



**MADRAS INSTITUTE OF TECHNOLOGY**  
**ANNA UNIVERSITY**



**DEPARTMENT OF INFORMATION TECHNOLOGY**

**NAAN MUDHALVAN**

**MERN STACK POWERED BY MONGODB**

**MIT FOODS - FOOD ORDERING APPLICATION**

**Team Members:**

**Lokesh M (2021506042)**

**Sankar Prasad P (2021506087)**

**Saravanakumar V (2021506089)**

**Suganth G (2021506111)**

# FOOD ORDERING APPLICATION

## Project Overview:

MIT Foods is a cutting-edge digital platform designed to transform the online food ordering experience. With our user-friendly web app, foodies can easily explore, discover, and order dishes tailored to their tastes. Comprehensive details about each dish, including descriptions, pricing, and discounts ensure well-informed choices. The ordering process is simple: enter your name, delivery address, and payment method, then place your order. You'll receive instant confirmation with no long queues or complex steps. MIT Foods makes food ordering quick, efficient, and hassle-free.

## Tech Stack:

### 1) Frontend:

**UI (React):** The frontend of the application is built using React, a popular JavaScript library for building user interfaces. React enables the creation of dynamic and interactive web pages, ensuring a seamless user experience.

### 2) Backend:

**Node.js:** The backend is powered by Node.js, a JavaScript runtime environment that allows server-side JavaScript execution. It enables the backend to handle requests and responses efficiently.

**Database:** The backend communicates with a MongoDB database, which is a NoSQL database known for its flexibility and scalability. It stores data such as user information or stock-related data. Mongoose is an ODM (Object Data Modeling) library for MongoDB and Node.js, providing a straightforward way to interact with MongoDB by defining models and schemas.

## Interface Setup:

### 1) Setup React Application:

- Create a React app in the client folder.
- Install required libraries
- Create required pages and components and add routes.

### 2) Design UI components:

- Create Components.
- Implement layout and styling.
- Add navigation.

### 3) Implement frontend logic:

- Integration with API endpoints.
- Implement data binding.

### Application 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.

### Frontend (React):

#### 1) Project Initialization:

- After creating the frontend project with `npx create-react-app client`, the necessary dependencies are installed.

#### 2) Folder Structure:

**/src:**

- **components/:** Contains UI components like `RestaurantList`, `RestaurantCard`, `DishesMenu`, `DishCard`, and `Cart`.
- **contexts/:** Holds `RestaurantContext.js`, which provides shared state management across the app.
- **App.js:** Main application component, which uses `React Router` to navigate between screens (like the restaurant list and cart).
- **index.js:** Wraps the app in the `RestaurantProvider`, allowing access to the context throughout.

### 3) Context API Setup:

#### **RestaurantContext:**

- Defines global state management for restaurant and cart data.
- Enables components to access and update shared states like the selected restaurant, menu items, and cart contents.
- Uses `React.createContext()` to initialize and `useContext` to access data within components.

### 4) Component Structure:

#### **RestaurantList:**

- Displays a list of all available restaurants, using `RestaurantCard` to render individual items.

#### **RestaurantCard:**

- Shows details about each restaurant (e.g., name, location).
- Clicking a restaurant could trigger navigation to the restaurant's menu page.

#### **DishesMenu:**

- Shows the dishes available for the selected restaurant.
- Uses `DishCard` to render individual dishes, with an "Add to Cart" button.

#### **Cart:**

- Lists items added to the cart and provides options to increase/decrease item quantities or remove items.
- Total cost calculation is handled here.

#### **App.js:**

- Routes are managed here, linking pages like the restaurant list, menu, and cart.
- `BrowserRouter` and `Route` from `react-router-dom` provide navigation.

### **Backend (Node.js + Express):**

#### 1) Project Initialization:

- Set up the backend by creating a folder and initializing a package with `npm init -y`, then installing required dependencies.

#### 2) Folder Structure:

**server.js:** Main server file where Express is configured.

**/models:** Contains Mongoose models like `Restaurant.js` and `Dish.js`, which define schemas for storing data in MongoDB.

**/routes:** Contains route files (e.g., `restaurantRoutes.js`, `menuRoutes.js`), which define

endpoints for fetching and manipulating restaurant, menu, and cart data.

**/controllers:** Optional, but recommended. Contains logic for handling requests (e.g., fetching restaurant data or handling cart actions).

### 3) Backend Dependencies:

**cors:** Middleware for enabling cross-origin requests from the frontend.

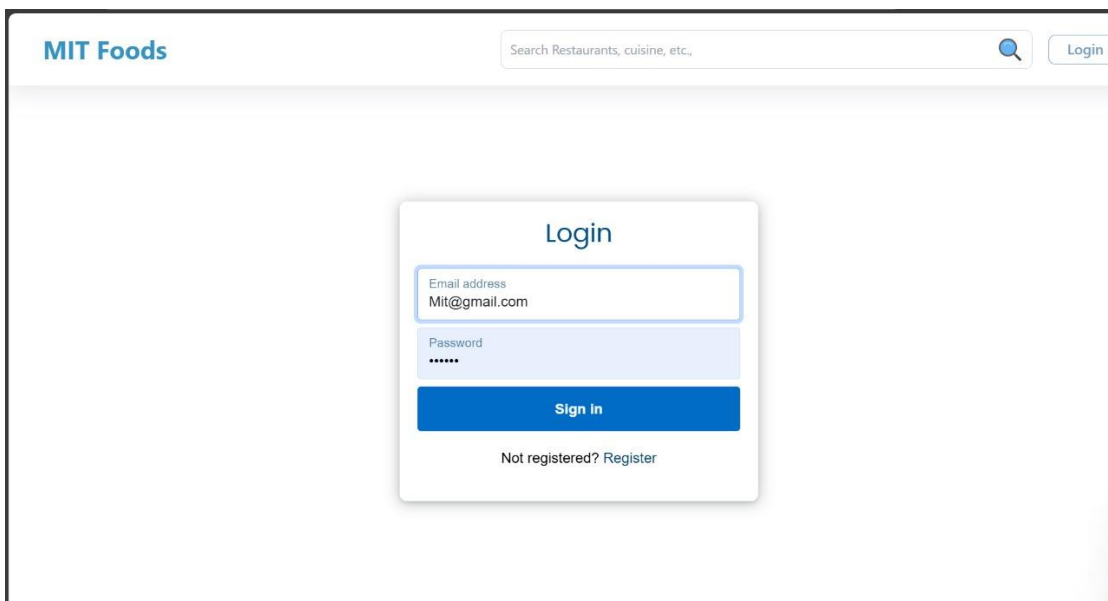
**express:** Framework for creating server and handling routes.

**mongoose:** ODM for MongoDB, making data interaction more efficient.

**nodemon:** Development tool for automatically restarting the server on file changes.

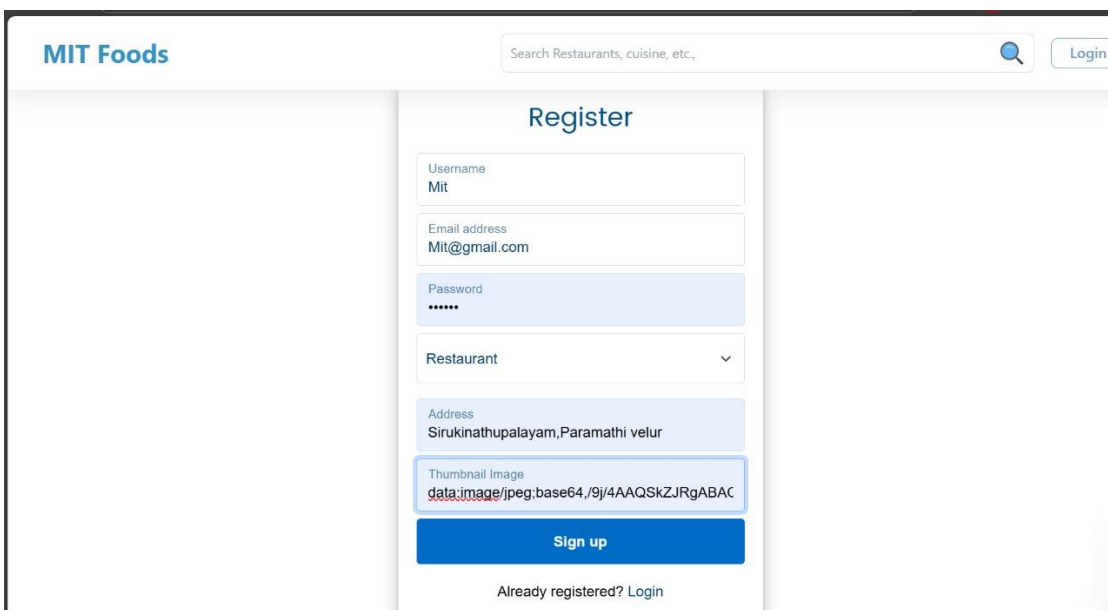
### Screenshots:

#### User Login:



The screenshot shows the MIT Foods login interface. At the top, there is a header with the MIT Foods logo on the left, a search bar with the placeholder text "Search Restaurants, cuisine, etc." in the center, and a "Login" button on the right. Below the header, a central white box contains the "Login" form. The form has two input fields: "Email address" with the value "Mit@gmail.com" and "Password" with masked characters "\*\*\*\*\*". Below these fields is a blue "Sign in" button. At the bottom of the box, there is a link that says "Not registered? Register".

#### User Registration:



The screenshot shows the MIT Foods registration interface. At the top, there is a header with the MIT Foods logo on the left, a search bar with the placeholder text "Search Restaurants, cuisine, etc." in the center, and a "Login" button on the right. Below the header, a central white box contains the "Register" form. The form has five input fields: "Username" with the value "Mit", "Email address" with the value "Mit@gmail.com", "Password" with masked characters "\*\*\*\*\*", "Restaurant" with a dropdown arrow, and "Address" with the value "Sirukinathupalayam, Paramathi velur". Below these fields is a "Thumbnail Image" field containing a base64 encoded string: "data:image/jpeg;base64,/9j/4AAQSkZJRgABAC". Below the form is a blue "Sign up" button. At the bottom of the box, there is a link that says "Already registered? Login".

## Admin Page:

MIT Foods (admin)

HomeUsersOrdersRestaurantsLogout

Total users  
7  
[View all](#)

All Restaurants  
6  
[View all](#)

All Orders  
2  
[View all](#)

Popular Restaurants(promotions)

☒ Jaya Mess

☐ Bhilal

☐ Anjappar

☐ Muniyandi Vilas

☒ Kurinji

[Update](#)

Approvals


No new requests...

## Customers' Orders:

MIT Foods (admin)

HomeUsersOrdersRestaurantsLogout

Orders



Jaya mess

Userid: 6741f601aecbdf8457948e2f Name: Suganth.G Mobile: 7825071868 Email: suganthskp@gmail.com

Quantity: 1 Total Price: ₹ 114 ~~₹ 120~~ Payment mode: cod


Address: 3/168 , Arasamarathuthottam,Sirukinathupalayam Pincode: 637204 Ordered on: 2024-11-23 Time: 15:36

status: order placed

Update order status

Update

Cancel



Chicken 65

Jaya Mess

Userid: 6741f601aecbdf8457948e2f Name: Suganth.G Mobile: 7825071868 Email: suganthskp@gmail.com

Quantity: 2 Total Price: ₹ 196 ~~₹ 200~~ Payment mode: cod

Address: 3/168 , Arasamarathuthottam,Sirukinathupalayam Pincode: 637204 Ordered on: 2024-11-23 Time: 15:36

status: order placed

Update order status

Update

Cancel

## Adding food items in Restaurant:

MIT Foods (Restaurant)

HomeOrdersMenuNew ItemLogout

New Product

Product name  
Dosa

Product Description  
Crispy...

Thumbnail Img url

Gender

☒ Veg ☐ Non Veg ☐ Beverages

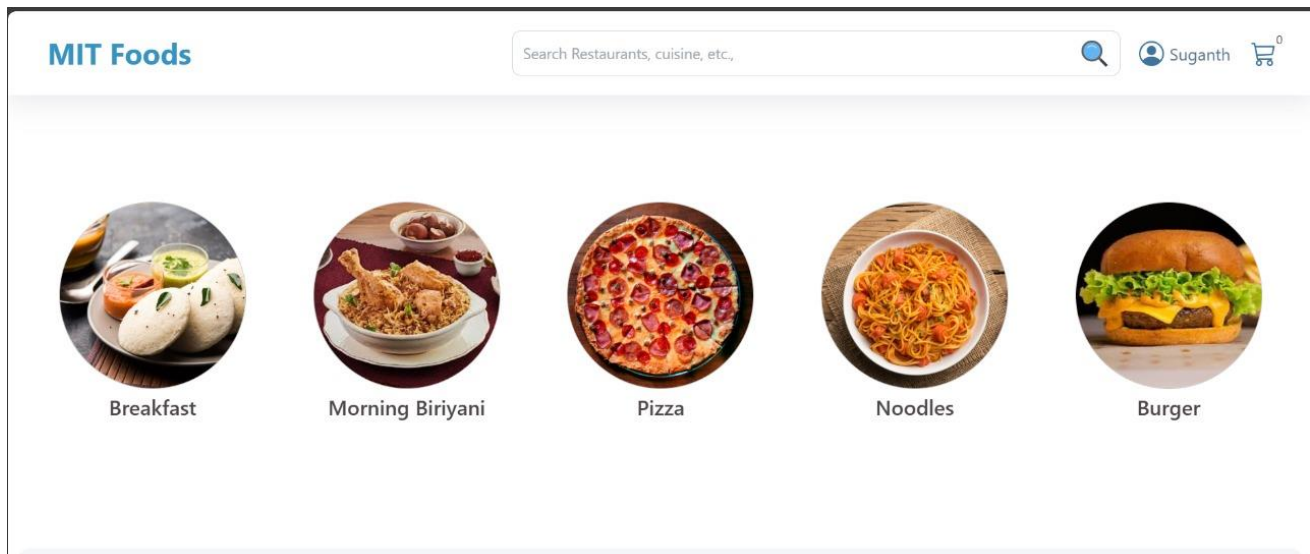
Category  
Spicy Indian

Price  
125

Discount (in %)  
0

Add product

## Main Page:



## Page Of Restaurants:

### MIT Foods

Search Restaurants, cuisine, etc.,



Suganth



#### All restaurants



**Jaya Mess**  
Chrompet



**Bhilal**  
Chrompet, Near Saravana stores



**Anjappar**  
Chrompet, Opposite to Vetri Theatre



**Muniyandi Vilas**  
Chrompet, Mit main gate



## Food Items In Corresponding Restaurants:

### MIT Foods

Search Restaurants, cuisine, etc.,



Suganth



### Kurinji

Chittlapakkam, Near Police Station

#### All Items

##### Filters

##### Sort By

- ☐ Popularity
- ☐ low-price
- ☐ high-price
- ☐ Discount
- ☐ Rating

##### Food Type

- ☐ Veg
- ☐ Non Veg
- ☐ Beverages

##### Categories

- ☐ Spicy Indian
- ☐ Breakfast
- ☐ Pizza
- ☐ Burger



**Parotta**  
Soft.....  
₹ 15 ~~15~~

Add item



**Meals**  
Unlimited Food.....  
₹ 80 ~~100~~

Add item



**Prawn 65**  
Crispy.....  
₹ 70 ~~70~~

Add item



**Biryani**  
Morning Biryani.....  
₹ 150 ~~150~~

Add item




## Ordering Of Food Items:

**MIT Foods**

Search Restaurants, cuisine, etc.,

Suganth

1



**Parotta**  
Kurinji  
Quantity: 2 Price: ₹ 15 ₹15  
Remove

**Price Details**

Total MRP: ₹ 30  
Discount on MRP: - ₹ 0  
Delivery Charges: + ₹ 50

**Final Price: ₹ 80**

Place order


## Payment Page:

**MIT Foods**

Search Restaurants, cuisine, etc.,

Suganth

1



**Parotta**  
Kurinji  
Quantity: 2  
Remove

**Checkout**

**Checkout details**

Name  
Suganth.G

Mobile  
7825071868

Email  
suganthskp@gmail.com

Address  
3/168 , Arasamarathuthottam,Siruk

Pincode  
637204

**Payment method**

Choose Payment method  
card payments

cancelOrder

**Price Details**

Total MRP: ₹ 30  
Discount on MRP: - ₹ 0  
Delivery Charges: + ₹ 50

**Final Price: ₹ 80**

Place order

## Conclusion:

In conclusion, the project successfully integrates a user-friendly frontend with React, a robust backend powered by Node.js and Express.js, and a flexible database solution with MongoDB and Mongoose. This architecture ensures efficient real-time data handling, scalability, and seamless interaction between the user interface and backend.