NANDHA ENGINEERING COLLEGE
ERODE–638052 (Autonomous)

(Affiliated to Anna University, Chennai)

PROJECT BASED LEARNING REPORT

Submitted by

KEERTHIVARSAN D (22AI022)

in partial fulfilment for the award of the degree of

B.Tech (ARTIFICIAL INTELLIGENCE AND DATA SCIENCE)

22AIC15 – FULL STACK DEVELOPMENT

PROJECT BASED LEARNING ON
CAKE BAKERY WEBSITE

NANDHA ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

This is to certify that the project work entitled**"CAKE BAKERY WEBSITE"** is the Bonafide work of **KEERTHIVARSAN D(22AI022)** who carried out the work under my supervision.

Signature of the HOD

Dr. P. Karunakaran,
Head of the Department,
Department of AI & DS,
Nandha Engineering College,
Erode – 638052.

Signature of the Supervisor

Ms. M. Senthamarai,
Assistant Professor,
Department of AI & DS,
Nandha Engineering College,
Erode – 638052.

Submitted for the project Viva Voce examination held on_____

ACKNOWLEDGEMENT

I express my thanks to our beloved Chairman of Sri Nandha Educational Trust **Thiru.V.Shanmugan** and our beloved Secretaries, **Thiru**. **S.NandhakumarPradeep** of Sri Nandha Educational Trust and **Thiru**. **S**.**Thirumoorthi** of Nandha Educational Institutions for providing me all the basic amenities to complete the course successfully.

I specially thank **Dr. S.Arumugam**, Chief Executive Officer of Nandha Educational Institutions for his affection and support in all aspects that have made me complete the course successfully.

I wish to convey my earnest gratefulness to our cherished Principal of Nandha Engineering College, **Dr. U. S.Ragupathy**, M.E., Ph.D., for his constant support in my successful completion of my project work.

I articulate my genuine and sincere thanks to our dear hearted Head of the Department of Artificial Intelligence and Data Science **Dr. P. Karunakaran**, M.E., Ph.D., and **Ms. T. Indhumathi**,M.E., Assistant Professor,Department of Artificial Intelligence and Data Science who has been the key spring of motivation to me throughout the completion of my course and my project work.

I wish to convey my heartfelt thanks to my beloved Project Supervisor **Ms. M. Senthamarai**, M.E., Assistant Professor,Department of Artificial Intelligence and Data Science for her continuous monitoring of the project work.

I am very much gratified to all teaching and non-teaching staff of our department who were direct and indirect strokes throughout my progress. I would like to acknowledge my heartfelt thanks to my parents and my friends who have supported me with their unconditional love and encouragement. Finally, I would like to thank the Almighty for his blessings.

TABLE OF CONTENTS

CHAPTER 1

ABSTRACT

The Cake Bakery Website is a full-stack web application developed using the MERN (MongoDB, Express.js, React.js, Node.js) stack. This project is a modern, responsive, and scalable web application developed using the latest front-end technologies, including Vite, TypeScript, and Tailwind CSS. The goal of the project is to provide a clean, efficient, and developer-friendly framework that can serve as a foundation for dynamic web applications such as dashboards, real-time monitoring systems, and admin panels.

The use of Vite as the build tool significantly enhances the development experience by offering fast hot module replacement (HMR), lightning-fast cold starts, and optimized production builds. This makes the application extremely efficient during development and ready for scalable deployment in production environments.

At the core of the application is TypeScript, which adds static typing to JavaScript and helps catch potential bugs at compile-time. This improves the code's maintainability, readability, and robustness, making it ideal for large-scale applications or teams working collaboratively on complex systems.

The project also incorporates Tailwind CSS, a utility-first CSS framework that enables rapid UI development without writing custom CSS for every component. This results in consistent design practices and a responsive user interface that works seamlessly across different devices and screen sizes.

Additionally, the project structure includes environment-specific configuration (.env), linting (eslint.config.js), and build customization (postcss.config.js and tailwind.config.js), making it highly configurable and adaptable to a variety of use cases.

This project lays a strong foundation for future enhancements like API integration, real-time data streaming, and full-stack extension. It is a testament to modern web development practices focused on performance, scalability, and developer productivity.

CHAPTER 2

SOFTWARE AND HARDWARE REQUIREMENTS

HARDWARE REQUIREMENTS

System	: ASUS
Processor	: INTEL I5
RAM	: 8GB RAM
Hard Disk Capacity	: 512 GB

SOFTWARE REQUIREMENTS

Operating System	: Windows 11
Frontend	: React JS
Backend	: Node.js with express
Database	: MongoDB
IDE Used	: Visual Studio Code
Browser	: Chrome/ Microsoft Edge

CHAPTER 3

SOFTWARE DESCRIPTION

React JS (Frontend Framework):

React JS is an open-source JavaScript library developed by Facebook for building fast, interactive user interfaces for web applications. It follows a component-based architecture, allowing developers to build encapsulated UI components that manage their own state and can be composed to create complex UIs. React uses a virtual DOM to efficiently update and render components in response to data changes, resulting in high performance and responsiveness. It is widely adopted for its scalability, flexibility, and active community support.

Node.js with Express (Backend Framework):

Node.js is a powerful, open-source JavaScript runtime environment built on Chrome's V8 engine, designed for building scalable network applications. It uses an event-driven, non-blocking I/O model which makes it lightweight and efficient-perfect for real-time applications.

Express.js is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications. It simplifies routing, middleware integration, and handling HTTP requests and responses. Together, Node.js and Express form a powerful backend platform for developing full stack web applications.

MongoDB (Database):

MongoDB is a NoSQL, document-oriented database designed for high availability, scalability, and performance. It stores data in flexible, JSON-like documents, making it easier to work with complex data structures compared to traditional relational databases. MongoDB allows for horizontal scaling and is widely used in modern web applications for storing user information, bookings, and transactional records in a schema-less format. It integrates seamlessly with Node.js using the Mongoose library, enabling structured data modelling.

Visual Studio Code (IDE):

Visual Studio Code (VS Code) is a lightweight yet powerful source-code editor developed by Microsoft. It supports JavaScript, TypeScript, Node.js, and many other programming languages and frameworks. VS Code offers intelligent code completion (IntelliSense), syntax highlighting, debugging tools, version control integration (Git), and a wide range of extensions for web development. Its built-in terminal and compatibility with modern development tools make it a preferred choice for full stack developers.

CHAPTER 4

PROJECT DESCRIPTION

1. Contents of the Project

The key files and folders included in the project are:

- **Frontend (React)**
  - /src/components – Contains reusable components (e.g., Reviews, DashProfile, Header, Orders)
  - /src/pages – Contains individual pages (e.g., Home.tsx, Orders.tsx, Login.tsx, Checkout.tsx, Cart.tsx, Product.txt)
  - /src/App.jsx – Main routing and layout logic
  - /src/api – Axios functions for API communication
- **Backend (Node.js + Express)**
  - /routes – Contains route files for users, Orders, Reviews (e.g., user.route.js, post.route.js)
  - /controllers – Business logic handlers for the routes
  - /models – Mongoose schemas for User, Orders, Reviews
  - /utils – contains **helper functions** or **reusable logic** used across the application.
  - index.js – Main server entry point
- **Database (MongoDB), Firebase**
  - Collections: Users, Orders, Reviews
  - Stores: Login credentials, Order data, Review data

2. Project Initialization and Setup

- The frontend is initialized using **Vite** with **TypeScript** support.
- The backend server is built using **Node.js** and **Express**, initialized via npm init and set up in the server/index.js file.
- Environment-specific variables are securely stored in a .env file to manage sensitive data (e.g., API base URLs).

- The development workflow includes using npm/yarn to install and manage dependencies.

3. User Authentication

- **User Registration & Login**: Users can sign up and log in using email and password through secure forms.
- **Token-Based Authentication**: Upon login, the server issues a JWT token to maintain user sessions securely.
- **Protected Routes**: API endpoints are secured using middleware that verifies the token before granting access.
- **Role Management**: The system supports role-based access, allowing admins to manage catalogues and users.

4. Cake Bakery Website(CRUD Operations)

- Users can:
  - Browse and read all thecatalogues
  - View detailed price content, View Integrated Projects
  - Sign up, log in, and update their profile information
  - Log out or delete their account
- Admin can:
  - Create new catalogues using a rich-text editor
  - Edit or delete existing catalogues,
  - View all users, orders, and likes from the admin dashboard
  - Delete orders, and likes

5. User Activity and Content Management

- **Users can** read catalogues and manage their personal profile information.
- **User activity data** (e.g., orders, likes, and profile updates) is stored in the MongoDB Users and reviewscollections.
- **Users can** update their profiles, while only **admins** have permission to manage content and moderate user interactions through the Admin Dashboard.

6. Frontend Features

- Clean and interactive UI designed with React and Tailwind CSS
- Pages include:
    - Home – Overview of the bakeryplatform
    - All catalogues – Grid view of all published category entries
    - Order Details – Order the cakes based on their choice
    - Profile Page – Allows users to view and edit their personal info
    - Cart– For users can add their product and place order
- Axios is used for frontend-backend communication.
- Firebase Auth Integration enables login/logout and user session tracking on the frontend.

7. API Routes and Integration

- GET /api/posts, GET /api/posts/:id– Fetch all published catalogues, and by profile
- POST /api/posts– Create a new catalogues (admin only)
- PUT /api/posts/:id - Update a catalogues (admin only)
- DELETE /api/posts/:id - Delete a catalogues(admin only)
- POST /api/users,GET /api/users/:id - Create or update user profile, get user details by profile
- POST /auth/register, POST /auth/login – Authentication routes

Integration with the frontend would occur via Axios or fetch inside the React components.

CHAPTER 5

APPENDIX

```
import React, { useState, useEffect } from 'react';
import { Link } from 'react-router-dom';
import { Star, ArrowRight, Heart, Award, Truck, Shield } from 'lucide-react';
import axios from 'axios';

interface Product {
  _id: string;
  name: string;
  description: string;
  price: number;
  originalPrice?: number;
  image: string;
  rating: number;
  category: string;
  tags: string[];
}

const Home: React.FC = () => {
  const [featuredProducts, setFeaturedProducts] = useState<Product[]>([]);
  const [loading, setLoading] = useState(true);

  const API_URL = import.meta.env.VITE_API_URL || 'http://localhost:5000/api';

  useEffect(() => {
    fetchFeaturedProducts();
  }, []);
```

```
const fetchFeaturedProducts = async () => {
  try {
    const response = await axios.get(`${API_URL}/products?sort=rating`);
    setFeaturedProducts(response.data.slice(0, 6));
  } catch (error) {
    console.error('Error fetching products:', error);
  } finally {
    setLoading(false);
  }
};

const features = [
  {
    icon: Award,
    title: 'Premium Quality',
    description: 'Only the finest ingredients for exceptional taste'
  },
  {
    icon: Truck,
    title: 'Fast Delivery',
    description: 'Fresh cakes delivered to your doorstep in 2-4 hours'
  },
  {
    icon: Shield,
    title: 'Secure Payment',
    description: '100% secure transactions with multiple payment options'
  },
  {
```

```
    icon: Heart,
    title: '5-Star Reviews',
    description: 'Loved by thousands of satisfied customers'
  }
];

const categories = [
  {
    name: 'Chocolate Cakes',
    image: 'https://images.pexels.com/photos/291528/pexels-photo-291528.jpeg',
    href: '/products?category=chocolate'
  },
  {
    name: 'Vanilla Cakes',
    image: 'https://images.pexels.com/photos/1126359/pexels-photo-1126359.jpeg',
    href: '/products?category=vanilla'
  },
  {
    name: 'Red Velvet',
    image: 'https://images.pexels.com/photos/1721932/pexels-photo-1721932.jpeg',
    href: '/products?category=red-velvet'
  },
  {
    name: 'Fruit Cakes',
    image: 'https://images.pexels.com/photos/1070850/pexels-photo-1070850.jpeg',
    href: '/products?category=fruit'
  }
];
```

```
  return (
<div className="min-h-screen">
    {/* Hero Section */}
<section className="relative min-h-screen flex items-center justify-center overflow-hidden">
<div className="absolute inset-0 bg-gradient-to-br from-amber-50 via-rose-50 to-purple-50"></div>
<div className="absolute inset-0 bg-[url('https://images.pexels.com/photos/1126359/pexels-photo-1126359.jpeg')] bg-cover bg-center opacity-10"></div>

<div className="relative z-10 max-w-7xl mx-auto px-4 sm:px-6 lg:px-8 text-center">
<div className="max-w-4xl mx-auto">
<h1 className="text-5xl md:text-7xl font-bold mb-6">
<span className="bg-gradient-to-r from-amber-600 via-rose-600 to-purple-600 bg-clip-text text-transparent">
        Sweet Delights
</span>
</h1>
<p className="text-xl md:text-2xl text-gray-700 mb-8 leading-relaxed">
      Crafting moments of joy with every bite. Premium handmade cakes delivered fresh to your doorstep.
</p>
<div className="flex flex-col sm:flex-row gap-4 justify-center items-center">
<Link
        to="/products"
        className="px-8 py-4 bg-gradient-to-r from-amber-500 to-rose-500 text-white rounded-full font-semibold text-lg hover:from-amber-600 hover:to-rose-600 transform hover:scale-105 transition-all duration-200 shadow-lg"
>
        Order Now <ArrowRight className="inline w-5 h-5 ml-2" />
</Link>
<Link
```

```
            to="/products"
            className="px-8 py-4 border-2 border-amber-500 text-amber-600 rounded-full font-semibold text-lg hover:bg-amber-50 transition-all duration-200"
>
            View Menu
</Link>
</div>
</div>
</div>
</section>

    {/* Features Section */}
<section className="py-20 bg-white">
<div className="max-w-7xl mx-auto px-4 sm:px-6 lg:px-8">
<div className="text-center mb-16">
<h2 className="text-4xl font-bold text-gray-900 mb-4">Why Choose Sweet Delights?</h2>
<p className="text-xl text-gray-600 max-w-3xl mx-auto">
          We're committed to delivering exceptional quality and service with every order
</p>
</div>

<div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-4 gap-8">
        {features.map((feature, index) => (
<div key={index} className="text-center group">
<div className="w-16 h-16 bg-gradient-to-br from-amber-100 to-rose-100 rounded-full flex items-center justify-center mx-auto mb-4 group-hover:scale-110 transition-transform duration-200">
<feature.icon className="w-8 h-8 text-amber-600" />
</div>
<h3 className="text-xl font-semibold text-gray-900 mb-2">{feature.title}</h3>
```

```
<p className="text-gray-600">{feature.description}</p>
</div>
        ))}
</div>
</div>
</section>

    {/* Categories Section */}
<section className="py-20 bg-gradient-to-br from-gray-50 to-amber-50">
<div className="max-w-7xl mx-auto px-4 sm:px-6 lg:px-8">
<div className="text-center mb-16">
<h2 className="text-4xl font-bold text-gray-900 mb-4">Shop by Category</h2>
<p className="text-xl text-gray-600">Discover our delicious range of premium cakes</p>
</div>

<div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-4 gap-8">
        {categories.map((category, index) => (
<Link
            key={index}
            to={category.href}
            className="group relative overflow-hidden rounded-2xl shadow-lg hover:shadow-2xl transition-all duration-300 transform hover:scale-105"
>
<div className="aspect-w-1 aspect-h-1 h-64">
<img
              src={category.image}
              alt={category.name}
              className="w-full h-full object-cover group-hover:scale-110 transition-transform duration-300"
            />
```

```
<div className="absolute inset-0 bg-gradient-to-t from-black/60 to-transparent"></div>
<div className="absolute bottom-4 left-4">
<h3 className="text-xl font-bold text-white">{category.name}</h3>
</div>
</div>
</Link>
          ))}
</div>
</div>
</section>

    {/* Featured Products Section */}
<section className="py-20 bg-white">
<div className="max-w-7xl mx-auto px-4 sm:px-6 lg:px-8">
<div className="text-center mb-16">
<h2 className="text-4xl font-bold text-gray-900 mb-4">Featured Cakes</h2>
<p className="text-xl text-gray-600">Our most popular and highly-rated creations</p>
</div>

        {loading ? (
<div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-8">
            {[...Array(6)].map((_, index) => (
<div key={index} className="bg-gray-200 rounded-2xl h-96 animate-pulse"></div>
            ))}
</div>
        ) : (
<div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-8">
            {featuredProducts.map((product) => (
```

```
<div key={product._id} className="group bg-white rounded-2xl shadow-lg hover:shadow-2xl transition-all duration-300 overflow-hidden">

<div className="relative overflow-hidden">

<img

src={product.image}

alt={product.name}

className="w-full h-64 object-cover group-hover:scale-110 transition-transform duration-300"

/>

{product.tags.includes('bestseller') && (

<div className="absolute top-4 left-4 bg-rose-500 text-white px-3 py-1 rounded-full text-sm font-semibold">

Bestseller

</div>

)}

{product.originalPrice && (

<div className="absolute top-4 right-4 bg-green-500 text-white px-3 py-1 rounded-full text-sm font-semibold">

Save ₹{product.originalPrice - product.price}

</div>

)}

</div>


<div className="p-6">

<h3 className="text-xl font-bold text-gray-900 mb-2">{product.name}</h3>

<p className="text-gray-600 mb-4 line-clamp-2">{product.description}</p>


<div className="flex items-center mb-4">

<div className="flex items-center">

{[...Array(5)].map((_, i) => (

<Star
```

```
                    key={i}
                    className={`w-4 h-4 ${
                      i < Math.floor(product.rating)
                        ? 'text-yellow-400 fill-current'
                        : 'text-gray-300'
                    }`}
                  />
                ))}
</div>
<span className="ml-2 text-sm text-gray-600">({product.rating})</span>
</div>

<div className="flex items-center justify-between">
<div className="flex items-center space-x-2">
<span className="text-2xl font-bold text-gray-900">₹{product.price}</span>
                {product.originalPrice && (
<span className="text-lg text-gray-500 line-through">₹{product.originalPrice}</span>
                )}
</div>
<Link
                to={`/product/${product._id}`}
                className="px-6 py-2 bg-gradient-to-r from-amber-500 to-rose-500 text-white rounded-full font-semibold hover:from-amber-600 hover:to-rose-600 transition-all duration-200"
>
                View Details
</Link>
</div>
</div>
</div>
```

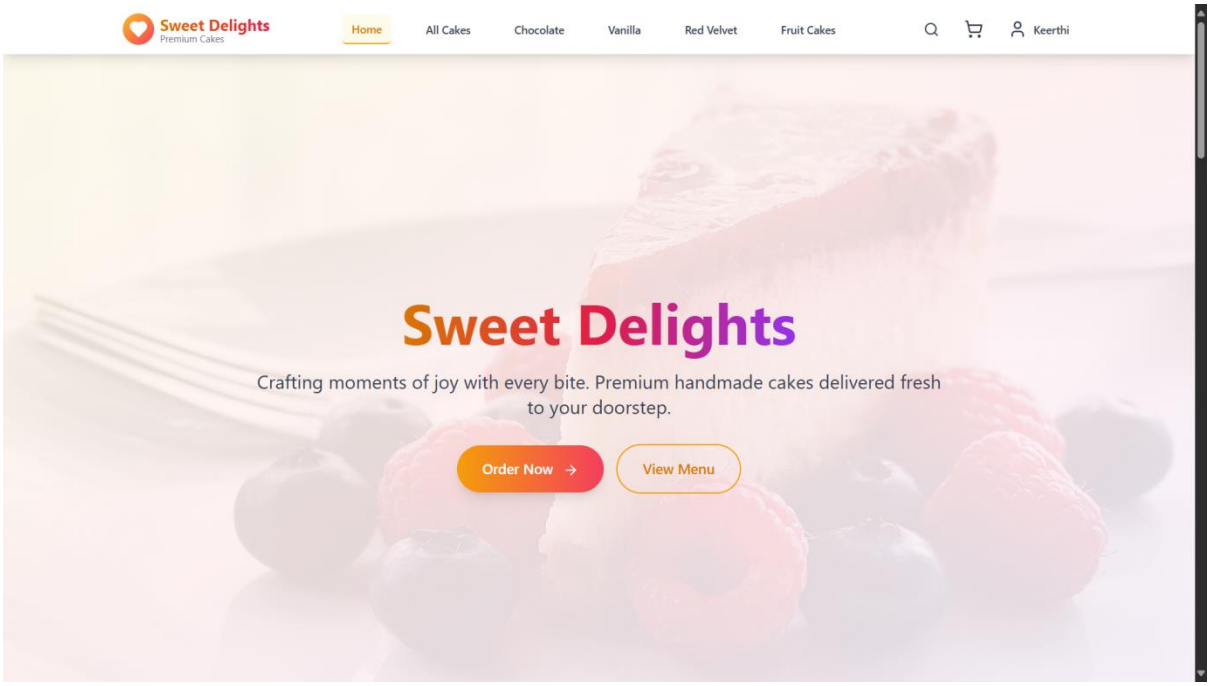
```
          ))}
</div>
        )}

<div className="text-center mt-12">
<Link
          to="/products"
          className="px-8 py-3 bg-gradient-to-r from-amber-500 to-rose-500 text-white rounded-full font-semibold text-lg hover:from-amber-600 hover:to-rose-600 transition-all duration-200"
>
          View All Cakes <ArrowRight className="inline w-5 h-5 ml-2" />
</Link>
</div>
</div>
</section>

    {/* CTA Section */}
<section className="py-20 bg-gradient-to-r from-amber-500 to-rose-500 text-white">
<div className="max-w-7xl mx-auto px-4 sm:px-6 lg:px-8 text-center">
<h2 className="text-4xl font-bold mb-4">Ready to Order Your Perfect Cake?</h2>
<p className="text-xl mb-8 opacity-90">
          Join thousands of happy customers who trust us for their special moments
</p>
<div className="flex flex-col sm:flex-row gap-4 justify-center">
<Link
          to="/products"
          className="px-8 py-4 bg-white text-amber-600 rounded-full font-semibold text-lg hover:bg-gray-100 transition-all duration-200"
>
```

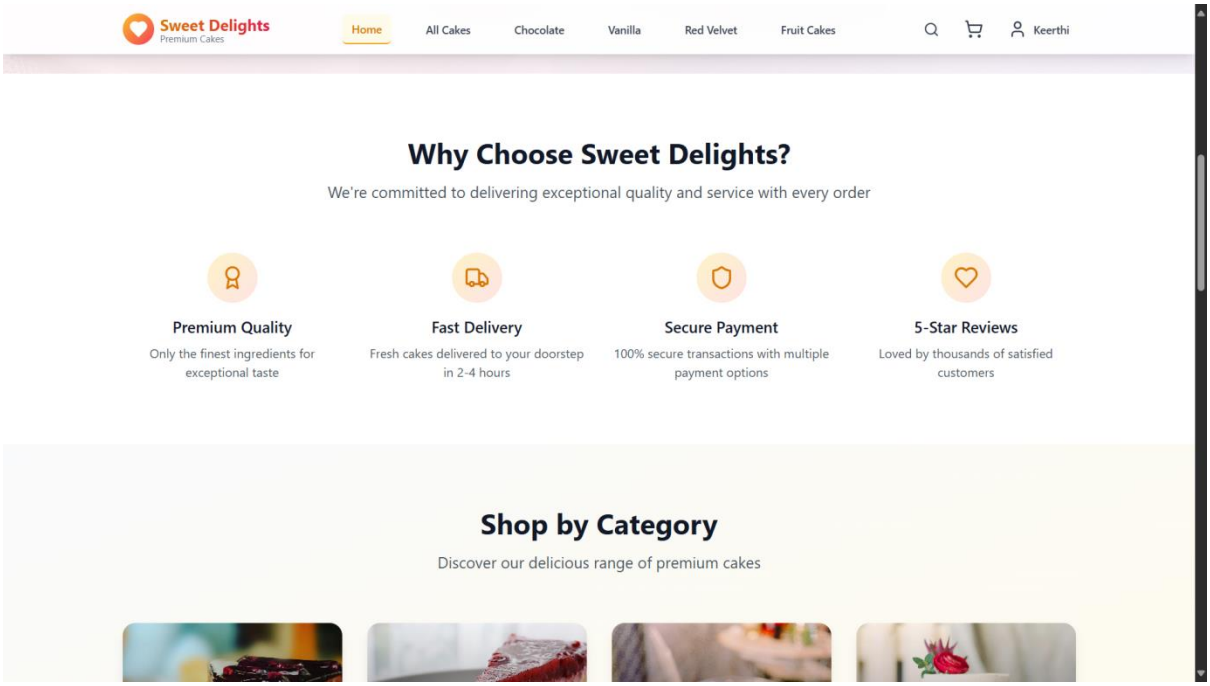
```
            Browse Cakes
</Link>
<Link
            to="/register"
            className="px-8 py-4 border-2 border-white text-white rounded-full font-semibold text-lg hover:bg-white hover:text-amber-600 transition-all duration-200"
>
            Sign Up Now
</Link>
</div>
</div>
</section>
</div>
  );
};
export default Home;
```

CHAPTER 6

SCREENSHOTS

1. Home page:





Sweet Delights
Premium Cakes

Home | All Cakes | Chocolate | Vanilla | Red Velvet | Fruit Cakes | Keerthi

Featured Cakes

Our most popular and highly-rated creations

Black Forest Special

Traditional German cake with chocolate, cherries, and whipped cream

★★★★☆ (4.9)

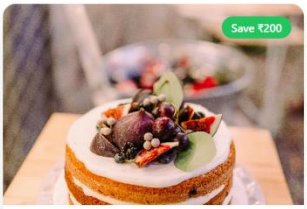
₹999 ~~₹1199~~ View Details

Chocolate Fudge Delight

Rich, decadent chocolate cake with layers of fudge and chocolate ganache

★★★★☆ (4.8)

₹899 ~~₹1099~~ View Details



Red Velvet Romance

Velvety red cake with cream cheese frosting and a hint of cocoa

★★★★☆ (4.7)

₹799 ~~₹999~~ View Details

Sweet Delights
Premium Cakes

Home | All Cakes | Chocolate | Vanilla | Red Velvet | Fruit Cakes | Keerthi

Ready to Order Your Perfect Cake?

Join thousands of happy customers who trust us for their special moments

Browse Cakes | Sign Up Now

Sweet Delights

Creating memorable moments with our premium handcrafted cakes since 2020. Every bite tells a story of love and craftsmanship.

Quick Links

- Home
- All Cakes
- Chocolate Cakes
- Vanilla Cakes
- Track Order

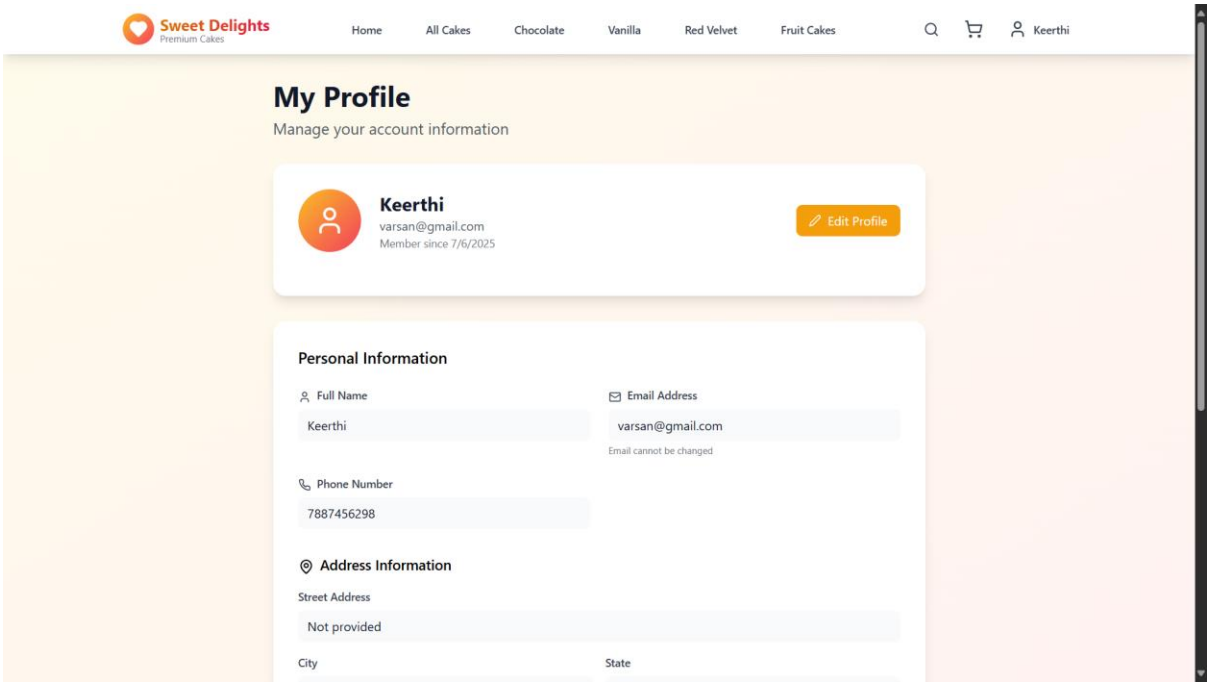
Customer Service

- Contact Us
- FAQ
- Shipping Info
- Returns
- Privacy Policy

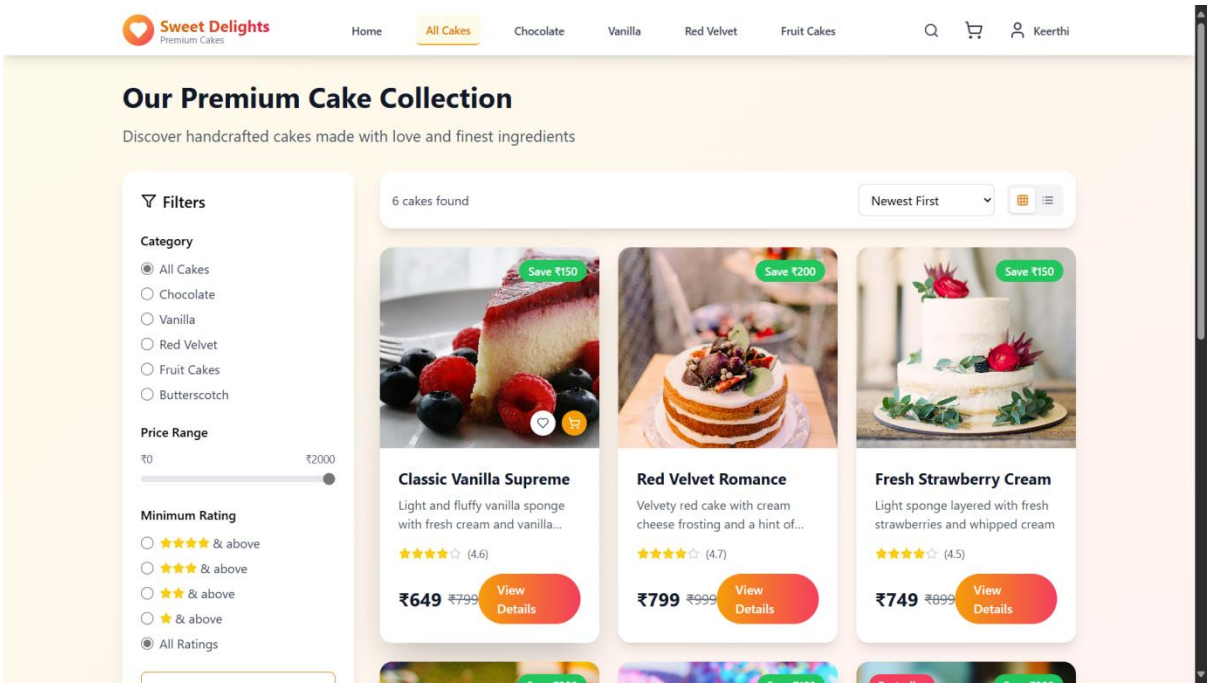
Contact Info

+91 98765 43210
orders@sweetdelights.com
123 Baker Street
Mumbai, Maharashtra 400001
India

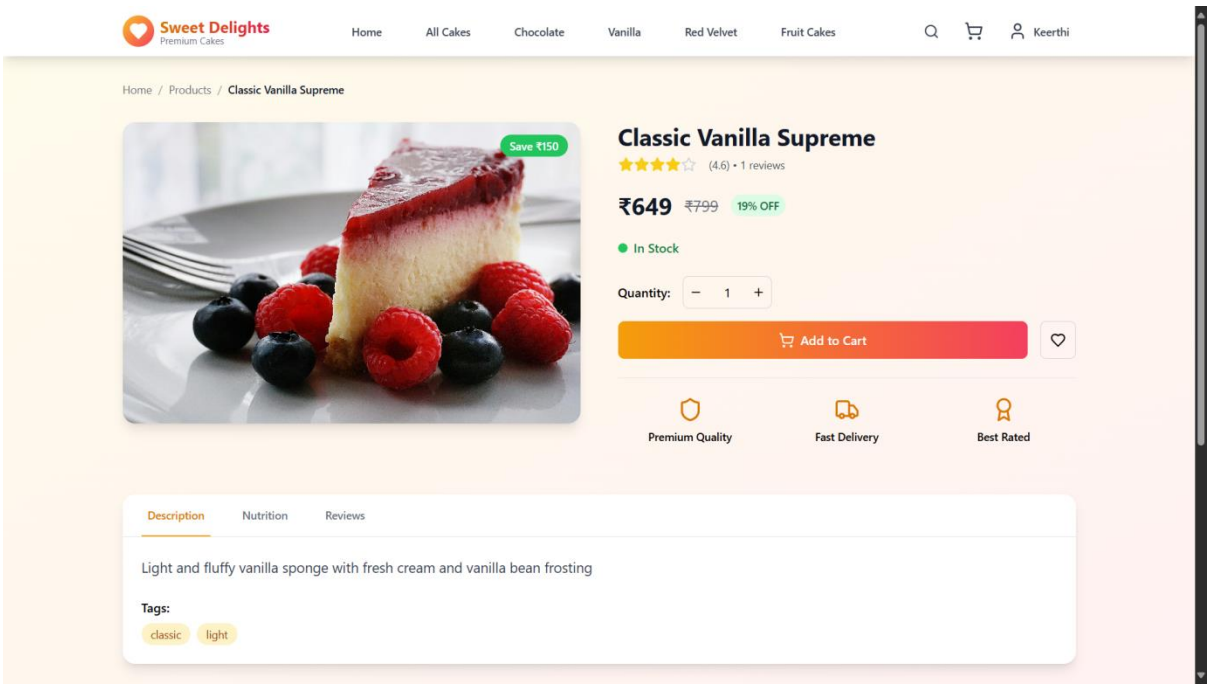
2. Profile page



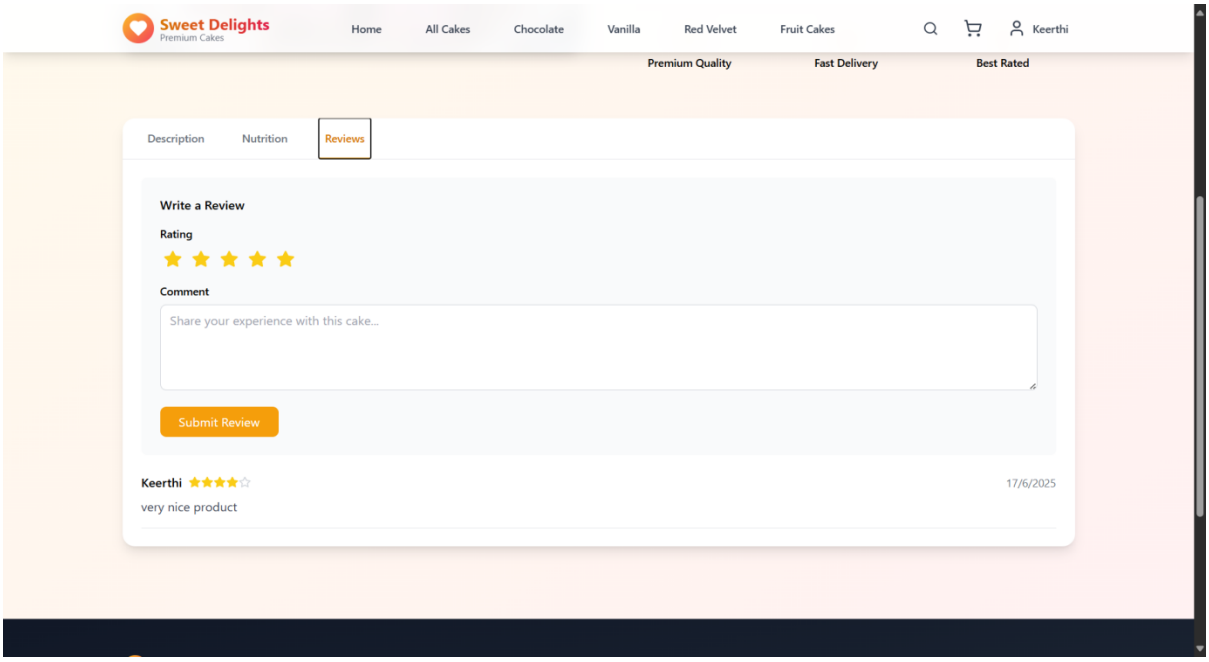
3. Catalogue Page:



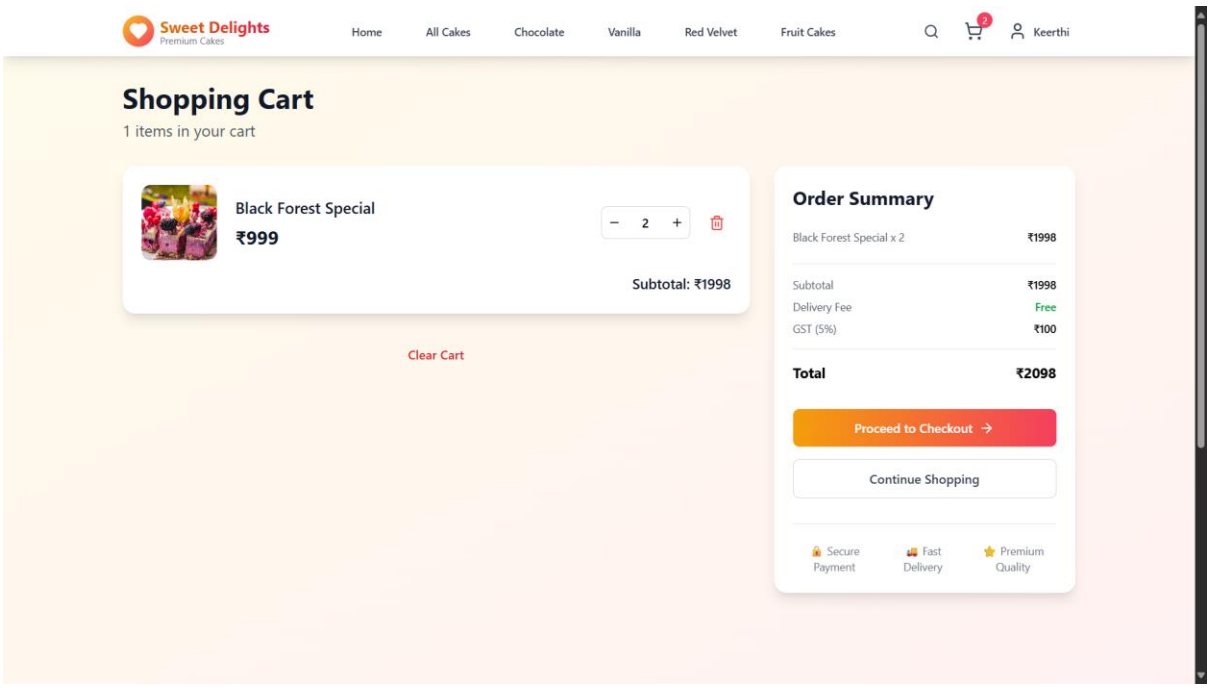
4. Product Page:



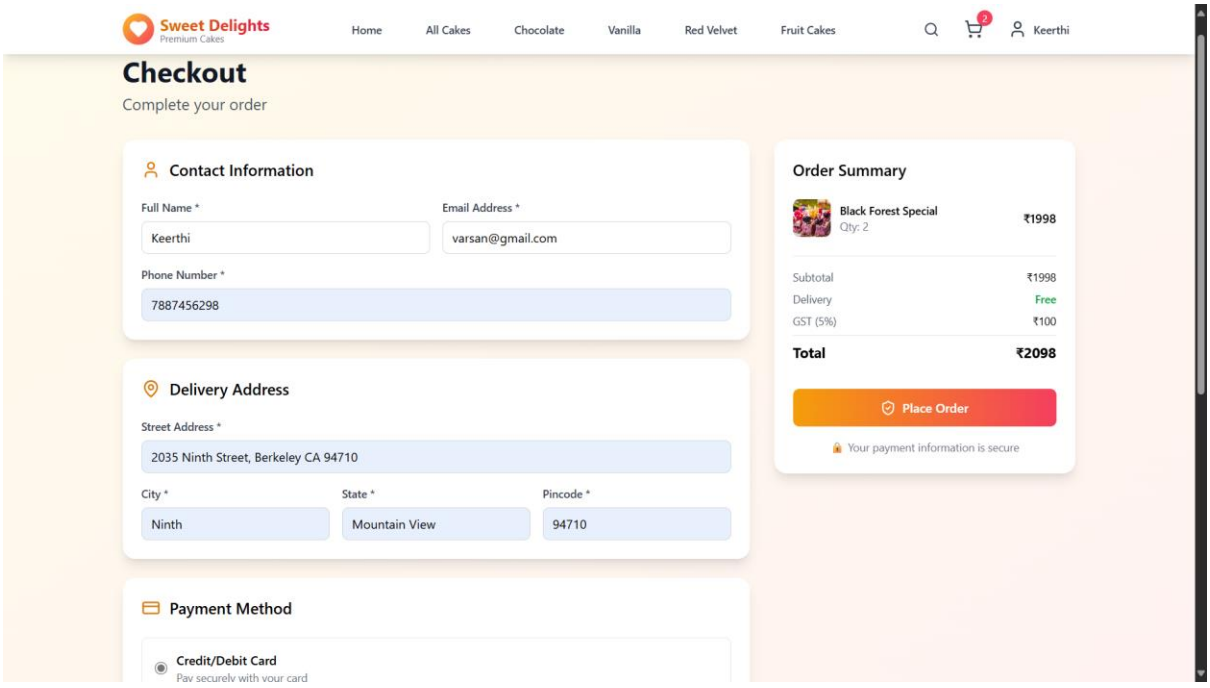
5. ReviewPage:



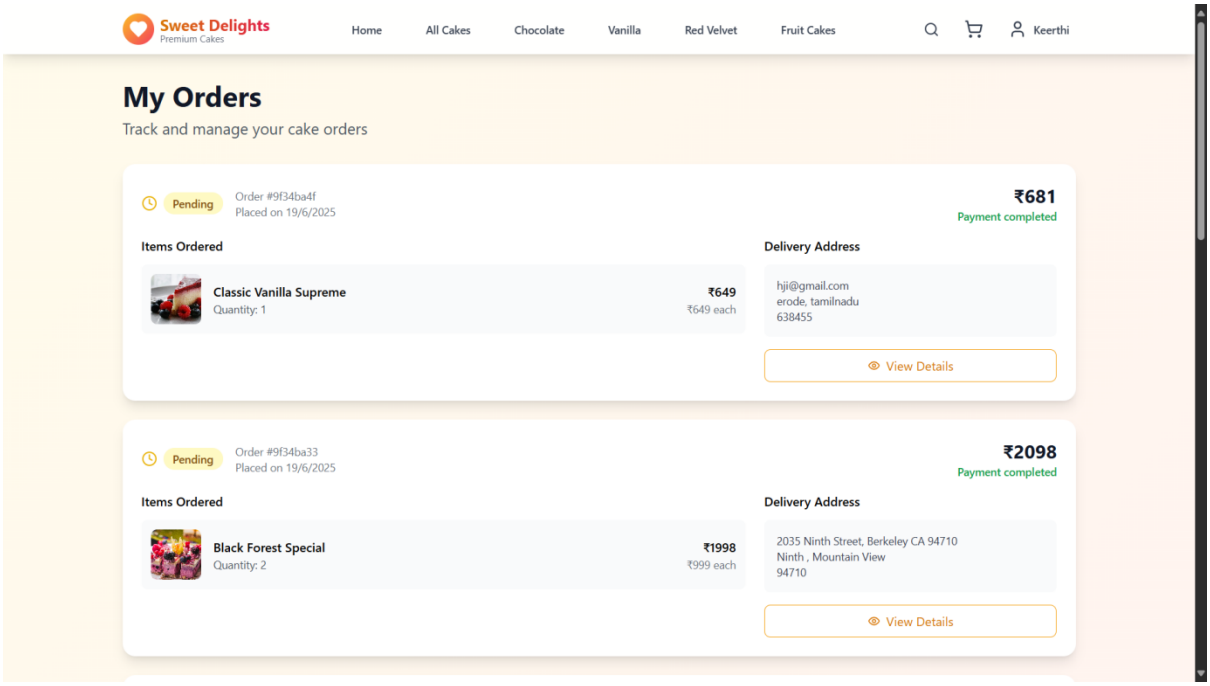
6. Cart page:



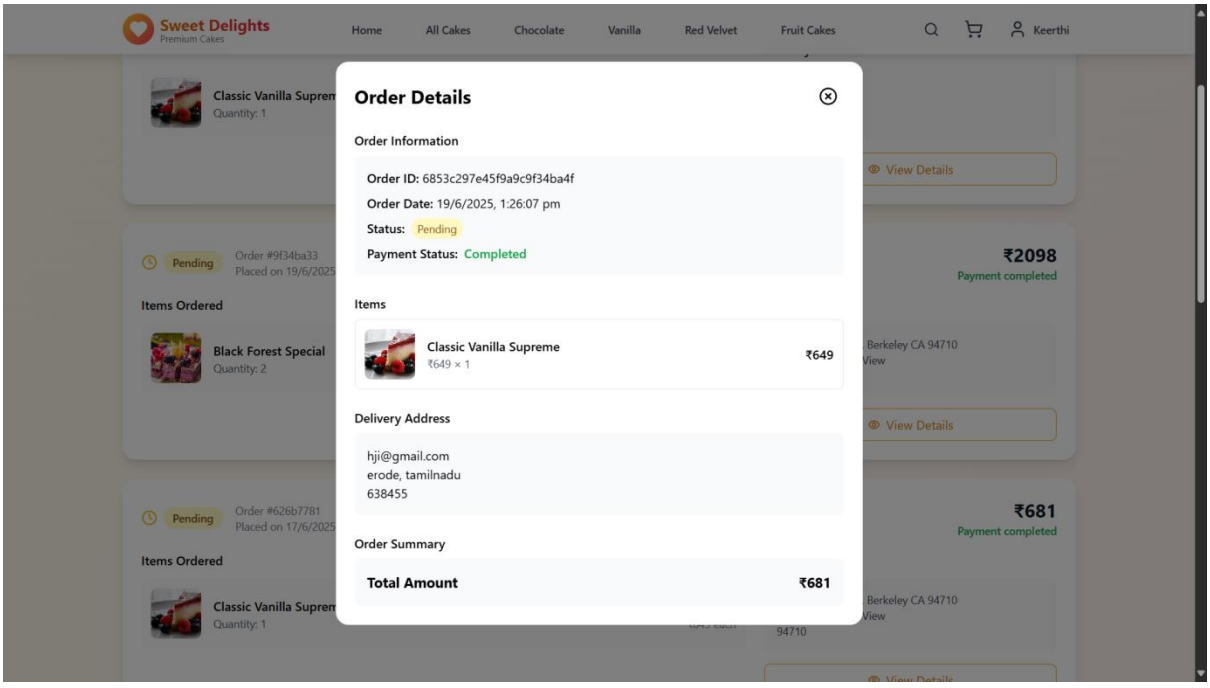
7. Checkout Page:



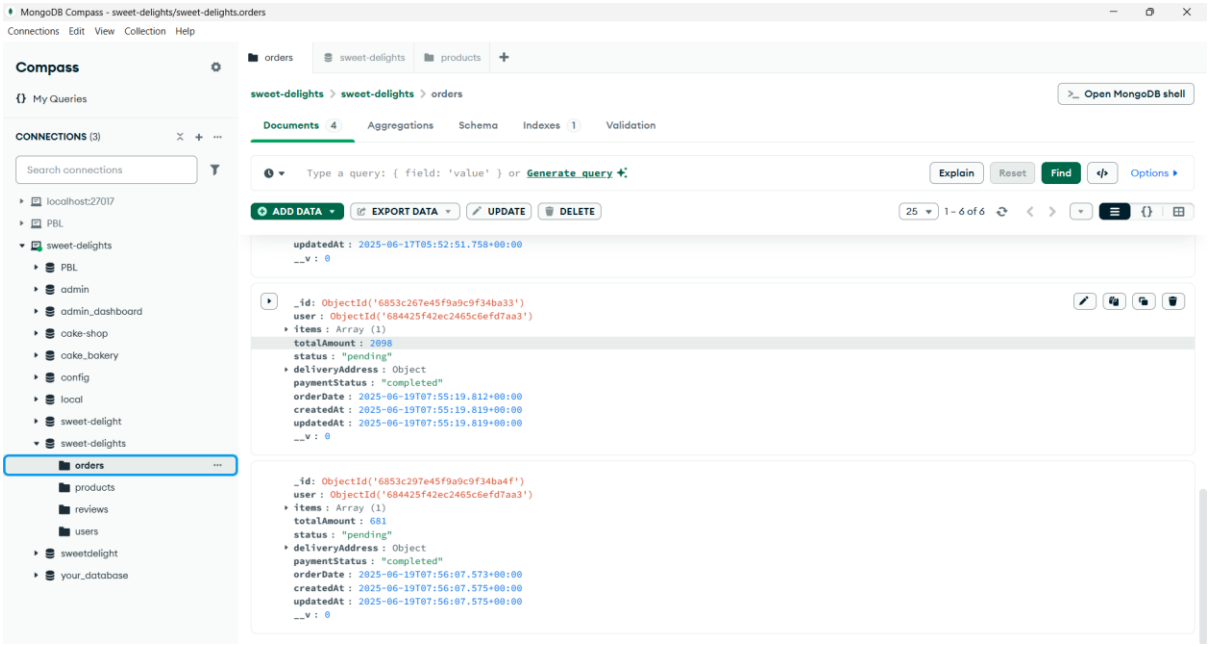
8. My Orders Page:



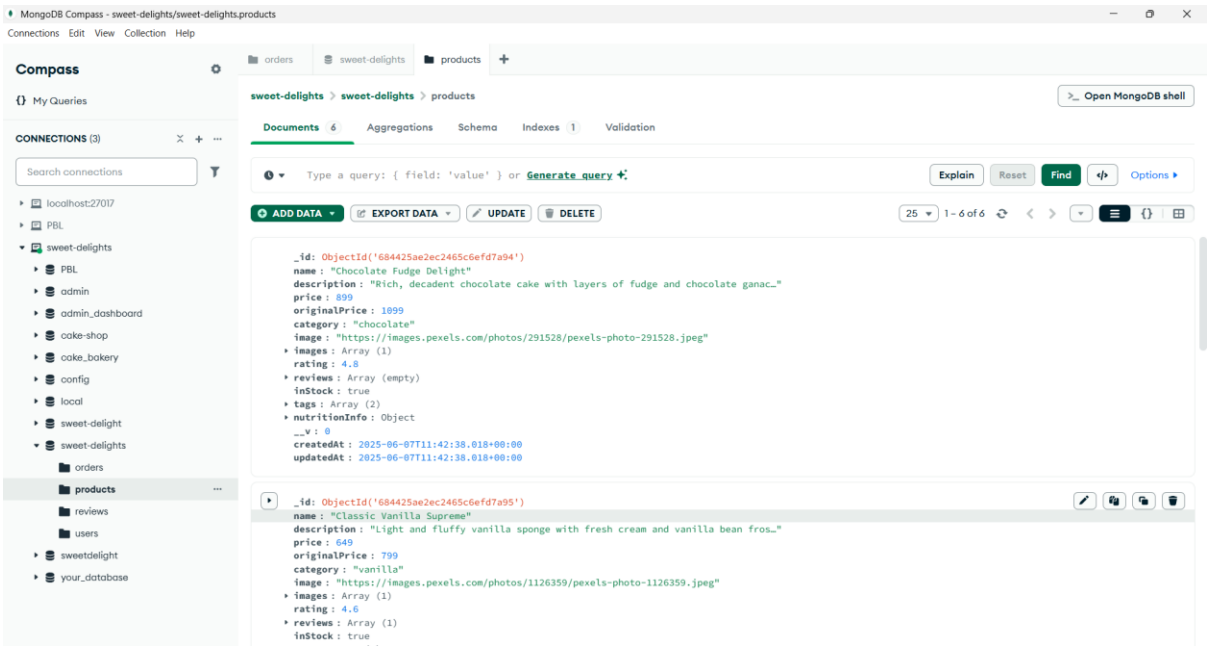
9. Order details:



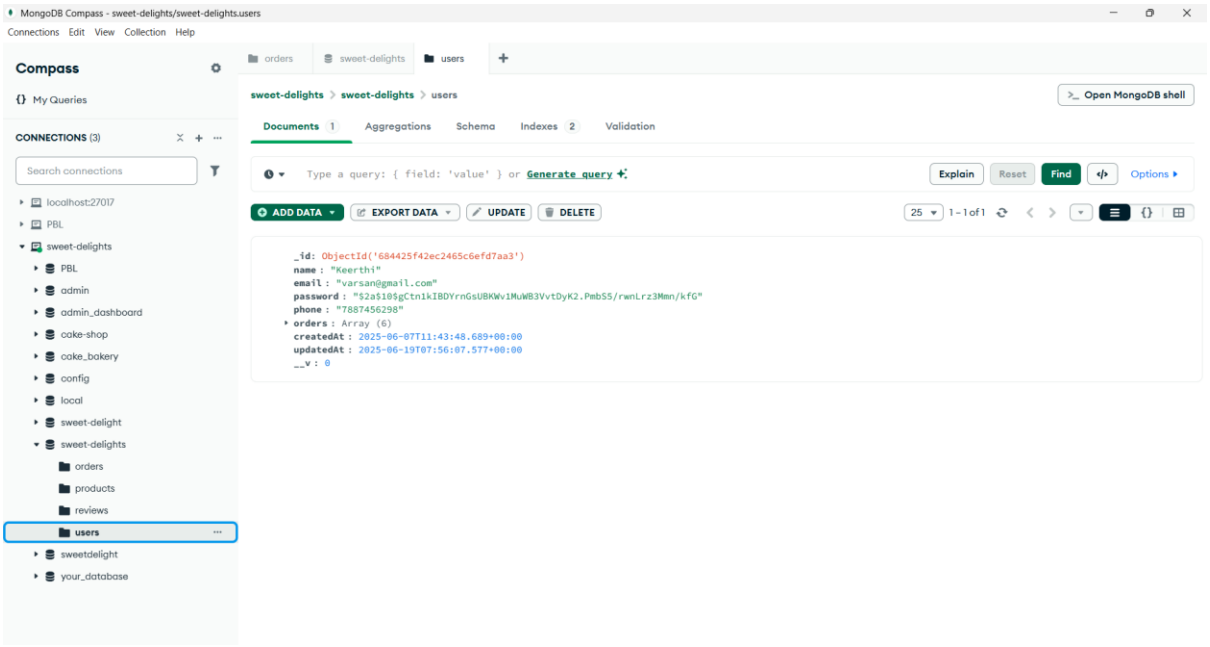
10. Orders Collection:



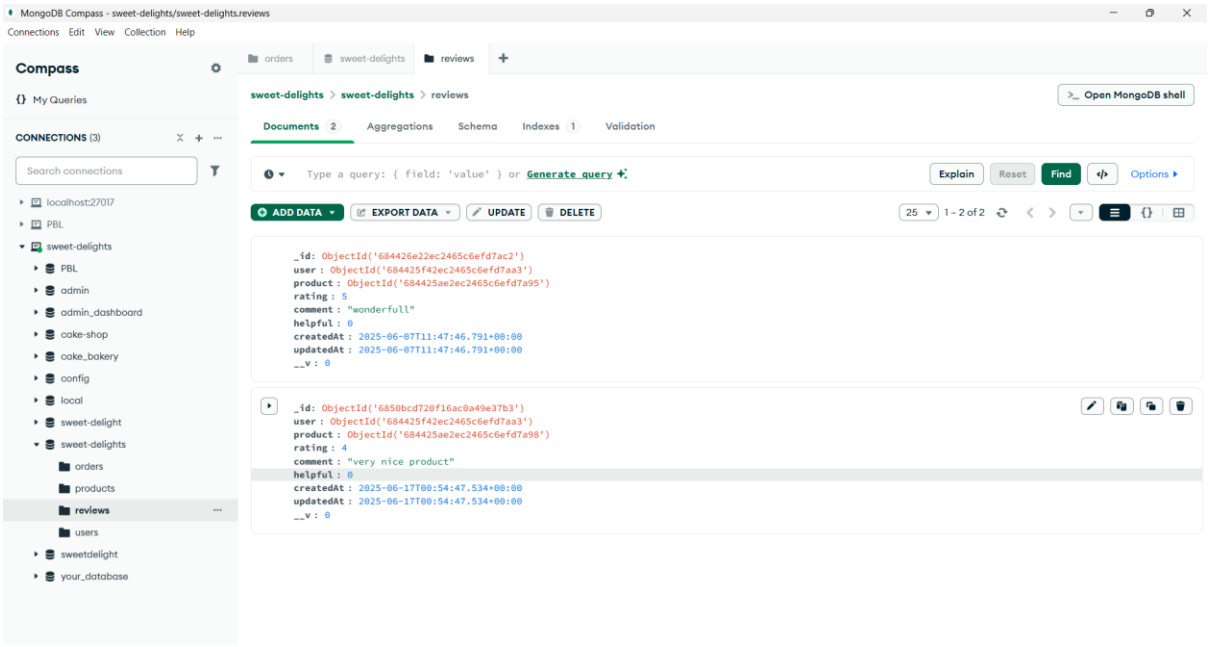
11. Product Collection:



12.UsersCollection:



13.Review Collection:



CHAPTER 7

CONCLUSION

The Cake bakery Website is a fully functional MERN stack-based platform.This project successfully implements a full-stack web application integrating user authentication, content management, and activity tracking. The React frontend provides a dynamic and responsive user interface, while the backend ensures secure and efficient data handling. Features like login/signup, profile management, and activity logging enhance the overall user experience. Content creation, editing, and deletion are streamlined through a robust and user-friendly interface. The modular component-based design ensures easy maintenance and scalability. By leveraging modern web development frameworks and best practices, the project meets industry standards. Furthermore, the integration of backend logic with database operations ensures data persistence and reliability. Security considerations like protected routes and session handling are implemented. Overall, the project demonstrates practical knowledge of full-stack development and readiness for real-world deployment.

CHAPTER 8

SCOPE FOR FUTURE ENHANCEMENT

1. **Role-Based Access Control:**

Introduce advanced user roles (admin, editor, viewer) to control access to different features and content modules.

2. **Notification System:**

Add real-time notifications for user activity, content updates, and system alerts using WebSockets or Firebase.

3. **Analytics Dashboard:**

Implement data visualization tools to track user engagement, activity, trends, and system performance.

4. **Offline Mode:**

Incorporate service workers to allow limited functionality and content access without internet connectivity.

CHAPTER 9
REFERENCES

[1] *Eloquent JavaScript: A Modern Introduction to Programming* – Marijn Haverbeke, No Starch Press, 3rd Edition, 2018.

[2] *Learning React: Functional Web Development with React and Redux* – Alex Banks and Eve Porcello, O'Reilly Media, 2nd Edition, 2020.

[3] *Web Development with Node and Express* – Ethan Brown, O'Reilly Media, 2nd Edition, 2019.

[4]*MongoDB: The Definitive Guide* – Shannon Bradshaw, Kristina Chodorow, O'Reilly Media, 3rd Edition, 2019.

[5] *Professional JavaScript for Web Developers* – Nicholas C. Zakas, Wrox Publications, 3rd Edition, 2012.

[6] React.js Documentation – https://reactjs.org/docs

[7] Roy, S., & Biswas, S. (2019). *A study on modern web application frameworks and security issues*. International Journal of Computer Applications, 177(31), 15-21.

[8] Sharma, V., & Bhardwaj, D. (2020). *User Authentication Techniques in Web Applications*. International Journal of Engineering Research & Technology (IJERT), Vol. 9 Issue 05.