|  |
| --- |
|  |
| Food Box |
| S Aparna |
|  |

XII A

|  |
| --- |
|  |

****

**NATIONAL PUBLIC SCHOOL**

**Gopalapuram , Chennai**

**BONAFIDE CERTIFICATE**

Certified to be the bonafide **Project Work** done by

Master/Miss S APARNA of Class **XII A** in **COMPUTER SCIENCE** during the academic year **2020-2021** at **NATIONAL PUBLIC SCHOOL, CHENNAI- 86.**

Signature of Internal Examiner: ………………………………………………

Date :­­­­­­­­­­­­­­­­­­­­ ……………………………………………….

**Submitted for All-India Senior School Certificate Practical Examination.**

**Board Roll number: …………………………………………**

**Date of Examination: ………………………………………..**

**Centre of Examination: ……………………………………..**

**Signature of External Examiner:** **……………………………**

**Date:……………………………..**

**ACKNOWLEDGEMENT**

I gratefully acknowledge my sincere thanks to our Computer Science Teacher for her valuable guidance and supervision throughout the project work. I also thank Madam Principal and the School Management for providing the necessary support to complete the project.

**INDEX**

|  |  |  |
| --- | --- | --- |
| S No | Topic | Page no. |
| 1. | Introduction | 1 |
| 2. | Project Listing | 4 |
| 3. | Outputs | 15 |
| 4. | Bibliography | 17 |

**INTRODUCTION**

1. **INTRODUCTION TO PYTHON:**

A programming language create by Guido Van Rossum in 1991, Python has seen a surge in it’s popularity in the past few years. An easy-to-learn object oriented language, Python serves many purposes such as web development, software development, connecting to database systems etc.

The surge in Python’s popularity in the past few years can be attributed to the following reasons:

* It can work easily on various platforms such as Linux, Mac etc.
* It has a simpler syntax.
* It is an interpreted language which makes code execution faster and easier to debug.
* It’s freely available.
* To ensure a smooth functioning one does not have to download additional libraries when using python. This is because all types of required functionality is available through various modules of Python library.

1. **INTRODUCTION TO THE PROJECT:**

As mentioned under introduction to python, the project is based on Python’s ability to connecting database servers. The project aims to facilitate exchange of information between the administration of a food company and the user/customer.

There are two types of user levels- administrator and user. The project also has two modules- inventory and sales module.

* **INVENTORY MODULE:**

The inventory module, as the name suggests, keeps a record of various information of a particular food item. The information take into consideration is the cuisine of the food, food name, price and quantity available. Each food item is identified using a food number which is unique for all the food items and serves as the primary key.

The inventory module consists of the following fields:

* Food number
* Food name
* Food type (cuisine of the food item like Chinese, south Indian, north Indian etc.)
* Price
* Quantity

It should be noted here that the access to the inventory module is restricted only till the administrator’s level. The administrator can create, modify and delete.

* **SALES MODULE:**

The access of sales module is given to both the administrator and the user. However, the control that the user has is confined to create and modifying, whereas, the administrator, in addition to create and modify also has the power to delete items. The sale module consists of following fields:

* Date
* Food number
* Food name
* Quantity
* Price
* Total Cost

At the end of the day,

The sum of quantities of a particular Food in the sales module should be tallied with the Inventory module and report should be generated on the current stock of Foods.

The total cost in the sales module should be tallied with sum of the prices of individual Foods sold.

**PROJECT LISTING**

Database: Proj\_sample

Tables:

1. Users table :

|  |  |  |
| --- | --- | --- |
| **Field** | **Data type** | **Constraints** |
| Users\_id | Integer(3) | Primary key |
| Name | Varchar(20) | Not null |
| User \_Type | Enum(“Admin”,”User”) | Not null |
| username | char(10) | Not null |
| password | Char(10) | Not null |

1. Inventory:

|  |  |  |
| --- | --- | --- |
| **Field** | **Data Type** | **Constraints** |
| Food\_Number | Varchar(7) | Primary Key |
| Food\_Name | Varchar(40) |  |
| Food\_type | Varchar(20) |  |
| Price | Float(4,2) |  |
| Initial\_Stock | Integer(5) |  |
| Current\_Stock | Integer(5) |  |

1. Sales Table:

|  |  |  |
| --- | --- | --- |
| **Field** | **Data Type** | **Constraint** |
| Food\_Number | Varchar(7) | Foreign Key |
| Food\_Name | Varchar(20) |  |
| Quantity | Integer(5) |  |
| Price | Float(4,2) |  |
| Total\_Cost | Float(6,2) |  |
| Date | date |  |

**SOURCE CODE**

import mysql.connector,sys

from tabulate import tabulate

from easygui import passwordbox,enterbox,msgbox

mydb=mysql.connector.connect(host='localhost',user='root',password='aparna',database='proj\_sample')

mycursor=mydb.cursor()

def Admin\_main\_menu():

logged\_in=True

while True:

print("What do you want to do today?")

print("1. Go to sales")

print("2. Go to inventory")

print("3. Log out")

ch=int(input("Enter choice:"))

if ch==1:

sales\_main\_menu\_admin()

elif ch==2:

inventory\_main\_menu()

elif ch==3:

log\_out()

break

else:

print("Invalid choice.Please try again")

def sales\_main\_menu\_admin():

print("Welocme Admin. Choose your option to continue.")

try:

while True:

print("1.Add an order")

print("2.Update the order")

print("3.Delete the order")

print("4.Generate the report on sales of the day")

print("5.Go to main menu")

ch=int(input("Enter your option number"))

if ch==1:

add\_sales()

elif ch==2:

update\_sales()

elif ch==3:

delete\_sales()

elif ch==4:

report\_sales()

elif ch==5:

Admin\_main\_menu()

break

else:

print("invalid choice number")

except ValueError:

print("Not a choice number")

def sales\_main\_menu\_user():

print("Welocme User . Choose your option to continue.")

try:

while logged\_in==True:

print("1.Add an order")

print("2.Update the order")

print("3.Delete the order")

print("4.Generate the report on sales of the day")

print("5.Log out")

ch=int(input("Enter your option number"))

if ch==1:

add\_sales()

elif ch==2:

update\_sales()

elif ch==3:

delete\_sales()

elif ch==4:

report\_sales()

elif ch==5:

log\_out()

break

else:

print("invalid choice number")

except ValueError:

print("Not a choice number")

def inventory\_main\_menu():

print("Welcome Admin . Choose an option to continue.")

try:

logged\_in=True

while logged\_in==True:

print("1.Create a new food item")

print("2.Update the food item")

print("3.Delete the food item")

print("4.Search the food item")

print("5.Go back to main menu")

ch=int(input("Enter your option number"))

if ch==1:

add\_inventory()

elif ch==2:

update\_inventory()

elif ch==3:

delete\_inventory()

elif ch==4:

search\_inventory()

elif ch==5:

Admin\_main\_menu()

break

else:

print("invalid choice number")

except ValueError:

print("Not a choice number")

def add\_inventory():

Food\_number=str(input("Enter food number: "))

mycursor.execute("select \* from inventory")

result=mycursor.fetchall()

exists=0

for i in result:

if i[0]==Food\_number:

print("Food number already exists")

exists=1

break

if exists==0:

Food\_Name=str(input("Enter the food name:"))

Food\_Type=str(input("Enter the food type"))

Price=float(input("Enter price: "))

Initial\_stock=int(input("Enter quantity: "))

Current\_Stock=Initial\_stock

try:

sql="insert into inventory(Food\_Number,Food\_Name,Food\_Type,Price,Initial\_stock,Current\_stock)\

values('{}','{}','{}',{},{},{})".format(Food\_number,Food\_Name,Food\_Type,Price,Initial\_stock,Current\_Stock)

mycursor.execute(sql)

mydb.commit()

print("Added Food item to inventory")

except:

print("Unable to add Food item")

def update\_inventory():

fno=input("Enter the food number of the record to be updated: ")

mycursor.execute("select food\_number from inventory")

result=mycursor.fetchall()

found=0

for i in result:

if i[0]==fno:

found=1

mycursor.execute("select \* from inventory where Food\_Number='{}'".format(fno))

details=mycursor.fetchall()

print("Which one of the following fields would you like to update?")

print("1.Food Name")

print("2.Food Type")

print("3.Price")

print("4.Initial Stock")

print("5.Back to main menu")

ch=int(input("Enter the option number to proceed:")) # variables of the format up\_field\_name are variables containing updated field value

if ch==1:

try:

print("Old Food Name:",details[0][1])

up\_food\_name=input("Enter the new food name:")

mycursor.execute("update inventory set Food\_Name='{}' where Food\_Number='{}'".format(up\_food\_name,fno))

mydb.commit()

print("Updated the food name!")

except:

print("Couldn't Update the food name.")

elif ch==2:

try:

print("Old Food Type:",details[0][2])

up\_food\_type=input("Enter the new food type:")

mycursor.execute("update inventory set Food\_Type='{}' where Food\_Number='{}'".format(up\_food\_type,fno))

mydb.commit()

print("Updated the food type!")

except:

print("Couldn't Update the food type.")

elif ch==3:

try:

print("Old Food Price:",details[0][3],"Rupees")

up\_price=float(input("Enter the new price:"))

mycursor.execute("update inventory set Price={} where Food\_Number='{}'".format(up\_price,fno))

mydb.commit()

print("Updated the Price!")

except:

print("Couldn't Update the Price.")

elif ch==4:

try:

print("Old Stock:",details[0][4])

up\_inistk=int(input("Enter the new stock Quantity:"))

mycursor.execute("update inventory set Initial\_Stock={},Current\_Stock={} where Food\_Number='{}'".format(up\_inistk,up\_inistk,fno))

mydb.commit()

print("Updated the initial stock!")

except:

print("Couldn't Update the initial stock.")

elif ch==5:

inventory\_main\_menu()

if found==0:

print("Food number not found")

def delete\_inventory():

fno=input("Enter food number to delete")

mycursor.execute("select Food\_Number from inventory")

result=mycursor.fetchall()

found=0

for i in result:

if i[0]==fno:

found=1

mycursor.execute("select Food\_Number,food\_Name,Food\_type,Price,Initial\_Stock from inventory where Food\_Number='{}'".format(fno))

details=mycursor.fetchall()

print("Your Food Details:")

print()

print("Food Number:",details[0][0])

print("Food Name:",details[0][1])

print("Food Type:",details[0][2])

print("Price:",details[0][3])

print("Initial Stock:",details[0][4])

print()

confirm=input("Are you sue that you want to delete this food item? Enter y(yes) or n(no)")

if confirm=='y' or confirm=='Y':

try:

mycursor.execute("delete from inventory where food\_number='{}'".format(fno))

mydb.commit()

print("Deleted the food item.")

except:

print("Couldn't delete the food item")

break

elif confirm=='n' or confirm=='N':

break

if found==0:

print("Couldn't find the food number")

def search\_inventory():

fno=input("Enter the food number to search for :")

mycursor.execute("select Food\_Number from inventory")

result=mycursor.fetchall()

found=0

for i in result:

if i[0]==fno:

found=1

mycursor.execute("select Food\_Number,food\_Name,Food\_type,Price,Initial\_Stock,Current\_Stock from inventory where Food\_Number='{}'".format(fno))

details=mycursor.fetchall()

print("Your Food Details:")

print()

print("Food Number:",details[0][0])

print("Food Name:",details[0][1])

print("Food Type:",details[0][2])

print("Price:",details[0][3])

print("Initial Stock:",details[0][4])

print("Current Stock:",details[0][5])

print()

break

if found==0:

print("Couldn't find the food number")

def add\_sales():

Food\_number=input('Food number-')

mycursor.execute("select count(Food\_Number) from inventory where Food\_Number= '%s'"%(Food\_number))

count=mycursor.fetchall()

if count==[(0,)]: # checks the existence of food\_no

print('Not Available')

else:

mycursor.execute("select current\_stock from inventory where Food\_Number='%s'"%(Food\_number))

cur\_stk=mycursor.fetchall() # cur\_stk is the value of the current stock in the inventory

if int(cur\_stk[0][0])==0: # checks in food item requested for is available in the inventory or not

print('Food not available')

else:

Quantity=int(input('Quantity-'))

if Quantity==0 : # if Quantity=0 then it means you don't want that food. so ensured only non zero quantity was entered

print('Cannot place a order')

elif Quantity>int(cur\_stk[0][0]): # if qty requested for was greater than what was available

print("Available amount is only-",int(cur\_stk[0][0]))

ch=input("Do you want to continue with the available quantity?(y/n)") # re-asks the user if he/she wants to replace the order

if ch=='y':

new\_qty=int(input("Enter new Quantity?"))

if new\_qty==0 or new\_qty>int(cur\_stk[0][0]):

print("Sorry, cannot place the order")

else:

try:

sales=int(cur\_stk[0][0])-new\_qty

mycursor.execute("insert into sales(Food\_Number,Quantity) values('%s',%d)"%(Food\_number,new\_qty))

mycursor.execute("update sales s , inventory i set s.Food\_Name=i.Food\_Name, s.Price=i.Price,Date=now(),\

Total\_cost=i.Price\*s.Quantity where s.Food\_Number=i.Food\_Number")

mycursor.execute("update inventory set current\_stock= %d where Food\_Number='%s'"%(sales,Food\_number))

mydb.commit()

print("Added the order")

except:

print("Unable to add the order")

elif ch=='n':

pass

else:

try:

sales=int(cur\_stk[0][0])-Quantity

mycursor.execute("insert into sales(Food\_Number,Quantity) values('%s',%d)"%(Food\_number,Quantity))

mycursor.execute("update sales s , inventory i set s.Food\_Name=i.Food\_Name, s.Price=i.Price,Date=current\_timestamp,\

Total\_cost=i.Price\*s.Quantity where s.Food\_Number=i.Food\_Number")

mycursor.execute("update inventory set current\_stock= %d where Food\_Number='%s'"%(sales,Food\_number))

mydb.commit()

print("Added the order")

except:

print("Unable to add order")

def update\_sales():

food\_no=input("Enter Food number -")

mycursor.execute("select \* from sales where Food\_Number='%s'"%(food\_no))

result=mycursor.fetchall()

if mycursor.rowcount<=0:

print("Food not found")

elif mycursor.rowcount>0:

mycursor.execute("select current\_stock from inventory where Food\_Number='%s'"%(food\_no))

count\_qty=mycursor.fetchall()

if int(count\_qty[0][0])==0:

print("Food is not available")

else:

print("Your food details:")

print("Food Number-",result[0][0])

print("Food Name-",result[0][1])

print("Quantity-",result[0][2])

old\_qty=int(result[0][2])

print("Total cost-",result[0][4],"rupees")

ch=input("Do you want to re-enter a new qty(y/n)?")

if ch=='y'or ch=='Y':

new\_qty=int(input("Enter the qty to update:"))

if new\_qty==0:

print("Cannot place the order.")

elif new\_qty>int(count\_qty[0][0]):

print("Available Quantity is -",int(count\_qty[0][0]))

choice=input("Do you still want to re-enter the quantity?(y/n)")

re\_new\_qty=0

if choice=='y' or choice=='Y':

re\_new\_qty=0 # re-enter new qty

re\_new\_qty=int(input("Enter the qty to update:"))

if re\_new\_qty==0 or re\_new\_qty>int(count\_qty[0][0]):

print("Sorry, cannot place the order")

else:

mycursor.execute("update sales set Quantity=%s,Total\_cost=Price\*%s where Food\_Number='%s'"%(re\_new\_qty,re\_new\_qty,food\_no))

qty\_diff=0

if re\_new\_qty>old\_qty:

qty\_diff=re\_new\_qty-old\_qty

mycursor.execute("update inventory set current\_stock=current\_stock-%s where food\_number='%s'"%(qty\_diff,food\_no))

mydb.commit()

elif re\_new\_qty<old\_qty:

qty\_diff=old\_qty-re\_new\_qty

mycursor.execute("update inventory set current\_stock=current\_stock+%s where food\_number='%s'"%(qty\_diff,food\_no))

mydb.commit()

print("Updated the order")

elif choice=='n'or choice=='N':

pass

else:

print("Invalid input")

else:

mycursor.execute("update sales set Quantity=%s,Total\_cost=Price\*%s where Food\_Number='%s'"%(new\_qty,new\_qty,food\_no))

qty\_diff=0

if new\_qty>old\_qty:

qty\_diff=new\_qty-old\_qty

mycursor.execute("update inventory set current\_stock=current\_stock-%s where food\_number='%s'"%(qty\_diff,food\_no))

mydb.commit()

elif new\_qty<old\_qty:

qty\_diff=old\_qty-new\_qty

mycursor.execute("update inventory set current\_stock=current\_stock+%s where food\_number='%s'"%(qty\_diff,food\_no))

mydb.commit()

print("Updated the order")

elif ch=='n'or ch=='N':

pass # main menu back

else:

print("invalid option")

'''---------------------------------------------------------------------------------------------------------------------------------------------------'''

def delete\_sales():

try:

itno=input("Enter the item number of the order to be deleted")

mycursor.execute("select \* from sales where Food\_Number='%s'"%(itno))

result=mycursor.fetchall()

if mycursor.rowcount<=0:

print("Sorry! The above mentioned item no. has no matches")

else:

for i in result:

print("Your Food Details:")

print("Food Name:", i[1])

print("Quantity:",i[2])

print("Price:",i[3])

print("Total Cost:",i[4])

print("Date:",i[5])

del\_qty=i[4]

ans=input("Are you sure you want to delete the above mentioned item(y/n)")

if ans=='y' or ans=='Y':

mycursor.execute("delete from Sales where Food\_Number='%s'"%(itno))

mycursor.execute("update inventory set current\_stock=current\_stock+'%d' where Food\_Number='%s'"%(del\_qty,itno))

mydb.commit()

print("Item has been deleted")

elif ans=='N' or ans=='n':

break

else:

print("Invalid input")

except:

print("Error occured!Could not delete item")

'''---------------------------------------------------------------------------------------------------------------------------------------------'''

def report\_sales():

mycursor.execute("select food\_number from sales s")

r1=list(mycursor.fetchall())

if r1==[]:

print("Report cannot be generated")

else:

mycursor.execute("select food\_number from inventory")

r2=list(mycursor.fetchall())

food\_no\_sales=[]

food\_no\_inventory=[]

for i in r1: # converting the food\_nos from sales as a list of string elements for easier tranversal

food\_no\_sales.append(i[0])

for i in r2: # converting the food\_nos from inventory as a list of string elements for easier tranversal

food\_no\_inventory.append(i[0])

total\_sales=0 #total\_sales is the total sales in a day

headers=["SNo","Food Name","Food Type","Initial Stock","Current Stock","Sales","Total (Sales+Current Stock)","Unit Price","Total Cost"]

table=[]

row=[]

for j in range(len(food\_no\_sales)):

if food\_no\_sales[j] in food\_no\_inventory:

mycursor.execute("select s.food\_name,i.Food\_Type,i.initial\_stock,i.current\_stock,s.Quantity,i.current\_stock+s.Quantity,s.Price,s.Total\_cost from sales s ,inventory i where s.food\_number='%s' and i.food\_number='%s'"%(food\_no\_sales[j],food\_no\_sales[j]))

result=mycursor.fetchall()

food\_name=result[0][0]

food\_type=result[0][1] #initialising variables for printing

initial\_stk=result[0][2]

current\_stk=result[0][3]

sales=result[0][4]

total\_salcurstk=result[0][5]

unit\_price=result[0][6]

tot\_cost=result[0][7]

total\_sales+=tot\_cost

sno=j+1

row=[sno,food\_name,food\_type,initial\_stk,current\_stk,sales,total\_salcurstk,unit\_price,tot\_cost] #Making a row

table.append(row) #Making the table

print(tabulate(table,headers,tablefmt="psql")) #\*\*-tabulated data is printed

print("%70s"%"Total Sales of the day is =",total\_sales,"Rupees","%70s"%" ") # total sales of the day is the sum of the total\_cost column values

def log\_out():

print("Logging out")

sys.exit(0)

#login

for i in range(3):

username=enterbox("Enter Username:","Welcome to Star Restauraunt")

password=passwordbox("Enter Password:")

mycursor.execute("select Name,USer\_Type,username,password from users")

result=mycursor.fetchall()

logged\_in=False

for i in result:

if i[2]==username and i[3]==password:

print("Welcome",i[0])

logged\_in=True

break

if logged\_in==True:

break

if logged\_in==False:

msgbox("Your username or password is incorrect. Please try again.","Incorrect Username or Password")

continue

if logged\_in==False:

msgbox("Sorry, all your attempts are exhausted.")

#main\_menu() function call

try:

if logged\_in==True:

for i in result:

if i[2]==username:

user\_type=i[1]

break

except NameError:

sys.exit(0)

if user\_type=="Admin":

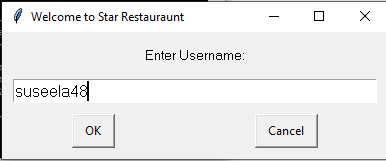
Admin\_main\_menu()

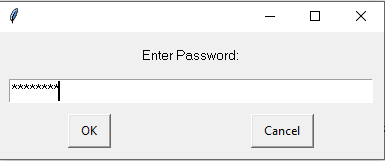
if user\_type=='User':

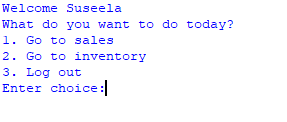
sales\_main\_menu\_user()

OUTPUTS

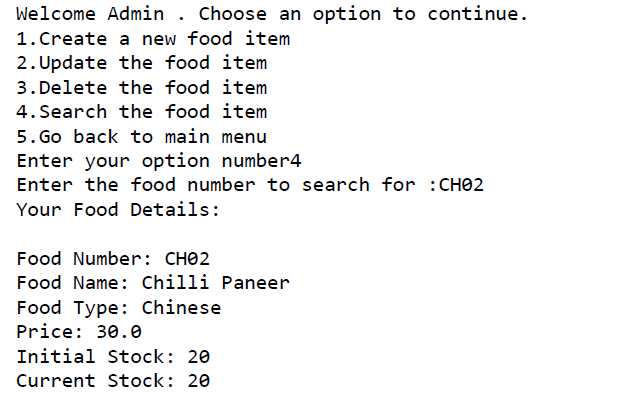
Logging in:



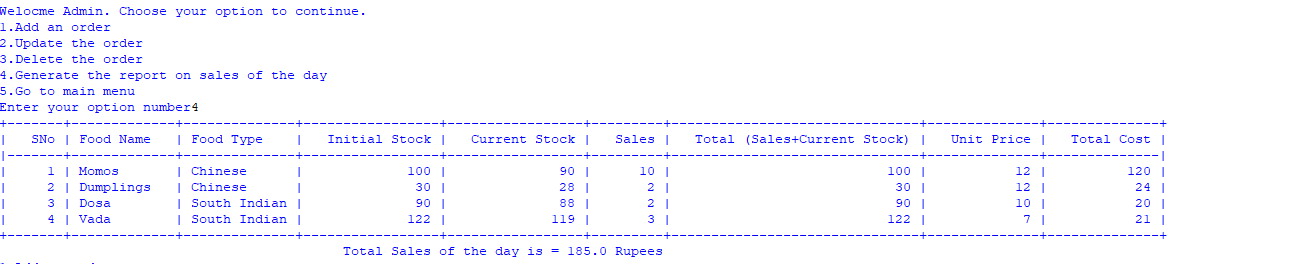




**Admin main menu – search operation**



**Sales report**



**BIBILIOGRAPHY**

1. <https://www.geeksforgeeks.org/python-easygui-module-introduction/>
2. <https://www.geeksforgeeks.org/sys-path-in-python/>
3. Computer Science with Python – Sumita Arora (2020 edition)

Published By:Gagan Kapur

Dhanpat Rai & Co.

1. Computer Science with Python – Preeti Arora (2020 edition)

Published by: Sultan Chand & Sons