
```

FOOTER

```
import React from "react";
import Link from "next/link";
import { Facebook, Twitter, Instagram } from "lucide-react";

const Footer = () => {
  return (
    <div className="bg-black text-white py-6">
      <div className="container mx-auto flex flex-col md:flex-row justify-between items-center">
        <div className="font-bold text-lg mb-4 md:mb-0">
          <Link href="/">
            Timeless <span className="text-red-500 underline">Haven</span>
          </Link>
        </div>

        <div className="flex flex-row gap-6 mb-4 md:mb-0">
          <Link href="/about" className="hover:text-red-500">About Us</Link>
          <Link href="/contact" className="hover:text-red-500">Contact</Link>
          <Link href="/privacy" className="hover:text-red-500">Privacy Policy</Link>
          <Link href="/terms" className="hover:text-red-500">Terms of Service</Link>
        </div>

        <div className="flex gap-4">
          <Link href="https://facebook.com" className="hover:text-red-500">
            <Facebook />
          </Link>
          <Link href="https://twitter.com" className="hover:text-red-500">
            <Twitter />
          </Link>
          <Link href="https://instagram.com" className="hover:text-red-500">
            <Instagram />
          </Link>
        </div>
      </div>
      <div className="text-center text-sm mt-4">
        &copy; {new Date().getFullYear()} Timeless Heaven. All rights reserved.
      </div>
    </div>
  );
};

export default Footer;
```

ABOUT

```
import React from "react";
import Link from "next/link";
import { Facebook, Twitter, Instagram } from "lucide-react";

const Footer = () => {
  return (
    <div className="bg-black text-white py-6">
      <div className="container mx-auto flex flex-col md:flex-row justify-between items-center">
        <div className="font-bold text-lg mb-4 md:mb-0">
          <Link href="/">
            Timeless <span className="text-red-500 underline">Haven</span>
          </Link>
        </div>

        <div className="flex flex-row gap-6 mb-4 md:mb-0">
          <Link href="/about" className="hover:text-red-500">About Us</Link>
          <Link href="/contact" className="hover:text-red-500">Contact</Link>
          <Link href="/privacy" className="hover:text-red-500">Privacy Policy</Link>
          <Link href="/terms" className="hover:text-red-500">Terms of Service</Link>
        </div>

        <div className="flex gap-4">
          <Link href="https://facebook.com" className="hover:text-red-500">
            <Facebook />
          </Link>
          <Link href="https://twitter.com" className="hover:text-red-500">
            <Twitter />
          </Link>
          <Link href="https://instagram.com" className="hover:text-red-500">
            <Instagram />
          </Link>
        </div>
      </div>

      <div className="text-center text-sm mt-4">
        &copy; {new Date().getFullYear()} Timeless Heaven. All rights reserved.
      </div>
    </div>
  );
};

export default Footer;
```

BEDSHEETS

```
"use client";
import { useCart } from "@/context/CartContext";
import { useRouter } from "next/navigation";
import React, { useState, useEffect } from "react";

const Home = () => {
  const images = [
    "/img5.jpg",
    "/Velvet new _page-0015.jpg",
    "/Velvet new _page-0012.jpg",
    "/Velvet new _page-0012.jpg",
    "/Velvet new _page-0014.jpg",
    "/Velvet new _page-0008.jpg",
    "/img31.jpg",
  ];

  const [currentIndex, setCurrentIndex] = useState(0);
  const [products, setProducts] = useState([]);
  const [filteredProducts, setFilteredProducts] = useState([]);
  const [category, setCategory] = useState("all");
  const [priceRange, setPriceRange] = useState([0, 5000]); // Example price range [min, max]
  const [fabric, setFabric] = useState("all");
  const [bedSize, setBedSize] = useState("all");
  const { addToCart } = useCart();
  const router = useRouter();

  useEffect(() => {
    const interval = setInterval(() => {
      setCurrentIndex((prevIndex) =>
        prevIndex === images.length - 1 ? 0 : prevIndex + 1
      );
    }, 3000);

    const fetchProducts = async () => {
      const res = await fetch("/productbed.json");
      const data = await res.json();
      setProducts(data);
      setFilteredProducts(data);
    };

    fetchProducts();
    return () => clearInterval(interval);
```

```
}, [images.length]);

const handleAddToCart = (product) => {
  addToCart(product);
  router.push("/cart");
};

const handleBuyNow = (product) => {
  addToCart(product);
  router.push("/cart");
};

const handleCategoryChange = (e) => {
  setCategory(e.target.value);
};

const handlePriceChange = (e) => {
  const { name, value } = e.target;
  if (name === "minPrice") {
    setPriceRange([Number(value), priceRange[1]]);
  } else if (name === "maxPrice") {
    setPriceRange([priceRange[0], Number(value)]);
  }
};

const handleFabricChange = (e) => {
  setFabric(e.target.value);
};

const handleBedSizeChange = (e) => {
  setBedSize(e.target.value);
};

useEffect(() => {
  const filtered = products.filter((product) => {
    const isCategoryMatch = category === "all" || product.category === category;
    const isPriceInRange =
      product.price >= priceRange[0] && product.price <= priceRange[1];
    const isFabricMatch = fabric === "all" || product.fabric === fabric;
    const isBedSizeMatch = bedSize === "all" || product.bedSize === bedSize;

    return isCategoryMatch && isPriceInRange && isFabricMatch && isBedSizeMatch;
  });

  setFilteredProducts(filtered);
}, [category, priceRange, fabric, bedSize, products]);

// Function to navigate to the previous image
const goToPrevious = () => {
  setCurrentIndex((prevIndex) =>
    prevIndex === 0 ? images.length - 1 : prevIndex - 1
    );
  };

  // Function to navigate to the next image
  const goToNext = () => {
    setCurrentIndex((prevIndex) =>
      prevIndex === images.length - 1 ? 0 : prevIndex + 1
    );
  };

  return (
    <div className="h-auto w-auto bg-gray-200 flex flex-col justify-center items-center py-10">
      {/* Hero Image Section */}
      <div className="bg-black flex mx-auto w-[80vw] h-[70vh] justify-center overflow-hidden
relative">
        <img
          src={images[currentIndex]}
          className="w-full h-full object-cover"
          alt={`Slide ${currentIndex + 1}`}
        />
        {/* Left Arrow */}
        <button
          onClick={goToPrevious}
          className="absolute left-5 top-1/2 transform -translate-y-1/2 bg-black text-white p-2 rounded-
full hover:bg-gray-700 transition"
        >
          &#8249;
        </button>
        {/* Right Arrow */}
        <button
          onClick={goToNext}
          className="absolute right-5 top-1/2 transform -translate-y-1/2 bg-black text-white p-2 rounded-
full hover:bg-gray-700 transition"
        >
          &#8250;
        </button>
        <div className="absolute bottom-5 left-1/2 transform -translate-x-1/2 space-x-4">
          <button
            onClick={() => handleBuyNow(products[0])} // You can set a specific product for Buy Now
            className="bg-red-500 text-white px-6 py-2 rounded-lg hover:bg-red-600 transition"
          >
            Buy Now
          </button>
        </div>
      </div>

      {/* Slogan Section */}
      <div className="mt-8 text-center">
        <h2 className="text-4xl font-extrabold text-red-600">
          Feel the Luxury, Love the Comfort.
        </h2>
  <p className="text-xl text-gray-700 mt-2 font-medium">
    Discover the perfect blend of comfort and elegance with our premium collection.
  </p>
</div>

{/* Filter Section */}
<div className="w-[80vw] max-w-8xl mt-8 flex gap-8">
  <div className="bg-white p-6 rounded-lg shadow-lg w-1/4">
    <h3 className="text-xl font-semibold mb-4">Filters</h3>

    {/* Category Filter */}
    <div className="mb-4">
      <label className="block text-sm font-medium mb-2">Category</label>
      <select
        value={category}
        onChange={handleCategoryChange}
        className="w-full p-2 border border-gray-300 rounded-md"
      >
        <option value="all">All</option>
        <option value="bedsheets">Bedsheets</option>
        <option value="pillow">Pillow</option>
        <option value="mattress">Mattress</option>
        <option value="blanket">Blanket</option>
      </select>
    </div>

    {/* Price Filter */}
    <div className="mb-4">
      <label className="block text-sm font-medium mb-2">Price Range</label>
      <div className="flex gap-4">
        <input
          type="number"
          name="minPrice"
          value={priceRange[0]}
          onChange={handlePriceChange}
          className="w-1/2 p-2 border border-gray-300 rounded-md"
          placeholder="Min"
        />
        <input
          type="number"
          name="maxPrice"
          value={priceRange[1]}
          onChange={handlePriceChange}
          className="w-1/2 p-2 border border-gray-300 rounded-md"
          placeholder="Max"
        />
      </div>
    </div>

    {/* Fabric Filter */}
    <div className="mb-4">
```

```
            <label className="block text-sm font-medium mb-2">Fabric</label>
            <select
              value={fabric}
              onChange={handleFabricChange}
              className="w-full p-2 border border-gray-300 rounded-md"
            >
              <option value="all">All</option>
              <option value="cotton">Cotton</option>
              <option value="silk">Silk</option>
              <option value="polyester">Polyester</option>
              <option value="linen">Linen</option>
            </select>
          </div>

          {/* Bed Size Filter */}
          <div className="mb-4">
            <label className="block text-sm font-medium mb-2">Bed Size</label>
            <select
              value={bedSize}
              onChange={handleBedSizeChange}
              className="w-full p-2 border border-gray-300 rounded-md"
            >
              <option value="all">All</option>
              <option value="single">Single</option>
              <option value="double">Double</option>
            </select>
          </div>
        </div>

        {/* Card Section with 3 Columns */}
        <div className="w-3/4 grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 gap-6">
          {filteredProducts.map((product) => (
            <div
              key={product.id}
              className="bg-white p-5 rounded-lg shadow-lg flex flex-col items-center hover:scale-105
hover:shadow-lg transition duration-300 ease-in-out"
            >
              <img
                src={product.image}
                alt={product.title}
                className="w-full h-[200px] object-cover mb-4 rounded-md"
              />
              <h2 className="text-2xl font-bold mb-2">{product.title}</h2>
              <p className="text-gray-600 mb-2">Price: ₹{product.price}</p>
              <p className="text-gray-600 mb-4 text-center">{product.description}</p>
              <button
                onClick={() => handleAddToCart(product)}
                className="bg-red-500 text-white px-6 py-2 rounded-lg hover:bg-red-600 transition"
              >
                Add to Cart
              </button>
          </div>
        ))}
      </div>
    </div>
  </div>
 );
};

export default Home;
```

BLANKET

```
"use client";
import { useCart } from "@/context/CartContext";
import { useRouter } from "next/navigation";
import React, { useState, useEffect } from "react";

const Home = () => {
 const images = [
  "/bt1.jpg",
  "/bt2.jpg",
  "/bt3.jpg",
  "/bt4.jpg",
  "/bt5.jpg",
  "/bt6.jpg",
  "/bt4.jpg",
 ];

 const [currentIndex, setCurrentIndex] = useState(0);
 const [products, setProducts] = useState([]);
 const [filteredProducts, setFilteredProducts] = useState([]);
 const [category, setCategory] = useState("all");
 const [priceRange, setPriceRange] = useState([0, 5000]); // Example price range [min, max]
 const [fabric, setFabric] = useState("all");
 const [bedSize, setBedSize] = useState("all");
 const { addToCart } = useCart();
 const router = useRouter();

 useEffect(() => {
  const interval = setInterval(() => {
   setCurrentIndex((prevIndex) =>
    prevIndex === images.length - 1 ? 0 : prevIndex + 1
   );
  }, 3000);

  const fetchProducts = async () => {
   const res = await fetch("/productsbt.json");
   const data = await res.json();
   setProducts(data);
   setFilteredProducts(data);
   };

   fetchProducts();
   return () => clearInterval(interval);
 }, [images.length]);

 const handleAddToCart = (product) => {
   addToCart(product);
   router.push("/cart");
 };

 const handleBuyNow = (product) => {
   addToCart(product);
   router.push("/cart");
 };

 const handleCategoryChange = (e) => {
   setCategory(e.target.value);
 };

 const handlePriceChange = (e) => {
   const { name, value } = e.target;
   if (name === "minPrice") {
     setPriceRange([Number(value), priceRange[1]]);
   } else if (name === "maxPrice") {
     setPriceRange([priceRange[0], Number(value)]);
   }
 };

 const handleFabricChange = (e) => {
   setFabric(e.target.value);
 };

 const handleBedSizeChange = (e) => {
   setBedSize(e.target.value);
 };

 useEffect(() => {
   const filtered = products.filter((product) => {
     const isCategoryMatch =
       category === "all" || product.category === category;
     const isPriceInRange =
       product.price >= priceRange[0] && product.price <= priceRange[1];
     const isFabricMatch = fabric === "all" || product.fabric === fabric;
     const isBedSizeMatch = bedSize === "all" || product.bedSize === bedSize;

     return (
       isCategoryMatch && isPriceInRange && isFabricMatch && isBedSizeMatch
     );
   });
```

```
    setFilteredProducts(filtered);
  }, [category, priceRange, fabric, bedSize, products]);

  // Function to navigate to the previous image
  const goToPrevious = () => {
    setCurrentIndex((prevIndex) =>
      prevIndex === 0 ? images.length - 1 : prevIndex - 1
    );
  };

  // Function to navigate to the next image
  const goToNext = () => {
    setCurrentIndex((prevIndex) =>
      prevIndex === images.length - 1 ? 0 : prevIndex + 1
    );
  };

  return (
    <div className="h-auto w-auto bg-gray-200 flex flex-col justify-center items-center py-10">
      {/* Hero Image Section */}
      <div className="bg-black flex mx-auto w-[80vw] h-[70vh] justify-center overflow-hidden relative">
        <img
          src={images[currentIndex]}
          className="w-full h-full object-cover"
          alt={`Slide ${currentIndex + 1}`}
        />
        {/* Left Arrow */}
        <button
          onClick={goToPrevious}
          className="absolute left-5 top-1/2 transform -translate-y-1/2 bg-black text-white p-2 rounded-full hover:bg-gray-700 transition"
        >
          &#8249;
        </button>
        {/* Right Arrow */}
        <button
          onClick={goToNext}
          className="absolute right-5 top-1/2 transform -translate-y-1/2 bg-black text-white p-2 rounded-full hover:bg-gray-700 transition"
        >
          &#8250;
        </button>
        <div className="absolute bottom-5 left-1/2 transform -translate-x-1/2 space-x-4">
          <button
            onClick={() => handleBuyNow(products[0])} // You can set a specific product for Buy Now
            className="bg-red-500 text-white px-6 py-2 rounded-lg hover:bg-red-600 transition"
          >
            Buy Now
          </button>
        </div>
```

```
</div>

{/* Slogan Section */}
<div className="mt-8 text-center">
  <h2 className="text-4xl font-extrabold text-red-600">
  "Stay Warm, Stay Cozy, Stay Home."
  </h2>
  <p className="text-xl text-gray-700 mt-2 font-medium">
    Discover the perfect blend of comfort and elegance with our premium
    collection.
  </p>
</div>

{/* Filter Section */}
<div className="w-[80vw] max-w-8xl mt-8 flex gap-8">
  <div className="bg-white p-6 rounded-lg shadow-lg w-1/4">
    <h3 className="text-xl font-semibold mb-4">Filters</h3>

    {/* Category Filter */}
    <div className="mb-4">
      <label className="block text-sm font-medium mb-2">Category</label>
      <select
        value={category}
        onChange={handleCategoryChange}
        className="w-full p-2 border border-gray-300 rounded-md"
      >
        <option value="all">All</option>
        <option value="bedsheets">Bedsheets</option>
        <option value="pillow">Pillow</option>
        <option value="mattress">Mattress</option>
        <option value="blanket">Blanket</option>
      </select>
    </div>

    {/* Price Filter */}
    <div className="mb-4">
      <label className="block text-sm font-medium mb-2">
        Price Range
      </label>
      <div className="flex gap-4">
        <input
          type="number"
          name="minPrice"
          value={priceRange[0]}
          onChange={handlePriceChange}
          className="w-1/2 p-2 border border-gray-300 rounded-md"
          placeholder="Min"
        />
        <input
          type="number"
          name="maxPrice"
```

```
                value={priceRange[1]}
                onChange={handlePriceChange}
                className="w-1/2 p-2 border border-gray-300 rounded-md"
                placeholder="Max"
              />
            </div>
          </div>

          {/* Fabric Filter */}
          <div className="mb-4">
            <label className="block text-sm font-medium mb-2">Fabric</label>
            <select
              value={fabric}
              onChange={handleFabricChange}
              className="w-full p-2 border border-gray-300 rounded-md"
            >
              <option value="all">All</option>
              <option value="cotton">Cotton</option>
              <option value="silk">Silk</option>
              <option value="polyester">Polyester</option>
              <option value="linen">Linen</option>
            </select>
          </div>

          {/* Bed Size Filter */}
          <div className="mb-4">
            <label className="block text-sm font-medium mb-2">Bed Size</label>
            <select
              value={bedSize}
              onChange={handleBedSizeChange}
              className="w-full p-2 border border-gray-300 rounded-md"
            >
              <option value="all">All</option>
              <option value="single">Single</option>
              <option value="double">Double</option>
            </select>
          </div>
        </div>

        {/* Card Section with 3 Columns */}
        <div className="w-3/4 grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 gap-6">
          {filteredProducts.map((product) => (
            <div
              key={product.id}
              className="bg-white p-5 rounded-lg shadow-lg flex flex-col items-center hover:scale-105
hover:shadow-lg transition duration-300 ease-in-out"
            >
              <img
                src={product.image}
                alt={product.title}
                className="w-full h-[200px] object-cover mb-4 rounded-md"
              />
              <h2 className="text-2xl font-bold mb-2">{product.title}</h2>
              <p className="text-gray-600 mb-2">Price: ₹{product.price}</p>
              <p className="text-gray-600 mb-4 text-center">
                {product.description}
              </p>
              <button
                onClick={() => handleAddToCart(product)}
                className="bg-red-500 text-white px-6 py-2 rounded-lg hover:bg-red-600 transition"
              >
                Add to Cart
              </button>
            </div>
          ))}
        </div>
      </div>
     </div>
     );
};

export default Home;
```

CART

```
"use client";
import { useCart } from "@/context/CartContext";
import React, { useState } from "react";

const Cart = () => {
  const { cart, removeFromCart, getTotalPrice } = useCart();
  const [isModalOpen, setIsModalOpen] = useState(false);

  // Open modal
  const handlePayNow = () => {
    setIsModalOpen(true);
  };

  // Close modal and complete "payment"
  const handleCompletePayment = () => {
    alert("Payment successful! Thank you for your purchase.");
    setIsModalOpen(false);
    // Reset the cart or proceed with further actions as needed
  };

  return (
    <div className="h-auto w-auto bg-gray-200 flex flex-col justify-center items-center py-10">
      <h2 className="text-3xl font-bold mb-6">Your Cart</h2>
      {cart.length === 0 ? (
        <p className="text-gray-600">Your cart is empty.</p>
      ) : (
        <div className="w-[80vw] max-w-8xl bg-white p-6 rounded-lg shadow-lg">
          <ul>
            {cart.map((item, index) => (
              <li key={index} className="border-b border-gray-300 py-4">
                <div className="flex justify-between items-center">
                  <div className="flex items-center gap-4">
                    <img
                      src={item.image}
                      alt={item.title}
                      className="w-20 h-20 object-cover rounded-md"
                    />
                    <div>
                      <h3 className="text-lg font-semibold">{item.title}</h3>
                      <p className="text-gray-600">{item.description}</p>
                      <p className="text-gray-700 font-semibold">
                        Price: ₹{item.price} x {item.quantity} = ₹
                        {item.price * item.quantity}
                      </p>
                    </div>
                  </div>
                  <button
                    onClick={() => removeFromCart(item.id)}
                    className="text-red-500 hover:text-red-700"
                  >
                    Remove
                  </button>
                </div>
              </li>
            ))}
          </ul>
          <div className="text-xl font-bold text-right mt-6">
            Total: ₹{getTotalPrice()}
          </div>
          <button
            onClick={handlePayNow}
            className="mt-4 w-full bg-green-500 text-white py-2 rounded-lg hover:bg-green-600
transition"
          >
            Pay Now
          </button>
        </div>
      )}

      {/* Payment Modal */}
      {isModalOpen && (
        <div className="fixed inset-0 bg-black bg-opacity-50 flex justify-center items-center">
          <div className="bg-white p-6 rounded-lg shadow-lg w-[90vw] max-w-md">
            <h3 className="text-xl font-semibold mb-4">Payment Details</h3>
            <p className="text-gray-600 mb-4">Total: ₹{getTotalPrice()}</p>
            {/* Payment Modal */}
{isModalOpen && (
  <div className="fixed inset-0 bg-black bg-opacity-50 flex justify-center items-center">
    <div className="bg-white p-6 rounded-lg shadow-lg w-[90vw] max-w-md">
      <h3 className="text-xl font-semibold mb-4 text-center">Make a Payment</h3>

      {/* QR Code Section */}
      <div className="flex justify-center mb-4">
        <img
          src="/qrcode.jpg" // Replace with your QR code image path
          alt="QR Code for Payment"
          className="w-40 h-40"
        />
      </div>

      <p className="text-center text-gray-600 mb-4">
        Scan the QR code with your Google Pay or compatible UPI app to make a payment.
      </p>

      {/* Confirm Payment Button */}
      <form
        onSubmit={(e) => {
          e.preventDefault();
          handleCompletePayment();
        }}
        className="flex flex-col items-center gap-4"
      >
        <button
          type="submit"
          className="bg-green-500 text-white w-full py-2 rounded hover:bg-green-600 transition"
        >
          Confirm Payment
        </button>

        <p className="text-center text-gray-500 text-sm">
          After completing the payment, please click "Confirm Payment" to proceed.
        </p>

        {/* Cancel Button */}
        <button
          type="button"
          onClick={() => setIsModalOpen(false)}
          className="text-gray-600 mt-4 hover:underline"
        >
          Cancel
        </button>
      </form>
    </div>
  </div>
)}

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

export default Cart;
```

CONTACT

```
"use client";
import React, { useState } from "react";

const page = () => {
 const [formData, setFormData] = useState({
  name: "",
  email: "",
  message: "",
 });

 const handleChange = (e) => {
  const { name, value } = e.target;
  setFormData({ ...formData, [name]: value });
 };

 const handleSubmit = (e) => {
  e.preventDefault();
  // Logic for handling form submission
  console.log("Form submitted:", formData);
 };

 return (
  <div className="min-h-screen bg-gray-100 py-10">
   <div className="container mx-auto px-4">
    <div className="text-center mb-12">
     <h1 className="text-3xl font-semibold text-gray-800">Contact Us</h1>
     <p className="text-gray-600 mt-2">We'd love to hear from you!</p>
    </div>

    <div className="bg-white p-8 rounded-lg shadow-lg max-w-4xl mx-auto">
     <form action="" onSubmit={handleSubmit} className="space-y-6">
      <div>
       <label htmlFor="name" className="block text-gray-700">Name</label>
       <input
        type="text"
        id="name"
        name="name"
        value={formData.name}
        onChange={handleChange}
        required
        className="w-full mt-2 p-3 border border-gray-300 rounded-md focus:outline-none
```

```
focus:ring-2 focus:ring-red-500"
              />
            </div>

            <div>
              <label htmlFor="email" className="block text-gray-700">Email</label>
              <input
                type="email"
                id="email"
                name="email"
                value={formData.email}
                onChange={handleChange}
                required
                className="w-full mt-2 p-3 border border-gray-300 rounded-md focus:outline-none
focus:ring-2 focus:ring-red-500"
              />
            </div>

            <div>
              <label htmlFor="message" className="block text-gray-700">Message</label>
              <textarea
                id="message"
                name="message"
                value={formData.message}
                onChange={handleChange}
                required
                rows="6"
                className="w-full mt-2 p-3 border border-gray-300 rounded-md focus:outline-none
focus:ring-2 focus:ring-red-500"
              />
            </div>

            <button
              type="submit"
              className="w-full py-3 bg-red-500 text-white rounded-md hover:bg-red-600 transition"
            >
              Send Message
            </button>
          </form>
        </div>

        <div className="mt-12 text-center">
          <h2 className="text-2xl font-semibold text-gray-800">Our Contact Information</h2>
          <p className="text-gray-600 mt-2">Feel free to reach us at any time.</p>

          <div className="mt-8 flex justify-center space-x-8">
            <div>
              <h3 className="text-xl font-semibold text-gray-800">Email</h3>
              <p className="text-gray-600">contact@timelessheaven.com</p>
            </div>
            <div>
              <h3 className="text-xl font-semibold text-gray-800">Phone</h3>
              <p className="text-gray-600">+1 234 567 890</p>
            </div>

            <div>
              <h3 className="text-xl font-semibold text-gray-800">Address</h3>
              <p className="text-gray-600">123 Timeless Ave, Heaven City, HC 12345</p>
            </div>
          </div>
        </div>
      </div>
    </div>
   );
};

export default page;
```

HANDPRINT

```
"use client";
import { useCart } from "@/context/CartContext";
import { useRouter } from "next/navigation";
import React, { useState, useEffect } from "react";

const Home = () => {
  const images = [
    "/hp6.jpg",
    "/hp2.jpg",
    "/hp3.jpg",
    "/hp4.jpg",
    "/hp5.jpg",
    "/hp6.jpg",
    "/hp7.jpg",
  ];

  const [currentIndex, setCurrentIndex] = useState(0);
  const [products, setProducts] = useState([]);
  const [filteredProducts, setFilteredProducts] = useState([]);
  const [category, setCategory] = useState("all");
  const [priceRange, setPriceRange] = useState([0, 5000]); // Example price range [min, max]
  const [fabric, setFabric] = useState("all");
  const [bedSize, setBedSize] = useState("all");
  const { addToCart } = useCart();
  const router = useRouter();

  useEffect(() => {
    const interval = setInterval(() => {
      setCurrentIndex((prevIndex) =>
        prevIndex === images.length - 1 ? 0 : prevIndex + 1
      );
    }, 3000);

    const fetchProducts = async () => {
      const res = await fetch("/productshp.json");
      const data = await res.json();
      setProducts(data);
      setFilteredProducts(data);
    };

    fetchProducts();
    return () => clearInterval(interval);
  }, [images.length]);

  const handleAddToCart = (product) => {
    addToCart(product);
    router.push("/cart");
  };

  const handleBuyNow = (product) => {
    addToCart(product);
    router.push("/cart");
  };

  const handleCategoryChange = (e) => {
    setCategory(e.target.value);
  };

  const handlePriceChange = (e) => {
    const { name, value } = e.target;
    if (name === "minPrice") {
      setPriceRange([Number(value), priceRange[1]]);
    } else if (name === "maxPrice") {
      setPriceRange([priceRange[0], Number(value)]);
    }
  };

  const handleFabricChange = (e) => {
    setFabric(e.target.value);
  };

  const handleBedSizeChange = (e) => {
    setBedSize(e.target.value);
  };

  useEffect(() => {
    const filtered = products.filter((product) => {
      const isCategoryMatch =
        category === "all" || product.category === category;
      const isPriceInRange =
        product.price >= priceRange[0] && product.price <= priceRange[1];
      const isFabricMatch = fabric === "all" || product.fabric === fabric;
      const isBedSizeMatch = bedSize === "all" || product.bedSize === bedSize;

      return (
        isCategoryMatch && isPriceInRange && isFabricMatch && isBedSizeMatch
      );
    });

    setFilteredProducts(filtered);
  }, [category, priceRange, fabric, bedSize, products]);

  // Function to navigate to the previous image
  const goToPrevious = () => {
    setCurrentIndex((prevIndex) =>
      prevIndex === 0 ? images.length - 1 : prevIndex - 1
    );
  };

  // Function to navigate to the next image
  const goToNext = () => {
    setCurrentIndex((prevIndex) =>
      prevIndex === images.length - 1 ? 0 : prevIndex + 1
    );
  };

  return (
    <div className="h-auto w-auto bg-gray-200 flex flex-col justify-center items-center py-10">
      {/* Hero Image Section */}
      <div className="bg-black flex mx-auto w-[80vw] h-[70vh] justify-center overflow-hidden relative">
        <img
          src={images[currentIndex]}
          className="w-full h-full object-cover"
          alt={`Slide ${currentIndex + 1}`}
        />
        {/* Left Arrow */}
        <button
          onClick={goToPrevious}
          className="absolute left-5 top-1/2 transform -translate-y-1/2 bg-black text-white p-2 rounded-full hover:bg-gray-700 transition"
        >
          &#8249;
        </button>
        {/* Right Arrow */}
        <button
          onClick={goToNext}
          className="absolute right-5 top-1/2 transform -translate-y-1/2 bg-black text-white p-2 rounded-full hover:bg-gray-700 transition"
        >
          &#8250;
        </button>
  <div className="absolute bottom-5 left-1/2 transform -translate-x-1/2 space-x-4">
    <button
      onClick={() => handleBuyNow(products[0])} // You can set a specific product for Buy Now
      className="bg-red-500 text-white px-6 py-2 rounded-lg hover:bg-red-600 transition"
    >
      Buy Now
    </button>
  </div>
</div>

{/* Slogan Section */}
<div className="mt-8 text-center">
  <h2 className="text-4xl font-extrabold text-red-600">
    Feel the Luxury, Love the Comfort.
  </h2>
  <p className="text-xl text-gray-700 mt-2 font-medium">
    Discover the perfect blend of comfort and elegance with our premium
    collection.
  </p>
</div>

{/* Filter Section */}
<div className="w-[80vw] max-w-8xl mt-8 flex gap-8">
  <div className="bg-white p-6 rounded-lg shadow-lg w-1/4">
    <h3 className="text-xl font-semibold mb-4">Filters</h3>

    {/* Category Filter */}
    <div className="mb-4">
      <label className="block text-sm font-medium mb-2">Category</label>
      <select
        value={category}
        onChange={handleCategoryChange}
        className="w-full p-2 border border-gray-300 rounded-md"
      >
        <option value="all">All</option>
        <option value="bedsheets">Bedsheets</option>
        <option value="pillow">Pillow</option>
        <option value="mattress">Mattress</option>
        <option value="blanket">Blanket</option>
      </select>
    </div>

    {/* Price Filter */}
    <div className="mb-4">
      <label className="block text-sm font-medium mb-2">
        Price Range
      </label>
      <div className="flex gap-4">
        <input
          type="number"
          name="minPrice"
          value={priceRange[0]}
          onChange={handlePriceChange}
          className="w-1/2 p-2 border border-gray-300 rounded-md"
          placeholder="Min"
        />
        <input
          type="number"
          name="maxPrice"
          value={priceRange[1]}
          onChange={handlePriceChange}
          className="w-1/2 p-2 border border-gray-300 rounded-md"
          placeholder="Max"
        />
      </div>
    </div>

    {/* Fabric Filter */}
    <div className="mb-4">
      <label className="block text-sm font-medium mb-2">Fabric</label>
      <select
        value={fabric}
        onChange={handleFabricChange}
        className="w-full p-2 border border-gray-300 rounded-md"
      >
        <option value="all">All</option>
        <option value="cotton">Cotton</option>
        <option value="silk">Silk</option>
        <option value="polyester">Polyester</option>
        <option value="linen">Linen</option>
      </select>
    </div>

    {/* Bed Size Filter */}
    <div className="mb-4">
      <label className="block text-sm font-medium mb-2">Bed Size</label>
      <select
        value={bedSize}
        onChange={handleBedSizeChange}
        className="w-full p-2 border border-gray-300 rounded-md"
      >
        <option value="all">All</option>
        <option value="single">Single</option>
        <option value="double">Double</option>
      </select>
    </div>
  </div>

  {/* Card Section with 3 Columns */}
  <div className="w-3/4 grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 gap-6">
    {filteredProducts.map((product) => (
      <div
```

```
              key={product.id}
              className="bg-white p-5 rounded-lg shadow-lg flex flex-col items-center hover:scale-105
hover:shadow-lg transition duration-300 ease-in-out"
            >
              <img
                src={product.image}
                alt={product.title}
                className="w-full h-[200px] object-cover mb-4 rounded-md"
              />
              <h2 className="text-2xl font-bold mb-2">{product.title}</h2>
              <p className="text-gray-600 mb-2">Price: ₹{product.price}</p>
              <p className="text-gray-600 mb-4 text-center">
                {product.description}
              </p>
              <button
                onClick={() => handleAddToCart(product)}
                className="bg-red-500 text-white px-6 py-2 rounded-lg hover:bg-red-600 transition"
              >
                Add to Cart
              </button>
            </div>
          ))}
        </div>
      </div>
    </div>
  );
};

export default Home;
```

LOGIN

```
"use client";
import { signIn } from "next-auth/react";
import { useState } from "react";
import { useRouter } from "next/navigation";

const page = () => {
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");

  const router = useRouter();

  const handleSubmit = async (e) => {
    e.preventDefault();
    // Sign in using email and password (custom credentials)
    await signIn("credentials", { email, password });
  };

  const handleGoogleSignIn = async () => {
```

```
  // Sign in using Google
  await signIn("google", { callbackUrl: "/" });
 };

 return (
  <div className="min-h-screen flex items-center justify-center bg-gray-100">
   <div className="bg-white p-8 rounded-lg shadow-lg w-[90%] max-w-md">
    <h1 className="text-2xl font-bold mb-6 text-center text-gray-800">
     Login
    </h1>
    <form onSubmit={handleSubmit} className="flex flex-col gap-4">
     {/* Email Input */}
     <input
      type="email"
      placeholder="Email"
      value={email}
      onChange={(e) => setEmail(e.target.value)}
      required
      className="p-3 border rounded-lg focus:outline-none focus:ring-2 focus:ring-red-500"
     />
     {/* Password Input */}
     <input
      type="password"
      placeholder="Password"
      value={password}
      onChange={(e) => setPassword(e.target.value)}
      required
      className="p-3 border rounded-lg focus:outline-none focus:ring-2 focus:ring-red-500"
     />
     {/* Login Button */}
     <button
      type="submit"
      className="bg-red-500 text-white py-3 rounded-lg hover:bg-red-600 transition"
     >
      Login
     </button>
    </form>
    {/* Divider */}
    <div className="flex items-center my-4">
     <div className="flex-grow h-px bg-gray-300"></div>
     <span className="px-3 text-gray-500 text-sm">OR</span>
     <div className="flex-grow h-px bg-gray-300"></div>
    </div>
    {/* Sign in with Google Button */}
    <button
     onClick={handleGoogleSignIn}
     className="bg-blue-500 text-white py-3 w-full rounded-lg hover:bg-blue-600 transition flex
items-center justify-center gap-2"
    >
     <img
      src="/google-icon.svg" // Add a Google icon to your public folder
```

```
          alt="Google"
          className="w-5 h-5"
        />
        Sign in with Google
      </button>
      {/* Signup Link */}
      <p className="mt-4 text-center text-gray-600">
        Don't have an account?{" "}
        <a href="/register" className="text-red-500 hover:underline">
          Sign up
        </a>
      </p>
    </div>
  </div>
 );
};

export default page;
```

LOGOUT

```
"use client";
import { signOut, useSession } from "next-auth/react";

const page = () => {
 const { data: session } = useSession();

 const handleLogout = async () => {
  await signOut(); // Logs out the user
 };

 return (
  <div className="min-h-screen flex items-center justify-center bg-gray-100">
   <div className="bg-white p-8 rounded-lg shadow-lg w-[90%] max-w-md text-center">
    <h1 className="text-2xl font-bold mb-4 text-gray-800">
     Welcome Back, {session?.user?.name || "User"}!
    </h1>
    <p className="text-gray-600 mb-6">
     You are logged in as <span className="font-medium">{session?.user?.email}</span>.
    </p>
    <button
     onClick={handleLogout}
     className="bg-red-500 text-white py-3 w-full rounded-lg hover:bg-red-600 transition"
    >
     Logout
    </button>
   </div>
  </div>
 );
};
```

```
export default page;
```

PILLOW

```
"use client";
import { useCart } from "@/context/CartContext";
import { useRouter } from "next/navigation";
import React, { useState, useEffect } from "react";

const page = () => {
  const images = [
    "/p1.webp",
    "/p2.webp",
    "/p3.webp",
    "/p4.webp",
    "/p5.jpg",
    "/p5.webp",
    "/p7.webp",
  ];

  const [currentIndex, setCurrentIndex] = useState(0);
  const [products, setProducts] = useState([]);
  const [filteredProducts, setFilteredProducts] = useState([]);
  const [category, setCategory] = useState("all");
  const [priceRange, setPriceRange] = useState([0, 5000]); // Example price range [min, max]
  const [fabric, setFabric] = useState("all");
  const [bedSize, setBedSize] = useState("all");
  const { addToCart } = useCart();
  const router = useRouter();

  useEffect(() => {
    const interval = setInterval(() => {
      setCurrentIndex((prevIndex) =>
        prevIndex === images.length - 1 ? 0 : prevIndex + 1
      );
    }, 3000);

    const fetchProducts = async () => {
      const res = await fetch("/pillow.json");
      const data = await res.json();
      setProducts(data);
      setFilteredProducts(data);
    };

    fetchProducts();
    return () => clearInterval(interval);
  }, [images.length]);

  const handleAddToCart = (product) => {
```

```
  addToCart(product);
  router.push("/cart");
};

const handleBuyNow = (product) => {
  addToCart(product);
  router.push("/cart");
};

const handleCategoryChange = (e) => {
  setCategory(e.target.value);
};

const handlePriceChange = (e) => {
  const { name, value } = e.target;
  if (name === "minPrice") {
    setPriceRange([Number(value), priceRange[1]]);
  } else if (name === "maxPrice") {
    setPriceRange([priceRange[0], Number(value)]);
  }
};

const handleFabricChange = (e) => {
  setFabric(e.target.value);
};

const handleBedSizeChange = (e) => {
  setBedSize(e.target.value);
};

useEffect(() => {
  const filtered = products.filter((product) => {
    const isCategoryMatch =
      category === "all" || product.category === category;
    const isPriceInRange =
      product.price >= priceRange[0] && product.price <= priceRange[1];
    const isFabricMatch = fabric === "all" || product.fabric === fabric;
    const isBedSizeMatch = bedSize === "all" || product.bedSize === bedSize;

    return (
      isCategoryMatch && isPriceInRange && isFabricMatch && isBedSizeMatch
    );
  });

  setFilteredProducts(filtered);
}, [category, priceRange, fabric, bedSize, products]);

// Function to navigate to the previous image
const goToPrevious = () => {
  setCurrentIndex((prevIndex) =>
    prevIndex === 0 ? images.length - 1 : prevIndex - 1
    );
  };

  // Function to navigate to the next image
  const goToNext = () => {
    setCurrentIndex((prevIndex) =>
      prevIndex === images.length - 1 ? 0 : prevIndex + 1
    );
  };

  return (
    <div className="h-auto w-auto bg-gray-200 flex flex-col justify-center items-center py-10">
      {/* Hero Image Section */}
      <div className="bg-black flex mx-auto w-[80vw] h-[70vh] justify-center overflow-hidden relative">
        <img
          src={images[currentIndex]}
          className="w-full h-full object-cover"
          alt={`Slide ${currentIndex + 1}`}
        />
        {/* Left Arrow */}
        <button
          onClick={goToPrevious}
          className="absolute left-5 top-1/2 transform -translate-y-1/2 bg-black text-white p-2 rounded-full hover:bg-gray-700 transition"
        >
          &#8249;
        </button>
        {/* Right Arrow */}
        <button
          onClick={goToNext}
          className="absolute right-5 top-1/2 transform -translate-y-1/2 bg-black text-white p-2 rounded-full hover:bg-gray-700 transition"
        >
          &#8250;
        </button>
        <div className="absolute bottom-5 left-1/2 transform -translate-x-1/2 space-x-4">
          <button
            onClick={() => handleBuyNow(products[0])} // You can set a specific product for Buy Now
            className="bg-red-500 text-white px-6 py-2 rounded-lg hover:bg-red-600 transition"
          >
            Buy Now
          </button>
        </div>
      </div>

      {/* Slogan Section */}
      <div className="mt-8 text-center">
        <h2 className="text-4xl font-extrabold text-red-600">
          Feel the Luxury, Love the Comfort.
        </h2>
  <p className="text-xl text-gray-700 mt-2 font-medium">
    Discover the perfect blend of comfort and elegance with our premium
    collection.
  </p>
</div>

{/* Filter Section */}
<div className="w-[80vw] max-w-8xl mt-8 flex gap-8">
  <div className="bg-white p-6 rounded-lg shadow-lg w-1/4">
    <h3 className="text-xl font-semibold mb-4">Filters</h3>

    {/* Category Filter */}
    <div className="mb-4">
      <label className="block text-sm font-medium mb-2">Category</label>
      <select
        value={category}
        onChange={handleCategoryChange}
        className="w-full p-2 border border-gray-300 rounded-md"
      >
        <option value="all">All</option>
        <option value="bedsheets">Bedsheets</option>
        <option value="pillow">Pillow</option>
        <option value="mattress">Mattress</option>
        <option value="blanket">Blanket</option>
      </select>
    </div>

    {/* Price Filter */}
    <div className="mb-4">
      <label className="block text-sm font-medium mb-2">
        Price Range
      </label>
      <div className="flex gap-4">
        <input
          type="number"
          name="minPrice"
          value={priceRange[0]}
          onChange={handlePriceChange}
          className="w-1/2 p-2 border border-gray-300 rounded-md"
          placeholder="Min"
        />
        <input
          type="number"
          name="maxPrice"
          value={priceRange[1]}
          onChange={handlePriceChange}
          className="w-1/2 p-2 border border-gray-300 rounded-md"
          placeholder="Max"
        />
      </div>
    </div>
```

```
          {/* Fabric Filter */}
          <div className="mb-4">
            <label className="block text-sm font-medium mb-2">Fabric</label>
            <select
              value={fabric}
              onChange={handleFabricChange}
              className="w-full p-2 border border-gray-300 rounded-md"
            >
              <option value="all">All</option>
              <option value="cotton">Cotton</option>
              <option value="silk">Silk</option>
              <option value="polyester">Polyester</option>
              <option value="linen">Linen</option>
            </select>
          </div>

          {/* Bed Size Filter */}
          <div className="mb-4">
            <label className="block text-sm font-medium mb-2">Bed Size</label>
            <select
              value={bedSize}
              onChange={handleBedSizeChange}
              className="w-full p-2 border border-gray-300 rounded-md"
            >
              <option value="all">All</option>
              <option value="single">Single</option>
              <option value="double">Double</option>
            </select>
          </div>
        </div>

        {/* Card Section with 3 Columns */}
        <div className="w-3/4 grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 gap-6">
          {filteredProducts.map((product) => (
            <div
              key={product.id}
              className="bg-white p-5 rounded-lg shadow-lg flex flex-col items-center hover:scale-105
hover:shadow-lg transition duration-300 ease-in-out"
            >
              <img
                src={product.image}
                alt={product.title}
                className="w-full h-[200px] object-cover mb-4 rounded-md"
              />
              <h2 className="text-2xl font-bold mb-2">{product.title}</h2>
              <p className="text-gray-600 mb-2">Price: ₹{product.price}</p>
              <p className="text-gray-600 mb-4 text-center">
                {product.description}
              </p>
              <button
```

```
                onClick={() => handleAddToCart(product)}
                className="bg-red-500 text-white px-6 py-2 rounded-lg hover:bg-red-600 transition"
              >
                Add to Cart
              </button>
            </div>
          ))}
        </div>
      </div>
    </div>
  );
};

export default page;
```

PRIVACY

```
"use client";
import React from "react";

const page = () => {
  return (
    <div className="min-h-screen bg-gray-100 py-10">
      <div className="container mx-auto px-4">
        {/* Title Section */}
        <div className="text-center mb-12">
          <h1 className="text-3xl font-semibold text-gray-800">Privacy Policy</h1>
          <p className="text-gray-600 mt-2">Your privacy is important to us</p>
        </div>

        {/* Privacy Policy Content */}
        <div className="bg-white p-8 rounded-lg shadow-lg max-w-4xl mx-auto space-y-8">
          <section>
            <h2 className="text-2xl font-semibold text-gray-800">Introduction</h2>
            <p className="text-gray-600 mt-2">
              This Privacy Policy explains how we collect, use, and protect your information when you visit our website or use our services. By using our site, you agree to the collection and use of information in accordance with this policy.
            </p>
          </section>

          <section>
            <h2 className="text-2xl font-semibold text-gray-800">Information Collection</h2>
            <p className="text-gray-600 mt-2">
              We collect various types of information in order to provide and improve our services to you. This includes:
            </p>
            <ul className="list-disc list-inside text-gray-600 mt-2">
              <li>Personal identification information (name, email address, etc.)</li>
              <li>Non-personal identification information (browser type, IP address, etc.)</li>
              <li>Cookies and usage data</li>
            </ul>
          </section>

          <section>
            <h2 className="text-2xl font-semibold text-gray-800">How We Use Your Data</h2>
            <p className="text-gray-600 mt-2">
              We use the collected data for various purposes, including:
            </p>
            <ul className="list-disc list-inside text-gray-600 mt-2">
              <li>To personalize and improve your experience</li>
              <li>To send updates, offers, and other relevant information</li>
              <li>To monitor and analyze the usage of our service</li>
              <li>To comply with legal obligations and resolve disputes</li>
            </ul>
          </section>

          <section>
            <h2 className="text-2xl font-semibold text-gray-800">Cookies</h2>
            <p className="text-gray-600 mt-2">
              We use cookies to enhance your experience on our website. Cookies are small files stored on your device that help us remember your preferences, analyze web traffic, and improve the functionality of the site.
            </p>
            <p className="text-gray-600 mt-2">
              You can choose to disable cookies by adjusting your browser settings, but this may limit some functionality of our website.
            </p>
          </section>

          <section>
            <h2 className="text-2xl font-semibold text-gray-800">Data Security</h2>
            <p className="text-gray-600 mt-2">
              We take the security of your personal data seriously and use reasonable measures to protect it. However, please note that no method of electronic transmission or storage is 100% secure. While we strive to protect your information, we cannot guarantee its absolute security.
            </p>
          </section>

          <section>
            <h2 className="text-2xl font-semibold text-gray-800">Your Rights</h2>
            <p className="text-gray-600 mt-2">
              You have the right to access, update, or delete your personal information. If you would like to exercise these rights, please contact us using the information provided below.
            </p>
          </section>

          <section>
            <h2 className="text-2xl font-semibold text-gray-800">Contact Us</h2>
            <p className="text-gray-600 mt-2">
              If you have any questions or concerns about this Privacy Policy, please contact us:
```

```
            </p>
            <div className="mt-4 text-gray-600">
              <p>Email: contact@timelessheaven.com</p>
              <p>Phone: +1 234 567 890</p>
              <p>Address: 123 Timeless Ave, Heaven City, HC 12345</p>
            </div>
          </section>
        </div>
      </div>
    </div>
  );
};

export default page;
```

PROFILE

Module.css

```
.link {
  cursor: pointer;
}

.burger {
  width: 20px;
  height: 16px;
  flex-direction: column;
  justify-content: space-between;
  cursor: pointer;
  display: none;
}

.line {
  width: 100%;
  height: 2px;
  background-color: var(--textColor);
}

.responsiveMenu{
  position: absolute;
  top: 100px;
  left: 0;
  background-color: var(--bg);
  height: calc(100vh - 100px);
  width: 100%;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  gap: 50px;
  font-size: 36px;
  z-index: 999;
}

@media screen and (max-width: 640px) {
  .burger {
    display: flex;
  }

  .link{
    display: none;
  }
}
```

Profile.jsx

```
"use client";
import Link from "next/link";
// import styles from "./authLinks.module.css";
import { useState } from "react";
import { signOut, useSession } from "next-auth/react";

const AuthLinks = () => {
  const [open, setOpen] = useState(false);

  const { status } = useSession();

  return (
    <>
      {status === "unauthenticated" ? (
        <Link href="/login" className={"styles.link"}>
          Login
        </Link>
      ) : (
        <>
          <Link href="/logout" className={"styles.link"}>
            Write
          </Link>
          <span className={"styles.link"} onClick={signOut}>
            Logout
          </span>
        </>
      )}
      <div className={"styles.burger"} onClick={() => setOpen(!open)}>
        <div className={"styles.line"}></div>
        <div className={"styles.line"}></div>
        <div className={"styles.line"}></div>
      </div>
      {open && (
        <div className={"styles.responsiveMenu"}>
          <Link href="/">Homepage</Link>
          <Link href="/">About</Link>
          <Link href="/">Contact</Link>
          {status === "notauthenticated" ? (
            <Link href="/login">Login</Link>
          ) : (
            <>
              <Link href="/write">Write</Link>
              <span className={"styles.link"}>Logout</span>
            </>
          )}
        </div>
      )}
    </>
  );
};

export default AuthLinks;
```

TERMS

```
"use client";
import React from "react";

const page = () => {
  return (
    <div className="min-h-screen bg-gray-100 py-10">
      <div className="container mx-auto px-4">
        {/* Title Section */}
        <div className="text-center mb-12">
          <h1 className="text-3xl font-semibold text-gray-800">Terms of Service</h1>
          <p className="text-gray-600 mt-2">Please read our terms carefully</p>
        </div>

        {/* Terms of Service Content */}
        <div className="bg-white p-8 rounded-lg shadow-lg max-w-4xl mx-auto space-y-8">
          <section>
            <h2 className="text-2xl font-semibold text-gray-800">Introduction</h2>
            <p className="text-gray-600 mt-2">
              Welcome to Timeless Heaven! By accessing or using our website, you agree to comply with and be bound by these Terms of Service. If you do not agree to these terms, please do not use our website.
            </p>
          </section>

          <section>
            <h2 className="text-2xl font-semibold text-gray-800">User Responsibilities</h2>
            <p className="text-gray-600 mt-2">
              As a user of our website, you agree to use our services in compliance with applicable laws and regulations. You also agree not to engage in:
            </p>
            <ul className="list-disc list-inside text-gray-600 mt-2">
              <li>Using the site for any unlawful purpose.</li>
              <li>Interfering with or disrupting the functioning of the website.</li>
              <li>Posting false or misleading information.</li>
            </ul>
          </section>

          <section>
            <h2 className="text-2xl font-semibold text-gray-800">Intellectual Property</h2>
            <p className="text-gray-600 mt-2">
              All content, including text, images, logos, and trademarks, on this website is the property of Timeless Heaven and is protected by intellectual property laws. You may not copy, modify, or distribute any of the content without explicit permission.
            </p>
          </section>

          <section>
            <h2 className="text-2xl font-semibold text-gray-800">Limitation of Liability</h2>
            <p className="text-gray-600 mt-2">
              Timeless Heaven will not be liable for any direct, indirect, incidental, or consequential damages resulting from the use or inability to use our services. We do not guarantee that our website will be error-free or uninterrupted.
            </p>
          </section>

          <section>
            <h2 className="text-2xl font-semibold text-gray-800">Governing Law</h2>
            <p className="text-gray-600 mt-2">
              These Terms of Service are governed by the laws of the jurisdiction in which Timeless Heaven operates. Any disputes arising from these terms will be resolved in the competent courts of that jurisdiction.
            </p>
          </section>

          <section>
            <h2 className="text-2xl font-semibold text-gray-800">Changes to Terms</h2>
            <p className="text-gray-600 mt-2">
              We reserve the right to modify these Terms of Service at any time. Any changes will be posted on this page, and the updated terms will be effective immediately upon posting.
            </p>
          </section>

          <section>
            <h2 className="text-2xl font-semibold text-gray-800">Contact Us</h2>
            <p className="text-gray-600 mt-2">
              If you have any questions regarding these Terms of Service, please contact us at:
            </p>
            <div className="mt-4 text-gray-600">
              <p>Email: contact@timelessheaven.com</p>
              <p>Phone: +1 234 567 890</p>
              <p>Address: 123 Timeless Ave, Heaven City, HC 12345</p>
```

```
          </div>
        </section>
      </div>
    </div>
  </div>
 );
};

export default page;
```

LAYOUT

```
import localFont from "next/font/local";
import "./globals.css";
import Header from "@/components/Header";
import Footer from "@/components/Footer";
import { CartProvider } from "@/context/CartContext";
import AuthProvider from "@/providers/AuthProvider";

const geistSans = localFont({
 src: "./fonts/GeistVF.woff",
 variable: "--font-geist-sans",
 weight: "100 900",
});
const geistMono = localFont({
 src: "./fonts/GeistMonoVF.woff",
 variable: "--font-geist-mono",
 weight: "100 900",
});

export const metadata = {
 title: "Timeless Heaven",
 description: "Generated by create next app",
};

export default function RootLayout({ children }) {
 return (
  <html lang="en">
   <body
    className={`${geistSans.variable} ${geistMono.variable} antialiased`}
   >
    <AuthProvider>
     <Header />
     <CartProvider>{children}</CartProvider>
     <Footer />
    </AuthProvider>
   </body>
  </html>
 );
}
```

PAGE.JSX

```
/* eslint-disable @next/next/no-img-element */
"use client";
import { useCart } from "@/context/CartContext";
import { useRouter } from "next/navigation";
import React, { useState, useEffect } from "react";

const Home = () => {
  const images = [
    "/hp1.jpg",
    "/p1.webp",
    "/bt1.jpg",
    "/Velvet new _page-0001.jpg",
    "/img23.jpg",
    "/p5.jpg",
    "/bt4.jpg",
  ];

  const [currentIndex, setCurrentIndex] = useState(0);
  const [products, setProducts] = useState([]);
  const [filteredProducts, setFilteredProducts] = useState([]);
  const [searchQuery, setSearchQuery] = useState(""); // New state for search
  const [priceRange, setPriceRange] = useState([0, 10000]); // State for price range filter
  const [category, setCategory] = useState(""); // State for category filter
  const { addToCart } = useCart();
  const router = useRouter();

  useEffect(() => {
    const interval = setInterval(() => {
      setCurrentIndex((prevIndex) =>
        prevIndex === images.length - 1 ? 0 : prevIndex + 1
      );
    }, 3000);

    const fetchProducts = async () => {
      const res = await fetch("/products.json");
      const data = await res.json();
      setProducts(data);
      setFilteredProducts(data); // Initialize with all products
    };

    fetchProducts();
    return () => clearInterval(interval);
  }, [images.length]);

  useEffect(() => {
    // Filter products based on search query, price range, and category
    const filtered = products.filter((product) => {
      const matchesSearch =
        product.title.toLowerCase().includes(searchQuery.toLowerCase()) ||
        product.description.toLowerCase().includes(searchQuery.toLowerCase());
      const matchesPrice =
        product.price >= priceRange[0] && product.price <= priceRange[1];
      const matchesCategory =
        category ? product.category.toLowerCase() === category.toLowerCase() : true;

      return matchesSearch && matchesPrice && matchesCategory;
    });
    setFilteredProducts(filtered);
  }, [searchQuery, priceRange, category, products]);

  const handleAddToCart = (product) => {
    addToCart(product);
    router.push("/cart");
  };

  const handleBuyNow = (product) => {
    addToCart(product);
    router.push("/cart");
  };

  // Function to navigate to the previous image
  const goToPrevious = () => {
    setCurrentIndex((prevIndex) =>
      prevIndex === 0 ? images.length - 1 : prevIndex - 1
    );
  };

  // Function to navigate to the next image
  const goToNext = () => {
    setCurrentIndex((prevIndex) =>
      prevIndex === images.length - 1 ? 0 : prevIndex + 1
    );
  };

  return (
    <div className="h-auto w-auto bg-gray-200 flex flex-col justify-center items-center py-10">
      {/* Hero Image Section */}
      <div className="bg-black flex mx-auto w-[80vw] h-[70vh] justify-center overflow-hidden
relative">
        <img
          src={images[currentIndex]}
          className="w-full h-full object-cover"
          alt={`Slide ${currentIndex + 1}`}
        />
        {/* Left Arrow */}
        <button
          onClick={goToPrevious}
          className="absolute left-5 top-1/2 transform -translate-y-1/2 bg-black text-white p-2 rounded-
full hover:bg-gray-700 transition"
        >
```

```
          &#8249;
        </button>
        {/* Right Arrow */}
        <button
          onClick={goToNext}
          className="absolute right-5 top-1/2 transform -translate-y-1/2 bg-black text-white p-2 rounded-full hover:bg-gray-700 transition"
        >
          &#8250;
        </button>
        <div className="absolute bottom-5 left-1/2 transform -translate-x-1/2 space-x-4">
          <button
            onClick={() => handleBuyNow(products[0])} // You can set a specific product for Buy Now
            className="bg-red-500 text-white px-6 py-2 rounded-lg hover:bg-red-600 transition"
          >
            Buy Now
          </button>
        </div>
      </div>

      {/* Slogan Section */}
      <div className="mt-8 text-center">
        <h2 className="text-4xl font-extrabold text-red-600">
          Feel the Luxury, Love the Comfort.
        </h2>
        <p className="text-xl text-gray-700 mt-2 font-medium">
          Discover the perfect blend of comfort and elegance with our premium collection.
        </p>
      </div>

      {/* Search Section */}
      <div className="w-[80vw] max-w-8xl mt-8 mb-6">
        <input
          type="text"
          value={searchQuery}
          onChange={(e) => setSearchQuery(e.target.value)}
          placeholder="Search for products..."
          className="w-full p-4 text-lg border border-gray-300 rounded-lg focus:outline-none focus:ring-2 focus:ring-red-500"
        />
      </div>

      {/* Filter Section */}
      <div className="w-[80vw] max-w-8xl mt-4 mb-8 flex flex-wrap justify-between gap-4">
        {/* Price Range Filter */}
        <div className="flex flex-col w-full sm:w-[45%] md:w-[30%] lg:w-[22%]">
          <label className="text-lg font-medium mb-2">Price Range</label>
          <input
            type="range"
            min="0"
            max="10000"
            step="100"
            value={priceRange[0]}
            onChange={(e) => setPriceRange([Number(e.target.value), priceRange[1]])}
            className="mb-2"
          />
          <input
            type="range"
            min="0"
            max="10000"
            step="100"
            value={priceRange[1]}
            onChange={(e) => setPriceRange([priceRange[0], Number(e.target.value)])}
          />
          <p>
            ₹{priceRange[0]} - ₹{priceRange[1]}
          </p>
        </div>

        {/* Category Filter */}
        <div className="flex flex-col w-full sm:w-[45%] md:w-[30%] lg:w-[22%]">
          <label className="text-lg font-medium mb-2">Category</label>
          <select
            value={category}
            onChange={(e) => setCategory(e.target.value)}
            className="p-2 border rounded-md"
          >
            <option value="">All Categories</option>
            <option value="Mattress">Mattress</option>
            <option value="Pillow">Pillow</option>
            <option value="Bed">Bed</option>
            <option value="Bedding">Bedding</option>
          </select>
        </div>
      </div>

      {/* Card Section */}
      <div className="w-[80vw] max-w-8xl mt-8 flex flex-wrap justify-center gap-6">
        {filteredProducts.map((product) => (
          <div
            key={product.id}
            className="bg-white w-full sm:w-[45%] md:w-[30%] lg:w-[22%] p-5 rounded-lg shadow-lg flex flex-col items-center hover:scale-105 hover:shadow-lg transition duration-300 ease-in-out"
          >
            <img
              src={product.image}
              alt={product.title}
              className="w-full h-[200px] object-cover mb-4 rounded-md"
            />
            <h2 className="text-2xl font-bold mb-2">{product.title}</h2>
            <p className="text-gray-600 mb-2">Price: ₹{product.price}</p>
            <p className="text-gray-600 mb-4 text-center">{product.description}</p>
            <button
              onClick={() => handleAddToCart(product)}
              className="bg-red-500 text-white px-6 py-2 rounded-lg hover:bg-red-600 transition"
            >
              Add to Cart
            </button>
          </div>
        ))}
      </div>
    </div>
  );
};

export default Home;
```

CHAPTER 11

TESTING

10.1 Introduction

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionalities of components, sub-assemblies, and/or a finished product it is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptablemanner. There are various types of tests. Each test type addresses a specific testing requirement.

10.2 Types of Testing

10.2.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design,the module. The unit testing, we have is white box oriented and some modules the steps are conducted in parallel.

10.2.2 Integration Testing

Testing is done for each module. After testing all the modules, the modules are integrated and testing of the final system is done with the test data, specially designed toshow that the system will operate successfully in all its aspects conditions. Thus, the system testing is a confirmation that all is correct and an opportunity to show the user that the system works.

The purpose of integration testing is to verify functional, performance and reliability requirements placed on major design items. These "design items", i.e. assemblages (or groups of units), are exercised through their interfaces using black box testing, success and error cases being simulated via appropriate parameter and data inputs. Simulated usage of shared data areas and inter-process communication is tested and individual subsystems are exercised through their input interface.

10.2.3 System Testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. Anexample of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process linksand integration points.

FUTURE SCOPE AND FURTHER ENHANCEMENT OF THE PROJECT

- The future scope of E commerce websites lies in leveraging advanced technologies such as AI, VR, and AR to offer personalized and immersive shopping experiences. These websites will focus on sustainability, eco-friendly options, and authentic cultural exchanges. They will also provide AI-driven assistance, AR-enhanced navigation, and digital storytelling platforms to inspire and guide customers in their journeys.
- Another enhancement would be allowed for Customer also.

CONCLUSION & REFERNCES

The successful design and development of our project on Web Based Timeless Haven using React mark a significant milestone in our journey. Through this endeavor, we have gained invaluable experience in website development, honing our skills and understanding of the intricacies of creating engaging online platforms. As we move forward, we anticipate further growth and learning opportunities, leveraging our newfound expertise to continue innovating and delivering exceptional experiences for our users. Our commitment to excellence and passion for storytelling will drive us to push boundaries, inspiring wanderlust and fostering connections through the captivatingworld of luxury and comfort.

References

Coding phase: **-**

1. React

Referenced Sites:

www.w3school. com

www. https://react.dev/learn

https://nextjs.org/

https://tailwindcss.com/

https://github.com/

https://developer.mozilla.org/en-US/docs/Web/JavaScript

https://vercel.com/aamirs-projects-0fc0307d