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

INTRODUCTION

Welcome to **OrderOnTheGo**, a modern food ordering web application designed to simplify the process of browsing, ordering, and receiving food online. The system streamlines the customer experience with a responsive UI, diverse menu filters, and robust backend logic.

Whether you're craving a quick snack, a full meal, or a sweet dessert, **OrderOnTheGo** connects users with a wide variety of options from different categories including Starters, Main Course, Desserts, Beverages, and more. With secure payment integration including **Cash on Delivery** and **UPI**, users enjoy a seamless experience from cart to checkout.

**SCENARIO:**

**Late-Night Craving Resolution**

Meet Lisa, a college student burning the midnight oil to finish her assignment. As the clock strikes midnight, her stomach grumbles, reminding her that she skipped dinner. Lisa doesn't want to interrupt her workflow by cooking, nor does she have the energy to venture outside in search of food.

Solution with Food Ordering App:

1. Lisa opens the Food Ordering App on her smartphone and navigates to the late-night delivery section, where she finds a variety of eateries still open for orders.

2. She scrolls through the options, browsing menus and checking reviews until she spots her favorite local diner offering comfort food classics.

3. Lisa selects a hearty bowl of chicken noodle soup and a side of garlic bread, craving warmth and satisfaction in each bite.

4. With a few taps, she adds the items to her cart, specifies her delivery address, and chooses her preferred payment method.

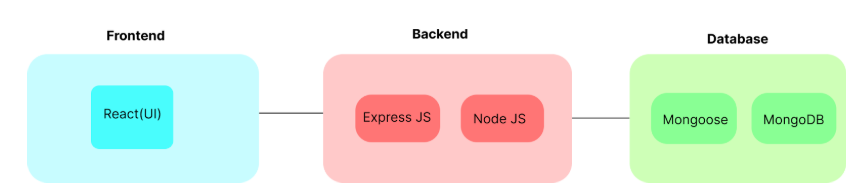
5. Lisa double-checks her order details on the confirmation page, ensuring everything looks correct, before tapping the "Place Order" button.

6. Within minutes, she receives a notification confirming her order and estimated delivery time, allowing her to continue working with peace of mind.

7. As promised, the delivery arrives promptly at her doorstep, and Lisa eagerly digs into her piping hot meal, grateful for the convenience and comfort provided by the Food Ordering App during her late-night study session.

This scenario illustrates how a Food Ordering App caters to users' needs, even during unconventional hours, by offering a seamless and convenient solution for satisfying late-night cravings without compromising on quality or convenience.

**TECHNICAL ARCHITECTURE:**



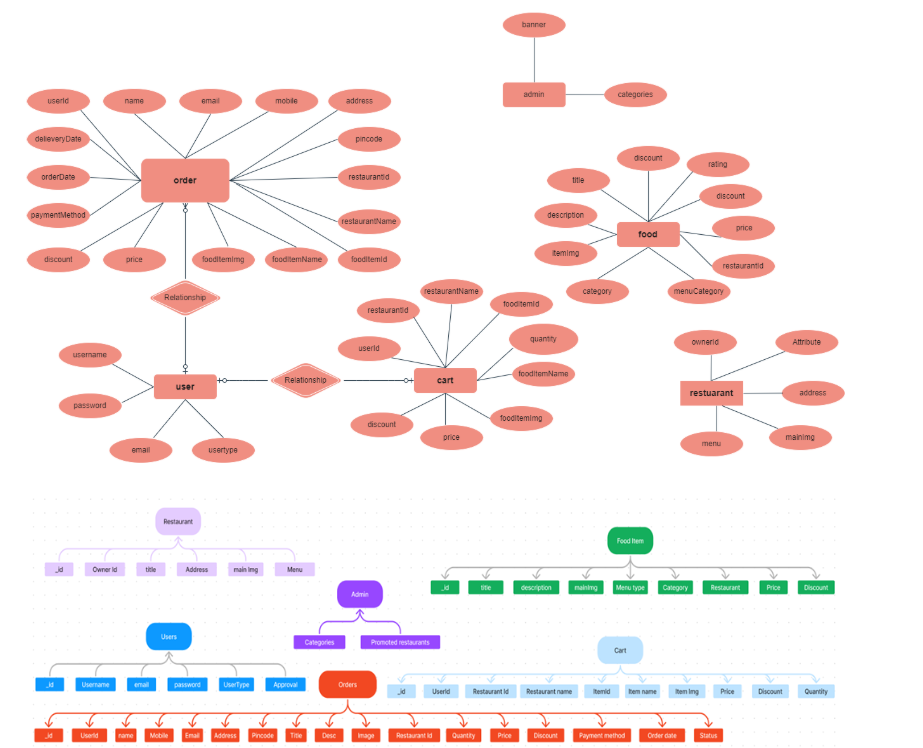
In this architecture diagram:

•        The frontend is represented by the "Frontend" section, including user interface components such as User Authentication, Cart, Products, Profile, Admin dashboard, etc.,

•        The backend is represented by the "Backend" section, consisting of API endpoints for Users, Orders, Products, etc., It also includes Admin Authentication and an Admin Dashboard.

•        The Database section represents the database that stores collections for Users, Admin, Cart, Orders, and products.

**ER DIAGRAM:**



**User:** Represents the individuals or entities who are registered in the platform.

**Restaurant**: This represents the collection of details of each restaurant in the platform. **Admin:** Represents a collection with important details such as promoted restaurants and Categories.

**Products:** Represents a collection of all the food items available in the platform.

**Cart:** This collection stores all the products that are added to the cart by users. Here, the elements in the cart are differentiated by the user Id.

**Orders:** This collection stores all the orders that are made by the users in the platform.

**FEATURES**

* Menu filtering by category
* User authentication
* Admin dashboard to monitor orders
* Add to cart, place order
* Cash on Delivery and UPI
* Profile page with order count and details

**PREREQUISITES**

* Node.js and npm
* MongoDB Atlas
* Git
* React.js
* Express.js
* VS Code IDE

Useful references:

* https://nodejs.org/
* https://www.mongodb.com/
* <https://reactjs.org/docs/create-a-new-react-app.html>

**USER & ADMIN FLOW:**

**1. User Flow:**

• Users start by registering for an account.

• After registration, they can log in with their credentials.

• Once logged in, they can check for the available products in the platform.

• Users can add the products they wish to their carts and order.

• They can then proceed by entering address and payment details.

• After ordering, they can check them in the profile section.

**2. Restaurant Flow:**

• Restaurants start by authenticating with their credentials.

• They need to get approval from the admin to start listing the products.

• They can add/edit the food items.

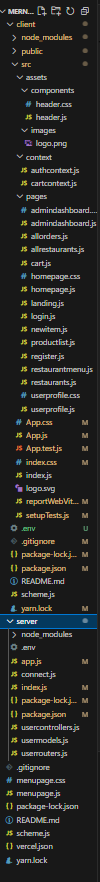
**3. Admin Flow:**

• Admins start by logging in with their credentials.

• Once logged in, they are directed to the Admin Dashboard.

• Admins can access the users list, products, orders, etc.

**PROJECT STRUCTURE**



This structure assumes a React app and follows a modular approach. Here's a brief explanation of the main directories and files:

• src/components: Contains components related to the application such as, register, login, home, etc.,

• src/pages has the files for all the pages in the application.

**PROJECT SETUP AND CONFIGURATION:**

**Install required tools and software:**

• Node.js.

Reference Article: <https://www.geeksforgeeks.org/installation-of-node-js-on-windows/>

• Git.

Reference Article: <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>

**Create project folders and files:**

• Client folders.

• Server folders

**DATABASE DEVELOPMENT:**

* **User**: username, email, password, usertype, approval
* **Admin**: categories, promotedRestaurants
* **Restaurant**: ownerId, title, address, menu[]
* **FoodItem**: title, description, itemImg, price, category, discount, rating
* **Orders**: userId, items[], paymentMethod, status
* **Cart**: userId, items[]

(Ref: scheme.js file)

**SETUP INSTRUCTIONS**

1. Clone the repo

git clone https://github.com/Niteesha/mernstack-project.git

1. Navigate into the project:

cd mernstack-project

1. Install dependencies:

* Server:

cd server

npm install

* Client:

cd ../client

npm install

1. Start backend server:

cd server

node index.js

1. Start frontend:

cd client

npm start

1. Connect MongoDB:

* Use .env file to place your connection string

**MODULE FLOW**

**USER MODULE:**

* Register/Login
* Browse items
* Add to cart
* Place orders
* View profile and order count

**ADMIN MODULE:**

* Login
* View all orders, users, restaurants
* Approve restaurants

**RESTAURANT MODULE:**

* Add/edit menu
* See orders

**COMPONENT STRUCTURE**

* **client/src/pages/**: productlist.js, cart.js, login.js, register.js, userprofile.js, admindashboard.js
* **client/src/components/**: header.js
* **client/src/context/**: cartcontext.js
* **server/**: index.js, scheme.js

**DEMO SCREENS**

* Home Page
* Menu Page with filters
* Cart Page (with UPI/Cash payment)
* Login/Register Page
* Admin Dashboard
* User Profile

**GITHUB REPO & DEPLOYMENT**

* GitHub: <https://github.com/TunuguntlaGouriNiteesha/MERNSTACK-final-project>
* Backend Deployed: https://food-0us6.onrender.com
* Frontend Vercel Setup: (see vercel.json)

**Conclusion:**  
The *OrderOnTheGo* food ordering application provides a seamless digital platform for users to browse, order, and enjoy food from a variety of categories. With features like user authentication, dynamic filtering, cart management, and admin control, it ensures a complete and efficient experience. The integration of MongoDB, Express, React, and Node.js (MERN stack) delivers robust performance and flexibility. Its user-friendly interface enhances accessibility for both customers and restaurant admins. Overall, this project modernizes food delivery by combining technology with convenience.

IMAGES:

