PES UNIVERSITY, BENGALURU

Department of Computer Science and Engineering

B. Tech (CSE) – 5th Semester – Aug-Dec 2024

UE22CS341A - Software Engineering Synopsis / Abstract

ZapFlavor: Online Food Ordering System

TEAM:

NAME	SRN	SECTION
N S TUSHAR	PES2UG22CS327	F
MRUNAL RAVISHANKAR A	PES2UG22CS323	F

Problem Statement:

In today's fast-paced lifestyle, individuals face challenges when it comes to ordering food conveniently and efficiently. Customers often struggle to find local restaurants, place orders, and track deliveries in real-time. On the other hand, restaurants may find it challenging to efficiently manage their online orders and payments while ensuring a seamless delivery experience. Traditional food ordering processes lack convenience, real-time tracking, and efficient management, leading to a disconnect between customers and restaurants, and ultimately impacting the overall food ordering experience.

Solution:

The Online Food Ordering System provides a user-friendly platform for customers to browse restaurants, place orders, and track deliveries. Restaurants can efficiently manage their listings, process orders, and track payments, while delivery partners can accept orders and track the delivery route in real time.

Tech Stack:

- **Frontend:** EJS Templating, HTML5, CSS3
- **Backend:** Node.js, Express.js
- **Database:** MySQL
- **Authentication:** Passport Authentication
- **Deployment:** Render
- **Version Control:** Git, GitHub
- **APIs:** RESTful APIs
- **Testing:** Jest, ThunderClient

Impact:

The Online Food Ordering System has the potential to significantly improve the food ordering and delivery process by making it more transparent, efficient, and convenient. By connecting customers with the right restaurants and delivery partners, the platform can enhance the overall dining experience, support local businesses, and contribute positively to the food industry. The streamlined processes and real-time tracking can also lead to cost savings for restaurants and a more satisfying ordering experience for customers..

UE22CS341A: Software Engineering
Software Requirements Specification (SRS)

Online Food Ordering System

TEAM:

NAME	SRN	SECTION
N S TUSAHR	PES2UG22CS327	F
MRUNAL RAVISHANKAR A	PES2UG22CS323	F

A Software Requirements Specification (SRS) document for **Online Food Ordering System.** For the development of this system, the Agile-Model has been selected as the SDLC methodology.
Below is an outline of the SRS document, along with an example of a Requirements Traceability Matrix (RTM).

1. Introduction:

1.1 Purpose

The **Online Food Ordering System** aims to provide an efficient and user-friendly platform for customers, restaurants to interact seamlessly. It facilitates food ordering, payment processing, and delivery tracking. The purpose of this document is to describe the software system in detail, ensuring that the requirements are clear for developers, stakeholders, testers, and users.

1.2 Intended Audience and Reading Suggestions

This document is intended for the following audience:

- Developers: To guide the implementation of the system.
- Project Managers: To track the project's progress and ensure that it meets the specified requirements.
- Testers: To design test cases based on the requirements outlined.
- Users: To understand the features and functionality of the system.
- Documentation Writers: To produce user manuals and help guides.

1.3 Product Scope

The system is designed to:

- Simplify online food ordering through a web and mobile interface.
- Help restaurants manage orders and streamline online sales.
- Provide delivery partners with tools to track orders and optimize delivery routes.

Ultimately, it aims to deliver a seamless experience, combining features like real-time order tracking, secure payments, and quick browsing.

1.4 References

The system adheres to best practices and standards, as outlined in:

- IEEE Software Engineering Standards for structured and clear requirement specifications.
- MySQL for handling databases..
- React.js and Node.js documentation for implementing the frontend and backend.

2. Overall Description:

2.1 Product Perspective

The **Online Food Ordering System** is positioned as an independent system that provides web-based interfaces for three primary user types (customers, restaurants, and delivery partners). It integrates external services for payments and delivery tracking, making it an interconnected system but capable of functioning independently with its own data storage and business logic.

2.2 Product Functions

- **User Registration and Login**: Secure sign-in for all user roles (customers, restaurants, delivery partners).
- **Browse Restaurants**: Search based on location, cuisine, or ratings.
- **Order Placement**: Customers can add items to a cart, specify delivery instructions, and confirm their orders.

2.3 User Classes and Characteristics

- **Customers**: Need a simple, intuitive interface for browsing restaurants, placing orders, and tracking deliveries.
- **Restaurants**: Require a robust back-end to manage menus, orders, and payments.
- **Delivery Partners**: Need a mobile-friendly interface to accept orders and update delivery status in real-time.
- **Administrators**: Oversee platform operations, ensuring system stability, handling disputes, and managing users.

2.4 Operating Environment

The system will operate in a web environment, accessible through major browsers (Chrome, Firefox, Safari, Edge) on both desktop and mobile devices. It will be hosted on a cloud platform (Render) with MySQL as the database and Node.js/Express.js for the backend.

2.5 Design and Implementation Constraints

- **Responsiveness**: The system must adapt to both mobile and desktop screen sizes.
- **Security**: HTTPS is mandatory for all communications, and user authentication will use OAuth 2.0.
- **Scalability**: The system must be able to handle high traffic loads during peak times, especially on weekends or holidays when food orders typically spike.
- **Localization:** The system should be designed in such a way that it can easily support multiple languages and currencies as the platform expands geographically. Assumptions and Dependencies
  - ❖ Users will have access to the internet and modern web browsers.
  - ❖ The system will rely on third-party services such as Passport Authentication.
  - ❖ Development will depend on timely feedback from stakeholders.

3. External Interface Requirements:

3.1 User Interfaces

- **Login/Registration Page**: This page allows users to create accounts or sign in using existing credentials. It should also allow social logins (Google, Facebook).
- **Restaurant Listings Page**: A list of available restaurants based on the user's location. Restaurants can be filtered by distance, cuisine, and rating.
- **Order Placement Page:** Once users select an Item they can place a order for that.

3.2 Software Interfaces

- ❖ Integration with passport module of NodeJS for Authentication.
- ❖ RESTful APIs for communication between frontend and backend.
- ❖ ❖ MySQL database for data storage.

3.3 Communications Interfaces

- ❖ HTTPS protocol for secure data transmission.
- ❖ Email notifications for account activities and in app updates.

4. System Features:

4.1 System Feature 1: User Authentication

4.1.1 Description and Priority

- ❖ Secure registration and login for looking up restaurants and placing orders. Priority: High

4.1.2 Stimulus/Response Sequences

- ❖ Users enter their credentials; the system verifies them and grants access.

4.1.3 Functional Requirements

- ❖ REQ-1: The system shall allow users to register using an email address or ID.
- ❖ REQ-2: The system shall allow users to log in using their registered email or ID and password.
- ❖ REQ-3: The system shall allow the customer to add and update their food order before ordering.

4.2 System Feature 2: Browse and Search Food Items

4.2.1 Description and Priority

Customers need an interface to explore food items available across different restaurants. The system should allow users to search for specific food items based on their preferences, including filters such as cuisine type, price range, and dietary restrictions.

Priority: **High**

Functional Requirements:

REQ-4: The system must allow users to browse through food items from multiple restaurants. **REQ-5**: A search bar should enable users to find specific food items by name (e.g., "pizza", "sushi", "burger").

REQ-6: Users must be able to filter food items based on price, cuisine, popularity, and special dietary needs (e.g., vegetarian, gluten-free, etc.).

REQ-7: Each food item must display relevant information, such as a short description, price, ingredients, and restaurant offering it.

REQ-8: Users should be able to sort food items by popularity, rating, or price.

System Feature 3: Job Posting and Management

4.3.1 System Feature 3: Order Placement

Description and Priority

Customers can browse available food items, add them to a cart, and place orders for delivery or pickup.

Priority: High

Functional Requirements:

- **REQ-9**: Users must be able to add items from multiple restaurants and categories (e.g., appetizers, mains, desserts) into a single order.
- **REQ-10**: An order confirmation page should summarize selected food items, prices, delivery options, and special instructions.
- **REQ-11**: The system must provide options for users to customize orders (e.g., extra toppings, sauce choices) where applicable.
- **REQ-12**: Users must be able to specify delivery or pickup options, along with the estimated delivery time.

5. Other Non-Functional Requirements:

5.1 Performance Requirements

- The system must handle simultaneous browsing of food items by at least large users with minimal delays.
- The search functionality for food items should return results within a few seconds.

5.2 Usability

- The user interface for searching and filtering food items must be intuitive, with clear buttons and easy-to-use filters.
- The design must be responsive, allowing users to comfortably browse on both desktop and mobile devices.
- The system should provide clear feedback to the user during interactions (e.g., loading indicators when searching for food items).

5.3 Scalability Requirements

- The system must be capable of scaling to accommodate up to very large thousands of users during peak times without degradation in performance.
- As the number of food items increases, the system must maintain quick search functionality, able to scale to accommodate up to 100,000 food items across different restaurants.
- The infrastructure should allow seamless addition of new restaurants and food items without requiring system downtime.

5.4 Reliability

- The system should have 99.9% uptime, ensuring that the platform is available for users to place orders at any time.
- The system should include mechanisms for error handling and fallback strategies, ensuring that in the event of a partial system failure (e.g., one service goes down), other essential functionalities (like browsing food items) are still available.

5.5 Maintainability

- The system should be built with modular components, allowing individual features (e.g., the search function or payment gateway) to be updated without affecting other parts of the system.
- Documentation for both the codebase and APIs should be thorough, making it easy for developers to maintain and expand the system.

6. Other Requirements:

6.1 Database Requirements

- The system shall use MySQL for data storage.
- The database should be normalized.

6.2 Internationalization Requirements

- The system should support multiple languages in the future.

6.3 Legal Requirements

- The system must comply with data protection laws applicable in the regions it operates.

7. Appendices:

Appendix A: Glossary

- **Customer**: An individual who uses the platform to browse restaurant menus, place food orders, and track deliveries.

❖ **Restaurant**: A food service provider that lists its menu, accepts orders, and manages delivery or pickup through the platform.

❖ **Admin**: A user with special privileges to manage the platform, including restaurant accounts, customer profiles, orders, and system settings.

❖ **Profile**: The collection of personal or business information created and managed by customers or restaurants on the platform.

❖ **Menu**: A list created by restaurants showcasing available food items along with descriptions, prices, and special offers.

❖ **Order**: The process through which a customer selects menu items and submits a request for food delivery or pickup from a restaurant.

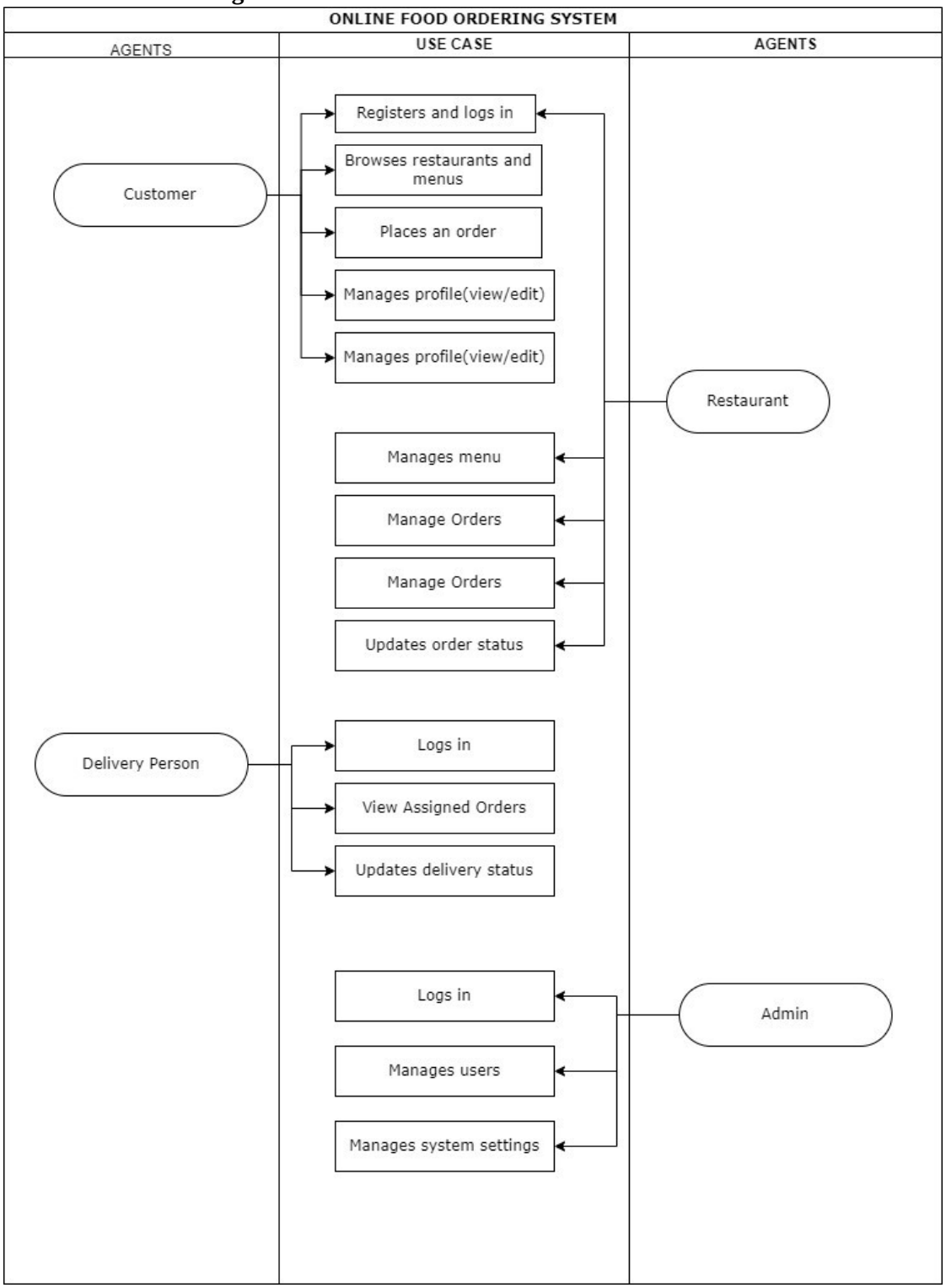
❖ **Dashboard**: The user interface provided to customers and restaurants for managing their interactions with the platform, including order placement, and account management.

Appendix B: Requirement Traceability Matrix (RTM)

The RTM ensures that all requirements are covered by design, development, and testing activities. Below is a rough RTM for the corresponding project.

Requirement ID	Description	Design Specification	Implementation module	Test Case ID
FR-1	User Authentication	UserAuth-Design	UserAuth-Module.js	TC-01
FR-2	Customer Profile Management	ProfileMgt-Design	ProfileMgtModule.js	TC-02
FR-3	Restaurant Menu Management	MenuMgt-Design	MenuMgtModule.js	TC-03
FR-4	Order Placement	Order-Design	OrderModule.js	TC-04
FR-5	Admin Panel	AdminPanel-Design	AdminPanelModule.js	TC-05
NFR-1	Usability	Usability-Design	Usability-Module.js	TC-06

Appendix C: Use Case Diagram



Project Plan Documentation(PPD)

Life Cycle Used

Spiral Model :

The decision to utilize the Spiral Model for this project stems from its systematic and iterative approach to development, which prioritizes risk analysis and refinement at each phase. This model is particularly suitable for the Comprehensive Recruitment System due to its ability to accommodate evolving requirements, frequent prototyping, and effective risk management.

Reason :

1. **Risk management**: The system's complexity, including payment integration, order tracking, and discount management, makes it essential to identify and mitigate risks at each phase.
2. **Iterative refinement**: Since requirements such as premium customer discounts and secure payments may change based on feedback, iterative development allows continuous refinement.
3. **Flexibility**: The model allows adaptability to changing business requirements, ensuring that the project evolves based on real-world feedback.
4. **Prototyping**: Each cycle of the Spiral Model allows the development of prototypes, which will help ensure functionality and usability, especially for customer and restaurant management interfaces.

Tools to Be Used Throughout the Lifecycle

- Planning Tool:
  - **Jira or Trello**: For task management, sprint planning, and tracking progress using Kanban boards or sprints.
- Design Tool:
  - **Lucidchart or Microsoft Visio**: For creating ER diagrams, flowcharts, and architectural designs.

- Version Control:
  - **Git/GitHub**: For source code versioning, branch management, and collaboration among team members.
- Development Tool:
  - **Visual Studio Code**: A lightweight, feature-rich IDE for writing and managing code, with integrations for linting, debugging, and live collaboration.
- Bug Tracking:
  - **Jira**: For tracking bugs, issues, and tasks within sprints and generating reports.
- Testing Tool:
  - **Selenium**: For automated functional testing.
  - **Postman**: For testing and automating API calls.
  - **Jest**: For unit testing of the backend and frontend logic.

Deliverables and Categorization

The project deliverables are divided into two categories: **Reuse Components** and **Build Components**.

Reuse Components:

- **Third-party libraries:** For payment processing, geolocation, and user authentication(Passport).
- **Design patterns:** For common architectural elements. Eg: client-server, MVC
- **Code snippets:** For reusable functions or algorithms.

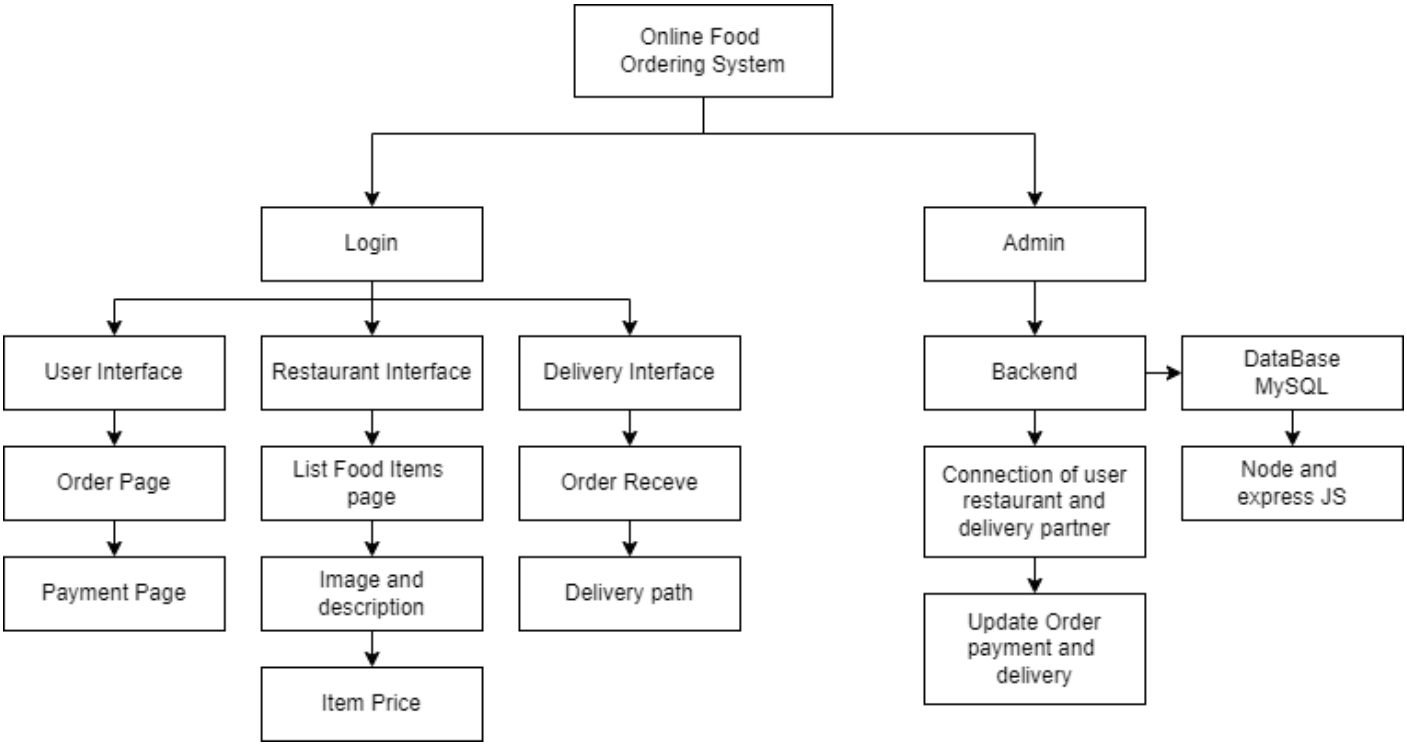
Justification: Reusing existing components (libraries and services) for authentication, and UI ensures security and efficiency, as these components are already well-tested and reliable.

Build Components:

- **Customer/User Interface**: A custom-built interface for customers to browse restaurants, place orders, and track their order status**.**
- **Restaurant Interface**: A management dashboard for restaurants to update menus, process orders, and track sales**.**
- **Delivery Partner Interface:** A dedicated interface for delivery personnel to receive orders, track deliveries, and update delivery statuses.
- **Admin Interface:** A comprehensive control panel for system administrators to manage customers, restaurants, delivery partners, and overall system **monitoring.**

Justification: These components are specific to the business requirements and need to be custom-built to meet the exact functional needs of the system.

Work Break Down Structure

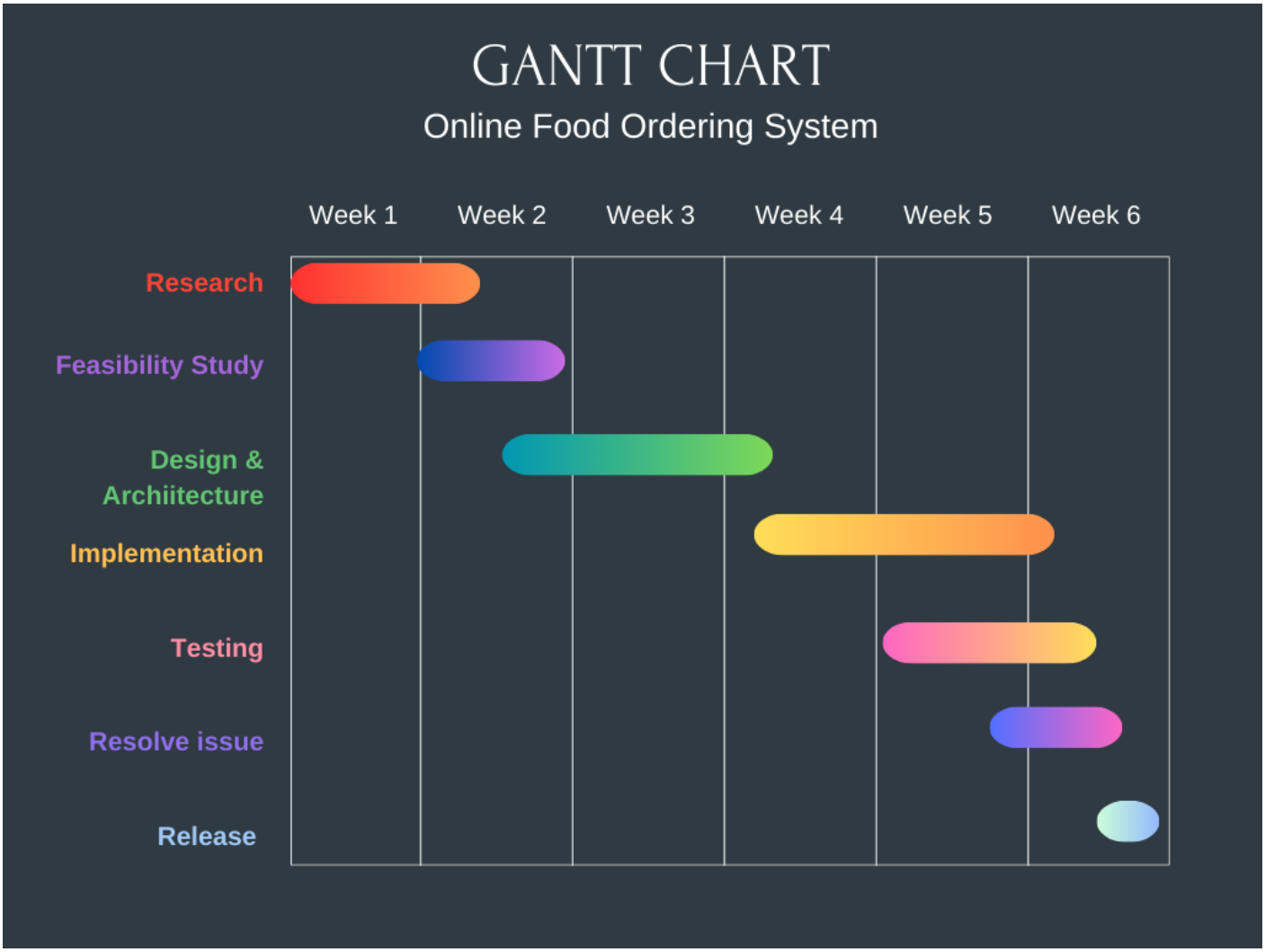


Effort Estimation

Task	Effort (Person Months)
Requirement Analysis	0.10
System Design	0.25
Database Design	0.10
Frontend Development	0.15
Backend Development	0.25
Testing	0.10

Task	Effort (Person Months)
Documentation	0.30
Total Effort	**1.25 Person Months**

Gantt Chart



Coding Details

1. **Frontend Code**:
   - Written in **Ejs Templating**, utilizing components for reusable UI elements.
   - Implement forms for order placement, restaurant management, and payment

processing.
- Handle user authentication and session management.

2. **Backend Code**:
   - Written in **Node.js** with **Express.js** for handling API requests.
   - Use **MySQL** schema design and database operations.
   - Implement RESTful APIs for interacting with the database, managing orders, users, and payments.
   - Apply business logic for premium customer discounts.

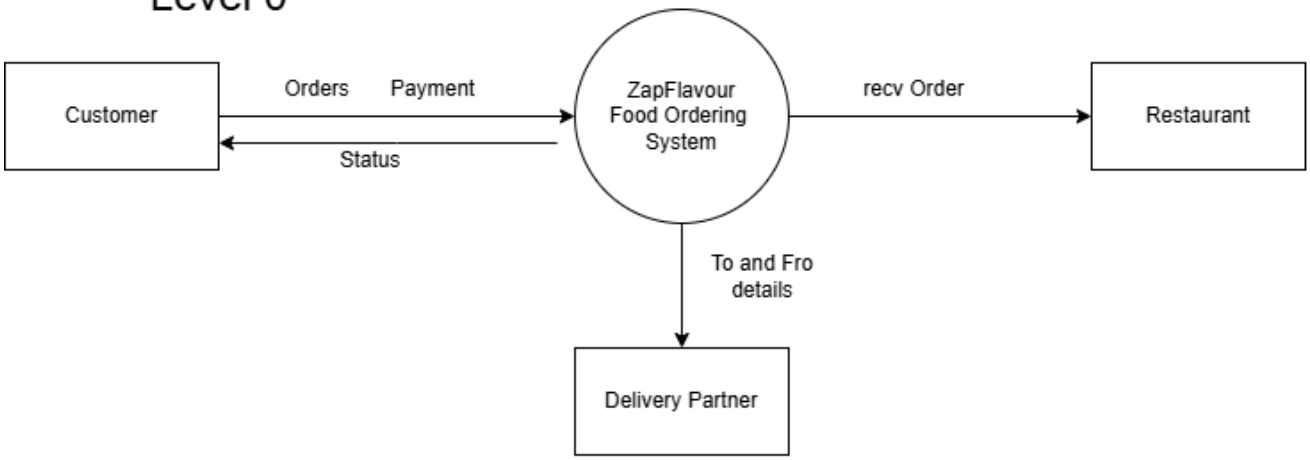
3. **Database Code**:
   - Design MySQL collections and schemas for entities like Users, Restaurants, MenuItems, Orders, Payments, Delivery partners.
   - Implement database triggers and stored procedures for managing order and payment workflows.
4. **Testing Code**:
   - **Jest** for unit testing individual components.
   - **Postman** for API testing.
   - **Selenium** for automated UI testing.

Design Document

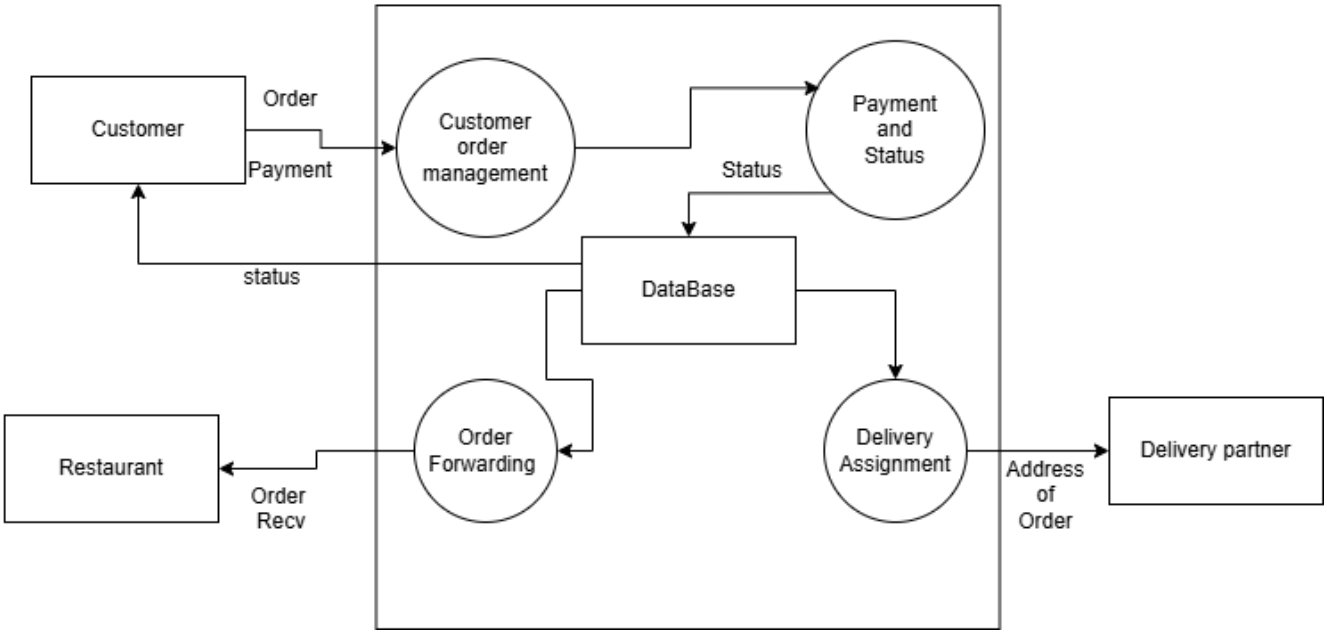
Design Diagrams

Diagrams of Levels of DFD

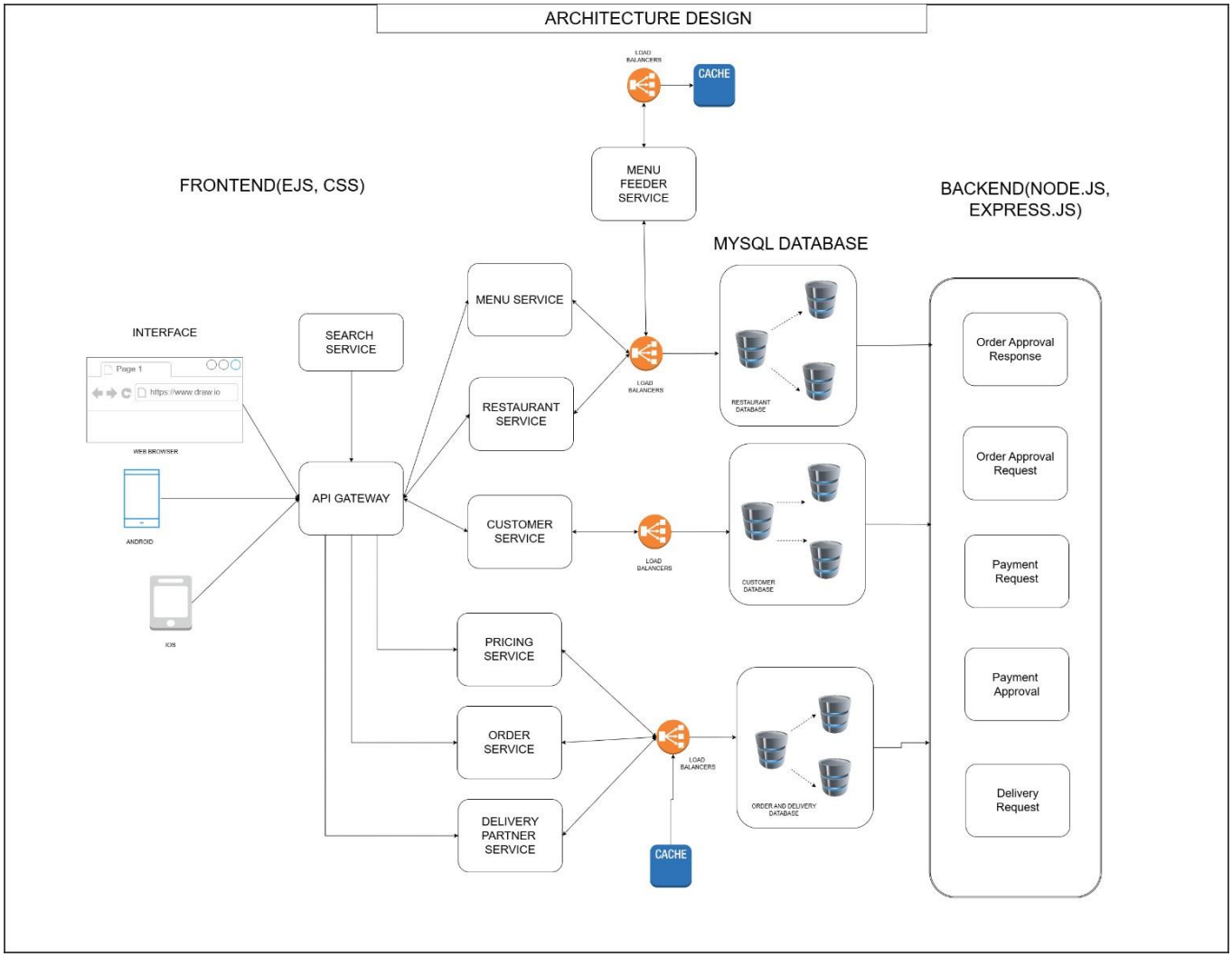
Level 0



Level 1

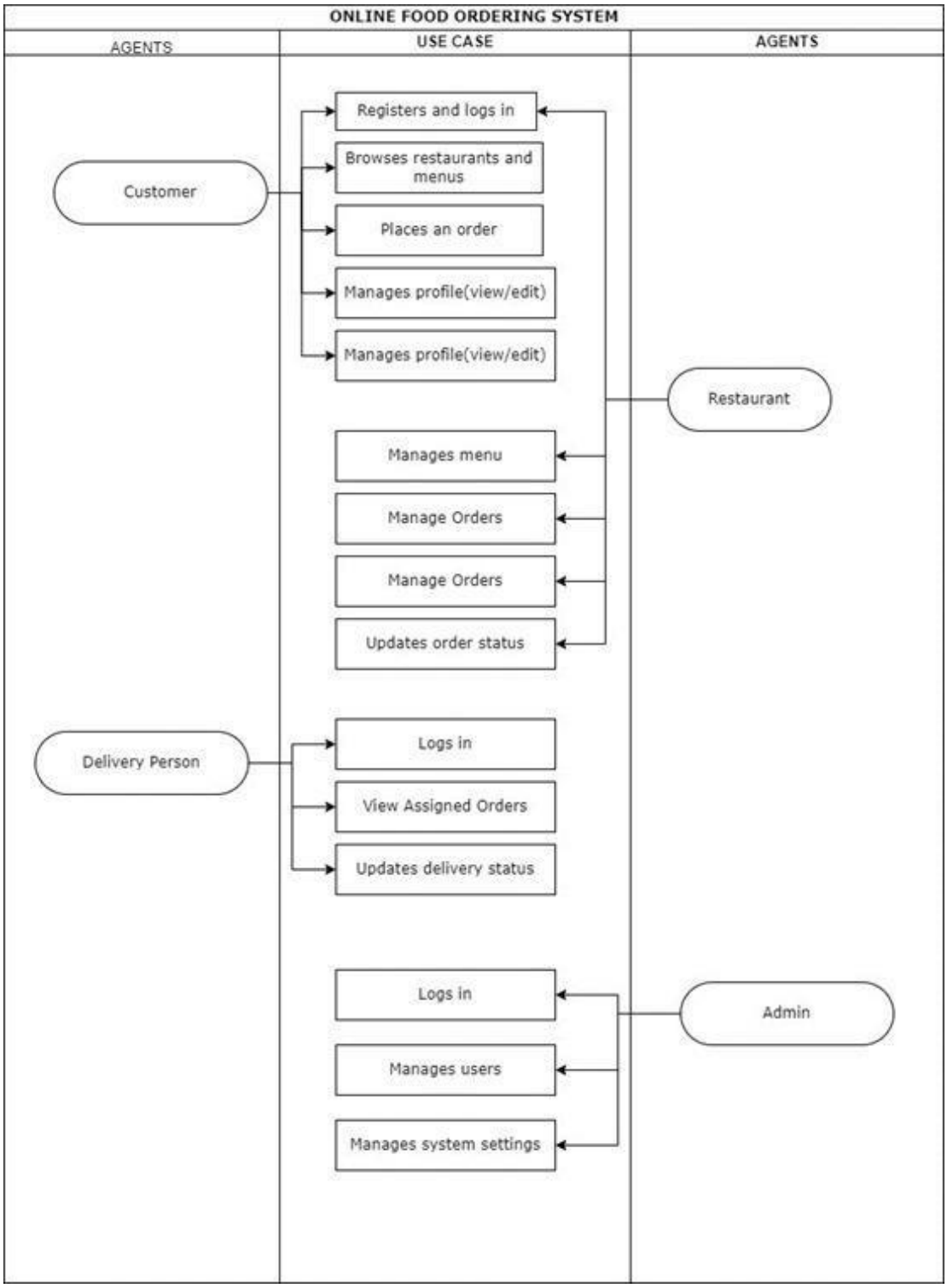


Architectural Design

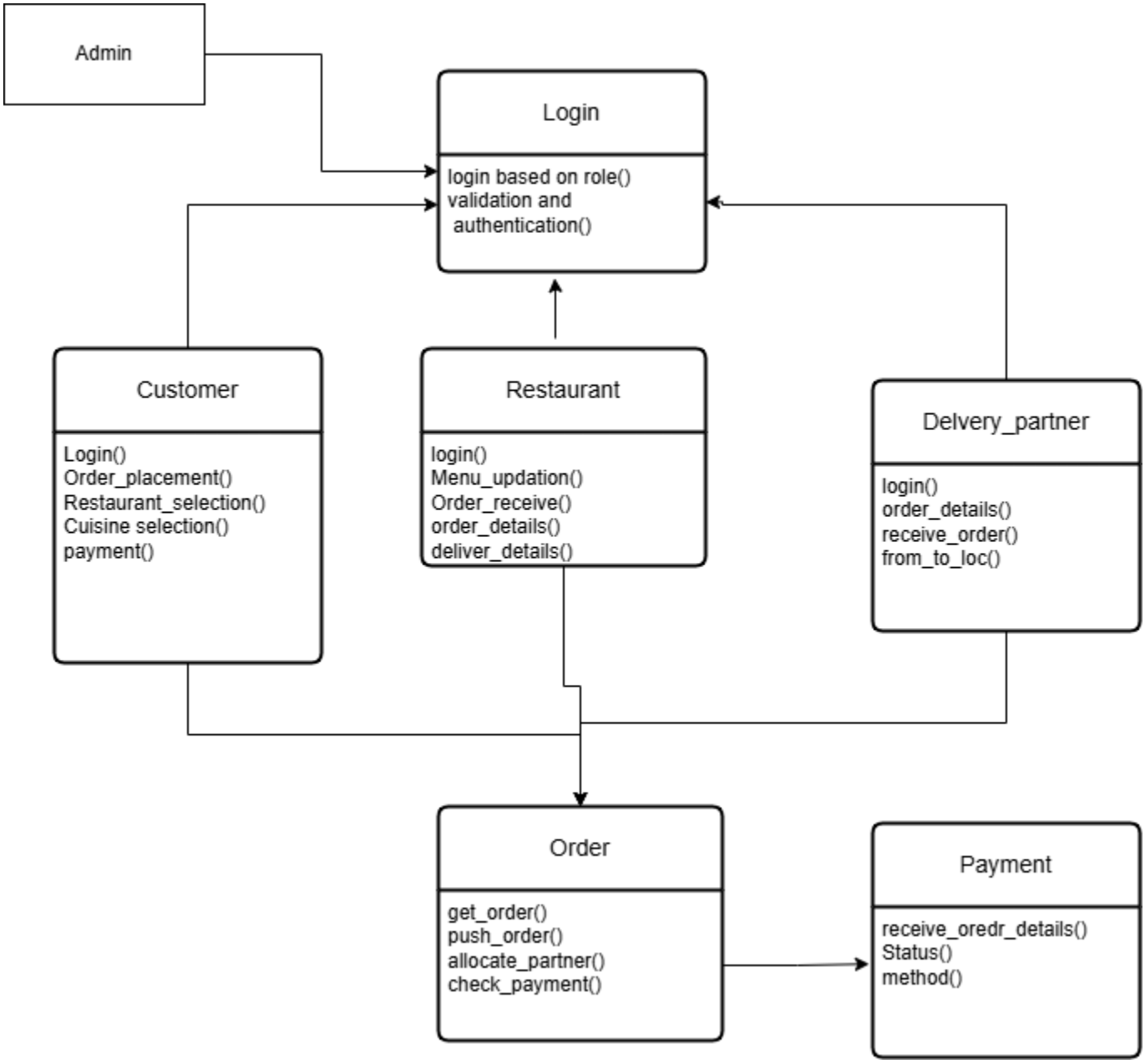


UML

Use Case Diagram



Class Diagram



Test Document

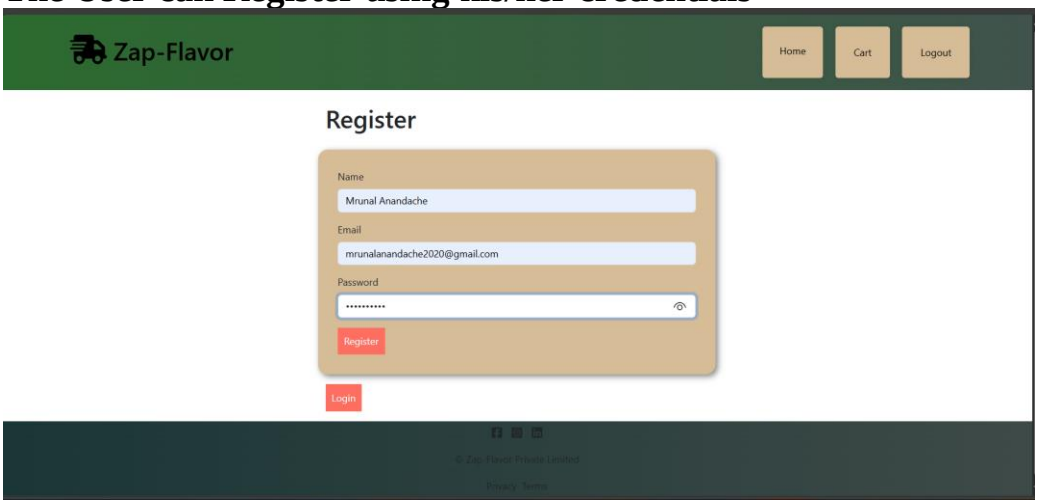
Test CaseID	Name of Module	Test Case Description	Pre Conditions	Test Steps	Test Data	Expected Results	Actual Results	Test Results
TC 01	User Registration	Verify that a new user can successfully register an account.	User is on the registration page.	Enter username,email and password	Username:Mrunal, email:mrunal@gmail.com, password:mru123	User account should be created successfully	Account creation isdone successfully	Pass
TC 02	User Login	Verify that a user who has registeredcan now login	User is on the Login page	Enter email and password	Username:Mrunal, email:mrunal@gmail.com, password:mru123	User should be logged in and redirected to home page	Login and redirect to home page	Pass
TC 03	Listing/Updating New Restaurants	Verify that the admin has the access to list restaurants, and also edit the information	Admin should be on the Home page	Click the Edit/ Delete button to carry out this functionality	Click on the edit/ delete btn to render a form to proceed	A form with all the details from where the admin can edit / delete	Updation of the restaurant details / Deletion of the restaurant	Pass
TC 04	Listing/Updat i ng New MenuItems	Verify that the admin is able to modify the menu items / delete/ update	Admin should be on the page of specific restaurant	Click the Edit/ Delete button to carry out this functionality	Click on the edit/ delete btn to render a form to proceed	A form with all the details from where the admin can edit / delete	Updation of the restaurant details / Deletion of the restaurant	Pass
TC 05	Cart Updation	Verify that user is able to choose theitems and push to cart	User should be on the page of the restaurant and should choose a valid quantity of item	choose a valid number of food items to order and then press add to cart	no test data, thereis a procedure to do follow to achieve this functionality	the items selected by the user should be updated into the cart section	The user can view the updated itemslist in the cart section along with the total amount	Pass
TC 06	Order Placement	Verify that user canplace order after inserting the items into the cart	User should be on the cart page	insert food items into cart and then press place order btn	no test data, thereis a procedure to do follow to achieve this functionality	the user should be redirected to other page where he will see the delivery partner assigned	the order is placed and a delivery partner is assigned	Pass

TC 07	Make Delivery Partner Available	Verify that the admin can make any delivery partner as available	Admin should click on the navbar button of delivery partners and should be on that redirected page	Admin should press the set available button in front of a delivery partner who is assigned	no test data, thereis a procedure to do follow to achieve this functionality	the delivery partners who are assigned to ordersshould be made available to take up other orders	the delivery partner is made availableto take up other orders	Pass

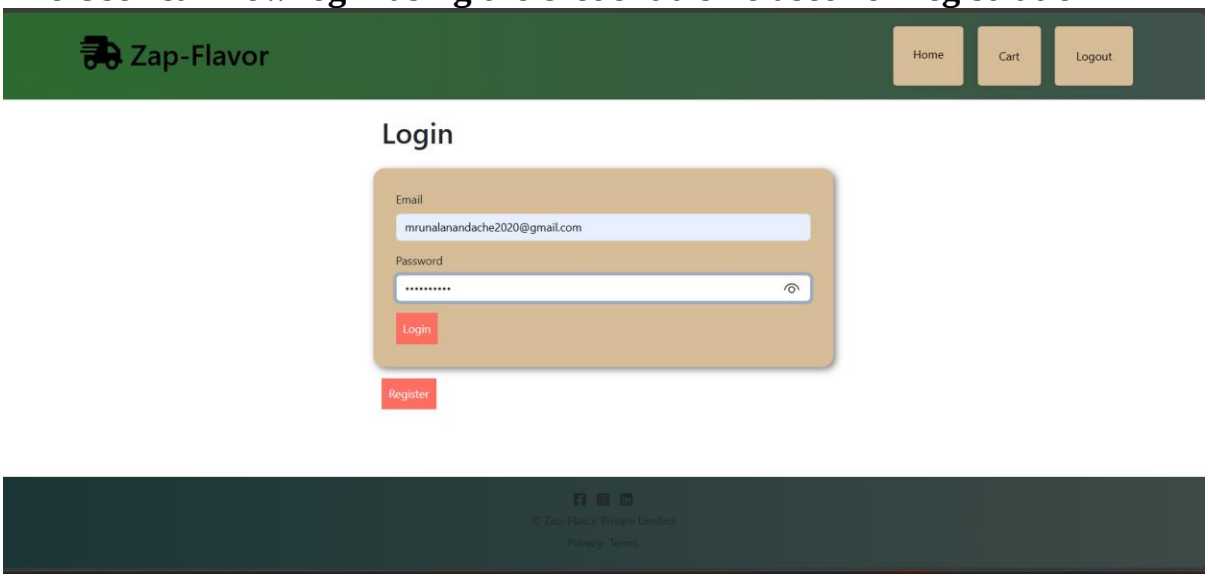
FINAL RESULTS (SCREENSHOTS):

- **From Users Point Of View(FRONTEND):**

Registration Page:
The User can Register using his/her credentials

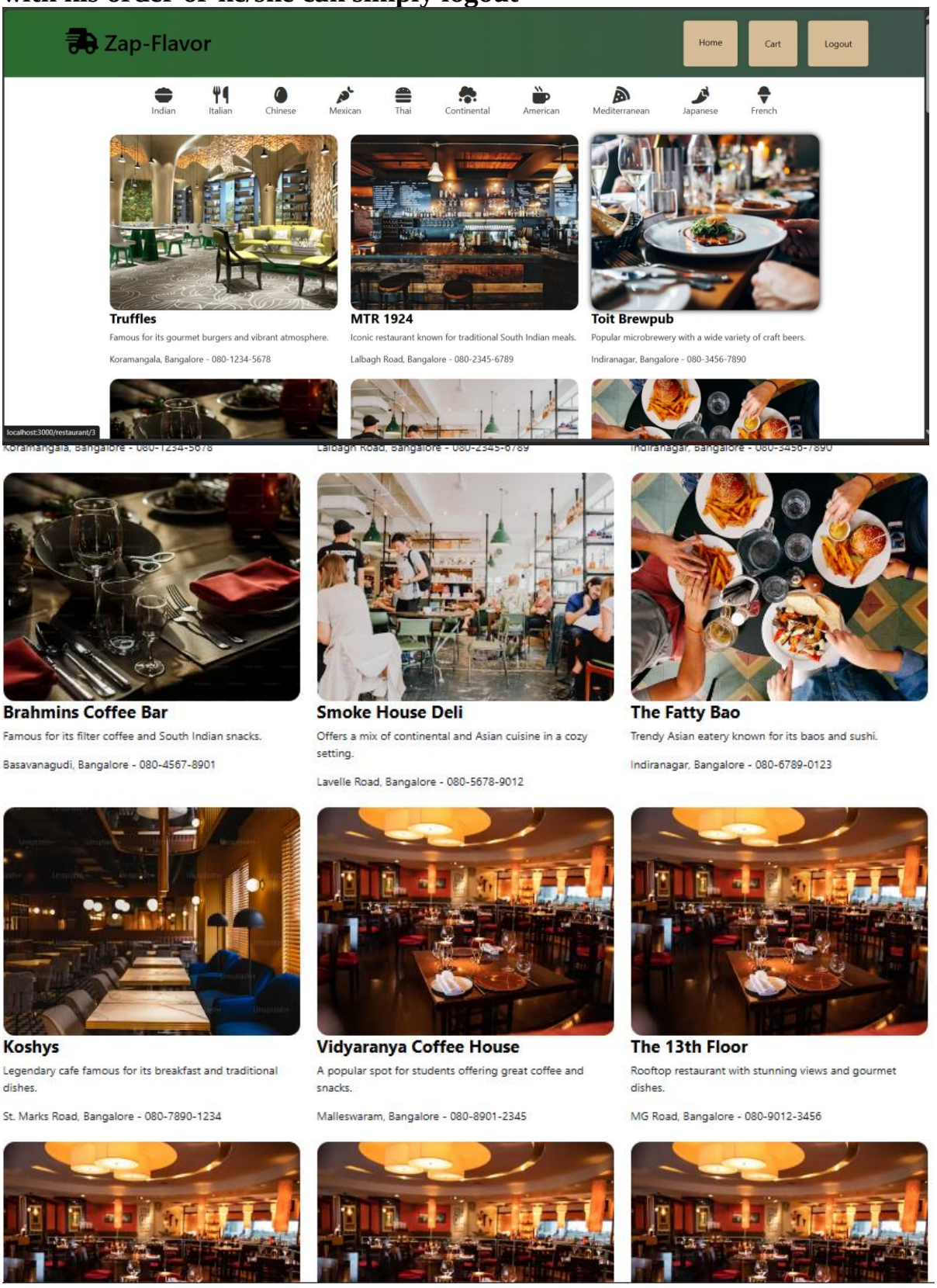


Login Page:
The User can now login using the credentials he used for registration



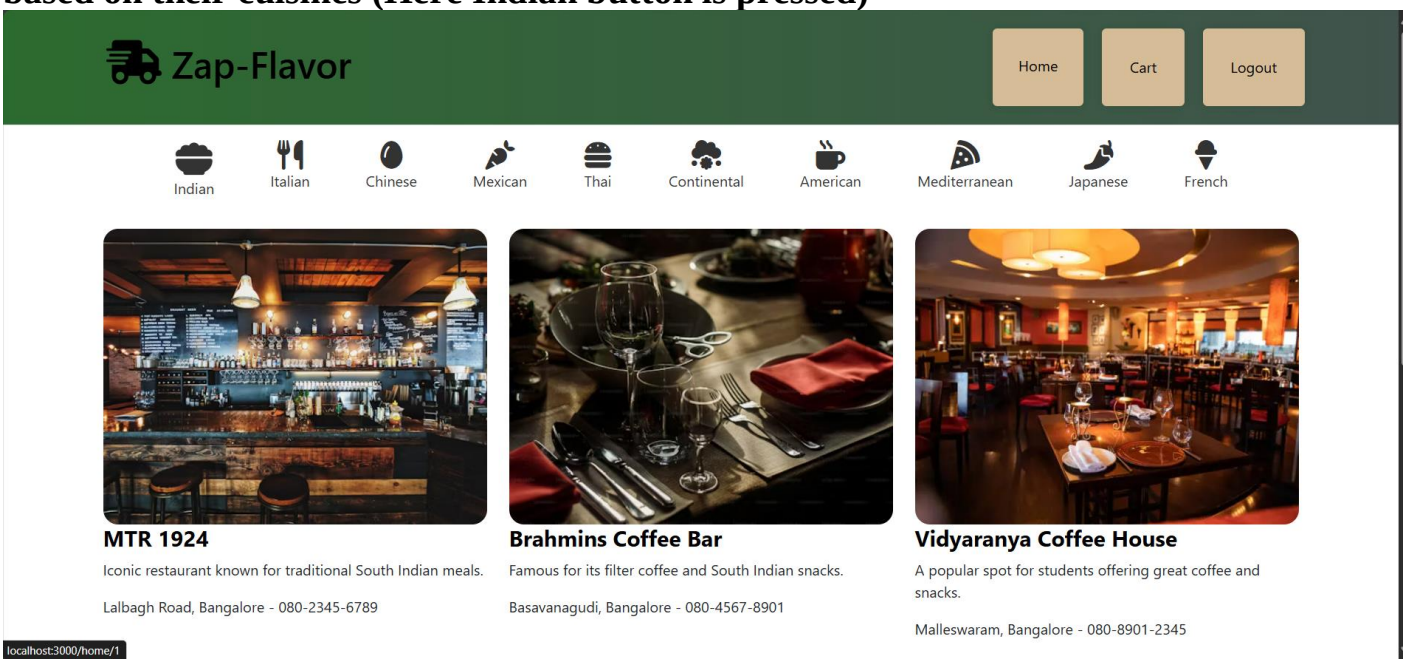
Home Page:

Once Logged In the user can view the home page with all the listings of restaurants, the user can click on any restaurant and can go the page where he/she can view the menu items available , also on the navbar he can directly jump to the cart section to proceed with his order or he/she can simply logout



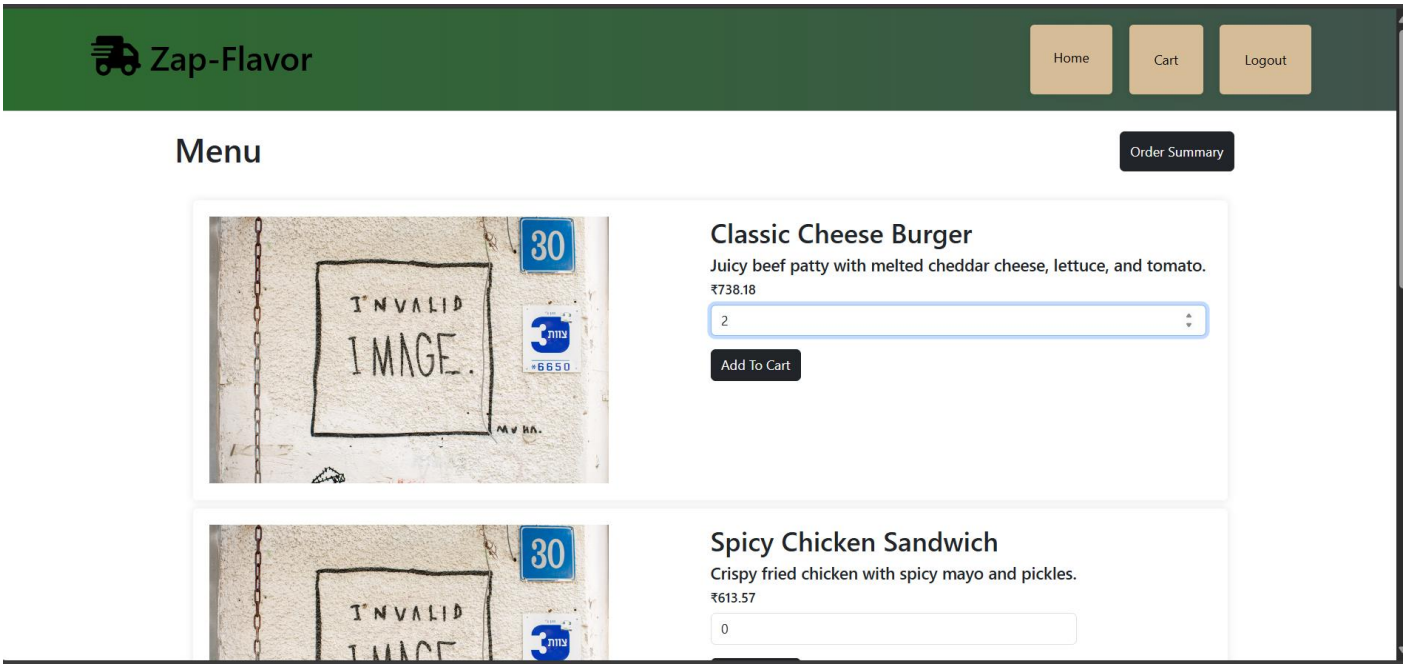
Cuisines:

The User can choose the cuisines shown just below the navbar to filter out restaurants based on their cuisines (Here Indian button is pressed)

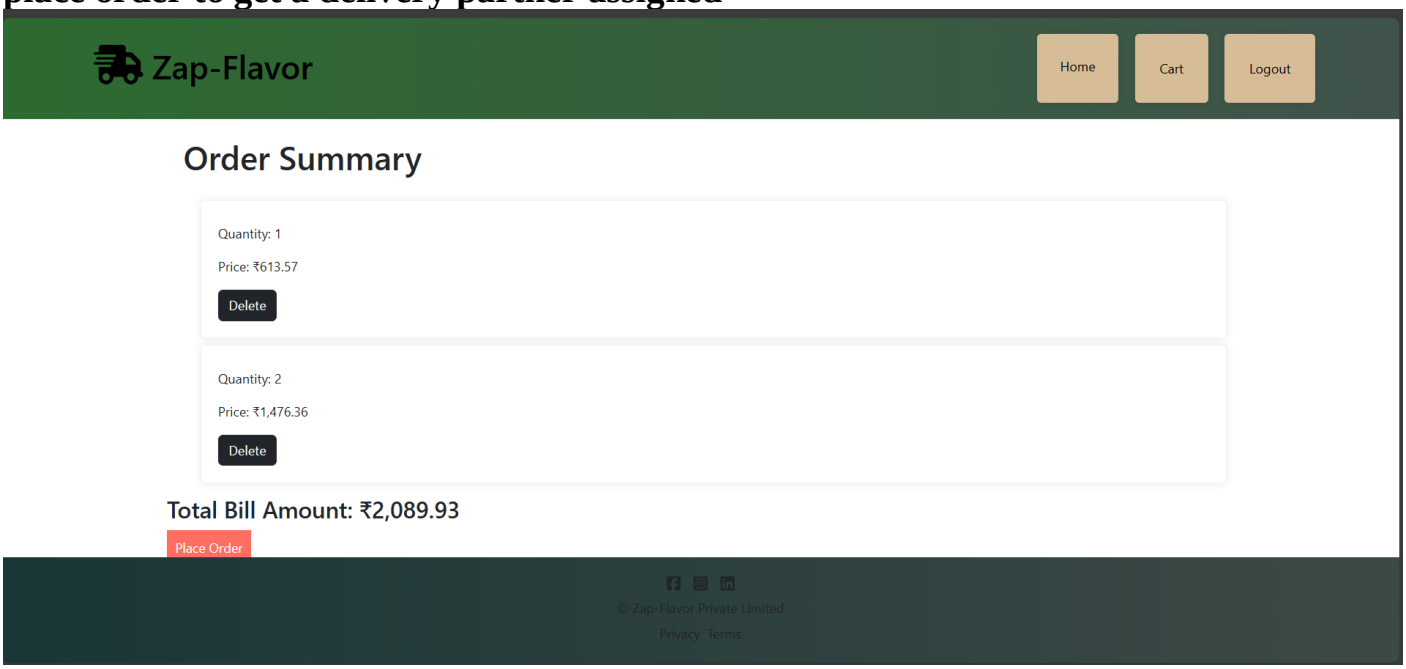


Menu Page:

On clicking on any of the Restaurants images , the user will be redirected to the menu page from where he/she can proceed with choosing the food items and then pushing them to cart



Order Summary Page:
After adding the food items to the cart , the user can jump to the order summary section by using the button from where he can view all his cart item and then press place order to get a delivery partner assigned

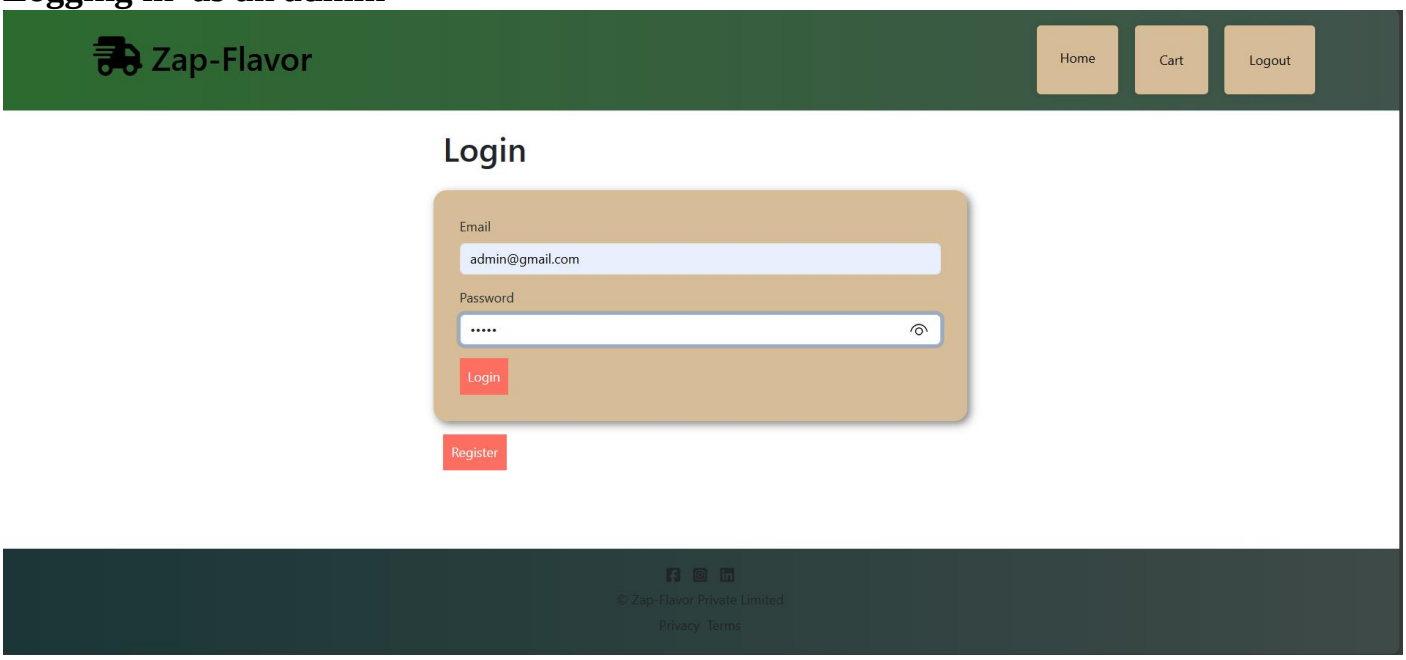


Order Confirmation Page:
Once placed order a delivery partner will be assigned with all his details displayed

- **From Admin's Point Of View(FRONTEND):**

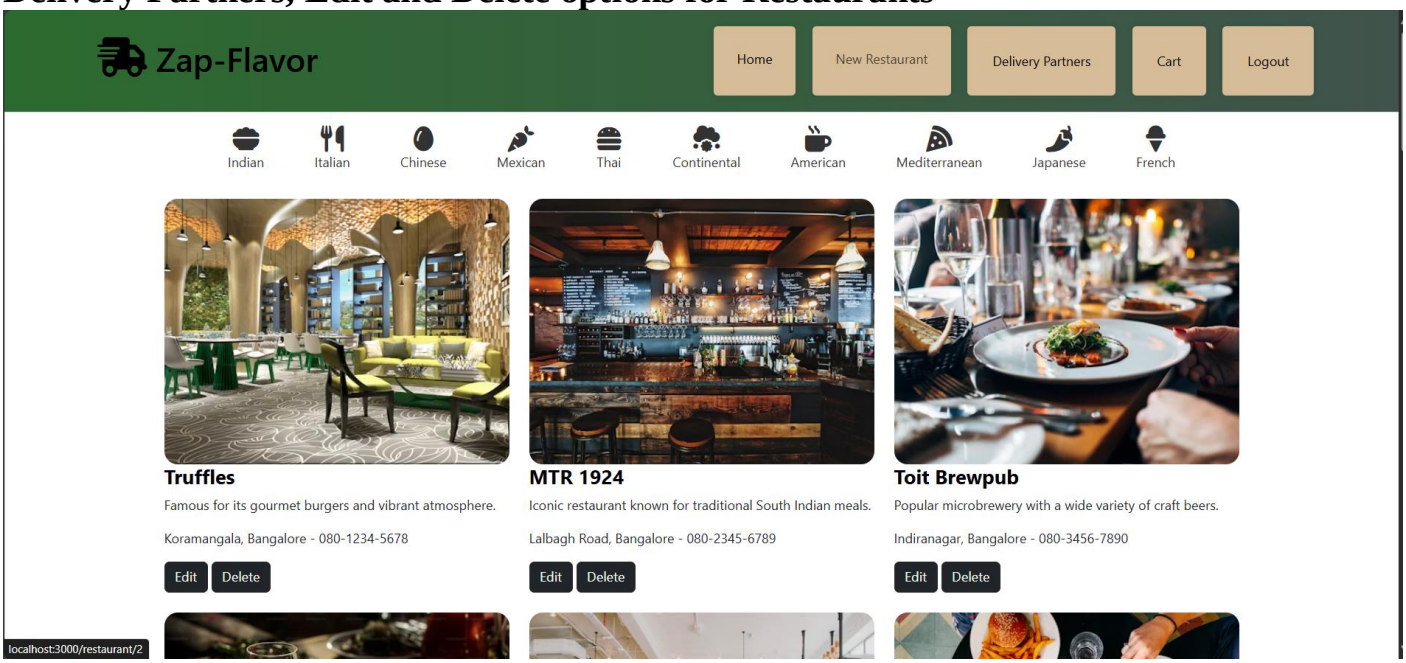
Login Page:

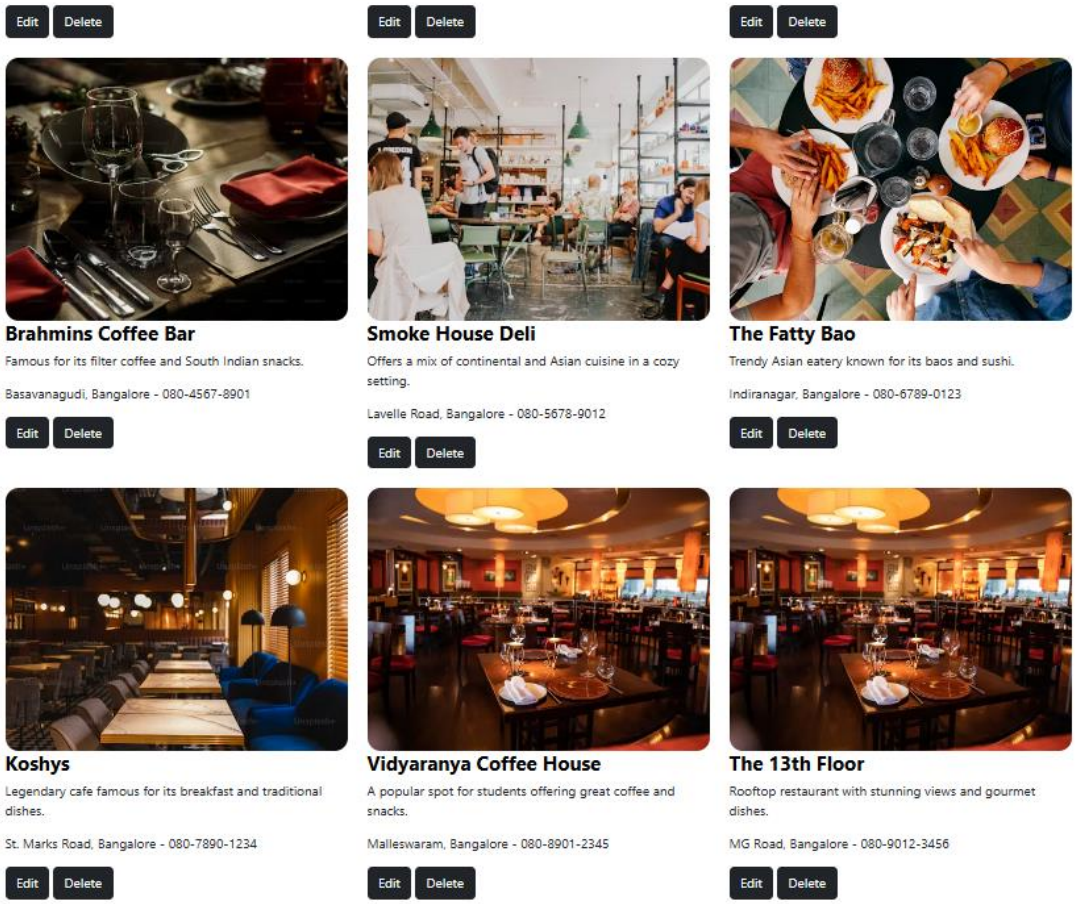
Logging in as an admin



Home Page:

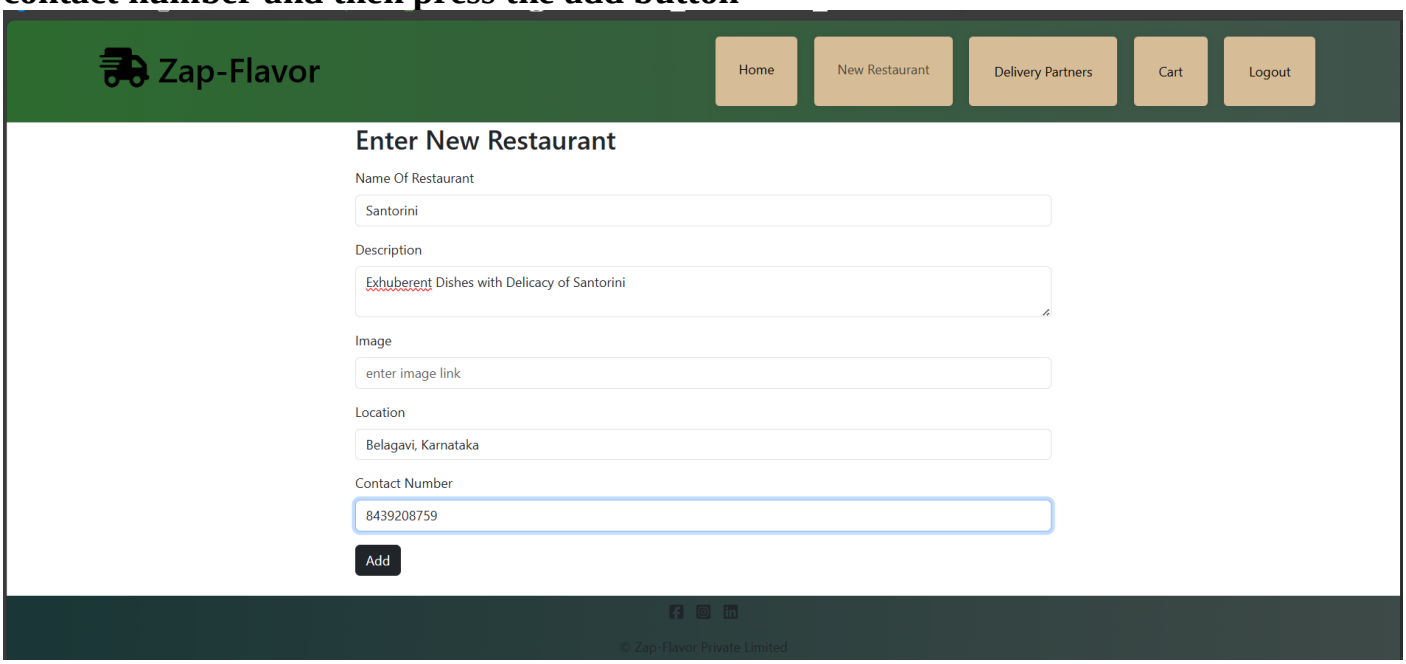
The Admin can view additional buttons on the home page like New Restaurant, Delivery Partners, Edit and Delete options for Restaurants



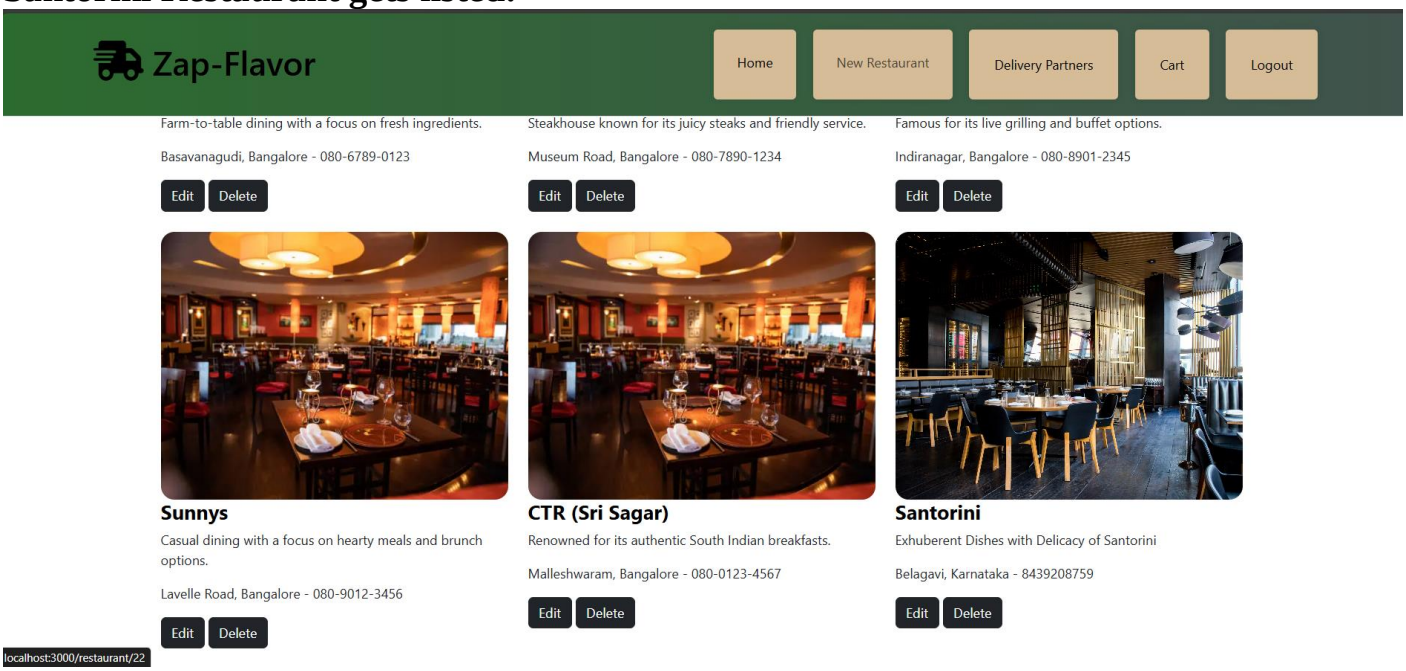


New Restaurant Creation Page:

The Admin has to Fill the form with all the necessary details like name of the restaurant, description, image link(if not then default image will be used), location and contact number and then press the add button

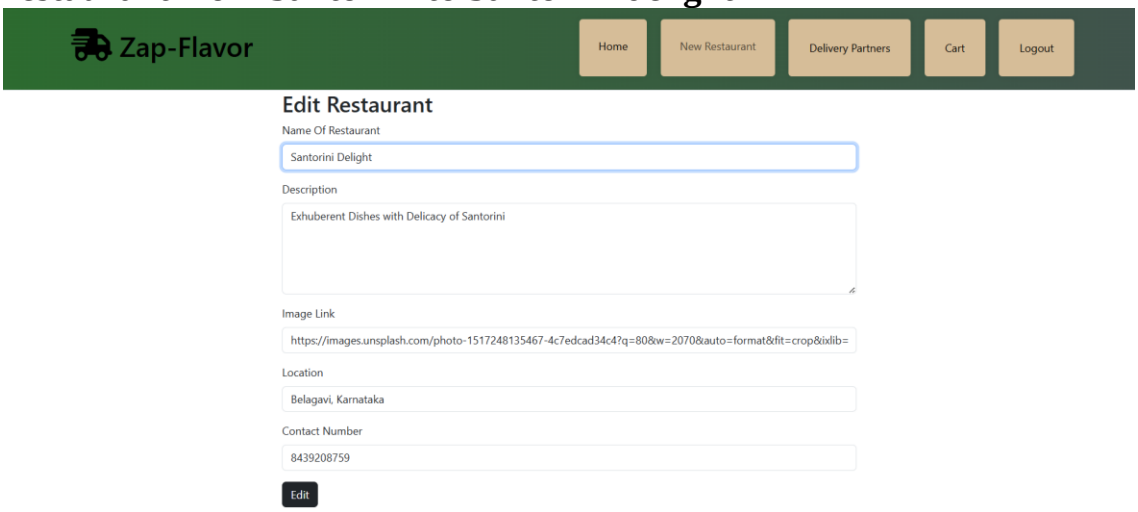


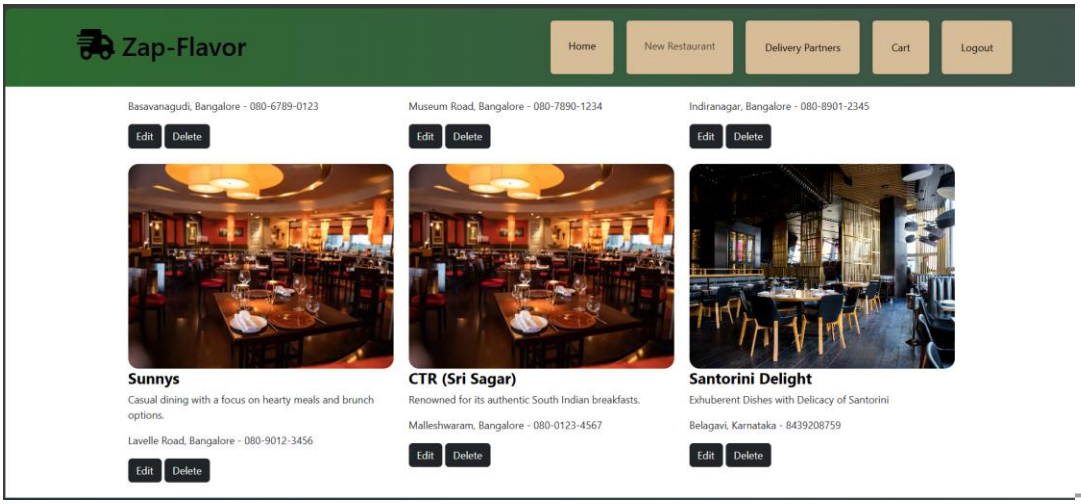
Santorini Restaurant gets listed:



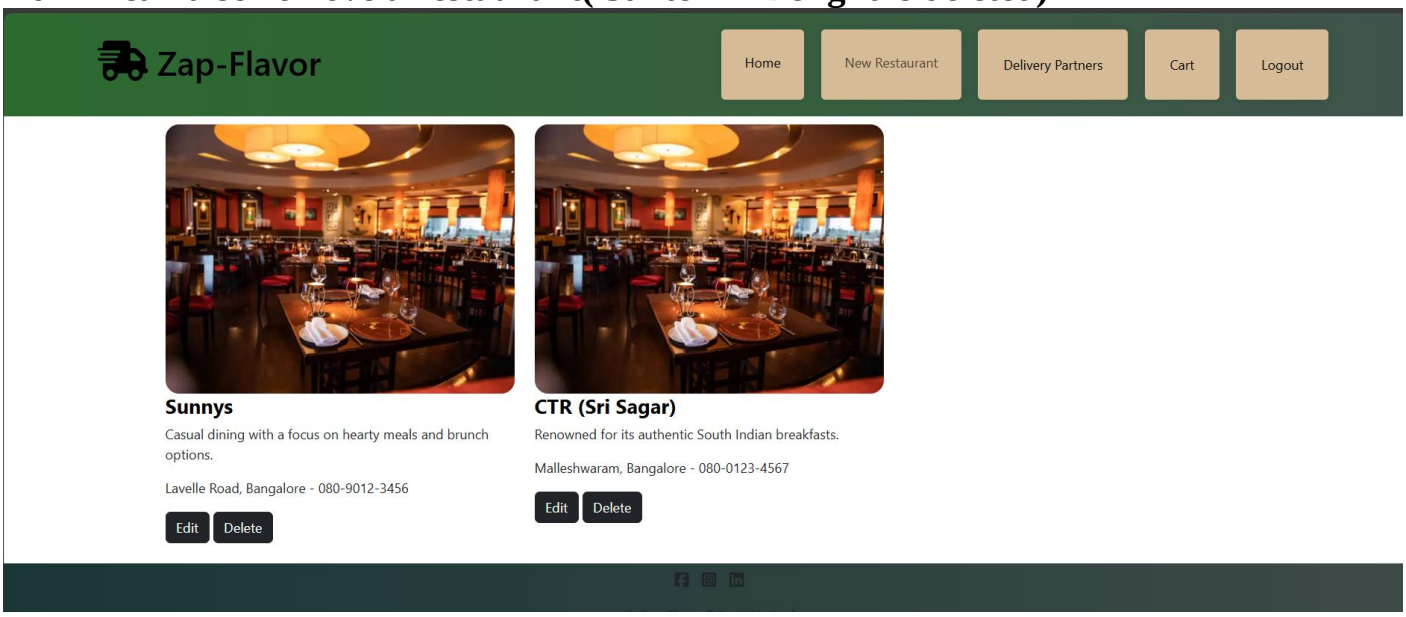
Edit Existing Restaurants Details:

The admin can click the edit button to edit the details lets say he edits the name of the restaurant from Santorini to Santorini delight

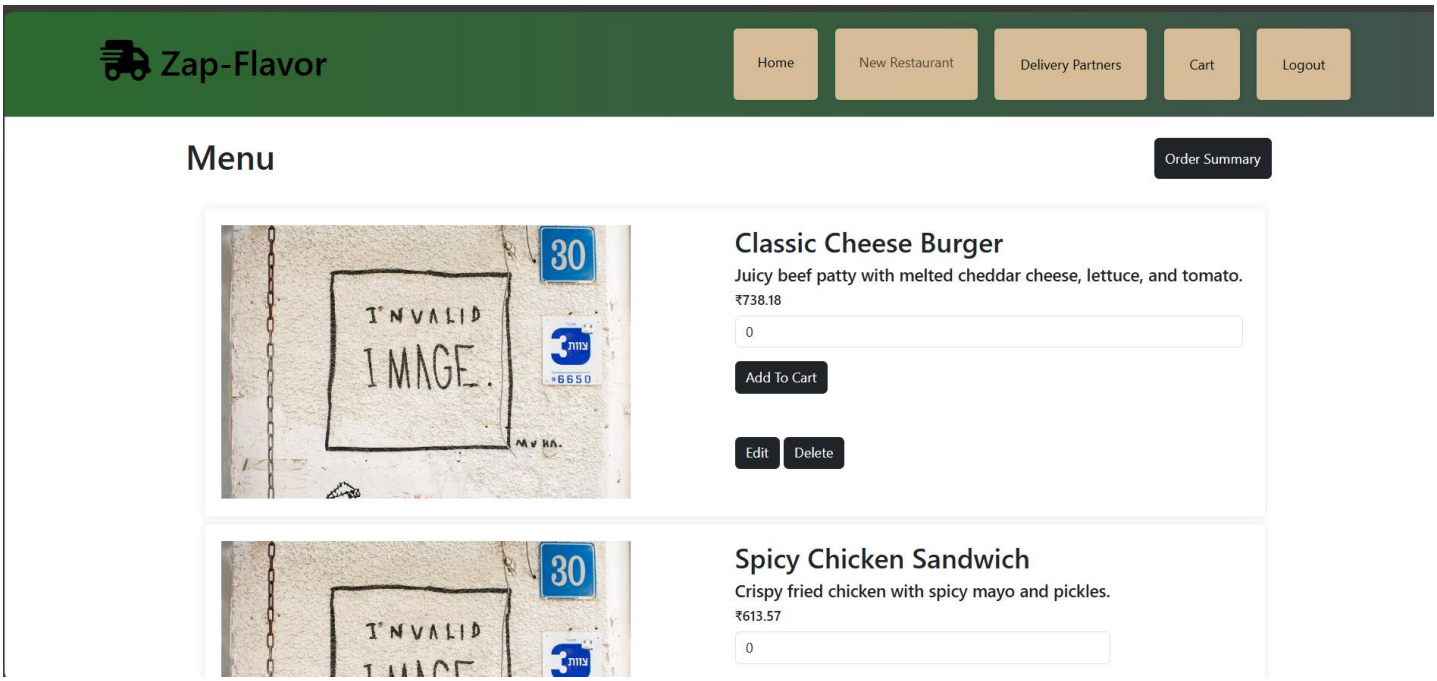




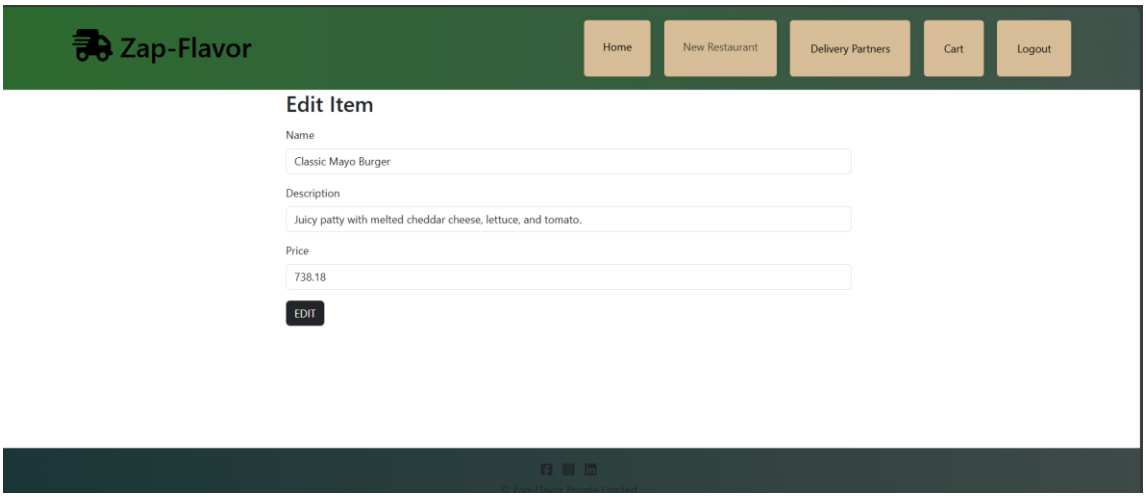
Delete Existing Restaurant:
Admin can also remove a restaurant(Santorini Delight is deleted)

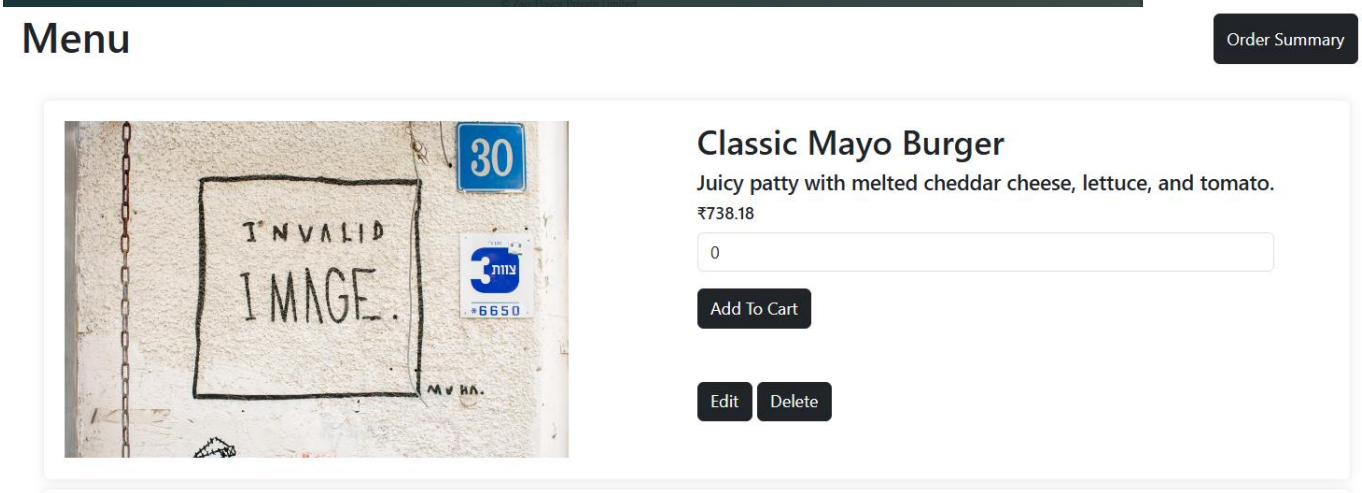


Menu Page:
There are two more buttons available to the admin to add and to delete a menu item

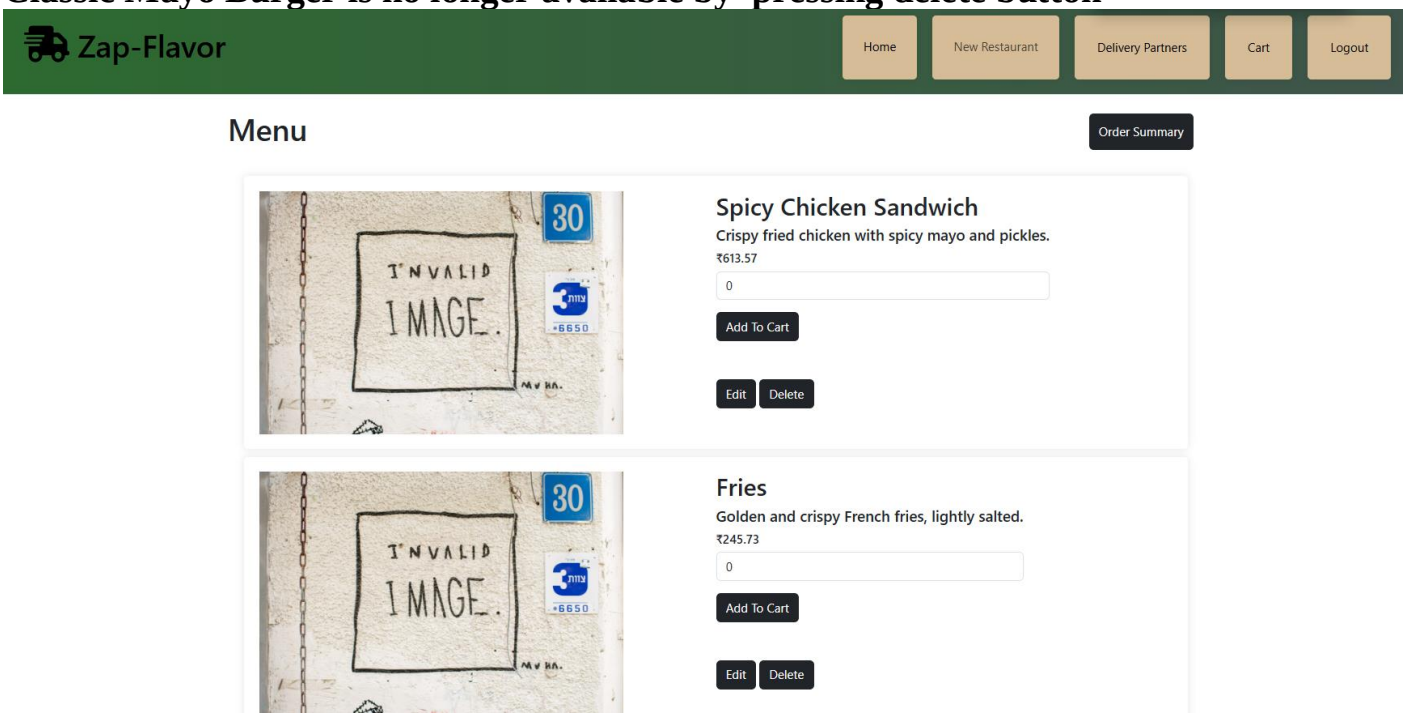


Menu item Edit page:
Edit the details of the food item listed here (classic cheese burger to classic mayo burger)

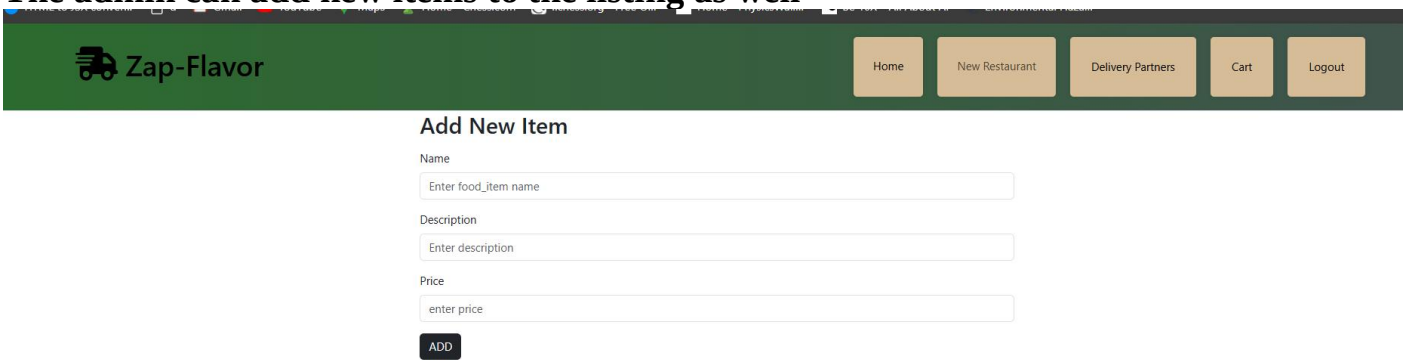




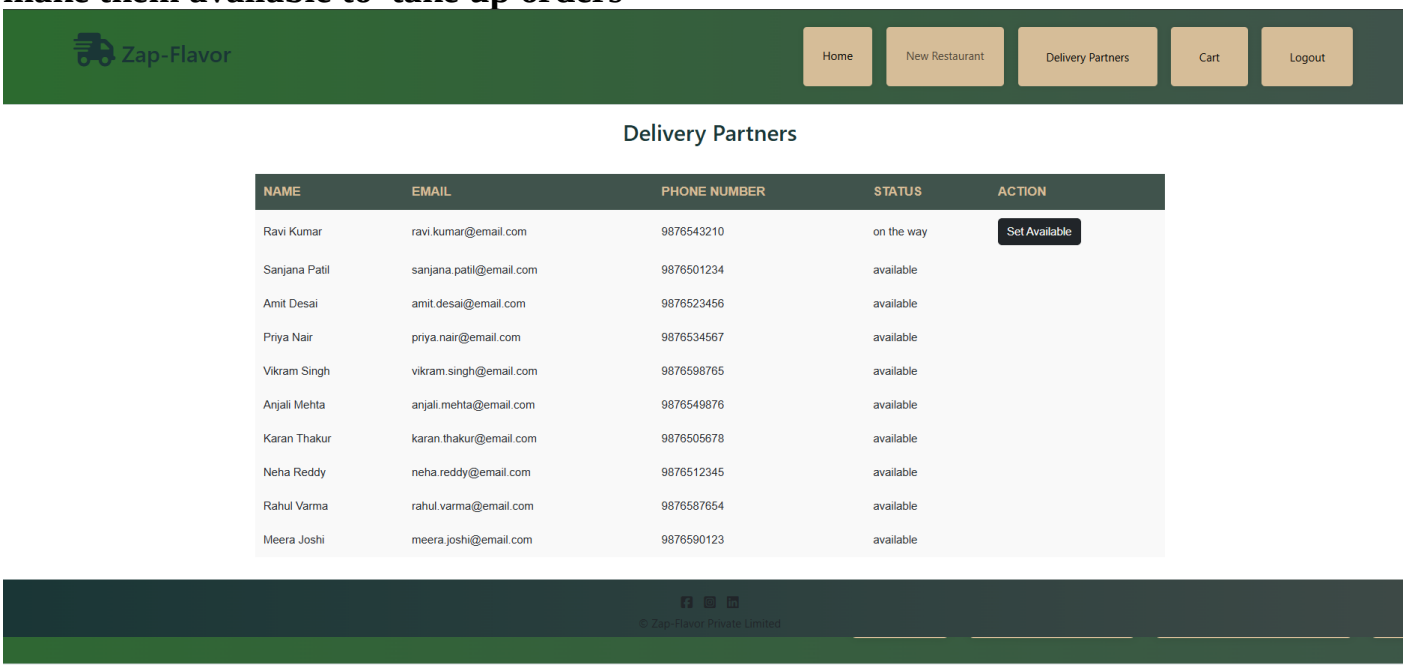
Deletion of Food item:
Classic Mayo Burger is no longer available by pressing delete button

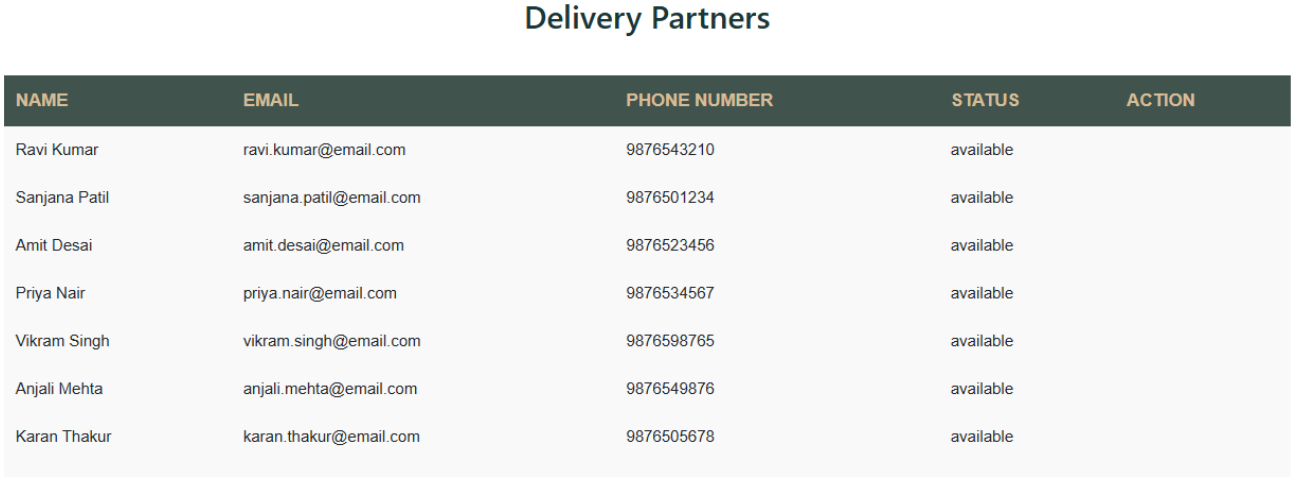


Add New Food item page:
The admin can add new items to the listing as well

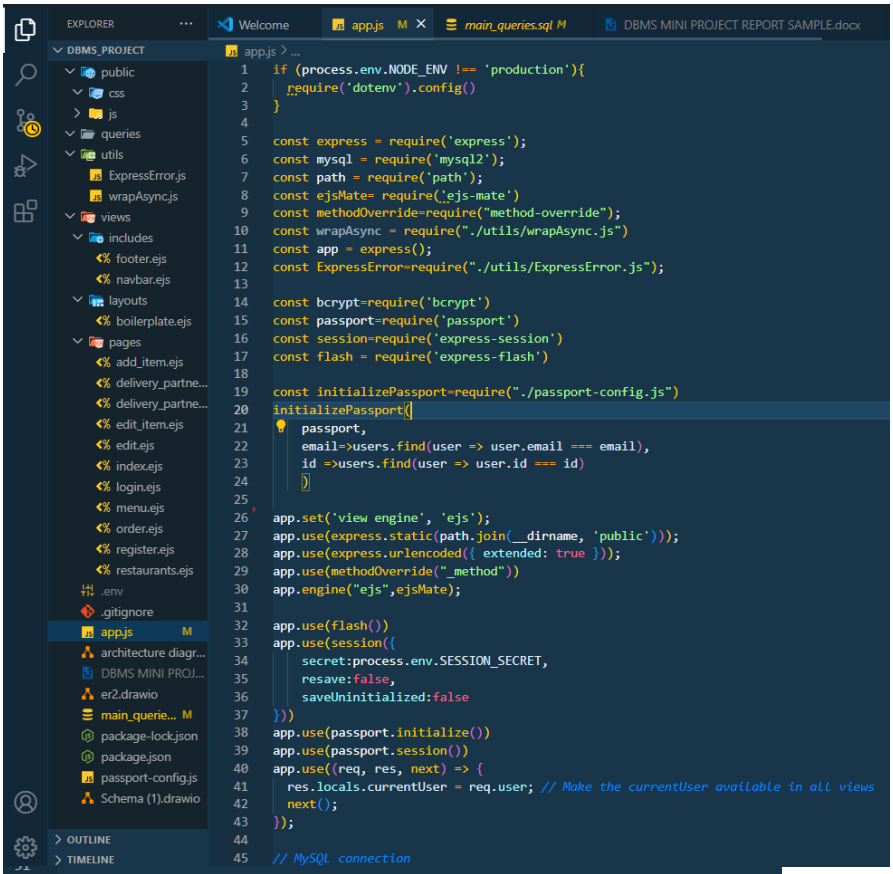


Delivery Partners Page:
By clicking the button on the navbar the delivery partners are listed and the admin can make them available to take up orders





BACKEND CODE:




```
52 // MySQL connection
53 const db = mysql.createConnection({
54   host: 'localhost',
55   user: 'root',
56   password: 'CodeMrunal2004',
57   database: 'food_ordering'
58 });
59
60 db.connect((err) => {
61   if (err) {
62     console.error('Database connection failed:', err);
63   } else {
64     console.log('Connected to the database.');
65   }
66 });
67
68 //Login Route
69 app.get("/",checkAuthenticated,(req,res)=>{
70   db.query('SELECT * FROM restaurants', (err, results) => {
71     if (err) throw err;
72     // console.log(results);
73     res.render('pages/index', { restaurants: results });
74   });
75 })
76
77 app.get("/login",checkNotAuthenticated,(req,res)=>{
78   res.render("pages/login.ejs")
79 })
80
81 app.get("/register",checkNotAuthenticated,(req,res)=>{
82   res.render("pages/register.ejs")
83 })
84
85 // Registration Route
86 app.post("/register", async (req, res) => {
87   const { name, email, password } = req.body;
88
89   try {
90     const hashedPassword = await bcrypt.hash(password, 10);
91     const insertUserQuery = 'INSERT INTO users (name, email, password) VALUES (?, ?, ?)';
92
93     db.query(insertUserQuery, [name, email, hashedPassword], (err, result) => {
94       if (err) {
95         console.log(err);
96         return res.redirect('/register');
```

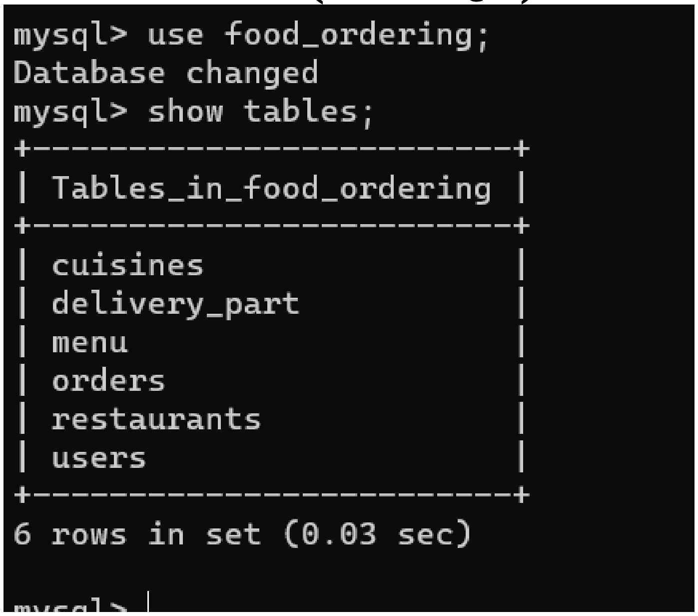
```
// Login Route (No change needed for post "/login" since passport handles it)
app.post("/login", passport.authenticate('local', {
  successRedirect: '/home',
  failureRedirect: '/login',
  failureFlash: true
}));


app.delete('/logout', (req, res, next) => {
  req.logout(err => {
      if (err) { return next(err); }
      res.redirect('/login');
  });
});


// Home Route
app.get("/home", checkAuthenticated, (req, res) => {
  db.query('SELECT * FROM restaurants', (err, results) => {
    if (err) throw err;
    res.render('pages/index', {
      restaurants: results,
      currentUser: req.user // Pass the user information to EJS
    });
  });
});


//New Restaurant Listing
app.get("/home/new",checkAuthenticated,(req,res)=>{
  res.render("pages/restaurants");
})

app.get("/home/:id", checkAuthenticated, (req, res) => {
  let { id } = req.params;
  const sel_cuisine = `
    SELECT restaurants.*, cuisines.name AS cuisine_name
    FROM restaurants
    INNER JOIN cuisines ON restaurants.cuisine_id = cuisines.cuisine_id
    WHERE restaurants.cuisine_id = ?;
  `;
```

```
app.post("/home/new", checkAuthenticated, (req, res) => {
  let { name, description, location, contact_no } = req.body;

  // Define the query to call the stored function
  const insert_query = `SELECT add_new_restaurant(?, ?, ?, ?) AS message`;

  // Execute the function
  db.query(insert_query, [name, description, location, contact_no], (err, results) => {
    if (err) {
      console.log(err);
    } else {
      console.log(results[0].message); // Output success message from the function
    }
    res.redirect("/home");
  });
});


//Edit Restaurant
app.get("/home/:id/edit",checkAuthenticated,(req,res)=>{
  let {id}=req.params;
  specific_rest=`Select * from restaurants where id=?`
  db.query(specific_rest,id,(err,result)=>{
    if(err){
      console.log(err);
    }
    else{
      console.log(result)
      res.render("pages/edit",{restaurant:result[0],currentUser: req.user});
    }
  })
})

app.put("/home/:id", checkAuthenticated, (req, res) => {
  let { id } = req.params;
  let { name, description, location, contact_no } = req.body;

  // Define the query to call the stored procedure
  const update_query = `CALL update_restaurant(?, ?, ?, ?, ?)`;

  // Execute the procedure
```

```
//Delete Restaurant
app.delete("/home/:id/delete",checkAuthenticated,(req,res)=>{
  let {id}=req.params;
  del_query=`DELETE FROM restaurants where id=?;`
  db.query(del_query,id,(err,result)=>{
    if(err){
      console.log(err);
    }else{
      // console.log(result);
      res.redirect("/home")
    }
  })
})

// Restaurant Menu
app.get('/restaurant/:id',checkAuthenticated, (req, res) => {
  const restaurantId = req.params.id;
  db.query('SELECT * FROM menu WHERE restaurant_id = ?', [restaurantId], (err, results) => {
    if (err) throw err;
    res.render('pages/menu', { menu: results, restaurantId });
  });
});

//add menu item
app.get("/home/:id/add_item",checkAuthenticated,(req,res)=>{
  // here id is restaurant id
  let {id}=req.params;
  res.render("pages/add_item",{restaurant_id:id})
})

app.post("/home/:id/add_item",checkAuthenticated,(req,res)=>{
  //here id is restaurant id
  let {id}=req.params;
  let {name,description,price}=req.body;
  insert_item_query=`INSERT INTO menu (restaurant_id,name,description,price) values (?,?,?,?)`;
  db.query(insert_item_query,[id,name,description,price],(err)=>{
    if(err){
      console.log(err);
    }else{
      res.redirect(`/restaurant/${id}`)
    }
  })
})
```

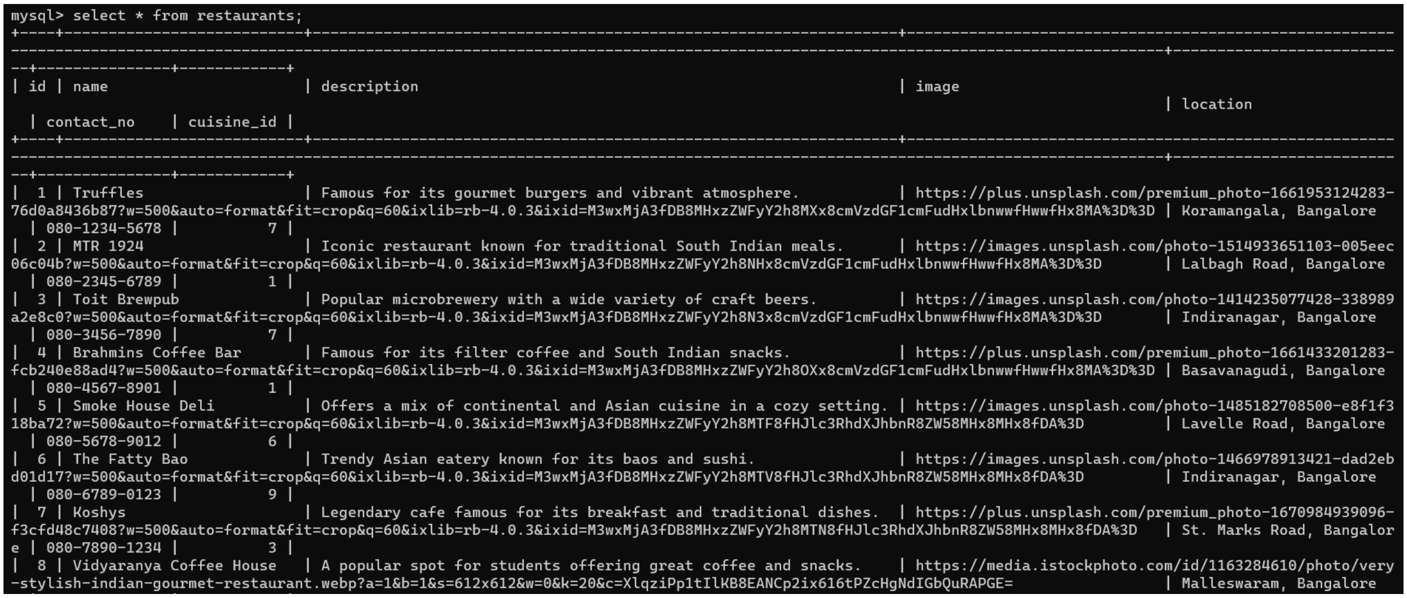
```

app.all("*",(req,res,next)=>{
  next(new ExpressError(404,"Page not found!"))
})

app.use((err,req,res,next)=>{
    let {statusCode=500,message="Something went wrong"}=err
    res.status(statusCode).send(message)

})

function checkAuthenticated(req, res, next) {
  if (req.isAuthenticated()) {
    return next();
  }
  res.redirect("/login");
}

function checkNotAuthenticated(req, res, next) {
  if (req.isAuthenticated()) {
    return res.redirect("/home");
  }
  next();
}


// Start the server
app.listen(3000, () => {
  console.log('Server started on http://localhost:3000');
});

```

DATABASE (MYSQL):

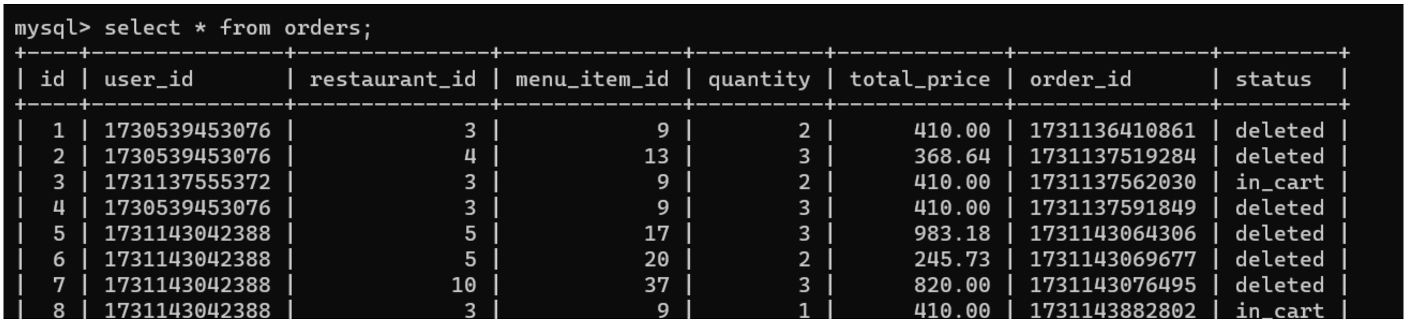


```
mysql> use food_ordering;
Database changed
mysql> show tables;
+--------------------------+
| Tables_in_food_ordering |
+--------------------------+
| cuisines                 |
| delivery_part            |
| menu                     |
| orders                   |
| restaurants              |
| users                    |
+--------------------------+
6 rows in set (0.03 sec)

mysql> 
```

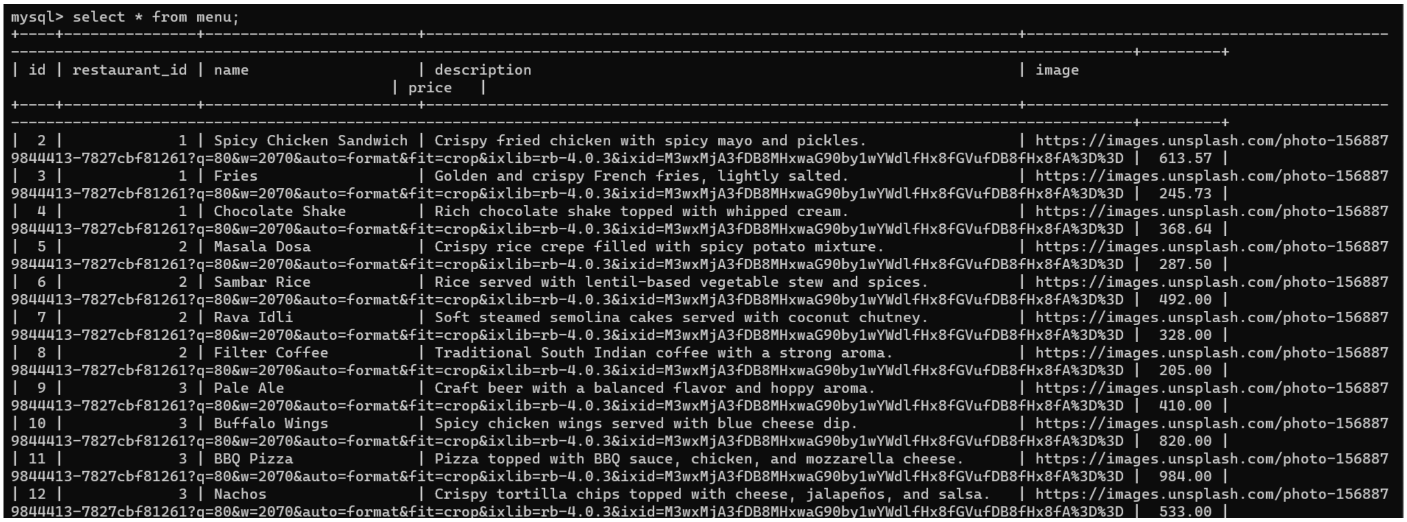
```
mysql> select * from users;
+----+------------------+-------------------------------+--------------------------------------------------------------+
| id | name             | email                         | password                                                     |
+----+------------------+-------------------------------+--------------------------------------------------------------+
|  1 | admin            | admin@gmail.com               | $2b$10$ocCdwccDJ78uKwb4OkWYkeGCLxhLexjwyWCzNLRlX0JdPZyFQ4Iui |
|  2 | Mrunal Anandache | mrunalanandache2020@gmail.com | $2b$10$XYi6tWu6Al9XF27UA7MlSuGgYGvOvdtOveACB3zq2B0YYZjCAEE5q |
+----+------------------+-------------------------------+--------------------------------------------------------------+
2 rows in set (0.00 sec)
```



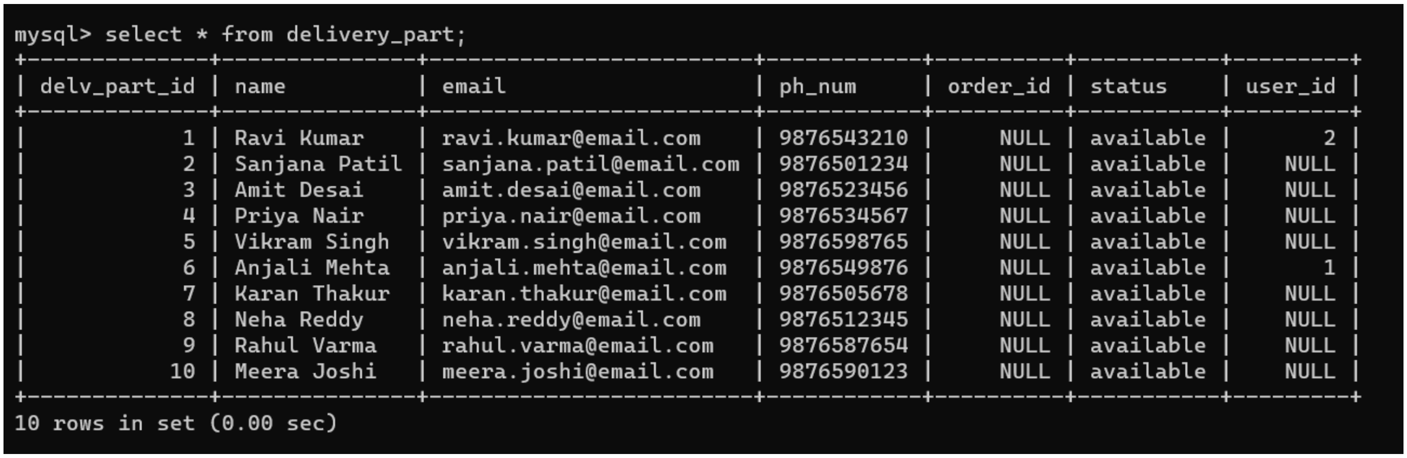
```
mysql> select * from restaurants;
+----+---------------------------+------------------------------------------------------------------+-------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------+-------------------------
--+--------------+------------+
| id | name                      | description                                                      | image
                                                                                                                                   | location
  | contact_no   | cuisine_id |
+----+---------------------------+------------------------------------------------------------------+-------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------+-------------------------
--+--------------+------------+
|  1 | Truffles                  | Famous for its gourmet burgers and vibrant atmosphere.           | https://plus.unsplash.com/premium_photo-1661953124283-
76d0a8436b87?w=500&auto=format&fit=crop&q=60&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxzZWFyY2h8MXx8cmVzdGF1cmFudHxlbnwwfHwwfHx8MA%3D%3D | Koramangala, Bangalore
  | 080-1234-5678 |          7 |
|  2 | MTR 1924                  | Iconic restaurant known for traditional South Indian meals.      | https://images.unsplash.com/photo-1514933651103-005eec
06c04b?w=500&auto=format&fit=crop&q=60&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxzZWFyY2h8NHx8cmVzdGF1cmFudHxlbnwwfHwwfHx8MA%3D%3D       | Lalbagh Road, Bangalore
  | 080-2345-6789 |          1 |
|  3 | Toit Brewpub              | Popular microbrewery with a wide variety of craft beers.         | https://images.unsplash.com/photo-1414235077428-338989
a2e8c0?w=500&auto=format&fit=crop&q=60&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxzZWFyY2h8N3x8cmVzdGF1cmFudHxlbnwwfHwwfHx8MA%3D%3D       | Indiranagar, Bangalore
  | 080-3456-7890 |          7 |
|  4 | Brahmins Coffee Bar       | Famous for its filter coffee and South Indian snacks.            | https://plus.unsplash.com/premium_photo-1661433201283-
fcb240e88ad4?w=500&auto=format&fit=crop&q=60&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxzZWFyY2h8OXx8cmVzdGF1cmFudHxlbnwwfHwwfHx8MA%3D%3D | Basavanagudi, Bangalore
  | 080-4567-8901 |          1 |
|  5 | Smoke House Deli          | Offers a mix of continental and Asian cuisine in a cozy setting. | https://images.unsplash.com/photo-1485182708500-e8f1f3
18ba72?w=500&auto=format&fit=crop&q=60&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxzZWFyY2h8MTF8fHJlc3RhdXJhbnR8ZW58MHx8MHx8fDA%3D         | Lavelle Road, Bangalore
  | 080-5678-9012 |          6 |
|  6 | The Fatty Bao             | Trendy Asian eatery known for its baos and sushi.                | https://images.unsplash.com/photo-1466978913421-dad2eb
d01d17?w=500&auto=format&fit=crop&q=60&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxzZWFyY2h8MTV8fHJlc3RhdXJhbnR8ZW58MHx8MHx8fDA%3D         | Indiranagar, Bangalore
  | 080-6789-0123 |          9 |
|  7 | Koshys                    | Legendary cafe famous for its breakfast and traditional dishes.  | https://plus.unsplash.com/premium_photo-1670984939096-
f3cfd48c7408?w=500&auto=format&fit=crop&q=60&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxzZWFyY2h8MTN8fHJlc3RhdXJhbnR8ZW58MHx8MHx8fDA%3D   | St. Marks Road, Bangalor
e | 080-7890-1234 |          3 |
|  8 | Vidyaranya Coffee House   | A popular spot for students offering great coffee and snacks.    | https://media.istockphoto.com/id/1163284610/photo/very
-stylish-indian-gourmet-restaurant.webp?a=1&b=1&s=612x612&w=0&k=20&c=XlqziPp1tIlKB8EANCp2ix616tPZcHgNdIGbQuRAPGE=                  | Malleswaram, Bangalore
```



```
mysql> select * from orders;
+----+---------------+---------------+--------------+----------+-------------+---------------+---------+
| id | user_id       | restaurant_id | menu_item_id | quantity | total_price | order_id      | status  |
+----+---------------+---------------+--------------+----------+-------------+---------------+---------+
|  1 | 1730539453076 |             3 |            9 |        2 |      410.00 | 1731136410861 | deleted |
|  2 | 1730539453076 |             4 |           13 |        3 |      368.64 | 1731137519284 | deleted |
|  3 | 1731137555372 |             3 |            9 |        2 |      410.00 | 1731137562030 | in_cart |
|  4 | 1730539453076 |             3 |            9 |        3 |      410.00 | 1731137591849 | deleted |
|  5 | 1731143042388 |             5 |           17 |        3 |      983.18 | 1731143064306 | deleted |
|  6 | 1731143042388 |             5 |           20 |        2 |      245.73 | 1731143069677 | deleted |
|  7 | 1731143042388 |            10 |           37 |        3 |      820.00 | 1731143076495 | deleted |
|  8 | 1731143042388 |             3 |            9 |        1 |      410.00 | 1731143882802 | in_cart |
```



```
mysql> select * from menu;
+----+---------------+------------------------+-------------------------------------------------------------------+-----------------------------------------
----------------------------------------------------------------------------------------------------------------------------------+---------+
| id | restaurant_id | name                   | description                                                       | image
                                                                                                                                  | price   |
+----+---------------+------------------------+-------------------------------------------------------------------+-----------------------------------------
----------------------------------------------------------------------------------------------------------------------------------+---------+
|  2 |             1 | Spicy Chicken Sandwich | Crispy fried chicken with spicy mayo and pickles.                 | https://images.unsplash.com/photo-156887
9844413-7827cbf81261?q=80&w=2070&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D |  613.57 |
|  3 |             1 | Fries                  | Golden and crispy French fries, lightly salted.                   | https://images.unsplash.com/photo-156887
9844413-7827cbf81261?q=80&w=2070&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D |  245.73 |
|  4 |             1 | Chocolate Shake        | Rich chocolate shake topped with whipped cream.                   | https://images.unsplash.com/photo-156887
9844413-7827cbf81261?q=80&w=2070&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D |  368.64 |
|  5 |             2 | Masala Dosa            | Crispy rice crepe filled with spicy potato mixture.               | https://images.unsplash.com/photo-156887
9844413-7827cbf81261?q=80&w=2070&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D |  287.50 |
|  6 |             2 | Sambar Rice            | Rice served with lentil-based vegetable stew and spices.          | https://images.unsplash.com/photo-156887
9844413-7827cbf81261?q=80&w=2070&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D |  492.00 |
|  7 |             2 | Rava Idli              | Soft steamed semolina cakes served with coconut chutney.          | https://images.unsplash.com/photo-156887
9844413-7827cbf81261?q=80&w=2070&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D |  328.00 |
|  8 |             2 | Filter Coffee          | Traditional South Indian coffee with a strong aroma.              | https://images.unsplash.com/photo-156887
9844413-7827cbf81261?q=80&w=2070&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D |  205.00 |
|  9 |             3 | Pale Ale               | Craft beer with a balanced flavor and hoppy aroma.                | https://images.unsplash.com/photo-156887
9844413-7827cbf81261?q=80&w=2070&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D |  410.00 |
| 10 |             3 | Buffalo Wings          | Spicy chicken wings served with blue cheese dip.                  | https://images.unsplash.com/photo-156887
9844413-7827cbf81261?q=80&w=2070&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D |  820.00 |
| 11 |             3 | BBQ Pizza              | Pizza topped with BBQ sauce, chicken, and mozzarella cheese.      | https://images.unsplash.com/photo-156887
9844413-7827cbf81261?q=80&w=2070&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D |  984.00 |
| 12 |             3 | Nachos                 | Crispy tortilla chips topped with cheese, jalapeños, and salsa.   | https://images.unsplash.com/photo-156887
9844413-7827cbf81261?q=80&w=2070&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D |  533.00 |
```



```
mysql> select * from delivery_part;
+--------------+---------------+-------------------------+------------+----------+-----------+---------+
| delv_part_id | name          | email                   | ph_num     | order_id | status    | user_id |
+--------------+---------------+-------------------------+------------+----------+-----------+---------+
|            1 | Ravi Kumar    | ravi.kumar@email.com    | 9876543210 |     NULL | available |       2 |
|            2 | Sanjana Patil | sanjana.patil@email.com | 9876501234 |     NULL | available |    NULL |
|            3 | Amit Desai    | amit.desai@email.com    | 9876523456 |     NULL | available |    NULL |
|            4 | Priya Nair    | priya.nair@email.com    | 9876534567 |     NULL | available |    NULL |
|            5 | Vikram Singh  | vikram.singh@email.com  | 9876598765 |     NULL | available |    NULL |
|            6 | Anjali Mehta  | anjali.mehta@email.com  | 9876549876 |     NULL | available |       1 |
|            7 | Karan Thakur  | karan.thakur@email.com  | 9876505678 |     NULL | available |    NULL |
|            8 | Neha Reddy    | neha.reddy@email.com    | 9876512345 |     NULL | available |    NULL |
|            9 | Rahul Varma   | rahul.varma@email.com   | 9876587654 |     NULL | available |    NULL |
|           10 | Meera Joshi   | meera.joshi@email.com   | 9876590123 |     NULL | available |    NULL |
+--------------+---------------+-------------------------+------------+----------+-----------+---------+
10 rows in set (0.00 sec)
```

```
mysql> select * from cuisines;
+------------+---------------+
| cuisine_id | name          |
+------------+---------------+
|          1 | Indian        |
|          2 | Italian       |
|          3 | Chinese       |
|          4 | Mexican       |
|          5 | Thai          |
|          6 | Continental   |
|          7 | American      |
|          8 | Mediterranean |
|          9 | Japanese      |
|         10 | French        |
+------------+---------------+
10 rows in set (0.00 sec)
```