

# **RESTAURANT AND FOOD SERVICES MANAGEMENT SYSTEM**

## **EDS6397 – DATABASE MANAGEMENT TOOLS**



### **GROUP-2**

Bharathwaj Anumula Guru 229815

, Jerusha Gnanamani Bhaskaran 1599927

Keerthi Yadav Eeraboina 2311912

Naga Venkata Vignesh Reddy Upputuri 2313040

Pooja Bejjanki 2311609

Praneeth Achanta 2302324

Sharath Kumar Reddy 2313772

Vara Lakshmi Mamillapalli 2315749

Vasista Tummala 2316063

Victor Paul Buddha 2298520

## **A. INTRODUCTION**

The goal of this project is to develop a comprehensive Restaurant and Food Services Management System that empowers restaurant owners and staff to efficiently manage daily operations. From menu creation to customer orders and reservations, this system will streamline processes and enhance the overall dining experience. We will use PHP, HTML, CSS, JavaScript for the frontend and MySQL and phpMyAdmin as a relational database. The UI is intuitive, and the SQL database is well-structured for data management.

The Restaurant and Food Services Management System is designed to function as a central hub, allowing restaurant proprietors to efficiently optimize their processes and deliver outstanding dining experiences to their customers. Utilizing intuitive functionalities and easy-to-navigate interfaces, our platform aims to foster smooth communication between restaurant personnel and guests, thereby boosting overall productivity and contentment.

We provide a user-friendly and intuitive interface for everyone involved. Our development approach will follow a detailed process, encompassing gathering business needs, crafting logical ER diagrams, relational diagrams, and building the database. Through integrating the database with the website, our goal is to ensure smooth data entry and retrieval processes, thereby improving the platform's usability and functionality.

## **B. Modules**

1. User Registration and Authentication
2. Menu and categories
3. Reservation
4. Order Management
5. Payment gateway
6. Admin Panel and Inventory
7. Reporting and Analytics
8. Staff and Manager roles

## **C. BUSINESS RULES**

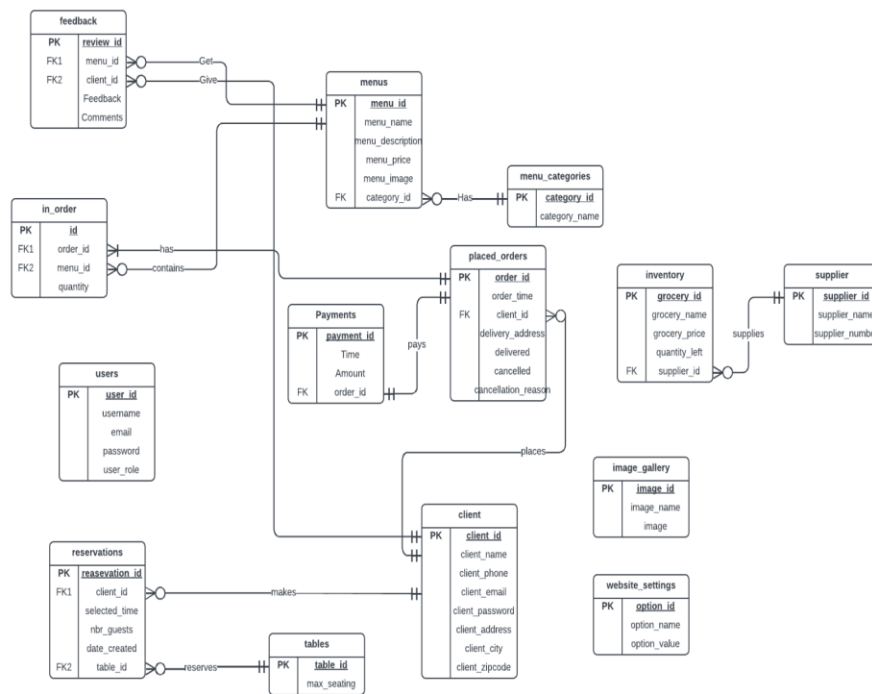
- **User Registration and Authentication:**
  - Users must create an account with a unique username and strong password.
  - Password complexity requirements should be enforced (e.g., minimum length, combination of uppercase and lowercase letters, numbers, and symbols).
  - User roles (Customer, Staff, Manager, Administrator) should be assigned during registration.
- **Menu Management:**
  - Only Managers can create and Update menu items.
  - Each menu item must have a unique name, description, price, calories per serving, rating, and Category.

- Menu items should have their availability marked.
- Images for the menu items can be uploaded for better presentation.
- **Reservation Management:**
  - Customers can search for available reservations based on date, time, and party size.
  - Reservations can be made online or through phone calls.
  - Managers and staff can view and manage all reservations, including accepting, modifying, or canceling them.
- **Cart Management:**
  - Customers can select the required items online in case of takeout and delivery orders.
  - Menu items along with their quantity can be added to the cart.
  - Customers can modify, delete and update all the items in the cart.
  - Customers can check the availability of the menu items as well.
- **Order Management:**
  - Staff (Servers) can take orders for dine-in, takeout, and delivery.
  - Takeout and delivery orders can be placed online or through the phone.
  - Order status updates (e.g., preparing, ready, delivered) should be communicated to customers.
- **Payment Processing:**
  - Secure payment gateway integration is required for various payment methods (e.g., credit cards, debit cards, digital wallets).
  - Takeout and delivery orders can be prepaid online or paid upon receiving the order.
  - Customers can pay for dine-in orders at the table or checkout counter.
  - Invoices with detailed order information (OrderID, Order time) should be generated for all transactions.
- **Customer Feedback and Reviews:**
  - Customers can submit feedback and reviews after their dining experience.
  - Manager can respond to feedback and reviews,
  - Inappropriate or offensive reviews can be moderated or removed.
- **Inventory and Supplier Management:**
  - Inventory levels for ingredients and supplies should be tracked.
  - The grocery cost of the current stock must be available to be compared with the supplier cost.
  - Managers can create purchase orders for suppliers based on inventory needs.
  - Supplier information (contact details, pricing) should be managed within the system.
- **Reporting and Analytics:**
  - Reports should be easily accessible to Administrators for data-driven decision making.
  - System should generate comprehensive reports on various aspects:
    - Sales by menu items, categories, and time periods.
    - Inventory usage and cost analysis.
    - Reservation trends and booking patterns.
    - Customer feedback and review analysis.
- **Admin Panel:**
  - Admins must have full access to manage all aspects of the system:

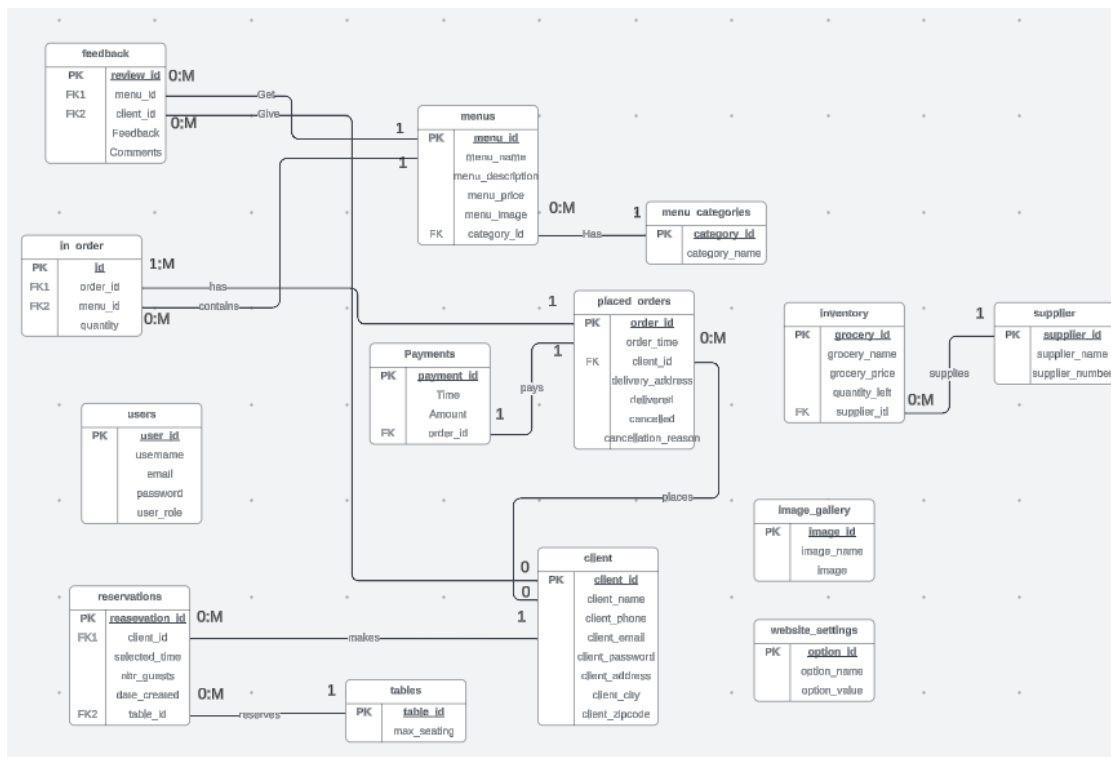
- Staff and manager accounts (create, edit, delete)
- Menu Items (pricing, availability)
- Reservations (view, accept, modify, cancel)
- Orders (review, track status, resolve issues)
- Access and analyze reports and data.

## D. ER DIAGRAM

Entity-Relationship Diagram (ERD) consists of fourteen entities, each fulfilling a key function within the database. These entities are connected through foreign keys and are distinguished by their primary keys. The ERD provided below offers a comprehensive overview of the database's functionality.



## E. RELATIONAL DIAGRAM



## ENTITIES

We have thirteen entities in our ERD. We'll have a look at each entity in detail along with their primary and foreign keys.

- **Client:** This entity stores any information regarding the client with attributes names of Client's ID, which is the Primary key (PK), Name, Email, Phone Number, Password, Address with zip code.
- **In\_order:** This is an associative entity used to break placed orders and menus from being a many to many.
- **Menus:** This entity stores all menu items served by the restaurant. Item ID is used as the primary key. Category ID is used to indicate what category the menu item belongs to, which also acts as a foreign key for this table from the Category table. There are other attributes associated with this table such as Price, Name, Description, Image, Calories per serving and Rating of the item which provides more details about the item.
- **menu\_categories:** This contains information of which category each menu item falls in. Primary key for this entity is category id. It also contains category name.
- **placed\_orders:** This entity tracks all placed orders with order ID being the primary key. There is a foreign key of client id. It contains order time, delivery address, whether the order was delivered or cancelled and the cancellation reason.
- **Reservations:** This entity has information regarding reservations that the client made. Reservation ID is the primary key and two foreign keys of client ID and table ID. The

other attributes are selected time, number of guests and the date the reservation was created.

- **Tables:** This entity places table numbers on the reservations being made. The primary key is table id and has one other attribute of max seating.
- **Users:** This entity stores information about admin, staff and managers. It has a primary key called user ID. It also has attributes in username, password, email and user roles.
- **Website\_settings:** This entity keeps information regarding the data structure or object within a website's backend that stores and manages user settings. This entity is responsible for handling user preferences, configurations, and other settings related to the website's functionality and user experience.
- **Supplier:** Entity tracks the suppliers, including details such as Supplier ID (Primary Key), Name and their Phone number. All the suppliers can be added or deleted from the table by the admin.
- **Inventory:** Entity records all the inventory items along with the Quantity available and their Cost. Grocery ID acts as a primary key in this case. This table can be used to reference all stocks present in the inventory.
- **Feedback:** This entity stores all feedback given by customers and contains five attribute. The primary key is review id. There are two foreign keys menu id and client id. It has two other attributes, feedback about the restaurant or comments on the menu item.
- **Payment:** This entity stores all payment-related information which is related to each order. The attributes in this entity include Payment ID (Primary Key), time, amount, method of payment and Order ID. Order ID is a foreign key in this case from the order table, as each payment must be related to some order.

## F. DATA DICTIONARY

| Table Name    | Attribute       | Data Type    | Required | PK/FK/UK | FK Referenced Table |
|---------------|-----------------|--------------|----------|----------|---------------------|
| Client        |                 |              |          |          |                     |
|               | client_id       | INT(5)       | Y        | PK       |                     |
|               | client_name     | Varchar(50)  | Y        |          |                     |
|               | client_phone    | VARCHAR(50)  | Y        |          |                     |
|               | client_email    | VARCHAR(100) | Y        | UK       |                     |
|               | client_password | VARCHAR(100) | Y        |          |                     |
|               | client_address  | VARCHAR(100) | Y        |          |                     |
|               | client_city     | VARCHAT(15)  | Y        |          |                     |
|               | client_zipcode  | INT(6)       | Y        |          |                     |
| Image_gallery |                 |              |          |          |                     |
|               | image_id        | int(2)       | Y        | PK       |                     |

|                 |                     |              |   |    |              |
|-----------------|---------------------|--------------|---|----|--------------|
|                 | image_name          | varchar(30)  | Y |    |              |
|                 | image               | varchar(255) | Y |    |              |
| In_order        |                     |              |   |    |              |
|                 | id                  | int(5)       | Y | PK |              |
|                 | order_id            | int(5)       | Y | FK | Placed_order |
|                 | menu_id             | int(5)       | Y | FK | menu         |
|                 | quantity            | int(3)       | Y |    |              |
| Menus           |                     |              |   |    |              |
|                 | menu_id             | int(5)       | Y | PK |              |
|                 | menu_name           | varchar(100) | Y |    |              |
|                 | menu_description    | varchar(255) | Y |    |              |
|                 | menu_price          | decimal(6,2) | Y |    |              |
|                 | menu_image          | decimal(6,2) | N |    |              |
|                 | category_id         | int(5)       | Y | FK | category     |
| menu_categories |                     |              |   |    |              |
|                 | category_id         | int(3)       | Y | PK |              |
|                 | category_name       | varchar(50)  | Y |    |              |
| placed_orders   |                     |              |   |    |              |
|                 | order_id            | int(5)       | Y | PK |              |
|                 | order_time          | datetime     | Y |    |              |
|                 | client_id           | int(5)       | Y | FK | client       |
|                 | delivery_address    | varchar(255) | Y |    |              |
|                 | delivered           | tinyint(1)   | Y |    |              |
|                 | canceled            | tinyint(1)   | Y |    |              |
|                 | cancellation_reason | varchar(255) | Y |    |              |
| reservations    |                     |              |   |    |              |
|                 | reservation_id      | int(5)       | Y | PK |              |
|                 | date_created        | datetime     | Y |    |              |
|                 | client_id           | int(5)       | Y | PK | clients      |
|                 | selected_time       | datetime     | Y |    |              |
|                 | nbr_guests          | int(2)       | Y |    |              |
|                 | table_id            | int(3)       | Y | FK | tables       |
|                 | liberated           | tinyint(1)   | Y |    |              |

|                  |                     |                                 |   |    |          |
|------------------|---------------------|---------------------------------|---|----|----------|
|                  | canceled            | tinyint(1)                      | Y |    |          |
|                  | cancellation_reason | varchar(255)                    | Y |    |          |
| tables           |                     |                                 |   |    |          |
|                  | table_id            | int(3)                          | Y | PK |          |
|                  | max_seating         | int(3)                          | Y |    |          |
| users            |                     |                                 |   |    |          |
|                  | user_id             | int(2)                          | Y | PK |          |
|                  | username            | varchar(20)                     | Y | UK |          |
|                  | user_number         | varchar(20)                     | N | UK |          |
|                  | email               | varchar(30)                     | Y |    |          |
|                  | password            | varchar(100)                    | Y |    |          |
|                  | user_role           | enum('manager','staff','admin') | Y |    |          |
| website settings |                     |                                 |   |    |          |
|                  | option_id           | int(5)                          | Y | PK |          |
|                  | option_name         | varchar(255)                    | Y |    |          |
|                  | option_value        | varchar(255)                    | Y |    |          |
| supplier         |                     |                                 |   |    |          |
|                  | supplier_id         | int(2)                          | Y | PK |          |
|                  | supplier_name       | varchar(30)                     | Y |    |          |
|                  | supplier_number     | int(15)                         | Y |    |          |
| inventory        |                     |                                 |   |    |          |
|                  | grocery_id          | int(2)                          | Y | PK |          |
|                  | grocery_name        | varchar(30)                     | Y |    |          |
|                  | grocery_price       | decimal(10,2)                   | Y |    |          |
|                  | quantity_left       | decimal(10,2)                   | Y |    |          |
|                  | supplier_id         | int(3)                          | Y | PK | supplier |
| feedback         |                     |                                 |   |    |          |
|                  | review_id           | decimal(10,2)                   | Y | PK |          |
|                  | menu_id             | int(2)                          | Y | FK | menus    |
|                  | client_id           | int(2)                          | Y | FK | client   |
|                  | feedback            | decimal(10,2)                   | Y |    |          |
|                  | comments            | varchar(255)                    | Y |    |          |



| payment |            |               |   |    |          |
|---------|------------|---------------|---|----|----------|
|         | payment_id | int(2)        | Y | PK |          |
|         | time       | datetime      | Y |    |          |
|         | amount     | int(2)        | Y |    |          |
|         | order_id   | decimal(10,2) | Y | FK | in_order |

### Presentation Question:

For the attributes, delivered and cancelled, in the placed\_orders table, we used the Tinyint(1) as the datatype instead of Boolean. Boolean uses true/ false conditions to assign a 0 to false and 1 to true. Tinyint can have multiple range of values but in our table is set to just 0 or 1. Using Tinyint offers more flexibility and efficiency to the database. In terms of flexibility, right now the data base is set to display either deliver or not delivered but in the future we could have a third option of on transit or attempting to deliver. For efficiency, Tinyint typically uses only 1byte of storage while Boolean can use more depending on how the database is implemented. While both can be used in this situation, these are the reasons we chose to go with Tinyint as the datatype for these attributes.

## G. QUERIES

**Question 1:** Retrieve the order id and category (dine-in, takeout, and delivery orders)

| order_id | order_type |
|----------|------------|
| 7        | dine in    |
| 8        | delivery   |
| 9        | dine in    |
| 10       | delivery   |
| 11       | delivery   |
| 12       | delivery   |
| 14       | dine in    |
| 20       | dine in    |
| 21       | dine in    |
| 23       | dine in    |
| 28       | delivery   |
| 29       | dine in    |
| 30       | take away  |
| 31       | delivery   |
| 32       | delivery   |
| 33       | delivery   |
| 34       | take away  |
| 35       | take away  |
| 36       | dine in    |

```

SELECT order_id,
CASE
    WHEN order_id = 1 THEN 'dine in'
    WHEN 'Vincent' IN delivery_address THEN 'take away'
    ELSE 'delivery'
END AS order_type
FROM placed_orders;

```

**Question 2:** All delivery orders are supposed to provide a delivery address. All addresses should have a zip code. Retrieve customer names having invalid addresses.

| client_name | client_zipcode |
|-------------|----------------|
| mamilla     | 23             |
| Hermione    | 123            |
| Luke        | 9877           |
| Candice     | 12             |

```
SELECT a.client_id, a.client_zipcode, b.order_id, b.delivery_address
```

```
FROM clients a JOIN placed_orders b ON a.client_id = b.client_id
```

```
WHERE LENGTH(a.client_zipcode) <> 5;
```

**Question 3:** Retrieve all customers who make the same order within the same time slot.

| menu_id | client_ids | client names                    | menu_name         |
|---------|------------|---------------------------------|-------------------|
| 1       | 1,22       | Vincent,mamilla                 | Moroccan Couscous |
| 2       | 1,20,21,22 | Vincent,Pra,Richie Rich,mamilla | Beef Hamburger    |
| 3       | 1,20,21,22 | Vincent,Pra,Richie Rich,mamilla | Ice Cream         |
| 5       | 1,20,21,22 | Vincent,Pra,Richie Rich,mamilla | Coffee            |
| 6       | 1,20,22    | Vincent,Pra,mamilla             | Ice Tea           |
| 7       | 1,20,22    | Vincent,Pra,mamilla             | Bucatini          |
| 8       | 1,20,22    | Vincent,Pra,mamilla             | Cannelloni        |
| 9       | 1,22       | Vincent,mamilla                 | Margherita        |
| 11      | 21,22      | Richie Rich,mamilla             | Moroccan Tajine   |
| 12      | 1,21,22    | Vincent,Richie Rich,mamilla     | Moroccan Bissara  |
| 16      | 1,20,21,22 | Vincent,Pra,Richie Rich,mamilla | Couscous          |

```
SELECT
    t1.menu_id,
    GROUP_CONCAT(DISTINCT t1.client_id ORDER BY t1.client_id) AS client_ids,
    GROUP_CONCAT(DISTINCT c.client_name ORDER BY t1.client_id) AS client_names,
    m.menu_name
FROM
    in_order t1
JOIN
    placed_orders t2 ON t1.client_id = t2.client_id

JOIN
    clients c ON t1.client_id = c.client_id
JOIN
    menus m ON t1.menu_id = m.menu_id
WHERE
    DATE(t2.order_time) = '2024-04-25'
GROUP BY
    t1.menu_id
HAVING
    COUNT(DISTINCT t1.client_id) > 1;
```

**Question 4:** Retrieve the customer id, customer name, and total bill paid for all customers in

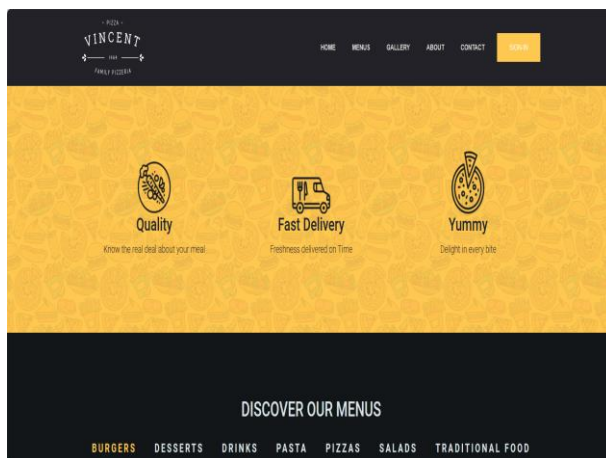
| client_id | total_amount_paid |
|-----------|-------------------|
| 20        | 51                |
| 21        | 71                |
| 1         | 248               |
| 22        | 287               |

ascending order.

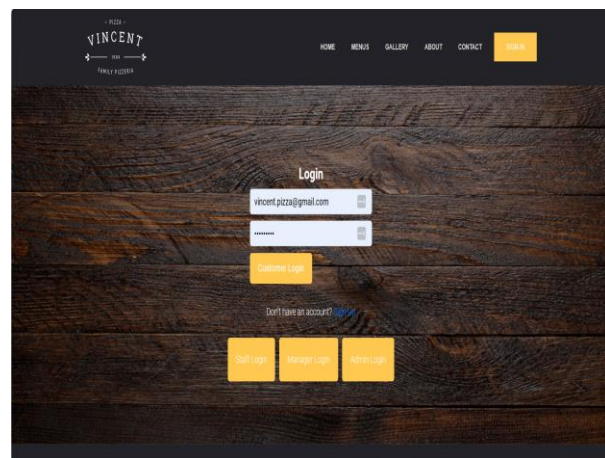
Select c.client\_id, c.client\_name, b.amount as total\_bill from clients c join (select client\_id, amount from placed\_orders a join (select amount,order\_id from payment) b on a.order\_id=b.order\_id) b on c.client\_id=b.client\_id)

## I. PAGES:

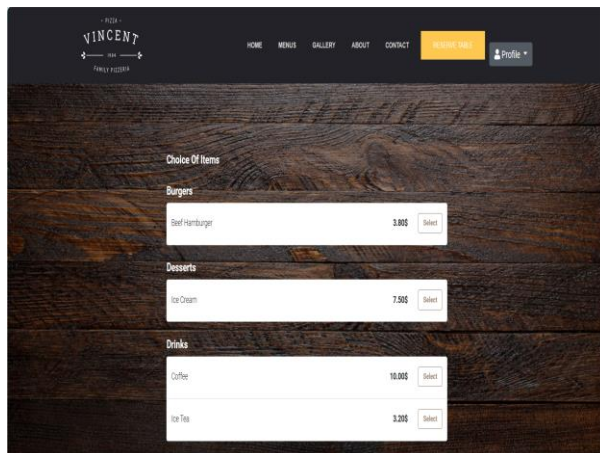
### Home Page:



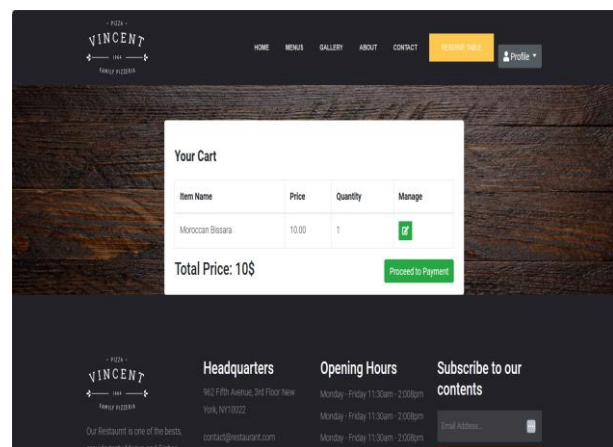
### Customer Login Page:



### Order Page:

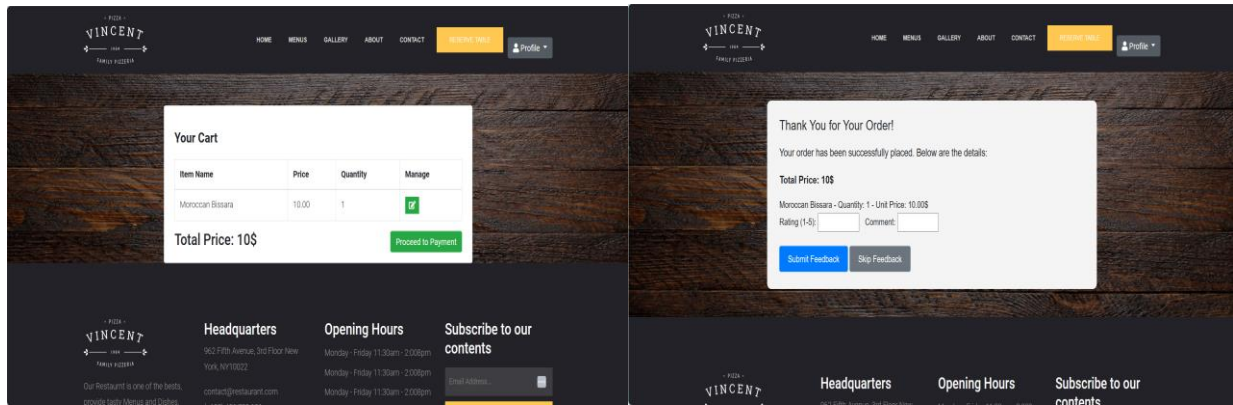


### Cart Page:



### Payment Page:

### Feedback Page:



## Customer details edit page:

**EDIT Profile**

Email:

Name:

Phone Number:

Password:

Street Address:







## Employee Panel:

**Staff Panel**

**Dashboard**

- Dashboard** (9 Clients, 11 Menu, 4 Appointments, 10 Orders)

**Recent Orders**

| Order Time Created  | Selected Menus                                  | Total Price | Client | Manage  |
|---------------------|---|-------------|--------|---|
| 2023-07-01 04:02:00 | Chickens  | 200         | 1      |   |
| 2023-10-30 20:09:00 | Moroccan Bissara<br>Chickens                    | 300         | 2      |   |
| 2023-10-30 21:46:00 | Moroccan Tagine<br>Moroccan Bissara<br>Chickens | 500         | 3      |   |

## **I. CONCLUSION**

In conclusion, we have successfully achieved our objective of developing a comprehensive Restaurant and Food Services Management System. Our system empowers restaurant owners and staff to efficiently manage daily operations, from menu creation to customer orders and reservations. By utilizing PHP and CSS for the frontend, and MySQL and phpMyAdmin for the relational database, we have created an intuitive UI and a well-structured SQL database.

Together, we've acquired the skills to construct a database from scratch using MySQL and phpMyAdmin, alongside crafting a user interface using PHP and CSS code. Additionally, we've mastered the art of linking the frontend and backend components through XXAMP. Throughout this process, our exceptional teamwork and effective communication have been instrumental in achieving our collective goals.

Our Restaurant and Food Services Management System serves as a central hub, allowing proprietors to optimize processes and deliver outstanding dining experiences. With intuitive functionalities and easy-to-navigate interfaces, our platform facilitates smooth communication between restaurant personnel and guests, enhancing overall productivity and contentment.

## **G. FUTURE WORK:**

There are a lot of avenues in which we can expand the database and website for this project. Currently, Implement barcode scanning or RFID technology for inventory tracking to reduce stockouts, prevent over-ordering, and streamline replenishment. Additionally, introduce promotions and loyalty programs to incentivize repeat business and attract new customers.

## **TEAM CONTRIBUTION**

Each team member played a vital role for this project. Each member of the team learned valuable insights from their peers and contributed to various tasks, even those that may not have been explicitly assigned to them.

Bharathwaj Anumula Guru – SQL Queries, Security aspects

Jerusha Gnanamani Bhaskaran – DB verification , Data Dictionary, Documentation,

Keerthi Yadav Eeraboina – Database (SQL), Cohesive tech stack

Vignesh Upputuri – Integration of Frontend and Database (SQL)

Pooja Bejjanki – Testing, SQL code, Documentation

Praneeth Achanta – Database design &ERD

Sharath Kumar – Frontend and project management

Vara Lakshmi Mamillapalli – Frontend and project management

Vasista Tummala – Security aspects and authentication

Victor Paul Buddha – Data integrity, Validation, Queries

## REFERENCES

**Github Link:** [https://uhlibrary.qualtrics.com/jfe/form/SV\\_eKgetxbgNrxcnYu](https://uhlibrary.qualtrics.com/jfe/form/SV_eKgetxbgNrxcnYu)

1. “PHP: PHP Manual - Manual,” *www.php.net*. <https://www.php.net/manual/en/>
2. campcodes.com, “You Searched For Restaurant | CampCodes.”  
[https://www.campcodes.com/?s=restaurant&post\\_type=post](https://www.campcodes.com/?s=restaurant&post_type=post) (accessed Mar. 20, 2024).
3. campcodes.com, “You Searched For Restaurant | CampCodes.”  
[https://www.campcodes.com/?s=restaurant&post\\_type=post](https://www.campcodes.com/?s=restaurant&post_type=post) (accessed Mar. 20, 2024).
4. campcodes.com, “Complete Restaurant Food Ordering System Using PHP MySQL Source Code | CampCodes,” May 30, 2021. <https://www.campcodes.com/projects/php/restaurant-food-ordering-system-using-php-mysql-source-code/> (accessed Apr. 1, 2024).