

# Full Stack Development with MERN

## Project Documentation

### 1. Introduction

- **Project Title:** Orderonthego: your on-demand food ordering solution
- **Team Members:**
  1. MANVITH SAI YARLAGADDA – FULL STACK DEVELOPER
  2. HARSHA DODDA – FRONTEND DEVELOPER
  3. PARDHASARADHI PUJALA – BACKEND DEVELOPER
  4. YUSUF VALI SHAIK – DATABASE ADMINISTRATOR

### 2. Project Overview

#### Purpose

Steamy Pot is a multi-role food ordering and delivery platform designed to provide a smooth and reliable food delivery experience similar to Swiggy and Zomato. The system supports users, restaurants, delivery drivers, and administrators, enabling seamless order placement, real-time tracking, payments, reviews, and administrative control.

#### Goals

- Deliver smooth and intuitive user experience
- Maintain reliable order lifecycle tracking
- Provide role-specific dashboards
- Ensure INR-first pricing model
- Enable scalable and secure backend architecture

#### Key Features

- Restaurant discovery and filtering
- Menu browsing and cart management
- Checkout with offers and discounts
- Payment state management (COD/Card/UPI states)
- Real-time order status timeline
- Driver assignment and tracking
- Reviews and restaurant replies
- Admin dashboard with search, filters, and metrics

### 3. Architecture

#### Technology Stack

MERN Stack with Socket.io

- MongoDB
- Express.js
- React.js
- Node.js
- Socket.io for real-time communication

REST APIs are used for core operations.

Socket connections are used for real-time notifications.

## Frontend Architecture (React + Vite)

- Routing via React Router
- Role-protected routes using ProtectedRoute
- Global state management using AuthContext (JWT handling)
- Toast notifications via ToastContext
- TailwindCSS for styling
- Axios client configured with /api base URL

### Pages include:

- User pages (restaurants list, menu, cart, checkout, orders, profile)
- Restaurant dashboard
- Driver dashboard
- Admin dashboard
- Shared landing and authentication pages

## Backend Architecture (Node.js + Express)

- RESTful API design
- Controllers organized by domain (auth, order, cart, admin, etc.)
- JWT authentication middleware
- Role-based access control
- Centralized error handling
- Async wrapper for controller functions
- Socket.io gateway with authenticated rooms

## Database Architecture (MongoDB + Mongoose)

### Collections include:

#### User

- username
- email
- role
- approval status
- blocked flag

#### FoodItem

- name
- description
- price
- imageUrl
- restaurantId

#### Cart

- userId
- restaurantId

- items (foodId, name, price, quantity)

### Order

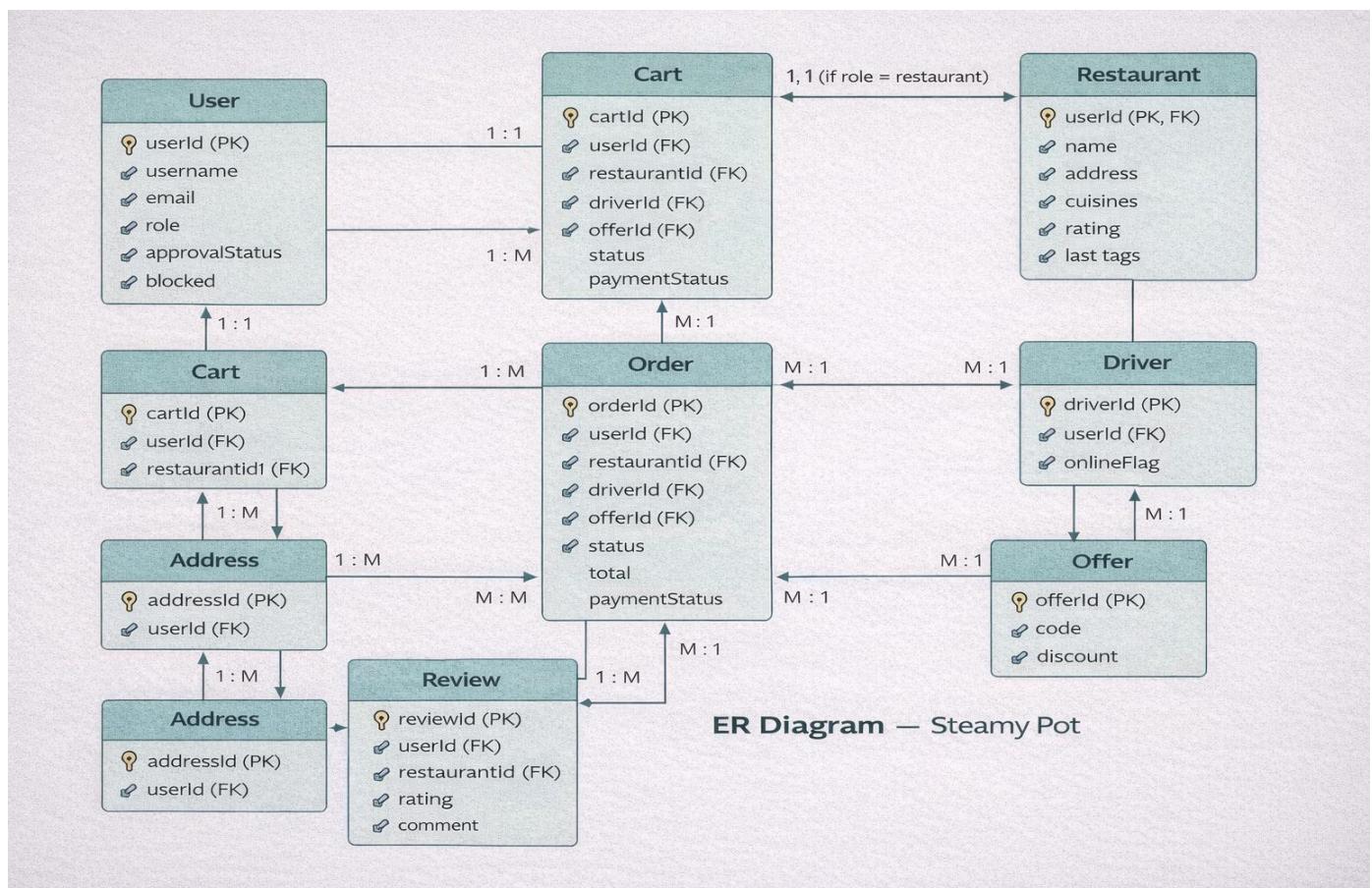
- userId
- restaurantId
- items
- subtotal
- discount
- deliveryFee
- total
- paymentMethod
- paymentStatus
- driverId
- status
- statusHistory

### Driver

- userId
- online flag
- currentOrderId

### Additional Collections:

- Address
- Offer
- Review



## 4. Setup Instructions

### Prerequisites

- Node.js (v18 or higher)
- MongoDB instance
- npm
- Git

### Installation Steps

#### Clone Repository

```
git clone <repo-url>
```

#### Navigate to project

```
cd FINAL YEAR/STEAMY POT
```

#### Server Setup

```
cd server  
npm install
```

#### Create .env file

```
MONGO_URI=<mongodb-connection-string>  
JWT_SECRET=<strong-secret>  
PORT=5000
```

#### Run backend

```
npm run dev  
or  
npm start
```

#### Client Setup

```
cd client  
npm install  
npm run dev
```

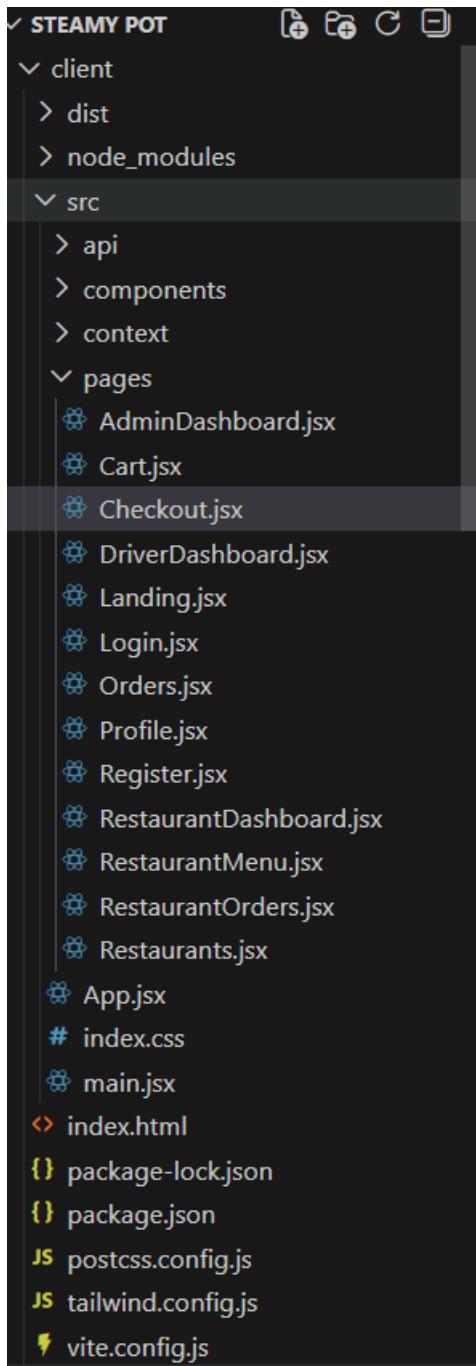
#### Frontend runs at:

<http://localhost:5173>

## 5. Folder Structure

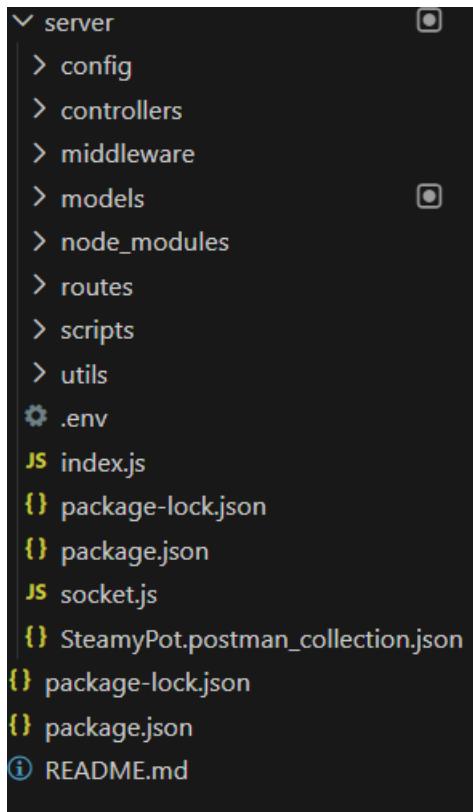
### client/

- React application
- Routing, contexts, pages, Tailwind config



## server/

- Express API
- Controllers
- Middleware
- Models
- Routes
- Socket.io setup



#### server/scripts/seed.js

- Optional seed data

## 6. Running the Application

### Frontend

From client folder

npm run dev

```
PS D:\FINAL YEAR\STEAMY POT\client> npm run dev

> sb-foods-client@0.1.0 dev
> vite

VITE v5.4.21 ready in 706 ms

→ Local: http://localhost:5173/
→ Network: use --host to expose
→ press h + enter to show help
```

## Backend

From server folder

npm start

```
PS D:\FINAL YEAR\STEAMY POT\server> npm start
> start
> node index.js

[dotenv@17.2.3] injecting env (2) from .env -- tip: ⚡ enable debug logging with { debug: true }
Connected to MongoDB
Server running on port 5000
[]
```

Ensure MongoDB is connected and environment variables are configured.

## 7. API Documentation

### Base Path: /api

Authentication uses JWT Bearer tokens.

Authorization Header Format:

Authorization: Bearer <token>

### Auth

POST /auth/register

POST /auth/login

### User

GET /user/profile

GET /user/orders

POST /user/addresses

PATCH /user/addresses/:id

DELETE /user/addresses/:id

### Cart

GET /cart

POST /cart/add

PATCH /cart/item

DELETE /cart/item/:foodId

### Orders

POST /orders

GET /orders

PATCH /orders/:id/status

### Restaurant

POST /restaurant/food

GET /restaurant/menu

PATCH /restaurant/food/:id

DELETE /restaurant/food/:id

### **Driver**

PATCH /driver/online  
GET /driver/orders  
PATCH /driver/orders/:id/status

### **Admin**

GET /admin/users  
GET /admin/orders  
PATCH /admin/users/:id/block  
GET /admin/metrics

### **Health**

GET /health  
Returns:  
{ status: "ok" }

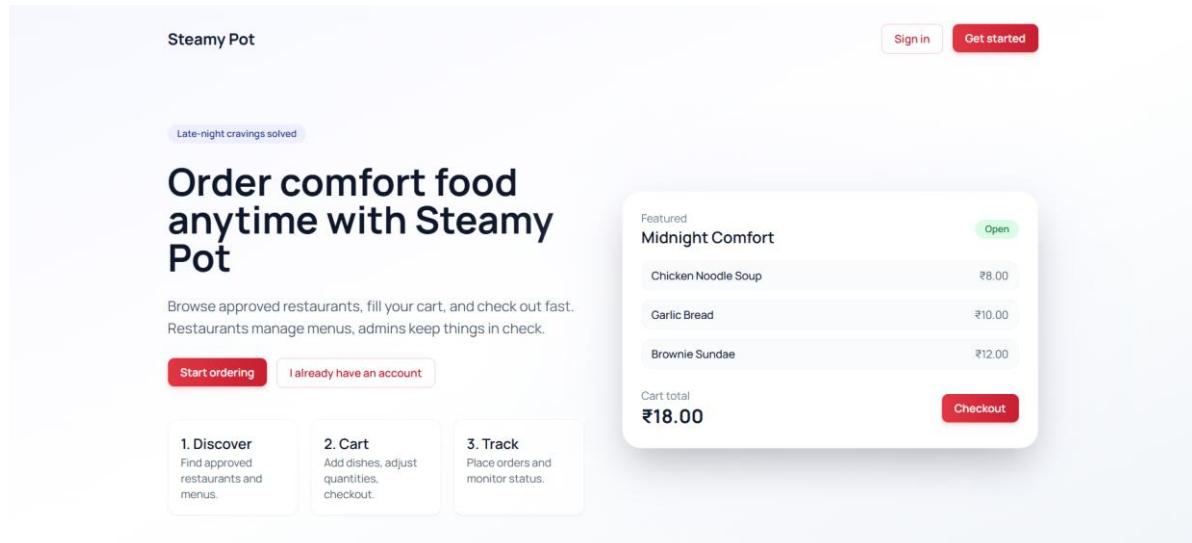
## **8. Authentication**

- JWT tokens issued during login
- Token stored client-side
- Middleware verifies token
- Role-based access control enforced
- Passwords hashed using bcrypt
- Token expiry: 7 days

## **9. User Interface**

### **Includes:**

- Landing page



## • Login and Registration

The image contains two screenshots of the Steamy Pot application's user interface. The top screenshot shows the 'Create account' form with fields for Username, Email, Password, and Role (set to User), followed by a 'Create account' button and a link to 'Sign in'. The bottom screenshot shows the 'Sign in' form with fields for Email (containing 'user@example.com') and Password (represented by dots), followed by a 'Sign in' button.

## • Restaurant listing

The screenshot shows the Steamy Pot restaurant listing page. At the top, there is a navigation bar with links for 'SteamyPot', 'Restaurants', 'Orders', 'Cart', and 'Profile'. To the right of the navigation is a 'User (user)' dropdown and a 'Logout' button. Below the navigation is a search section titled 'Restaurants' with fields for 'Search' (containing 'Restaurant or cuisine'), 'Cuisines (comma)' (containing 'Indian, Chinese'), and 'Tags' (containing 'Biryani, Pizza'). There is also a 'Sort by' dropdown set to 'Top rated' and an 'Open now' checkbox. A red 'Apply' button is located at the bottom of this section. Below the search section is a card for a restaurant. The card displays the restaurant's name, a 5.0 star rating with '2 ratings', a fee of '₹0.00', an ETA of '30 mins', and an 'Open' status. A red 'View menu' button is at the bottom of the card.

## • Menu listing

Screenshot of the SteamyPot app showing the menu listing for a restaurant. The top navigation bar includes 'SteamyPot' and links for 'Restaurants', 'Orders', 'Cart', and 'Profile'. A user status 'User (user)' and a 'Logout' button are also present.

The main section is titled 'Restaurant' with a green 'Open' button. It displays three menu items:

- Chicken Biriyani Full**: Flavoured rice with 3-piece chicken. Price: ₹320.00. Quantity: 1. Buttons for - and +. An 'Add to cart' button is visible.
- chicken curry**: juicy chicken curry (250 gms). Price: ₹150.00. An 'Add to cart' button is visible.
- CHICKEN 65**: juicy chicken 65 (250 gms). Price: ₹200.00. An 'Add to cart' button is visible.

## • Cart and checkout

Screenshot of the SteamyPot app showing the cart and checkout process. The top navigation bar includes 'SteamyPot' and links for 'Restaurants', 'Orders', 'Cart', and 'Profile'. A user status 'User (user)' and a 'Logout' button are also present.

The main section is titled 'Your Cart' with a red 'Checkout' button. It shows one item:

- Chicken Biriyani Full**: ₹320.00. Quantity: 1. Buttons for - and +. An 'Remove' button is visible.

The total amount is displayed as **Total: ₹320.00**.

## Checkout

Screenshot of the SteamyPot app showing the checkout process. The top navigation bar includes 'SteamyPot' and links for 'Restaurants', 'Orders', 'Cart', and 'Profile'. A user status 'User (user)' and a 'Logout' button are also present.

The main section is titled 'Delivery'.

Delivery options:

- Choose saved address: No saved addresses. Add one below.
- Or enter a new delivery address: Input field for 'House, street, city'.

Promo code: Input field for 'e.g. SAVE50' with an 'Apply' button.

Payment method: Select dropdown currently set to 'Cash on delivery'.

Summary table:

Subtotal	₹320.00
Delivery fee	₹0.00
Discount	-₹0.00
Total	₹320.00

At the bottom, it says **Pay: ₹320.00** and has a red 'Place order' button.

## • Order timeline view

SteamyPot Restaurants Orders Cart Profile

User (user) Logout

### Your Orders

Order #7e1cf3	ETA ~ 12:29	placed	Cancel
Progress			
● Placed	● Confirmed	● Preparing	● Out For Delivery
Chicken Biriyani Full x1			
Payment: COD pending			
Subtotal	₹320.00	Delivery	₹0.00
Discount	-₹0.00	Total	₹320.00
placed	20/02/2026, 11:54:16		
Order #b0acbc	delivered		
Driver: Driver			
Driver assignment: accepted			
Progress			
● Placed	● Confirmed	● Preparing	● Out For Delivery
Chicken Biriyani Full x1	₹320.00		
Payment: COD paid			

## • Restaurant dashboard

SteamyPot Restaurant Orders

Restaurant (restaurant) Logout

### Restaurant Dashboard

<b>Open status</b>	<b>Set to Closed</b>	
Control whether customers can place orders.		
<b>Performance</b>		
Total orders 0	Delivered 0	Cancelled 0
GMV ₹0.00	Delivery fees ₹0.00	Avg delivery (mins) 0.0
<b>Add menu item</b>		
Name <input type="text"/>	Description <input type="text"/>	Price <input type="text"/>
Image URL <input type="text"/>		<b>Add</b>

## Restaurant Orders

Order #7e1cf3		User: User
Status:	<a href="#">Accept</a>	<a href="#">Reject</a>
Driver:	Unassigned	Select driver <input type="button" value="▼"/>
Chicken Biriyani Full x1		₹320.00
Payment:	COD – pending	
Subtotal	₹320.00	Delivery ₹0.00
Discount	-₹0.00	Total ₹320.00
placed		20/02/2026, 11:54:16
Order #b0acbc		User: User
Driver acceptance: accepted		
Status:	<a href="#">preparing</a>	<a href="#">cancelled</a>
Driver:	Driver	Select driver <input type="button" value="▼"/>
Driver:	Driver	<a href="#">Assign</a>
Chicken Biriyani Full x1		₹320.00

## • Driver dashboard

SteamyPot	Driver	Driver (driver)	<a href="#">Logout</a>
<h3>Driver Dashboard</h3> <p>Manage your availability and assigned orders.</p>			
<a href="#">Go offline</a>			
Order #7e1cf3		preparing	
Driver acceptance: pending			
<a href="#">placed</a>	<a href="#">confirmed</a>	<a href="#">preparing</a>	<a href="#">out for delivery</a>
<a href="#">delivered</a>			
Customer			
User			
user@example.com			
Chicken Biriyani Full x1			₹320.00
<a href="#">Accept</a>	<a href="#">Decline</a>		
placed			20/02/2026, 11:54:16
confirmed			20/02/2026, 11:56:07
preparing			20/02/2026, 11:56:10
preparing			20/02/2026, 11:56:48
Order #b0acbc		delivered	
Driver acceptance: accepted			
<a href="#">placed</a>	<a href="#">confirmed</a>	<a href="#">preparing</a>	<a href="#">out for delivery</a>
<a href="#">delivered</a>			
Customer			
User			
Restaurant			
Restaurant			
Restaurant@gmail.com			

- Admin analytics dashboard

The screenshot shows the Admin Dashboard interface. At the top, there's a header with 'SteamyPot Admin' on the left and 'Admin (admin)' with a 'Logout' button on the right. Below the header, the title 'Admin Dashboard' is displayed. Underneath, a section titled 'Platform metrics' contains six boxes with the following data:

Total orders 6	Delivered 4	Cancelled 0	GMV ₹1690.00
Discounts ₹0.00	Delivery fees ₹0.00	Avg delivery (mins) 0.1	

Below the metrics, a section titled 'Top restaurants (by revenue)' lists two entries:

Restaurant 228x1a42h3	₹1190.00 (4 orders)
	₹500.00 (2 orders)

On the left side, there's a sidebar titled 'Restaurants' which shows a single entry for 'Restaurant'. On the right side, there's a 'Users' section with filters for 'Search user', 'All roles', 'Blocked?', and 'Approval'.

## 10. Testing

### Manual Testing:

- Postman collection used

### Future Testing:

- Jest and Supertest for API testing
- React Testing Library for UI

## 11. Screenshots or Demo

Video Link : [\(LINK\)](#)

The screenshot shows the Steamy Pot food delivery app. At the top, there's a header with 'Steamy Pot' on the left and 'Sign in' and 'Get started' buttons on the right. Below the header, a banner says 'Late-night cravings solved.' and features the tagline 'Order comfort food anytime with Steamy Pot'. It encourages users to 'Browse approved restaurants, fill your cart, and check out fast. Restaurants manage menus, admins keep things in check.' There are 'Start ordering' and 'I already have an account' buttons. At the bottom, there are three numbered steps: '1. Discover', '2. Cart', and '3. Track'.

To the right, a detailed view of the 'Cart' screen is shown. It displays a list of items from a 'Featured' menu called 'Midnight Comfort':

Chicken Noodle Soup	₹8.00
Garlic Bread	₹10.00
Brownie Sundae	₹12.00

The total cart amount is ₹18.00, and there's a 'Checkout' button at the bottom right.

## Restaurants

Search

Cuisines (comma)

Tags

Sort by

Top rated

Open now

Apply

**Restaurant**

5.0 ★  
2 ratings

Fee: ₹0.00      ETA: 30 mins      Open

View menu

## Restaurant

ETA 30 mins Fee ₹0.00 5.0 ★ (2)

Open

 <b>Chicken Biryani Full</b> Flavoured rice with 3-piece chicken	₹320.00 <span style="border: 1px solid #ccc; border-radius: 50%; padding: 2px 5px;">-</span> <span style="border: 1px solid #ccc; border-radius: 50%; padding: 2px 5px;">1</span> <span style="border: 1px solid #ccc; border-radius: 50%; padding: 2px 5px;">+</span>	 <b>chicken curry</b> juicy chicken curry (250 gms)	₹150.00 <span style="border: 1px solid red; border-radius: 5px; padding: 2px 10px;">Add to cart</span>
 <b>CHICKEN 65</b> juicy chicken 65 (250 gms)	₹200.00 <span style="border: 1px solid red; border-radius: 5px; padding: 2px 10px;">Add to cart</span>		

## Your Orders

Order #7e1cf3

ETA ~ 12:29

placedCancel

## Progress

 Placed —————  Confirmed —————  Preparing —————  Out For Delivery —————  Delivered
 

Chicken Biryani Full x1

₹320.00

Payment: COD pending

Subtotal	₹320.00	Delivery	₹0.00
Discount	-₹0.00	Total	₹320.00

placed

20/02/2026, 11:54:16

Order #b0acbc

delivered

Driver: Driver

Driver assignment: accepted

## Progress

 Placed —————  Confirmed —————  Preparing —————  Out For Delivery —————  Delivered
 

Chicken Biryani Full x1

₹320.00

Payment: COD paid

## Restaurant Dashboard

### Open status

Control whether customers can place orders.

[Set to Closed](#)

### Performance

Total orders  
0

Delivered  
0

Cancelled  
0

GMV  
₹0.00

Delivery fees  
₹0.00

Avg delivery (mins)  
0.0

### Add menu item

Name

Description

Price

Image URL

<https://...>

[Add](#)

## Driver Dashboard

Manage your availability and assigned orders.

[Go offline](#)

Order #7e1cf3

Driver acceptance: pending

placed confirmed preparing out for delivery delivered

Customer  
User  
[user@example.com](mailto:user@example.com)

Restaurant  
Restaurant  
[Restaurant@gmail.com](mailto:Restaurant@gmail.com)

Chicken Biriyani Full x1

₹320.00

[Accept](#) [Decline](#)

placed  
confirmed  
preparing  
preparing

20/02/2026, 11:54:16  
20/02/2026, 11:56:07  
20/02/2026, 11:56:10  
20/02/2026, 11:56:48

Order #b0acbc

delivered

Driver acceptance: accepted

placed confirmed preparing out for delivery delivered

Customer  
User

Restaurant  
Restaurant

## Admin Dashboard

### Platform metrics

Total orders  
6

Delivered  
4

Cancelled  
0

GMV  
₹1690.00

Discounts  
₹0.00

Delivery fees  
₹0.00

Avg delivery (mins)  
0.1

### Top restaurants (by revenue)

Restaurant  
228x1a42h3

₹1190.00 (4 orders)  
₹500.00 (2 orders)

### Restaurants

Restaurant  
[Restaurant@gmail.com](mailto:Restaurant@gmail.com)  
Approved: Yes

### Users

Search user

All roles

Blocked?

Approval

## **12. Known Issues**

- Real-time updates need full socket integration
- Payment flow uses mock state transitions
- Requires MongoDB running locally

## **13. Future Enhancements**

- Real-time ETA calculations
- Full payment gateway integration
- Driver and restaurant verification
- Advanced analytics dashboards
- Mobile app or PWA support