

**NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY**

(AN AUTONOMOUS INSTITUTION, AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM, APPROVED BY AICTE & GOVT.OF KARNATAKA

**MINI PROJECT REPORT**

on

**ONLINE FOOD DELIVERY APPLICATION USING MERN STACK**

*Submitted in partial fulfilment of the requirement for the award of Degree of*

*Bachelor of Engineering*

*in*

*Information Science and Engineering*

*Submitted by:*

KANTA AISHWARYA 1NT20IS072

Under the Guidance of

Mr. Mohan Kumar T. G. Assistant Professors, Dept. of ISE, NMIT

Ms Akarsha DP, Mr.Veeranna Kotagi Assistant Professors, Dept. of ISE,NMIT

Department of Information Science and Engineering

**(Accredited by NBA Tier-1)**



2023-2024

(AN AUTONOMOUS INSTITUTION, AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM

Department of Information Science and Engineering

**(Accredited by NBA Tier-1)**

**CERTIFICATE**

This is to certify that the Project Report on “**ONLINE FOOD DELIVERY APPLICATION USING MERN STACK”** is an authentic work carried out by **KANTA AISHWARYA (1NT20IS072)** Bonafide students of Nitte Meenakshi Institute of Technology, Bangalore in partial fulfilment for the award of the degree of Bachelor of Engineering in Information Science and Engineering of Visvesvaraya Technological University, Belagavi during the academic year 2023-2024***.*** It is certified that all corrections and suggestions indicated during the internal assessment has been incorporated in the report.

**Web Tech Faculty Big Data Faculty**

Mr. Mohan T. G. Ms Akarsha DP

Assistant Professor, Dept. ISE, Assistant Professor, Dept. ISE, NMIT Bangalore NMIT Bangalore



## ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any taskwould be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crowned our effort with success. I express my sincere gratitude to our **Principal Dr. H. C. Nagaraj**, Nitte Meenakshi Institute of Technology for providing facilities.

We wish to thank our **HOD, Dr. Mohan S. G.** for the excellent environment created to further educational growth in our college. We also thank him for the invaluable guidance provided which has helped in the creation of a better project.

I hereby like to thank our **Mr. Mohan Kumar T. G., Assistant Professor, Ms Akarsha D P Assistant Professor,** Department of Information Science & Engineering on his periodic inspection, time to time evaluation of the project and help to bring the project to the present form.

Thanks to our Departmental Project coordinators. We also thank all our friends, teaching and non-teaching staff at NMIT, Bangalore, for all the direct and indirect help provided in the completion of the project.

|  |  |  |
| --- | --- | --- |
| **NAME** | **USN** | **Signature** |
| KANTA AISHWARYA | 1NT20IS072 | k.aishwarya |

Date: 30th May 2023



# ABSTRACT

The online food delivery application is a web-based platform that allows users to order food from their favorite restaurants and have it delivered to their doorstep. The application is built using the MERN stack, which includes MongoDB, Express, React, and Node.js.

Key features of the Online Food Delivery Application include user registration and authentication, restaurant and menu browsing, cart management, order placement, payment integration, and order tracking. Users can create personalized profiles, search for nearby restaurants, filter menus based on cuisine or dietary preferences, and add items to their cart. The application supports secure payment options and provides real-time updates on order status, estimated delivery time, and delivery tracking.

The frontend of the application is developed using React.js, a popular JavaScript library for building user interfaces. React.js provides a responsive and interactive user interface, allowing users to browse restaurants, view menus, add items to their cart, and place orders effortlessly. The application employs various UI components, such as search filters, menus, and shopping carts, to enhance the user experience and facilitate smooth navigation.

Node.js, a server-side JavaScript runtime environment, powers the backend of the application. It handles the business logic, authentication, and database operations necessary for processing user requests and managing orders. Node.js enables real-time updates and notifications, ensuring users receive timely information regarding order status and delivery updates.

The MERN stack, known for its robustness and scalability, serves as the foundation for building this application. MongoDB, a NoSQL database, is used to store and manage the application's data, including user profiles, restaurant information, menu items, and order details. Express.js, a flexible web application framework, handles server-side operations, routing, and request handling, ensuring efficient communication between the client and server.

Overall, the Online Food Delivery Application using the MERN stack aims to revolutionize the way people order food online. The application provides a seamless and efficient experience for users, enabling them to enjoy their favorite meals conveniently, securely, and with minimum hassle.



# Table of Contents

* Introduction
* Problem statement
* Technologies Used
* Implementation
* Results
* Conclusion

# CHAPTER 1: INTRODUCTION

In today's fast-paced world, convenience is key. With the rise of technology and changing lifestyles, online food delivery services have become an integral part of our daily routines. Whether you're a food lover seeking diverse culinary experiences or a busy professional. Our online food delivery application boasts an intuitive and visually appealing interface that ensures an effortless user experience. From the moment users land on our platform, they can easily navigate through various sections, browse menus, select items, customize orders, and proceed with secure online payments.

Our application simplifies the ordering process, allowing users to add items to their cart, modify quantities, apply discounts or promotional codes, and specify delivery preferences. A streamlined checkout system ensures quick and hassle-free payments, and users can track their orders in real-time.

Understanding that every user has unique preferences, our online food delivery application employs advanced algorithms to provide personalized recommendations. By analyzing previous orders, favorite cuisines, and dietary preferences, we offer tailored suggestions that cater to individual tastes.

Efficient delivery management is a priority for us. Our application incorporates features such as real-time order tracking, estimated delivery times, and delivery personnel management to ensure that food reaches customers' doorsteps in a timely and convenient manner.

To enhance the overall experience, our application seamlessly integrates with various third- party services. This includes secure payment gateways, mapping services for accurate delivery tracking, and customer support systems to address any queries or concern.

# Chapter 2: Problem Statement

The problem statement for an online food delivery application using the MERN stack could be to provide a convenient and efficient way for customers to order food from their favorite restaurants while also allowing restaurants to manage their orders and menus. One of the main challenges of building such an application is ensuring that it is scalable and can handle a large number of users and orders. Additionally, the application should be user-friendly and provide a seamless experience for customers, from browsing menus to placing orders and tracking deliveries. Another challenge is ensuring the security of the application, including user authentication and payment processing. Finally, the application should be easy to maintain and update, with a clear separation of concerns between the front-end and back-end components. By addressing these challenges, developers can create a successful online food delivery application that meets the needs of both customers and restaurants.

# Chapter 3: Technologies used

## REACT.JS:

With its component-based architecture, React.js enables the creation of interactive and user- friendly interfaces, ensuring a seamless and enjoyable customer experience.

React.js allows for creating responsive user interfaces that adapt to different screen sizes and devices. With the help of CSS frameworks like Material-UI or Bootstrap, you can build responsive layouts and design user-friendly interfaces that work well on both desktop and mobile devices. This is crucial for an online food delivery system as users may access it from various devices.

React.js is used in the front-end of an online food delivery application using the MERN stack to create a user-friendly interface for customers to browse menus, place orders, and track deliveries. React.js is a popular JavaScript library that allows developers to create reusable UI components and manage the state of the application efficiently. React.js works by breaking down the user interface into smaller components, each with its own state and behavior. These components can be reused throughout the application, making it easier to maintain and update. React.js also uses a virtual DOM, which allows it to update the UI efficiently without having to reload the entire page. This makes the application faster and more responsive, providing a better user experience. In addition, React.js can be used with other libraries and frameworks, such as Redux and React Router, to provide additional functionality and improve the performance of the application. By using React.js in the front-end of an online food delivery application, developers can create a scalable, efficient, and user-friendly interface that meets the needs of both customers and restaurants.

## NODE.JS:

Node.js, a JavaScript runtime environment, enables efficient handling of server-side operations, such as order processing, real-time updates, and authentication.

Node.js is used in the back-end of an online food delivery application using the MERN stack to handle HTTP requests and responses, as well as to execute JavaScript code on the server-side. Node.js is a popular runtime environment that allows developers to write server-side code in JavaScript, the same language used in the front-end. Node.js works by using an event-driven, non-blocking I/O model, which allows it to handle a large number of concurrent connections efficiently. This makes it ideal for building scalable and efficient web applications, such as online food delivery applications. In addition, Node.js can be used with other frameworks and libraries, such as Express and Mongoose, to provide additional functionality and improve the performance of the application. Node.js also has a large and active community of developers, which provides support and resources for building and maintaining web applications. By using Node.js in the back-end of an online food delivery application, developers can create a scalable, efficient, and secure application that meets the needs of both customers and restaurants.

## EXPRESS.JS:

This web application framework simplifies the development process by providing a set of tools and features for building robust APIs and handling server-side logic.

Express.js is used in the back-end of an online food delivery application using the MERN stack to handle HTTP requests and responses, as well as to provide middleware for authentication

and authorization. Express.js is a popular framework that handles HTTP requests and responses, as well as provides middleware for authentication and authorization. It is a Node.js HTTP framework that handles a lot of things out of the box and requires little code to create fully functional RESTful APIs. Express.js can be used with other libraries, such as Mongoose, to provide additional functionality and improve the performance of the application. Mongoose is an Object Data Modeling (ODM) library for MongoDB that provides a higher level of abstraction for working with MongoDB. By using Express.js in the back-end of an online food delivery application, developers can create a scalable, efficient, and secure application that meets the needs of both customers and restaurants. Express.js provides a simple and efficient way to handle HTTP requests and responses, as well as to provide middleware for

authentication and authorization, making it an ideal choice for building RESTful APIs for online food delivery applications.

## MONGODB:

As a NoSQL database, MongoDB offers flexibility and scalability, making it ideal for handling large amounts of data related to menus, orders, customer profiles, and more.

MongoDB is used in the back-end of an online food delivery application using the MERN stack to store information about users, restaurants, menus, and orders. MongoDB is a NoSQL document-oriented database that is designed to handle large volumes of unstructured data. MongoDB uses JSON as a way to turn data into something much more like code, which allows the structure of the data to be under the control of the developer. This makes it easier to work with in web applications and provides a higher level of abstraction for working with data. MongoDB is also highly scalable, which means it can handle the demands of even the largest applications. In addition, MongoDB provides several authentication mechanisms, such as SCRAM, x.509, and LDAP, to ensure the security of the data. By using MongoDB in the back- end of an online food delivery application, developers can create a scalable, efficient, and secure application that meets the needs of both customers and restaurants.

# Chapter 4: Implementation

Implementing an online food delivery application using the MERN stack involves several steps.

## Project Setup:

To set up a project for building an online food delivery application using the MERN stack (MongoDB, Express.js, React.js, and Node.js), you need to follow these steps:

1. Create a Project Directory:

* Create a new directory for your project on your local machine.
* Open a terminal or command prompt and navigate to the project directory.

1. Initialize the Project:

* Run the following command in the terminal to initialize a new Node.js project npm init-y
* This will create a `package.json` file in the project directory, which will store information about your project and its dependencies.

1. Backend Setup:

* Create a new folder within your project directory for the backend code.
* Navigate to the backend folder using the terminal cd backend

1. Install Backend Dependencies:
   * To set up the backend dependencies, you need to install Express.js and other required packages. Run the following command in the terminal

npm install express mongoose cors dotenv

* + This will install Express.js (web application framework), Mongoose (ODM library for MongoDB), cors (middleware for enabling cross-origin requests), and dotenv (for environment variable configuration).

1. Backend File Structure:

* Create the necessary files for the backend. Typical files include:
  + `index.js`: The entry point for your backend server.
  + `models/`: A folder to store the Mongoose models for your data.
  + `routes/`: A folder to define the routes and API endpoints.
  + `controllers/`: A folder to handle the logic for each route.
  + `config/`: A folder to store configuration files or constants.
  + `middlewares/`: A folder to define custom middleware functions.
  + `utils/`: A folder to store utility functions or helper files.
  + `middlewares/auth.js`: A file for authentication middleware (if required).

1. Frontend Setup:

* Create a new folder within your project directory for the frontend code.
* Navigate to the frontend folder using the terminal:

cd frontend

1. Install Frontend Dependencies:
   * To set up the frontend dependencies, you need to install React.js and other required packages. Run the following command in the terminal:

npx create-react-app .

- This will create a new React.js project within the current directory and install the required dependencies.

1. Frontend File Structure:
   * The `create-react-app` command automatically generates a file structure for your frontend code. Key files and folders include:
     + `src/`: A folder to store the source code for your React components.
     + `src/components/`: A folder to organize your React components.
     + `src/pages/`: A folder to define the main pages of your application.
     + `src/services/`: A folder to handle API calls and communication with the backend.
     + `src/utils/`: A folder to store utility functions or helper files.
     + `src/App.js`: The main component that renders other components.
     + `src/index.js`: The entry point of your React application.
2. Connect Backend and Frontend:
   * In your frontend code, you will make HTTP requests to the backend API. Open the `.env` file in the frontend folder and add the following line:

REACT\_APP\_API\_URL=http://localhost:3001/api

* + Replace `http://localhost:3001/api` with the appropriate URL of your backend server.

1. Development Environment:
   * You are now ready to start developing your online food delivery application using the MERN stack. You can open

## Backend Development:

* Define and create the necessary models using Mongoose to represent entities like users, restaurants, menu items, orders, etc.
* Set up the Express.js server to handle HTTP requests and responses.
* Implement API endpoints for user authentication (e.g., registration, login, logout), restaurant listing, menu item management, order placement, etc.
* Configure routes to map the incoming requests to the appropriate API handlers.
* Connect to MongoDB database using the MongoDB driver or Mongoose and handle data persistence.

## Frontend Development:

* Create React components for different sections of the application, such as the landing page, restaurant listings, menu, cart, order tracking, user profile, etc.
* Design the user interface using HTML, CSS, and UI component libraries like Material-UI or Bootstrap.
* Implement routing using React Router to enable navigation between different sections of the application.
* Integrate with backend APIs using libraries like Axios to fetch data, handle user

authentication, and submit orders.

* Manage component state using React's built-in useState and useEffect hooks or consider using a state management library like Redux for complex state management needs.

## User Authentication and Authorization:

* Implement user authentication mechanisms such as JWT-based authentication or session- based authentication.
* Set up routes and controllers for user registration, login, logout, and password reset.
* Implement middleware functions to handle authentication and authorization for protected routes.

## Chapter 5: RESULTS

**HOME PAGE:**

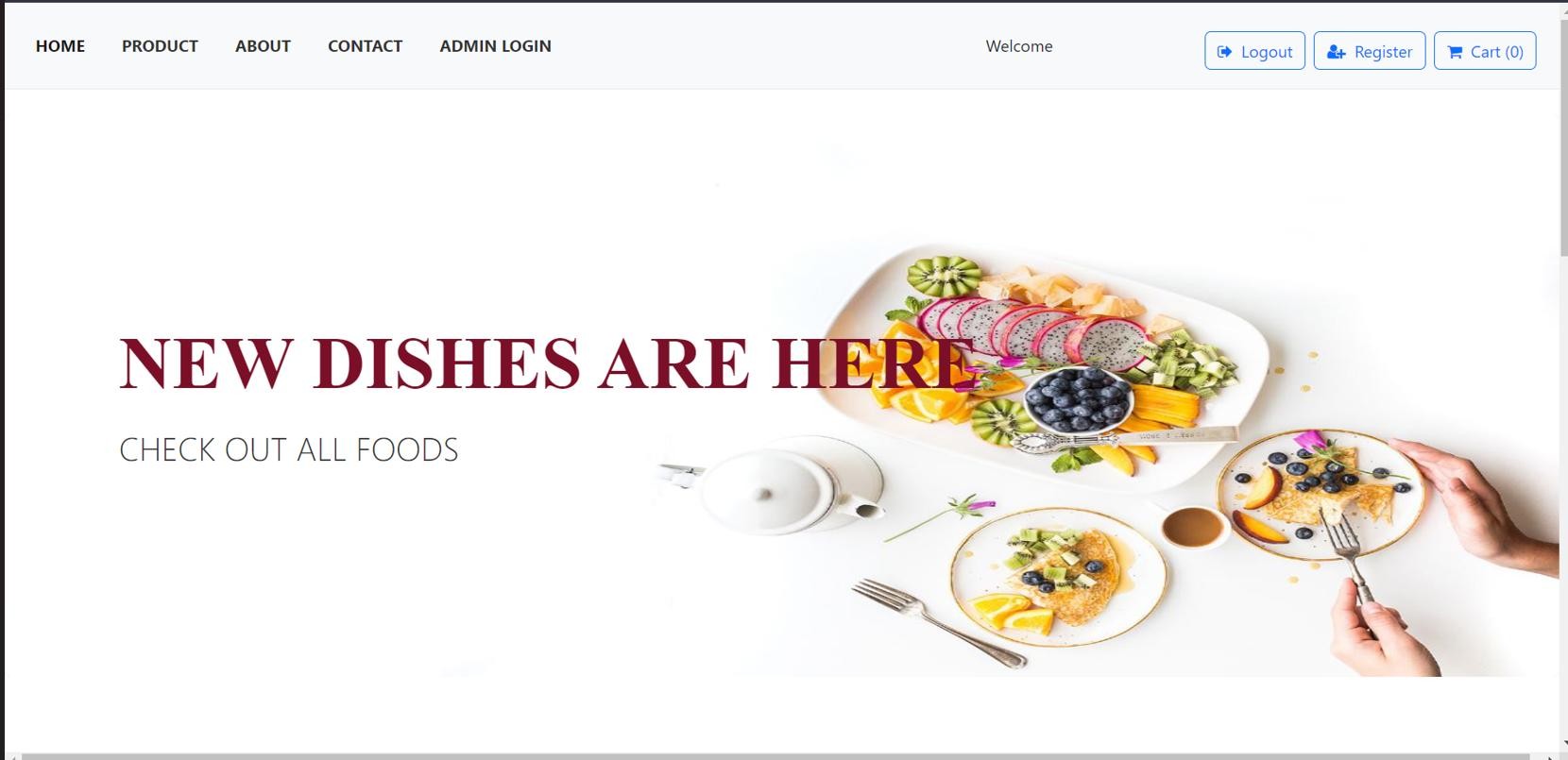


Figure:5.1

The home page of an online food delivery application serves as the landing page where users first arrive and provides an overview of the platform's features, enticing users to explore further.

## ADMIN LOGIN:

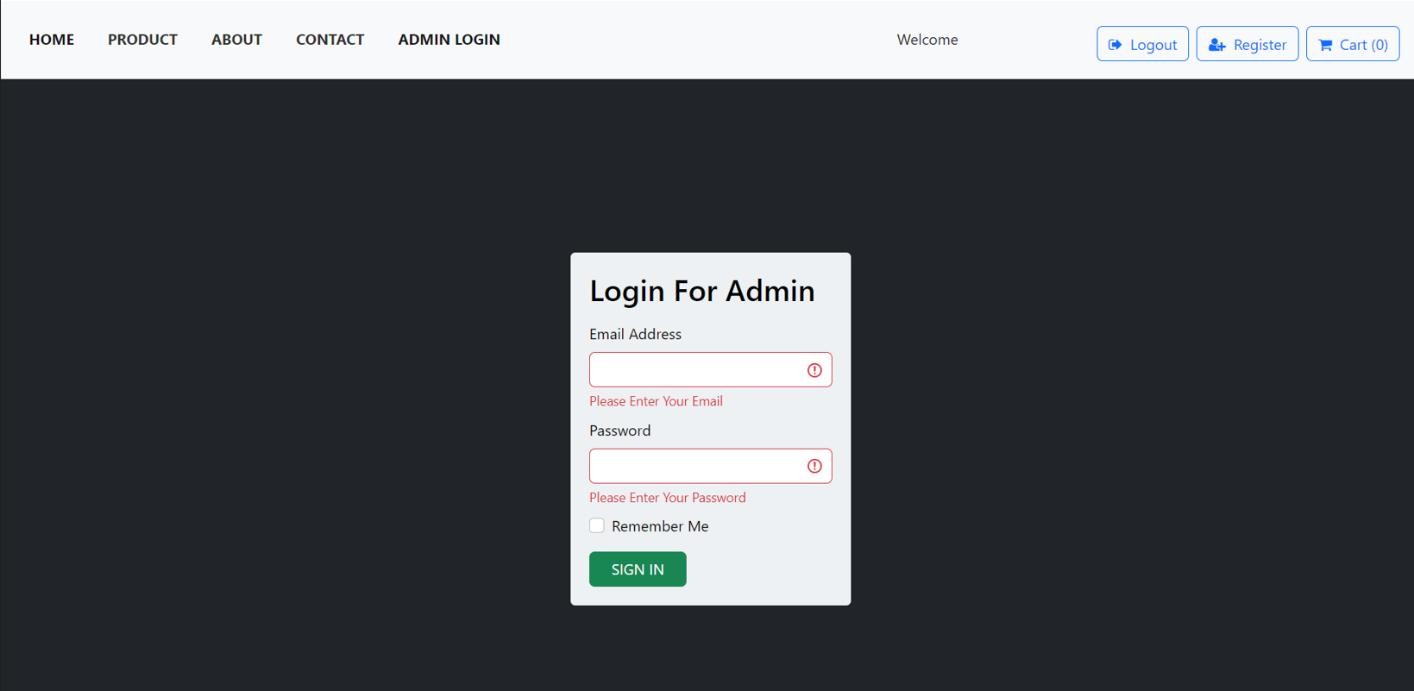


Figure:5.2

The admin login page for an online food delivery application provides a secure entry point for administrators to access the backend administration panel.

## REGISTRATION PAGE FOR CUSTOMERS

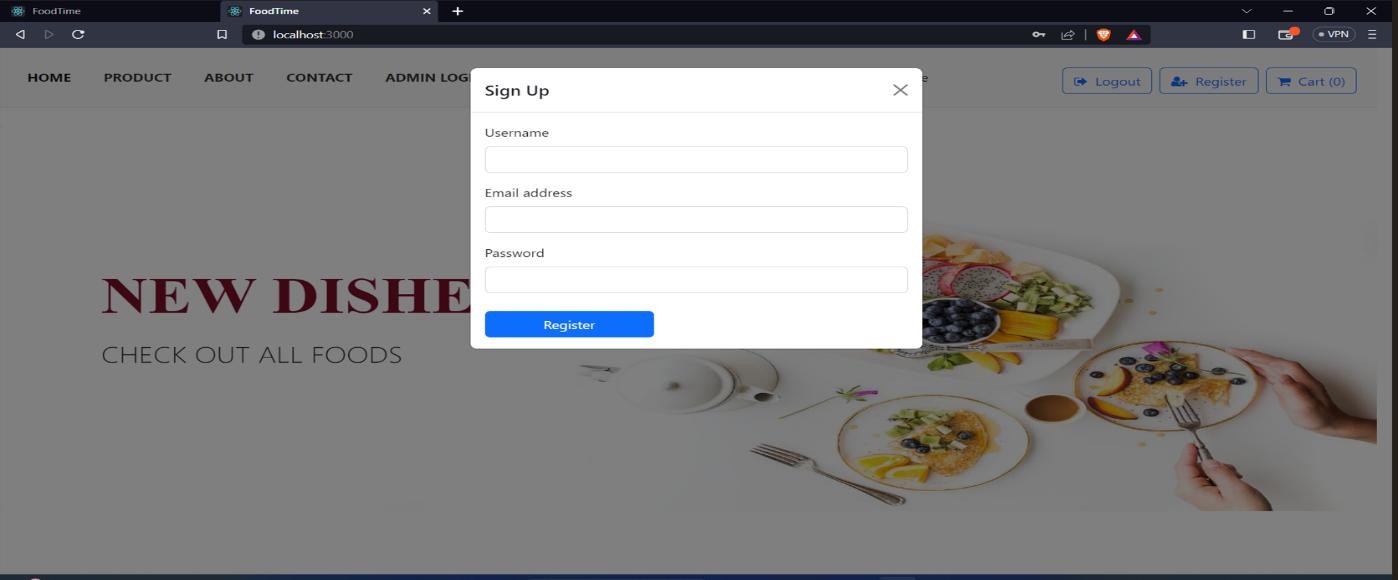


Figure:5.3

The registration page for customers in an online food delivery application allows new users to create an account and gain access to the platform's features.

## FOOD PRODUCT PAGE

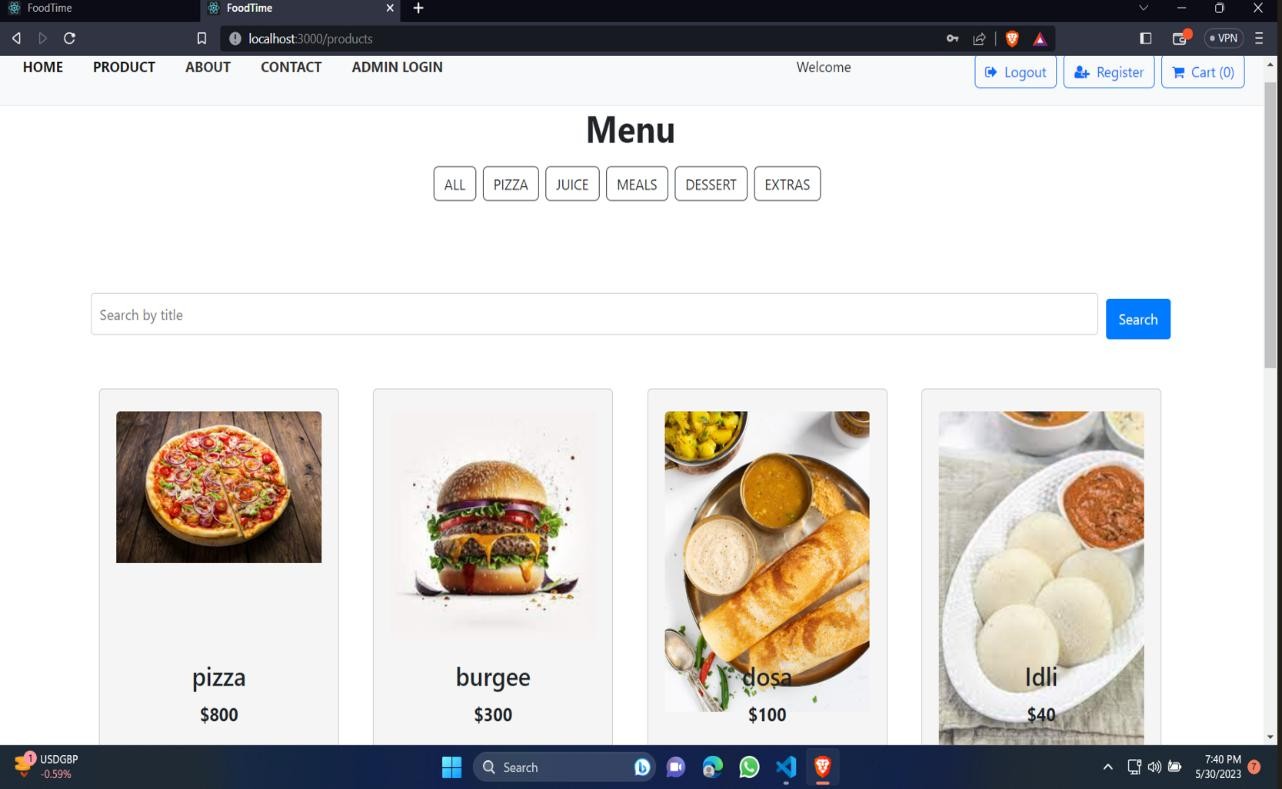


Figure:5.4

The food products page in an online food delivery application serves as a catalog or listing of available food items from different restaurants or food establishments. It allows users to browse and explore the various options before making their selections. Here's a description of the key elements typically found on a food products page:

1. Search and Filters:
   * The food products page often includes a search bar where users can enter keywords to search for specific food items or cuisines.
   * Filters are provided to help users narrow down their search based on criteria such as cuisine type, dietary preferences, price range, or restaurant ratings.
   * These search and filter options enable users to find the desired food products efficiently.
2. Food Product Cards:

* Each food product is represented by a card or tile containing key information and visuals.
* The card typically includes the food product's name, brief description, price, and an image to showcase its appearance.
* Additional details such as dietary information, tags, or ratings may also be included on the card.

1. Pagination or Infinite Scrolling:
   * If there are a large number of food products available, pagination or infinite scrolling may be implemented to manage the display of products on the page.
   * Pagination divides the food products into multiple pages, allowing users to navigate through them using page numbers or previous/next buttons.
   * Infinite scrolling continuously loads additional food products as the user scrolls down the page, providing a seamless browsing experience.
2. Sorting Options:

* Sorting options allow users to arrange the food products based on specific criteria.
* Common sorting options include sorting by popularity, price (low to high or high to low), ratings, or alphabetical order.
* Providing sorting options helps users find the food products that best match their preferences.

1. Add to Cart or Order:

- Each food product card typically includes an "Add to Cart" or "Order" button.

## ABOUT PAGE:

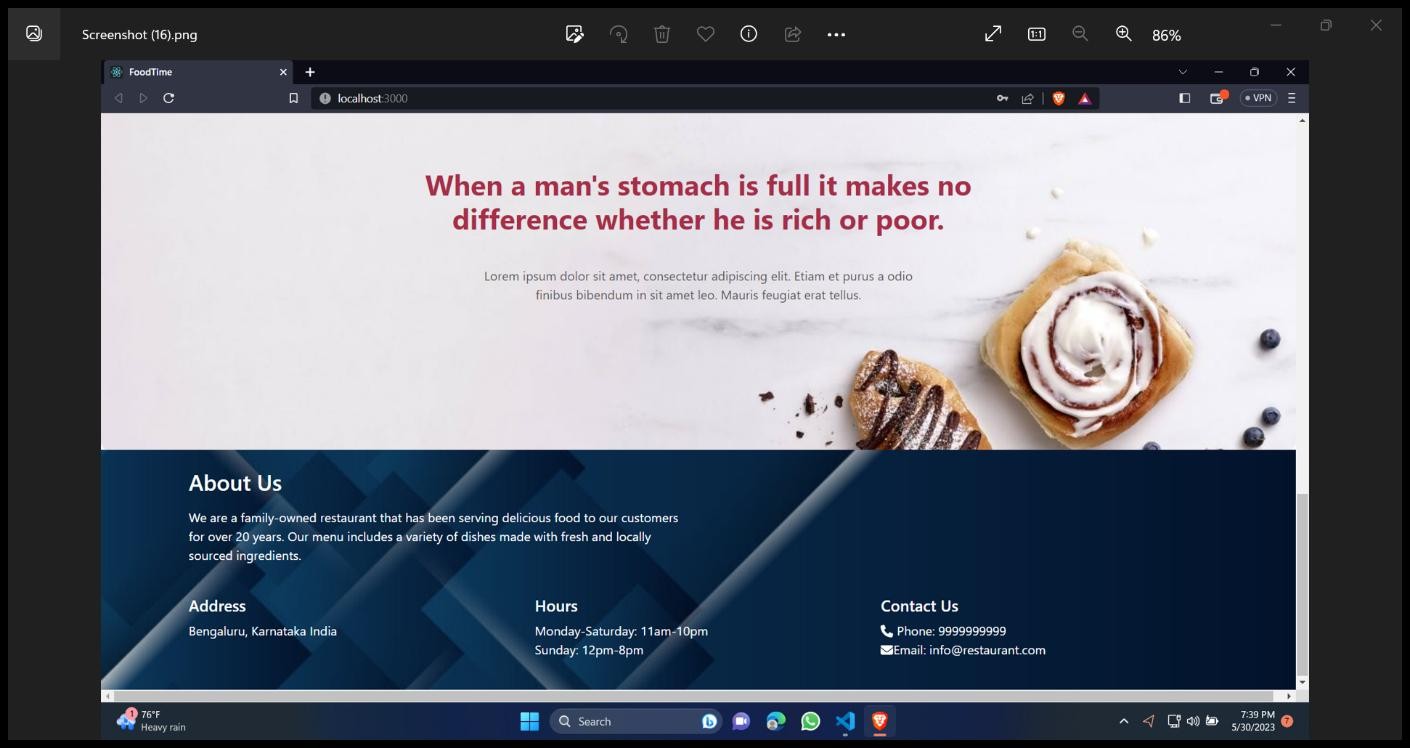


Figure:5.5

The About page in an online food delivery application provides users with information about the application itself, including its purpose, mission, and background. It serves as a platform to communicate the application's values, goals, and unique selling points. Here's a description of the key elements typically found on an About page:

1. Introduction:

* The About page starts with a concise and engaging introduction that gives users an overview of the application and its purpose.
* It may highlight the main features or benefits that set the application apart from competitors.

1. Company or Application Information:

* The About page provides details about the company or organization behind the application.
* This can include information about the company's history, founding members, or key milestones.
* If applicable, it may also include information about the application's team, their expertise, and their passion for delivering excellent food and service.

1. Mission and Vision:

* The About page often articulates the mission and vision of the application.
* The mission statement communicates the application's primary goal or purpose, such as providing convenient and high-quality food delivery services to customers.
* The vision statement outlines the long-term aspirations and direction of the application, such as becoming a leading platform in the food delivery industry.

1. Values and Commitments:

* The About page may list the core values and commitments of the application.
* This could include principles such as customer satisfaction, quality control, sustainability, supporting local businesses, or fostering positive user experiences.

1. Testimonials or Success Stories:
   * Including testimonials or success stories from satisfied customers or partner restaurants can build trust and credibility.
   * Testimonials can highlight positive experiences, exceptional service, or the convenience and reliability of the application.
2. Contact Information:
   * The About page may include contact information, such as an email address or customer support phone number.
   * This allows users to reach out with any questions, feedback, or partnership opportunities.
3. Social Media Links:

* The About page often includes links or icons to the application's social media profiles.
* These links encourage users to connect with the application's social media community, stay updated on news and promotions, and engage in conversations.

1. Styling and Visuals:
   * The About page should have a visually appealing design that aligns with the overall branding of the application.
   * It may include relevant images or graphics to enhance the page's aesthetic appeal and make it more engaging for users.

The About page provides users with essential information about the online food delivery application, allowing them to better understand its purpose, values, and mission. It helps build trust, showcases the application's unique features, and fosters a connection with users.

## CONTACT PAGE:

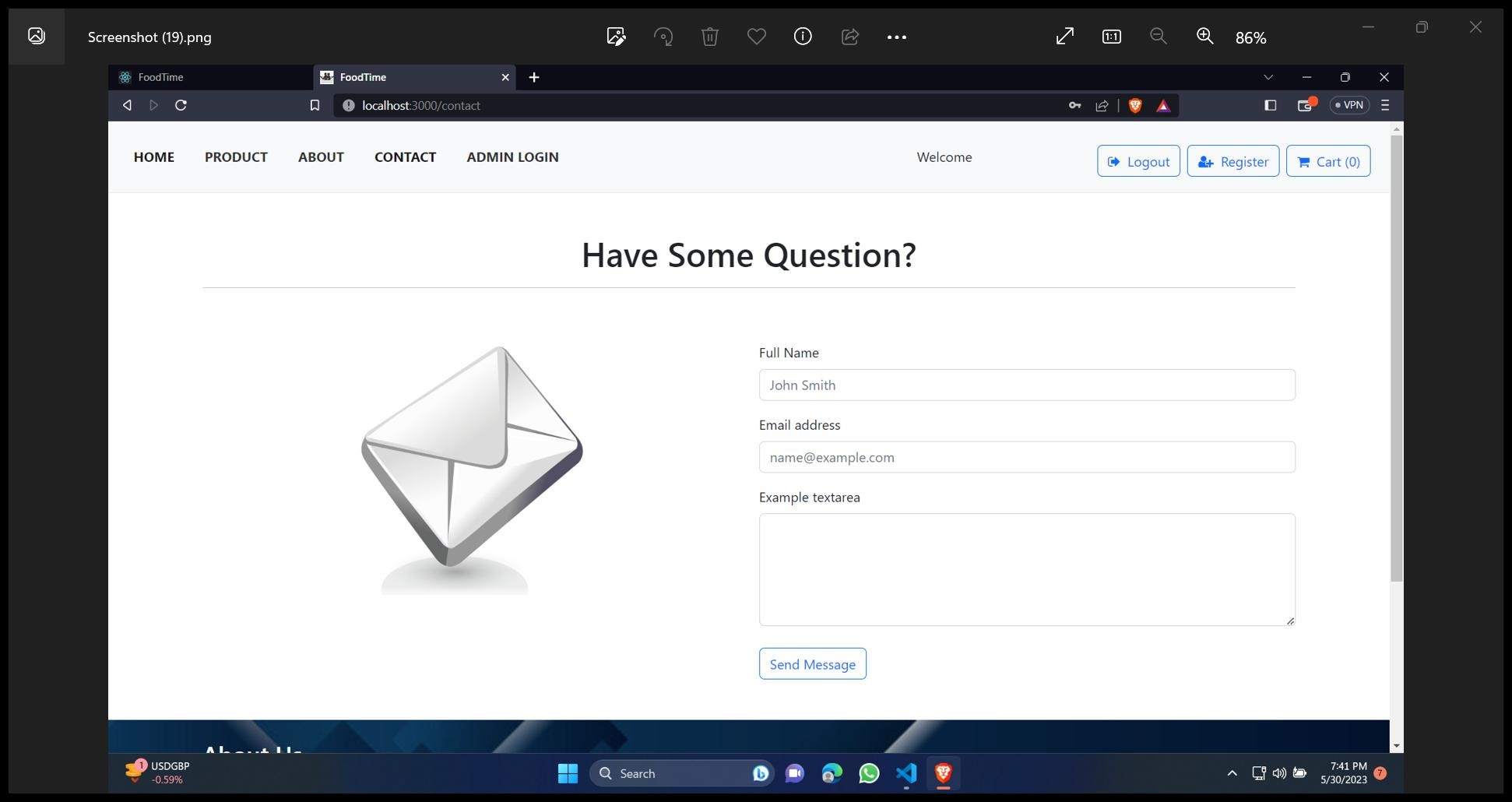


Figure:5.6

The contact page in an online food delivery application provides users with a means to get in touch with the application's customer support or restaurant management team. It serves as a channel for users to ask questions, report issues, provide feedback, or seek assistance. Here's a description of the key elements typically found on a contact page:

1. Contact Information:
   * The contact page displays the contact information for the application's customer support or restaurant management team.
   * This typically includes a phone number, email address, and physical address (if applicable).
   * Providing multiple contact options gives users flexibility in choosing their preferred method of communication.
2. Contact Form:

* A contact form allows users to send messages directly from the contact page.
* The form typically includes fields for users to enter their name, email address, subject, and message.
* Additional fields may be included based on the specific requirements of the application.

1. Message Submission:
   * Once users fill in the contact form, they can submit their message by clicking a "Submit" or "Send" button.
   * Upon submission, the message is typically sent to the designated email address or stored in a database for further processing.
2. Information or FAQ Section:
   * The contact page may include an information section or frequently asked questions (FAQ) to address common queries or provide essential details.
   * This section can cover topics such as delivery times, payment options, cancellation policies, or general inquiries.
   * Including relevant information can help reduce the number of support requests and provide immediate answers to user questions.
3. Social Media Links:

* Many online food delivery applications maintain a social media presence.
* The contact page may include links or icons to the application's social media profiles (e.g., Facebook, Twitter, Instagram) to encourage users to connect and engage through those channels.

1. Error and Confirmation Messages:
   * The contact page should provide appropriate error messages if users encounter any issues while filling out the contact form.
   * Upon successful submission of the form, a confirmation message should be displayed to acknowledge that the message has been sent successfully.
2. Styling and Responsiveness:
   * The contact page should be visually appealing, consistent with the application's design, and easy to navigate.
   * It should be responsive, ensuring it adapts well to different screen sizes and devices for a seamless user experience.

The contact page serves as an important communication channel for users to interact with the application's support team or restaurant management. It should provide clear contact information, a user-friendly contact form, and relevant information to assist users effectively. Prompt and helpful responses to user inquiries can enhance user satisfaction and build trust in the application's customer service.

## BACKEND:

### User Database:

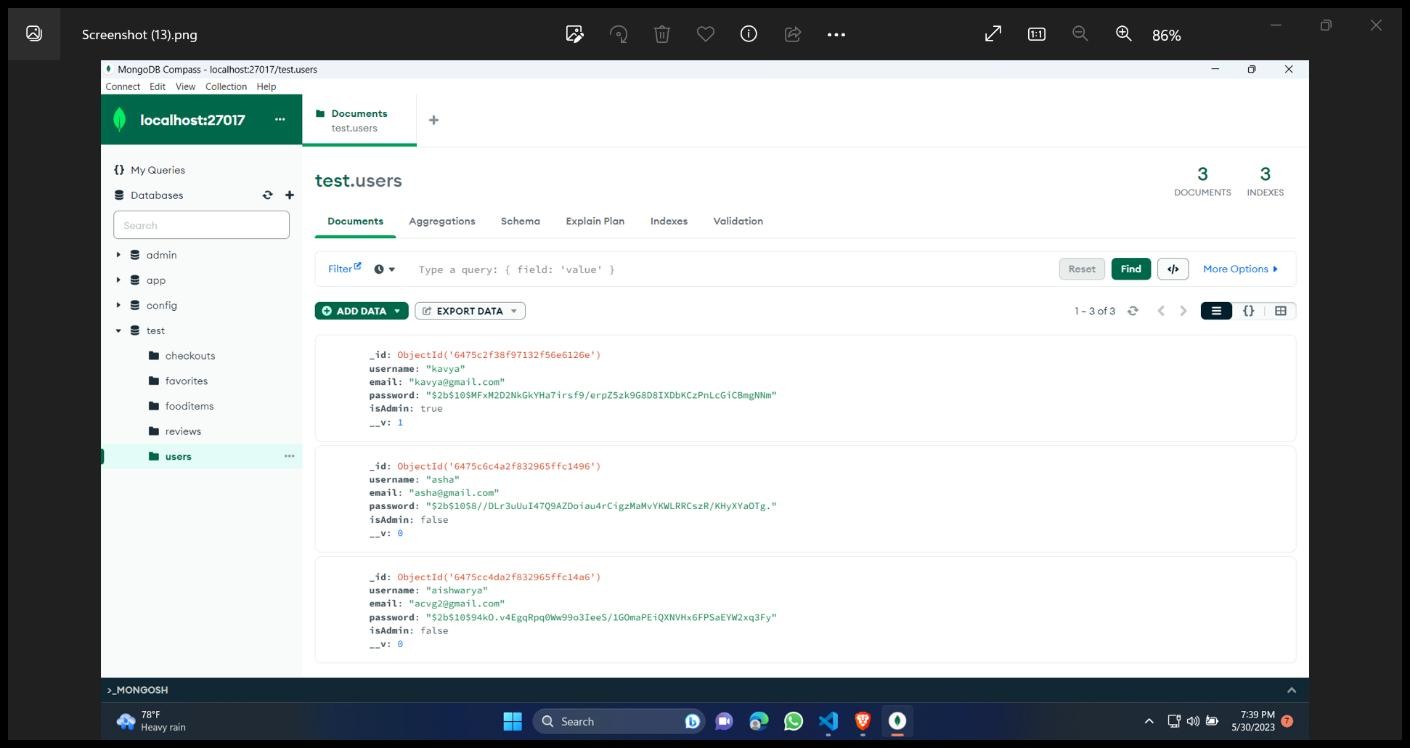


Figure:5.7

The user database in an online food delivery application is a central storage system that securely stores and manages user-related information. It serves as a foundation for user authentication, personalization, and order tracking. Here's a description of the key elements typically associated with a user database in an online food delivery application:

Each user has a profile in the database that contains their personal information, such as name, email address, phone number, and delivery address.

User profiles may also include additional details like preferred payment methods, dietary preferences, order history, or saved addresses.

### Product Database:

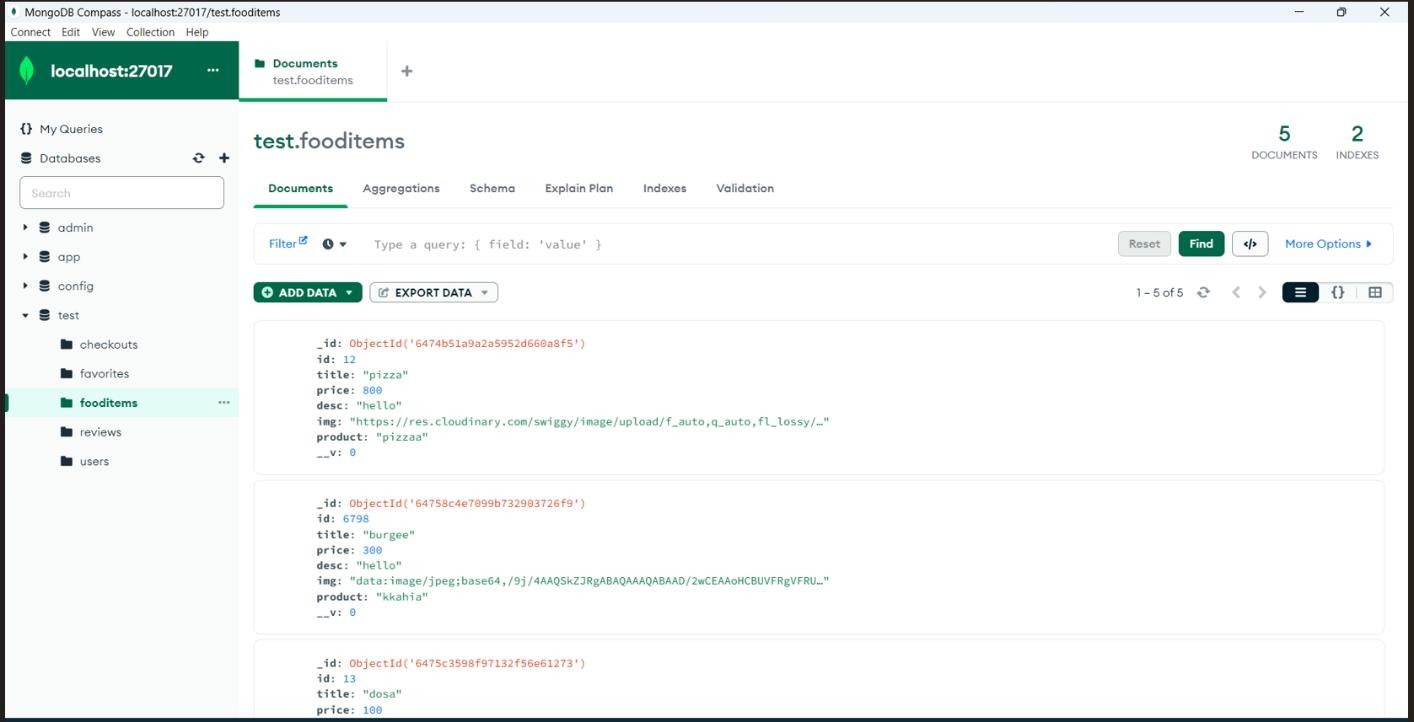


Figure:5.8

The food items database in an online food delivery application is a structured storage system that stores and manages information related to the various food products available for ordering. It serves as a central repository for storing details about menu items, prices, descriptions, images, and other relevant attributes. Here's a description of the key elements typically associated with a food items database in an online food delivery application:

* + - Each food item in the database has a unique identifier and includes information such as the item name, description, and category.
    - The description may provide details about the ingredients, cooking method, flavors, or any other relevant information that describes the food item.
    - The food items database stores pricing information for each item, indicating its cost.
    - Some food items may have different variations or options, such as size, toppings, or flavors. Each variation may have its own price associated with it.

The food items database may include the functionality to store customer reviews and ratings for each item.

* + - Users can leave feedback, ratings, and comments on the food item, allowing other users to make informed decisions based on the collective feedback.

### Checkouts Database:

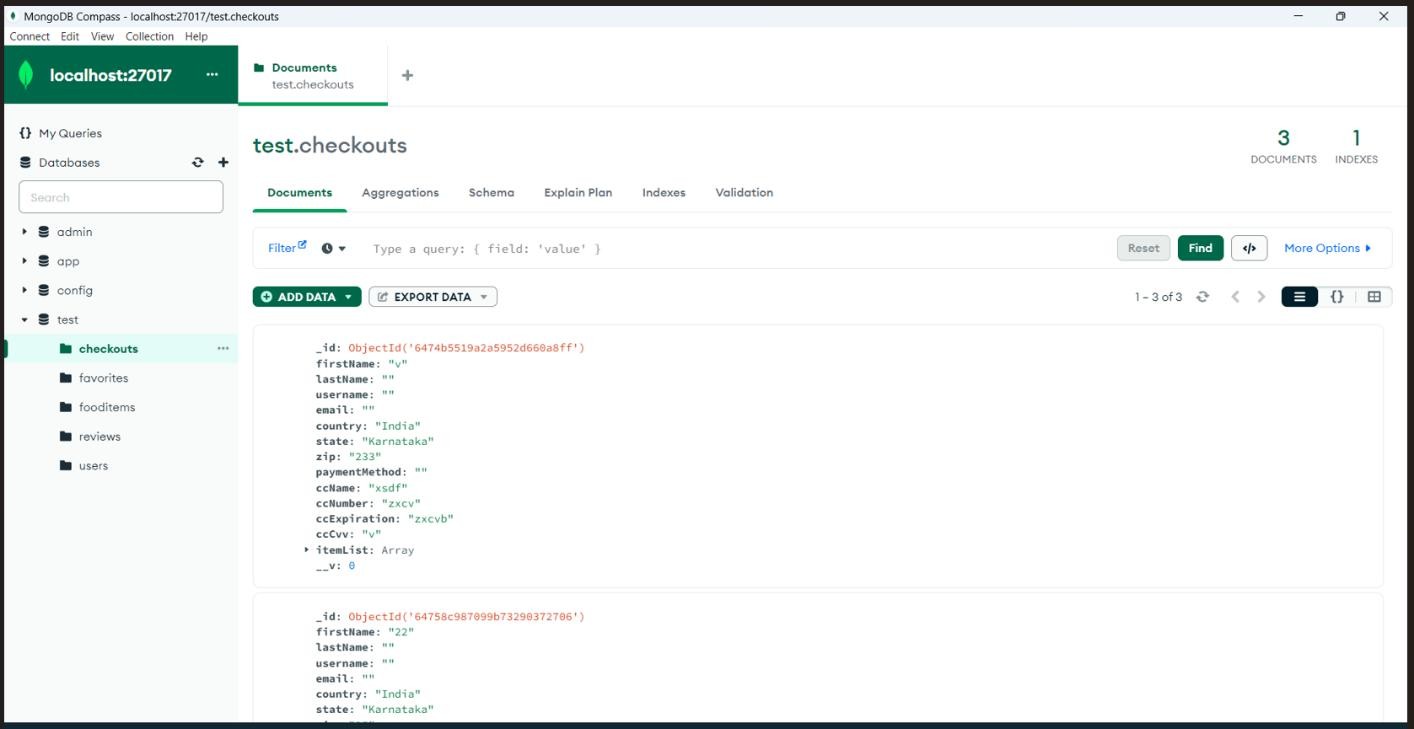


Figure:5.9

The checkout database in an online food delivery application is a component that manages the information related to the user's order and facilitates the payment process. It stores and tracks details about the items selected for purchase, delivery address, payment information, and order status. Here's a description of the key elements typically associated with a checkout database in an online food delivery application:

1. User Order Information:
   * The checkout database stores information about the user's order, including the selected food items, quantities, and any customizations or variations.
   * It associates the order with the user's unique identifier or session information to ensure proper tracking and retrieval.
2. Delivery Address:

* The checkout database includes the delivery address provided by the user for the order.
* It stores details such as the street address, city, state, and postal code to ensure accurate delivery.

1. Payment Information:
   * The checkout database securely stores the user's payment information for processing the transaction.
   * This can include details such as credit card information, PayPal account details, or other payment methods.
2. Order Total and Pricing:

The database tracks the total cost of the order, including item prices, taxes, delivery fees, and any discounts or promotional offers.

* + It calculates and stores the order total to facilitate payment processing and generate invoices if necessary.

1. Order Status and Tracking:
   * The checkout database maintains the status of the order, keeping track of its progress throughout the delivery process.
   * This can include statuses such as "Order Placed," "Preparing," "Out for Delivery," or "Delivered."
   * Order status updates can be stored along with timestamps to provide real-time tracking information to the user.

The checkout database plays a crucial role in managing the order and payment process in an online food delivery application. It securely stores user order information, handles payment processing, tracks order status, and maintains a history of past orders. By efficiently managing and organizing this information, the checkout database ensures a smooth and seamless checkout experience for users, contributing to overall customer satisfaction.

# CHAPTER 9: CONCLUSION

In conclusion, developing an online food delivery application using the MERN stack (MongoDB, Express.js, React.js, and Node.js) offers a powerful and efficient solution for building a robust and scalable platform. The MERN stack provides a full-stack JavaScript framework that allows for seamless communication between the client-side and server- side components, resulting in a cohesive and responsive user experience.

By leveraging MongoDB as the database, the application can store and manage large volumes of data related to users, food items, orders, and more. The flexibility of MongoDB's NoSQL structure allows for easy scalability and adaptability to changing business needs.

Express.js, as the backend framework, enables the creation of a RESTful API for handling user requests, managing authentication and authorization, and connecting with the database. It provides a lightweight and efficient approach to building server-side logic.

React.js, the frontend library, empowers the application with a component-based architecture and virtual DOM rendering. It facilitates the creation of a dynamic and interactive user interface, allowing users to browse food items, place orders, and track their delivery status in real-time.

Node.js, as the runtime environment, enables server-side JavaScript execution, providing high performance and scalability. It allows for non-blocking I/O operations, making the application responsive and capable of handling multiple concurrent requests.

Together, the MERN stack technologies provide a seamless development experience, allowing for efficient code reuse, rapid prototyping, and easy deployment. They offer a rich ecosystem of libraries, frameworks, and tools that further enhance the development process and enable the application to be tailored to specific business requirements.

Overall, building an online food delivery application using the MERN stack empowers businesses to create a feature-rich, scalable, and user-friendly platform that can cater to the growing demands of the food delivery industry. It enables seamless integration of various functionalities such as user authentication, menu management, order processing, and real-time tracking, resulting in an enhanced user experience and increased customer satisfaction.

## REFERENCES

1. <https://en.wikipedia.org/wiki/Online_food_ordering>
2. <https://react.dev/learn>
3. <https://expressjs.com/>
4. <https://nodejs.org/en>
5. <https://www.mongodb.com/>
6. <https://github.com/harshitkathuria/FoodTime>
7. <https://www.youtube.com/watch?v=v2ob3bpaLP8>
8. <https://youtu.be/3PXSI1fvJ3E>
9. <https://github.com/judygab/web-dev-pr>
10. <https://github.com/TheArhaam/MERN-FoodOrderingApp>