System Implementations

**Recommended System Requirements**

Processors: Intel® Core™ i3 processor 4300M at 2.60 GHz.

Disk space: 4 to 8 GB.

Operating systems: Windows® 10, MACOS, and UBUNTU.

Python Versions: 3.X.X or Higher.

**Minimum System Requirements**

Processors: Intel Atom® processor or Intel® Core™ i3 processor.

Disk space: 1 GB.

Operating systems: Windows 7 or later, MACOS, and UBUNTU.

Python Versions: 2.7.X, 3.9.X.

ACKNOWLEDGEMENTTTT

First and foremost, praises and thanks to the God, the Almighty, for His showers of blessings throughout my research work to complete the research successfully.

We would like to express my deep and sincere gratitude to my subject teacher, Mr. Amit Udiwal, for giving me the opportunity to do research and providing invaluable guidance throughout this research. His dynamism, vision, sincerity and motivation have deeply inspired me. He has taught me the methodology to carry out the research and to present the research works as clearly as and honour to work and study under his guidance. We are very much thankful to our Sr. Jasmin for giving valuable time and moral support to develop this software. We would like to take opportunity to extend my sincere thanks and gratitude to our parents for being a source of inspiration and providing time and freedom to develop this software project. We also feel indebted to my friends for the valuable suggestions during the project work.

Vishi Jain

[Roll No.

Class XII

CERTIFICATE

This is to certify that the project on ‘**Fashion Store Management System**’ is a work done by Vishi Jain fulfilment of CBSE’S AISSCE EXAMINATION 2020¢and has been carried out under my direct supervision and guidance. This report or a similar report on the topic has not been submitted for any other examination and does not form any other examination and does not form any other course undergone by the candidate.

Name:

Vishi Jain [Roll No.

………………….

Signature of Teacher / Guide

Name: Mr. Amit Udiwal

Designation:

………………. ….………………

REFERENCE

The order to work on this project on ‘**Fashion Store Management System**’ the following books & literature are referred by me during the various phrases of department of the project.

• http://www.python.org/.

• http://www.itsourcecode.org/.

• http://www.wikipedia.org/.

• Informatics Practices for Class XII

- By Sumita Arora

• Together with informatics practices.

Other than the above mentioned books, the suggestions and supervision of my teacher and my class experience also helped me to develop this software project.

Introduction

The **Fashion Store Management System Project in Python** was created specifically for adding Fashion item details. The system clarifies the fundamental principle of storing and creating item information.

A Fashion Store, often known as a clothing store, is an establishment that offers ready-made garments. A boutique is a small store that sells high-end or designer apparel. An outfitter is a store that sells clothing for a specific market, such as school uniforms or outdoor activities.

Objective and

Scope of The Project

The main objective of the python project on fashion store management is to manage the details of sales, discounts, payments, products, and inventory digitally. 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 sales, discounts, stock, and payments. It tracks all the details about stocks, products, and inventory; it also prints various reports as per input given by the user.

***Functions:***

• Manage the information of Inventory

• Shows the information and description of the Customer, Supplier.

• To increase efficiency of managing the Customer, Inventory.

Fashion Store

Management System

import os

import platform

import mysql.connector

import pandas as pd

import datetime

mydb=mysql.connector.connect(host="localhost",\

user="root",\

passwd="root",\

database="fashion")

mycursor=mydb.cursor()

def AddProduct():

L=[]

stk=[]

pid=input("Enter the Product ID : ")

L.append(pid)

IName=input("Enter the Product Name : ")

L.append(IName)

brnd=input("Enter the Product Brand Name : ")

L.append(brnd)

fr=input("Enter Male/Female/Kids : ")

L.append(fr)

sn=input("Enter Winter/Summer : ")

L.append(sn)

rate=int(input("Enter the Rates for Product :"))

L.append(rate)

product=(L)

sql="Insert into product (product\_id,PName,brand,Product\_for,Season,rate)values(%s,%s,%s,%s,%s,%s)"

mycursor.execute(sql,product)

mydb.commit()

stk.append(pid)

stk.append(0)

stk.append("No")

st=(stk)

sql="insert into stock(item\_id, Instock, status) values(%s,%s,%s)"

mycursor.execute(sql,st)

mydb.commit()

print("One Product inserted ")

def EditProduct():

pid=input("Enter product ID to be edited : ")

sql="select \* from product where product\_id=%s"

ed=(pid,)

mycursor.execute(sql,ed)

res=mycursor.fetchall()

for x in res:

print(x)

print("")

fld=input("Enter the field which you want to edit : ")

val=input("Enter the value you want to set : ")

sql="Update product set " + fld +"='" + val + "' where product\_id='" + pid + "'"

sq=sql

mycursor.execute(sql)

print("Editing Don : ")

print("After correction the record is : ")

sql="select \* from product where product\_id=%s"

ed=(pid,)

mycursor.execute(sql,ed)

res=mycursor.fetchall()

for x in res:

print(x)

mydb.commit()

def DelProduct():

pid=input("Enter the Product)id to be deleted : ")

sql="delete from sales where item\_id=%s"

id=(pid,)

mycursor.execute(sql,id)

mydb.commit()

sql="delete from purchase where item\_id=%s"

mycursor.execute(sql,id)

mydb.commit()

sql="delete from stock where item\_id=%s"

mycursor.execute(sql,id)

mydb.commit()

sql="delete from product where product\_id=%s"

mycursor.execute(sql,id)

mydb.commit()

print("One Item Deleted")

def ViewProduct():

print("Display Menu: Select the category to display the data")

print("1. All Details")

print("2. Product Name:")

print("3. Product Brand:")

print("4. Product For:")

print("5. Product Season:")

print("6. Product ID:")

x=0

ch=int(input("Enter your choice to display : "))

if ch==1:

sql="select \* from product"

mycursor.execute(sql)

res=mycursor.fetchall()

for x in res:

print(x)

x=1

elif ch==2:

var='PName'

val=input("Enter the name of Product : ")

elif ch==3:

var='brand'

val=input("Enter the name of Brand : ")

elif ch==4:

var='Product\_for'

val=input("Enter Male/Femal/Kids : ")

elif ch==5:

var='season'

val=input("Enter the Season : ")

elif ch==6:

var='product\_id'

val=input("Enter the Product\_id : ")

if x==0:

sql="select \* from product where " + var + " = %s"

sq=sql

tp=(val,)

mycursor.execute(sq,tp)

res=mycursor.fetchall()

for x in res:

print(x)

def PurchaseProduct():

mn=""

dy=""

now=datetime.datetime.now()

purchaseID="P"+str(now.year)+str(now.month)+str(now.day)+str(now.hour)+str(now.minute)+str(now.second)

L=[]

Lst=[]

L.append(purchaseID)

itemId=input("Enter Product ID : ")

L.append(itemId)

itemNo=int(input("Enter the number of Items : "))

L.append(itemNo)

sql="select rate from product where product\_id=%s"

pid=(itemId,)

mycursor.execute(sql,pid)

res=mycursor.fetchone()

for x in res:

print("rate is : ", x)

amount=x\*itemNo

print("Amount is :", amount)

L.append(amount)

mnth=now.month

if mnth<=9:

mn="0"+str(mnth)

else:

mn=str(mnth)

day=now.day

if day<=9:

dy="0"+str(day)

else:

dy=str(day)

dt=str(now.year)+"-"+mn+"-"+dy

L.append(dt)

tp=(L)

sql="insert into purchase(purchase\_id,item\_id,no\_of\_items,amount,Purchase\_date)values(%s,%s,%s,%s,%s)"

mycursor.execute(sql,tp)

mydb.commit()

sql="Select Instock from stock where item\_id=%s"

mycursor.execute(sql,pid)

res=mycursor.fetchall()

status="No"

for x in res:

print(x)

instock=x[0]+itemNo

if instock>0:

status="Yes"

Lst.append(instock)

Lst.append(status)

Lst.append(itemId)

tp=(Lst)

sql="update stock set instock=%s,status=%s where item\_id=%s"

mycursor.execute(sql,tp)

mydb.commit()

print("1 Item purchased and saved in Database")

def ViewPurchase():

item=input("Enter Product Name : ")

sql="select product.product\_id, product.PName,product.brand,purchase.no\_of\_items,purchase.purchase\_date,purchase.amount from product INNER JOIN purchase ON product.product\_id=purchase.item\_id and product.PName=%s"

itm=(item,)

mycursor.execute(sql,itm)

res=mycursor.fetchall()

for x in res:

print(x)

def ViewStock():

item=input("Enter Product Name : ")

sql="select product.product\_id,product.PName,stock.Instock,\

stock.status from stock, product where \

product.product\_id=stock.item\_id and product.PName=%s"

itm=(item,)

mycursor.execute(sql,itm)

res=mycursor.fetchall()

for x in res:

print(x)

def SaleProduct():

now=datetime.datetime.now()

saleID="S"+str(now.year)+str(now.month)+str(now.day)+str(now.hour)+str(now.minute)+str(now.second)

L=[]

L.append(saleID)

itemId=input("Enter Product ID : ")

L.append(itemId)

itemNo=int(input("Enter the number of Items : "))

L.append(itemNo)

sql="select rate from product where product\_id=%s"

pid=(itemId,)

mycursor.execute(sql,pid)

res=mycursor.fetchall()

for x in res:

print("The rate of item is :",x)

dis=int(input("Enter the discount : "))

saleRate=x[0]-(x[0]\*dis/100)

L.append(saleRate)

amount=itemNo\*saleRate

L.append(amount)

mnth=now.month

if mnth<=9:

mn="0"+str(mnth)

else:

mn=str(mnth)

day=now.day

if day<=9:

dy="0"+str(day)

else:

dy=str(day)

dt=str(now.year)+"-"+mn+"-"+dy

L.append(dt)

tp=(L)

sql="insert into sales (sale\_id, item\_id,no\_of\_item\_sold,\

sale\_rate,amount,date\_of\_sale) values(%s,%s,%s,%s,%s,%s)"

mycursor.execute(sql,tp)

mydb.commit()

sql="Select Instock from stock where item\_id=%s"

mycursor.execute(sql,pid)

res=mycursor.fetchall()

for x in res:

print("Total Items in Stock are : ",x)

instock=x[0]-itemNo

if instock>0:

status="Yes"

tp=(instock,status,itemId)

sql="update stock set instock=%s,status=%s where item\_id=%s"

print("Remaining Items in Stock are : ",instock)

mycursor.execute(sql,tp)

mydb.commit()

def ViewSales():

item=input("Enter Product Name : ")

sql="select product.product\_id, product.PName,product.brand,\

sales.no\_of\_item\_sold,sales.date\_of\_sale,sales.amount \

from sales, product where product.product\_id=sales.item\_id \

and product.PName=%s"

itm=(item,)

mycursor.execute(sql,itm)

res=mycursor.fetchall()

for x in res:

print(x)

def MenuSet(): #Function For The SFashion Store System

print("Enter 1 : To Add Product ")

print("Enter 2 : To Edit Product ")

print("Enter 3 : To Delete Product ")

print("Enter 4 : To View Product ")

print("Enter 5 : To Purchase Product")

print("Enter 6 : To View Purchases")

print("Enter 7 : To View Stock Detials")

print("Enter 8 : To Sale the item")

print("Enter 9 : To View Sales Detials")

try: #Using Exceptions For Validation

userInput = int(input("Please Select An Above Option: ")) #Will Take Input From User

except ValueError:

exit("\nHy! That's Not A Number") #Error Message

else:

print("\n") #Print New Line

if(userInput == 1):

AddProduct()

elif(userInput == 2):

EditProduct()

elif (userInput==3):

DelProduct()

elif (userInput==4):

ViewProduct()

elif (userInput==5):

PurchaseProduct()

elif (userInput==6):

ViewPurchase()

elif (userInput==7):

ViewStock()

elif (userInput==8):

SaleProduct()

elif (userInput==9):

ViewSales()

else:

print("Enter correct choice. . . ")

MenuSet()

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 Again Y/n: ")

runAgain()