

FULL STACK DEVELOPMENT WITH MERN PROJECT DOCUMENTATION

1. Introduction

Project Title: OrderOnTheGo: Your On-Demand Food Ordering Solution

Team ID: LTVIP2025TMID55113

Team Roles:

Team Leader: Kolla Yaraswini - Project coordination, GitHub management, system integration

Frontend Developer: Kemburu Surendra Kumar - React UI, component design, user interactions

Backend Developer: Kodeti Likhitha - RESTful APIs, authentication, server logic

Database Admin: Kasula Lakshmi Gnana - MongoDB schemas, data consistency

2. Project Overview

Purpose:

Revolutionize food ordering with real-time availability tracking, multi-role access, and seamless transactions.

Key Features:

User: Browse menus with filters, Cart functionality, Order history, Address/payment profiles

Restaurant: Menu management, Order fulfillment, Sales tracking, Performance reports

Admin: Restaurant approvals, User/order analytics, System configuration, Content moderation

3. Technical Architecture

Frontend: React, Redux, Material-UI

Backend: Node.js, Express.js, JWT

Database: MongoDB, Mongoose

4. Setup & Installation

Prerequisites: Node.js >= 18.x, MongoDB Atlas, npm/yarn

Setup:

git clone [https://github.com/Kemburu-](https://github.com/Kemburu-Surendrakumar/OrderOnTheGo-Your-On-Demand-Food-Ordering-Solution-/tree/main/project)

[Surendrakumar/OrderOnTheGo-Your-On-Demand-Food-Ordering-Solution-/tree/main/project](https://github.com/Kemburu-Surendrakumar/OrderOnTheGo-Your-On-Demand-Food-Ordering-Solution-/tree/main/project)

cd server && npm install

cd ../client && npm install

5. Folder Structure

SBFoods/

├── client/

│ ├── src/

│ │ ├── components/

│ │ ├── pages/

│ │ ├── redux/

│ │ ├── App.js

│ │ └── index.js

└── server/

├── controllers/

├── models/

├── routes/

├── server.js

└── .env

6. Application Execution

Frontend: `cd client && npm start` (`http://localhost:3000`)

7. API Documentation

POST `/api/users/register` - User registration (No Auth)

POST `/api/users/login` - User authentication (No Auth)

GET `/api/products` - Browse all dishes (No Auth)

POST `/api/orders` - Create new food order (Yes - User)

PATCH `/api/restaurants` - Update menu items (Yes - Restaurant)

GET `/api/admin/approvals` - Pending approvals (Yes - Admin)

8. Authentication System

JWT-based authentication with HTTP-only cookies

Role-based access control

Users: `{ role: 'user' }`

Restaurants: `{ role: 'restaurant', approved: false }`

Admins: `{ role: 'admin' }`

Password encryption with `bcryptjs`

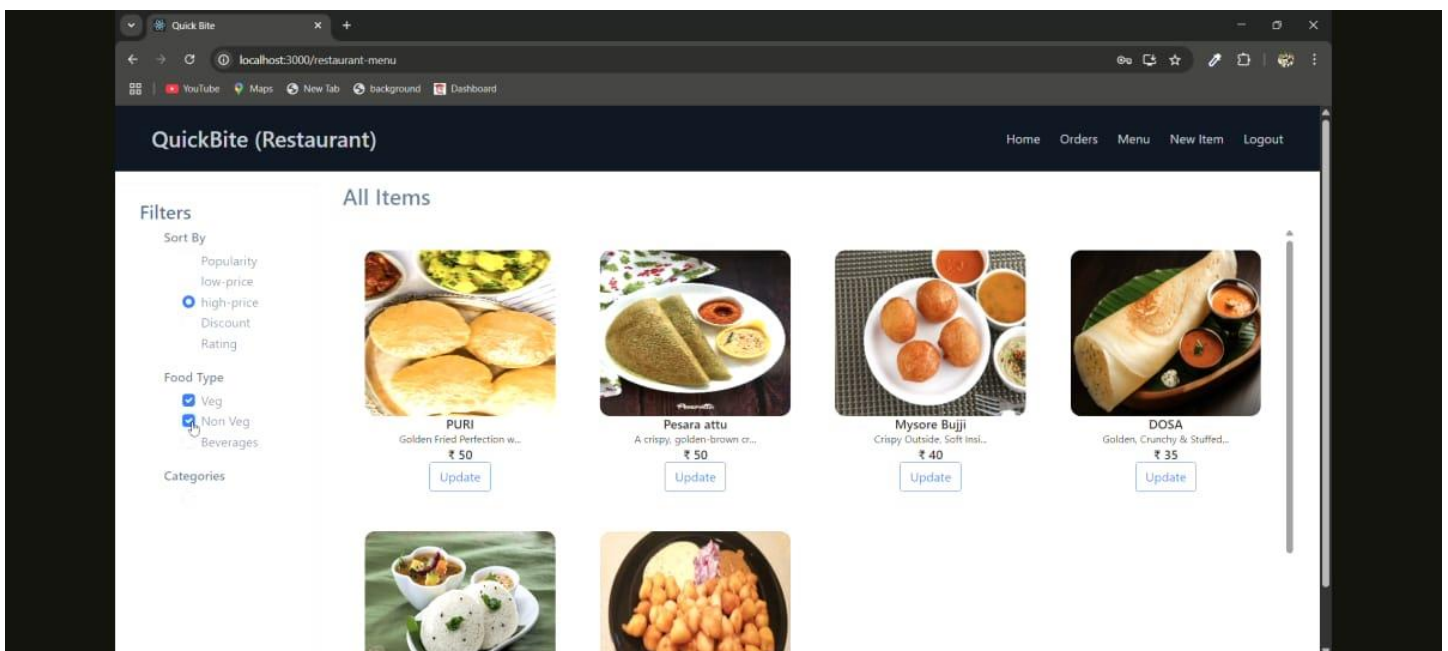
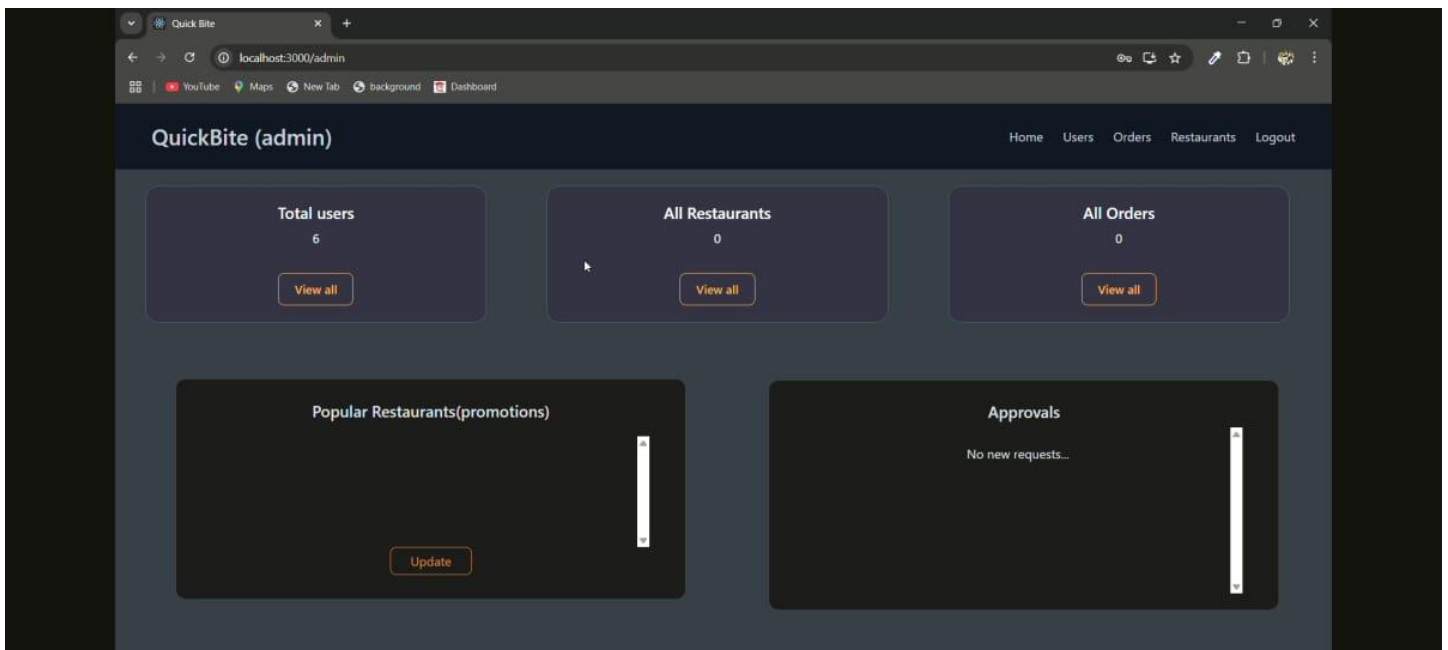
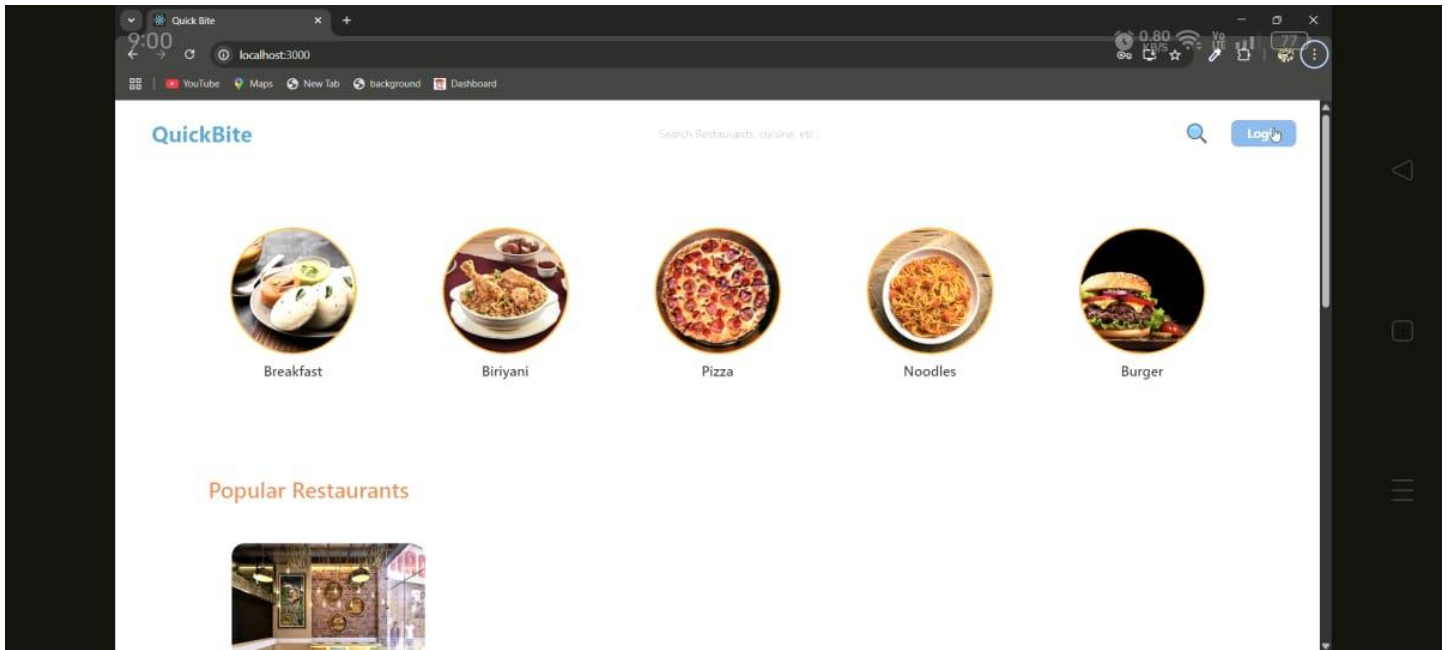
9. User Interface Workflows

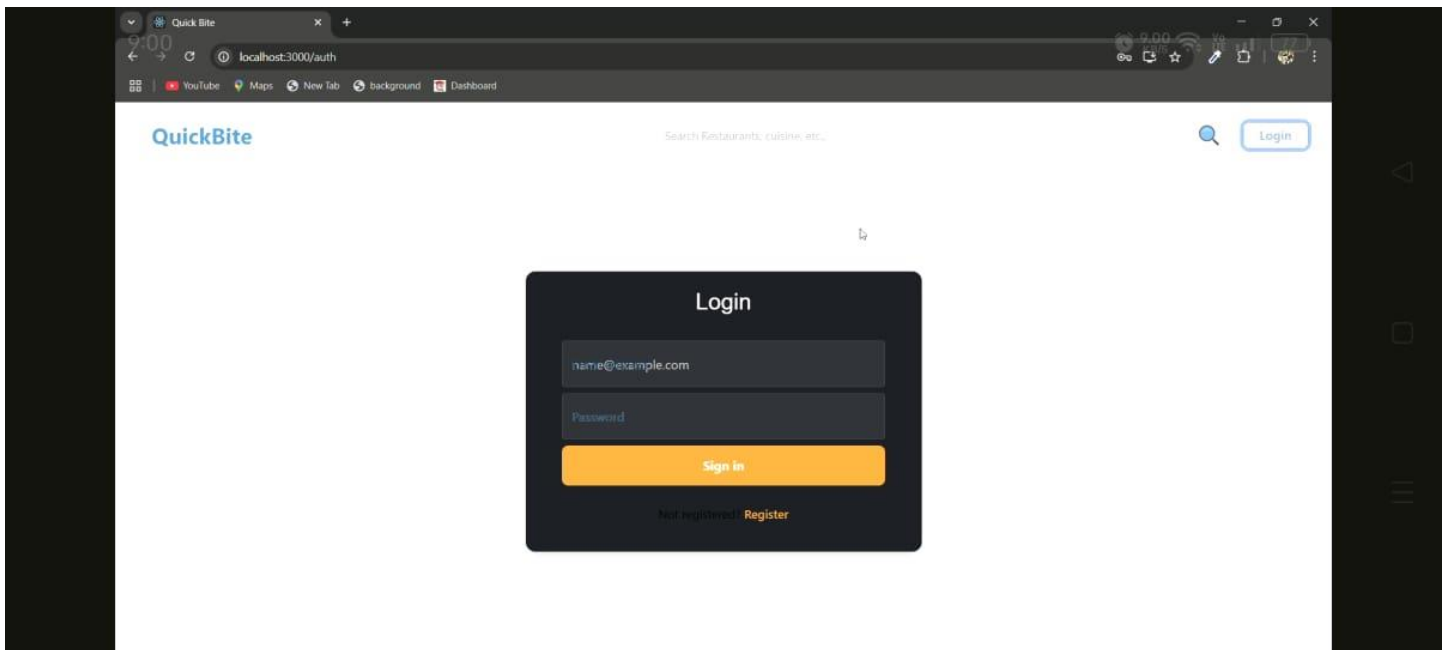
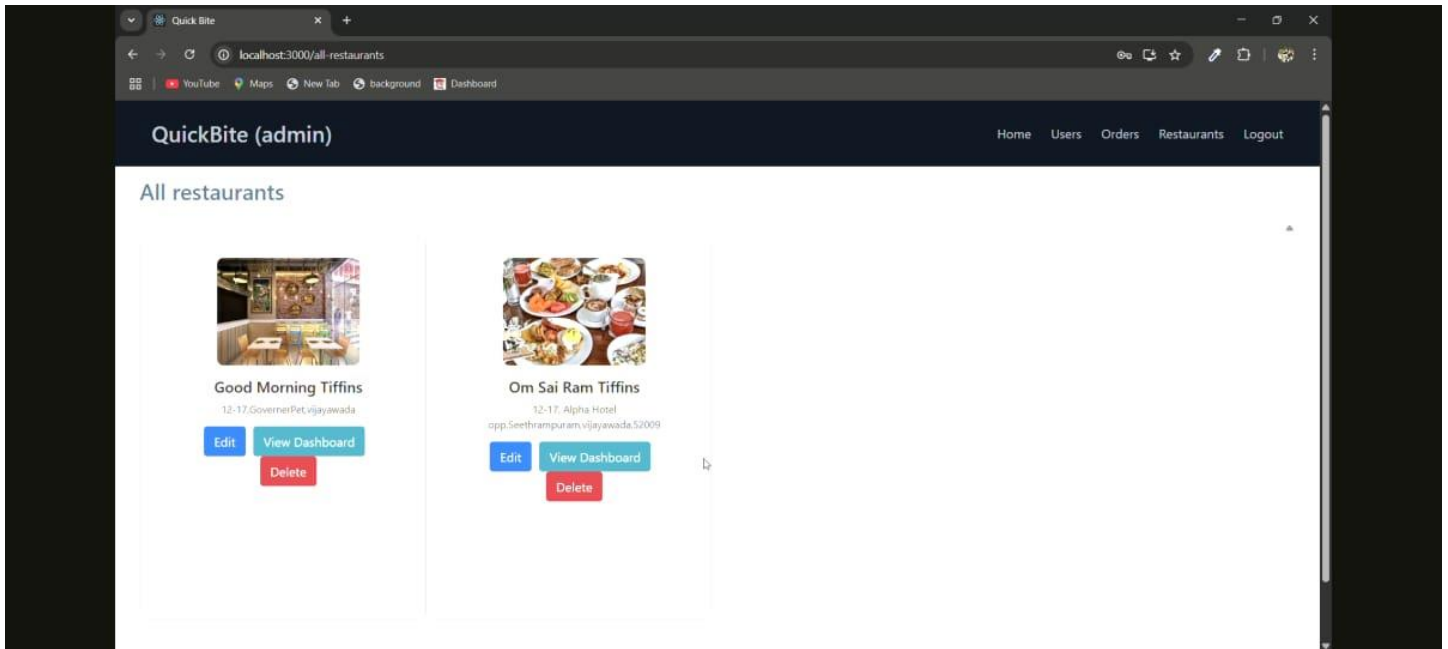
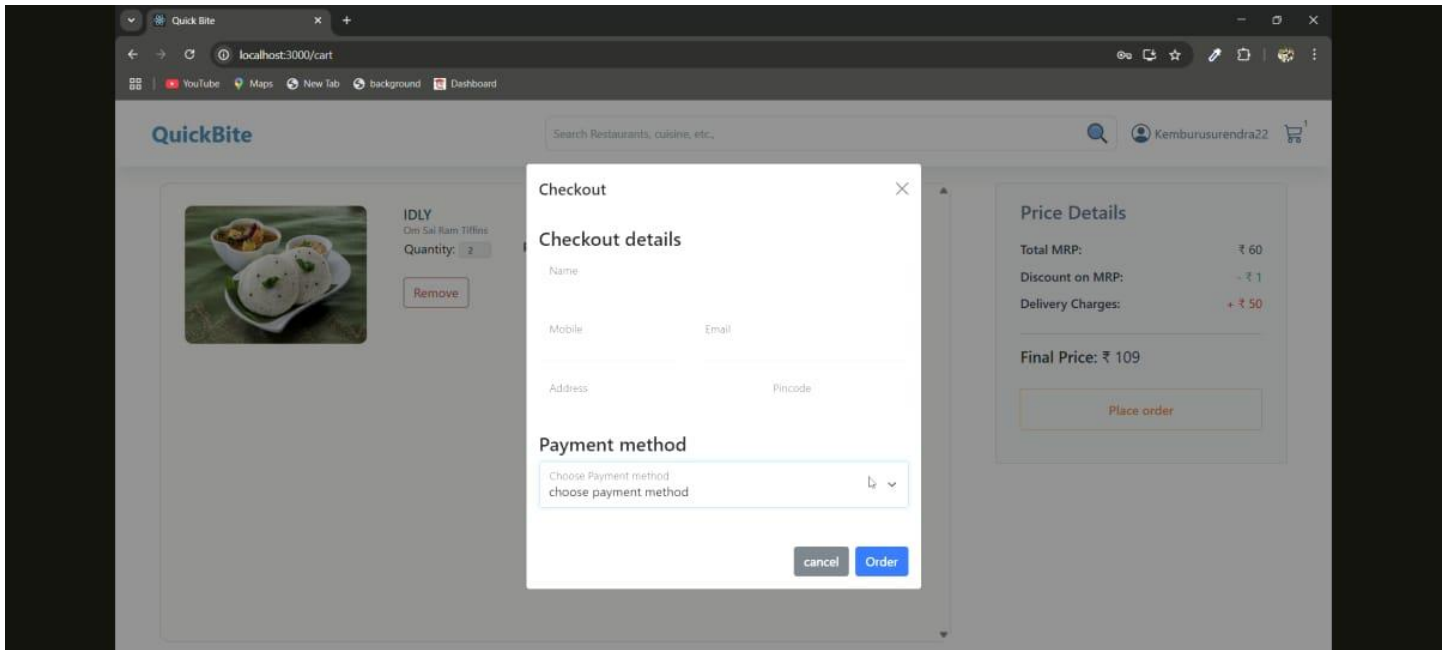
1. User Registration

2. Food Discovery

3. Order Processing

4. Admin Dashboard





10. Testing Protocol

API Testing: Postman, Jest

UI Testing: React Testing Library

Load Testing: Artillery.io

11. Demo Video

Demo

Link:

<https://drive.google.com/file/d/1FWtPXrCN1OmsjP9K5IWtcvb7ofhLoM0n/view?usp=drivesdk>

12. Known Issues

Mobile UI responsiveness - High - In progress

Image upload limitations - Medium - Planned

Password recovery - Medium - Backlog

13. Future Roadmap

1. Payment Integration

2. Real-time Tracking

3. Recommendation Engine

4. Multi-vendor Support

"Transforming food ordering experiences through innovative technology"

Security Testing: OWASP ZAP

Load Testing: Artillery.io

14. Demo Video

Demo

Link:

<https://drive.google.com/file/d/1FWtPXrCN1OmsjP9K5IWtcvb7ofhLoM0n/view?usp=drivesdk>

15. Known Issues

Mobile UI responsiveness - High - In progress

Image upload limitations - Medium - Planned

Password recovery - Medium - Backlog

16. Future Roadmap

5. Payment Integration

6. Real-time Tracking

7. Recommendation Engine

8. Multi-vendor Support

"Transforming food ordering experiences through innovative technology"