Aissce 2023-24

Computer science project

Restaurant management system

Certificate

This is to certify that Samriddhi Shah, Roll No. \_\_\_\_\_\_\_\_\_\_\_\_\_, Class XII Science, Year 2023-2024 of Sushila Birla Girl’s School, Kolkata has successfully completed the project titled Restaurant Management System.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Internal Examiner) (External Examiner)

Acknowledgement

I, Samriddhi Shah, would like to express my sincere gratitude towards our computer teacher, Ms. N. Pasari for her inspiration and support in the completion of this project.

Contents

1. Introduction
2. Hardware and Software used
3. Project Design

* Front End
* Back End

1. Technical Design
2. Source Code
3. Output
4. Bibliography

Introduction

“Restaurant Management System” aims to enable the user to manage tasks like storing and manipulating data that are relevant in terms of restaurant management. It is a comprehensive software designed to optimize restaurant operations.

This project features a user-friendly menu driven system designed for effective management of various aspects related to food, restaurant owner and customers.

It manages tasks like categorizing food, handling employee details, handling product details along with their respective suppliers, processing orders and generating the bill. It is a user-friendly interface that provides quick access to the required information.

The category, employee, supplier, product and food details incorporate different modules which can be used for various purposes like inserting, updating, searching, deleting and displaying its required details.

The application allows customers to order food from the available items. After the order is placed, a waiter is assigned to the customer and the bill is generated.

The application provides with an array of reports to help the owner to check product details, food details and details of employees working in the restaurant.

The modular design, backed by a SQL database, ensures clarity and efficiency by providing a robust framework for day-to-day restaurant operations.

The Restaurant Management System Software is a comprehensive solution designed to revolutionize the way restaurants operate.

Hardware

* Monitor
* Keyboard
* CPU
* Mouse
* Printer: Laser Jet
* Processor: Intel Core i5
* Processor Speed: 2.60 GHz

Software

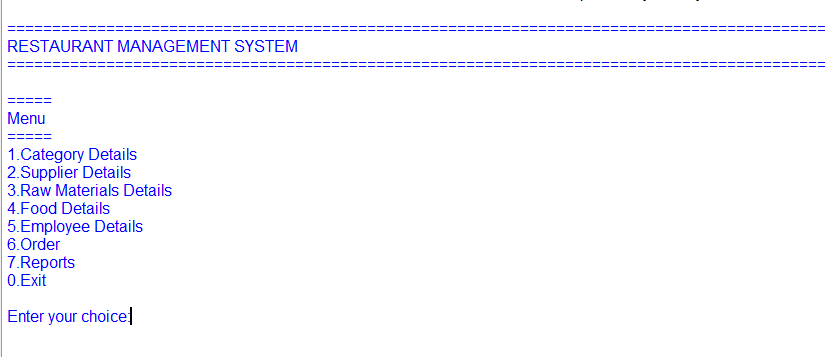
* Operating System: MS Windows 10 Professional
* Front End: Python 3.10.4
* Back End: MySQL 5.5 Server

Project Design

Front End

User Interface

**MODULE 1: MAIN MENU**

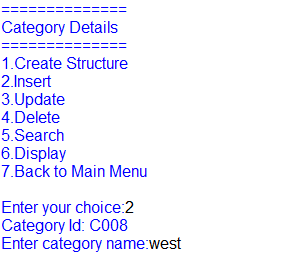


**MODULE 2: CATEGORY**

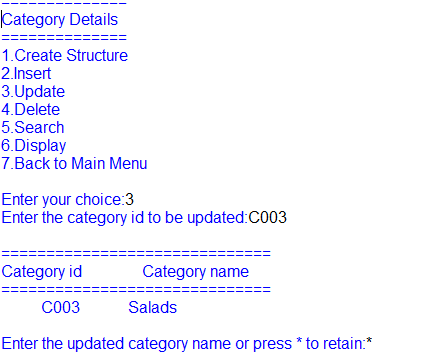
**Create Category**



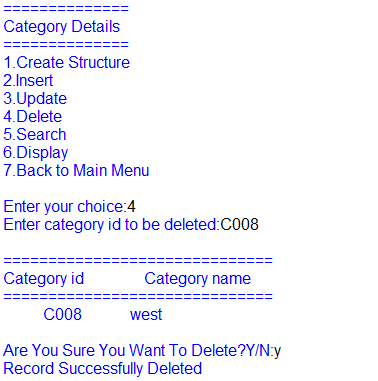
**Insert category**



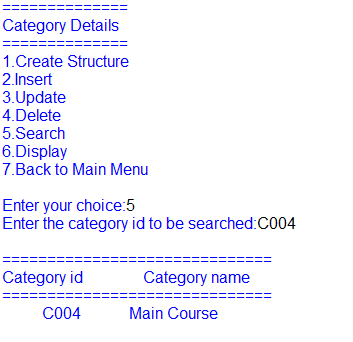
**Update category**



**Delete category**



**Search category**

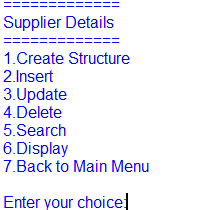


**Display category**

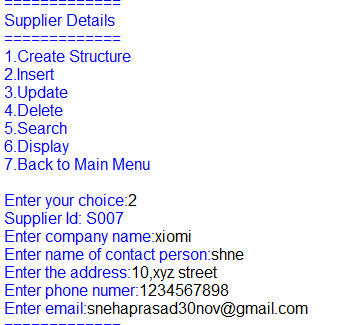


**Module 3: supplier**

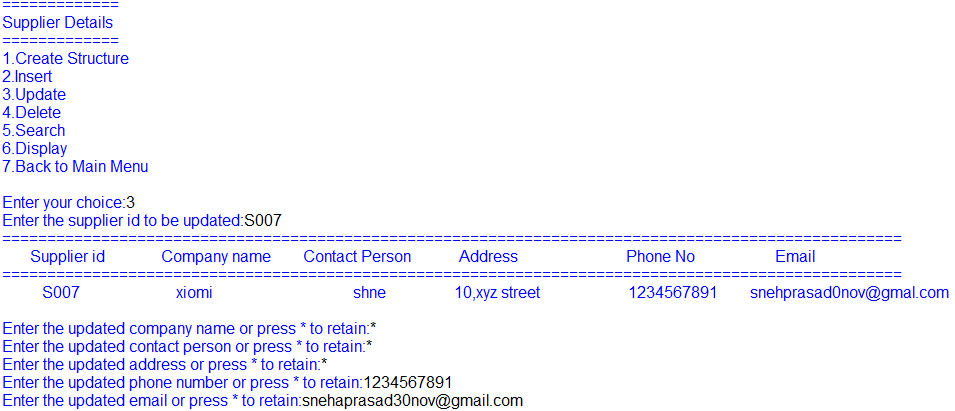
**Create supplier**

****

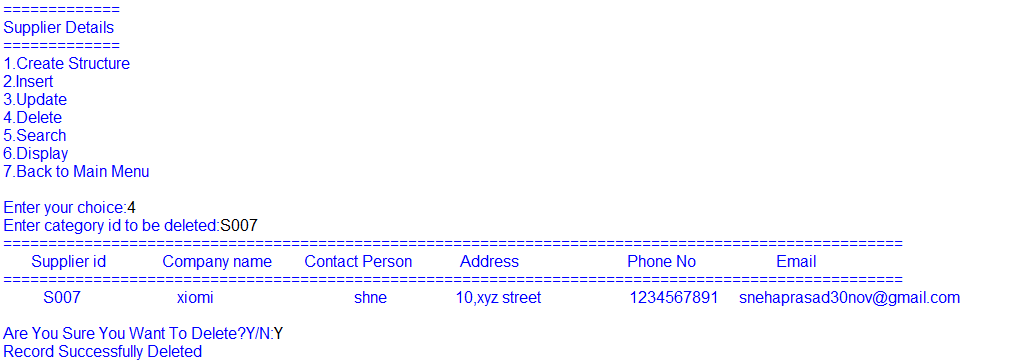
**Insert supplier**



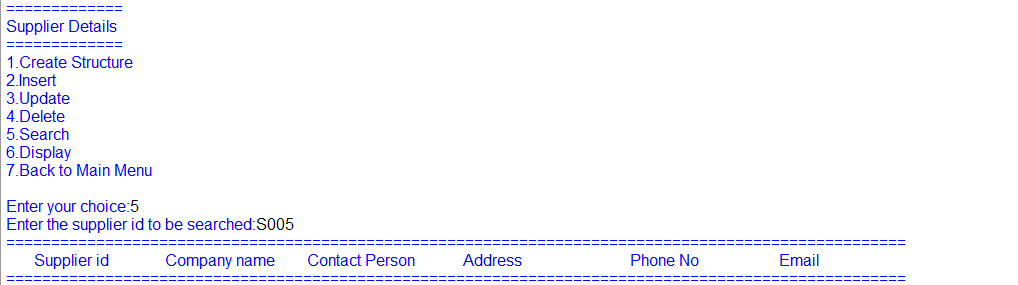
**Update supplier**

****

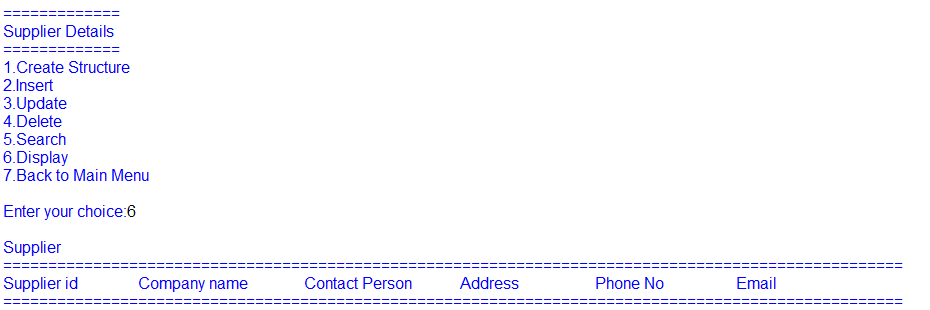
**Delete supplier**

****

**Search supplier**

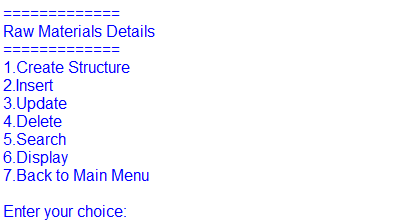
****

**Display supplier**

****

**Module 4: Raw Materials**

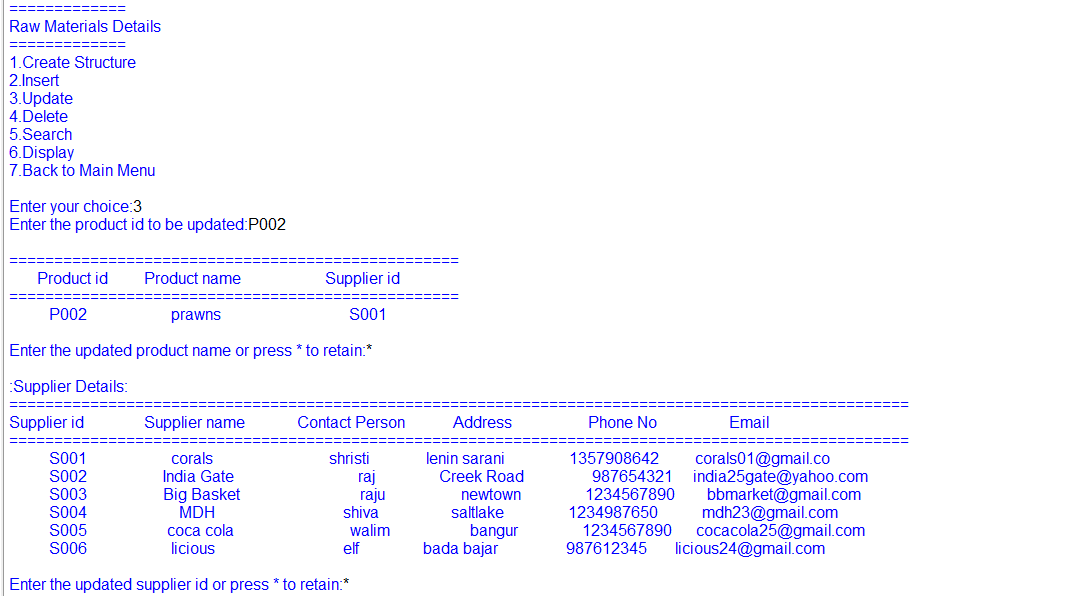
**Create raw materials**

****

**Insert raw materials**

****

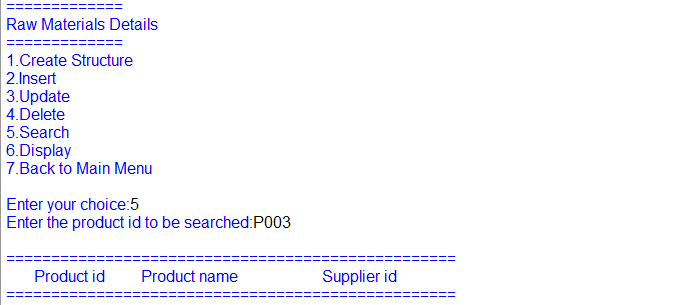
**Update raw materials**

****

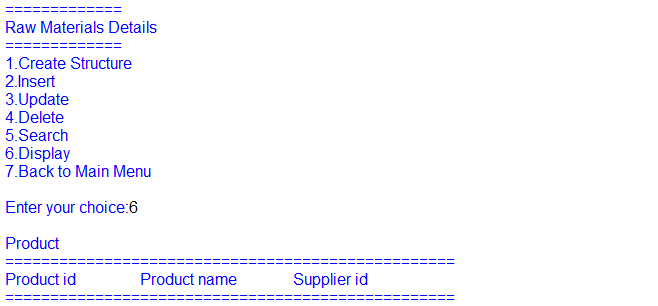
**Delete raw materials**

****

**Search raw materials**

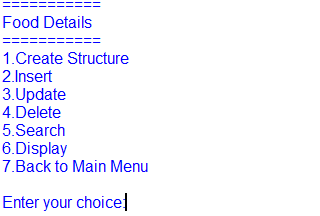
****

**Display raw materials**

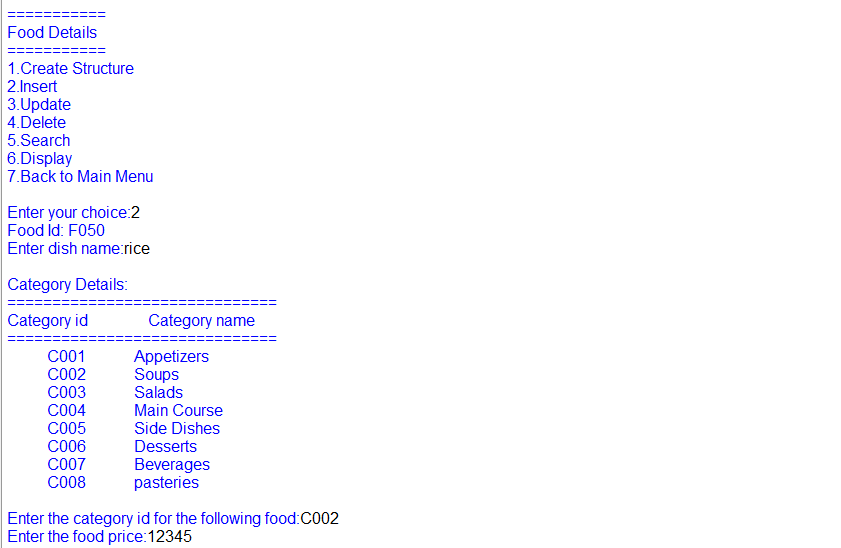
****

**Module 5: food**

**Create food**

****

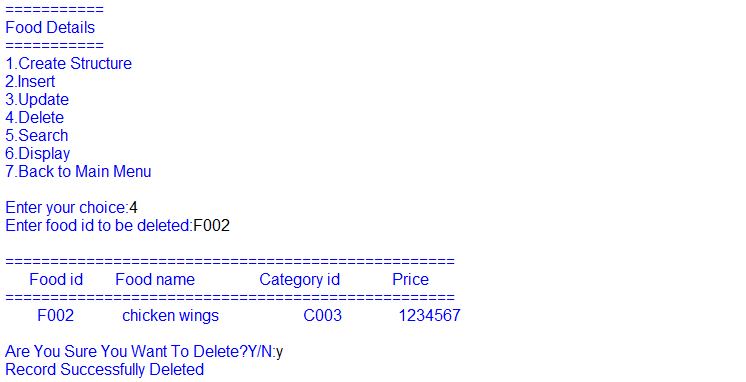
**Insert food**

****

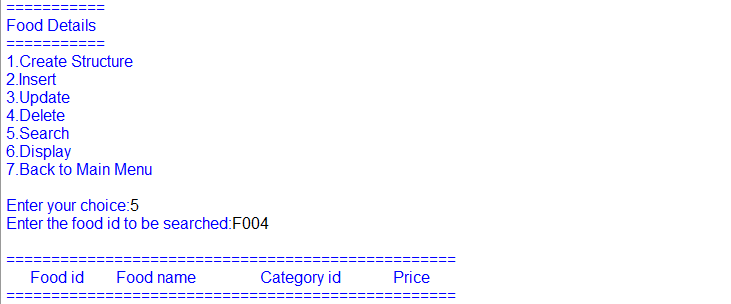
**Update food**

****

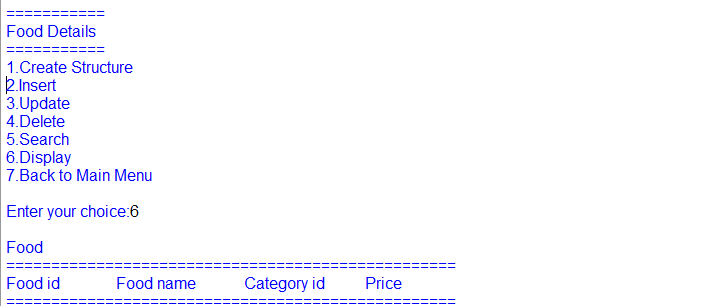
**Delete food**

****

**Search food**

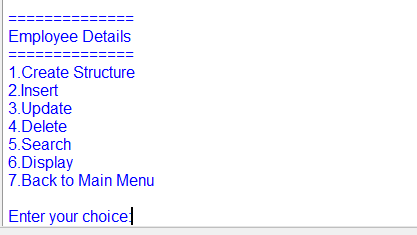
****

**Display food**

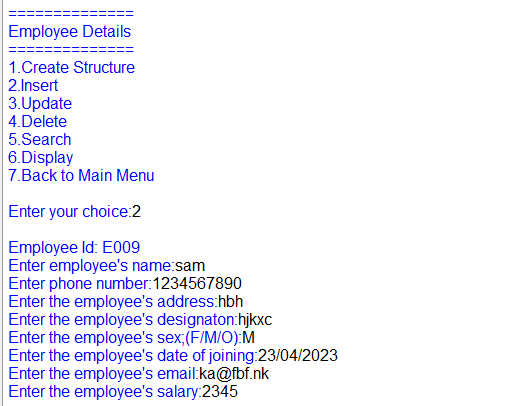
****

**Module 6: employee**

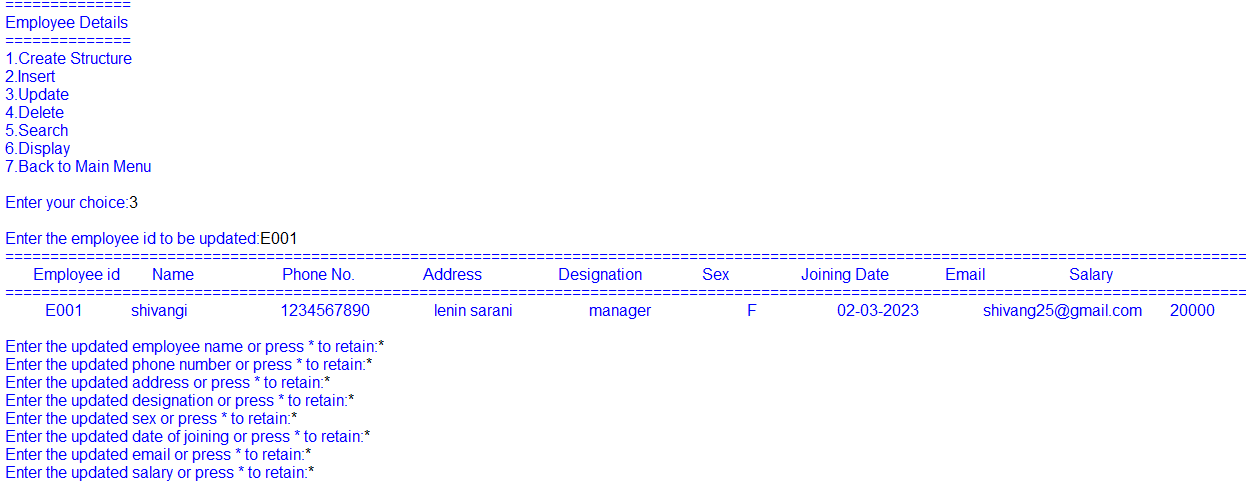
**Create employee**

****

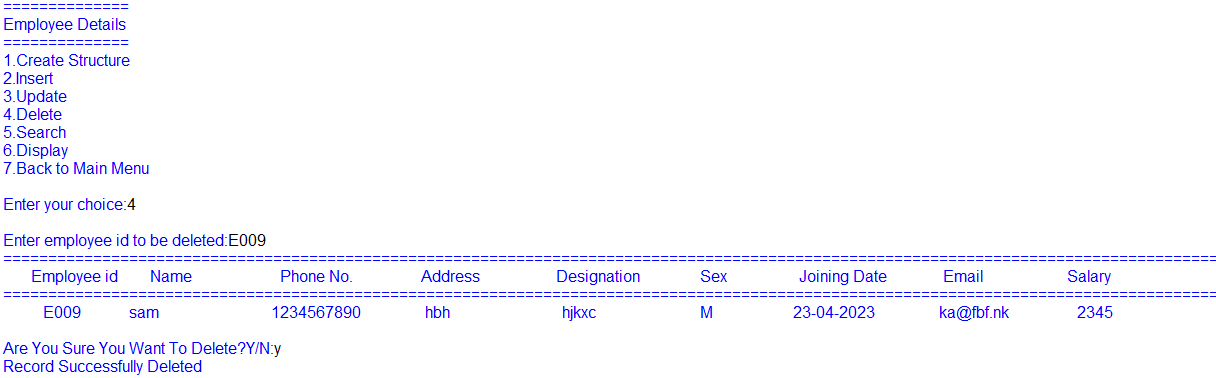
**Insert employee**

****

**Update employee**

****

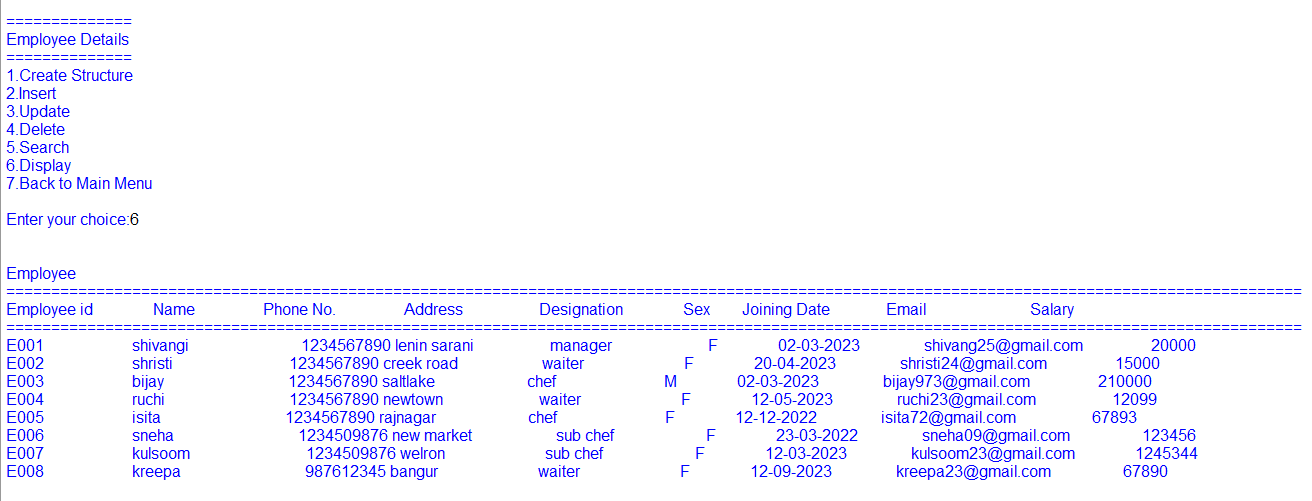
**Delete employee**

****

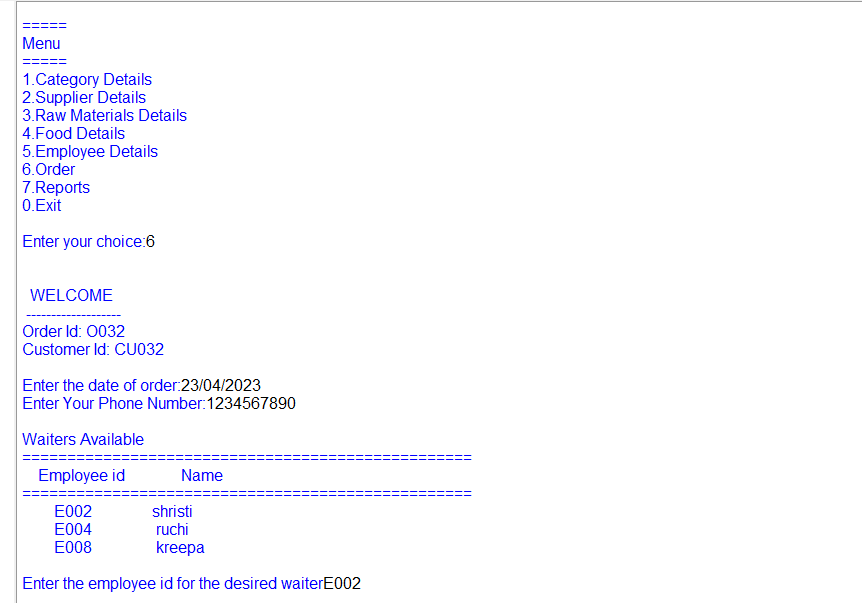
**Search employee**

****

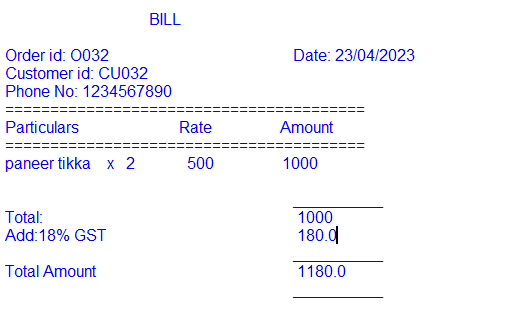
**Display employee**

****

**Module 6: order**

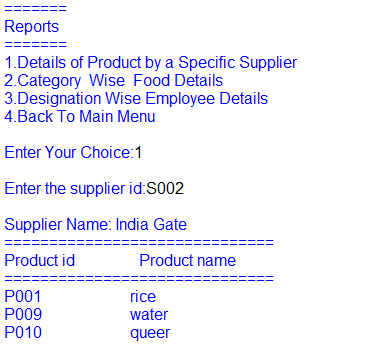
****

****

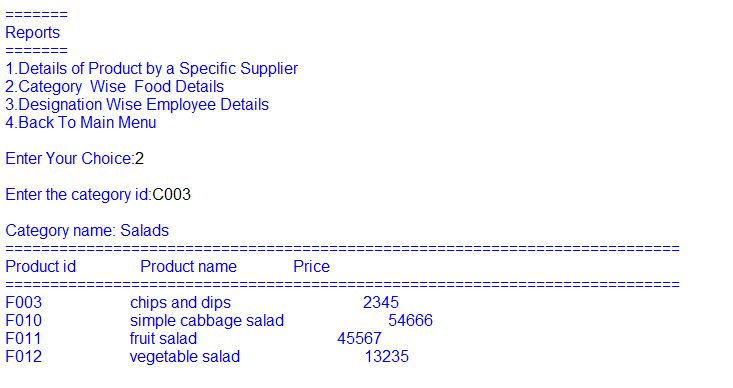
****

**Module 7: reports**

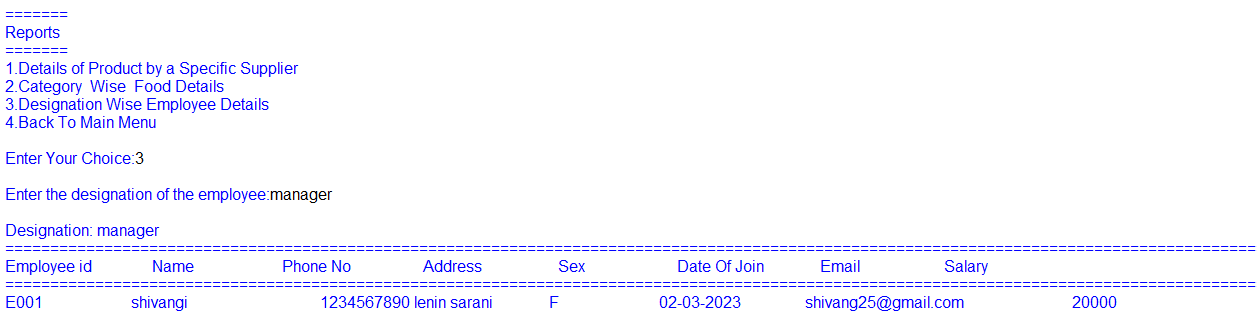
**Details of product by specific supplier**

****

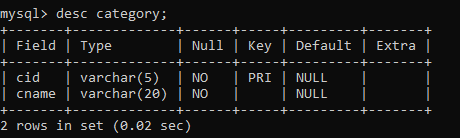
**Category wise food details**

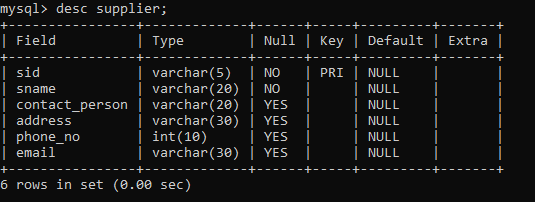
****

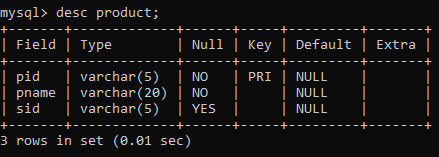
**Designation wise employee details**

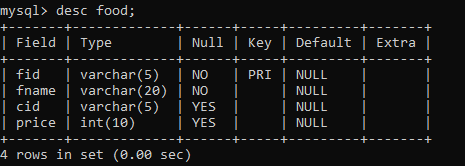
****

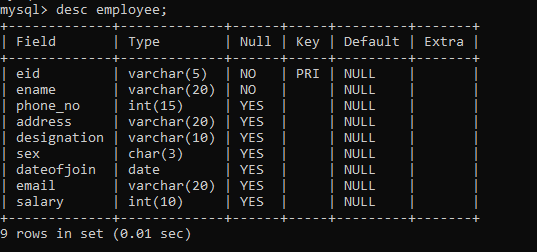
Back end

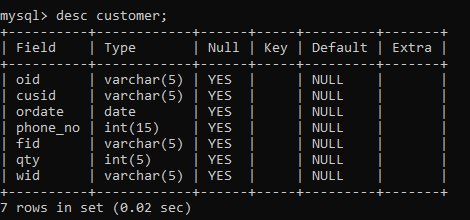












TECHNICAL DESIGN

**MODULE 1: MAIN MENU**

|  |  |
| --- | --- |
| FUCTION SIGNATURE | main() |
| Purpose and Description | To display the main menu and call modules based on the user’s choice. |
| Return value | void |

**MODULE 2: CATEGORY**

|  |  |
| --- | --- |
| FUCTION SIGNATURE | create\_cat() |
| Purpose and Description | To create the structure of the category table |
| Return value | void |

|  |  |
| --- | --- |
| FUCTION SIGNATURE | generate\_catid |
| Purpose and Description | To generate a category id |
| Return value | string |

|  |  |
| --- | --- |
| FUCTION SIGNATURE | cat\_ins() |
| Purpose and Description | To insert details into the category table |
| Return value | void |

|  |  |
| --- | --- |
| FUCTION SIGNATURE | cat\_upd() |
| Purpose and Description | To modify the details in the table for a valid category ID given by the user. |
| Return value | void |

|  |  |
| --- | --- |
| FUCTION SIGNATURE | cat\_del() |
| Purpose and Description | To delete the details from the table for a valid category ID given by the user. |
| Return value | void |

|  |  |
| --- | --- |
| FUCTION SIGNATURE | cat\_srch(n) |
| Purpose and Description | To search for the details of a category |
| Return value | string |

|  |  |
| --- | --- |
| FUCTION SIGNATURE | cat\_dis() |
| Purpose and Description | To display the details from the category table. |
| Return value | void |

**MODULE 3: SUPPLIER**

|  |  |
| --- | --- |
| FUCTION SIGNATURE | create\_sup() |
| Purpose and Description | To create the structure of the supplier table |
| Return value | void |

|  |  |
| --- | --- |
| FUCTION SIGNATURE | generate\_supid() |
| Purpose and Description | To generate a supplier id |
| Return value | string |

|  |  |
| --- | --- |
| FUCTION SIGNATURE | sup\_ins() |
| Purpose and Description | To insert details into the supplier table |
| Return value | void |

|  |  |
| --- | --- |
| FUCTION SIGNATURE | sup\_upd() |
| Purpose and Description | To modify the details of a supplier. |
| Return value | void |

|  |  |
| --- | --- |
| FUCTION SIGNATURE | sup\_del() |
| Purpose and Description | To delete the details of a supplier. |
| Return value | void |

|  |  |
| --- | --- |
| FUCTION SIGNATURE | sup\_srch(n) |
| Purpose and Description | To search the details of a supplier. |
| Return value | string |

|  |  |
| --- | --- |
| FUCTION SIGNATURE | sup\_dis() |
| Purpose and Description | To display the details from the supplier table |
| Return value | void |

**MODULE 4: PRODUCT**

|  |  |
| --- | --- |
| FUCTION SIGNATURE | create\_prod() |
| Purpose and Description | To create the structure of the Product table. |
| Return value | void |

|  |  |
| --- | --- |
| FUCTION SIGNATURE | generate\_proid() |
| Purpose and Description | To generate a product id. |
| Return value | string |

|  |  |
| --- | --- |
| FUCTION SIGNATURE | prod\_ins() |
| Purpose and Description | To insert the product details. |
| Return value | void |

|  |  |
| --- | --- |
| FUCTION SIGNATURE | prod\_upd() |
| Purpose and Description | To update the details of product. |
| Return value | void |

|  |  |
| --- | --- |
| FUCTION SIGNATURE | prod\_del() |
| Purpose and Description | To delete the details of a product. |
| Return value | void |

|  |  |
| --- | --- |
| FUCTION SIGNATURE | prod\_srch(n) |
| Purpose and Description | To search for the product details. |
| Return value | string |

|  |  |
| --- | --- |
| FUCTION SIGNATURE | prod\_dis() |
| Purpose and Description | To display the details of products. |
| Return value | void |

**MODULE 5: Food**

|  |  |
| --- | --- |
| FUCTION SIGNATURE | create\_food() |
| Purpose and Description | To create the structure of food table. |
| Return value | void |

|  |  |
| --- | --- |
| FUCTION SIGNATURE | generate\_foodid() |
| Purpose and Description | To generate a food id. |
| Return value | string |

|  |  |
| --- | --- |
| FUCTION SIGNATURE | food\_ins() |
| Purpose and Description | To insert food details in the table. |
| Return value | void |

|  |  |
| --- | --- |
| FUCTION SIGNATURE | food\_upd() |
| Purpose and Description | To update the details of a food. |
| Return value | void |

|  |  |
| --- | --- |
| FUCTION SIGNATURE | food\_del() |
| Purpose and Description | To delete the details of a food. |
| Return value | void |

|  |  |
| --- | --- |
| FUCTION SIGNATURE | food\_srch(n) |
| Purpose and Description | To search for food details. |
| Return value | string |

|  |  |
| --- | --- |
| FUCTION SIGNATURE | food\_dis() |
| Purpose and Description | To display the details of the food. |
| Return value | void |

**MODULE 6: EMPLOYEE**

|  |  |
| --- | --- |
| FUCTION SIGNATURE | create\_emp() |
| Purpose and Description | To create the structure of the employee table. |
| Return value | void |

|  |  |
| --- | --- |
| FUCTION SIGNATURE | generate\_empid() |
| Purpose and Description | To generate an employee id |
| Return value | string |

|  |  |
| --- | --- |
| FUCTION SIGNATURE | emp\_ins() |
| Purpose and Description | To insert employee details in the table. |
| Return value | void |

|  |  |
| --- | --- |
| FUCTION SIGNATURE | emp\_upd() |
| Purpose and Description | To update the details of an employee. |
| Return value | void |

|  |  |
| --- | --- |
| FUCTION SIGNATURE | emp\_del() |
| Purpose and Description | To delete the details of an employee. |
| Return value | void |

|  |  |
| --- | --- |
| FUCTION SIGNATURE | emp\_srch(n) |
| Purpose and Description | To search the details of an employee. |
| Return value | string |

|  |  |
| --- | --- |
| FUCTION SIGNATURE | emp\_dis() |
| Purpose and Description | To display the employee details. |
| Return value | void |

**Module 7: order**

|  |  |
| --- | --- |
| FUCTION SIGNATURE | create\_cus() |
| Purpose and Description | To create the structure of the table. |
| Return value | void |

|  |  |
| --- | --- |
| FUCTION SIGNATURE | generate\_ordid() |
| Purpose and Description | To generate the order id for the customers |
| Return value | string |

|  |  |
| --- | --- |
| FUCTION SIGNATURE | generate\_cusid() |
| Purpose and Description | To generate customer id. |
| Return value | string |

|  |  |
| --- | --- |
| FUCTION SIGNATURE | dine\_in() |
| Purpose and Description | To accept the details of the customers and food. It assigns a waiter to the customers after accepting the details. |
| Return value | void |

|  |  |
| --- | --- |
| FUCTION SIGNATURE | bill\_display(x) |
| Purpose and Description | To display the bill for the customer. |
| Return value | void |

**MODULE 8: DATE**

|  |  |
| --- | --- |
| FUCTION SIGNATURE | datevalidation(x) |
| Purpose and Description | To check if the date is valid or not. |
| Return value | boolean |

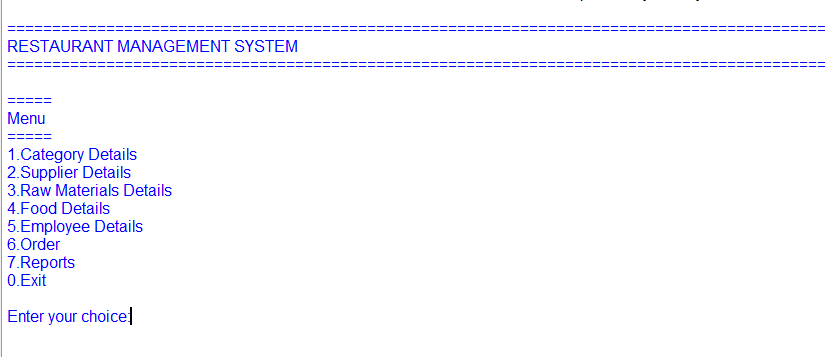
|  |  |
| --- | --- |
| FUCTION SIGNATURE | date\_convert(x) |
| Purpose and Description | To convert date from YYYY/MM/DD to DD/MM/YYYY. |
| Return value | string |

Source code

import mysql.connector as sql  
  
#DATE VALIDATION  
def datevalidation(x):  
    m=[31,28,31,30,31,30,31,31,30,31,30,31]  
    while True:  
        yy=int(x[6:])  
        mm=int(x[3:5])  
        dd=int(x[0:2])  
        if yy%4==0:     #LEAP YEAR  
            m[1]=29  
        if mm>=1 and mm<=12:  
            if dd>=1 and dd <=m[mm-1]:  
                return True  
            else:  
                return False        
        else:  
            return False  
  
#DATE CONVERT  
def date\_convert(x):  
    s=str(x)  
    y=s[0:4]  
    m=s[5:7]  
    d=s[8:]  
    s1=d+'-'+m+'-'+y  
    return s1  
     
#CATEGORY TABLE  
  
#CREATE CATEGORY TABLE  
def create\_cat():  
    try:  
        query='create table category(cid varchar(5) primary key NOT NULL,cname varchar(20) NOT NULL)'  
        cur.execute(query)  
        con.commit()  
        print('Table Successfully Created')  
    except BaseException:  
        print('Table Already Created')  
  
#GENERATE CATEGORY ID  
def generate\_catid():  
    query='select \* from category;'  
    n=''  
    count=0  
    cur.execute(query)  
    data=cur.fetchall()  
    for row in data:  
        count=count+1  
        n=row  
    if count==0:  
        cid='C001'  
    else:  
        s1=n[0]  
        cid=int(s1[1:])  
        if cid<9:  
            nid=cid+1  
            cid='C00'+str(nid)  
        elif cid<99:  
            nid=cid+1  
            cid='C0'+str(nid)  
        elif cid<999:  
            nid=cid+1  
            cid='C'+str(nid)  
    return cid  
  
#CATEGORY TABLE INSERT  
def cat\_ins():  
    cid=generate\_catid()  
    print('Category Id:',cid)  
    cname=input('Enter category name:')  
    query="insert into category values('{}','{}')".format(cid,cname)  
    cur.execute(query)  
    con.commit()  
  
#CATEGORY TABLE SEARCH  
def cat\_srch(n):  
    query="select \* from category where cid='{}'".format(n)  
    cur.execute(query)  
    data=cur.fetchall()  
    ct=cur.rowcount  
    if ct==0:  
        print('Id not found')  
        return ct  
    else:  
        print()  
        print('{:<30s}'.format('='\*30))  
        print('{:<25s} {:<25s}'.format('Category id','Category name'))  
        print('{:<30s}'.format('='\*30))  
        for row in data:  
            print('{:^25s} {:<25s}'.format(row[0],row[1]) )  
        print()  
        return 1  
  
#CATEGORY TABLE UPDATE  
def cat\_upd():  
    cid=input('Enter the category id to be updated:')  
    if cat\_srch(cid)==1:  
        cname=input('Enter the updated category name or press \* to retain:')  
        if cname!='\*':  
            query="update category set cname='{}' where cid='{}'".format(cname,cid)  
            cur.execute(query)  
            con.commit()  
  
#CATEGORY TABLE DELETE  
def cat\_del():  
    cid=input('Enter category id to be deleted:')  
    if cat\_srch(cid)==1:  
        ch=input('Are You Sure You Want To Delete?Y/N:')  
        if ch not in 'Nn':  
            query="delete from category where cid='{}';".format(cid)  
            cur.execute(query)  
            con.commit()  
            print('Record Successfully Deleted')  
  
#CATEGORY TABLE DISPLAY  
def cat\_dis():  
    query="select \* from category"  
    cur.execute(query)  
    data=cur.fetchall()  
    print()  
    print('Category ')  
    print('{:<30s}'.format('='\*30))  
    print('{:<25s} {:<25s}'.format('Category id','Category name'))  
    print('{:<30s}'.format('='\*30))  
    for row in data:  
        print('{:^25s} {:<25s}'.format(row[0],row[1]) )  
    print()  
  
#SUPPLIER TABLE  
  
#GENERATE SUPPLIER ID  
def generate\_supid():  
    query='select \* from supplier;'  
    n=''  
    count=0  
    cur.execute(query)  
    data=cur.fetchall()  
    for row in data:  
        count=count+1  
        n=row  
    if count==0:  
        sid='S001'  
    else:  
        s1=n[0]  
        sid=int(s1[1:])  
        if sid<9:  
            nid=sid+1  
            sid='S00'+str(nid)  
        elif sid<99:  
            nid=sid+1  
            sid='S0'+str(nid)  
        elif sid<999:  
            nid=sid+1  
            sid='S'+str(nid)  
    return sid  
  
#CREATE SUPPLIER TABLE  
def create\_sup():  
    try:  
        query='create table supplier(sid varchar(5) primary key NOT NULL,sname varchar(20) NOT NULL,contact\_person varchar(20),address varchar(30),phone\_no int(10),email varchar(30))'  
        cur.execute(query)  
        con.commit()  
        print('Table Successfully Created')  
    except BaseException:  
        print('Table Already Created')  
  
#SUPPLIER TABLE INSERT  
def sup\_ins():  
    sid=generate\_supid()  
    print('Supplier Id:',sid)  
    sname=input('Enter company name:')  
    contact=input('Enter name of contact person:')  
    add=input('Enter the address:')  
    while True:  
        no=input('Enter phone numer:')  
        if  len(no)==10:  
            while True:  
                email=input('Enter email:')  
                if '@'  in email and '.' in email :  
                    query="insert into supplier values('{}','{}','{}','{}',{},'{}')".format(sid,sname,contact,add,int(no),email)  
                    cur.execute(query)  
                    con.commit()  
                    break  
                else:  
                    print('Enter a valid email')  
            break  
        else:  
            print('Enter a valid 10-digit phone number')  
  
#SUPPLIER TABLE SEARCH  
def sup\_srch(n):  
    query="select \* from supplier where sid='{}'".format(n)  
    cur.execute(query)  
    data=cur.fetchall()  
    ct=cur.rowcount  
    if ct==0:  
        print('Id not found')  
        return ct  
    else:  
        print('{:<100s}'.format('='\*100))  
        print('{:^25s} {:^25s} {:<25s} {:<25s} {:^25s} {:^25s}'.format('Supplier id','Company name','Contact Person','Address','Phone No','Email'))  
        print('{:<100s}'.format('='\*100))  
        for row in data:  
            print('{:^25s} {:^30s} {:>25s} {:>30s} {:>30d} {:>30s}'.format(row[0],row[1],row[2],row[3],row[4],row[5]))  
        print()  
        return 1  
  
#SUPPLIER TABLE UPDATE  
def sup\_upd():  
    sid=input('Enter the supplier id to be updated:')  
    if sup\_srch(sid)==1:  
        sname=input('Enter the updated company name or press \* to retain:')  
        if sname!='\*':  
            query="update supplier set sname='{}' where sid='{}'".format(sname,sid)  
            cur.execute(query)  
            con.commit()  
        contact=input('Enter the updated contact person or press \* to retain:')  
        if contact!='\*':  
            query="update supplier set contact\_person='{}' where sid='{}'".format(contact,sid)  
            cur.execute(query)  
            con.commit()  
        add=input('Enter the updated address or press \* to retain:')  
        if add!='\*':  
            query="update supplier set address='{}' where sid='{}'".format(add,sid)  
            cur.execute(query)  
            con.commit()  
        while True:  
            no=input('Enter the updated phone number or press \* to retain:')  
            if no!='\*':  
                if len(no)==10:  
                    query="update supplier set phone\_no={} where sid='{}'".format(int(no),sid)  
                    cur.execute(query)  
                    con.commit()  
                    break  
            else:  
                print('Enter a valid 10-digit phone number')    
        while True:  
            email=input('Enter the updated email or press \* to retain:')  
            if email!='\*'  and  '@'   in email and '.' in email:  
                query="update supplier set email='{}' where sid='{}'".format(email,sid)  
                cur.execute(query)  
                con.commit()  
                break  
            else:  
                print('Enter a valid email')  
  
#SUPPLIER TABLE DELETE  
def sup\_del():  
    sid=input('Enter category id to be deleted:')  
    if sup\_srch(sid)==1:  
        ch=input('Are You Sure You Want To Delete?Y/N:')  
        if ch not in 'Nn':  
            query="delete from supplier where sid='{}';".format(sid)  
            cur.execute(query)  
            con.commit()  
            print('Record Successfully Deleted')  
  
#SUPPLIER TABLE DISPLAY  
def sup\_dis():  
    query="select \* from supplier"  
    cur.execute(query)  
    data=cur.fetchall()  
    print()  
    print('Supplier')  
    print('{:<100s}'.format('='\*100))  
    print('{:<25s} {:<25s} {:<25s} {:<25s} {:<25s} {:<25s}'.format('Supplier id','Company name','Contact Person','Address','Phone No','Email'))  
    print('{:<100s}'.format('='\*100))  
    for row in data:  
        print('{:<25s} {:<30s} {:<25s} {:<30s} {:>30d} {:<30s}'.format(row[0],row[1],row[2],row[3],row[4],row[5]))  
    print()  
  
#PRODUCT TABLE  
  
#CREATE PRODUCT TABLE  
def create\_prod():  
    try:  
        query='create table product(pid varchar(5) primary key NOT NULL,pname varchar(20) NOT NULL,sid varchar(5))'  
        cur.execute(query)  
        con.commit()  
        print('Table Successfully Created')  
    except BaseException:  
        print('Table Already Created')  
  
#GENERATE PRODUCT ID  
def generate\_prodid():  
    query='select \* from product;'  
    n=''  
    count=0  
    cur.execute(query)  
    data=cur.fetchall()  
    for row in data:  
        count=count+1  
        n=row  
    if count==0:  
        pid='P001'  
    else:  
        p1=n[0]  
        pid=int(p1[1:])  
        if pid<9:  
            nid=pid+1  
            pid='P00'+str(nid)  
        elif pid<99:  
            nid=pid+1  
            pid='P0'+str(nid)  
        elif pid<999:  
            nid=pid+1  
            pid='P'+str(nid)  
    return pid  
  
#PRODUCT TABLE INSERT  
def prod\_ins():  
    pid=generate\_prodid()  
    print('Product Id:',pid)  
    pname=input('Enter product name:')  
    query1="select \* from supplier"  
    cur.execute(query1)  
    data=cur.fetchall()  
    print()  
    print(':Supplier Details:')  
    print('{:<100s}'.format('='\*100))  
    print('{:<25s} {:<25s} {:<25s} {:<25s} {:<25s} {:<25s}'.format('Supplier id','Supplier name','Contact Person','Address','Phone No','Email'))  
    print('{:<100s}'.format('='\*100))  
    for row in data:  
        print('{:^25s} {:^25s} {:>25s} {:>25s} {:>25d} {:>25s}'.format(row[0],row[1],row[2],row[3],row[4],row[5]))  
    print()  
    while True:  
        sid=input('Enter the supplier id for the following product:')  
        query2="select \* from supplier where sid='{}'".format(sid)  
        cur.execute(query2)  
        data=cur.fetchall()  
        if len(data)==0:  
            print('Id not found')  
        else:  
            query="insert into product values('{}','{}','{}')".format(pid,pname,sid)  
            cur.execute(query)  
            con.commit()  
            break  
  
#PRODUCT TABLE SEARCH  
def prod\_srch(n):  
    query="select \* from product where pid='{}'".format(n)  
    cur.execute(query)  
    data=cur.fetchall()  
    ct=cur.rowcount  
    if ct==0:  
        print('Id not found')  
        return ct  
    else:  
        print()  
        print('{:<50s}'.format('='\*50))  
        print('{:^25s} {:<25s} {:^25s} '.format('Product id','Product name','Supplier id'))  
        print('{:<50s}'.format('='\*50))  
        for row in data:  
            print('{:^25s} {:^25s} {:>25s}'.format(row[0],row[1],row[2]))  
        print()  
        return 1  
  
#PRODUCT TABLE UPDATE  
def prod\_upd():  
    pid=input('Enter the product id to be updated:')  
    if prod\_srch(pid)==1:  
        pname=input('Enter the updated product name or press \* to retain:')  
        if pname!='\*':  
            query="update product set pname='{}' where pid='{}'".format(pname,pid)  
            cur.execute(query)  
            con.commit()  
        query1="select \* from supplier"  
        cur.execute(query1)  
        data=cur.fetchall()  
        print()  
        print(':Supplier Details:')  
        print('{:<100s}'.format('='\*100))  
        print('{:<25s} {:<25s} {:<25s} {:<25s} {:<25s} {:<25s}'.format('Supplier id','Supplier name','Contact Person','Address','Phone No','Email'))  
        print('{:<100s}'.format('='\*100))  
        for row in data:  
            print('{:^25s} {:^25s} {:>25s} {:>25s} {:>25d} {:>25s}'.format(row[0],row[1],row[2],row[3],row[4],row[5]))  
        print()  
        while True:  
            sid=input('Enter the updated supplier id or press \* to retain:')  
            if sid!='\*':  
                query2="select \* from supplier where sid='{}'".format(sid)  
                cur.execute(query2)  
                data=cur.fetchall()  
                if len(data)==0:  
                    print('Id not found')  
                else:  
                    query="update product set sid='{}' where pid='{}'".format(sid,pid)  
                    cur.execute(query)  
                    con.commit()  
                    break  
  
#PRODUCT TABLE DELETE  
def prod\_del():  
    pid=input('Enter product id to be deleted:')  
    if prod\_srch(pid)==1:  
        ch=input('Are You Sure You Want To Delete?Y/N:')  
        if ch not in 'Nn':  
            query="delete from product where pid='{}';".format(pid)  
            cur.execute(query)  
            con.commit()  
            print('Record Successfully Deleted')  
  
#PRODUCT TABLE DISPLAY              
def prod\_dis():  
    query="select \* from product"  
    cur.execute(query)  
    data=cur.fetchall()  
    print()  
    print('Product')  
    print('{:<50s}'.format('='\*50))  
    print('{:<25s} {:<25s} {:<25s} '.format('Product id','Product name','Supplier id'))  
    print('{:<50s}'.format('='\*50))  
    for row in data:  
        print('{:<25s} {:<25s} {:<25s}'.format(row[0],row[1],row[2]))  
    print()  
  
#FOOD TABLE  
  
#CREATE FOOD TABLE  
def create\_food():  
    try:  
        query='create table food(fid varchar(5) primary key NOT NULL,fname varchar(20) NOT NULL,cid varchar(5),price int(10))'  
        cur.execute(query)  
        con.commit()  
        print('Table Successfully Created')  
    except BaseException:  
        print('Table Already Created')  
         
#GENERATE FOOD ID  
def generate\_foodid():  
    query='select \* from food;'  
    n=''  
    count=0  
    cur.execute(query)  
    data=cur.fetchall()  
    for row in data:  
        count=count+1  
        n=row  
    if count==0:  
        fid='F001'  
    else:  
        f1=n[0]  
        fid=int(f1[1:])  
        if fid<9:  
            nid=fid+1  
            fid='F00'+str(nid)  
        elif fid<99:  
            nid=fid+1  
            fid='F0'+str(nid)  
        elif fid<999:  
            nid=fid+1  
            fid='F'+str(nid)  
    return fid  
  
#FOOD TABLE INSERT  
def food\_ins():  
    fid=generate\_foodid()  
    print('Food Id:',fid)  
    fname=input('Enter dish name:')  
    query1="select \* from category"  
    cur.execute(query1)  
    data=cur.fetchall()  
    print()  
    print('Category Details:')  
    print('{:<30s}'.format('='\*30))  
    print('{:<25s} {:<25s}'.format('Category id','Category name'))  
    print('{:<30s}'.format('='\*30))  
    for row in data:  
        print('{:^25s} {:<25s}'.format(row[0],row[1]) )  
    print()  
    while True:  
        cid=input('Enter the category id for the following food:')  
        query2="select \* from category where cid='{}'".format(cid)  
        cur.execute(query2)  
        data=cur.fetchall()  
        if len(data)==0:  
            print('Id not found')  
        else:  
            price=int(input('Enter the food price:'))  
            query="insert into food values('{}','{}','{}',{})".format(fid,fname,cid,price)  
            cur.execute(query)  
            con.commit()  
            break  
  
#FOOD TABLE SEARCH  
def food\_srch(n):  
    query="select \* from food where fid='{}'".format(n)  
    cur.execute(query)  
    data=cur.fetchall()  
    ct=cur.rowcount  
    if ct==0:  
        print('Id not found')  
        return ct  
    else:  
        print()  
        print('{:<50s}'.format('='\*50))  
        print('{:^20s} {:<20s} {:^20s} {:^20s}'.format('Food id','Food name','Category id','Price'))  
        print('{:<50s}'.format('='\*50))  
        for row in data:  
            print('{:^20s} {:^20s} {:>20s} {:>20d}'.format(row[0],row[1],row[2],row[3]))  
        print()  
        return 1  
  
#FOOD TABLE UPDATE  
def food\_upd():  
    fid=input('Enter the food id to be updated:')  
    if food\_srch(fid)==1:  
        fname=input('Enter the updated food name or press \* to retain:')  
        if fname!='\*':  
            query="update food set fname='{}' where fid='{}'".format(fname,fid)  
            cur.execute(query)  
            con.commit()  
    query1="select \* from category"  
    cur.execute(query1)  
    data1=cur.fetchall()  
    print()  
    print('Category Details:')  
    print('{:<30s}'.format('='\*30))  
    print('{:<25s} {:<25s}'.format('Category id','Category name'))  
    print('{:<30s}'.format('='\*30))  
    for row in data1:  
        print('{:^25s} {:<25s}'.format(row[0],row[1]) )  
    print()  
    while True:  
        cid=input('Enter the updated category id or press \* to retain:')  
        if cid!='\*':  
            query2="select \* from category where cid='{}'".format(cid)  
            cur.execute(query2)  
            data2=cur.fetchall()  
            if data2==[]:  
                print('Id not found')  
            else:  
                query="update food set cid='{}' where fid='{}'".format(cid,fid)  
                cur.execute(query)  
                con.commit()  
                break  
    price=input('Enter the updated food price or press \* to retain:')  
    if price!='\*':  
        query="update food set price='{}' where fid='{}'".format(int(price),fid)  
        cur.execute(query)  
        con.commit()  
  
#FOOD TABLE DELETE  
def food\_del():  
    fid=input('Enter food id to be deleted:')  
    if food\_srch(fid)==1:  
        ch=input('Are You Sure You Want To Delete?Y/N:')  
        if ch not in 'Nn':  
            query="delete from food where fid='{}';".format(fid)  
            cur.execute(query)  
            con.commit()  
            print('Record Successfully Deleted')  
  
#FOOD TABLE DISPLAY  
def food\_dis():  
    query="select \* from food"  
    cur.execute(query)  
    data=cur.fetchall()  
    print()  
    print('Food')  
    print('{:<50s}'.format('='\*50))  
    print('{:<20s} {:<20s} {:<20s} {:<20s}'.format('Food id','Food name','Category id','Price'))  
    print('{:<50s}'.format('='\*50))  
    for row in data:  
        print('{:<20s} {:<20s} {:<20s} {:>20d}'.format(row[0],row[1],row[2],row[3]))  
    print()  
  
#EMPLOYEE TABLE  
  
#CREATE EMPLOYEE TABLE  
def create\_emp():  
    try:  
        query='create table employee(eid varchar(5) primary key NOT NULL,ename varchar(20) NOT NULL,phone\_no int(15),address varchar(20),designation varchar(10),sex char(3),dateofjoin date,email varchar(20),salary int(10))'  
        cur.execute(query)  
        con.commit()  
        print('Table Successfully Created')  
    except BaseException:  
        print('Table Already Created')  
  
#EMPLOYEE ID GENERATION  
def generate\_empid():  
    query='select \* from employee'  
    n=''  
    count=0  
    cur.execute(query)  
    data=cur.fetchall()  
    for row in data:  
        count=count+1  
        n=row  
    if count==0:  
        eid='E001'  
    else:  
        e1=n[0]  
        eid=int(e1[1:])  
        if eid<9:  
            nid=eid+1  
            eid='E00'+str(nid)  
        elif eid<99:  
            nid=eid+1  
            eid='E0'+str(nid)  
        elif eid<999:  
            nid=eid+1  
            eid='E'+str(nid)  
    return eid  
  
#EMPLOYEE TABLE INSERT  
def emp\_ins():  
    eid=generate\_empid()  
    print('Employee Id:',eid)  
    ename=input("Enter employee's name:")  
    while True:  
        phno=input('Enter phone number:')  
        if  len(phno)==10:  
            no=int(phno)  
            break  
        else:  
            print("Enter a valid 10 digited number")  
    add=input("Enter the employee's address:")  
    desig=input("Enter the employee's designaton:")  
    while True:  
        gender=input("Enter the employee's sex;(F/M/O):")  
        if gender in 'FfMmOo':  
            break  
        else:  
            print("Invalid Gender")  
    while True:  
        join=input("Enter the employee's date of joining:")  
        if datevalidation(join)==True:  
            yy=int(join[6:])  
            mm=int(join[3:5])  
            dd=int(join[0:2])  
            jdate=str(yy)+'-'+str(mm)+'-'+str(dd)  
            break  
        else:  
            print("Invalid Date. Please reenter")  
    while True:  
        email=input("Enter the employee's email:")  
        if '@' in email and '.' in email:  
            break  
        else:  
            print('Enter a valid email')  
    sal=int(input("Enter the employee's salary:"))  
    query="insert into employee values('{}','{}',{},'{}','{}','{}','{}','{}',{})".format(eid,ename,no,add,desig,gender,jdate,email,sal)  
    cur.execute(query)  
    con.commit()  
             
#EMPLOYEE TABLE SEARCH  
def emp\_srch(n):  
    query="select \* from employee where eid='{}'".format(n)  
    cur.execute(query)  
    data=cur.fetchall()  
    ct=cur.rowcount  
    if ct==0:  
        print('Id not found')  
        return ct  
    else:  
        print('{:<144s}'.format('='\*144))  
        print('{:^25s} {:<25s} {:<25s} {:<25s} {:<25s} {:<20s} {:<25s} {:<25s} {:<25s}'.format('Employee id','Name','Phone No.','Address','Designation','Sex','Joining Date','Email','Salary'))  
        print('{:<144s}'.format('='\*144))  
        for row in data:  
            print('{:^25s} {:<30s} {:<25d} {:<30s} {:<30s} {:<20s} {:<25s} {:<25s} {:<25d}'.format(row[0],row[1],row[2],row[3],row[4],row[5],date\_convert(row[6]),row[7],row[8]))  
        print()  
        return 1  
  
#EMPLOYEE TABLE UPDATE  
def emp\_upd():  
    eid=input('Enter the employee id to be updated:')  
    if emp\_srch(eid)==1:  
        ename=input('Enter the updated employee name or press \* to retain:')  
        if ename!='\*':  
            query="update employee set ename='{}' where eid='{}'".format(ename,eid)  
            cur.execute(query)  
            con.commit()  
        while True:  
            no=input('Enter the updated phone number or press \* to retain:')  
            if no!='\*' :  
                if len(no)==10:                  
                    query="update employee set phone\_no={} where eid='{}'".format(int(no),eid)  
                    cur.execute(query)  
                    con.commit()  
                    break  
                else:  
                    print('Enter a valid 10-digit phone number')  
            else:  
                break  
        add=input('Enter the updated address or press \* to retain:')  
        if add!='\*':  
            query="update employee set address='{}' where eid='{}'".format(add,eid)  
            cur.execute(query)  
            con.commit()  
        desig=input('Enter the updated designation or press \* to retain:')  
        if desig!='\*':  
            query="update employee set designation='{}' where eid='{}'".format(desig,eid)  
            cur.execute(query)  
            con.commit()  
        sex=input('Enter the updated sex or press \* to retain:')  
        if sex!='\*':  
            if sex  not in 'FfMmOo':  
                print('Enter a valid sex')  
            else:  
                query="update employee set sex='{}' where eid='{}'".format(sex,eid)  
                cur.execute(query)  
                con.commit()  
        while True:  
            dateofjoin=input('Enter the updated date of joining or press \* to retain:')  
            if dateofjoin!='\*':  
                if datevalidation(dateofjoin)==True:  
                    yy=int(dateofjoin[6:])  
                    mm=int(dateofjoin[3:5])  
                    dd=int(dateofjoin[0:2])  
                    jdate=str(yy)+'-'+str(mm)+'-'+str(dd)  
                    query="update employee set dateofjoin='{}' where eid='{}'".format(jdate,eid)  
                    cur.execute(query)  
                    con.commit()  
                    break  
                else:  
                    print('Enter a valid date')  
            else:  
                break  
        while True:  
            email=input('Enter the updated email or press \* to retain:')  
            if email!='\*':  
                if '@' not in email and '.' not in email:  
                    print('Enter a valid email')  
                else:  
                    query="update employee set email='{}' where eid='{}'".format(email,eid)  
                    cur.execute(query)  
                    con.commit()  
                    break  
            else:  
                break  
        sal=input('Enter the updated salary or press \* to retain:')  
        if sal!='\*':  
            query="update employee set salary='{}' where eid='{}'".format(sal,eid)  
            cur.execute(query)  
            con.commit()  
  
#EMPLOYEE TABLE DELETE  
def emp\_del():  
    eid=input('Enter employee id to be deleted:')  
    if emp\_srch(eid)==1:  
        ch=input('Are You Sure You Want To Delete?Y/N:')  
        if ch not in 'Nn':  
            query="delete from employee where eid='{}';".format(eid)  
            cur.execute(query)  
            con.commit()  
            print('Record Successfully Deleted')  
  
#EMPLOYEE TABLE DISPLAY  
def emp\_dis():  
    query="select \* from employee"  
    cur.execute(query)  
    data=cur.fetchall()  
    print()  
    print('Employee')  
    print('{:<144s}'.format('='\*144))  
    print('{:<25s} {:<20s} {:<25s} {:<25s} {:<25s} {:<10s} {:<25s} {:<30s} {:<20s}'.format('Employee id','Name','Phone No.','Address','Designation','Sex','Joining Date','Email','Salary'))  
    print('{:<144s}'.format('='\*144))  
    for row in data:  
        print('{:<25s} {:<20s} {:>25d} {:<30s} {:<30s} {:<15s} {:<25s} {:<30s} {:>10d}'.format(row[0],row[1],row[2],row[3],row[4],row[5],date\_convert(row[6]),row[7],row[8]))  
    print()  
  
#CUSTOMER      
  
#CREATE CUSTOMER TABLE  
def create\_cus():  
    try:  
        query='create table customer(oid varchar(5), cusid varchar(5),ordate date,phone\_no int(15),fid varchar(5),qty int(5),wid varchar(5))'  
        cur.execute(query)  
        con.commit()  
    except BaseException:  
        print()  
  
#ORDER IF GENERATION  
def generate\_ordid():  
    query='select \* from customer'  
    n=''  
    count=0  
    cur.execute(query)  
    data=cur.fetchall()  
    for row in data:  
        count=count+1  
        n=row  
    if count==0:  
        oid='O001'  
    else:  
        ord1=n[0]  
        oid=int(ord1[1:])  
        if oid<9:  
            nid=oid+1  
            oid='O00'+str(nid)  
        elif oid<99:  
            nid=oid+1  
            oid='O0'+str(nid)  
        elif oid<999:  
            nid=oid+1  
            oid='O'+str(nid)  
    return oid  
  
#CUSTOMER ID GENERATION  
def generate\_cusid():  
    query='select \* from customer'  
    n=''  
    count=0  
    cur.execute(query)  
    data=cur.fetchall()  
    for row in data:  
        count=count+1  
        n=row  
    if count==0:  
        cusid='CU001'  
    else:  
        cus1=n[0]  
        cusid=int(cus1[1:])  
        if cusid<9:  
            nid=cusid+1  
            cusid='CU00'+str(nid)  
        elif cusid<99:  
            nid=cusid+1  
            cusid='CU0'+str(nid)  
        elif cusid<999:  
            nid=cusid+1  
            cusid='CU'+str(nid)  
    return cusid  
   
#BILL DISPLAY  
def bill\_display(oid,cusid,phno,date):  
    print()  
    print('\t\tBILL')  
    print()  
    print('Order id:',oid,"\t\t\tDate:",date)  
    print('Customer id:',cusid)  
    print('Phone No:',phno)  
    query2="select fname,price,qty from food ,customer  where food.fid=customer.fid and oid='{}' ".format(oid)  
    cur.execute(query2)  
    data2=cur.fetchall()  
    print('{:<40s}'.format('='\*40))  
    print('{:<35s} {:<20s} {:<20s}'.format('Particulars','Rate','Amount'))  
    print('{:<40s}'.format('='\*40))  
    tot=0  
    if data2==[]:  
        print("No data available")  
    else:  
        for row in data2:  
            qty=row[2]  
            price=row[1]  
            tprice=qty\*price  
            tot=tot+tprice  
            print('{:<15s} {:<2s} {:>2d} {:>15d} {:>20d}'.format(row[0],'x',qty,price,tprice))  
    print()  
    print('\t\t\t\t\_\_\_\_\_\_\_\_\_\_')  
    print("Total:","\t\t\t\t",tot)  
    gst=(18/100)\* tot  
    print('Add:18% GST','\t\t\t',gst)  
    total=tot+gst  
    print('\t\t\t\t\_\_\_\_\_\_\_\_\_\_')  
    print('Total Amount','\t\t\t',total)  
    print('\t\t\t\t\_\_\_\_\_\_\_\_\_\_')  
  
#ORDER  
def dine\_in():  
    create\_cus()  
    cusid=generate\_cusid()  
    oid=generate\_ordid()  
    print('  WELCOME ')  
    print(' -------------------')  
    print('Order Id:',oid)  
    print('Customer Id:',cusid)  
    print()  
    while True:  
        date=input('Enter the date of order:')  
        if datevalidation(date)==True:  
            yy=int(date[6:])  
            mm=int(date[3:5])  
            dd=int(date[0:2])  
            cdate=str(yy)+'-'+str(mm)+'-'+str(dd)  
            break  
        else:  
            print("Invalid Date. Please renter")  
             
    while True:  
        phno=int(input('Enter Your Phone Number:'))  
        if len(str(phno))!=10:  
               print('Enter a valid 10-digit no')  
        else:  
            break  
             
    while True:          #waiter check  
                query4="select eid,ename from employee where designation='waiter'"  
                cur.execute(query4)  
                data=cur.fetchall()  
                print()  
                print('Waiters Available')  
                print('{:<50s}'.format('='\*50))  
                print('{:^20s} {:^20s}'.format('Employee id','Name'))  
                print('{:<50s}'.format('='\*50))  
                for row in data:  
                    print('{:^20s} {:^20s}'.format(row[0],row[1]))  
                print()  
                watid=input('Enter the employee id for the desired waiter')  
                query5="select \* from employee where eid='{}' and designation='waiter' ".format(watid)  
                cur.execute(query5)  
                data5=cur.fetchall()  
                if data5==[]:  
                          print('Invalid Id')  
                else:  
                    break  
             
    while True:          
        query1="select \* from category"  
        cur.execute(query1)  
        data1=cur.fetchall()  
        #ct1=cur.rowcount  
        print()  
        print("Category's Available")  
        print('{:<30s}'.format('='\*30))  
        print('{:<25s} {:<25s}'.format('Category id','Category name'))  
        print('{:<30s}'.format('='\*30))  
        for row in data1:  
            print('{:^25s} {:<25s}'.format(row[0],row[1]) )  
        print()  
        if data1==[]:  
            print('Category Currently Not Available')  
            break  
        else:  
            while True:         #category check  
                cat=input('Enter the category id : ')  
                query2="select \* from food where cid='{}'".format(cat)  
                cur.execute(query2)  
                data2=cur.fetchall()  
                #ct2=cur.rowcount  
                if data2==[]:  
                    print('Food Currently Not Available For This Category')  
                else:  
                    break  
            while True:     #Food check  
                print()  
                print('Food Menu')  
                print('{:<50s}'.format('='\*50))  
                print('{:^20s} {:<20s} {:^20s} {:^20s}'.format('Food id','Food name','Category id','Price'))  
                print('{:<50s}'.format('='\*50))  
                for row in data2:  
                    print('{:^20s} {:^20s} {:>20s} {:>20d}'.format(row[0],row[1],row[2],row[3]))  
                print()  
                food=input('Enter the food id for which food you want to order:')  
                query3="select \* from food where fid='{}' and cid='{}'".format(food,cat)  
                cur.execute(query3)  
                data3=cur.fetchall()  
                if data3==[]:  
                   print('Invalid Food Id')  
                else:  
                    qty=int(input('Enter the no.of plates:'))  
                    query6="insert into customer values('{}','{}','{}',{},'{}',{},'{}')".format(oid,cusid,cdate,phno,food,qty,watid)  
                    cur.execute(query6)  
                    con.commit()  
                    ch=input('Do You Wish To Order More?(Y/N):')  
                    if ch in'Nn':  
                        break  
        ch1=input('Do You Wish To Order from another category?(Y/N): ')  
        if ch1 in 'Nn':                      
                 break  
    bill\_display(oid,cusid,phno,date)  
  
#REPORTS PRODUCT  
def report\_prod():  
    sid=input('Enter the supplier id:')  
    query1="select pid,pname from product  where sid='{}' ".format(sid)  
    cur.execute(query1)  
    data1=cur.fetchall()  
    if data1==[]:  
        print('No Product For the Supplier ',sid)  
    else:  
        query2="select sname from supplier where sid='{}' ".format(sid)  
        cur.execute(query2)  
        data2=cur.fetchall()  
        if data2==[]:  
            print('Invalid Id')  
        else:  
            name=data2[0][0]  
            print()  
            print('Supplier Name:',name)  
            print('{:<30s}'.format('='\*30))  
            print('{:<25s} {:<25s}'.format('Product id','Product name'))  
            print('{:<30s}'.format('='\*30))  
            for row in data1:  
                print('{:<25s} {:<25s}'.format(row[0],row[1]))  
            print()  
  
#REPORTS FOOD  
def report\_food():  
    cid=input('Enter the category id:')  
    query1="select fid,fname,price from food where cid='{}'".format(cid)  
    cur.execute(query1)  
    data1=cur.fetchall()  
    if data1==[]:  
        print('No Food for the Category ')  
    else:  
        query2="select cname from category where cid='{}' ".format(cid)  
        cur.execute(query2)  
        data2=cur.fetchall()  
        if data2==[]:  
            print('Invalid Id')  
        else:  
            name=data2[0][0]  
        print()  
        print('Category name:',name)  
        print('{:<75s}'.format('='\*75))  
        print('{:<25s} {:<25s} {:<25s}'.format('Product id','Product name','Price'))  
        print('{:<75s}'.format('='\*75))  
        for row in data1:  
            print('{:<25s} {:<25s} {:>25d}'.format(row[0],row[1],row[2]))  
        print()  
  
#REPORTS EMPLOYEE  
def report\_emp():  
    desig=input('Enter the designation of the employee:')  
    query="select eid,ename,phone\_no,address,sex,dateofjoin,email,salary from employee where designation='{}' ".format(desig)  
    cur.execute(query)  
    data=cur.fetchall()  
    if data==[]:  
        print('No Employee Available for the Designation ')  
    else:  
        print()  
        print('Designation:',desig)  
        print('{:<144s}'.format('='\*144))  
        print('{:<25s} {:<25s} {:<25s} {:<25s} {:<25s} {:<25s} {:<25s} {:<25s}'.format('Employee id','Name','Phone No','Address','Sex','Date Of Join','Email','Salary'))  
        print('{:<144s}'.format('='\*144))  
        for row in data:  
            print('{:<25s} {:<25s} {:>25d} {:<25s} {:<25s} {:<25s} {:<25s} {:>25d}'.format(row[0],row[1],row[2],row[3],row[4],date\_convert(row[5]),row[6],row[7]))  
        print()  
           
#MAIN MENU  
con=sql.connect(host='localhost',user='root',passwd='',database='project')  
if con.is\_connected()==False:  
    print('Not Connected')  
cur=con.cursor()  
while True:  
    print()  
    print('=' \* 144)  
    print('RESTAURANT MANAGEMENT SYSTEM')  
    print('=' \* 144)  
    print()  
    print('=====')  
    print('Menu')  
    print('=====')  
    print("1.Category Details")  
    print("2.Supplier Details")  
    print("3.Raw Materials Details")  
    print("4.Food Details")  
    print("5.Employee Details")  
    print("6.Order")  
    print("7.Reports")  
    print("0.Exit")  
    print()  
    ch=int(input('Enter your choice:'))  
    print()  
     
    if ch==1:  
        while True:  
            print('==============')  
            print('Category Details')  
            print('==============')  
            print('1.Create Structure')  
            print('2.Insert')  
            print('3.Update')  
            print('4.Delete')  
            print('5.Search')  
            print('6.Display')  
            print('7.Back to Main Menu')  
            print()  
            ch=int(input('Enter your choice:'))  
            if ch==1:  
                create\_cat()  
            elif ch==2:  
                cat\_ins()  
            elif ch==3:  
                cat\_upd()  
            elif ch==4:  
                cat\_del()  
            elif ch==5:  
                cid=input('Enter the category id to be searched:')  
                cat\_srch(cid)  
            elif ch==6:  
                cat\_dis()  
            elif ch==7:  
                print('Back to Main Menu')  
                print('')  
                break  
            else:  
                print('Enter a valid choice')  
                break  
             
    elif ch==2:  
        while True:  
            print('=============')  
            print('Supplier Details')  
            print('=============')  
            print('1.Create Structure')  
            print('2.Insert')  
            print('3.Update')  
            print('4.Delete')  
            print('5.Search')  
            print('6.Display')  
            print('7.Back to Main Menu')  
            print()  
            ch=int(input('Enter your choice:'))  
            if ch==1:  
                create\_sup()  
            elif ch==2:  
                sup\_ins()  
            elif ch==3:  
                sup\_upd()  
            elif ch==4:  
                sup\_del()  
            elif ch==5:  
                sid=input('Enter the supplier id to be searched:')  
                sup\_srch(sid)  
            elif ch==6:  
                sup\_dis()  
            elif ch==7:  
                print('Back to Main Menu')  
                print('')  
                break  
            else:  
                print('Enter a valid choice')  
                break  
  
    elif ch==3:  
        while True:  
            print('=============')  
            print('Raw Materials Details')  
            print('=============')  
            print('1.Create Structure')  
            print('2.Insert')  
            print('3.Update')  
            print('4.Delete')  
            print('5.Search')  
            print('6.Display')  
            print('7.Back to Main Menu')  
            print()  
            ch=int(input('Enter your choice:'))  
            if ch==1:  
                create\_prod()  
            elif ch==2:  
                prod\_ins()  
            elif ch==3:  
                prod\_upd()  
            elif ch==4:  
                prod\_del()  
            elif ch==5:  
                pid=input('Enter the product id to be searched:')  
                prod\_srch(pid)  
            elif ch==6:  
                prod\_dis()  
            elif ch==7:  
                print('Back to Main Menu')  
                print('')  
                break  
            else:  
                print('Enter a valid choice')  
                break  
     
    elif ch==4:  
        while True:  
            print('===========')  
            print('Food Details')  
            print('===========')  
            print('1.Create Structure')  
            print('2.Insert')  
            print('3.Update')  
            print('4.Delete')  
            print('5.Search')  
            print('6.Display')  
            print('7.Back to Main Menu')  
            print()  
            ch=int(input('Enter your choice:'))  
            if ch==1:  
                create\_food()  
            elif ch==2:  
                food\_ins()  
            elif ch==3:  
                food\_upd()  
            elif ch==4:  
                food\_del()  
            elif ch==5:  
                fid=input('Enter the food id to be searched:')  
                food\_srch(fid)  
            elif ch==6:  
                food\_dis()  
            elif ch==7:  
                print('Back to Main Menu')  
                print('')  
                break  
            else:  
                print('Enter a valid choice')  
                break  
  
    elif ch==5:  
        while True:  
            print('==============')  
            print('Employee Details')  
            print('==============')  
            print('1.Create Structure')  
            print('2.Insert')  
            print('3.Update')  
            print('4.Delete')  
            print('5.Search')  
            print('6.Display')  
            print('7.Back to Main Menu')  
            print()  
            ch=int(input('Enter your choice:'))  
            print()  
            if ch==1:  
                create\_emp()  
            elif ch==2:  
                emp\_ins()  
            elif ch==3:  
                emp\_upd()  
            elif ch==4:  
                emp\_del()  
            elif ch==5:  
                eid=input('Enter the employee id to be searched:')  
                emp\_srch(eid)  
            elif ch==6:  
                emp\_dis()  
            elif ch==7:  
                print('Back to Main Menu')  
                print()  
                break  
            else:  
                print('Enter a valid choice')  
                break  
  
    elif ch==6:  
            dine\_in()  
  
    elif ch==7:  
        while True:  
            print('=======')  
            print('Reports')  
            print('=======')  
            print('1.Details of Product by a Specific Supplier')  
            print('2.Category  Wise  Food Details')  
            print('3.Designation Wise Employee Details')  
            print('4.Back To Main Menu ')  
            print()  
            ch=int(input('Enter Your Choice:'))  
            print()  
            if ch==1:  
                report\_prod()  
            elif ch==2:  
                report\_food()  
            elif ch==3:  
                report\_emp()  
            elif ch==4:  
                print('Back to Main Menu')  
                print()  
                break  
            else:  
                print('Enter a valid choice')  
                break  
                 
    elif ch==0:  
        print('Thank You')  
        con.close()  
        break  
     
    else:  
        print('Enter a valid choice')

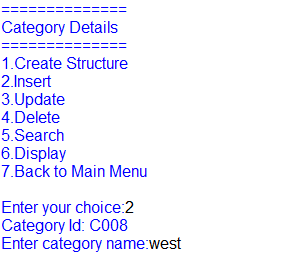
Output

**MODULE 1: MAIN MENU**

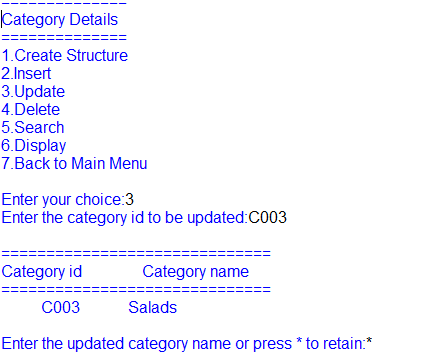


**MODULE 2: CATEGORY**

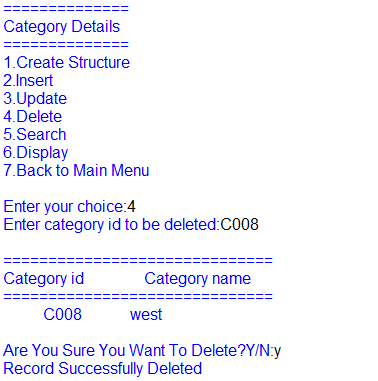
**Insert category**



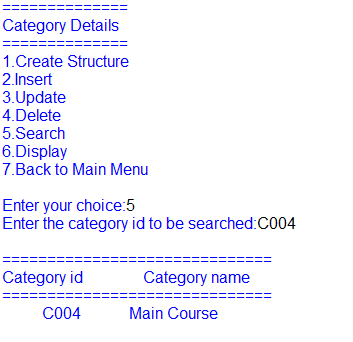
**Update category**



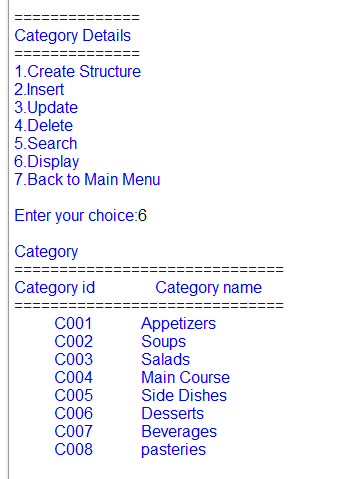
**Delete category**



**Search category**

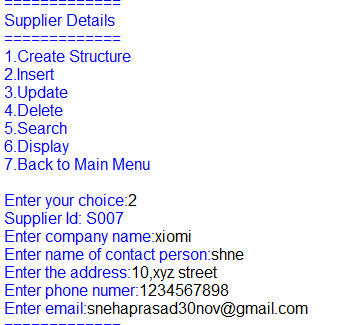


**Display category**



**Module 3: supplier**

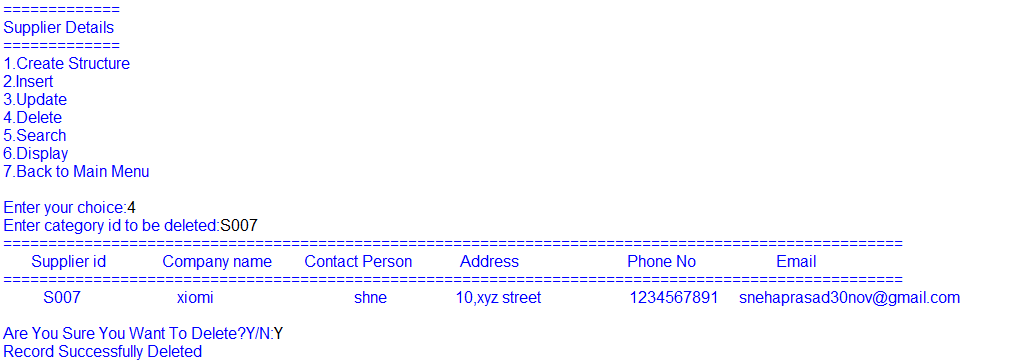
**Insert supplier**



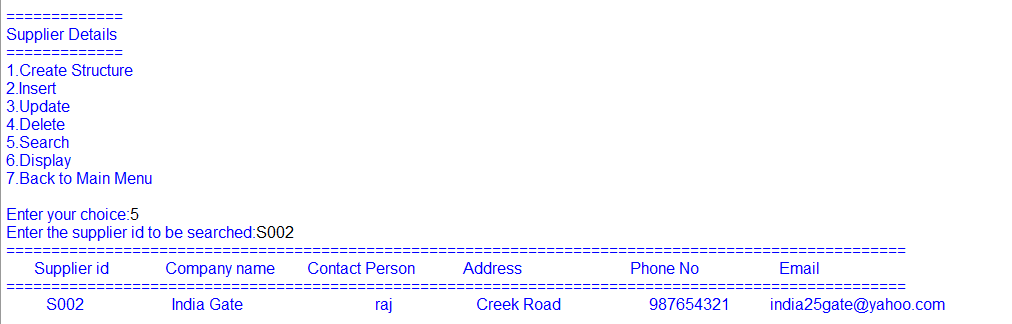
**Update supplier**

****

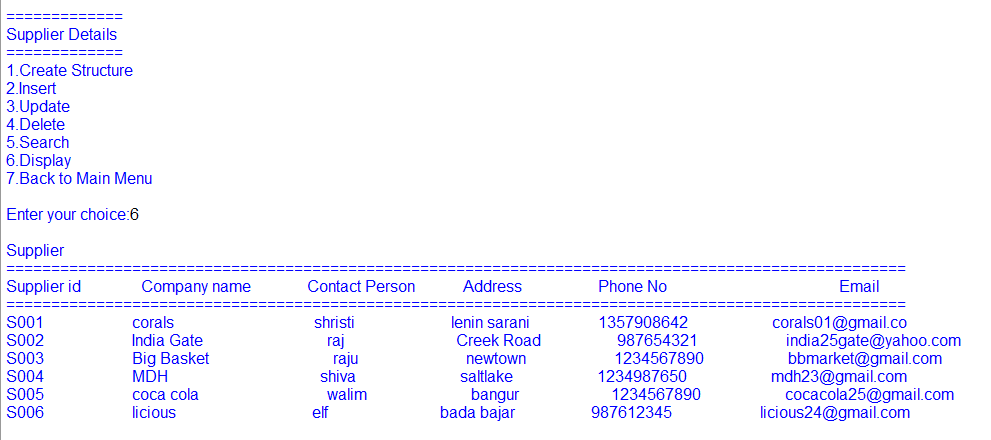
**Delete supplier**

****

**Search supplier**

****

**Display supplier**

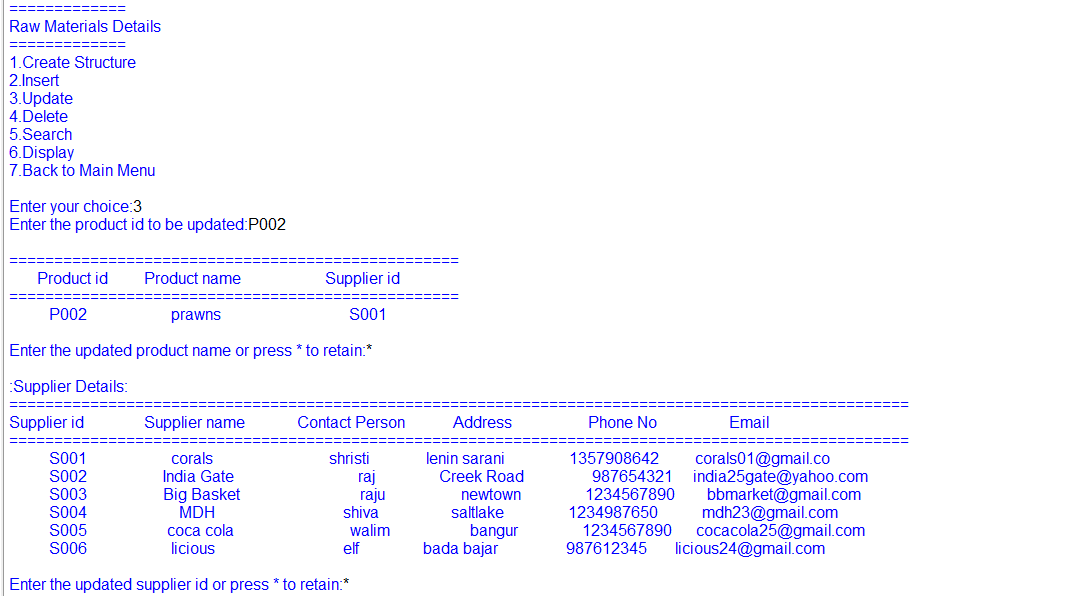
****

**Module 4: Raw Materials**

**Insert raw materials**

****

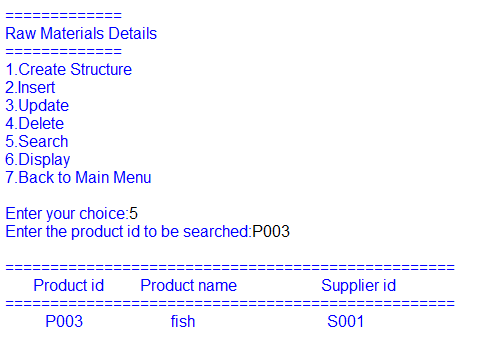
**Update raw materials**

****

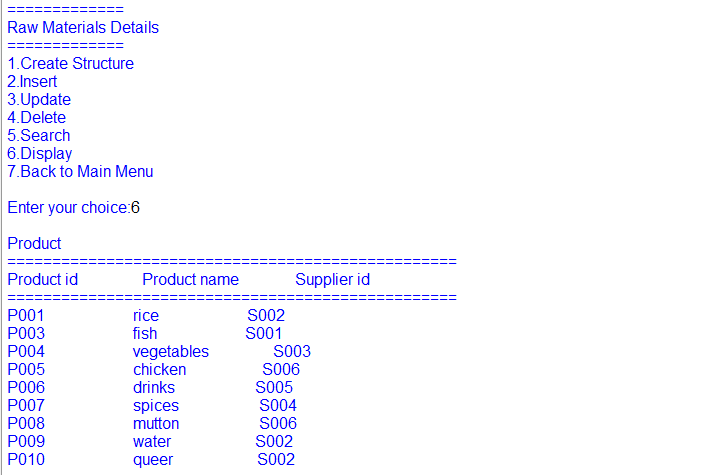
**Delete raw materials**

****

**Search raw materials**

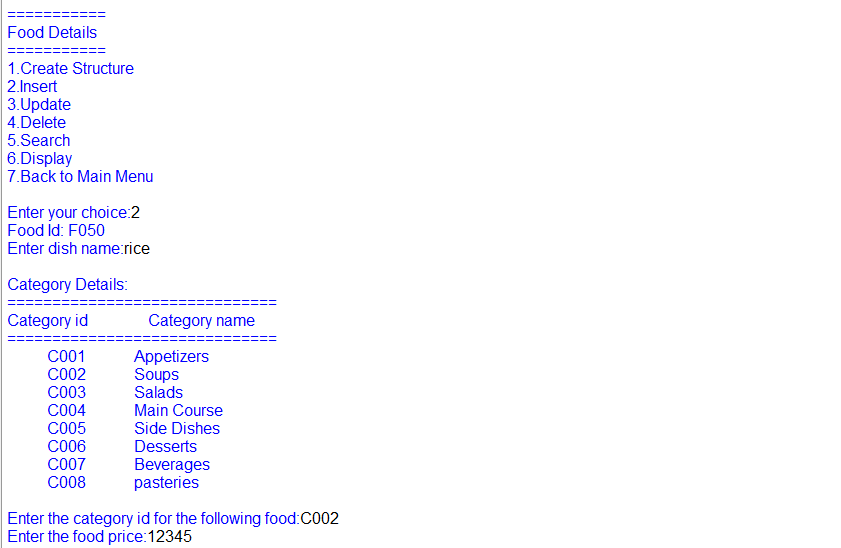
****

**Display raw materials**

****

**Module 5: food**

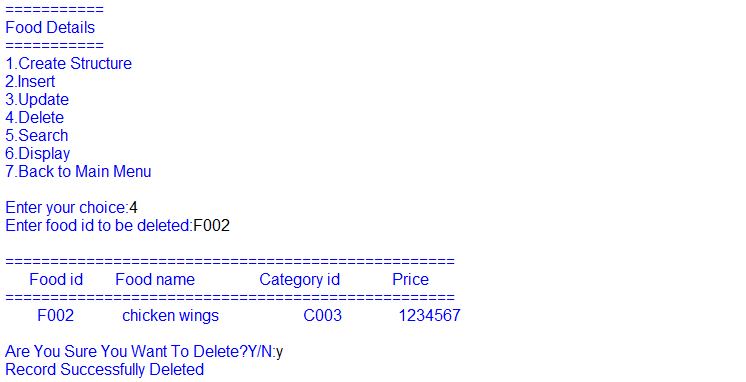
**Insert food**

****

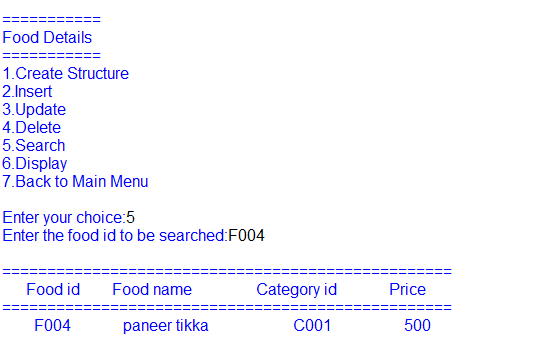
**Update food**

****

**Delete food**

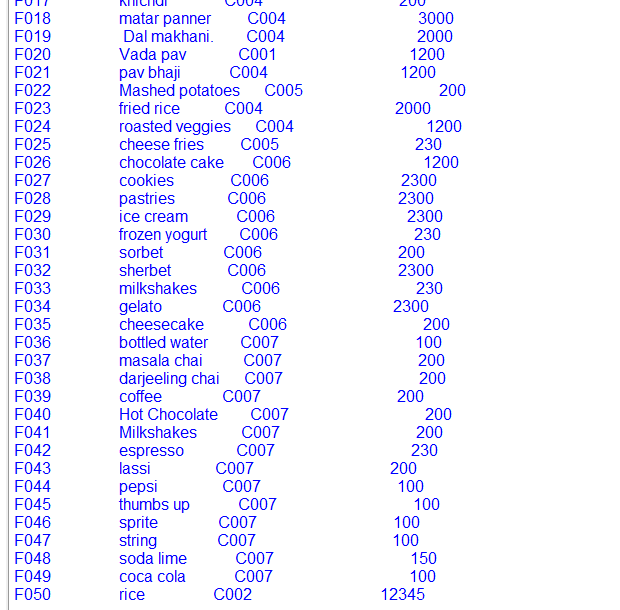
****

**Search food**

****

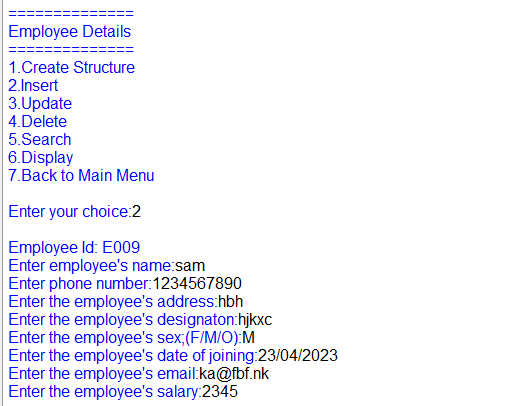
**Display food**

****

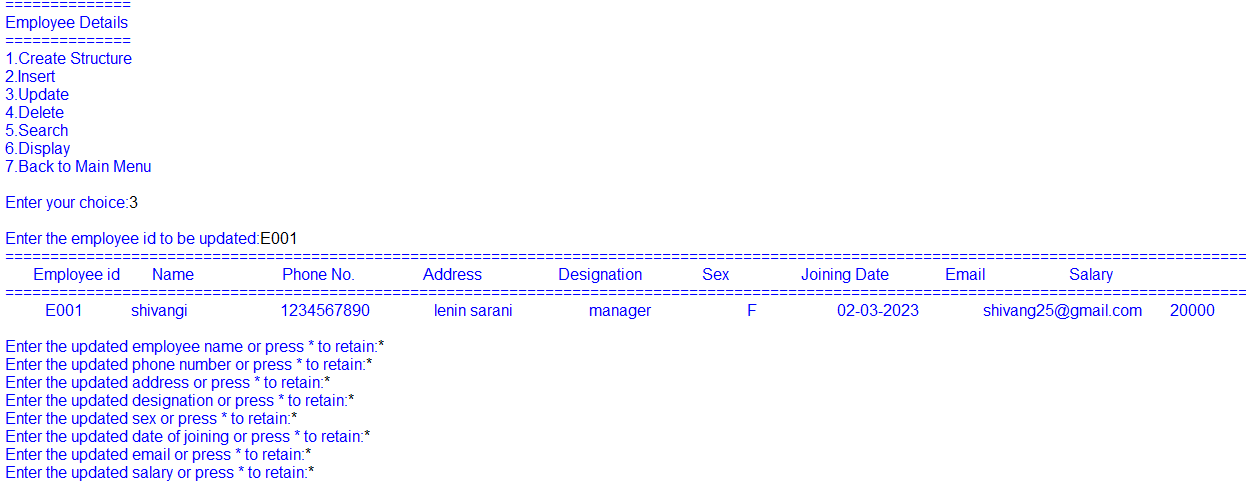
****

**Module 6: employee**

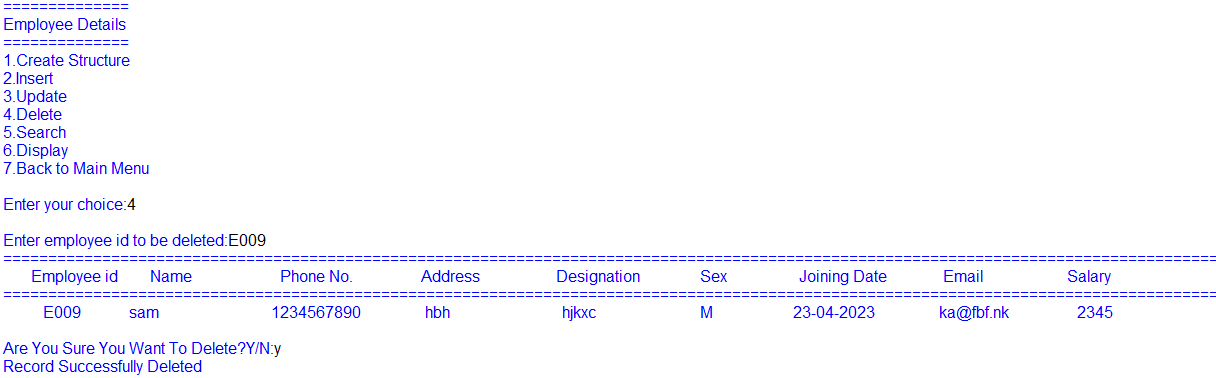
**Insert employee**

****

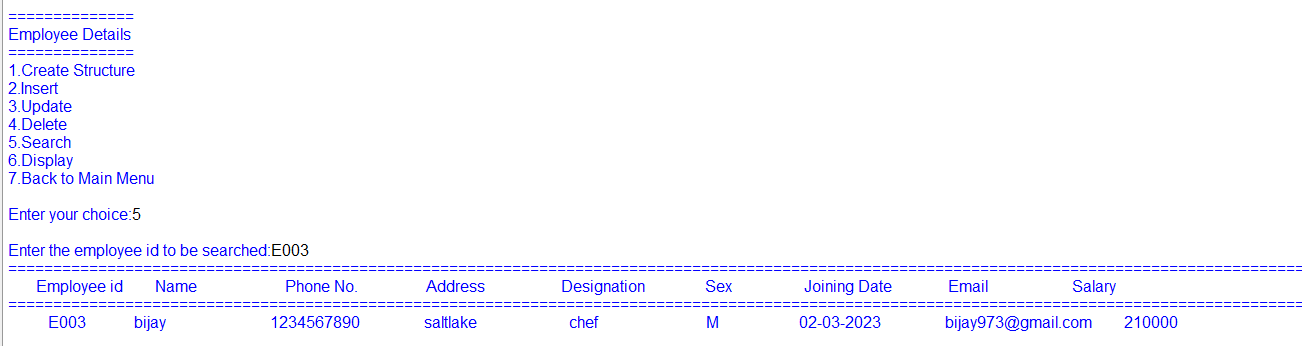
**Update employee**

****

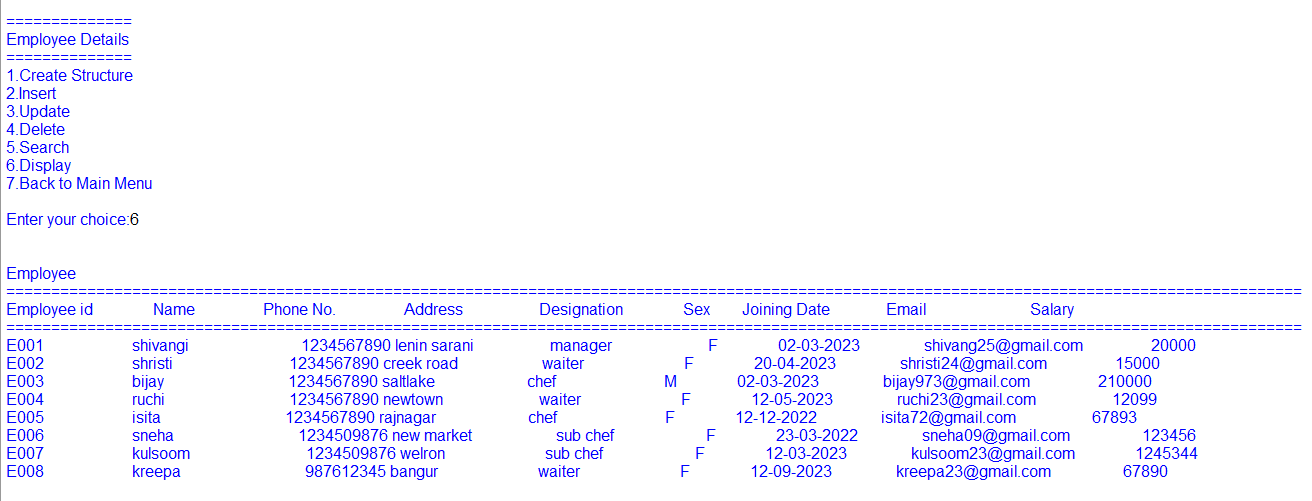
**Delete employee**

****

**Search employee**

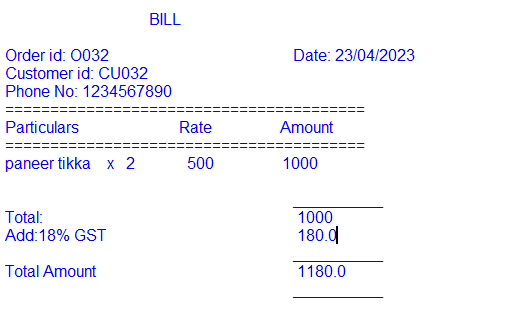
****

**Display employee**

****

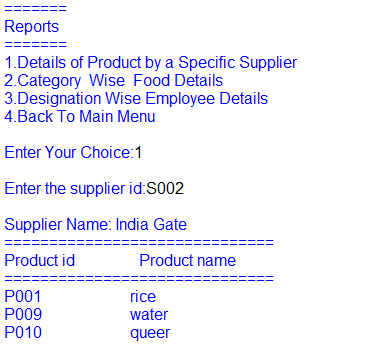
**Module 6: order**

****

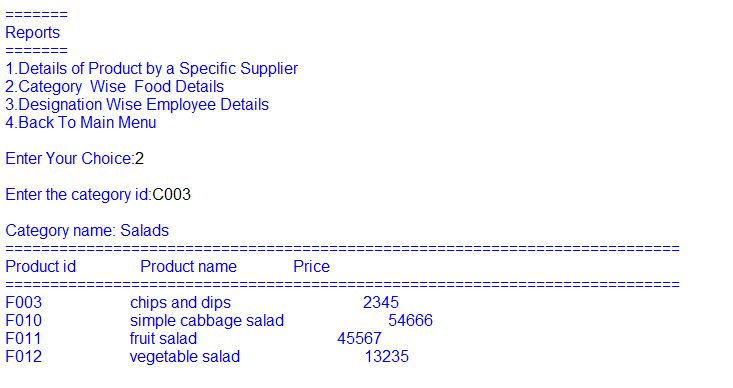
****

**Module 7: reports**

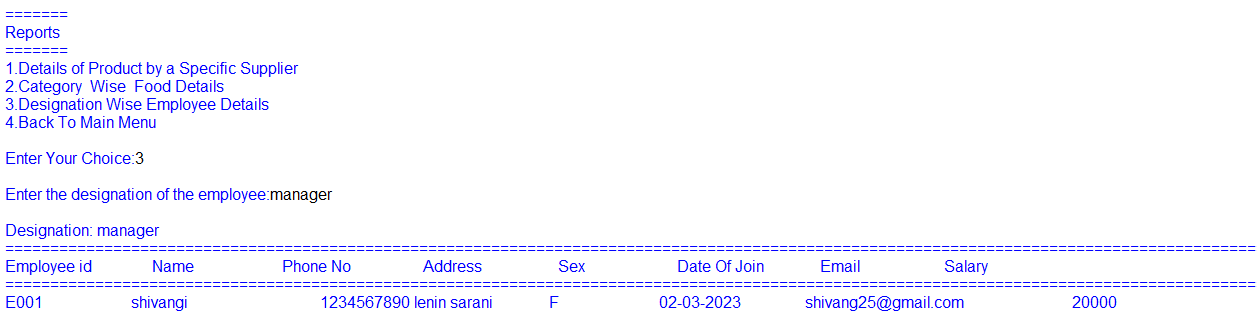
**Details of product by specific supplier**

****

**Category wise food details**

****

**Designation wise employee details**

****

BIBLIOGRAPHY

The project has been completed successfully with the help of the following sources:

* Computer Science with Python for class XI by Sumita Arora.

* Computer Science with Python for class XII by Sumita Arora.