**National University of Computer & Emerging Sciences**

**Karachi Campus**



**HU TAO RESTAURANT**

**Group Members:**

**Saud Ahmed Abbasi 19K-0229  
Arun Jai 19K-1437**

**INSTRUCTORS:**  
**Zulfiqar Ali Memon  
Erum Shaheen**

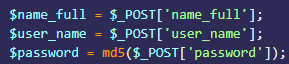
HU TAO RESTAURANT:



# ADMIN PORTAL:

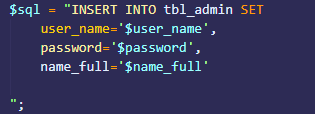
## Add New Admin:

1. Get the Data from form



Password Encryption with MD5 Used.

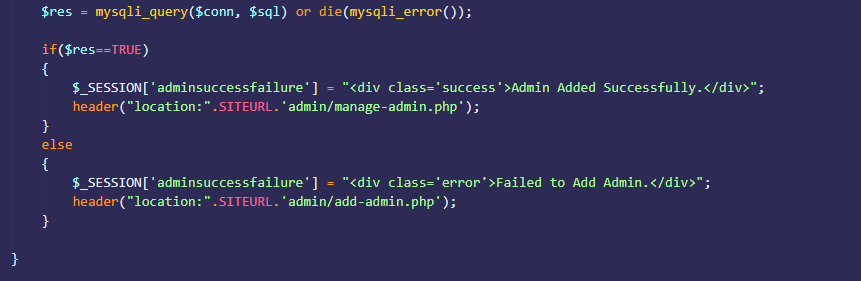
1. SQL Query to Save the data into database

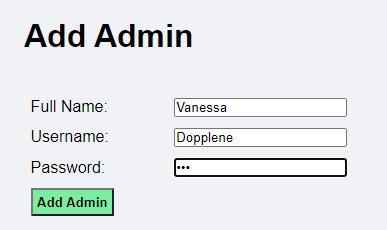


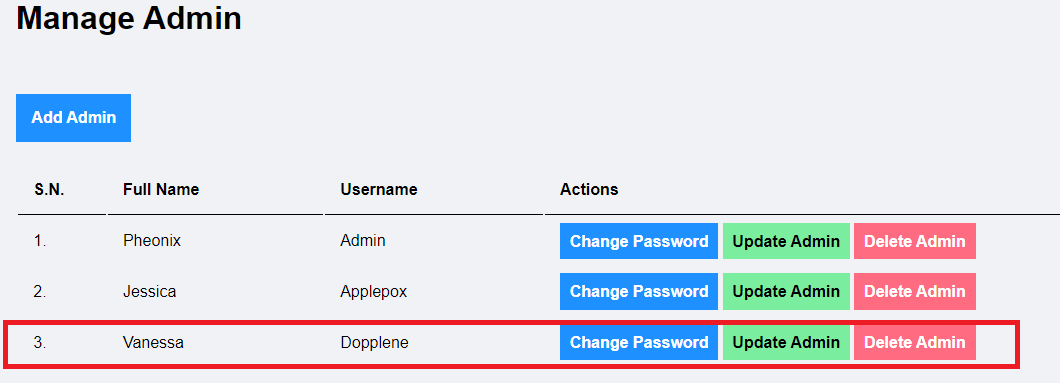
1. Executing Query and Saving Data into Database



1. Check whether the (Query is Executed) data is inserted or not and display appropriate message





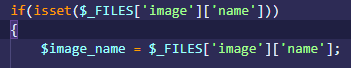


## Admin Category:

1. Check whether the image is selected or not and set the value for image name accordingly (You may use print\_r($\_FILES['image']); to check)

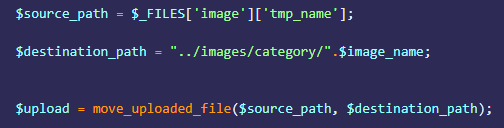


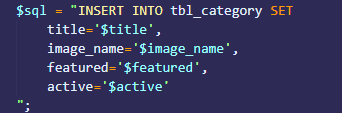
1. Upload the Image   
   (To upload image we need image name, source path and destination path)

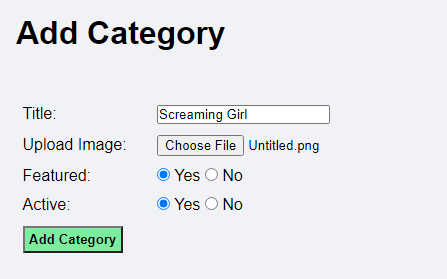


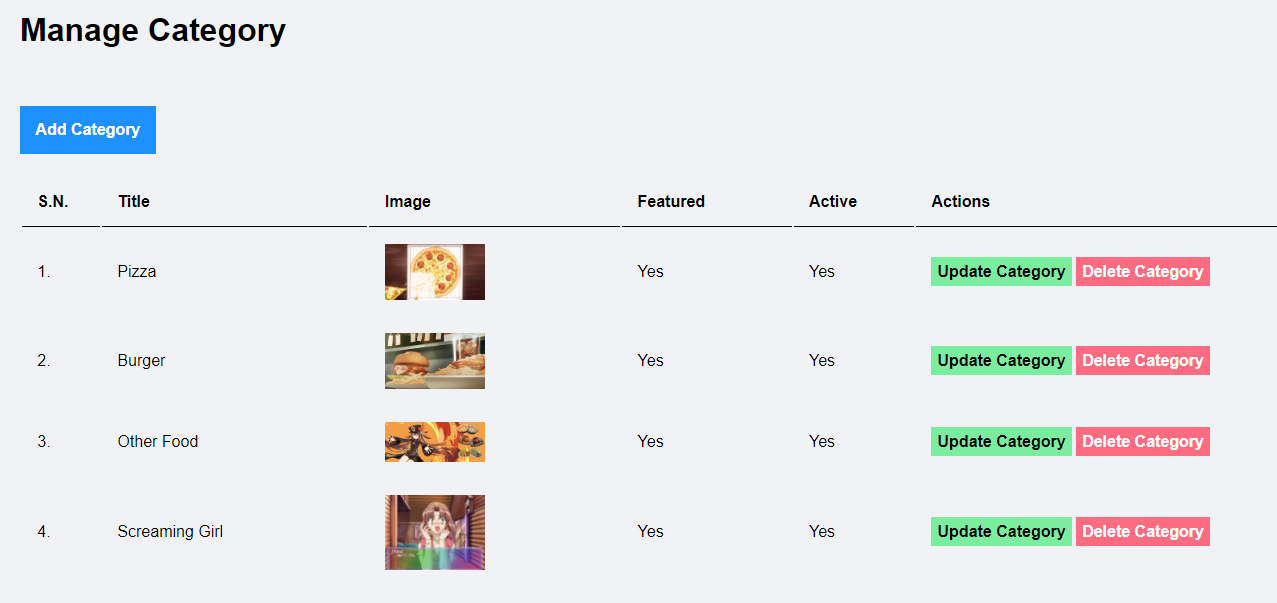
1. Auto Rename our Image  
   1. First get extension:  
     
   2. Rename the Image (Use number randomizer to set unique name)

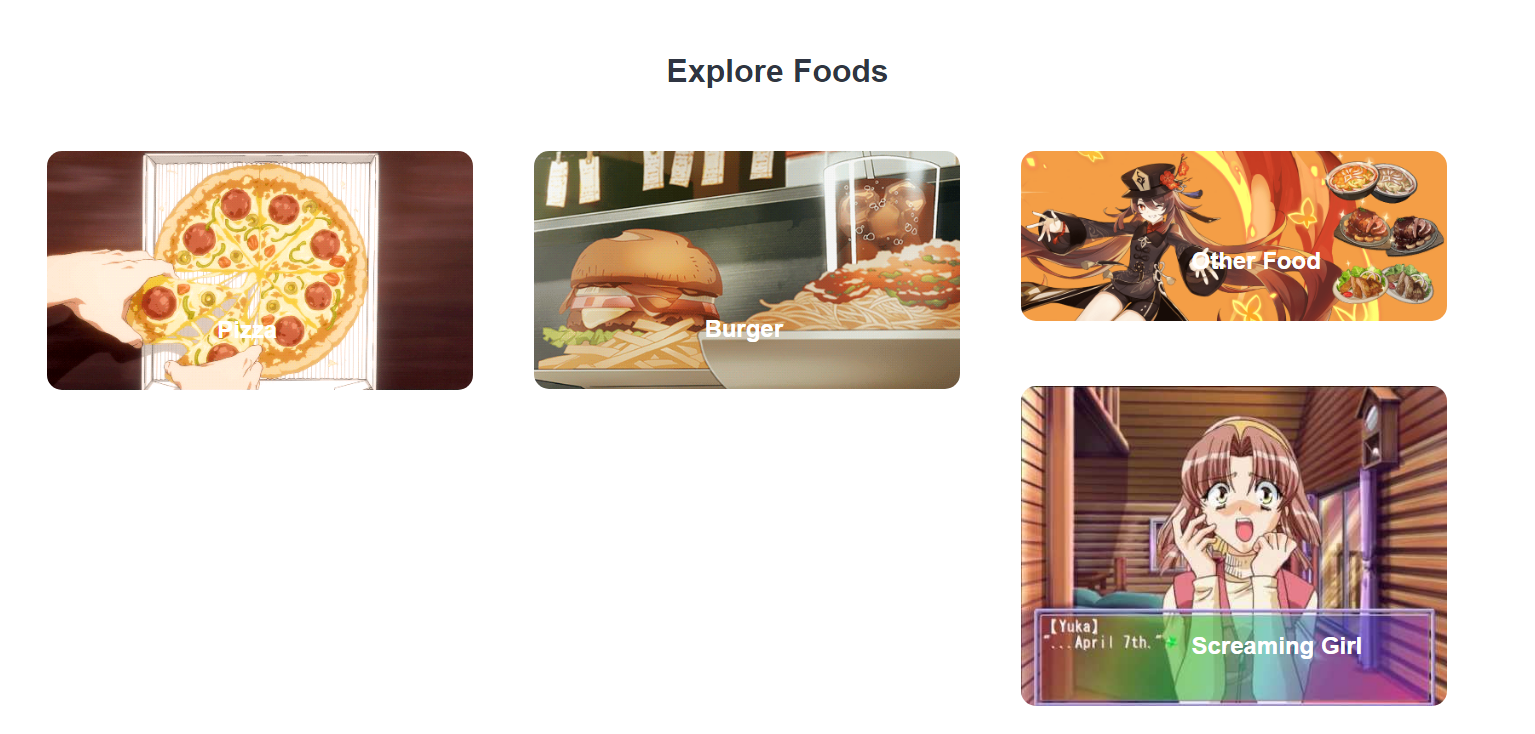


1. Finally Upload Image:  
   
2. Create SQL Query to Insert Category into Database









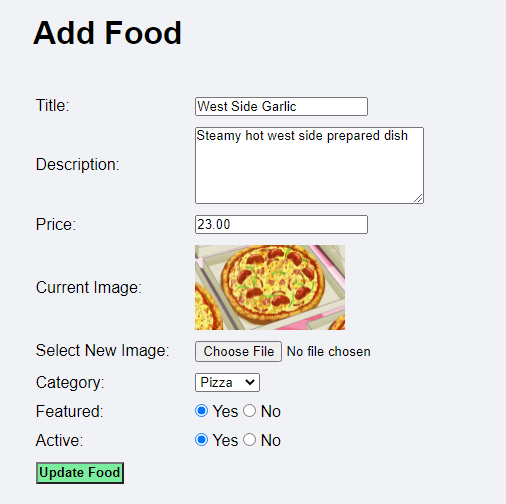
## Add Food:

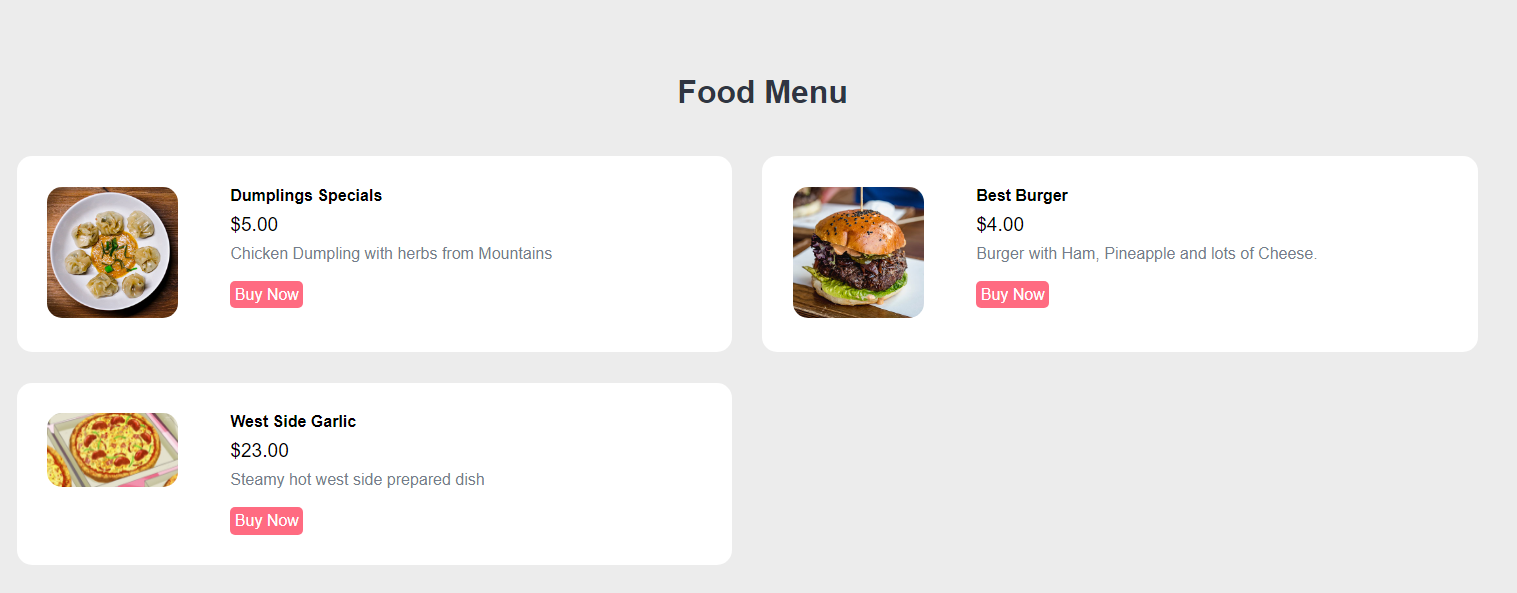
1. Create PHP Code to display categories from Database

1. Create SQL to get all active categories from database  


2. If count is greater than zero, we have categories else we do not have categories

1. Add food in database after getting form data:  
   Get data from form:  
     
   Image setting same as category:  
   

How it looks in form:  


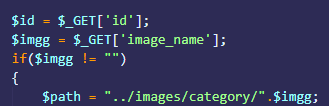
Official site result:  


## Admin Deletion:

1. Simply GET ID and then use deletion query:  
   

## Admin Category Deletion:

1. If image is available remove it



1. Remove Image



1. For the rest of the data simply run deletion query  
   

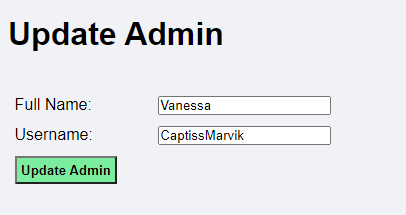
## Admin Food Deletion:

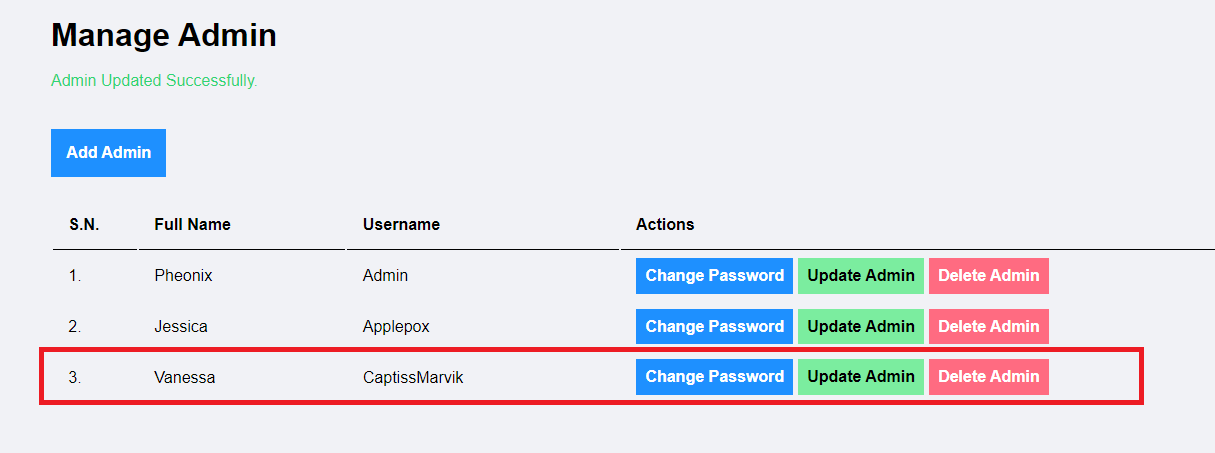
1. Get ID and image name, remove image first using same method as done in category:  
   
2. Use delete sql query for the rest of the data:  
   

## Admin Update:

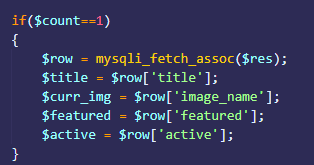
1. Get already set details from database:  
   
2. Set New Details







## Admin Category Update:

1. If category data available, fetch data, put image name inside (current image) variable:  
   
2. Set new image details

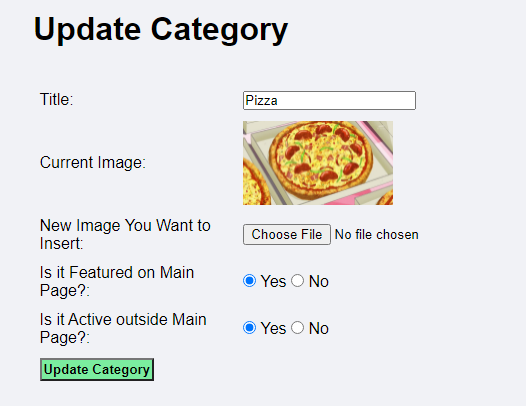


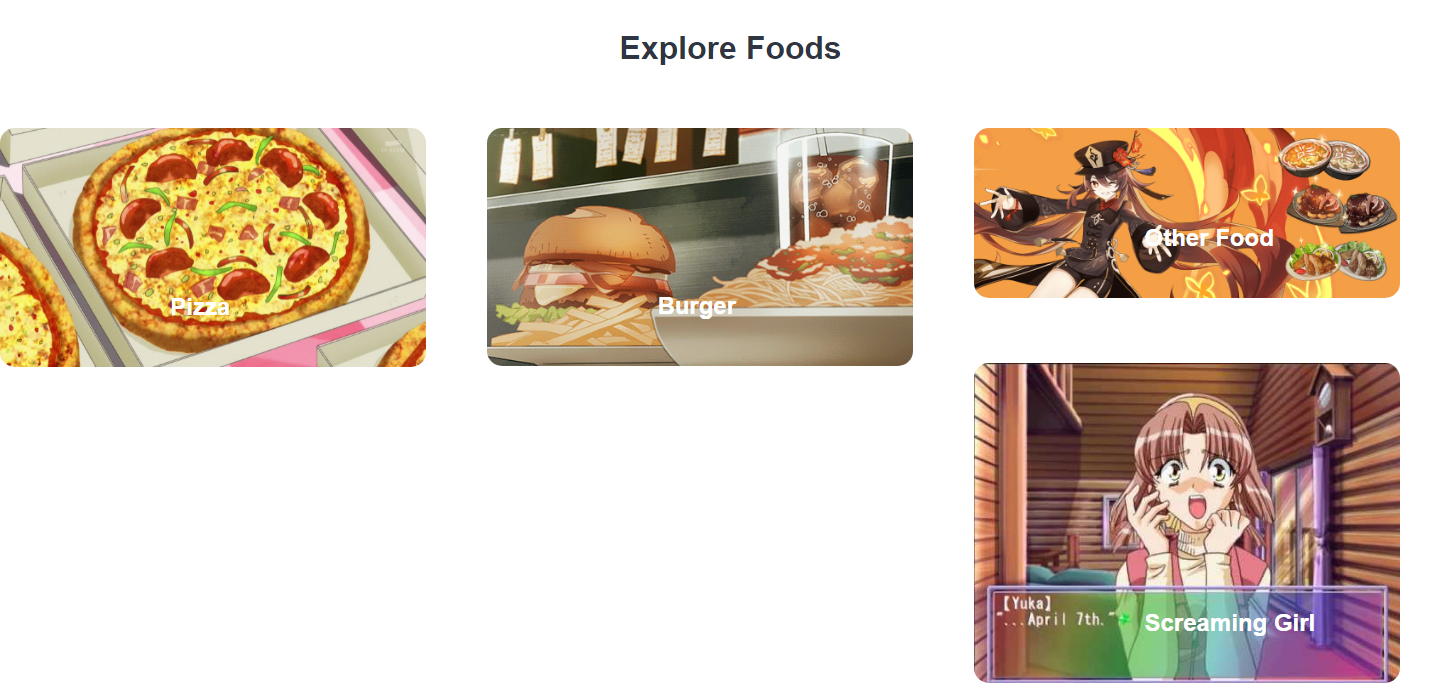
1. Delete current image (the one to change) on page:



1. Update rest of the data through query and data received from form







## Admin Food Update

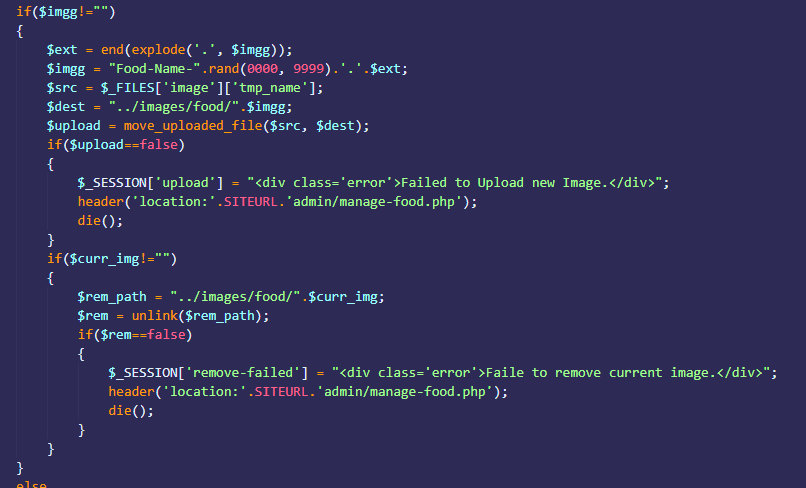
1. Get Food Information, store current category and image into temporary variables

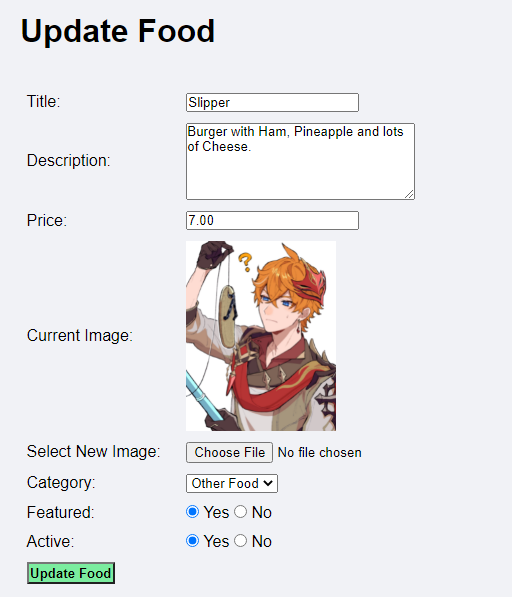


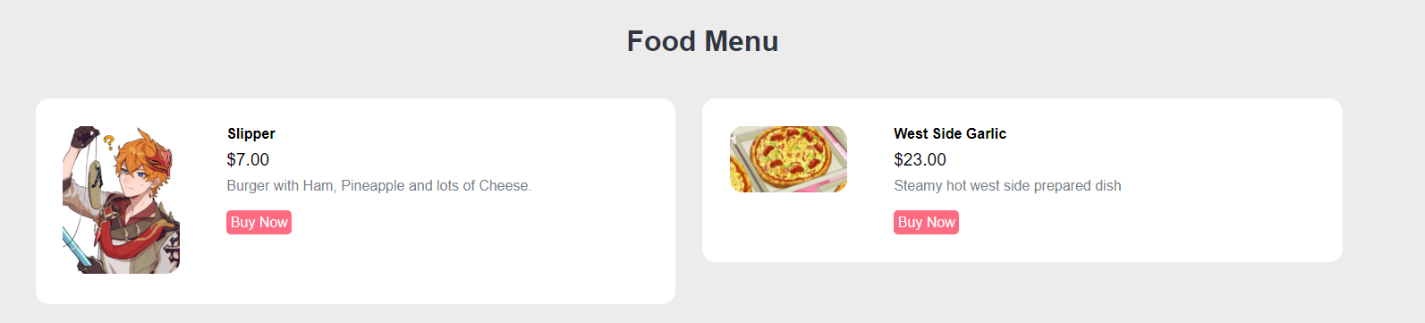
1. Set new category within form selection



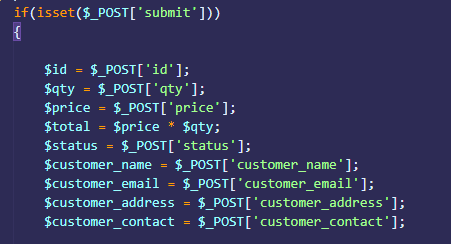
1. Image updating the same as done in category updation

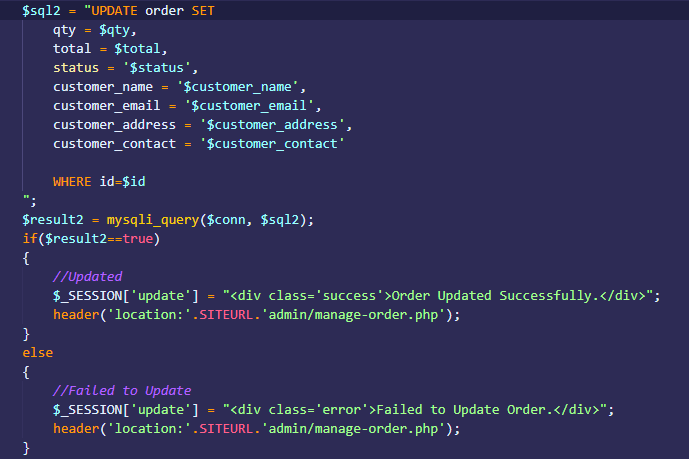


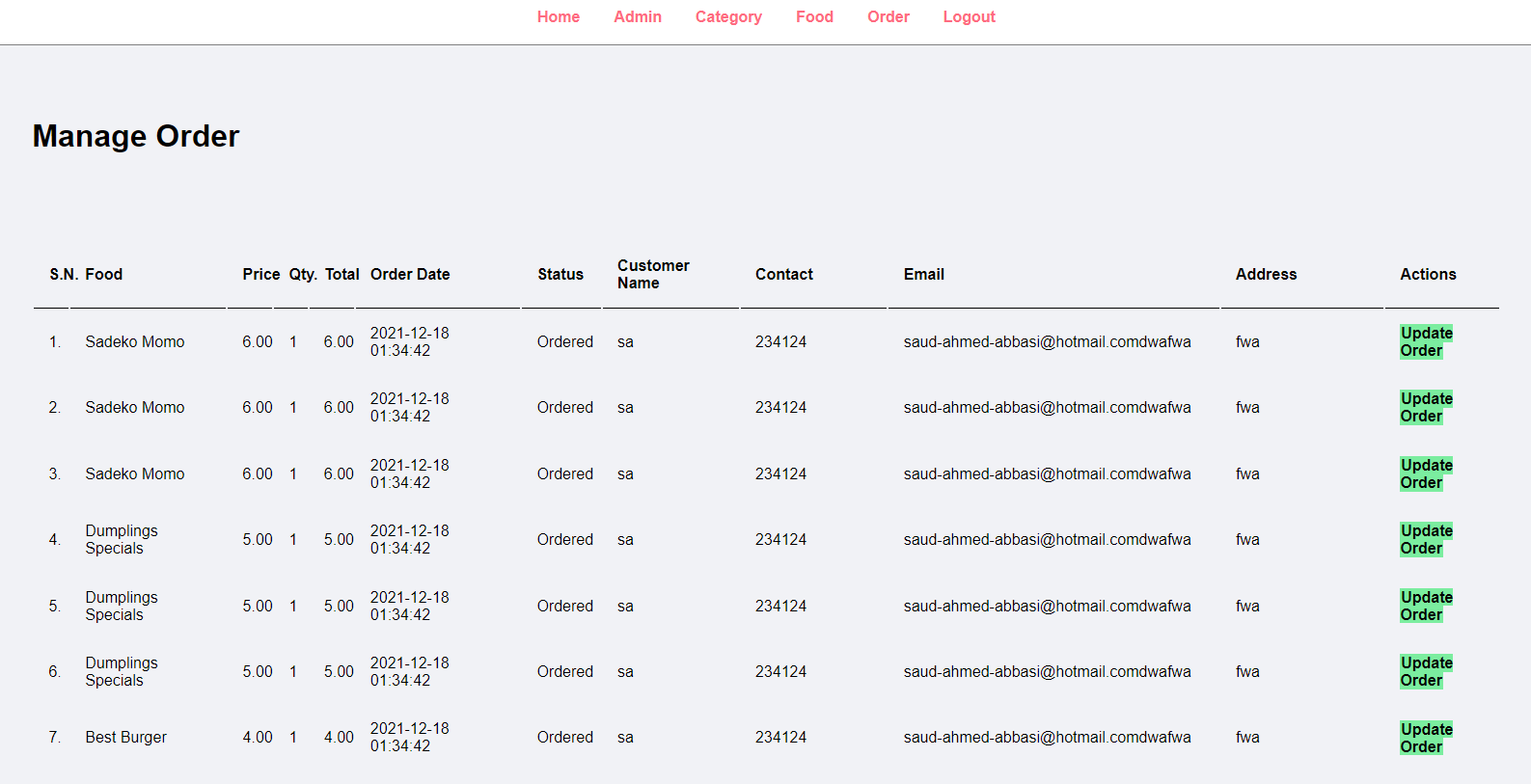




## Check And Update Orders:

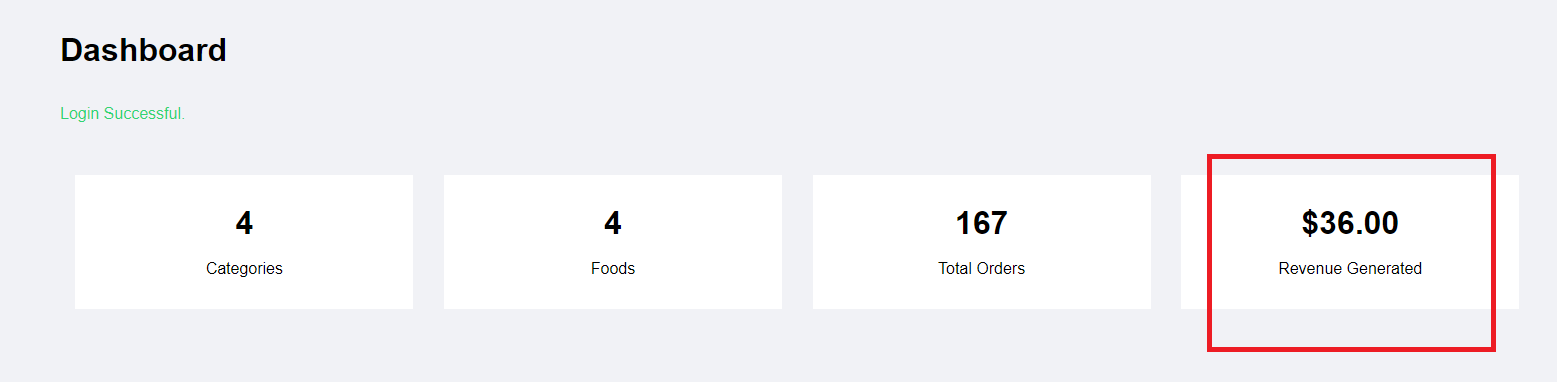
1. Get all order records:  
   
2. Simply get new order details from form and insert with query



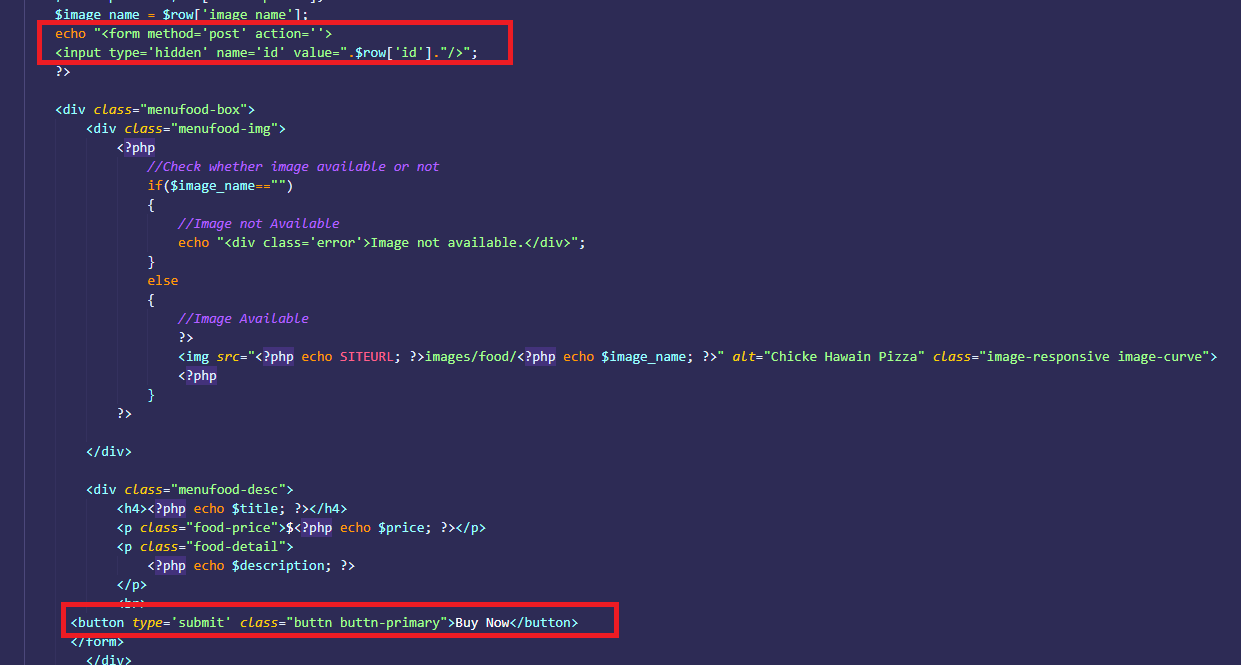
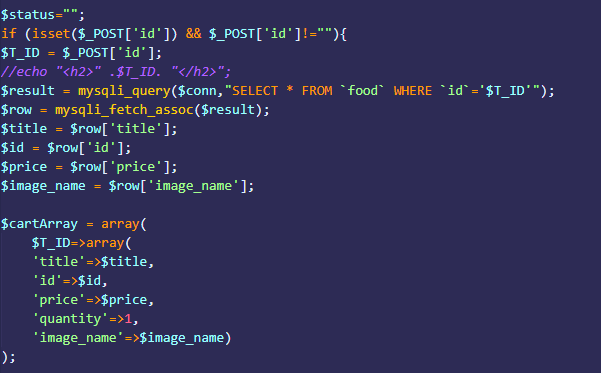


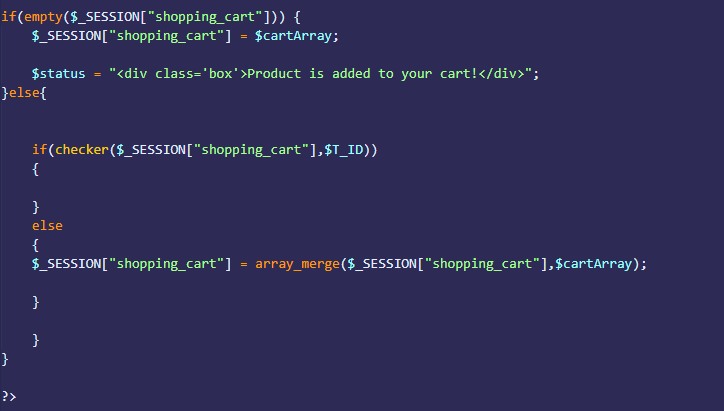


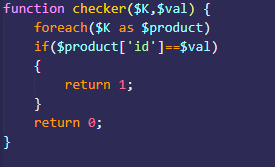
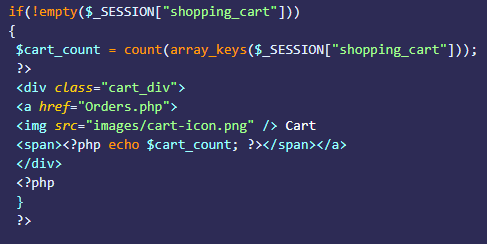
Using Aggregate function in SQL to find total revenue:  

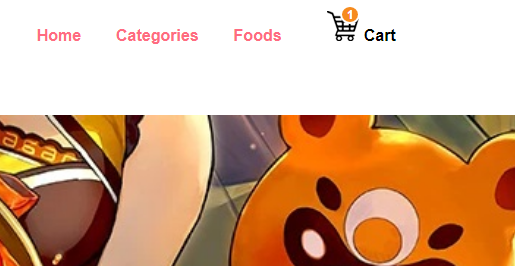



# Customer Side: Cart:

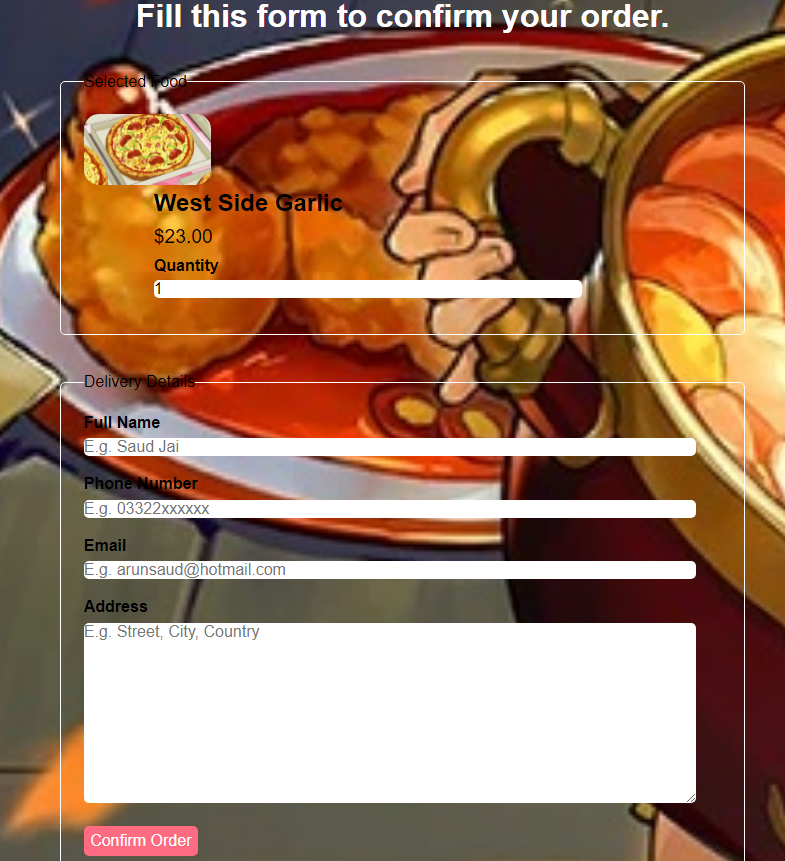
The form has the following declaration for the button and imaging of the Featured Food section on the main page:  
  
What this does is, it $\_POST’s the ID so it can be collected from the form and be processed in the file “cart.php”.  
Upon receiving the ID, the rest of the food data is fetched from the database into a temporary array called $cartArray:  


This $cartArray is then merged with the global array $shopping\_cart:  


So that no food item is repeated twice, a function checker is defined as follows:  
  
  
In the file “menu.php” if the global variable $shopping\_cart is not empty then it styled to show a small image of a cart icon with the number of items you have picked up:  


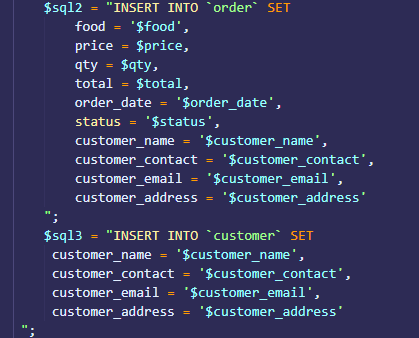
Visual:  


## Placing Order:



Once the customer is satisfied with the items they’ve picked they can fill the form above, the $shopping\_cart we defined earlier will be used to fetch all the data items the customer has picked up:  


To send the quantity unique to a food item, we use a unique trick, we use the product[id] to send the quantity as our shopping cart already contains the ID, the item $\_POST[“id”] will have will be the quantity only for that specific food ID.  


Rest of the data can simply be fed into the sql database as follows:  


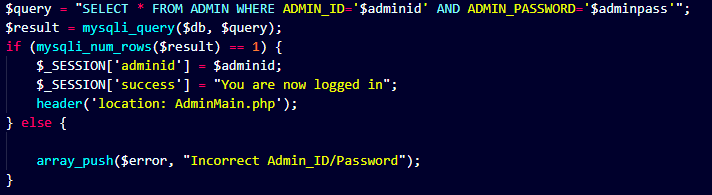
# Staff Side Report

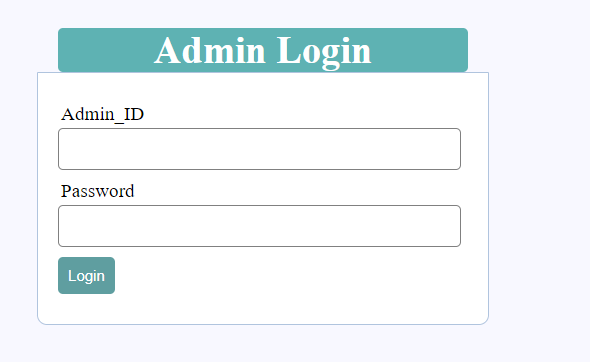
## Admin Login

Extracting Data from Front End:



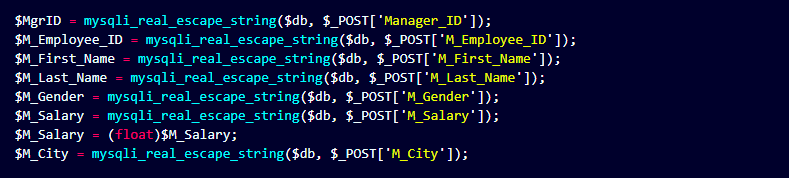
Checking From Data Base:





## Adding Staff Members

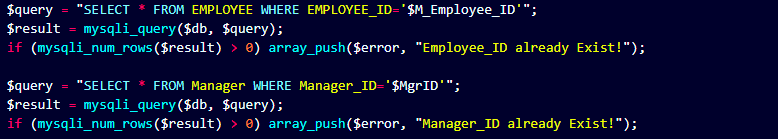
Extracting Data from Front End:



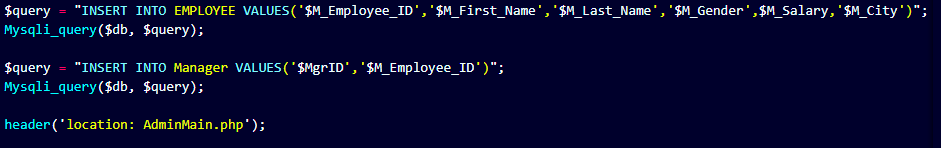
Checking the format of input:

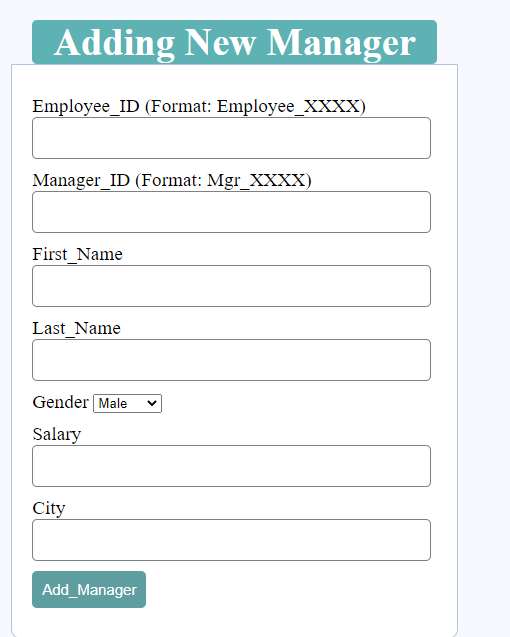


Checking whether the ids already exist:

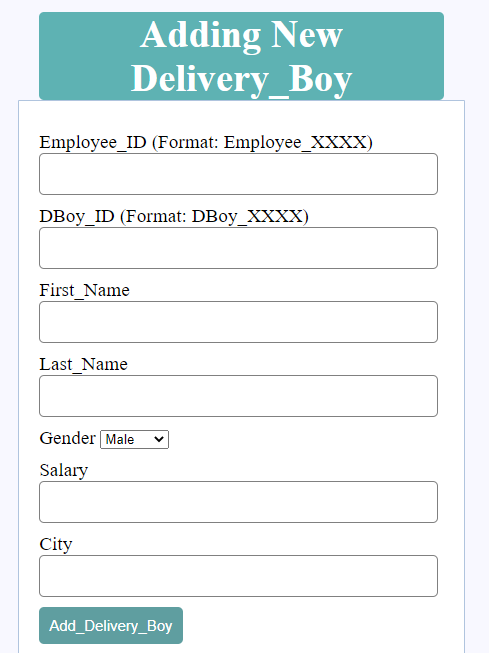


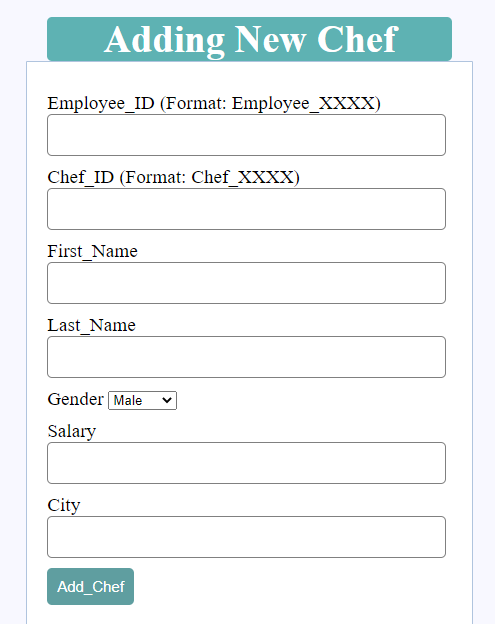
Inserting:





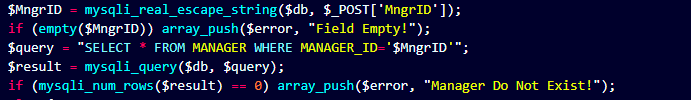
Same way other staff members are added as well, with different id formats.



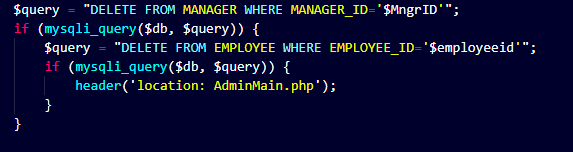


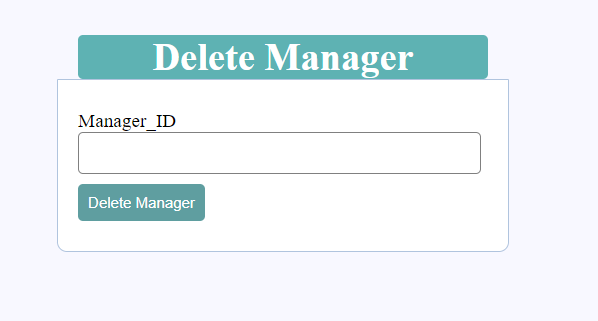
## Deleting Staff

Extracting and checking:

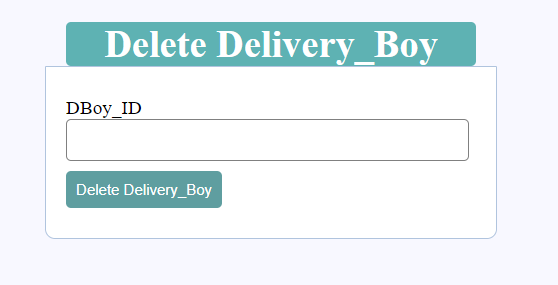


Deleting From Manager and Employee Table:





**Similar work is done for other staff members.**

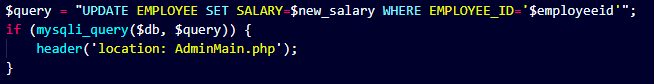


## Updating Staff Salary

Extracting data from front end and Checking:

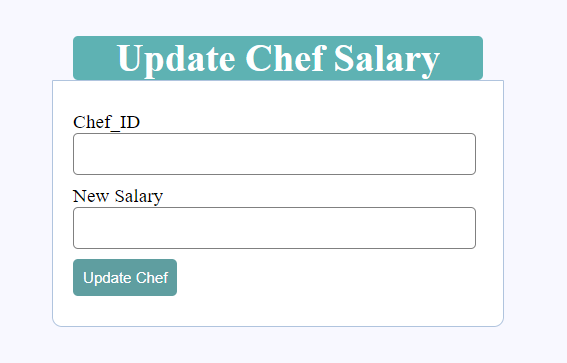


Updating Salary:





**Similar this is done for other sorts of employees**

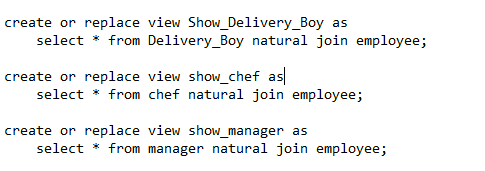
## Displaying Staff Tables (Use of Views and Joins)

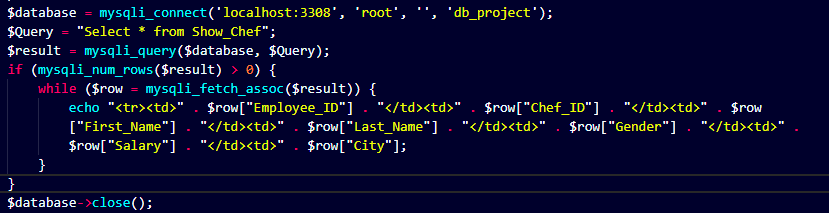
To show complete details of the Staff members there are 3 types of views created in the data base.

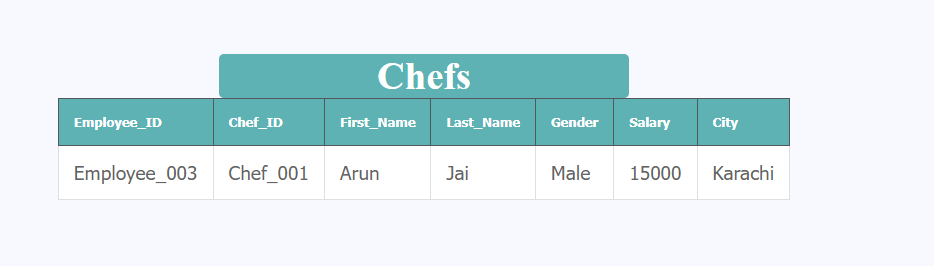


These view contain the join queries on different staff tables and employee table in order to print the whole information of the staff members.

**Sql Code for Creating Views with Join query:**







Similar Work is done for other staff tables.

