**DBMS PROJECT**

**RESTAURANT MANAGEMENT SYSTEM**

**FLAVOURS AND FABLES**

**Team members:**

**NIKKATH ZABEEN – 2023BCS0060**

**S S PAVAN CHARAN – 2023BCS0078**

**HARSHITHA – 2023BCS0069**

**ROSHITHA – 2023BCS0105**

**MAMALESH – 2023BCS0081**

**ROHITH – 2023BCY0030**

**TASK: To provide a user-friendly platform for a Restaurant to manage orders, Table-reservations, cart-items, user-Authentication and manage Employee-Authentication and Employee task.**

**Abstraction:**

This platform is a comprehensive solution for restaurant management, designed to integrate multiple key functions into a single, user-friendly interface. It addresses the operational needs of both customers and employees, providing seamless management of orders, table reservations, cart items, and secure authentication for both users and staff. The system enhances efficiency, reduces manual errors, and creates a smooth dining experience through the following features:

1. **Order Management:**

* Allows restaurant staff to efficiently handle and track customer orders in real-time.
* Includes features for marking orders as completed and updating the order status instantly.
* Offers employees an easy-to-navigate interface for managing active and completed orders, reducing delays and errors.

1. **Table Reservation System:**

* Enables customers to reserve tables ahead of time through an intuitive booking system.
* Allows restaurant staff to view, manage, and update reservations, ensuring proper table allocation and smooth customer seating.
* Integrates with the order management system, ensuring orders are linked with specific table reservations.

1. **Cart System:**

* Provides customers with an easy-to-use cart interface for adding items and making final submissions.
* Enhances user experience by offering real-time updates to cart contents and order summaries.

1. **User Authentication:**

* Ensures secure customer login and registration using robust authentication mechanisms.
* Generates unique sessions for each user, safeguarding personal data and providing a personalized experience.
* Supports user data retention, allowing returning customers to view past orders, preferences, and active reservations.

1. **Employee Authentication and Task Management:**

* Implements secure employee login and role-based access to the platform.
* Employees can manage tasks such as processing orders and updating reservation statuses.
* Ensures only authorized staff can access sensitive operations, protecting the integrity of the platform.

1. **Efficiency and Automation:**

* The platform automates routine tasks such as updating order statuses, managing reservations, and organizing employee workflows.
* By reducing manual input, it minimizes human errors and optimizes the time required to serve customers, resulting in faster service.
* All features are integrated into a single, cohesive system, ensuring smooth communication between customers, staff, and management.

**Objective:**

The main objective of this platform is to create an efficient and user-friendly system that simplifies restaurant management and enhances both customer and staff experiences. It aims to automate critical tasks such as order processing, table reservations, cart management, and secure authentication while optimizing staff workflow. The extended objectives of this platform include:

1. **Streamline Order Management:**

* Provide a centralized interface for restaurant staff to manage all customer orders efficiently.
* Ensure real-time updates on order status, allowing staff to track the progress of orders and handle them in a timely manner.
* Enable easy completion of orders, reducing confusion in busy kitchen and service areas.

1. **Enhance Table Reservation Process:**

* Allow customers to reserve tables directly through the platform, reducing the need for manual reservation processes.
* Automatically link table reservations with orders, ensuring that customers receive the appropriate service upon arrival.

1. **Improve Cart and Checkout System:**

* Provide customers with a simple, intuitive cart interface that allows them to browse the menu, add items, and place orders with ease.
* Allow customers to modify their orders before checkout, improving user satisfaction by offering flexibility.

1. **Strengthen User Authentication:**

* Automatically generate unique user sessions upon login, providing a personalized experience with access to previous orders and preferences.
* Offer an easy registration process to encourage new users to sign up, while maintaining high levels of security.

1. **Secure Employee Authentication and Role Management:**

* Secure employee login with individual credentials, safeguarding sensitive restaurant data from unauthorized access.
* Allow employees to view and manage their tasks easily, streamlining their workflow and improving task accountability

1. **Foster Scalability and Flexibility:**

* Design the platform to be scalable, ensuring it can handle increased customer and employee traffic as the restaurant grows.
* Ensure flexibility in system features so that the platform can be adapted to different types of restaurants, from small cafes to larger establishments.
* Support future expansions such as adding delivery services, loyalty programs, or advanced analytics without disrupting current operations

**Introduction**

Managing the operations of a restaurant requires seamless coordination between various functions such as order management, table reservations, customer service, and employee task assignments. In a fast-paced restaurant environment, manually handling these tasks can lead to inefficiencies, errors, and customer dissatisfaction. To address these challenges, this platform offers a comprehensive solution that integrates order processing, table reservations, cart management, and secure user and employee authentication into one unified system.

The platform aims to optimize restaurant workflows by automating routine tasks, improving communication between the kitchen and service staff, and enhancing the overall customer experience. By incorporating features such as table booking, cart functionality, and role-based access for employees, the system ensures that restaurants can efficiently manage operations even during peak hours. Ultimately, this platform is designed to streamline restaurant management processes, reduce manual workloads, and improve customer satisfaction.

**Methodology:**

The development of this platform follows a structured approach to

ensure that all functional requirements are met while maintaining an intuitive user experience. The methodology is divided into the following stages:

1. **Requirements Gathering and Analysis**:

* A detailed analysis of restaurant management needs was focused, focusing on key areas such as order handling, table reservations, customer and employee authentication, and task assignments.
* Specific use cases were developed to outline the system’s functionality, including scenarios for order management, reservation handling, and user authentication.

1. **System Design**:

* The platform was designed using a modular approach, allowing each feature (orders, reservations, cart, authentication, etc.) to function independently yet integrate seamlessly.
* A user-friendly interface was prioritized, ensuring both employees and customers can interact with the system effortlessly.
* Database design involved creating a relational schema that securely stores user, employee, order, and reservation data. Efficient indexing and query optimization were implemented to ensure fast response times, even under high traffic.

1. **Development**:

* The platform was developed using a combination of **Node.js** for the backend and **Express** to handle server-side operations. **MySQL** was used as the database to manage all records efficiently.
* **EJS** templates were used to render the frontend, ensuring dynamic content generation, while **Bootstrap** was employed to create a responsive, mobile-friendly interface.
* A robust authentication system was implemented, including role-based access for employees and secure session management for both users and employees.
* Real-time features, such as order status updates and table availability tracking, were developed to provide immediate feedback to both customers and staff.

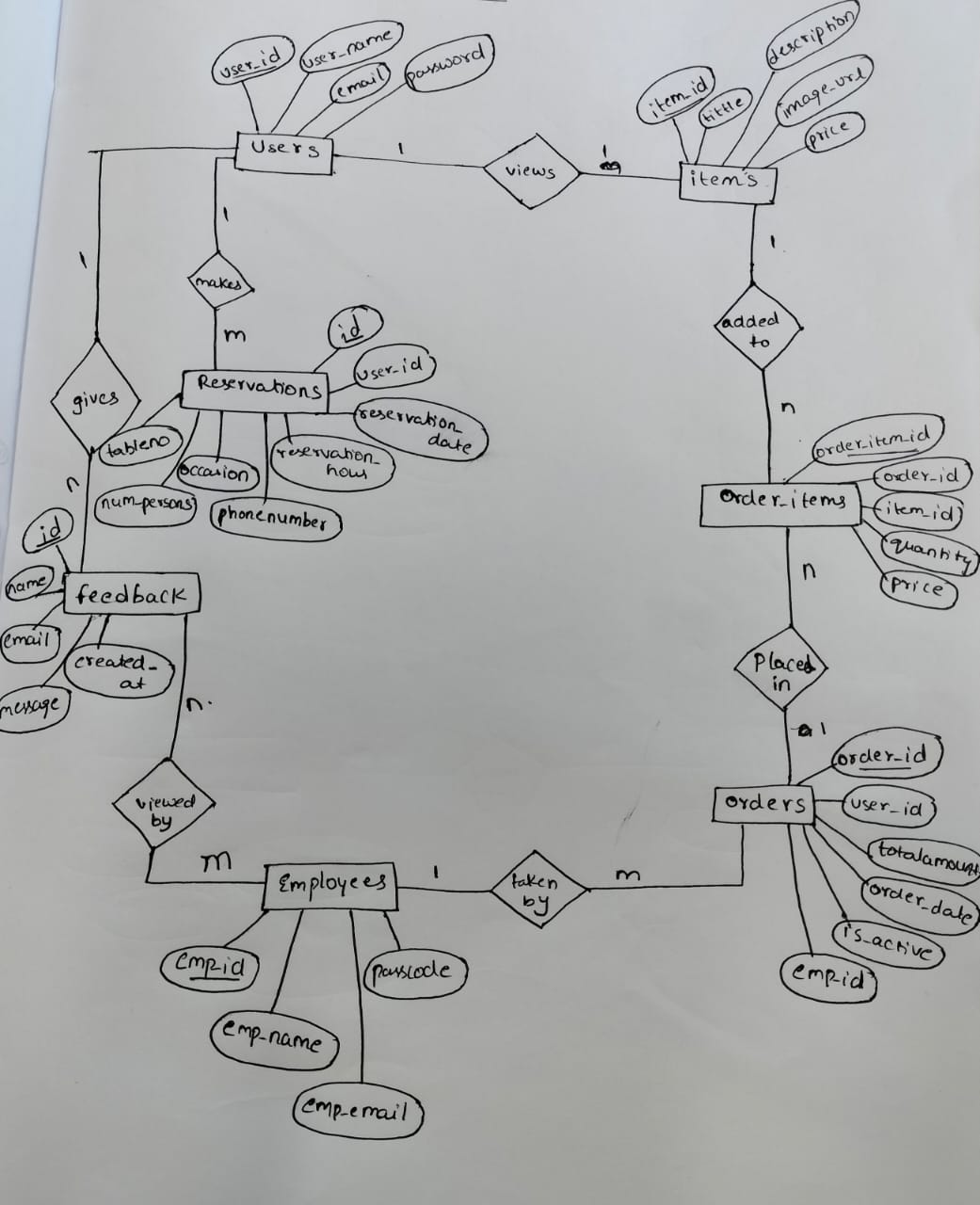
1. **Future Enhancements**:

* The platform is designed with scalability in mind, allowing future expansion to include additional features like delivery services, loyalty programs, and advanced data analytics for better decision-making.
* Based on ongoing feedback from users and staff, new features and improvements will be integrated into the system to meet evolving restaurant management needs.

By following this structured methodology, the platform ensures a high level of reliability, efficiency, and user satisfaction, making it an indispensable tool for modern restaurant management.

**DATABASE:**

* **ER-DIAGRAM:**

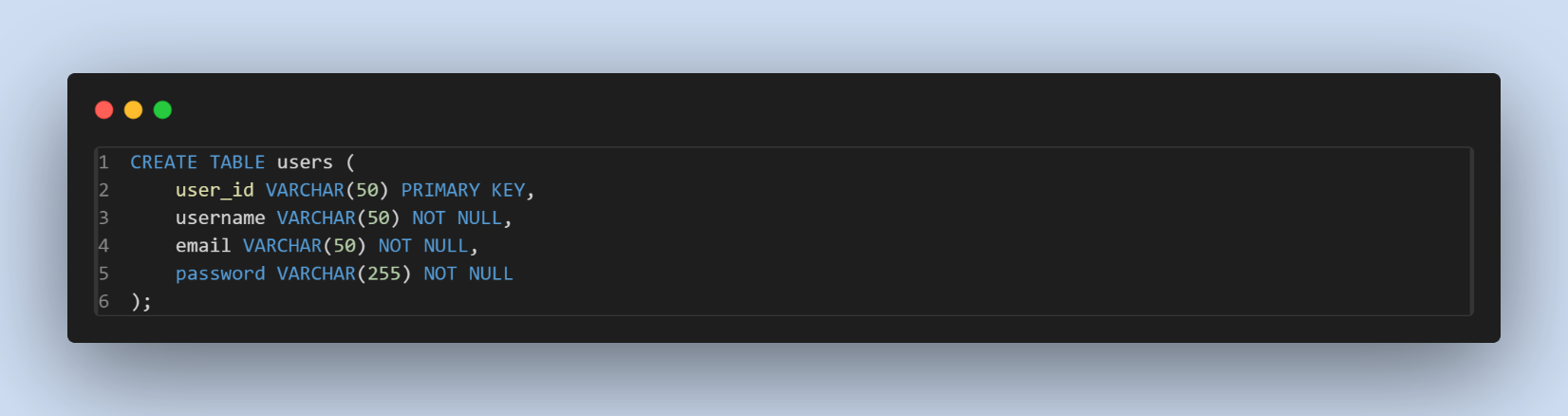


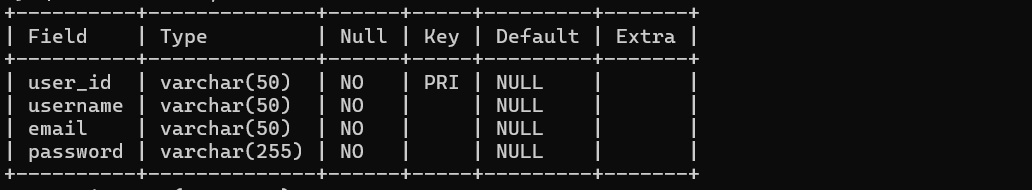
* **DATABASE CREATION:**

1. **User Table**

Where user details stores when the sign-up

And this table is used for used functionality purposes.



****

1. **Items table**

Here in this table all the menu items will be stored in this.

This table is used for viewing menu

****

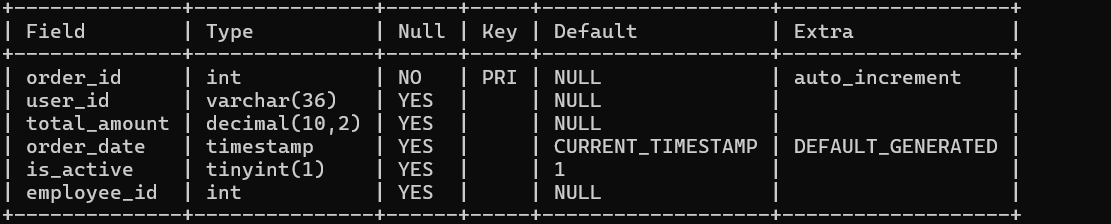
****

1. **Orders table**

Here in this table all the which were placed by user will be stored in this.

The is\_active is handled by employee

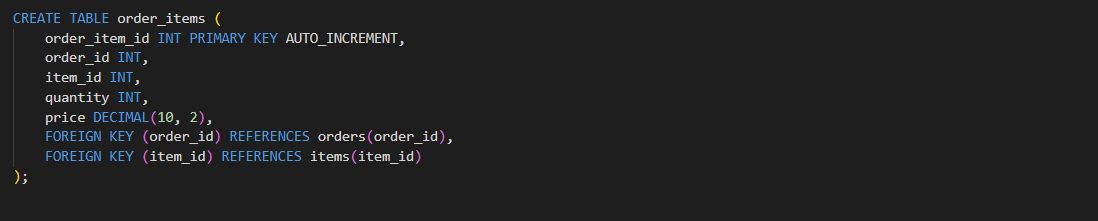
****

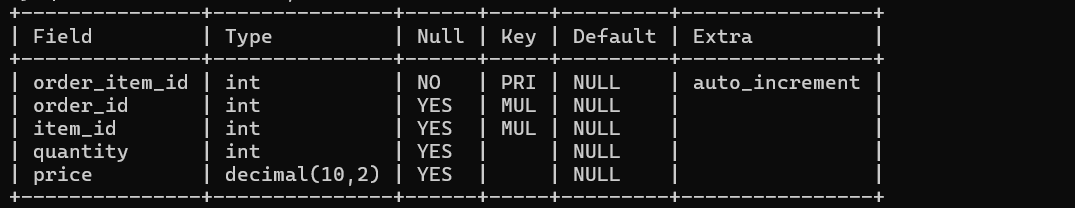
****

1. **Order\_items table**

Here in order\_items what are all the items been placed will be stored by his and order\_id created in the session.

This table is used for cart

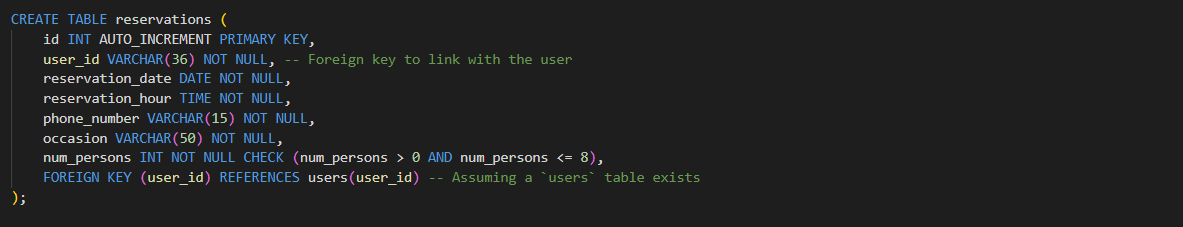
****

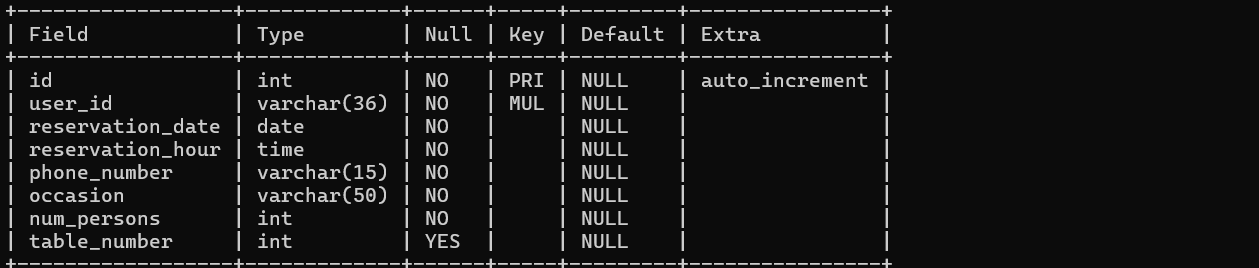
****

1. **Reservations table**

When user reserves table the details will be stored in this table

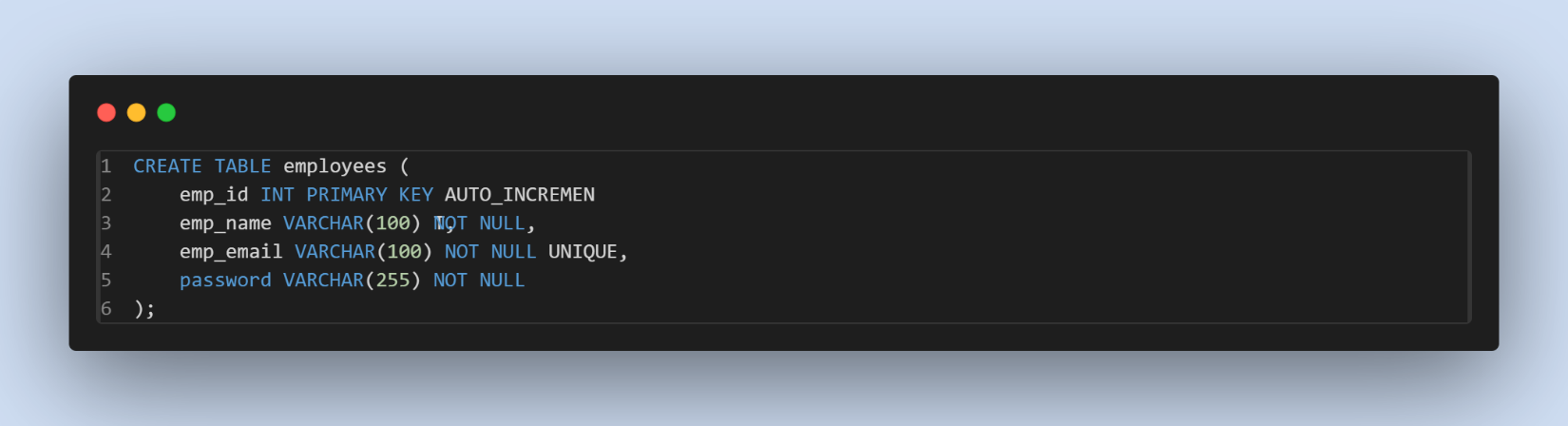
Further references to orders table

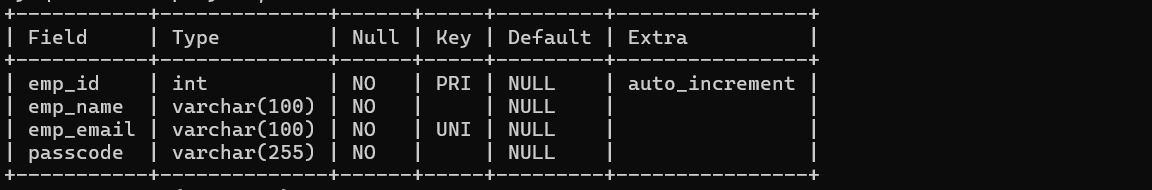




1. **Employee table**

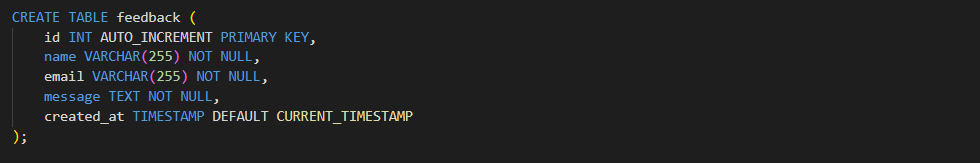
Pre existed details will be given in the employees table

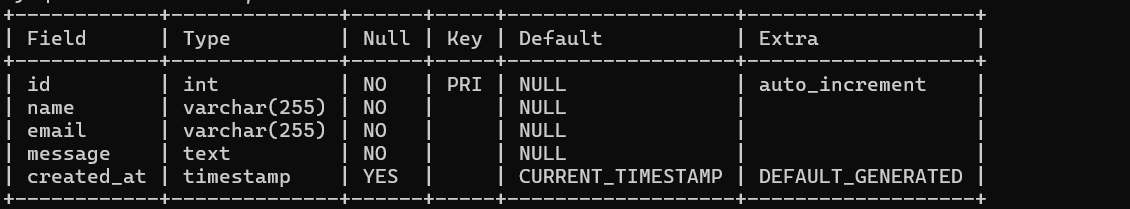
****

****

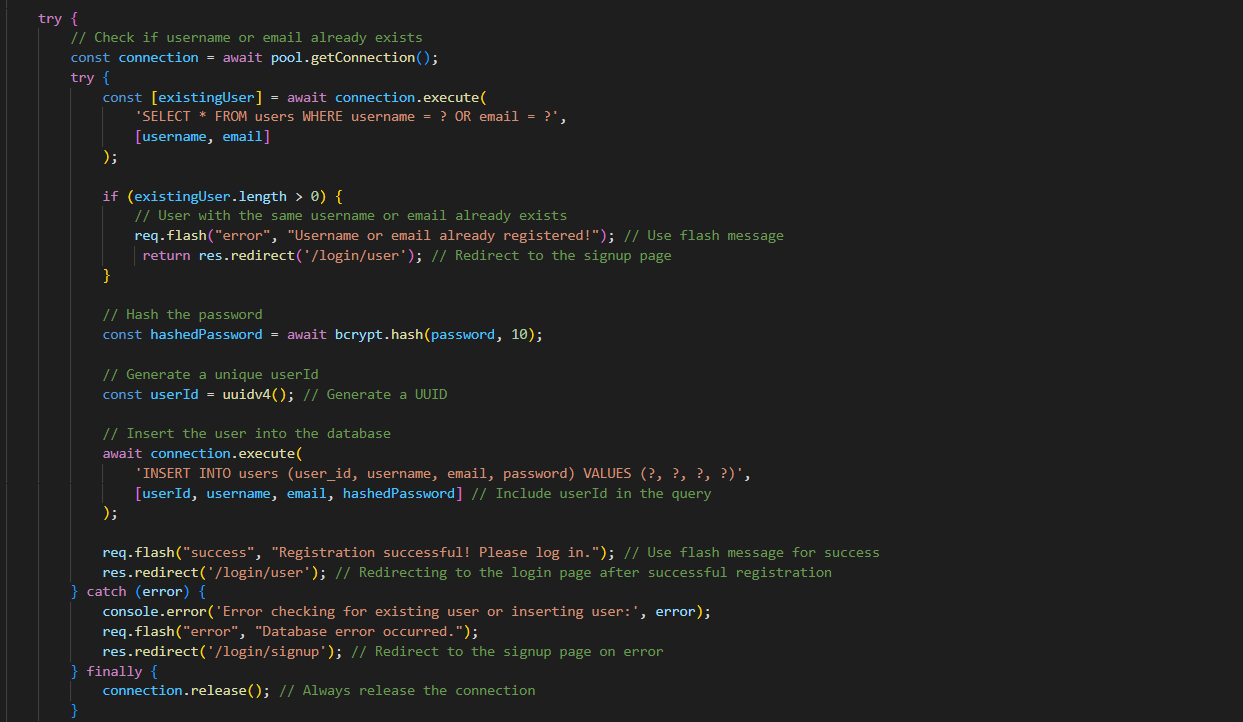
1. **Feedback table**

Here feedback is stored where user is given and will this feedback in his employee authentication page.

****

****

* **Queries used:**
* **User signup query:**

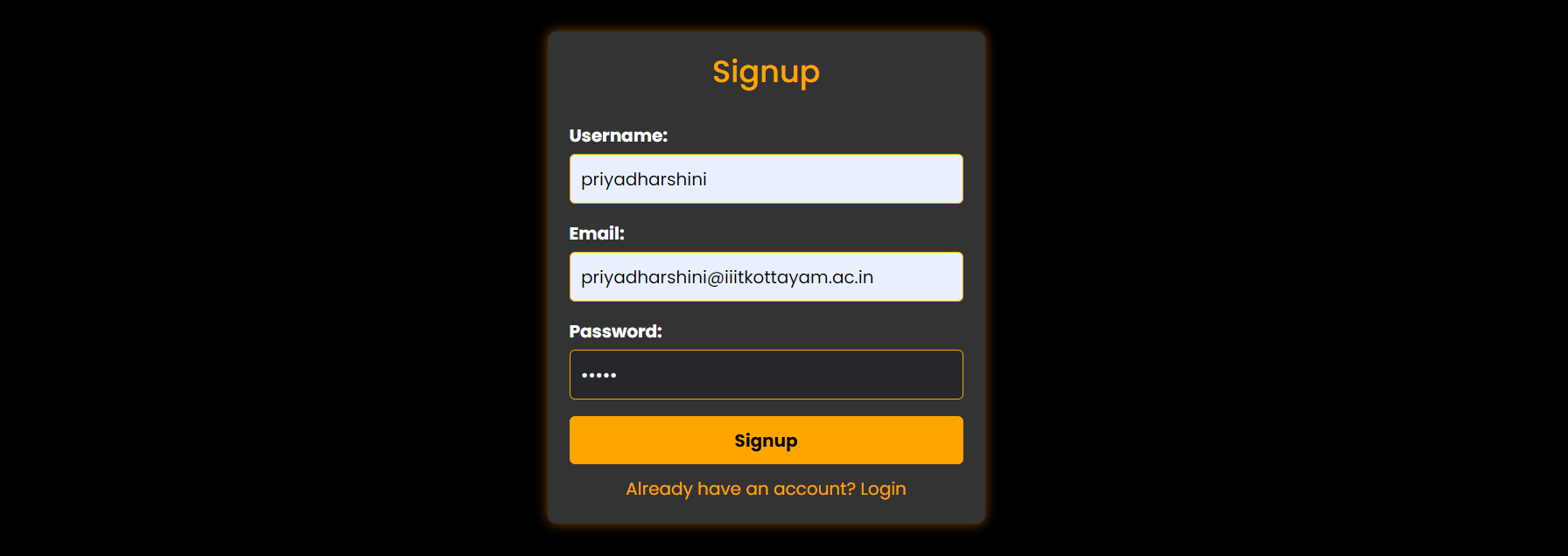
****

User first creates his account with his/her name , email, password here if the same username or email is already existed in the database his/her account can’t be created.

Here are the sql queries for the above scenario

Here when user creates his account the details are stored in the database.

**Testing:**

****

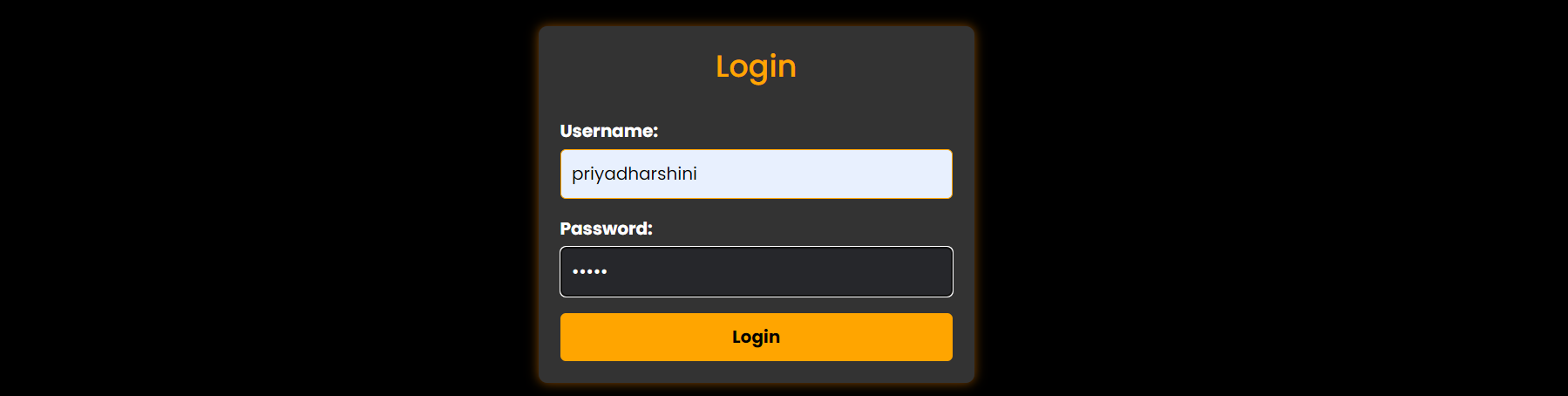
****

* **User login query:**

****

Here when ever user tries to login first it will check a particular user has signed in or not, if user doesn’t signed in and tries to login it shows pops up to signup or any username or password issues it shows invalid username or password.

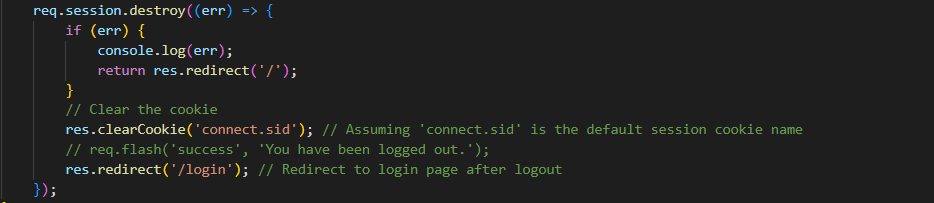
**Testing:**

****

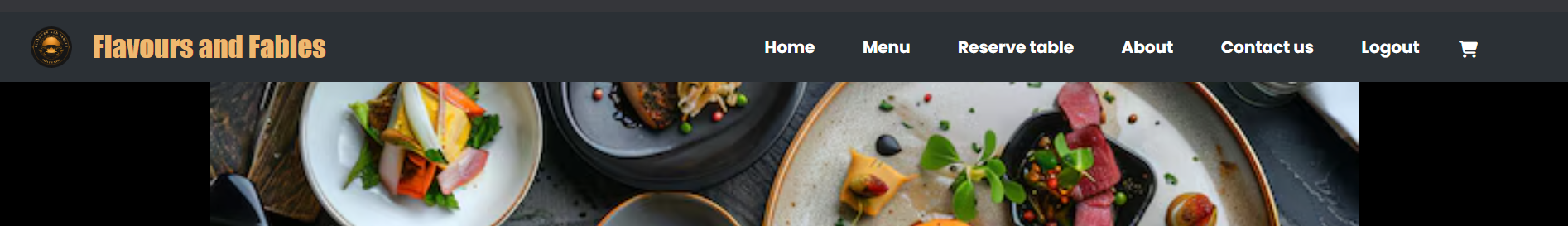
****

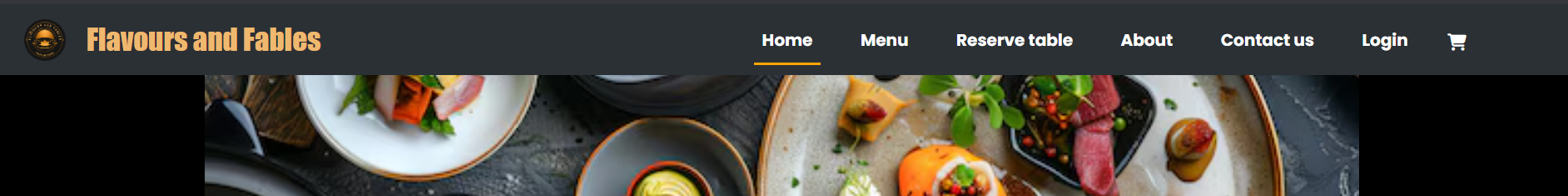
Here user is already sign up so it redirects to home page.

* **User logout**



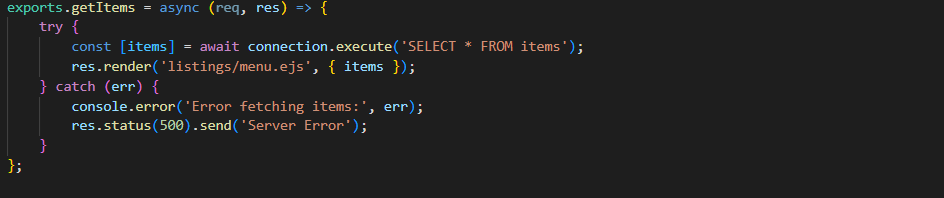
Here when user clicks logout in nav-bar users session will be destroyed





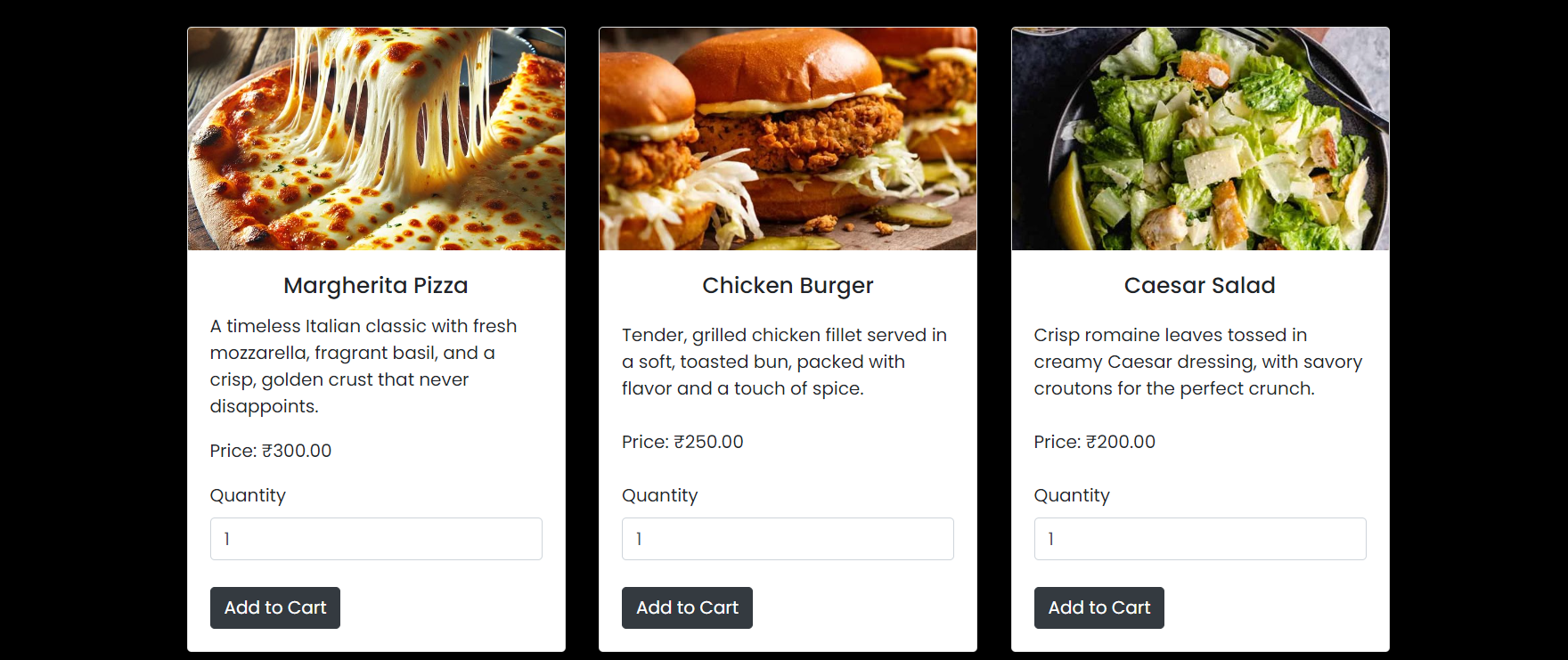
Here session of certain user ends and redirects home page.

* **Get items query**



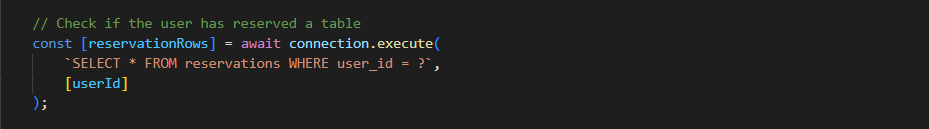
Here items from the items table will be fetched and displayed in the menu page.

Here we used cards to display the menu, in menu we can update the quantity and add to cart functionalities will be there.

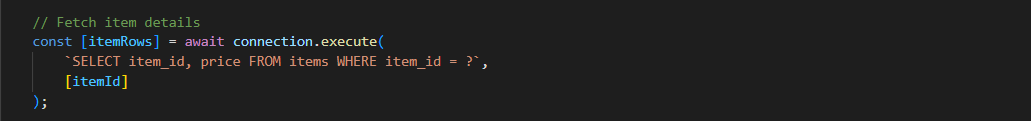


There are many rows of items but this is a sample items list from the menu page.

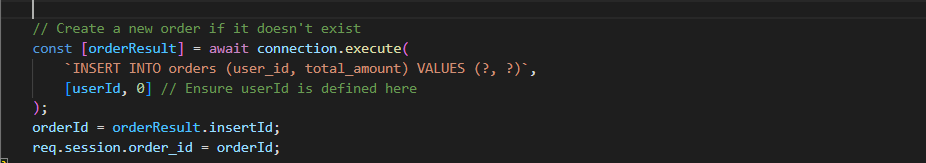
* **Add to cart query:**



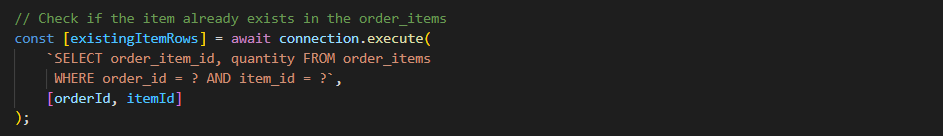
First when user tries to add the item to cart, it checks whether the user has reserved the table.



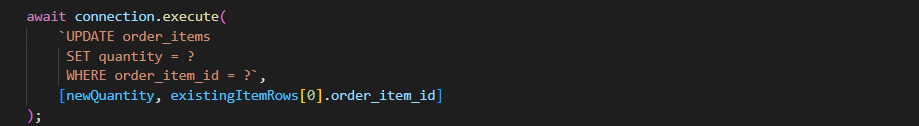
When user adds the item to cart, fetches the item details of item added to cart and stores in item\_id for further functionality.



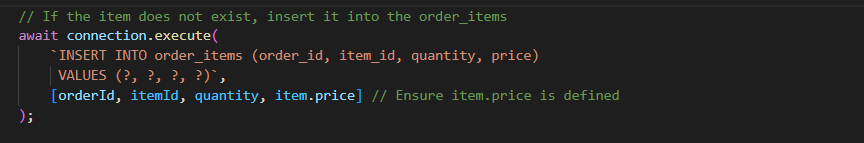
When item is added to cart a order\_id is created in orders table for that order\_id the session user\_id, total amount will be stored.



After adding item to cart the order\_items will be stored in the order\_items tables along with their unique order\_id fetched from session and quantity, amount of the certain item.



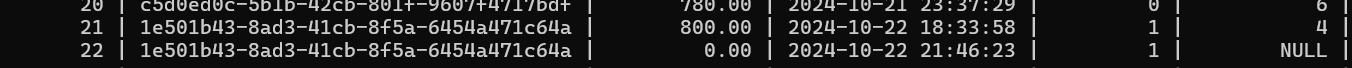
Whenever user tries to add the same item to the cart then the item in the cart will be updated by one.



And then all the items will be inserted into order\_items table.

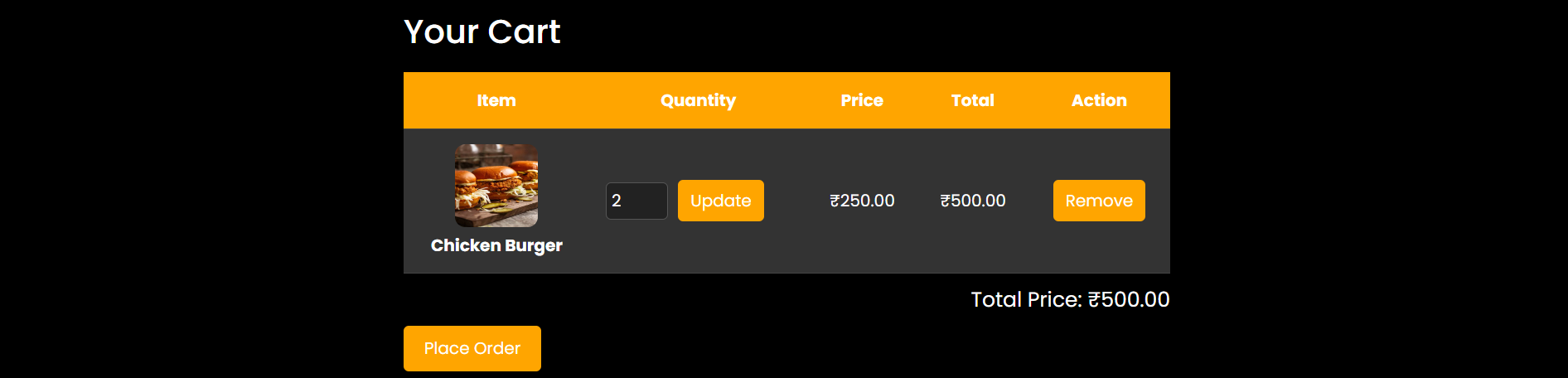
**Testing:**



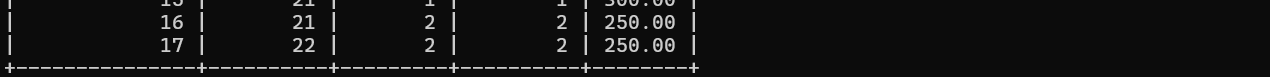
Here after placing order only a employee will be assigned to particular order.



This is the order\_items table whenever user adds to cart the item will be inserted to order\_items table with the order\_id, quantity.

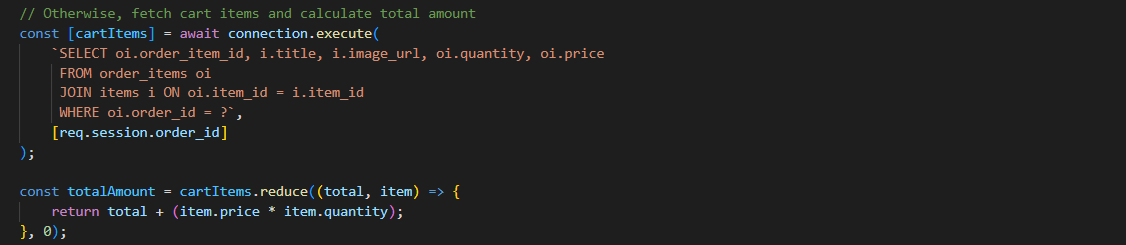


Here I updated the quantity to 2



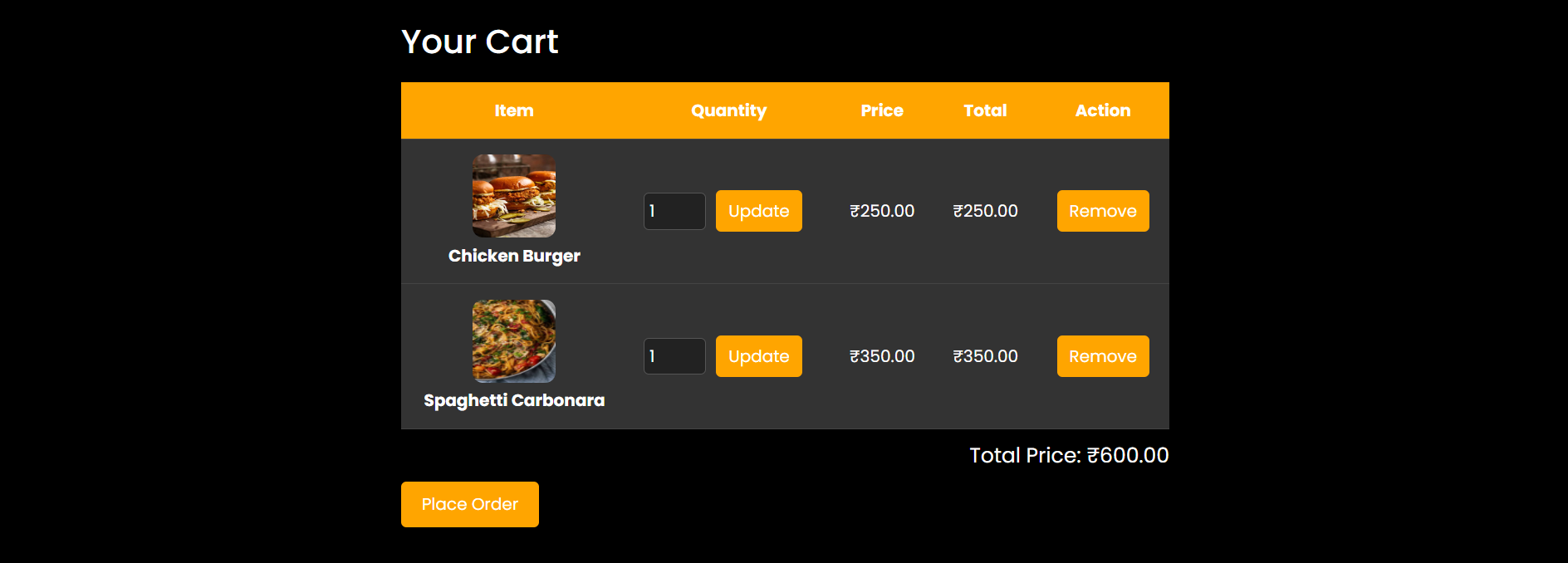
Here is the database change in order\_items table too.

* **View cart:**



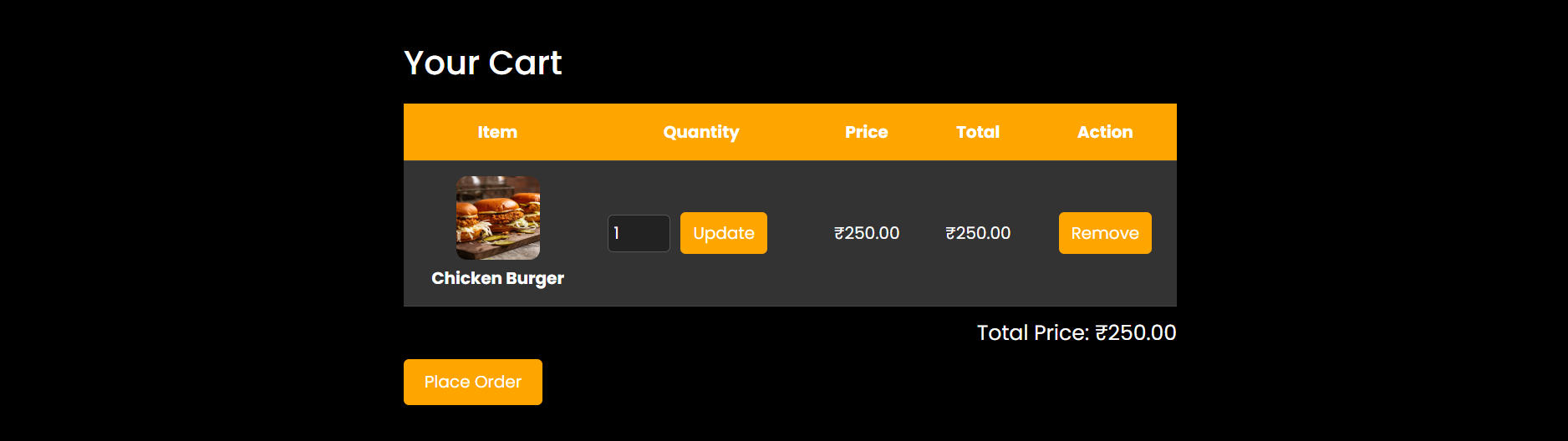
For viewing cart after adding the items.

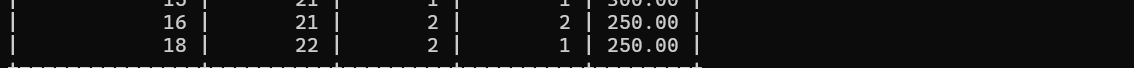
**Testing:**

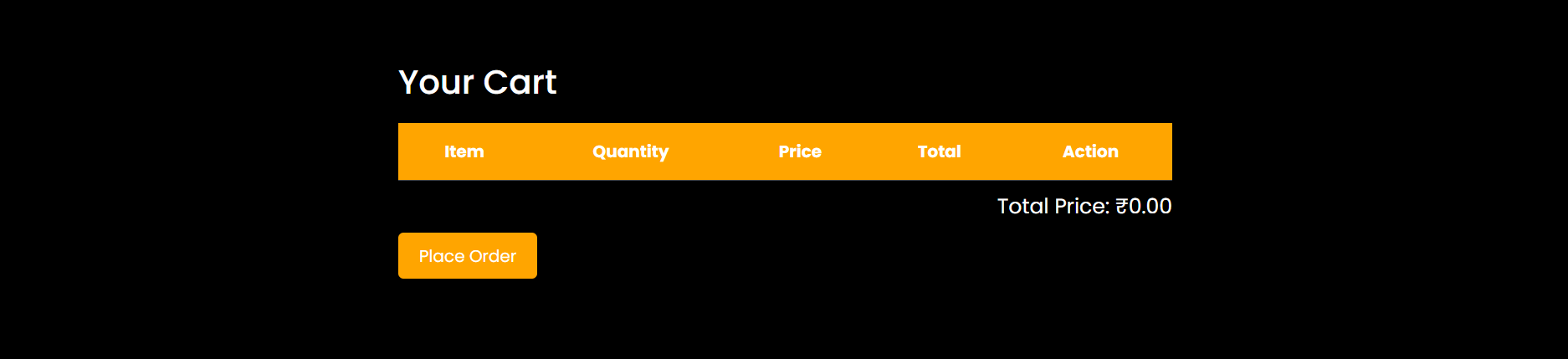
****

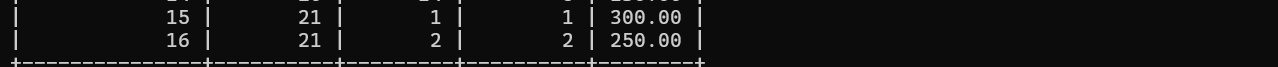
* **Delete query**

****

****

****

****

****

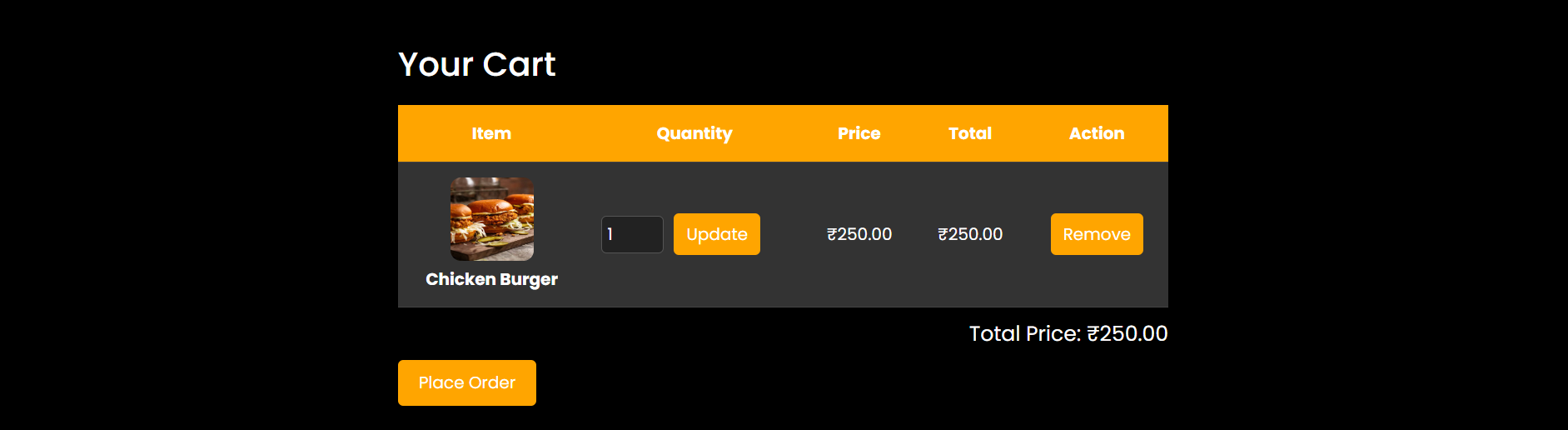
Whenever user removes the item from the cart the item is deleted from the order\_items table.

* **Place order query:**

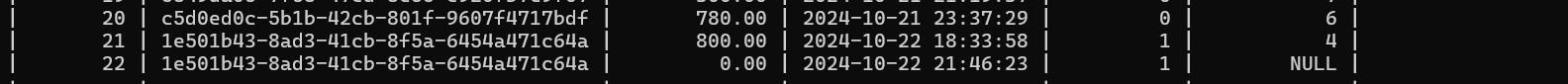


Here after placing order final updates will be done in orders table and order\_items table and here a random emp\_id will be given to the order so that a certain employee will takes the order.

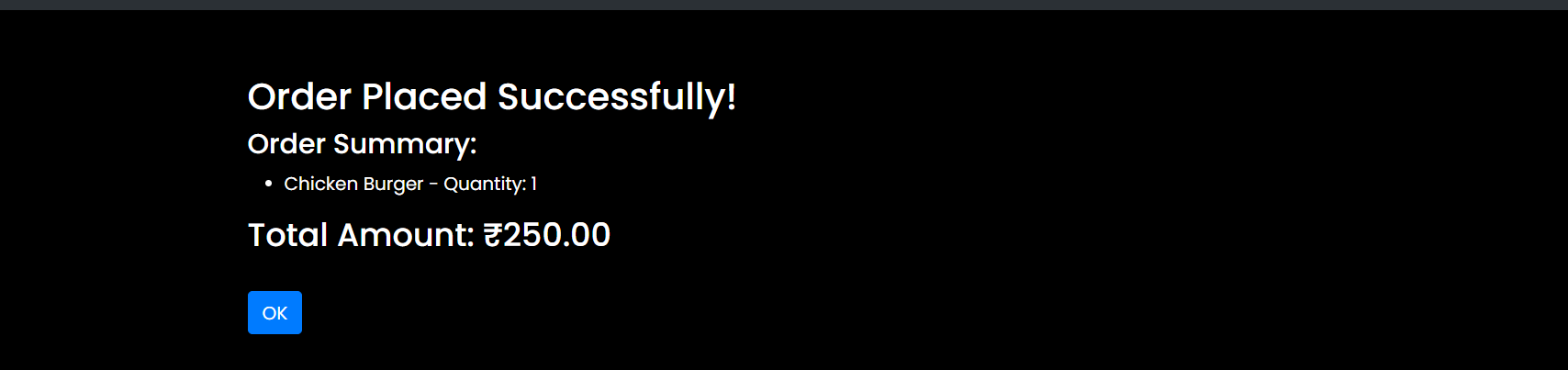
And the total amount will be updated in orders table.



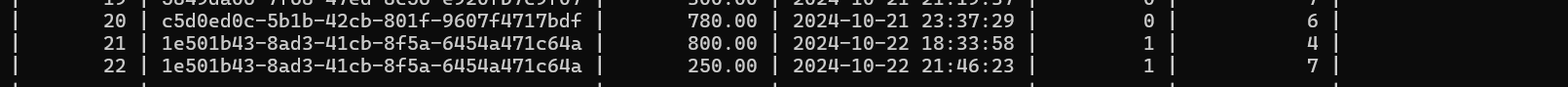
Before placing the order



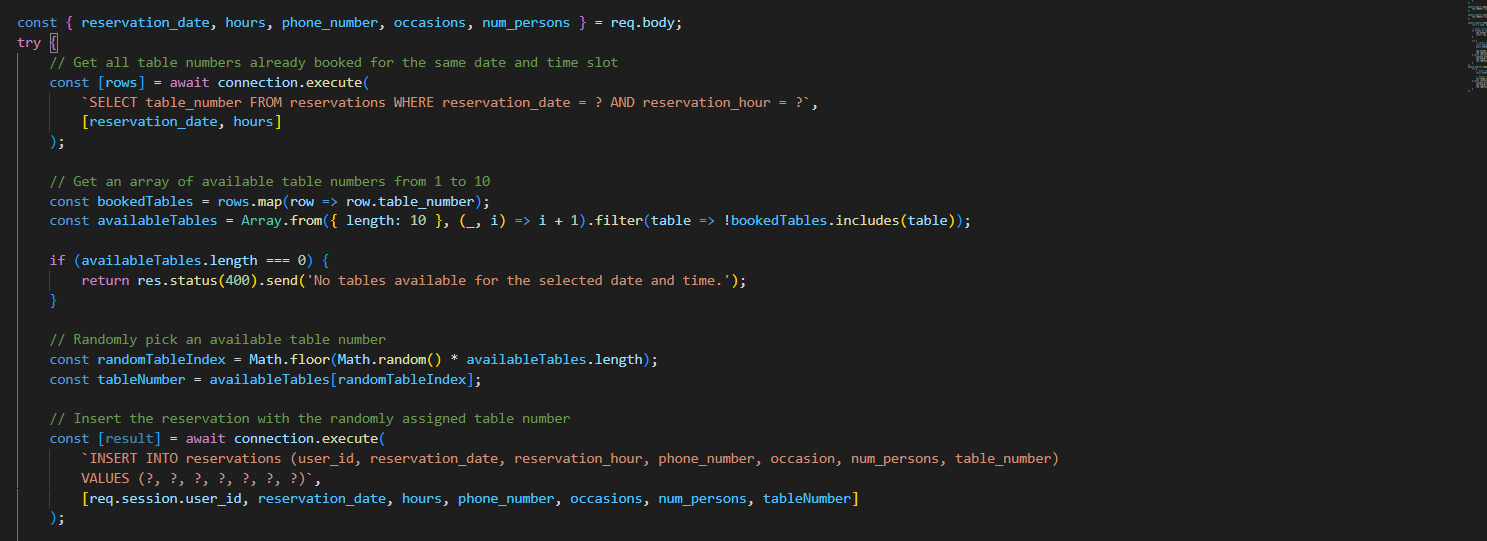
This the order confirmation page.



After placing the order, a emp\_id is assigned



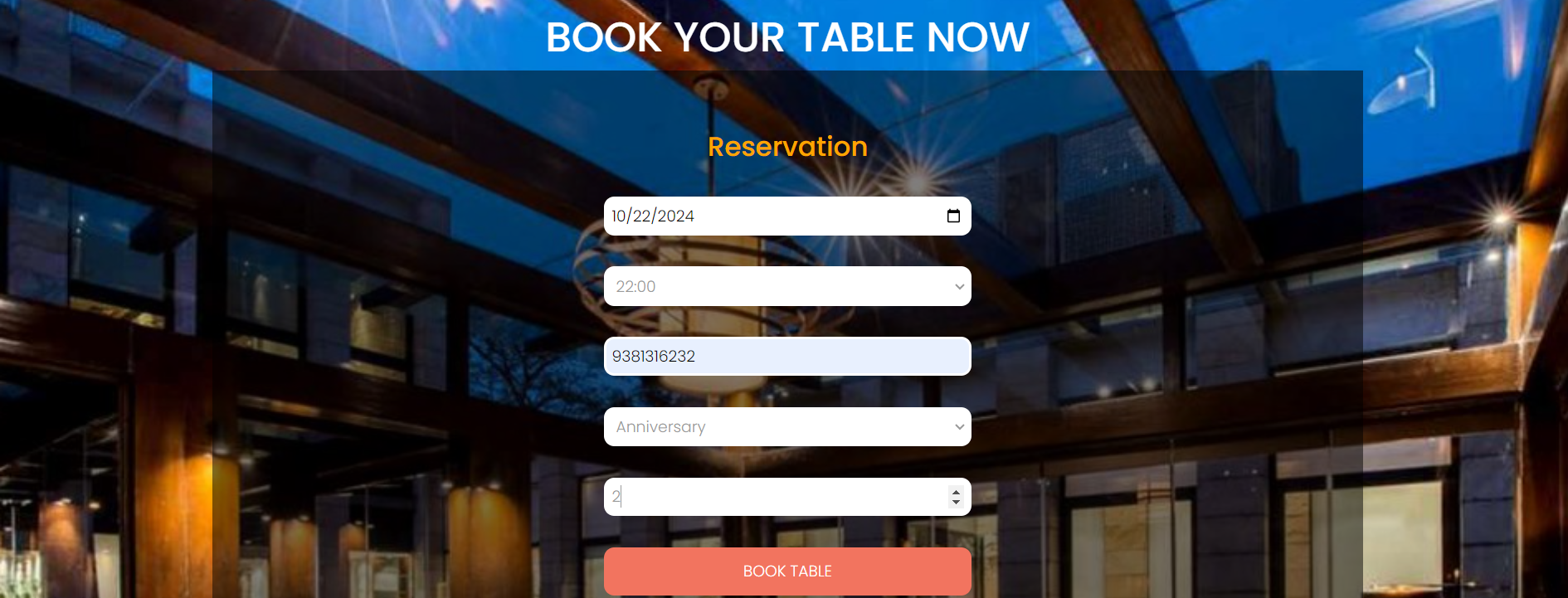
* Table reservation query

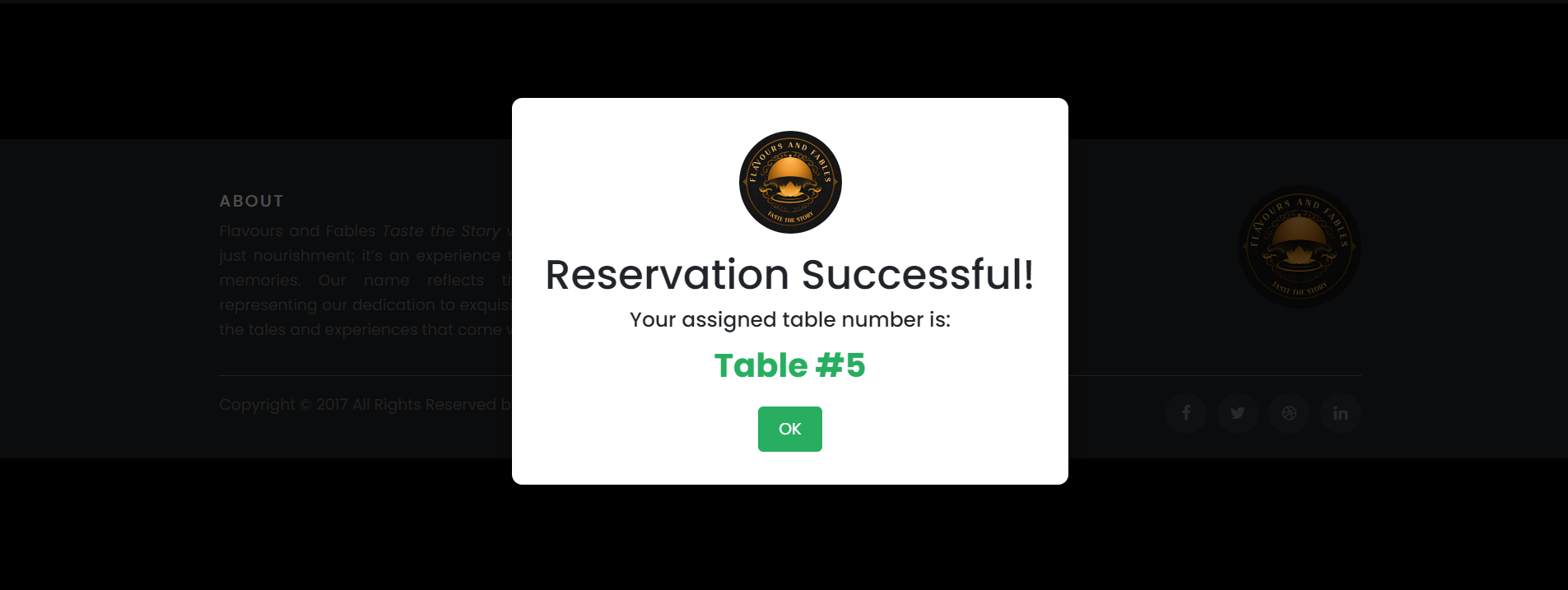


Here user can reserve a table after login

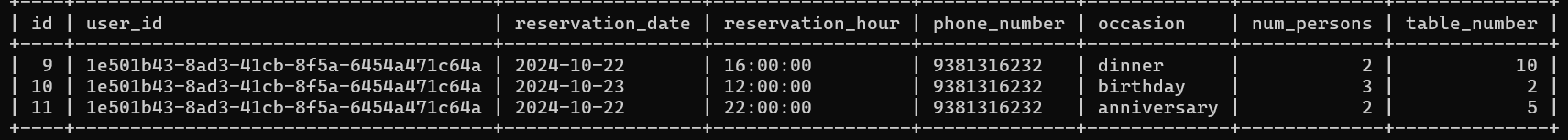
User fills the required details here query checks whether the table is free for the selected day and time slot if the table is free he will be ssigned a table else displays all tables are full for selected date and time slot.

**Testing:**

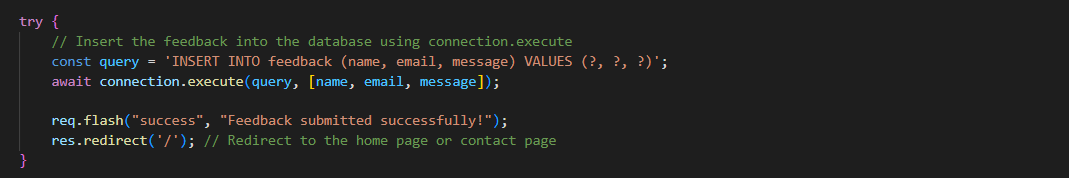
****

****

When reservation is done a popup will be displayed with the table\_no

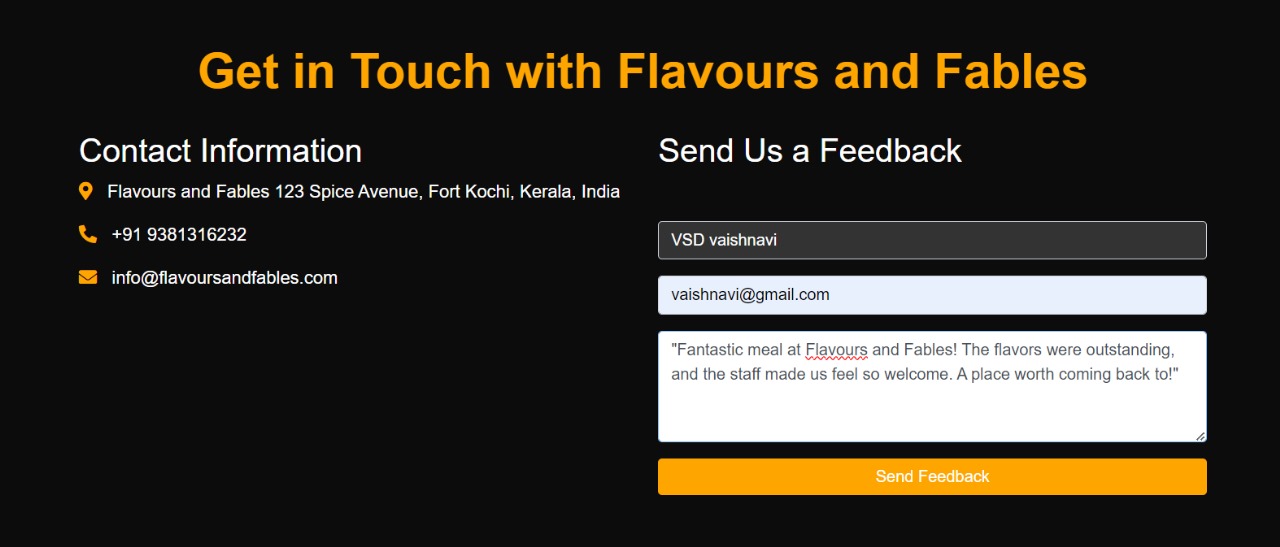
****

* **Feedback query:**

****

All the feedback details will be stored in feedback form

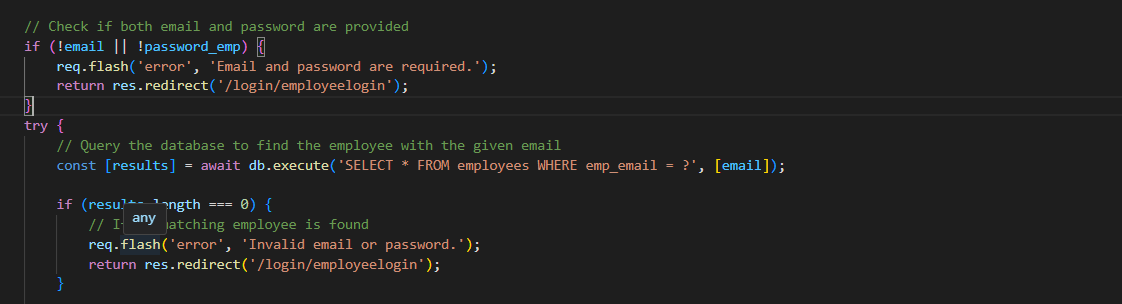
**Testing:**





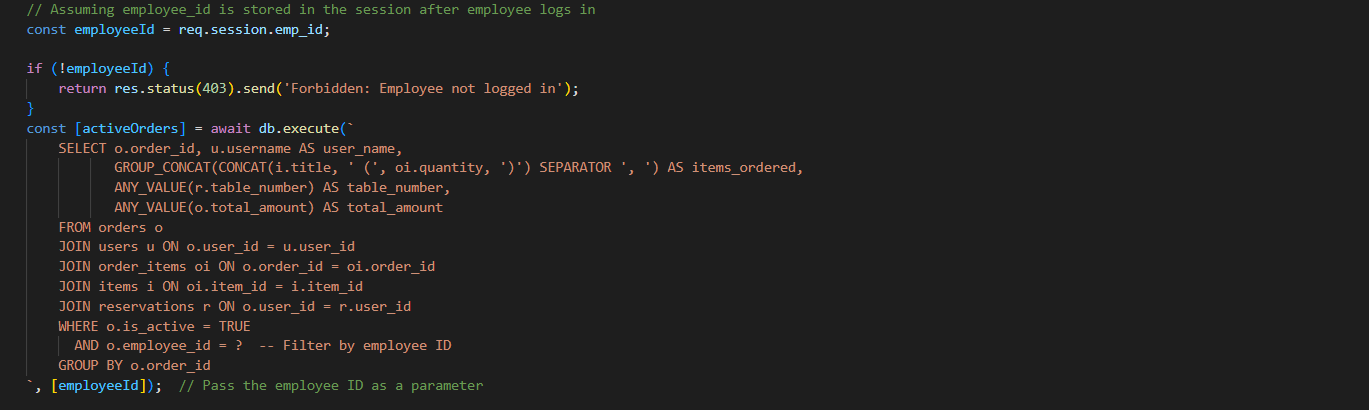
From the contact us feedback will be written and viewed by the employee.

* **Employee login query**



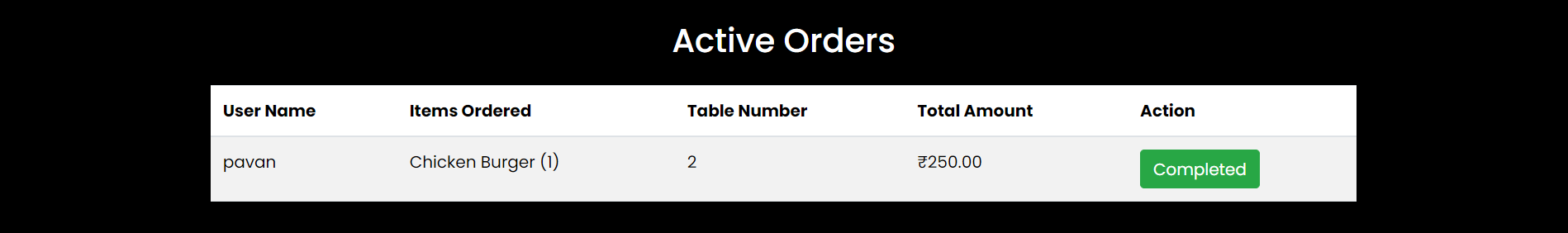
We have given a sample database for employee email and password by that credentials only a certain employee can login

* **Activeorders query:**

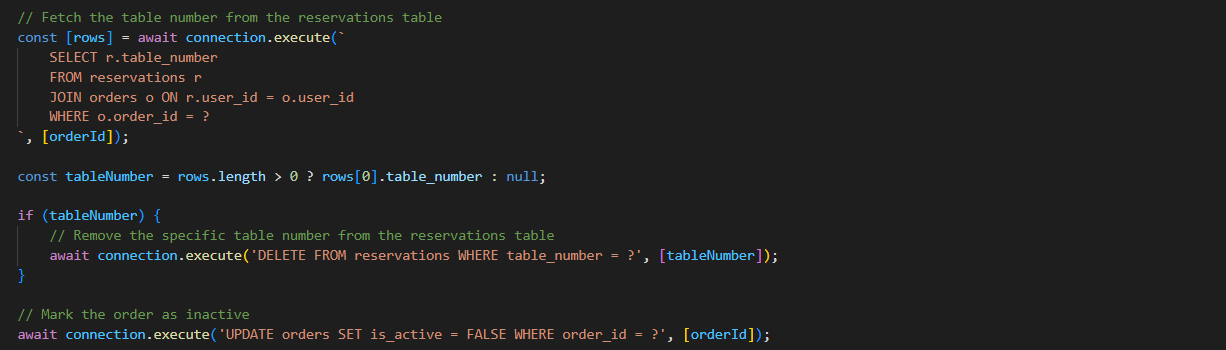


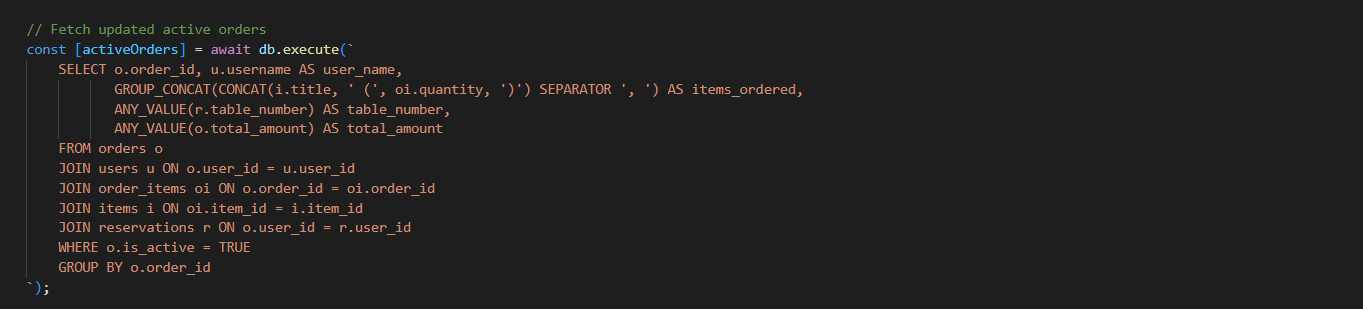
Here active orders will be fetched from orders table and displayed to particular employee of assigned orders

**Testing:**

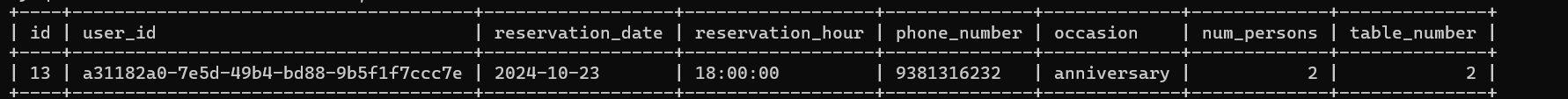
****

* **Completed order query:**

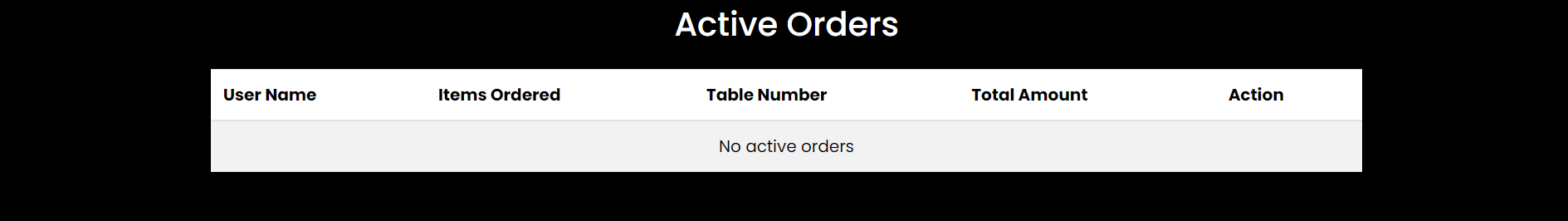




Here after clicking complete order the user\_id related to that order will be fetched and deletes the reservations table of that certain user\_id session.



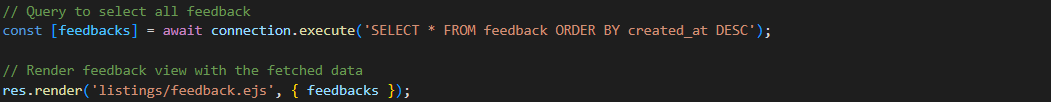
Before clicking complete orders



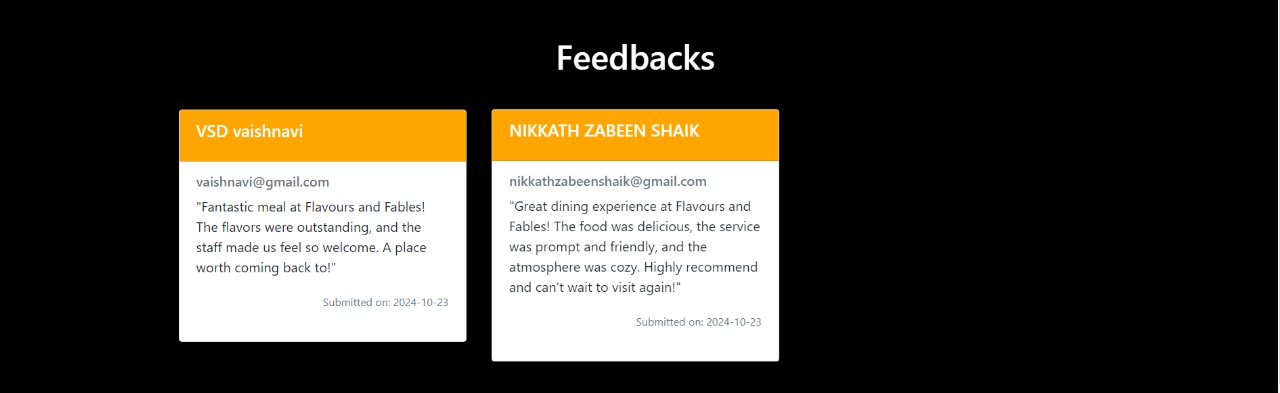


After clicking complete orders.

* **Feedback viewing qwery**



Here the feedbacks given by users will be viewed by employee



**Conclusion:**

Finaly this is our website where streamline all the tasks of employee , manage reservations, manage orders, and user, employee authentication.

**Thank you**