NEXT GEN EMPLOYABILITY PROGRAM

CREATING A FUTURE-READY WORKFORCE

Student Name :
Sandeep S K

Student ID :
STU67bb611cced0a1740333340

College Name :
Jain Institute of Technology,Davangere

CAPSTONE PROJECT SHOWCASE

A Service for Delivering Food (Zomato Clone) with MERN Technology

Abstract | Problem Statement | Project Overview | Proposed Solution | Technology Used | Modelling & Results | Conclusion | Q&A

Abstract

1 Food Delivery Platform – A MERN-based Zomato Clone enabling users to browse restaurants, add food items, and place orders.

2 Real-Time Updates & Order Management – Ensures instant reflection of restaurant listings, menu changes, and order processing in the system.

3 Secure Authentication & User Management – Implements JWT and bcrypt for user authentication, ensuring secure login, registration, and admin access

4 MERN Stack Deployment & Scalability – Uses MongoDB Atlas (database), Express.js & Node.js (backend), React.js (frontend), Render (backend hosting), and Vercel (frontend hosting) for a scalable and efficient system

Problem Statement

There is a need for a real-time, secure, and user-friendly food delivery system that allows users to browse restaurants, add food items, and place orders with instant updates. Existing platforms often lack real-time synchronization and secure authentication.

This project solves these issues by building a Zomato Clone using the MERN stack, enabling instant updates, secure user login, and efficient restaurant and order management.

Project Overview

The MERN-based Zomato Clone is a real-time, cloud-integrated food delivery system ensuring seamless restaurant management, dynamic menu updates, and efficient order processing.

❑ Secure Authentication – JWT-based security with bcrypt encryption for user and admin access.

❑ Real-Time Synchronization – Instant database updates via MongoDB Atlas & RESTful APIs.

❑ Scalable Cloud Deployment – Render (backend) & Vercel (frontend) for high-performance accessibility.

Proposed Solution

- Full-Stack Architecture – React.js (frontend), Node.js & Express.js (backend), and MongoDB Atlas (database) for seamless restaurant, menu, and order management.

- Real-Time Data Synchronization – Instant restaurant additions, menu updates, and order tracking using RESTful APIs & MongoDB Atlas.

- Secure Authentication & User Management – JWT-based authentication with bcrypt encryption for user and admin access control.

- Cloud Deployment & Scalability – Backend hosted on Render, frontend deployed on Vercel, ensuring high availability & performance.

Technology used

Frontend(client-side) :

React.js	Component-based UI for a dynamic and responsive interface
React Router	Navigation management for seamless routing between pages
Tailwind CSS / Bootstrap	Styling and responsive design
Axios / Fetch API	Handling HTTP requests to the backend

Database and storage:

MongoDB Atlas	Cloud-based NoSQL database for storing users, restaurants, menu items, and orders.

Technology used

Backend (Server-Side) :

Node.js	JavaScript runtime for handling server-side logic.
Express.js	Framework for building RESTful APIs and handling requests.
JWT/JSON Web Token	Secure authentication for users and admins.
bcrypt.js	Password hashing for enhanced security.

Deployment & Hosting:

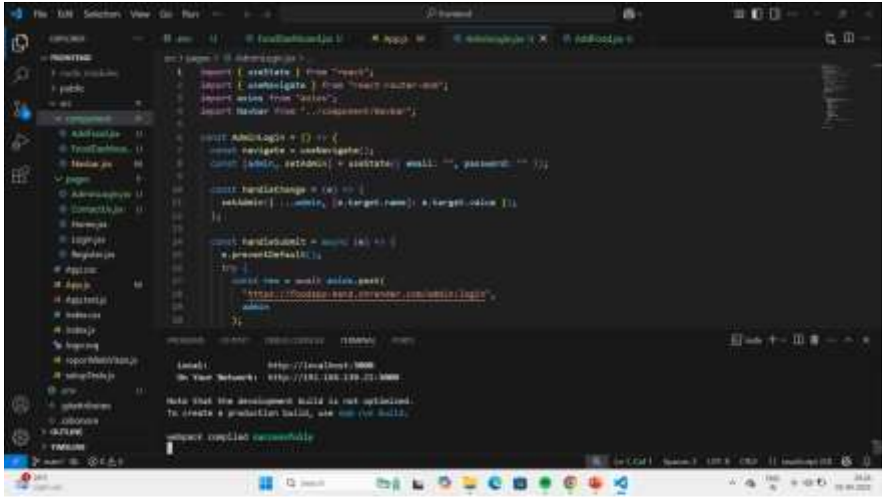
Render	Cloud platform for deploying the backend (Node.js & Express server).
Vercel	Hosting the frontend (React.js) for scalability and performance.

Technology used

Additional Tools & Technologies:

Postman	API testing for backend endpoints.
Git & GitHub	Version control and collaborative development.
Dotenv	Environment variable management for secure API keys and database credentials.

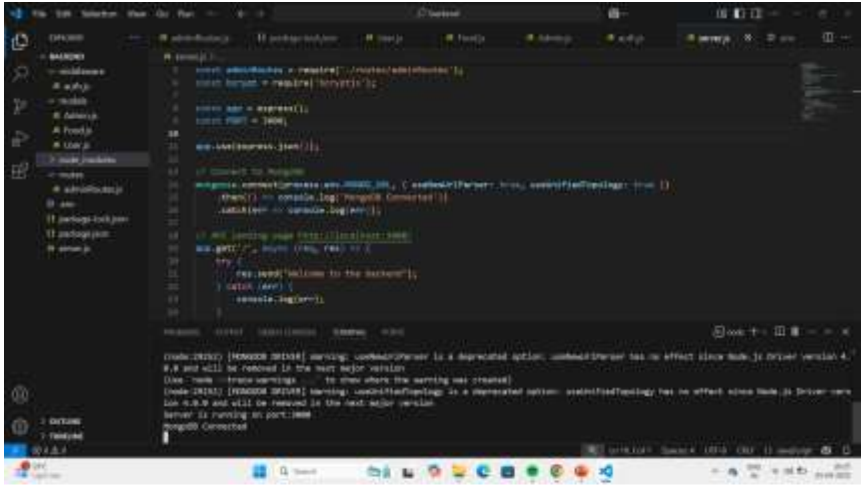
Modelling & Result



```
import { useState } from "react";
import { useNavigate } from "react-router-dom";
import axios from "axios";
import Navbar from "../component/Navbar";

const AdminLogin = () => {
  const navigate = useNavigate();
  const [admin, setAdmin] = useState({ email: "", password: "" });

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

  const handleSubmit = async (e) => {
    e.preventDefault();
    try {
      const res = await axios.post(
        "https://foodapp-bend.onrender.com/admin/login",
        admin
      );
```

```
  Local:            http://localhost:3000
  On Your Network:  http://192.168.139.21:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

Fig. Frontend Code Folder

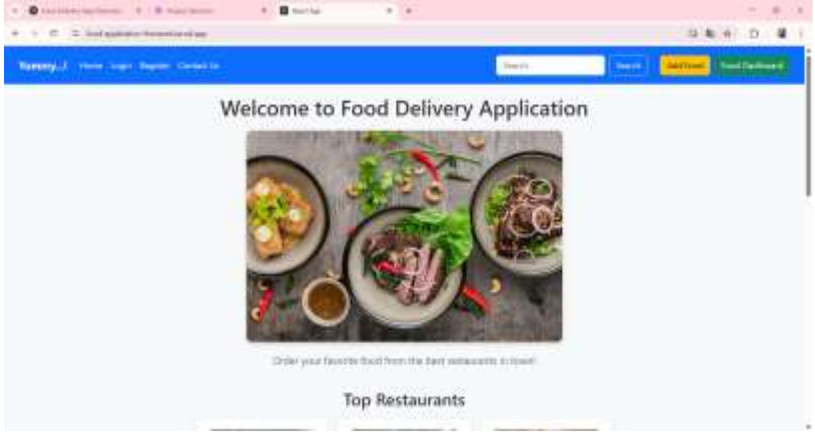


```
const adminRoutes = require('./routes/adminRoutes');
const bcrypt = require('bcryptjs');

const app = express();
const PORT = 3000;

app.use(express.json());

// Connect to MongoDB
mongoose.connect(process.env.MONGO_URI, { useNewUrlParser: true, useUnifiedTopology: true })
    .then(() => console.log('MongoDB Connected'))
    .catch(err => console.log(err));

// API landing page http://localhost:3000
app.get('/', async (req, res) => {
    try {
        res.send('Welcome to the backend');
    } catch (err) {
        console.log(err);
    }
```

```
(node:28152) [MONGODB DRIVER] Warning: useNewUrlParser is a deprecated option: useNewUrlParser has no effect since Node.js Driver version 4.
0.0 and will be removed in the next major version
(Use `node --trace-warnings ...` to show where the warning was created)
(node:28152) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect since Node.js Driver vers
ion 4.0.0 and will be removed in the next major version
Server is running on port:3000
MongoDB Connected
```

Fig. Backend code Folder

Modelling & Result

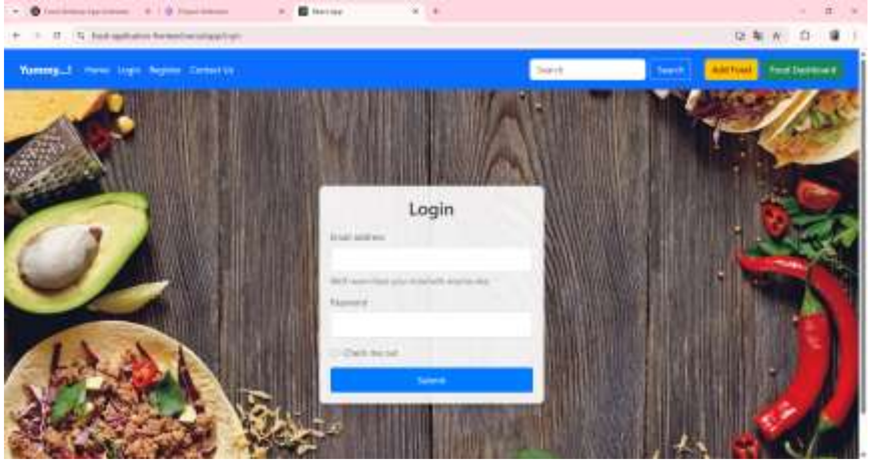


Welcome to Food Delivery Application

Order your favorite food from the best restaurants in town!

Top Restaurants

Fig. Home Page



Login

Email address

We'll never share your email with anyone else.

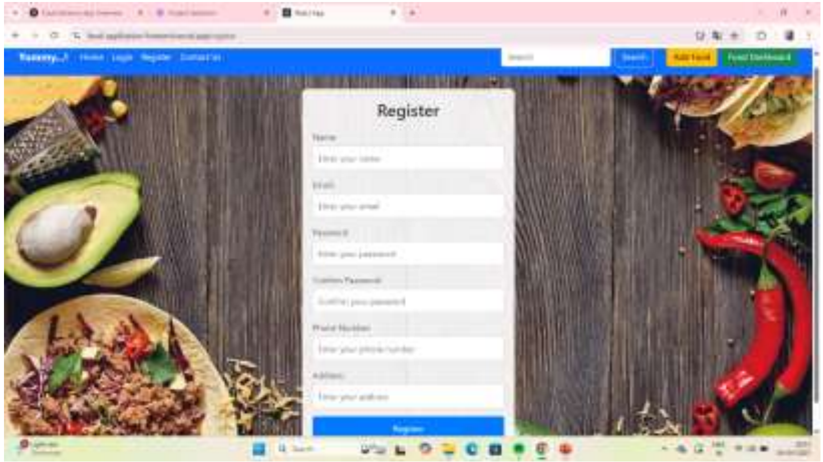
Password

□ Check me out

Submit

Fig. User Login Page

Modelling & Result



Yummy...! Home Login Register Contact Us

Search | Add Food | Food Dashboard

Register

Name
Enter your name

Email
Enter your email

Password
Enter your password

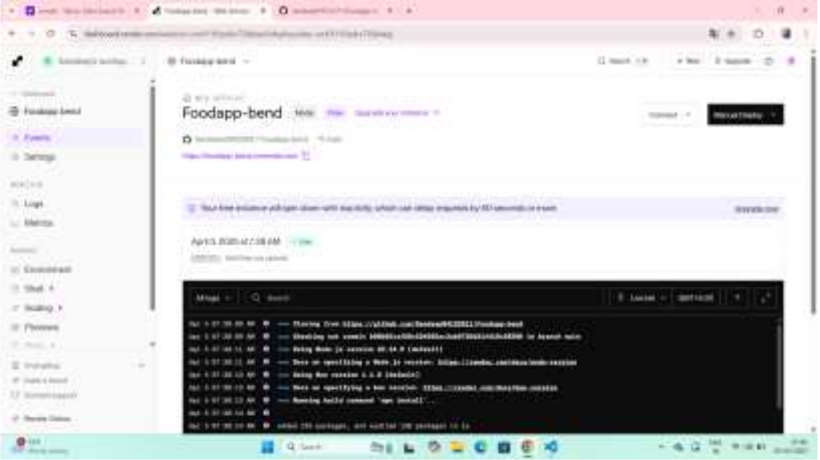
Confirm Password
Confirm your password

Phone Number
Enter your phone number

Address
Enter your address

Register

Fig. User Registration



Foodapp-bend

```
==> Cloning from https://github.com/[illegible]/foodapp-bend
==> Checking out commit [illegible] in branch main
==> Using Node.js version [illegible] (default)
==> Docs on specifying a Node.js version: [illegible]
==> Using Bun version [illegible] (default)
==> Docs on specifying a bun version: [illegible]
==> Running build command 'npm install'...
added [illegible] packages, and audited [illegible] packages in [illegible]
```

Fig. Deployment of Backend in Render

Conclusion

The Zomato Clone using MERN stack successfully demonstrates a full-stack, real-time food delivery application with features like secure authentication, restaurant and menu management, and instant order updates. By integrating MongoDB Atlas, Express.js, React.js, and Node.js, along with cloud deployment on Render and Vercel, the project ensures scalability, performance, and a smooth user experience.

This project not only fulfills the functional requirements of a food delivery system but also provides practical hands-on experience in modern web development, making it a strong foundation for building real-world, scalable applications.

Thank you!

edunet
foundation