**ORDER ON THE GO: YOUR ON-DEMAND FOOD ORDERING SOLUTION**

## 1. Introduction

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

* Niteesha Tunuguntla: Developer (Frontend + Backend+ All)
* Thota Harshitha
* Telluri Santhosh Kumar
* Tarun Venkat Sai

## 2. Project Overview

**Purpose**:  
OrderOnTheGo is a full-stack MERN application that simplifies food ordering. It allows users to browse a rich menu, filter based on food categories, add items to the cart, and checkout with cash or UPI. Admins can manage orders, users, and menu items.

**Key Features**:

* User registration/login
* Product listing with filters
* Add to cart and place order
* Payment options: Cash on Delivery & UPI
* User profile with order tracking
* Admin dashboard to manage users, products, and orders

## 3. Architecture

**Frontend**:

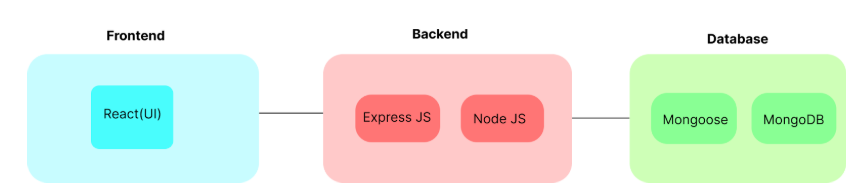
* React.js (Functional Components)
* React Router for routing
* Context API for cart management
* CSS for styling

**Backend**:

* Node.js with Express.js
* RESTful APIs
* Authentication logic (simple session-based)

**Database**:

* MongoDB
* Mongoose for ODM
* Collections: Users, Products, Orders, Admin



## 4. Setup Instructions

**Prerequisites**:

* Node.js v18+
* MongoDB (Atlas or local)
* Git

**Installation**:

1. Clone the repository:

<https://github.com/TunuguntlaGouriNiteesha/MERNSTACK-final-project>

1. Navigate to backend and install dependencies:

cd server  
 npm install

1. Navigate to frontend and install dependencies:

cd ../client  
 npm install

1. Set up MongoDB connection string in server/.env

MONGODB\_URI= <https://food-0us6.onrender.com>

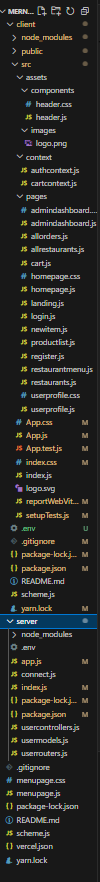
## 5. Folder Structure

**Client (React)**:

client/  
 src/  
 pages/ # Contains cart, login, register, productlist, etc.  
 components/ # Header, Navbar  
 context/ # Cart context  
 assets/images # Food images and logo

**Server (Node + Express)**:

server/  
 index.js # Server entry point  
 models/ # Mongoose schemas  
 routes/ # API route handlers



## 6. Running the Application

**Frontend**:

cd client  
npm start

**Backend**:

cd server

node index.js or node app.js

## 7. API Documentation

* **GET** /api/products: Fetch all food items
* **POST** /api/register: Register user
* **POST** /api/login: Login user
* **POST** /api/cart: Add to cart
* **POST** /api/orders: Place order
* **GET** /api/orders/: userId: Get user orders

## 8. Authentication

* Simple login logic with localStorage token simulation
* Only authenticated users can access cart and profile
* Admin routes restricted

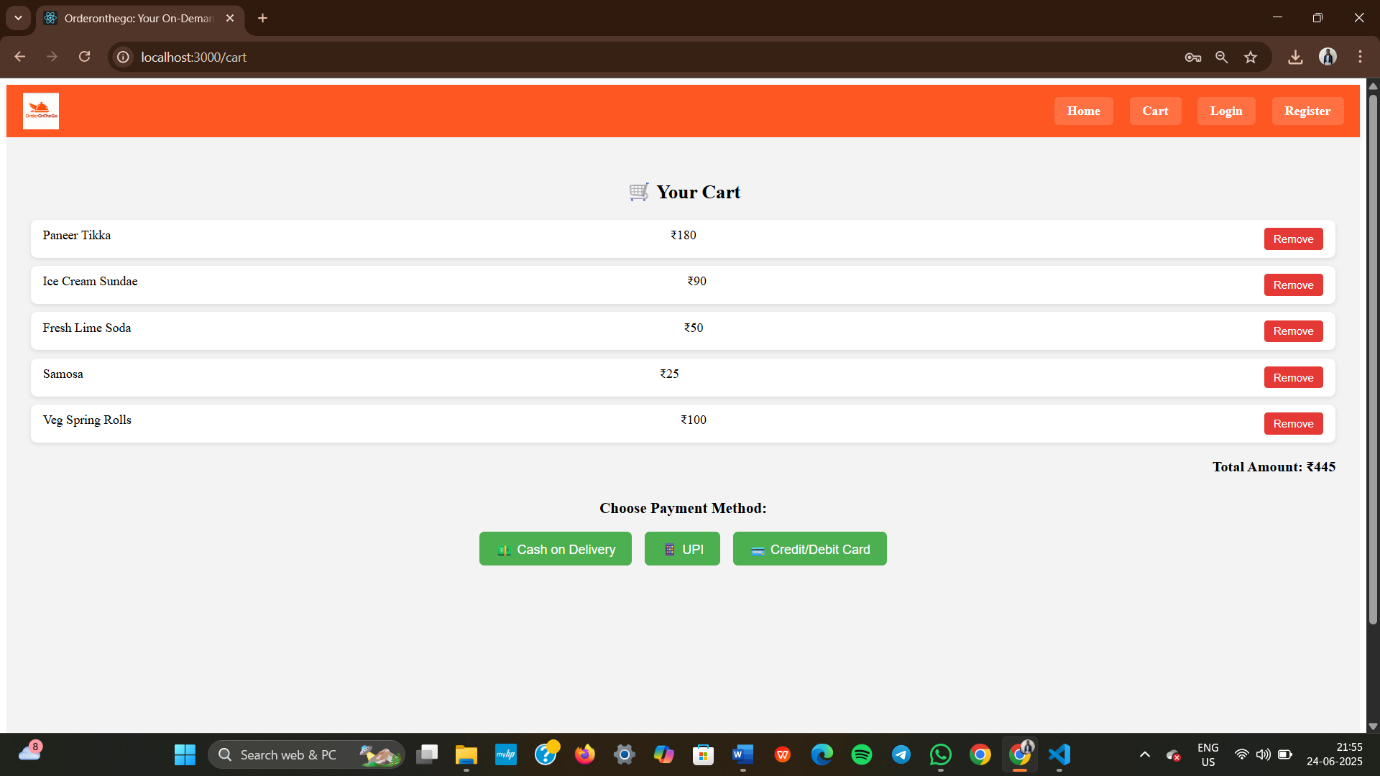
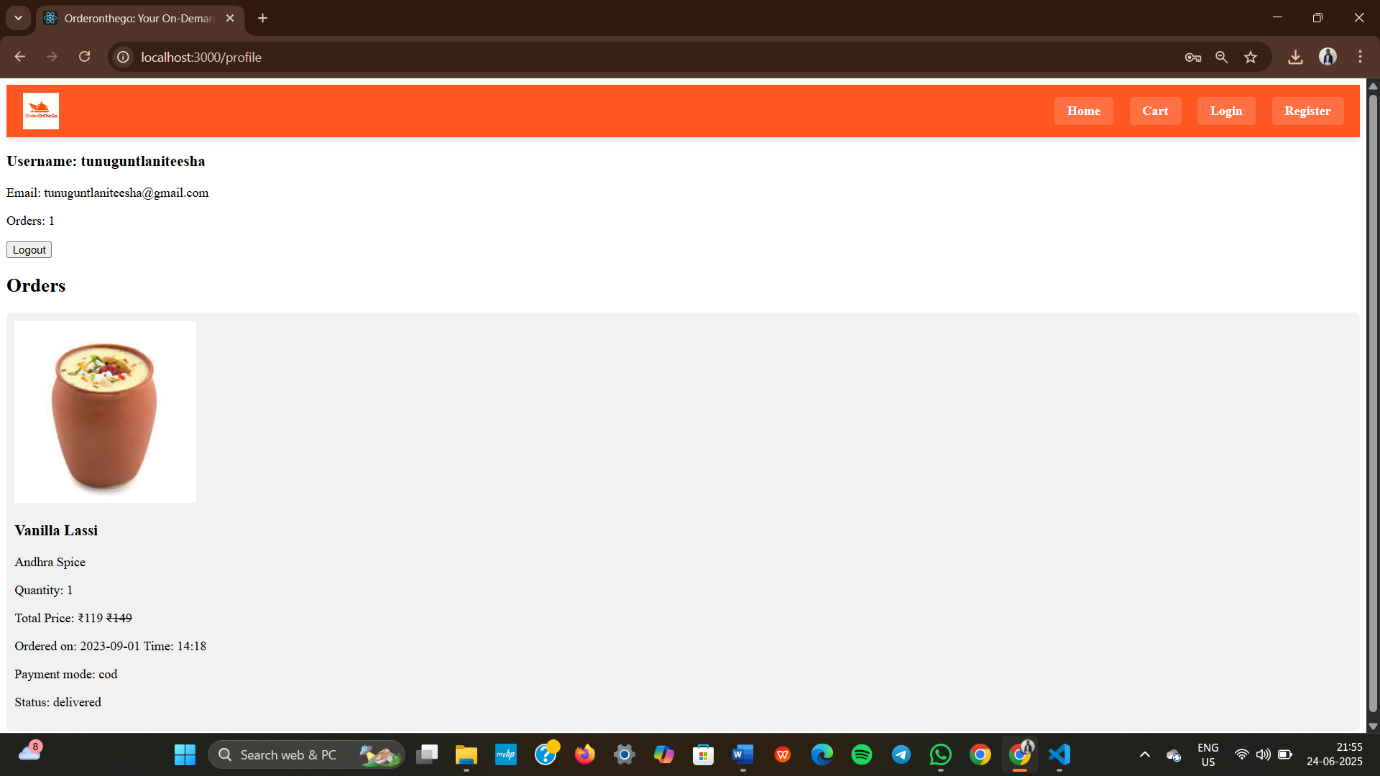
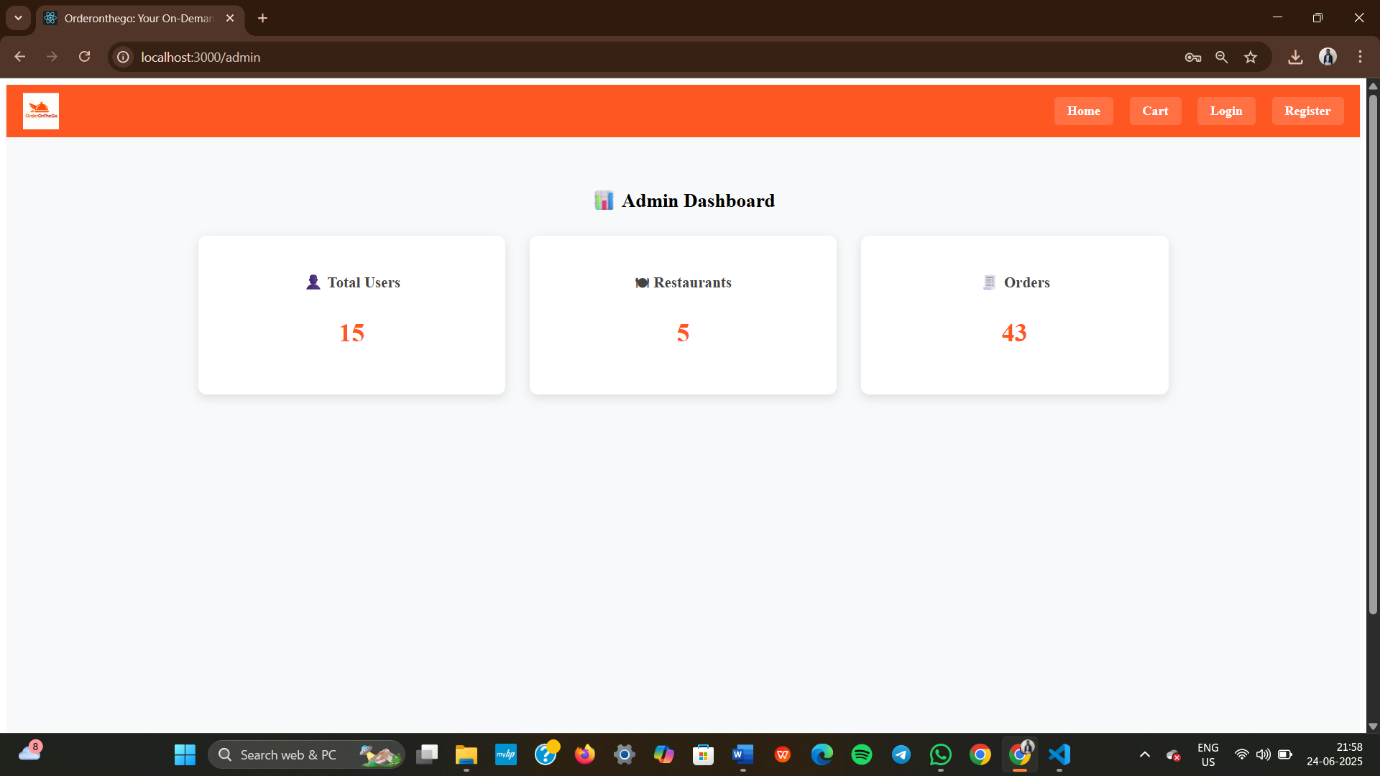
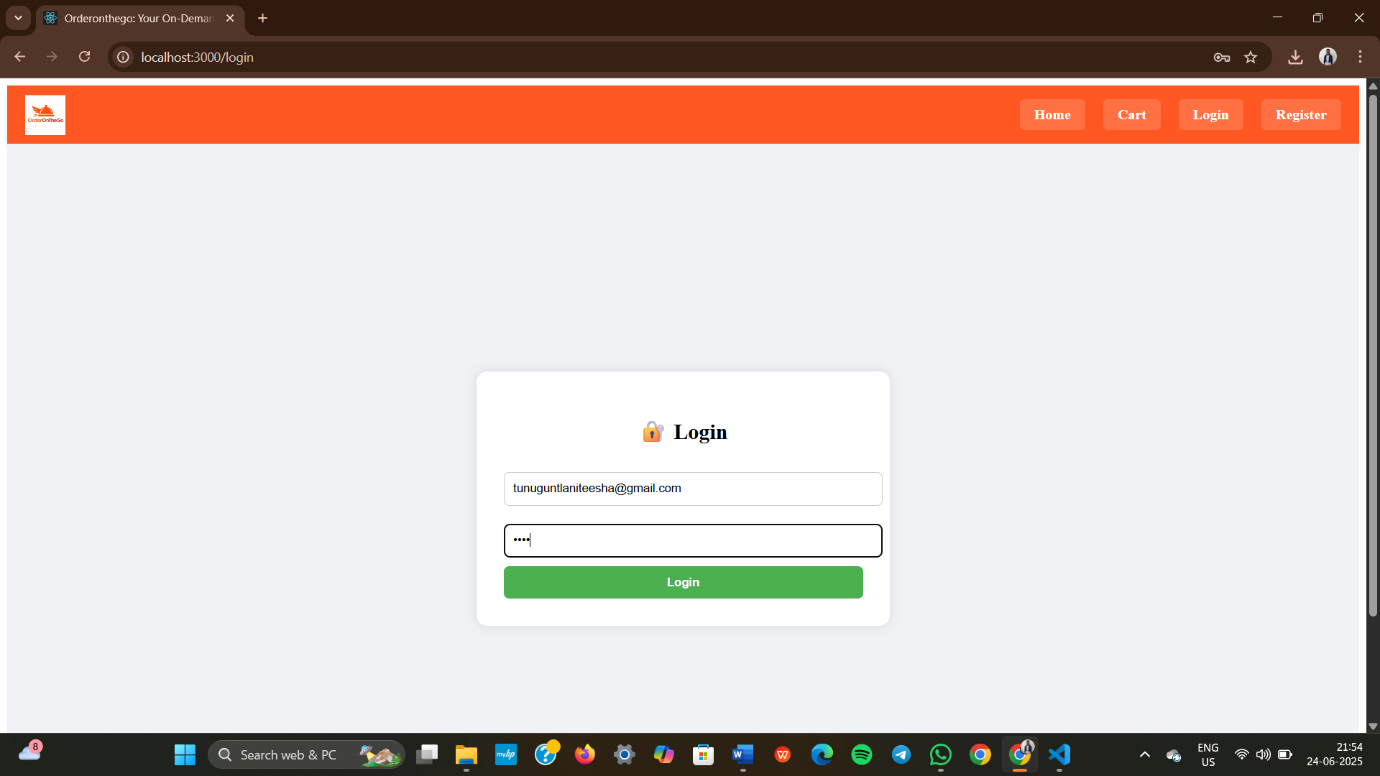
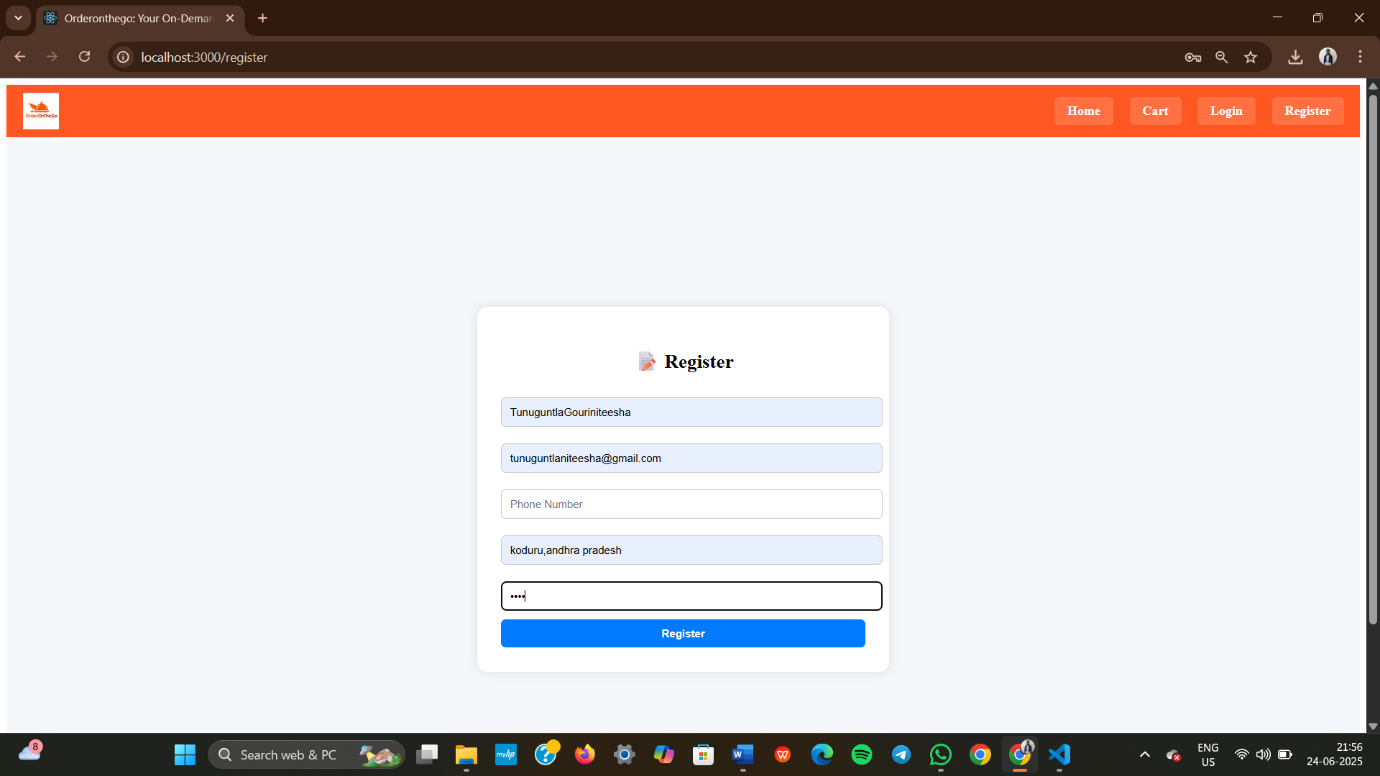
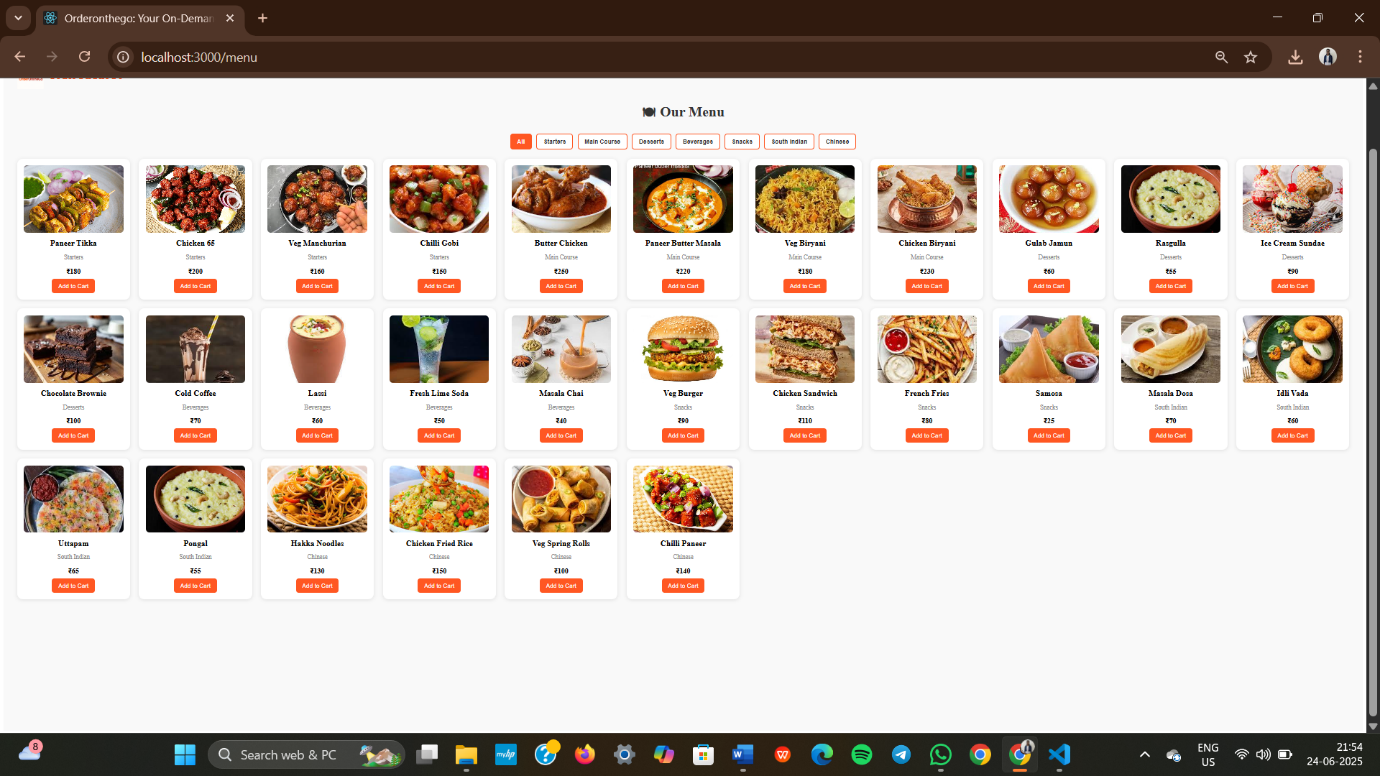
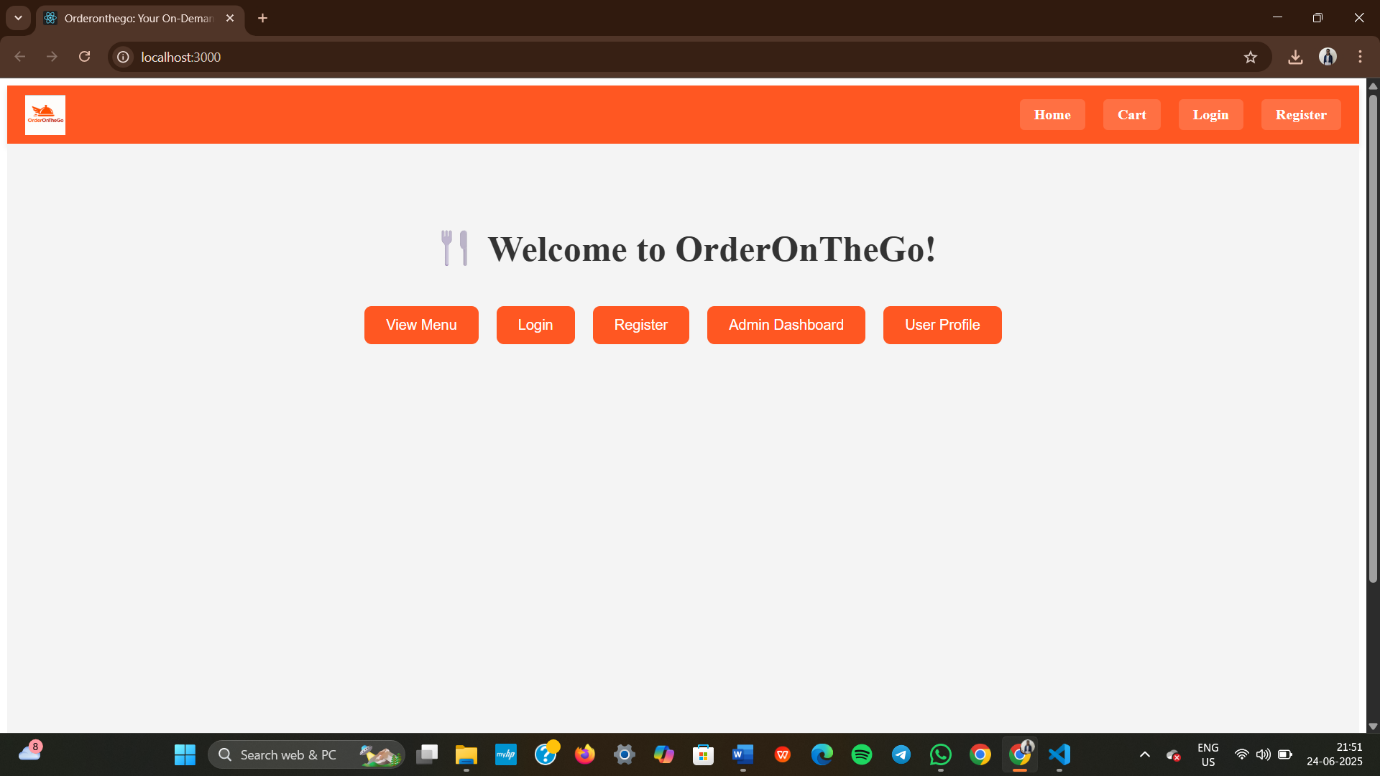
## 9. User Interface

* Responsive design
* Category filters (Starters, Main Course, etc.)
* Cart summary with price
* Profile page showing order history and user info

## 10. Testing

* Manual testing of all routes
* UI testing on Chrome

## 11. Screenshots or Demo

* **Landing Page**: Displays full menu
* **Cart Page**: Shows added items and total
* **Login/Register**: For user accounts
* **Admin Dashboard**: View all orders, users
* 

## 12. Known Issues

* No persistent login (uses frontend logic only)
* No image upload for new items

## 13. Future Enhancements

* JWT-based authentication
* Restaurant onboarding module
* Dynamic image uploads
* Delivery partner integration
* PWA support for mobile ordering

**Happy Ordering with OrderOnTheGo!**