PROF. V. B. SHAH INSTITUTE OF MANAGEMENT,
R. V. PATEL COLLEGE OF COMMERCE (ENG. MED.),
V. L. SHAH COLLEGE OF COMMERCE (GUJ. MED.) &
SUTEX BANK COLLEGE OF COMPUTER APPLICATIONS & SCIENCE
Managed by Jivan Jyot Trust, Amroli.

All Colleges are Affiliated to Veer Narmad South Gujarat University, Surat.
Self Financed B.B.A. - B.Com. (Eng. & Guj. Med.) - B.C.A & B.Sc. Data Science Degree Programme
Accredited by National Assessment and Accreditation Council with "B" Grade

Prin. Dr. Mukesh R. Goyani
M.Com., M.Ed., M.Phil., GSET, Ph.D.
Senate Member, VNSGU,Surat

At & Po. : Amroli, Surat,
Pin. 394107, Gujarat (India)
www.amrolicollege.ac.in

☎ (0261) 2494073, 2495643, 2496478
9427113947, 6354626492
principal@amrolicollege.ac.in
Principal_230@vnsgu.ac.in

Prof.V.B.Shah Institute of Management & R.V.Patel College of Commerce
(Eng.Med.) & V.L.Shah College of Commerce (Guj.Med.) &

Sutex Bank College of Computer Applications & Science, Amroli

(Accredited ‘B’ (CGPA 2.55) by NAAC Dec.-2009)

VEER NARMAD SOUTH GUJARAT UNIVERSITY

PROJECT REPORT

ON

GRAND RESTAURANT

Submitted By:

Chodavadiya Tiyansi Bharatbhai

Koladiya Jensi Hareshbhai

Guided By:

Ashish Solanki

Acknowledgement :

The successful completion of any project is an end result of the successful and coherent combination of ideas and efforts of the involved members. The same is true for this project. The reason for the successful completion of this project is unquestionably help and support achieved from my faculties and friends who guided me, advised me and finally made me capable to transform the ideas into successful work.

We are very grateful to **Prof. Ashish solanki** who helped us in the successful completion of this project.

We are highly thankful to them who sincerely provided us the necessary information and valuable suggestions and time for completing this project in the best possible way we also thank all our friends and family for being a wonderful support.

INDEX

1. Introduction

- Overview of the Project
- Objectives of the System

- Project Title : Grand Restaurant

1.1 Overview of the Project :

The Grand Restaurant Food Ordering System is a web application built with ReactJS frontends and Node.js/Express backend with MongoDB database, connecting customers with restaurant operations through separate user and admin portals for food ordering and management.

The system provides customers with food browsing, cart management, and order tracking, while administrators manage food inventory and orders through a dedicated dashboard, featuring secure authentication, payment integration, and real-time order processing capabilities.

1.2 Objectives of the System :

- Simplify the food ordering process through an online platform
- Provide complete transparency in food menu and pricing
- Enable secure user registration and login
- Allow admin to manage food items, orders, and customers
- Maintain proper records of food inventory, orders, and customer history
- Enhance communication between customers and restaurant management
- Promote digital dining experience and encourage convenient food ordering

2. System Review

- Hardware/Software Specification
- Feature of Proposed System
- Directory Structure

2.1 Hardware / Software Specification:

Project Title :	Grand Restaurant
Front-End :	React-Js
Back-End :	MongoDB
Browser :	All Browser Supported
Technology :	HTML,CSS,JavaScript
Platform :	VS code,Mongodb Atlas
Documentation tool :	MS Word
Guided By :	Prof. Ashish Solanki
Developed By:	**Chodavadiya Tiyansi** **Koladiya Jensi**

2.2 Feature of Proposed System:

➢ User Feature:

- Users can browse and view all available food items with detailed information such as name, description, price, and category
- They can add food items to cart and place orders
- Users can register and log in securely to access the system
- They can explore the home page, see featured food items, and view the menu
- Users can track their order status and view order history
- They can manage their profile and contact restaurant support

➢ Admin Feature:

- Admin can log in securely to access the system
- Admin can add new food items with details, update existing records, or delete items
- Admin can view and manage all registered users and their orders
- Admin can view all orders submitted by users
- Admin has the ability to update order status (Food Processing, Out for Delivery, Delivered)
- Admin can maintain order history and ensure the system is updated

2.3 Directory Structure:

```
Grand-restaurant/
├── admin/
│   ├── node_modules/
│   ├── public/
│   ├── src/
│   │   ├── assets/
│   │   ├── components/
│   │   │   ├── Navbar/
│   │   │   ├── Sidebar/
│   │   ├── pages/
│   │   │   ├── Add/
│   │   │   ├── List/
│   │   │   ├── Orders/
│   │   ├── App.jsx
│   │   ├── index.css
│   │   └── main.jsx
│   ├── index.html
│   ├── package-lock.json
│   ├── package.json
│   └── vite.config.js
├── backend/
```

```
│   ├── config/
│   │   └── db.js
│   ├── controllers/
│   ├── middleware/
│   ├── models/
│   │   ├── foodModel.js
│   │   ├── orderModel.js
│   │   └── userModel.js
│   ├── routes/
│   │   ├── foodroute.js
│   │   ├── userRoute.js
│   │   ├── cartroute.js
│   │   └── orderRoute.js
│   ├── uploads/
│   ├── .env
│   ├── index.js
│   ├── package-lock.json
│   ├── package.json
│   └── server.js
├── frontend/
│   ├── node_modules/
│   ├── public/
│   ├── src/
│   │   ├── assets/
│   │   ├── components/
│   │   │   ├── Navbar/
│   │   │   ├── Footer/
│   │   │   ├── LoginPopup/
│   │   ├── context/
│   │   ├── pages/
│   │   │   ├── Cart/
│   │   │   ├── Home/
│   │   │   ├── PlaceOrder/
│   │   │   ├── Verify/
│   │   │   └── MyOrders/
│   │   ├── App.jsx
│   │   ├── index.css
│   │   └── main.jsx
│   ├── index.html
│   ├── package-lock.json
│   ├── package.json
│   └── vite.config.js
└── .gitignore
```

3.Database Design

- Database Tables
- ER Diagram
- DFD
- Data Dictionary
- Process Flow Diagram

3.1 Database Tables:

1. UserTable (userModel.js)

Represents all registered users in the system.

- **Fields:**

- **name** → String, required
- **email** → String, required, unique
- **password** → String, required (hashed using bcrypt)
- **cartData** → Object, default: {}

Attribute	**Type**	**Constraints / Notes**
userId	ObjectId (MongoDB)	Primary Key, auto-generated
name	String	Required
email	String	Required, Unique
password	String	Required, stored hashed (bcrypt)
cartData	Object	Key-value structure for user cart; default {}
timestamps	Date (createdAt, updatedAt)	Auto-generated

userId	ObjectId (MongoDB)	Primary Key, auto-generated

2. FoodTable (foodModel.js)

Represents each food item stored in the system.

➢ **Fields:**

- **name** → String, required
- **description** → String, required
- **price** → Number, required
- **image** → String, required
- **category** → String, required

Attribute	Type	Constraints / Notes
foodId	ObjectId (MongoDB)	Primary Key, auto-generated
name	String	Required
description	String	Required
price	Number	Required
image	String	Required (image URL or file path)
category	String	Required (e.g., veg, non-veg, beverages, etc.)
timestamps	Date (createdAt, updatedAt)	Auto-generated

3. OrderTable (orderModel.js)

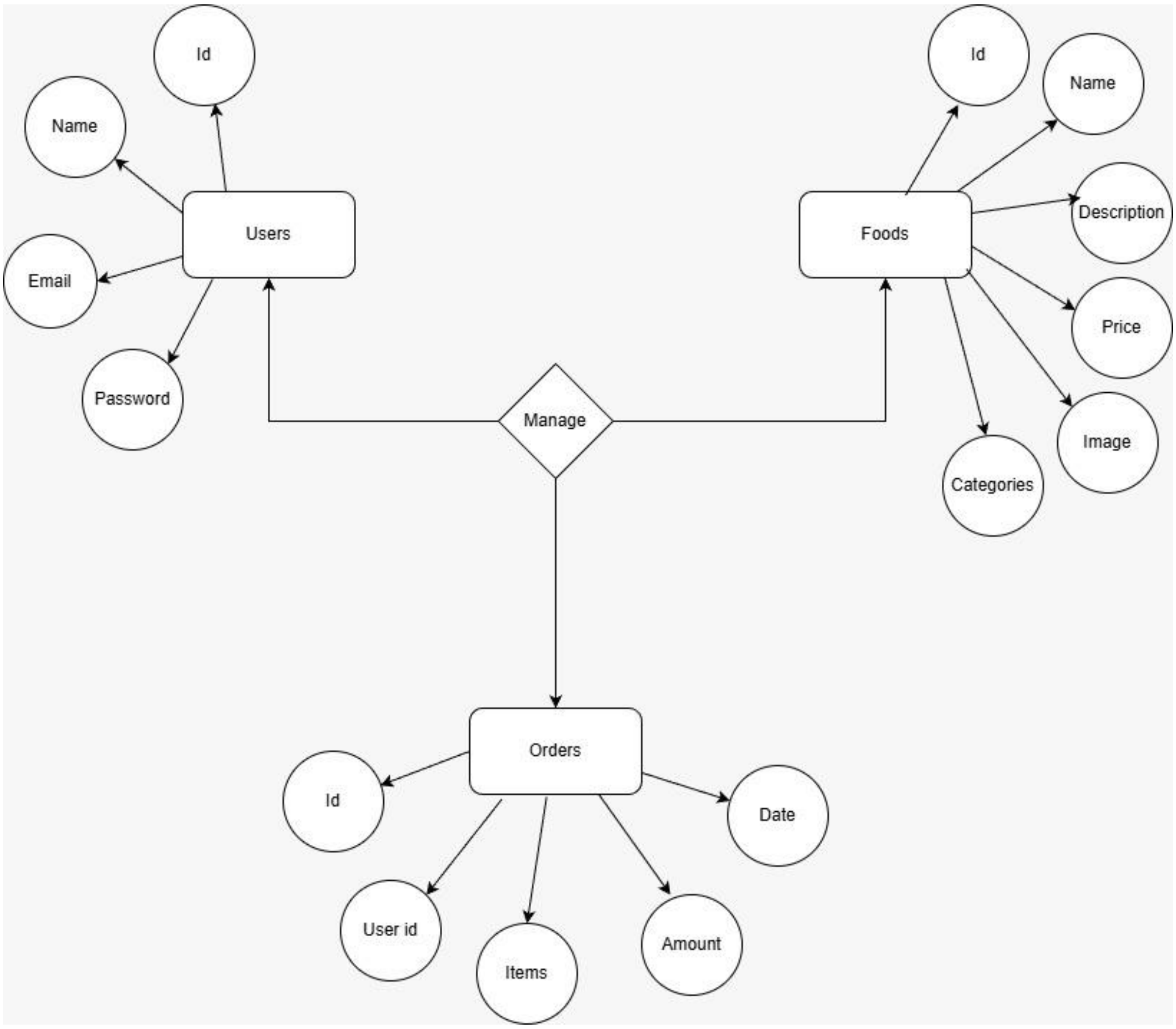
Represents orders placed by a user.

➢ **Fields:**

- **userId** → ObjectId, required (reference to User)
- **items** → Array, required
- **amount** → Number, required
- **address** → Object, required
- **status** → String, default: "Food Processing"
- **date** → Date, default: Date.now()
- **payment** → Boolean, default: false

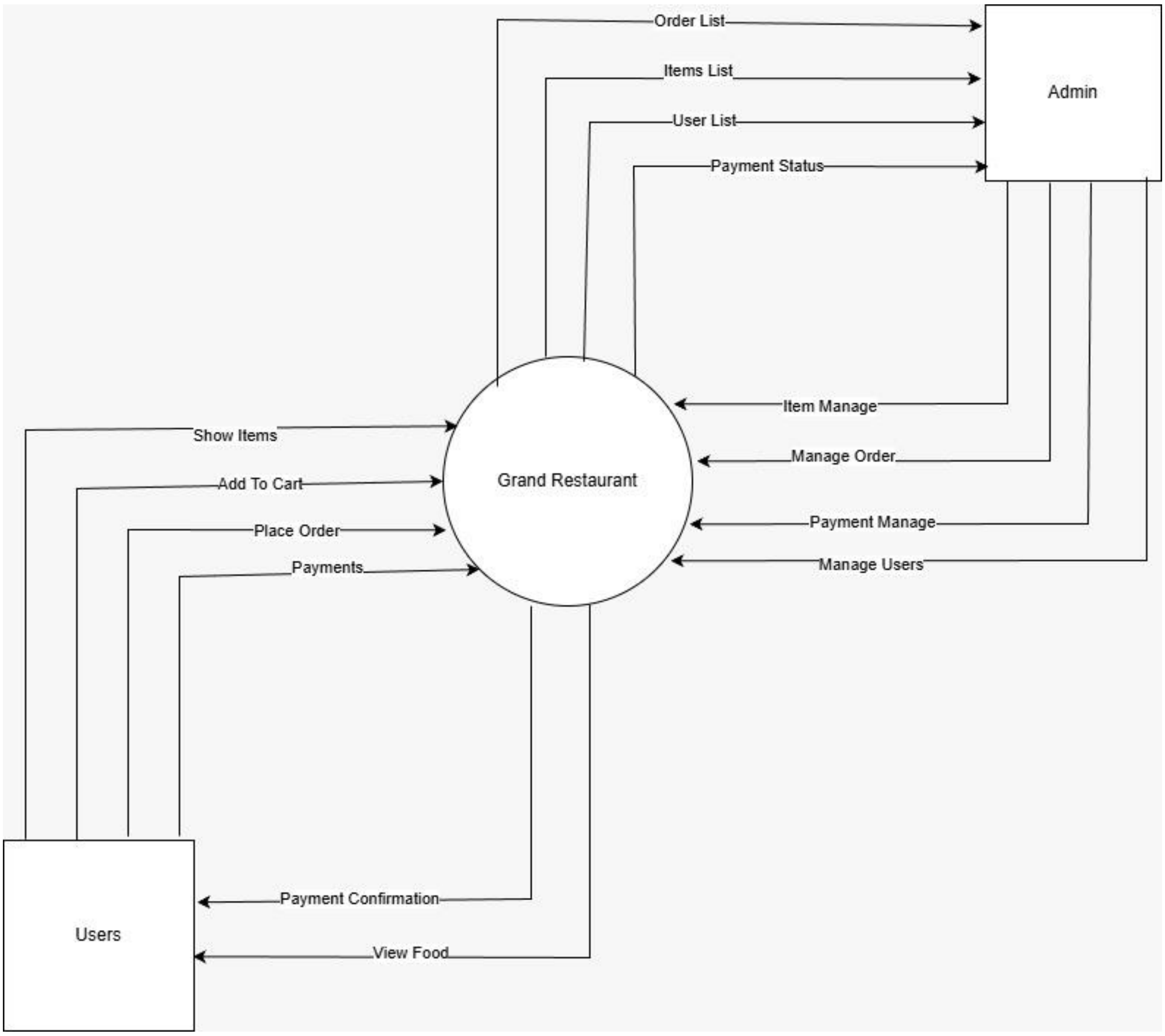
Attribute	Type	Constraints / Notes
orderId	ObjectId (MongoDB)	Primary Key, auto-generated
userId	ObjectId (User Ref)	Required; links order to the user
items	Array	Required; contains food items with quantities
amount	Number	Required; total bill amount
address	Object	Required; delivery address details
status	String	Default: "Food Processing" (can later be delivered/cancelled)
date	Date	Default: current timestamp
payment	Boolean	Default: false (paid/unpaid)
timestamps	Date (createdAt, updatedAt)	Auto-generated

3.2 ER Diagram:

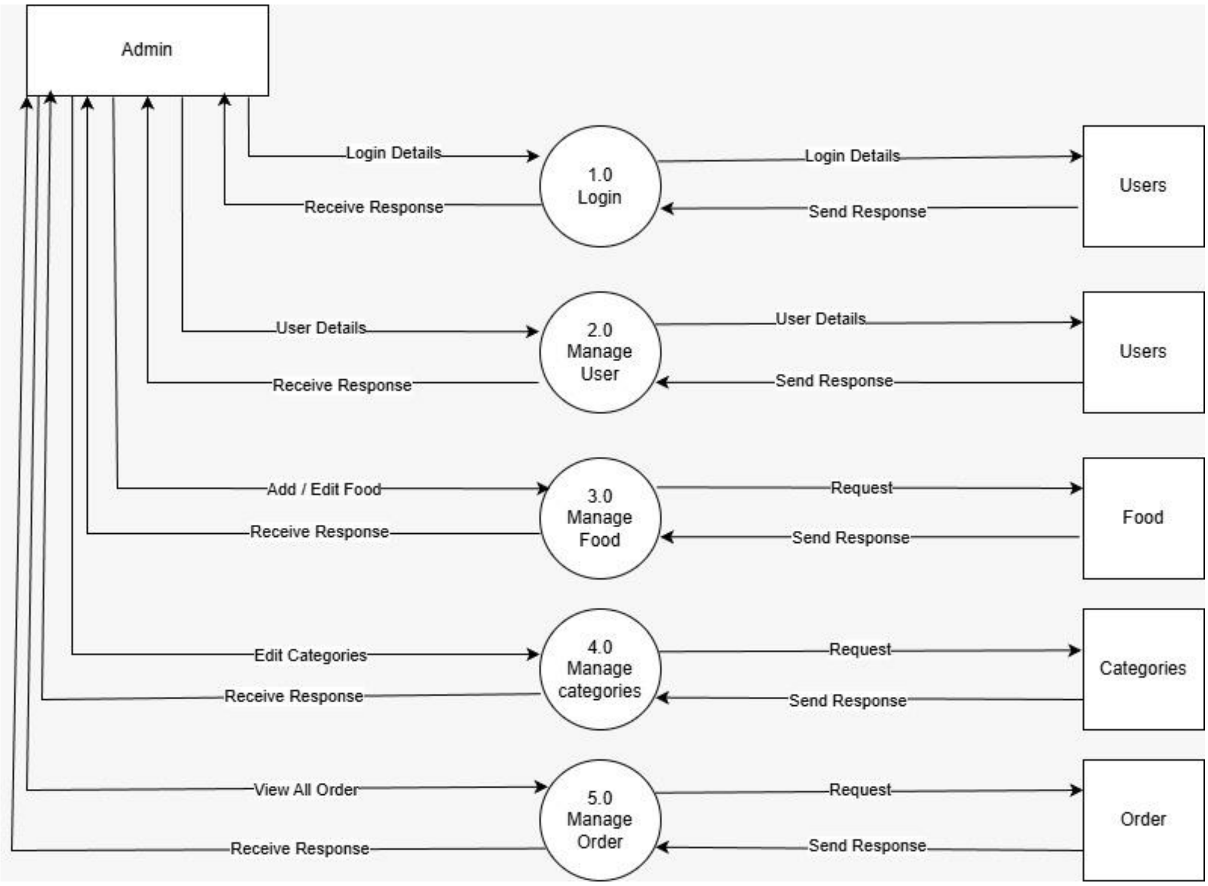


3.3 DFD:

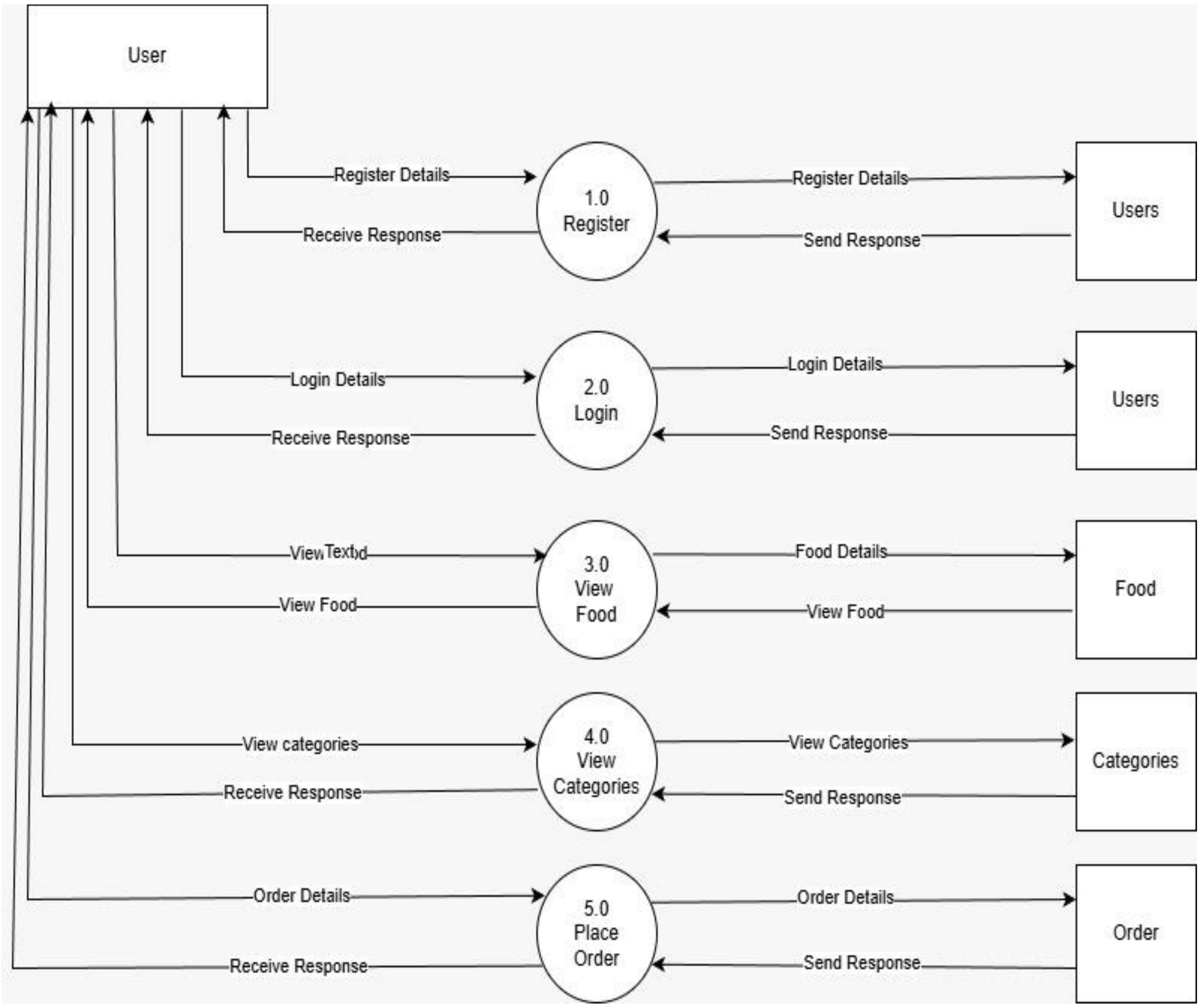
- **0 level Diagram:**



- **1st level Diagram(admin):**



- **1st level Diagram(user):**



3.4 Data Dictionary:

1. User Model

No	Field	Datatype	Size	Description / Constraint
1	_id	ObjectId	24 bytes	Primary key, auto-generated
2	name	String	Variable	User full name (required)
3	email	String	Up to 255 chars	User email (unique, required)
4	password	String	Variable	Hashed password (required)
5	cartData	Object	Variable	User cart data (default: {})

2. Food Model

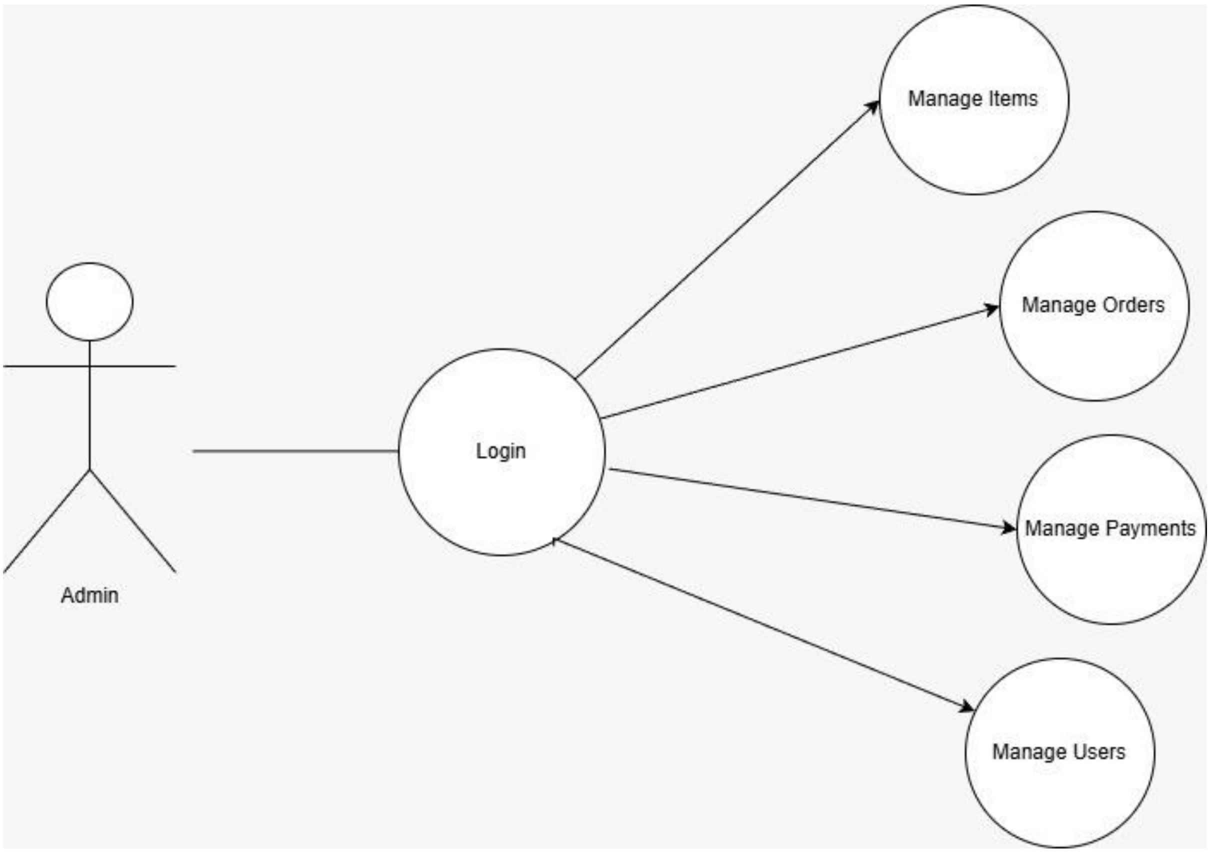
No	Field	Datatype	Size	Description / Constraint
1	_id	ObjectId	24 bytes	Primary key, auto-generated
2	name	String	Variable	Food name (required)
3	description	String	Variable	Food description (required)
4	price	Number	8 bytes	Food price (required)
5	image	String	Variable	Food image URL (required)
6	category	String	Variable	Food category (required)

3. Order Model

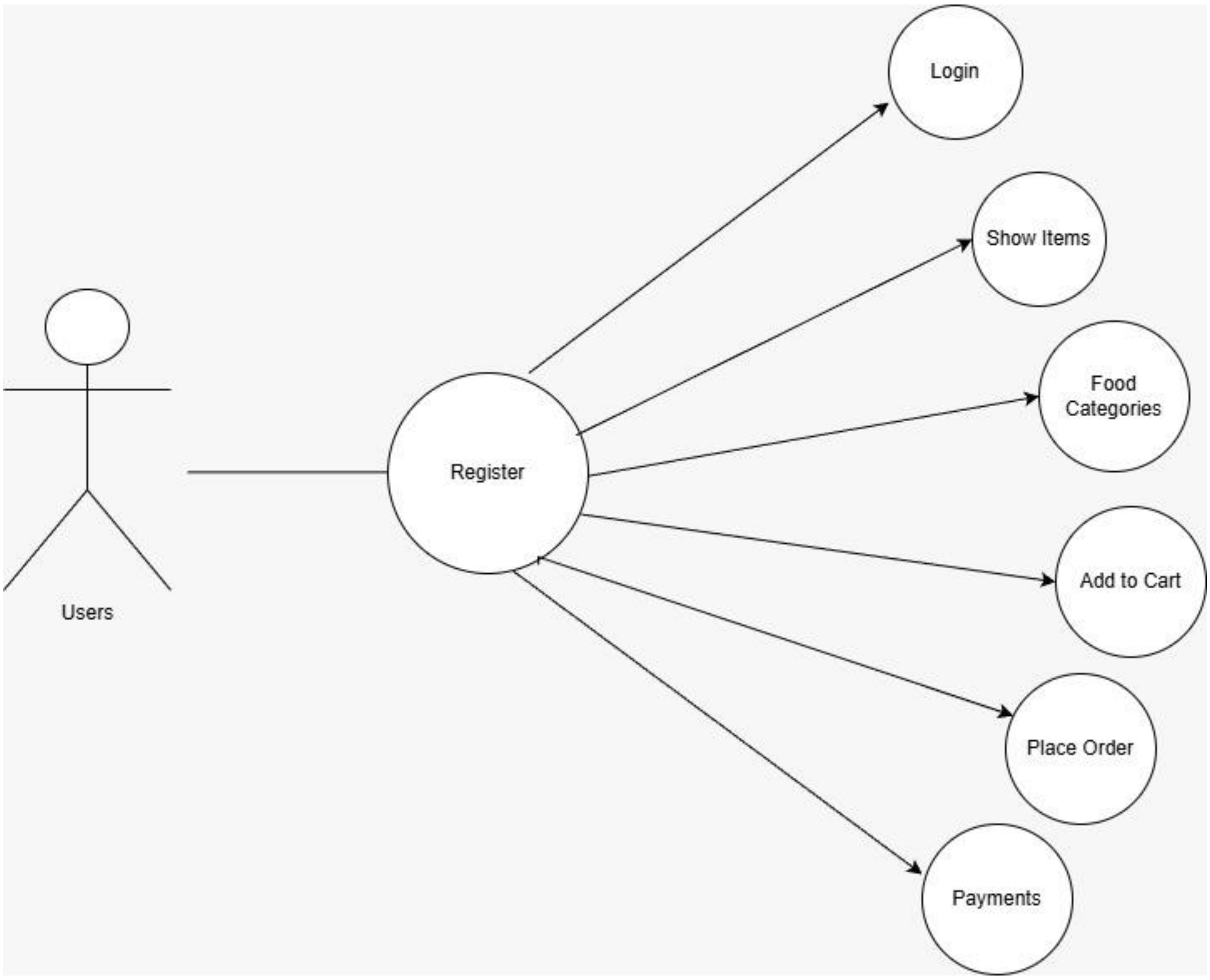
No	Field	Datatype	Size	Description / Constraint
1	_id	ObjectId	24 bytes	Primary key, auto-generated
2	userId	ObjectId	24 bytes	Reference to User (required)
3	items	Array	Variable	List of ordered items
4	amount	Number	8 bytes	Total order amount
5	address	Object	Variable	Delivery address (required)
6	status	String	Variable	Default: 'Food Processing'
7	date	Date	8 bytes	Order creation date (auto)
8	payment	Boolean	1 byte	Payment status (default: false)

3.5 Process Flow Diagram:

- **Admin Flow:**



- **User Flow:**



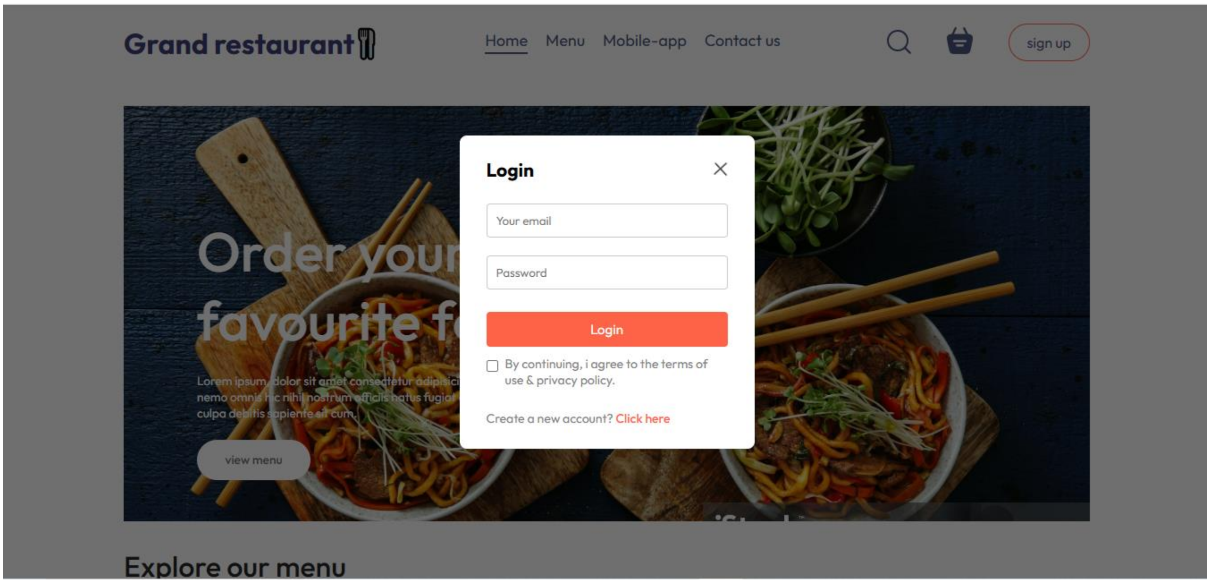
4.System Design

- ➢ Component wise Frontend Layouts
- ➢ Report Views

4.1 Component wise Frontend Layouts:

➢ User Side

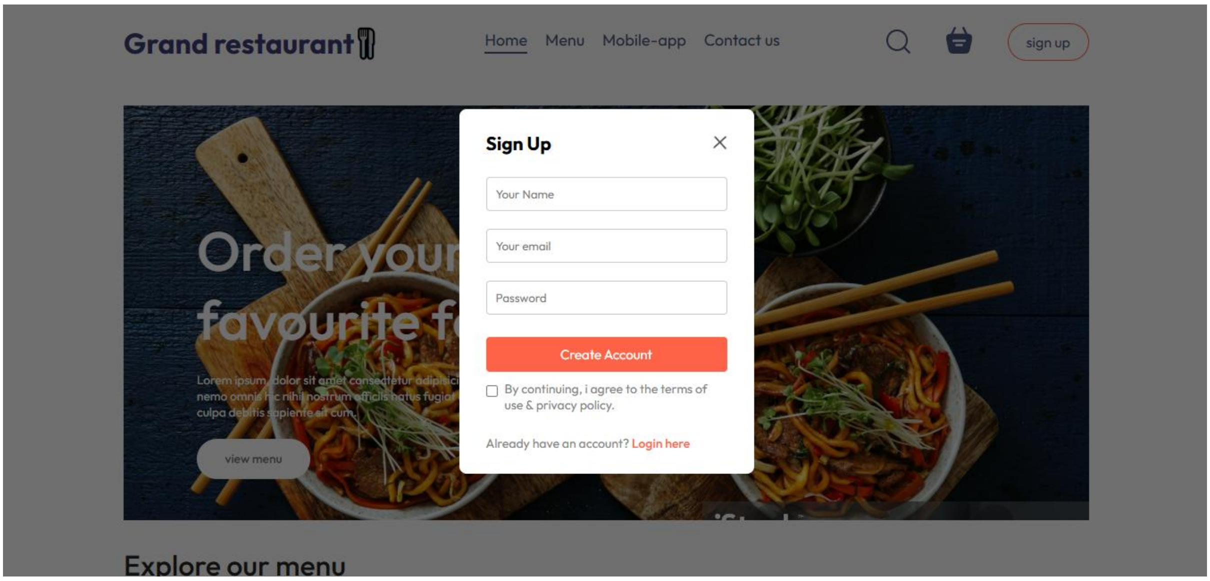
- User Login Screen



□ By continuing, i agree to the terms of use & privacy policy.

Create a new account? Click here

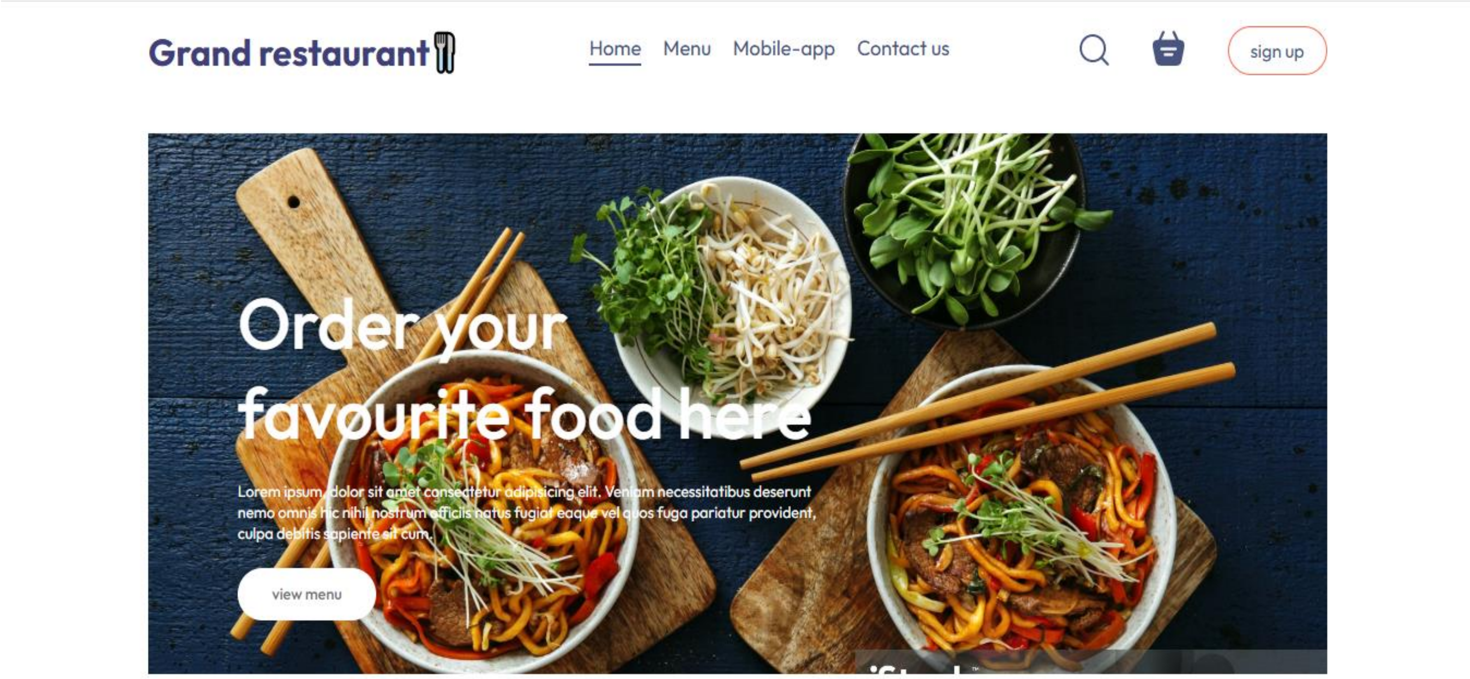
- User Register Screen



□ By continuing, i agree to the terms of use & privacy policy.

Already have an account? Login here

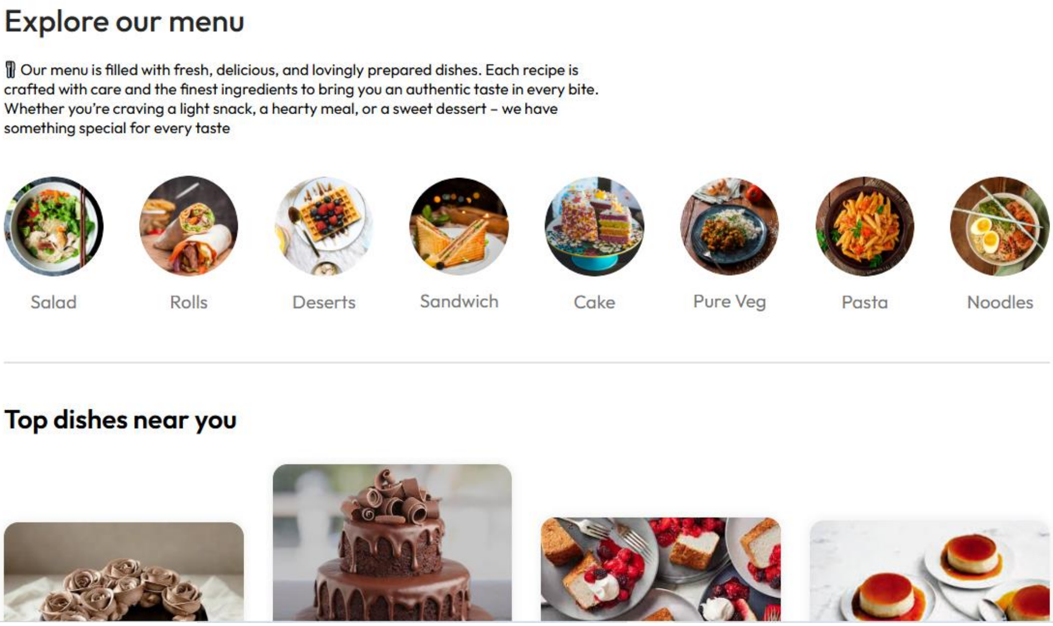
- Home screen



Order your favourite food here

Lorem ipsum dolor sit amet consectetur adipisicing elit. Veniam necessitatibus deserunt nemo omnis hic nihil nostrum officiis natus fugiat eaque vel quos fuga pariatur provident, culpa debitis sapiente sit cum.

- Menu page

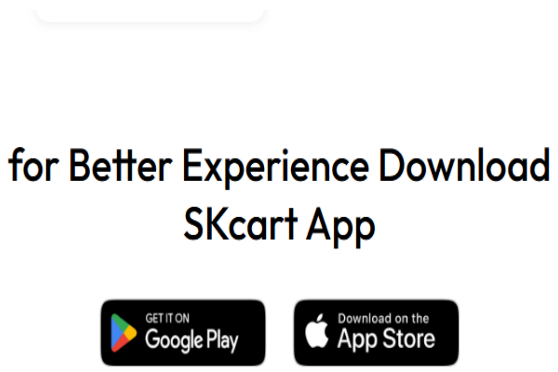


Explore our menu

🍴 Our menu is filled with fresh, delicious, and lovingly prepared dishes. Each recipe is crafted with care and the finest ingredients to bring you an authentic taste in every bite. Whether you're craving a light snack, a hearty meal, or a sweet dessert – we have something special for every taste

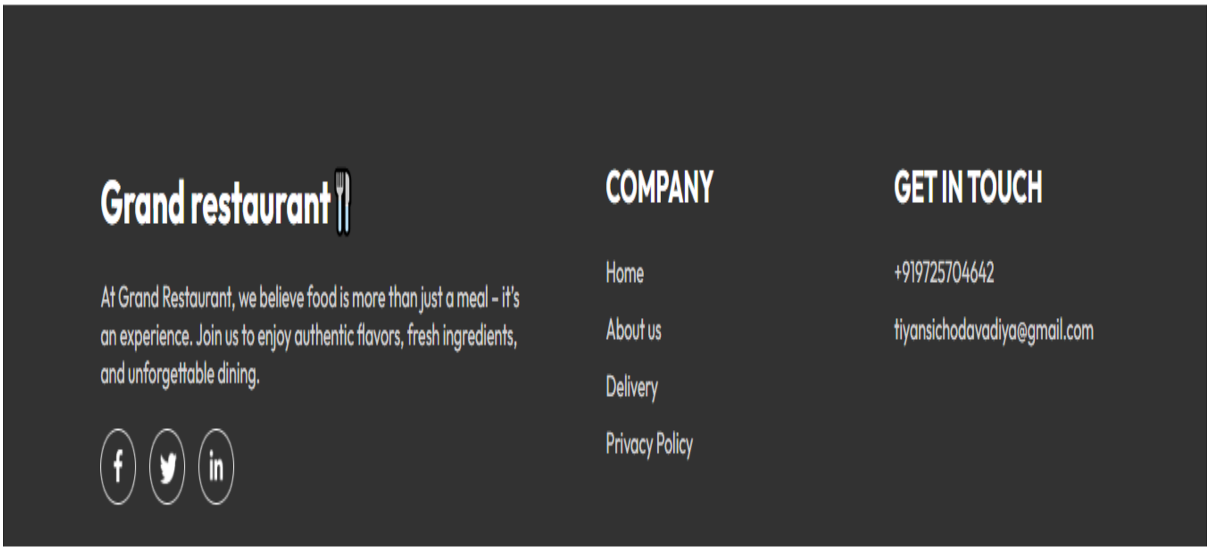
Top dishes near you

- Mobile –app page



for Better Experience Download SKcart App

- Contact Us page



Grand restaurant

At Grand Restaurant, we believe food is more than just a meal - it's an experience. Join us to enjoy authentic flavors, fresh ingredients, and unforgettable dining.

COMPANY

Home

About us

Delivery

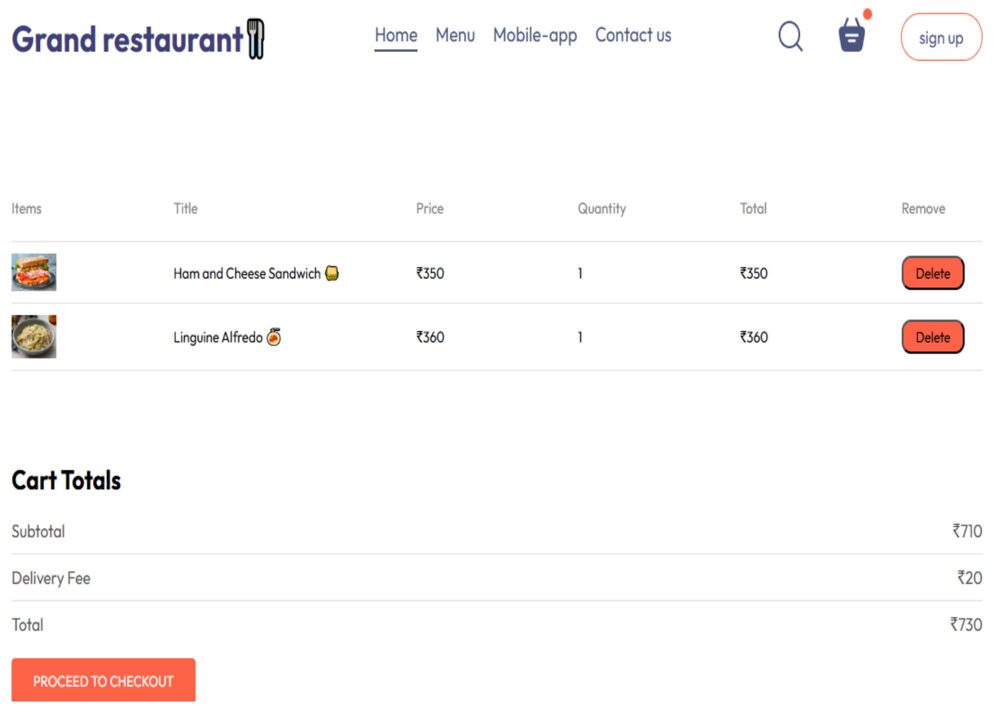
Privacy Policy

GET IN TOUCH

+919725704642

tiyansichodavadiya@gmail.com

- Add to cart

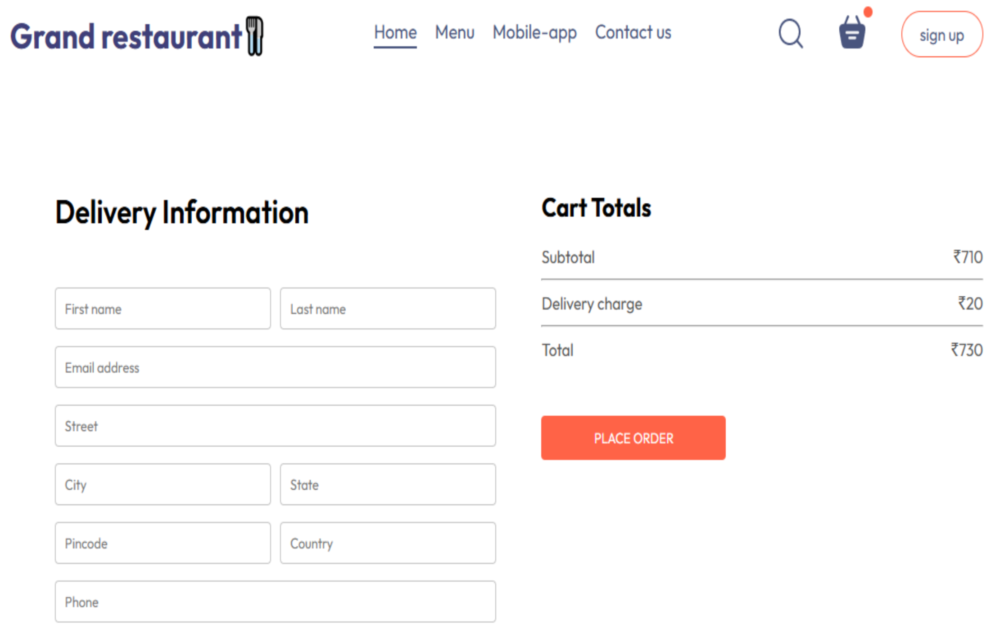


Items	Title	Price	Quantity	Total	Remove
	Ham and Cheese Sandwich	₹350	1	₹350	Delete
	Linguine Alfredo	₹360	1	₹360	Delete

Cart Totals

Subtotal	₹710
Delivery Fee	₹20
Total	₹730

- Paytment



Delivery Information

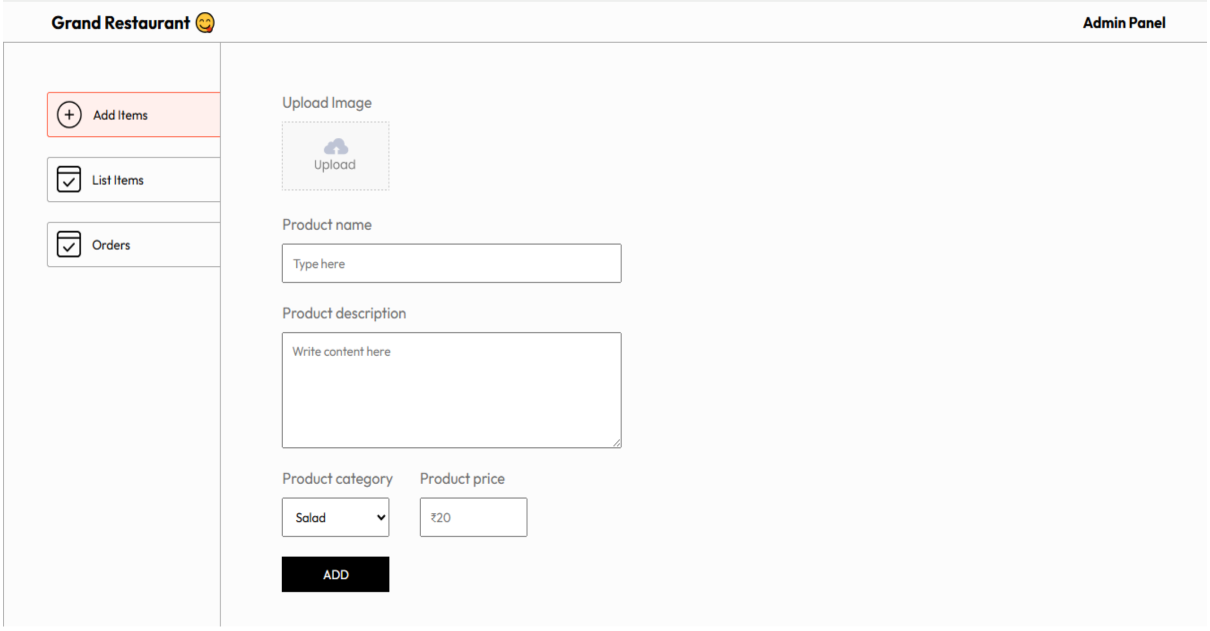
First name
Last name
Email address
Street
City
State
Pincode
Country
Phone

Cart Totals

Subtotal	₹710
Delivery charge	₹20
Total	₹730

➢ Admin Side

- Add items



Upload Image

Product name

Type here

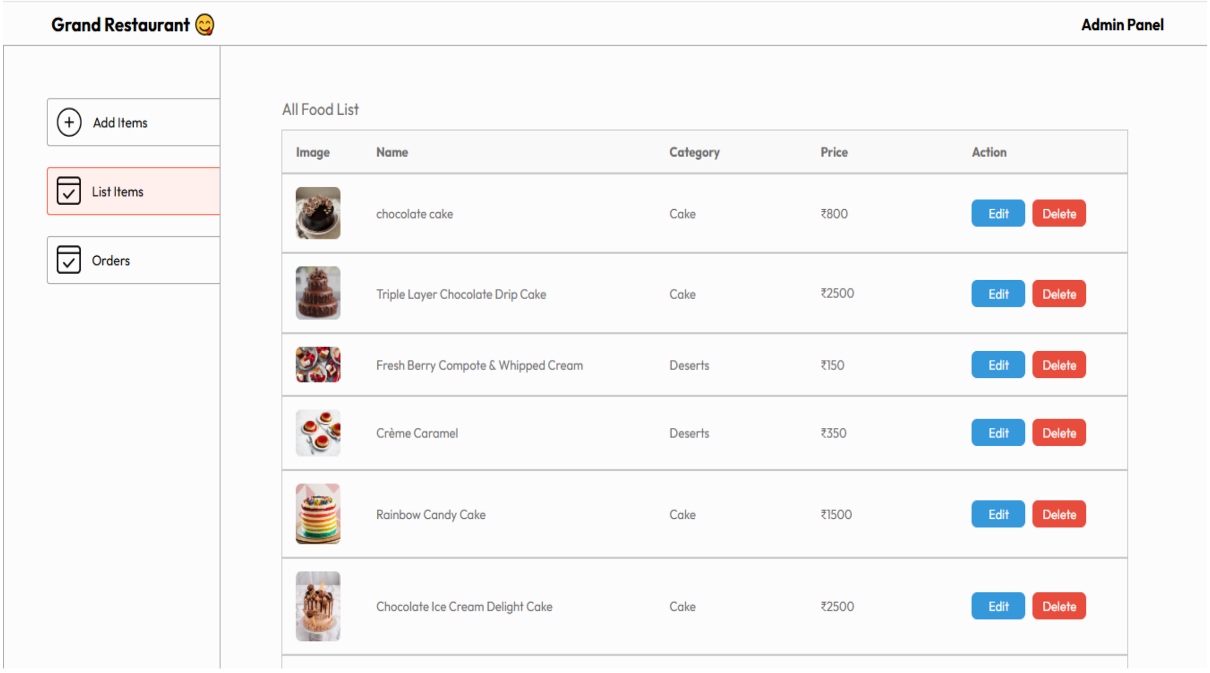
Product description

Write content here

Product category: Salad

Product price: ₹20

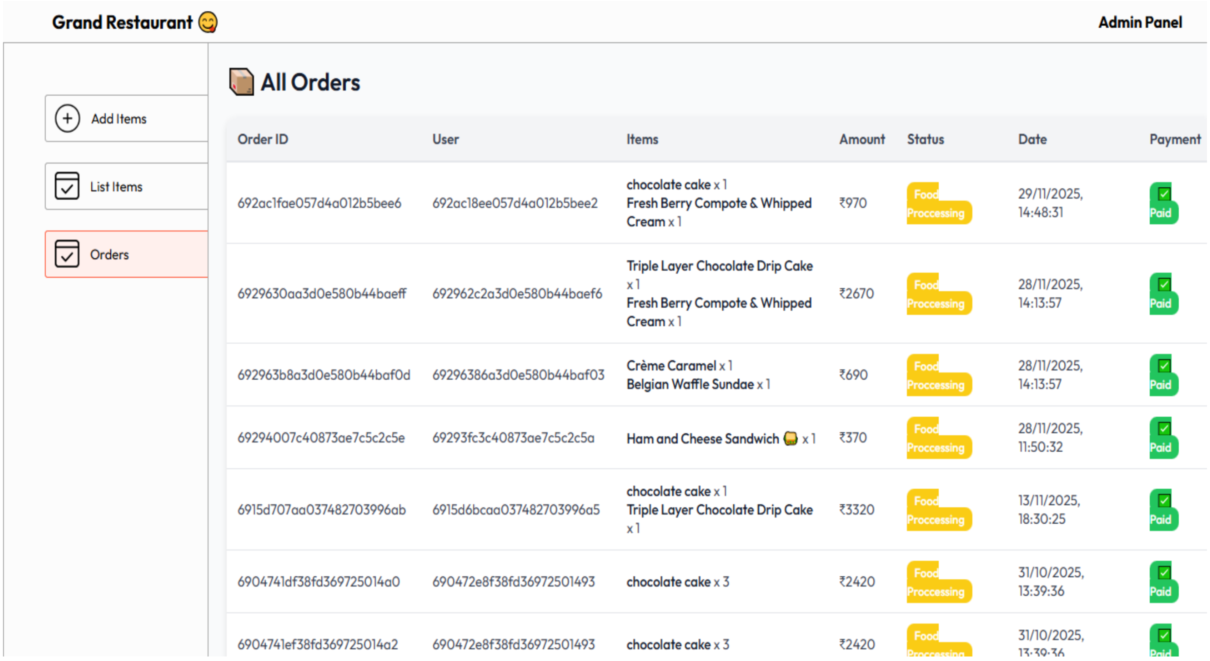
- List items



All Food List

Image	Name	Category	Price	Action
	chocolate cake	Cake	₹800	Edit Delete
	Triple Layer Chocolate Drip Cake	Cake	₹2500	Edit Delete
	Fresh Berry Compote & Whipped Cream	Deserts	₹150	Edit Delete
	Crème Caramel	Deserts	₹350	Edit Delete
	Rainbow Candy Cake	Cake	₹1500	Edit Delete
	Chocolate Ice Cream Delight Cake	Cake	₹2500	Edit Delete

- Orders



All Orders

Order ID	User	Items	Amount	Status	Date	Payment
692ac1fae057d4a012b5bee6	692ac18ee057d4a012b5bee2	chocolate cake x 1 Fresh Berry Compote & Whipped Cream x 1	₹970	Food Proccessing	29/11/2025, 14:48:31	✓ Paid
6929630aa3d0e580b44baeff	692962c2a3d0e580b44baef6	Triple Layer Chocolate Drip Cake x 1 Fresh Berry Compote & Whipped Cream x 1	₹2670	Food Proccessing	28/11/2025, 14:13:57	✓ Paid
692963b8a3d0e580b44baf0d	69296386a3d0e580b44baf03	Crème Caramel x 1 Belgian Waffle Sundae x 1	₹690	Food Proccessing	28/11/2025, 14:13:57	✓ Paid
69294007c40873ae7c5c2c5e	69293fc3c40873ae7c5c2c5a	Ham and Cheese Sandwich 🥪 x 1	₹370	Food Proccessing	28/11/2025, 11:50:32	✓ Paid
6915d707aa037482703996ab	6915d6bcaa037482703996a5	chocolate cake x 1 Triple Layer Chocolate Drip Cake x 1	₹3320	Food Proccessing	13/11/2025, 18:30:25	✓ Paid
6904741df38fd369725014a0	690472e8f38fd36972501493	chocolate cake x 3	₹2420	Food Proccessing	31/10/2025, 13:39:36	✓ Paid
6904741ef38fd369725014a2	690472e8f38fd36972501493	chocolate cake x 3	₹2420	Food Proccessing	31/10/2025, 13:39:36	✓ Paid

4.2 Report Views:

In the Grand Restaurant System, the report views are designed to provide both users and administrators with useful information.

1. User-side Reports

- Users can view a list of their submitted orders with items, amounts, and dates
- Order status is displayed (Food Processing, Completed, etc.)
- Users can access detailed information about food items they ordered, including name, price, category, and description

2. Admin-side Reports

- Admin can generate reports of all orders with details like user ID, items, amounts, and payment status
- Admin can view all food items in the system with attributes such as name, description, price, category, and images
- Admin can access all order requests submitted by users including order details and customer information
- Admin can track order history to see completed orders and maintain proper records

.

5.Conclusion

- Limitation
- Future Enhancement

5.1 Limitation:

- The system requires an active internet connection, so offline access is not possible.

- User and admin features are dependent on the database; if the database is large, performance may slow down.

- There is no real-time notification system for order status updates.

- The system does not include advanced search or AI-based recommendations for food items.

- Payment integration is not available in the current version.

5.2 Future Enhancement:

- Implement a real-time notification system to inform users about order status changes.

- Add advanced search and filtering options with AI-based food recommendations.

- Integrate payment functionality to support online ordering and transactions.

- Develop a mobile app version for easier access on smartphones.

- Include multi-language support to make the system accessible to a wider audience.

- Add a rating/review system for food items and restaurants to improve transparency and trust.

Annexures:

1. Progress Timeline:

- Week 1: Project planning, requirement analysis, and technology selection.

- Week 2: Designing system architecture, database design, and data dictionary.

- Week 3: Frontend development for User Panel – Home, Menu, and About pages.

- Week 4: Frontend development for User Panel – Register, Login, and Cart pages.

- Week 5: Admin Panel development – Login, Manage Food Items, and Manage Orders modules.

- Week 6: Admin Panel development – Order Management and Analytics modules.

- Week 7: Integration of frontend with backend and database connectivity.

- Week 8: Testing, debugging, and validation of both user and admin features.

- Week 9: Final documentation, report preparation

2. Technology Reference:

- Frontend: ReactJS, HTML5, CSS3, TailwindCSS, JavaScript

- Backend: Node.js, Express.js

- Database: MongoDB

- Tools: Visual Studio Code, MongoDB Compass, and Chrome

3.Testing Results:

User Registration & Login: Users can register and log in successfully – PASSED

Food Browsing & Ordering: Users can view menu items and place orders successfully – PASSED

Cart Management: Users can add/remove items from cart and proceed to checkout – PASSED

Order Status Tracking: Users can check the status of their orders – PASSED

Admin Login: Admin can log in securely to the admin panel – PASSED

Manage Food Items (Admin): Admin can add, update, and delete food records – PASSED

Manage Orders (Admin): Admin can view and manage all customer orders – PASSED

Order Processing (Admin): Admin can update order status and mark payments – PASSED

Data Integration: All user, food, and order data are correctly stored in the database – PASSED

4. Deployment Information

Platform: Localhost / Node.js server

Frontend Hosting: ReactJS build served via Vite dev server or hosting platform

Database: MongoDB (MongoDB Compass)

Monitoring: Browser console and server logs enabled for request tracking and debugging

5. User Guide

➢ For User:

1. Register or Login

2. Browse menu items (food categories, prices, descriptions)

3. Select items and place order

➢ For Admin:

1. Login as Admin

2. Manage food items

3. View and manage orders

4. Monitor user activities and system performance

References :

- https://stackoverflow.com/
- https://react.dev/
- https://www.mongodb.com/docs/
- https://expressjs.com/
- https://nodejs.org/docs/