



CS 457/CS 557 – Database Software Design

Online Menu Management System

GROUP FP G

1.0 Contribution of each member of your group.....	2
2.0 Project description.....	3
3.0 Modeling Scheme.....	4
4. SQL queries.....	7
4.1 SQL queries for table creation.....	7
4.2 SQL queries for Insertion.....	10
4.3 SQL queries for Deletion.....	10
5. Implementation.....	11
6.0 Screenshots of the user interface of your final project.....	12
6.1 User authentication Page.....	12
6.2 Dashboard Page.....	12
6.3 Category management Page.....	13
6.4 Food Menu Page.....	13
6.5 Feedback Review Page.....	14
6.6 User Management Page.....	15
6.7 Orders Listings management.....	15
6.8 Edit Product Listings.....	16
6.9 Food cart Ordering.....	17
6.9 Procced to Checkout Page.....	17
7.0 Conclusion and Future works.....	18

1.0 Contribution of each member of your group

Full name	Contribution
Victor Chukwuma Nwabufo 00234794	<ul style="list-style-type: none">• Requirement analysis• Database schema (DDL)• EER diagram design• Relation diagram• Testing
Ahmad Issa 002230777	<ul style="list-style-type: none">• Requirement analysis• Database schema(DDL)• EER diagram design• Relation diagram design & Testing
Marcus Eke Okwu 002349039	<ul style="list-style-type: none">• Requirement analysis• Code implementation• Relation diagram• Ui/UX design• Testing

2.0 Project description

Our project is all about making online menu management system easier and ordering smoother. We've created a web app that handles menus, tracks ingredients, and lets customers order efficiently. Plus, it automatically alerts staff about any changes in orders or ingredient stock. Managers can also get feedback from customers to improve their service.

Our app is made using PHP and HTML, with PostgreSQL for the database. It has a user-friendly web interface, making it easy to use whether you're on the cloud or using your own servers.

We followed a step-by-step process to make sure everything works smoothly:

- First, we figured out what the project needed.
- Then, we designed the software to meet those needs.
- Next, we built it.
- Finally, we tested everything to make sure it works as it should.

Concerning the needed time for each part, below is a approximate number of hours for each task:

Stages	Timeline	Description
Analysis Requirement	2 days	We did some elicitation requirements that helped us to develop the system requirement specifications.
Designing	3 days	We created a UML and mock ups that guided us in developing the system
Implementation	6 Days	We coded the system using PHP , HTML and Javascript for server side, front end and scripting side respectively.
Testing and Debugging	2 days	At this this, we compare the requirement specification to the solution delivered through a list of test cases.

3.0 Modeling Scheme

From the business requirement specified in our online menu management system, users interact with product lists, place orders, provide feedback through reviews, and have their information stored securely. System settings allow for customization and management of the platform's behavior. Categories help in organizing products for easy browsing, while carts facilitate the shopping experience by allowing users to add and remove items before finalizing their orders. we have several entities:

1. **User:** Individuals interacting with the system, possibly creating accounts, logging in, and making orders.
2. **User_info:** Information associated with users, including personal details, preferences, and order history.
3. **System_settings:** Configuration options and preferences that dictate the behavior of the system.
4. **Review:** Feedback provided by users regarding products or services.
5. **Product_list:** A catalog of items available for purchase or viewing.
6. **Order_list:** A record of orders placed by users.
7. **Orders:** Specific instances of orders made by users, including details such as items, quantities, and delivery information.
8. **Category_list:** Categorization of products to facilitate navigation and organization.
9. **Cart:** Temporary storage for items selected by users before checkout.

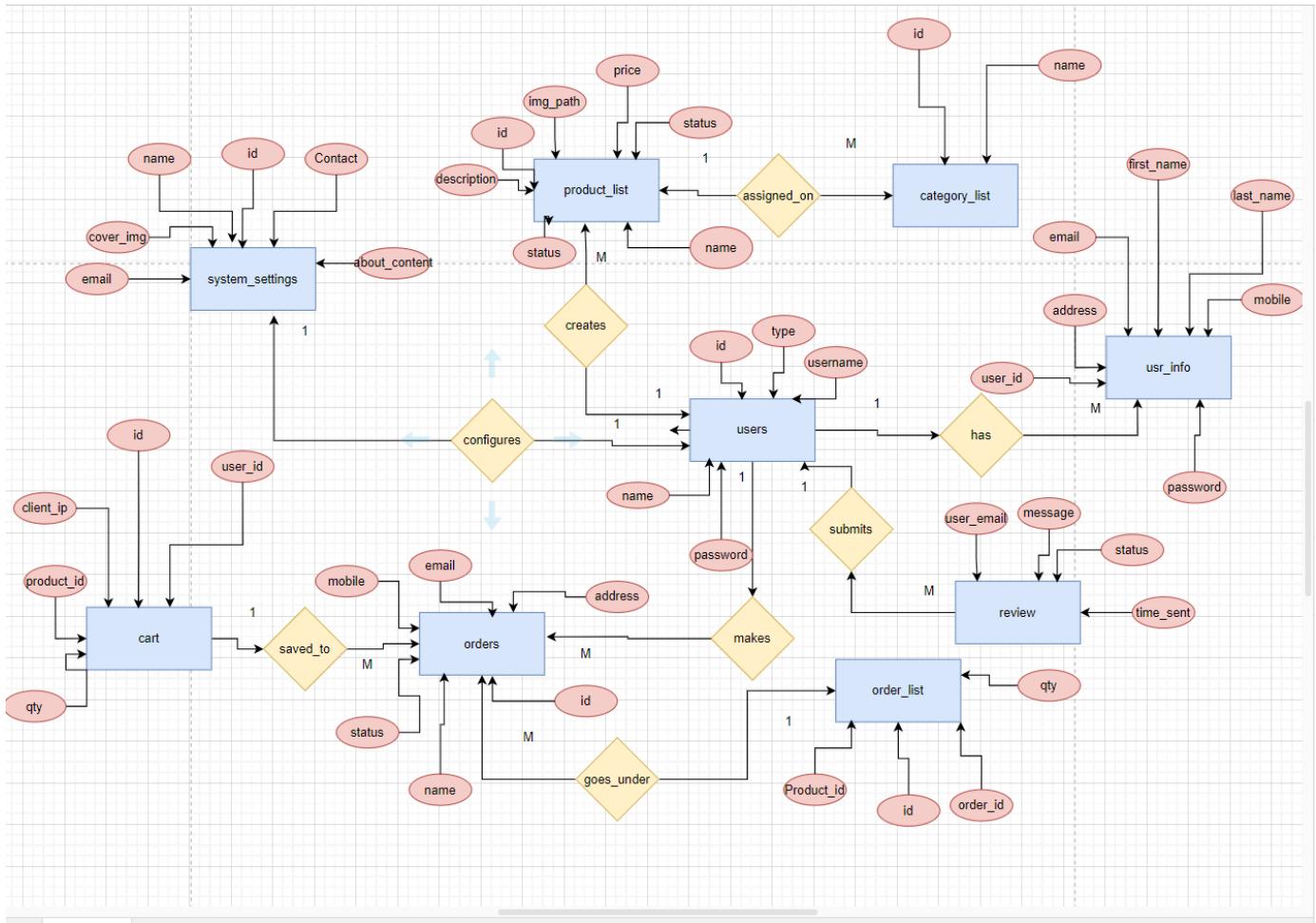


Figure 1: EER diagram of the online menu management system

From the EER diagram, the user entity comprises of both the Staff and the Customer which is differentiated by their roles within this relation. `User` and `User_info` have a one-to-one relationship, as each user has one corresponding user information. `System_Settings` and `Review` are independent entities. `Product_list` is associated with `Category_list` in a many-to-many relationship, meaning a product can belong to multiple categories, and a category can have multiple products. `Orders` represent specific instances of orders made by `users` and are related to `User` (many orders can be made by one user) and `Product_list` (each order consists of multiple products). `Cart` is associated with `User` since each user can have one cart, and it's also linked to `Product_list` since it contains selected items.

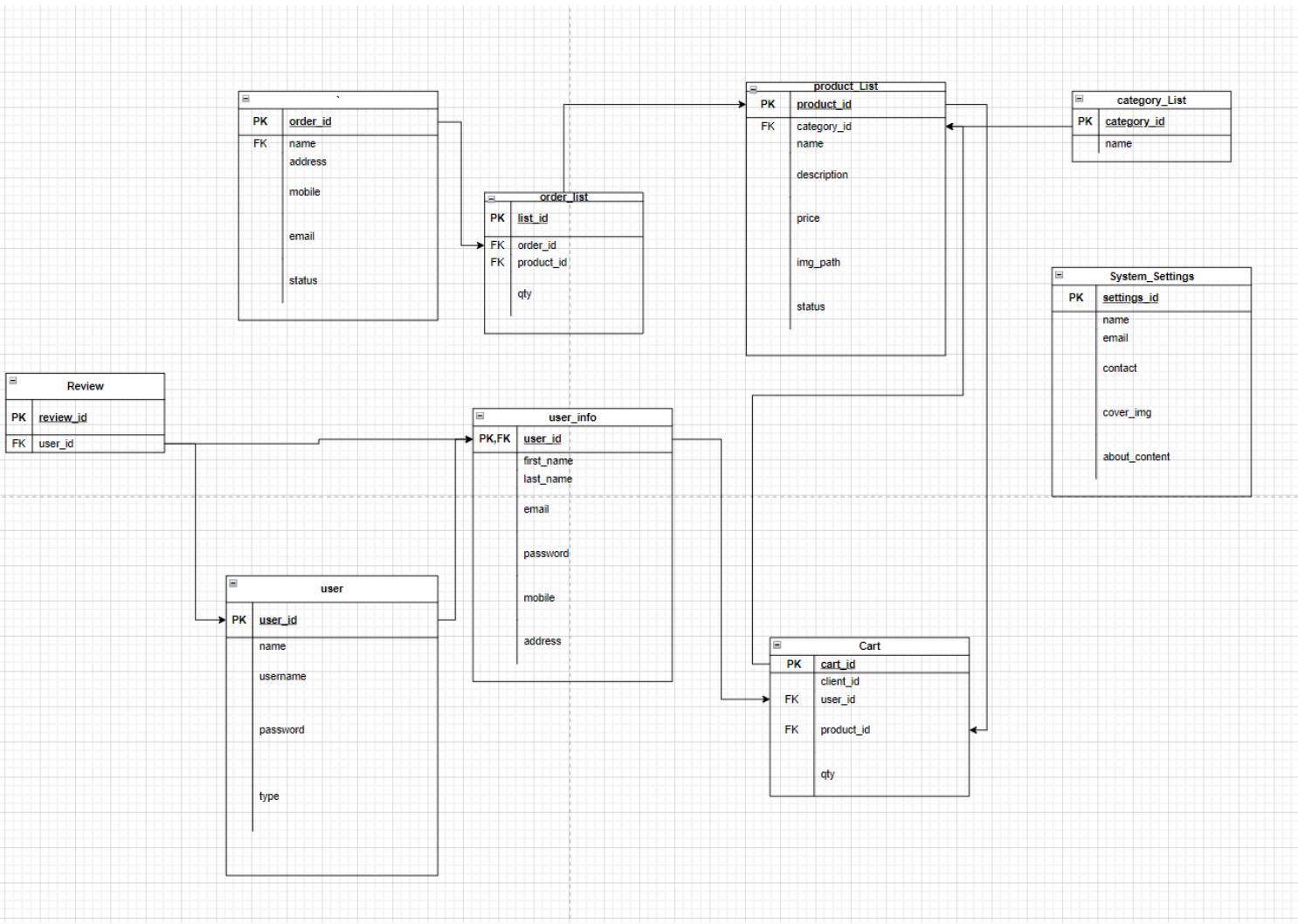


Figure 2: Relational Model of the online menu management system

4. SQL queries

4.1 SQL queries for table creation.

We created tables in POSTGRESQL specifying the data types and lengths of each field with appropriate relationships for data integrity.

Table structure for table *cart*

```
CREATE TABLE cart (
    id serial PRIMARY KEY,
    client_ip varchar(20) NOT NULL,
    user_id int NOT NULL,
    product_id int NOT NULL,
    qty int NOT NULL
);
```

Table structure for table *Category_list*

```
CREATE TABLE category_list (
    id serial PRIMARY KEY,
    name text NOT NULL
);
```

Table structure for table *orders*

```
CREATE TABLE orders (
    id serial PRIMARY KEY,
    name text NOT NULL,
    address text NOT NULL,
    mobile text NOT NULL,
    email text NOT NULL,
    status smallint NOT NULL DEFAULT 0
);
```

Table structure for table orders

```
CREATE TABLE order_list (  
    id serial PRIMARY KEY,  
    order_id int NOT NULL,  
    product_id int NOT NULL,  
    qty int NOT NULL  
);
```

Table structure for table product_list

```
CREATE TABLE product_list (  
    id serial PRIMARY KEY,  
    category_id int NOT NULL,  
    name varchar(100) NOT NULL,  
    description text NOT NULL,  
    price float NOT NULL DEFAULT 0,  
    img_path text NOT NULL,  
    status smallint NOT NULL DEFAULT 1  
);
```

Table structure for table review

```
CREATE TABLE review (  
    id serial PRIMARY KEY,  
    user_email varchar(50) NOT NULL,  
    message varchar(255) NOT NULL,  
    status smallint NOT NULL,  
    time_sent timestamp DEFAULT current_timestamp()  
);
```

Table structure for table system_settings

```
CREATE TABLE system_settings (  
    id serial PRIMARY KEY,  
    name text NOT NULL,
```

```
email varchar(200) NOT NULL,  
contact varchar(20) NOT NULL,  
cover_img text NOT NULL,  
about_content text NOT NULL );
```

Table structure for table users

```
CREATE TABLE users (  
  
id serial PRIMARY KEY,  
name varchar(200) NOT NULL,  
username text NOT NULL,  
password varchar(200) NOT NULL,  
type smallint NOT NULL DEFAULT 2  
);
```

Table structure for table user_info

```
CREATE TABLE user_info (  
  
user_id serial PRIMARY KEY,  
first_name varchar(100) NOT NULL,  
last_name varchar(100) NOT NULL,  
email varchar(300) NOT NULL,  
password varchar(300) NOT NULL,  
mobile varchar(10) NOT NULL,  
address varchar(300) NOT NULL  
);
```

4.2 SQL queries for Insertion.

No.	SQL
1	INSERT INTO user_info(user_id, first_name, last_name, email, password, mobile, address) VALUES(:user_id, :first_name, :last_name, :email, :password, :mobile, :address);
2	INSERT INTO cart(id, client_ip, user_id, product_id, qty) VALUES (:id, :client_ip, :user_id, :product_id, :qty);
3	INSERT INTO review(id, user_email, message, status,time_sent) VALUES (:id, :user_email, :message, :status ,:time_sent);
4	INSERT INTO "order"(id, name, mobile, address, email, status) VALUES (:id, :name, :mobile, :address, :email, :status);

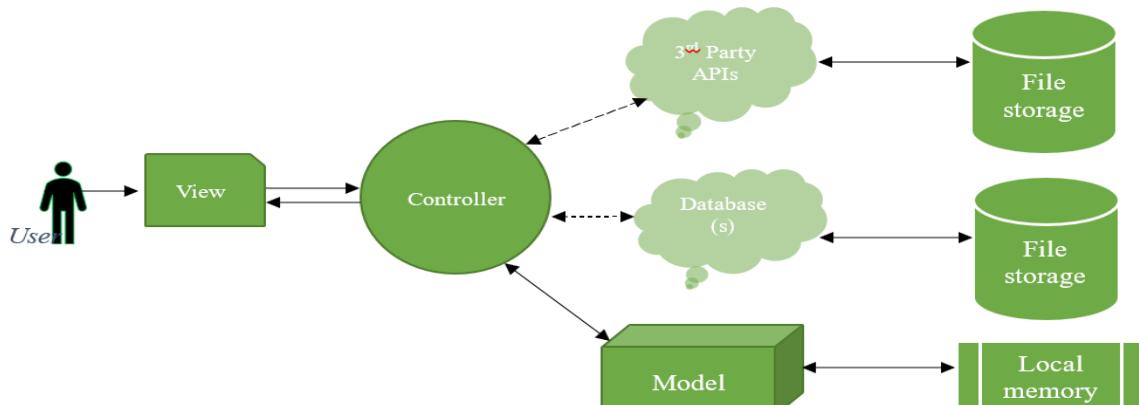
4.3 SQL queries for Deletion.

No.	SQL
1	DELETE FROM “order” WHERE id = order_id;
2	DELETE FROM “order_list” WHERE id=id;
3	DELETE FROM “product_list” WHERE id = id;
4	DELETE FROM "review" WHERE id=id;

5. Implementation

The software design pattern or architecture that we adopted in this project is the Model-View-Controller pattern (MVC). The MVC architectural pattern mostly relates to the UI/ interaction layer of our application. It enables us to break up the back-end and front-end code into separate components. The model's (data) job is to manage the data from a database. The view's (UI) job is to decide what the user sees on their screen, and how. While the job of the controller (Brain) is to pull and provide data to the user. Essentially, the controller is the link between the view and the model. The MVC provides a loose coupling between the presentation layer, data layer, and logic layer and also helped us to implement the SOLID principles for developing good and maintainable software. The diagram below illustrates the model of the MVC architecture.

To facilitate smooth collaboration and track tasks effectively, we relied on GitHub GIT and GitHub Issues/Projects. We hosted all our code on GitHub GIT, which simplified the process of implementing and integrating changes. We followed a trunk-based development workflow to boost our efficiency further. For precise project management, we consistently monitored all tasks using GitHub Issues/Projects. Additionally, we employed several well-established tools including JetBrains IntelliJ IDEA, JetBrains DataGrip, and Docker Desktop to accelerate our implementation process.

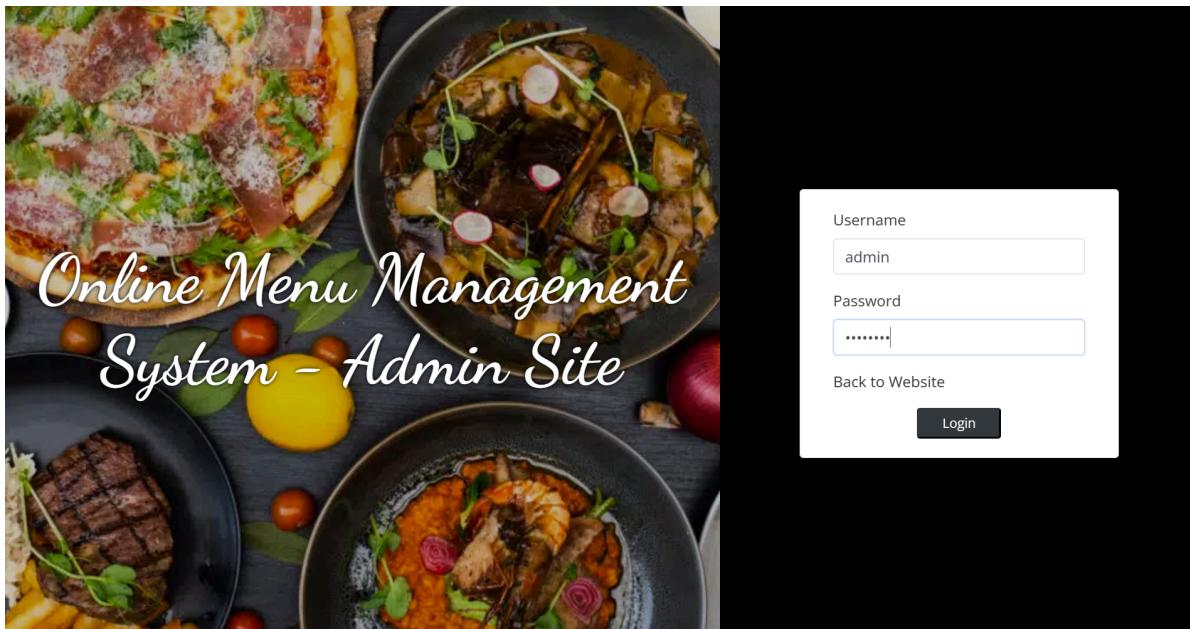


Model View Controller Design Pattern

6.0 Screenshots of the user interface of your final project

6.1 User authentication Page

Users must log in by username and password before accessing any function of this application. The credential (password) validation is done by the Spring Security framework. Our code responds to load user records by username from the database.



No.	SQL
1	SELECT * FROM "user" WHERE user_name = username and password=password;

6.2 Dashboard Page

A screenshot of a web browser showing the dashboard for the "Online Menu Management System - Admin Site". The dashboard has a dark sidebar with navigation links: Home, Orders, Menu, Category List, Reviews, Users, and Site Settings. The main area displays a welcome message "Welcome back Administrator!" and four summary cards: "Total Active Menu" (7), "Total Inactive Menu" (0), "Orders for Verification" (1), and "Confirmed Orders" (3).

Admin Home Page

6.3 Category management Page

The screenshot shows the Admin Home Page of the Online Menu Management System. The left sidebar contains navigation links: Home, Orders, Menu, Category List, Reviews, Users, and Site Settings. The main content area has two sections: 'Category Form' on the left with a text input field and 'Save'/'Cancel' buttons, and a 'Category List' table on the right displaying 10 categories with columns for #, Name, and Action (Edit/Delete). The table data is as follows:

#	Name	Action
1	Beverages	Edit Delete
2	Best Sellers	Edit Delete
3	Meals	Edit Delete
4	Snacks	Edit Delete
5	Dessert	Edit Delete
6	Appetizers	Edit Delete
7	Entrees	Edit Delete
8	Salads	Edit Delete
9	Sandwiches	Edit Delete
10	Soup	Edit Delete

No.	Query	SQL
1	SELECT	SELECT * FROM "category_list";
2	INSERT	INSERT INTO category_list (id, name) VALUES (:id, :name);

6.4 Food Menu Page

This menu page allow the user admin to setup the list of menu items, including the ingredients and amount of food.

localhost/fos/admin/index.php?page=menu

Administrator

			Ingredients :	
1		In Can	Price : \$20.00	Delete
2		Name : Lemon Iced Tea Category : Beverages Ingredients : Sample Price : \$15.00	Edit	Delete
3		Name : Chicken Category : Meals Ingredients : chicken breast, pepper, vegetable oil, mushroom, spices, mayonnaise, Salt. Price : \$150.00	Edit	Delete
4		Name : Steak Category : Meals Ingredients : Sample 2 Price : \$200.00	Edit	Delete
5		Name : Chicken2 Category : Meals Ingredients : Price : \$250.00	Edit	Delete

Menu Ingredients

Available

Category

Appetizers

Price

Image

Choose File No file chosen

Save Cancel

Home Orders Menu Category List Reviews Users Site Settings

No.	Query	SQL
1	SELECT	SELECT * FROM "category_list";
2	INSERT	INSERT INTO category_list (id, name) VALUES (:id, :name);

6.5 Feedback Review Page

localhost/fos/admin/index.php?page=feedbacks

Administrator

#	Email	Review Comments	Status
1	a@gmail.com	Sure, here are various cuisine types from around the world:	Delete
2	marcus@gmail.com	This menu platform is such an incredible platform that helped me navigate my needs as regards to making food choice	Delete
3	ahmad@gmail.com	Thank you guys for creating this awesome and flexible menu system for customer ease of placing orders.	Delete

Home Orders Menu Category List Reviews Users Site Settings

No.	Query	SQL
1	SELECT	SELECT * FROM "review";

6.6 User Management Page

The screenshot shows a web browser window for the 'Online Menu Management System - Admin Site' at the URL 'localhost/fos/admin/index.php?page=users'. The page title is 'Administrator'. On the left, there is a sidebar with navigation links: Home, Orders, Menu, Category List, Reviews, Users, and Site Settings. The main content area displays a table of users:

#	Name	Username	Action
1	Administrator	admin	Action ▾
2	Staff	staff	Action ▾
3	Super Admin	admin1	Action ▾

A blue button '+ New user' is located in the top right corner of the main content area.

No.	Query	SQL
1	SELECT	SELECT * FROM "review";

6.7 Orders Listings management

The screenshot shows a web browser window for the 'Online Menu Management System - Admin Site' at the URL 'localhost/fos/admin/index.php?page=orders'. The page title is 'Administrator'. On the left, there is a sidebar with navigation links: Home, Orders, Menu, Category List, Reviews, Users, and Site Settings. The main content area displays a table of orders:

#	Name	Address	Email	Mobile	Status	
1	Marcus Okwu	J1K 25F	jsmith@sample.com	4756463215	Confirmed	View Order
2	Victor Chukwuma	JkH 6H9	jsmith@sample.com	4756463215	Confirmed	View Order
3	Claire Blake	Sample Address	cblake@mail.com	0912365487	For Verification	View Order
4	Ahmad Issa	Sherbrooke	a@gmail.com	2638811501	Confirmed	View Order

No.	Query	SQL
1	SELECT	SELECT * FROM "review";

6.8 Edit Product Listings

Online Menu Management System - Admin Site

Administrator ⚙

No.	SQL
1	UPDATE product_list SET id=:id, category_id=:category_id, name=:name, description=:description, price=:price, img_path=:img_path, status WHERE id = :id;
2	DELETE FROM product_list WHERE id = :Id;
3	INSERT INTO product_list (id, category_id, name, description, price, img_path, status) VALUES (:id, :category_id, :name, :description, :price, :img_path, :status);

Home Orders Menu Category List Reviews Users Site Settings

Chicken

Menu Ingredients

chicken breast, pepper , vegetable oil, mushroom, spices, mayonnaise , Salt.,

Available

Category

Meals

Price

150

Image

Choose File No file chosen



1  Name : **Diet Coke**
Category : **Beverages**
Ingredients :
In Can
Price : **\$20.00**

2  Name : **Lemon Iced Tea**
Category : **Beverages**
Ingredients :
Sample
Price : **\$15.00**

3  Name : **Chicken**
Category : **Meals**
Ingredients :
chicken breast, pepper , vegetable oil, mushroom, spices, mayonnaise , Salt.
Price : **\$150.00**

4  Name : **Steak**
Category : **Meals**
Ingredients :
Sample 2
Price : **\$200.00**

5  Name : **Chicken2**
Category : **Meals**

No.	SQL
1	UPDATE product_list SET id=:id, category_id=:category_id, name=:name, description=:description, price=:price, img_path=:img_path, status WHERE id = :id;
2	DELETE FROM product_list WHERE id = :Id;
3	INSERT INTO product_list (id, category_id, name, description, price, img_path, status) VALUES (:id, :category_id, :name, :description, :price, :img_path, :status);

6.9 Food cart Ordering

The screenshot shows a web browser window for the "Online Menu Management System". The URL is `localhost/fos2/index.php?page=cart_list`. The page displays a shopping cart with a single item: "Food 101". The item details are: Desc: , Unit Price :55.00, QTY: 2. The total amount is 55.00. There is a "Proceed to Checkout" button. The top navigation bar includes links for Home, Cart, About, Login, and Admin Login.

6.9 Procced to Checkout Page

The screenshot shows a web browser window for the "Online Menu Management System". The URL is `localhost/fos2/index.php?page=cart_list`. The page displays a "Create an Account" form on the left and a shopping cart summary on the right. The shopping cart summary shows a total amount of 55.00 and a "Proceed to Checkout" button. The "Create an Account" form fields include: Firstname, Lastname, Contact, Address, Email, and Password. The top navigation bar includes links for Cart, About, Login, and Admin Login.

No.	SQL
1	INSERT INTO user (id, firstname, lastname, contact, address, email, password) VALUES (:id, :firstname, :lastname, :contact, :address, :email, :password);

7.0 Conclusion and Future works

In summary, we cannot directly use business concept names as database table names because the database management system often designates such terms as keywords, like 'user' and 'order.' Although escaping these reserved words allows their use as table names, it's preferable to select names not reserved. Additionally, maintaining and debugging complex queries proved challenging, and these queries can lead to performance issues in data-heavy scenarios. As a result, we advocate for using simpler queries whenever feasible. We also noticed redundant coding in creating CRUD functionalities for entities due to similar patterns. To address this, we recommend adopting established object-relationship mapping (ORM) tools such as JPA, Hibernate, and MyBatis, which help eliminate repetitive code and enhance the efficiency of database application development.

Also, we intend to integrate a payment model on the online order cart that can allow the customers to make payment from their comfort and also we would integrate a reminder /notification alert that notifies the admin of orders and customers when order is confirmed using CRON JOBS but this requires that this system is hosted on an active server host.