


A Project Report On

"FOOD BOOKING SYSTEM"

SUBMITTED BY:

NAME

CLASS :12

BOARD ROLL NO.:

UNDER THE GUIDANCE OF:

Mr.Mani

ACKNOWLEDGEMENT

This is to certify that **HARSHIT**
Of **class 12** has prepared the report on the Project entitled **"Food Booking System"**.The report is the result of his efforts and endeavors.The report is found worthy of acceptance as final project report for the subject **Computer Science** of **class XII**.He has prepared the project under the guidance of the subject teacher, **Mr.Mani**.

(Mr.Mani)

CERTIFICATE

The project report entitled

"FOOD BOOKING SYSTEM"

Submitted by **HARSHIT** of **Class XII** for the **C.B.S.E. Senior Secondary Examination class XII** of computer science under the guidance of the subject teacher, **Mr. Mani** is found to be correct under all means.

SIGNATURE

DECLARATION

I hereby declare that the project work entitled "**Food Booking System**", submitted to Department of **computer science,** _____ PUBLIC SCHOOL is prepared by me. All the coding are result of my personal efforts.

HARSHIT
12-D

INDEX

FOOD BOOKING SYSTEM

BRIEF OVERVIEW OF PROJECT..

The main objective of the python project on Food ordering is to manage the details of booking, menu payments, and order.
The project is totally built at administrative end and only administrator is guaranteed the access.
The purpose of the project is to build an application program to reduce the manual work for managing the booking, discounts ,and payments.
It tracks all the details about menu, discount , and payments; it also prints various reports as per input given by the user.

INPUT DATA AND VALIDATION OF PROJECT

1. All the fields such as order payments discounts are validated and does not take invalid values.
2. Each form of sales, discounts, bookings cannot accept the blank values.
3. Avoiding errors in data.
4. Controlling amount of input.

SOFTWARE AND HARDWARE REQUIREMENTS:

Data file handling :

has been effectively used in the program. The database is a collection of interrelated data to serve multiple applications. That is database programs create files of information. So we see that files are worked with most, inside the program.

DBMS:

The software required for the management of data is called as DBMS. It has3 models:

- Relation model
- Hierarchical model
- Network model

RELATIONAL MODEL:

It's based on the concept on relation. Relation is the table that consists of rows and columns. The rows of the table are called tuple and the

columns of the table are called attribute. Numbers of rows in the table is called as cardinality. Number of columns in the table is called as degree.

HIERARCHICAL MODEL:

In this type of model, we have multiple records for each record. A particular record has one parent record. No chide record can exist without parent record. In this, the records are organized in tree.

NETWORK MODEL:

In this, the data is represented by collection of records and relationship is represented by (ink or association.

CHARACTERISTICS OF DBMS:

- It reduces the redundancy
- Reduction of data in inconsistency
- Data sharing
- Data standardization

DIFFERENT TYPES OF FILES: -BASED ON ACCESS:

- Sequential file
- Serial file
- Random (direct access) file BASED ON STORAGE:-
- Text file
- Binary File

NEED OF COMPUTERISATION

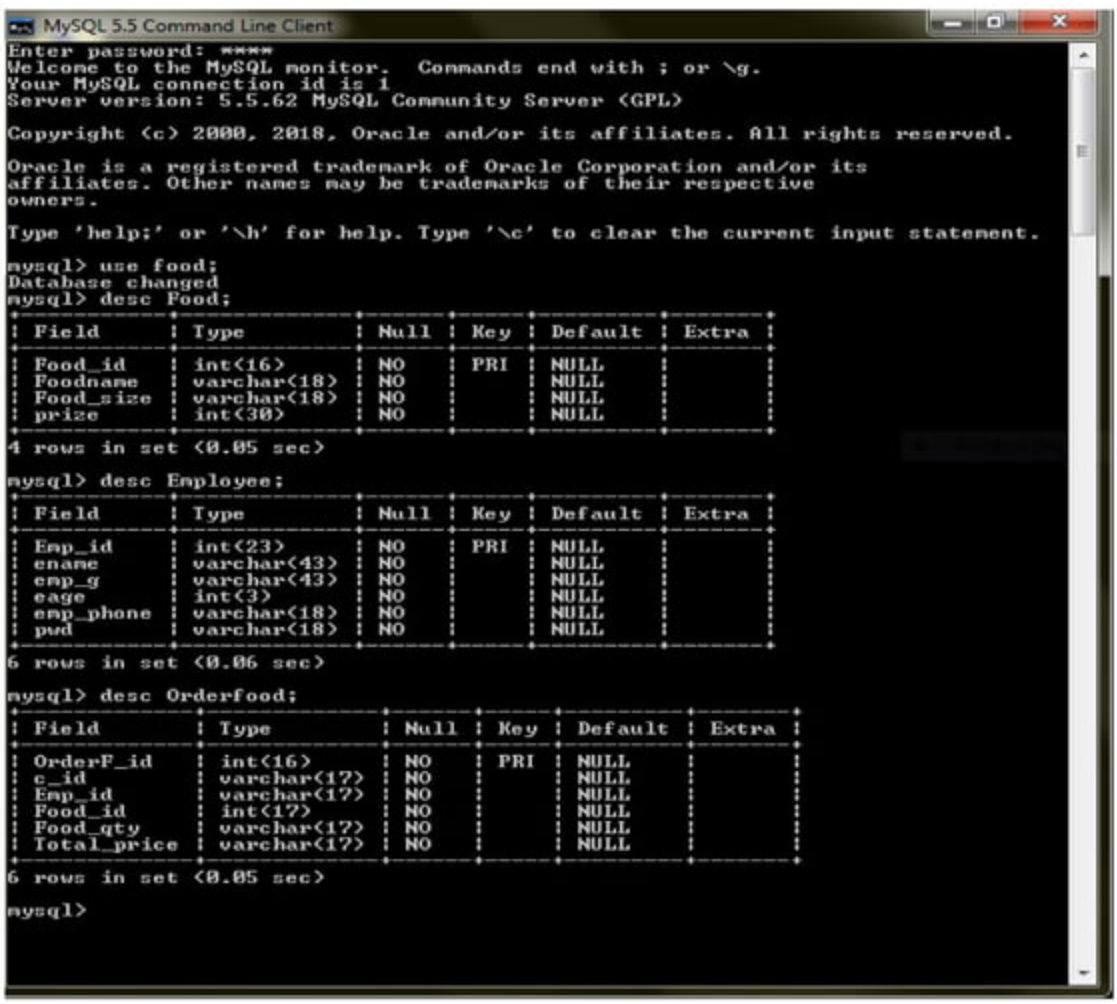
Over the decades computers and food bookings have developed gradually, changed with time. But nobody knew that a time will come when both these fields will complement each other so well. Today food booking has reached new heights by computer aided methods of design. As a result of which, computer industry has got its new customer. Computer technology is making waves in the food booking zone. Computers are a vital component of the food booking counters. Computer aided design (CAD) programs reduce the demand for manual sketches. New software programs continue to replace old manual skills. Those who lag in math can now breathe a little easier. Manually figuring of food insists that knowledge. Software programs constantly evolve. A program used today may be obsolete within several years. Being trained on today's software does not guarantee it will be used when you are ready to go out into the field. Understanding calculations is timeless, as is computer competency. Software, however, shifts rapidly.

ADVANTAGES

1. It generates the report on sales, discounts and menu.
2. Provides filter report on payments and food booking.
3. We can easily export PDF on sales, products and stocks
4. Applications can also provide excel export for bookings and discounts
5. It deals with monitoring the information and transaction of food bookings.
6. It increases the efficiency of food booking and discount.
7. It has higher efficiency of editing, adding and updating of records.
8. Provides the searching facilities on various factors.

SOURCE CODE SCREEN

DBMS: MySQL
Host: local host
User: root
Password: root
Database: Food
Table Structure: (Image below)



PYTHON CODE

```
import os
import platform
import mysql.connector
import pandas as pd
mydb=mysql.connector.connect(host="localhost",\
                             user="root",\
                             passwd ="root",\
                             database="food")
mycursor=mydb.cursor()

def Customer():
    L=[]
    c_id=int(input("Enter the customer ID number : "))
    L.append(c_id)
    name=input("Enter the Customer Name: ")
    L.append(name)
    cphone=int(input("Enter customer phone number : "))
    L.append(cphone)
    payment=int(input("Enter payment method ((1)credit
card/(2)Debit Card:) "))
    L.append(payment)
    pstatus=input("Enter the payment status : ")
    L.append(pstatus)
    email=input("Enter the email id")
    L.append(email)
    orderid=input("enter orderid")
    L.append(orderid)
    date=input("Enter the Date  : ")
    L.append(date)
    cust=(L)
    sql="insert into customer
(c_id,name,cphone,payment,pstatus,email,orderid,date)
values (%s,%s,%s,%s,%s,%s,%s,%s)"
    mycursor.execute(sql,cust)
```

```
    mydb.commit()
   # Customer Table :- C_id (PK  C_name  C_phonenum
    Payment_method (Cash/Credit Card)    Payment_status
(Paid/Unpaid)   Email   Emp_id (FK) OrderF_id (FK)  date


def Employee():
    L=[]
    Emp_id=int(input("Enter the Employee id : "))
    L.append(Emp_id)
    ename=input("Enter the Employee Name: ")
    L.append(ename)
    emp_g=input("Enter Employee Genderr : ")
    L.append(emp_g)
    eage=int(input("Enter Employee age"))
    L.append(eage)
    emp_phone=int(input("enter employee phone number"))
    L.append(emp_phone)
    pwd=input("Enter the password : ")
    L.append(pwd)
    EMP=(L)
    sql="insert into Employee
(Emp_id,ename,emp_g,eage,emp_phone,pwd) values
(%s,%s,%s,%s,%s,%s)"
    mycursor.execute(sql,EMP)
    mydb.commit()




def Food():
    L=[]
    Food_id=int(input("Enter the Food id : "))
    L.append(Food_id)
    Foodname=input("Enter the Food Name: ")
    L.append(Foodname)
    Food_size=input("Enter Food size : ")
    L.append(Food_size)
    prize=int(input("Enter Prize of Food"))
    L.append(prize)
    Food=(L)
```

```
    sql="insert into Food
(Food_id,Foodname,Food_size,prize ) values (%s,%s,%s,%s)"
    mycursor.execute(sql,Food)
    mydb.commit()

 #Food_id (PK    FoodnameFood_size    price



def OrderFood():
    L=[]
    OrderF_id=int(input("Enter the Food Order id : "))
    L.append(OrderF_id)
    C_id=input("Enter the Customer id : ")
    L.append(C_id)
    Emp_id=input("Enter Employee id: ")
    L.append(Emp_id)
    Food_id=int(input("Enter Food id"))
    L.append(Food_id)
    Food_qty=input("Enter Qty: ")
    L.append(Food_qty)
    Total_price=input("Enter Total_price")
    L.append(Total_price)
    OrderFood=(L)
    sql="insert into OrderFood
(OrderF_id,C_id,Emp_id,Food_id,Food_qty,Total_price )
values (%s,%s,%s,%s,%s,%s)"
    mycursor.execute(sql,OrderFood)
    mydb.commit()

#OrderF_id (PK) C_id (FK)    Employee_id (FK) Food_id (FK)
    Food_qtyTotal_price



def View():
    print("Select the search criteria : ")
    print("1. Employee")
    print("2. Customer")
    print("3. Food")
    print("4. Order Food")
```

```
ch=int(input("Enter the choice 1 to 4 : "))
if ch==1:
    s=int(input("enter Employee ID:"))
    rl=(s,)
    sql="select * from Employee where Emp_id=%s"
    mycursor.execute(sql,rl)
    res=mycursor.fetchall()
    for x in res:
        print(x)



elif ch==2:
    s=input("Enter Customer Name : ")
    rl=(s,)
    sql="select * from Customer where cname=%s"
    mycursor.execute(sql,rl)
    res=mycursor.fetchall()
    for x in res:
        print(x)

elif ch==3:

    sql="select * from Food"
    mycursor.execute(sql)
    res=mycursor.fetchall()
    for x in res:
        print(x)


elif ch==4:
    s=int(input("Enter Food id ID : "))
    rl=(s,)
    sql="select * from Foodorder where food_id=%s"
    mycursor.execute(sql,rl)
    res=mycursor.fetchall()
    for x in res:
        print(x)

#print("The Food details are as follows : ")
#print("(Custoemer ID, Food Name, quatity, Cost )")
```

```
    #for x in res:
        #print(x)

def feeDeposit():
    L=[]
    roll=int(input("Enter the roll number : "))
    L.append(roll)
    feedeposit=int(input("Enter the Fee to be deposited :
"))
    L.append(feedeposit)
    month=input("Enter month of fee : ")
    L.append(month)
    fee=(L)
    sql="insert into fee (roll,feedeposit,month) values
(%s,%s,%s)"
    mycursor.execute(sql,fee)
    mydb.commit()



def MenuSet():
    print("Enter 1 : To Add Employee")
    print("Enter 2 : To Add Cutomer details")
    print("Enter 3 : To Add Food Details ")
    print("Enter 4 : For Food Order")
    print("Enter 5 : For feeDeposit")
    print("Enter 6 : To view Food booking")

    try:

        userInput = int(input("Please Select An Above
Option: "))
    except ValueError:
        exit("\nHy! That's Not A Number")
    else:
        print("\n")
    if (userInput==1):
        Employee()
    elif (userInput==2):
        Customer()
    elif (userInput==3):
```

```
        Food()
    elif (userInput==4):
        OrderFood()
    elif (userInput==5):
        feeDeposit()
    elif (userInput==6):
        View()

    else:
        print("Enter correct choice. . . ")


def runAgain():
    runAgn=input("\nwant to run Again Y/N")
    while runAgn.lower()=='y':
        if(platform.system()=="Windows"):
            print(os.system('cls'))
        else:
            print(os.system('clear'))
        MenuSet()
        runAgn=input("\nwant to run Againy/n")
        print("Good Bye ... HAVE A NICE DAY")
MenuSet()
runAagain()
```

OUTPUT SCREEN

BIBLIOGRAPHY

1. http://www.google.com/
2. http://en.wikipedia.org
3. Computer science with python by Sumita Arora