CERTIFICATE

This is to certify that Chaitanya Sharma of class XII-H has successfully prepared and completed the project on the topic “Store Manager”. The report is the result of his efforts and endeavors and has been accepted as the final project for the subject Computer Science of class XII. He has prepared the project under my guidance.

Signature of Teacher

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

INDEX

1. Acknowledgement
2. Synopsis
3. Output
4. Python Code
5. MySQL Code and Tables

ACKNOWLEDGEMENT

I would like to express special thanks of gratitude to my Computer Science teacher, ***Miss Chanchal Chandna***, under whose guidance I was able to complete my project on the topic ***‘STORE MANAGER’.***

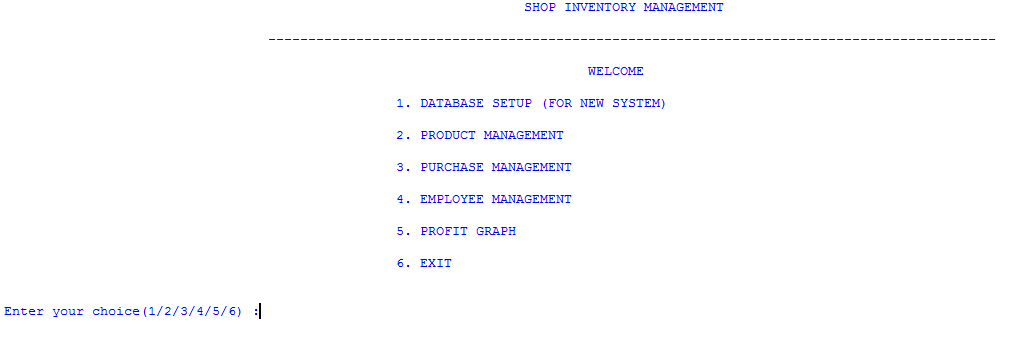
This project has proven to be a learning curve for me as I was able to gather so much information and I am very thankful for the opportunity.

SYNOPSIS

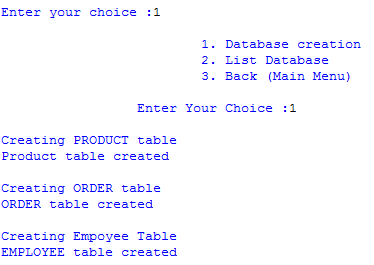
This is a powerful task manager. It helps the user manage various tasks such as managing products, orders, purchases and adding/updating Employee’s information. It is a fun way for the user to manage several tasks using a single program.

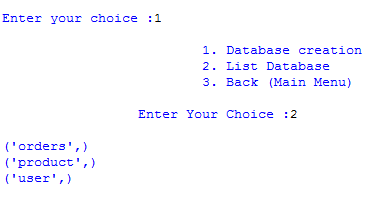
OUTPUT

Main Menu

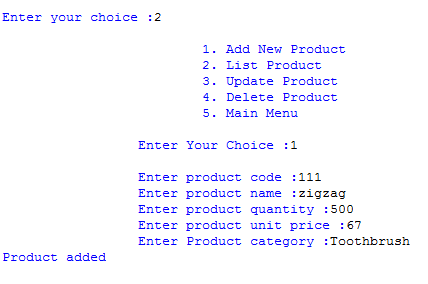


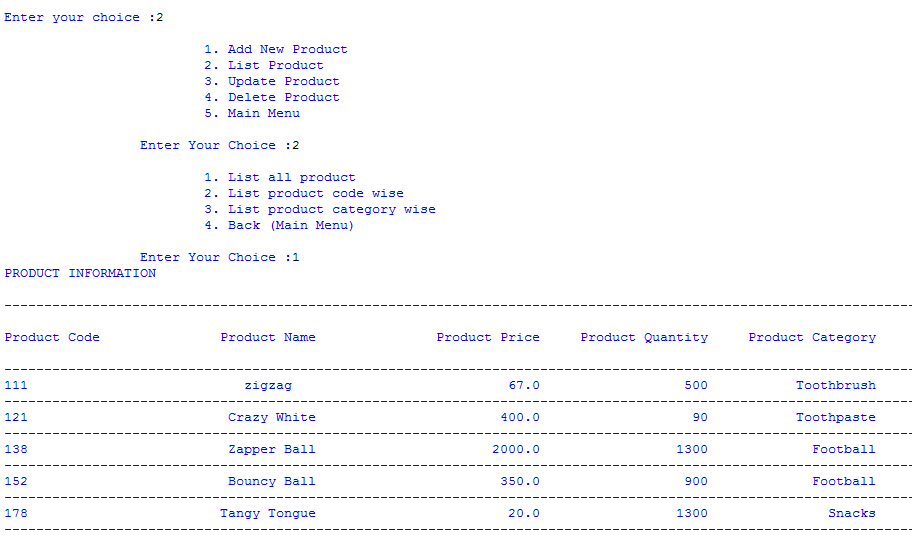
**1. Database setup**



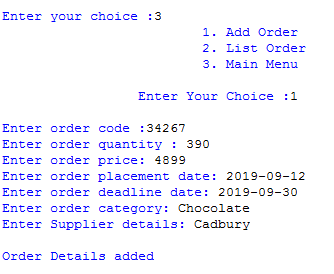


2. Product management

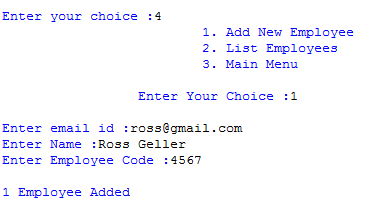


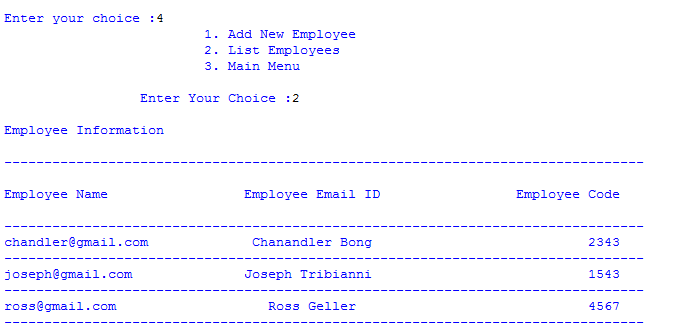


3. Purchase Management

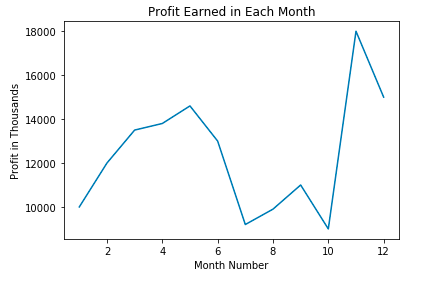


4. Employee Management





5. Profit Graph



PYTHON CODE

while True:

import os

import time

import pymysql

import mysql

import matplotlib.pyplot as plt

#PRODUCT MANAGEMENT MENU

def product\_mgmt():

print()

print("\t\t\t 1. Add New Product")

print("\t\t\t 2. List Product")

print("\t\t\t 3. Update Product")

print("\t\t\t 4. Delete Product")

print("\t\t\t 5. Main Menu")

print()

p = input("\t\t Enter Your Choice :")

print()

if p == '1':

add\_product()

elif p == '2':

search\_product()

elif p == '3':

update\_product()

elif p == '4':

delete\_product()

else :

main()

#PURCHASE MANAGEMENT MENU

def purchase\_mgmt():

print("\t\t\t 1. Add Order")

print("\t\t\t 2. List Order")

print("\t\t\t 3. Main Menu")

print()

o = input("\t\t Enter Your Choice :")

print()

if o == '1':

add\_order()

elif o == '2':

list\_order()

else :

main()

#USER MANAGEMENT MENU

def user\_mgmt():

print("\t\t\t 1. Add New Employee")

print("\t\t\t 2. List Employees")

print("\t\t\t 3. Main Menu")

print()

u = input("\t\t Enter Your Choice :")

print()

if u == '1':

add\_user()

elif u == '2':

list\_user()

else:

main()

#CREATING DATABASES

def create\_database():

mydb = pymysql.connect(host="localhost", user="root", password="rahul", database="school")

mycursor = mydb.cursor()

print("Creating PRODUCT table")

sql = "CREATE TABLE if not exists product(pcode int(4) PRIMARY KEY,pname char(30) NOT NULL,price float(8,2),pqty int(4),pcat char(30));"

mycursor.execute(sql)

time.sleep(0.5)

print("Product table created")

print()

print("Creating ORDER table")

sql = "CREATE TABLE if not exists orders(orderid int(4) PRIMARY KEY not null,orderdate DATE, orderfinaldate DATE, pprice float(8,2),pqty int(4),psupplier char(50),pcat char(30));"

mycursor.execute(sql)

time.sleep(0.5)

print("ORDER table created")

print()

print("Creating Empoyee Table")

sql = "CREATE TABLE if not exists user(uid char(40) PRIMARY KEY,uname char(30) NOT NULL,upwd char(30));"

mycursor.execute(sql)

time.sleep(0.5)

print("EMPLOYEE table created")

#LISTING DATABASES

def list\_database():

mydb = pymysql.connect(host="localhost", user="root", password="rahul", database="school")

mycursor = mydb.cursor()

sql = "show tables;"

mycursor.execute(sql)

for i in mycursor:

print(i)

#ADDING ORDERS

def add\_order():

mydb = pymysql.connect(host="localhost", user="root", password="rahul", database="school")

mycursor = mydb.cursor()

now = datetime.datetime.now()

sql = "INSERT INTO orders (orderid, orderdate, orderfinaldate, pprice, pqty, psupplier, pcat) values(%s,%s,%s,%s,%s,%s,%s)"

try:

code = int(input("Enter order code :"))

old = now.year + now.month + now.day + now.hour + now.minute + now.second

qty = input("Enter order quantity : ")

price = input("Enter order price: ")

orderdate = input("Enter order placement date: ")

orderfinaldate = input("Enter order deadline date: ")

cat = input("Enter order category: ")

supplier = input("Enter Supplier details: ")

print()

print("Order Details added")

val = (code, orderdate, orderfinaldate, price, qty, supplier, cat)

mycursor.execute(sql,val)

mydb.commit()

except:

print("Order Couldnt't be Added. Please try again")

mycursor.rollback()

#LISTING ORDERS

def list\_order():

mydb = pymysql.connect(host="localhost", user="root", password="rahul", database="school")

mycursor = mydb.cursor()

sql = "select\*from orders;"

try:

mycursor.execute(sql)

rows=mycursor.fetchall()

print("PRODUCT INFORMATION")

print()

print("-"\*150)

print()

print("{0:<20} {1:^25} {2:>20} {3:>20} {4:>20} {5:>20} {6:>20} "

.format("Order Code", "Order Placement Date", "Order Deadline", "Order Cost", "Order Quantity", "Supplier", "Order Category"))

print()

print("-"\*150)

for row in rows:

ocode=row[0]

opd=row[1]

opdd=row[2]

ocost=row[3]

oqty=row[4]

supp=row[5]

cat=row[6]

print("{0:<20} {1:^25} {2:>20} {3:>20} {4:>20} {5:>20} {6:>20}"\

.format(ocode, opd, opdd, ocost, oqty, supp, cat))

print("-"\*150)

except:

print("Data unable to load")

#DATABASE MANAGEMENT

def db\_mgmt():

print()

print("\t\t\t 1. Database creation")

print("\t\t\t 2. List Database")

print("\t\t\t 3. Back (Main Menu)")

print()

p = input("\t\t Enter Your Choice :")

print()

if p == '1':

create\_database()

elif p == '2':

list\_database()

else :

main()

#ADDING PRODUCTS

def add\_product():

mydb = pymysql.connect(host="localhost", user="root", password="rahul", database="school")

mycursor = mydb.cursor()

sql = "INSERT INTO product(pcode,pname,price,pqty,pcat) values(%s,%s,%s,%s,%s)"

code = input("\t\t Enter product code :")

search = "SELECT count(\*) FROM product WHERE pcode=%s;"

val = (code,)

mycursor.execute(search, val)

for x in mycursor:

cnt = x[0]

try:

if cnt == 0:

name = input("\t\t Enter product name :")

qty = input("\t\t Enter product quantity :")

price = input("\t\t Enter product unit price :")

cat = input("\t\t Enter Product category :")

val = (code, name, price, qty, cat)

mycursor.execute(sql, val)

mydb.commit()

print("Product added")

else:

print("\t\t Product already exist")

except:

print("Error occured...please try again")

#UPDATING PRODUCT

def update\_product():

mydb = pymysql.connect(host="localhost", user="root", password="rahul", database="school")

mycursor = mydb.cursor()

print()

try:

code = int(input("Enter the product code :"))

qty = int(input("Enter the quantity :"))

price = int(input("Enter product price :"))

sql = "UPDATE product SET pqty=pqty+% WHERE pcode=%s;"

val = [qty, code]

mycursor.execute(sql, val)

mydb.commit()

print("\t\t Product details updated")

except:

print("Error occured...please try again")

#DELETING PRODUCT INFORMATION

def delete\_product():

mydb = pymysql.connect(host="localhost", user="root", password="rahul", database="school")

mycursor = mydb.cursor()

try:

code = int(input("Enter the product code :"))

print()

sql = "DELETE FROM product WHERE pcode = %s;"

val = (code)

mycursor.execute(sql, val)

mydb.commit()

print(mycursor.rowcount, "record(s) deleted")

except:

print("Error occured...please try again")

def search\_product():

while True:

print("\t\t\t 1. List all product")

print("\t\t\t 2. List product code wise")

print("\t\t\t 3. List product category wise")

print("\t\t\t 4. Back (Main Menu)")

print()

s = input("\t\t Enter Your Choice :")

if s == '1':

list\_product()

if s == '2':

code = int(input(" Enter product code :"))

list\_prcode(code)

if s == '3':

cat = input("Enter category :")

list\_prcat(cat)

if s == '4':

main()

#LISTING PRODUCT INFORMATION

def list\_product():

mydb = pymysql.connect(host="localhost", user="root", password="rahul", database="school")

mycursor = mydb.cursor()

sql="SELECT\*FROM PRODUCT"

try:

mycursor.execute(sql)

rows=mycursor.fetchall()

print("PRODUCT INFORMATION")

print()

print("-"\*150)

print()

print("{0:<20} {1:^25} {2:>20} {3:>20} {4:>20} "

.format("Product Code", "Product Name", "Product Price", "Product Quantity", "Product Category"))

print()

print("-"\*150)

for row in rows:

pcode=row[0]

pname=row[1]

price=row[2]

pqty=row[3]

pcat=row[4]

print("{0:<20} {1:^25} {2:>20} {3:>20} {4:>20} "\

.format(pcode, pname, price, pqty, pcat))

print("-"\*150)

except:

print("Data unable to load")

#LISTING PRODUCT INFORMATION ORDERED BY PRODUCT CODE

def list\_prcode(code):

mydb = pymysql.connect(host="localhost", user="root", password="rahul", database="school")

mycursor = mydb.cursor()

sql = "SELECT \* from product WHERE pcode=%s"

try:

mycursor.execute(sql)

rows=mycursor.fetchall()

print("PRODUCT INFORMATION")

print()

print("-"\*150)

print()

print("{0:<20} {1:^25} {2:>20} {3:>20} {4:>20} "

.format("Product Code", "Product Name", "Product Price", "Product Quantity", "Product Category"))

print()

print("-"\*150)

for row in rows:

pcode=row[0]

pname=row[1]

price=row[2]

pqty=row[3]

pcat=row[4]

print("{0:<20} {1:^25} {2:>20} {3:>20} {4:>20} "\

.format(pcode, pname, price, pqty, pcat))

print("-"\*150)

except:

print("Data unable to load")

#LISTING PRODUCT INFORMATION ORDERED BY CATEGORY

def list\_prcat(cat):

mydb = pymysql.connect(host="localhost", user="root", password="rahul", database="school")

mycursor = mydb.cursor()

print(cat)

sql = "SELECT \* from product WHERE pcat =%s"

val = (cat,)

mycursor.execute(sql, val)

clrscr()

print("\t\t\t\t PRODUCT DETAILS")

print("\t\t", "-" \* 47)

print("\t\t code name price quantity category")

print("\t\t", "-" \* 47)

for i in mycursor:

print("\t\t", i[0], "\t", i[1], "\t", i[2], "\t", i[3], "\t\t", i[4])

print("\t\t", "-" \* 47)

#ADDING EMPLOYEE INFORMATION

def add\_user():

mydb = pymysql.connect(host="localhost", user="root", password="rahul", database="school")

mycursor = mydb.cursor()

uid = input("Enter email id :")

name = input("Enter Name :")

password = input("Enter Employee Code :")

print()

sql = "INSERT INTO user(uid,uname,upwd) values (%s,%s,%s);"

val = (uid, name, password)

mycursor.execute(sql, val)

mydb.commit()

print(mycursor.rowcount, "Employee Added")

#LISTING EMPLOYEE INFORMATION

def list\_user():

mydb = pymysql.connect(host="localhost", user="root", password="rahul", database="school")

mycursor = mydb.cursor()

sql = "SELECT\*from user"

try:

mycursor.execute(sql)

rows=mycursor.fetchall()

print("Employee Information")

print()

print("-"\*80)

print()

print("{0:<20} {1:^35} {2:>20} "

.format("Employee Name", "Employee Email ID", "Employee Code"))

print()

print("-"\*80)

for row in rows:

pcode=row[0]

pname=row[1]

price=row[2]

print("{0:<20} {1:^35} {2:>20} "\

.format(pcode, pname, price))

print("-"\*80)

except:

print("Data unable to load")

def clrscr():

print("\n" \* 5)

#DEF MAIN

def main ():

print("\t\t\t\t\t\t\t\t SHOP INVENTORY MANAGEMENT")

print()

print("\t\t\t\t -------------------------------------------------------------------------------------------\n")

time.sleep(1)

print("\t\t\t\t\t\t\t\t\t WELCOME")

print()

time.sleep(1)

print("\t\t\t\t\t\t 1. DATABASE SETUP (FOR NEW SYSTEM)")

time.sleep(1)

print()

print("\t\t\t\t\t\t 2. PRODUCT MANAGEMENT")

time.sleep(1)

print()

print("\t\t\t\t\t\t 3. PURCHASE MANAGEMENT")

time.sleep(1)

print()

print("\t\t\t\t\t\t 4. EMPLOYEE MANAGEMENT")

time.sleep(1)

print()

print("\t\t\t\t\t\t 5. PROFIT GRAPH")

time.sleep(1)

print()

print("\t\t\t\t\t\t 6. EXIT\n")

time.sleep(1)

print()

n = input("Enter your choice(1/2/3/4/5/6) :")

time.sleep(1)

if n == '2':

product\_mgmt()

elif n == '3':

purchase\_mgmt()

elif n == '4':

user\_mgmt()

elif n == '5':

X = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]

Y = [10000,12000,135000,7000,14600,13000,8200,6900,11000,9000,18000,15000]

print("Please enter the profit of each month:")

i=0

while i<=12:

profit = input("Please enter the profit:")

i=i+1

Y.append(profit)

plt.plot(x, y)

plt.xlabel('Month Number')

plt.ylabel('Profit in Thousands')

plt.title('Profit Earned in Each Month')

plt.show()

elif n == '1':

db\_mgmt()

elif n == '6':

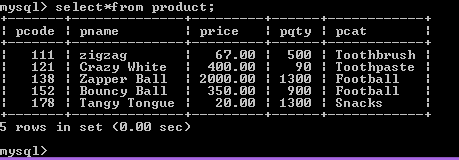
break;

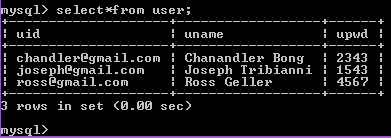
else :

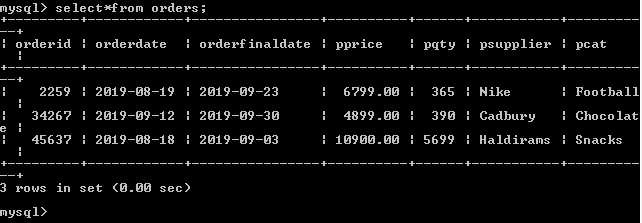
print("\t Incorrect choice")

main()

MySQL Code and Tables







BIBLIOGRAPHY

1. <https://developers.google.com/edu/python>
2. <https://www.edureka.co/blog/sql-joins-types>
3. <https://matplotlib.org/2.0.2/Matplotlib.pdf>
4. <https://jakevdp.github.io/PythonDataScienceHandbook/01.01-help-and-documentation.html>
5. <https://en.wikibooks.org/wiki/Python_Programming/Self_Help>