**ABSTRACT**

**TITLE :** Online Food Ordering System ( Fooderia )

**TEAM MEMBERS :**

1. Bharathi A - 18PD05
2. Shreenidhi N - 18PD33

**LANGUAGES USED :**

**DATABASE : MySQL**

**FRONT END : HTML, CSS, JAVASCRIPT**

**LOCAL HOST : PHP**

**SOFTWARES USED :**

**XAMPP, VISUAL STUDIO CODE(OR NOTEPAD++)**

**DESCRIPTION :**

Our project , titled ‘ **Online Food Ordering System**’ aims to simulate the working of a **database centric** web application to order food. A connection between the database and the User Interface is done and the User Interface interacts with the database efficiently, by means of Stored Procedures and Functions. There are a number of functions in our project .

They include :

* A Sign up System for Restaurant Admin and Customers.
* A Login System for Customers and Restaurant Employees.
* A Cart System to add items that includes Addition of items , Quantity Specification and Removal of Items
* A virtual payment system
* A facility for the Customers to share their reviews
* A Logout System
* A Session based approach
* Hiring of employees
* Addition of products to Restaurant menu
* A Mini Customer Analysis

The Login and Logout System are backed by Sessions and Session variables

**ENTITY RELATIONSHIP (ER) DIAGRAM**

**RELATIONSHIPS :**

1. Customer to Order - One to Many

2. Order to Menu - Many to Many

3. Order to Bill - One to One

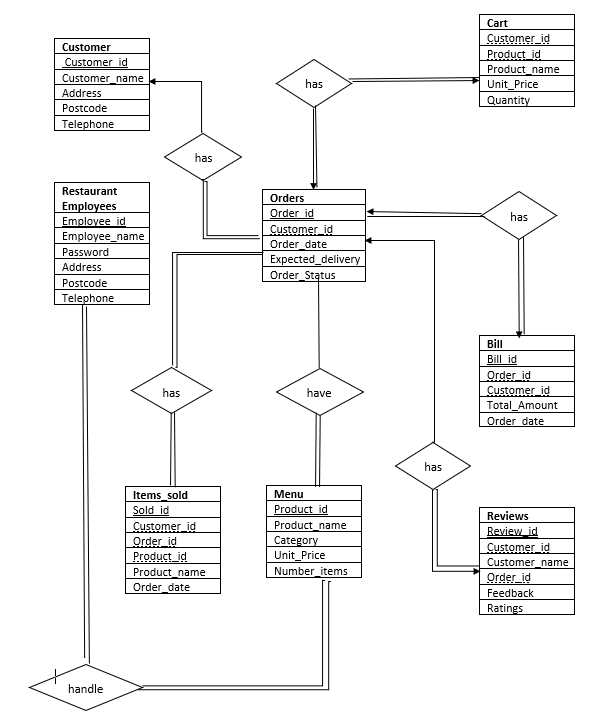
4. Order to Cart - One to One

5. Order to Items\_Sold - Many to Many

6. Order to Reviews - One to One

7. Restaurant\_employees to Menu - Many to Many (Modifies or Handles)

**ENTITY RELATIONSHIP (ER) DIAGRAM**



We have made use of 12 PHP Files to support the User view and to connect with the database.

They are as follows :

1. **index.PHP**

The first page , Index page serves two purposes :

* To Sign up as a new Customer/Restaurant Admin
* To Login as a new Customer/Restaurant Employee

1. **admin.PHP**

Only visible to the Restaurant Admin. Enables the Admin to either Add new products/ Add new Employees.

1. **03\_menu.PHP**

Displays a list of items and enables the customer to add products of different quantities into the cart via a stored procedure .

1. **cart.PHP**

Displays the items in a cart via a tabular format with the Total Price of each product and the Overall Total . It also has an option to remove any item from the cart .

1. **payment.PHP**

Enables confirmation of payment by populating tables such as Orders,Bill and Items Sold . The Order and Delivery time are also set . Cart is cleared at this stage

1. **feedback.PHP**

Enables the current customer to enter feedback via a stored procedure.

1. **toconnect.PHP**

Establishes a connection between the database and the front end (User view)

1. **session\_clear.PHP**

Empties the cart and destroys the current session of the customer, employee or admin.

1. **Bill Analysis.PHP**

A graph is plotted with the Total Amount against the Bill No

1. **Frequent items.PHP**

A graph is plotted with the Frequency of each item bought in the restaurant.

1. **Customer Analysis.PHP**

A page that calls and displays two graphs that are analyzed based on the collected data for the restaurant’s benefit.

1. **Thankyou.PHP**

Displays the Thank you message for the customer at the end of the ordering.

**PROCEDURES AND FUNCTIONS USED :**

1 logincheck

2 getPid

3 intoCart

4 enterFeedback

5 enterEmploy

6 getCname

7 addToOrders

8 getOrderID

9 insertIntoBill

10 Sign\_Up

11 addToItemsSold

12 removeFromCart

13 INSERTMENU

14 addToOrders

15 intoCart

16 createGraph

17 billsGraph

18 calculateTotal

(All the MySQL Codes are in a separate file named “FINAL CODE.docx”)