PROG8420-22W-Sec1-Programming for Big Data

Final Project P3: Travel Agency System

April 20th, 2022

Group Members

Taneja, Chandan (8748475)

Kuan, Sao I (8777987)

Text

Description automatically generatedDiagram

Description automatically generated

Text

Description automatically generated

main.py

import admin\_func a sa

from trip\_dbimport\*

print('main')

if\_\_name\_\_=='\_\_main\_\_':

dbPath=input("Sqlite3Path:")

create\_connection(r"{}".format(dbPath))

print("\*\*\*\*Hello Travel Agency\*\*\*\*")

login()

def login(): #pw:123

pw=input("admin password:")

if pw=="123":

print("login Successful")

print("[1]Location",

"[2]Customer",

"[3]Booking",

"[4]Report",

"[5]Audit")

option(input("action:"))

def option(opt):

if opt=="1":

loc\_profile()

elif opt=="2":

cust\_profile()

elif opt=="3":

booking()

elif opt=="4":

rep\_out()

elif opt=="5":

audit\_out()

else:

print("404")

Text

Description automatically generated

Text

Description automatically generated

admin\_func.py

import sqlite3

import trip\_db as t

def loc\_profile():

print("[1]List Location",

"[2]Enter Location:")

opt=input()

if opt=="1":

print("List Locations")

print(t.rtrv\_data('loc\_profile','Locations'))

elif opt=="2":

conn=sqlite3.connect('SysTA.db')

print("\*\*\*open database success\*\*\*")

loc\_country=input("Enter Country:")

loc\_city=input("Enter City:")

loc\_price=input("Enter Price:")

print("The trip package available period:")

loc\_start\_date=input("Start Date(MM-DD-YYYY):")

loc\_end\_date=input("End Date(MM-DD-YYYY):")

loc\_list=[(loc\_country,loc\_city,loc\_price,loc\_start\_date,loc\_end\_date)]

with conn:

conn=sqlite3.connect('SysTA.db')

c=conn.cursor()

executemany("INSERTIN TO\ loc\_profile(country,city,price,start\_date,end\_date)VALUES(?,?,?,?,?);",loc\_list)

conn.commit()

conn.close()

print("Enter successful")

print(t.rtrv\_data('loc\_profile','Locations'))

else:

print("404")

print("-End-")

trip\_db.py

def rtrv\_data(tbl,list\_name):

conn=sqlite3.connect('SysTA.db')

c=conn.cursor()

print("\*\*\*open data base success\*\*\*")

print(f"List{list\_name}")

cursor=c.execute(f"SELECT \* FROM{tbl};")

colnames=cursor.description

header=[]

for head in col names:

header.append(head[0])

print()

audit\_list=cursor.fetchall()

pd.set\_option('display.max\_columns',None)

df=pd.DataFrame(audit\_list,columns=header)

return df

conn.close()

Text

Description automatically generated

Text

Description automatically generated

admin\_func.py

def cust\_profile():

print("[1]List Customers",

"[2]Create Customer Profile")

opt=input()

if opt=="1":

print("List Customers")

print(t.rtrv\_data('cust\_profile','Customers'))

elif opt=="2":

conn=sqlite3.connect('SysTA.db')

cust\_name=input("Enter Customer Name:")

phone=input("Enter Customer Phone:")

cust\_list=[(cust\_name,phone)]

with conn:

conn=sqlite3.connect('SysTA.db')

c=conn.cursor()

executemany("INSERT INTO cust\_profile(cust\_name,phone)VALUES(?,?);",cust\_list)

conn.commit()

conn.close()

print("Enter successful")

print(t.rtrv\_data('cust\_profile','Customers'))

else:

print("404")

print("-End-")

Text

Description automatically generated

Text

Description automatically generated

admin\_func.py

def booking():#cust-loc-price-dateperiod

print("[1]List Booking Options",

"[2]Booking for customer")

opt=input()

if opt=="1":#[1]List Booking Options

print("List Bookings")

print(t.rtrv\_data('booking','booking'))

elif opt=="2":#[2]Booking for customer

conn=sqlite3.connect('SysTA.db')

fk\_cust\_id=eval(input("Enter Customer ID:"))

fk\_loc\_id=eval(input("Enter Location ID:"))

trvl\_date=input("Enter Travel Date(MM-DD-YYYY):")

paid=bool(eval(input("Customer Paid?('1'if Paid,'0'if not Paid):")))

booking\_list=([trvl\_date,paid,fk\_loc\_id,fk\_cust\_id])

with conn:

conn=sqlite3.connect('SysTA.db')

c=conn.cursor()

execute("INSERT INTO booking(trvl\_date,paid,fk\_loc\_id,fk\_cust\_id)\

VALUES(?,?,?,?)",booking\_list)

conn.commit()

conn.close()

print("booking\_id","cust\_id","trvl\_date","paid")

print(booking\_list)

print(t.rtrv\_data('booking','booking'))

else:

print("404")

print("-End-")

Text

Description automatically generated

admin\_func.py

def rep\_out():

t.orders\_city()

return

trip\_db.py

def orders\_city():#counting how many orders in a city in a dataframe as table

df=audit\_out()

df\_city=df.groupby(['city'])['city'].count().to\_frame('count')

print(df\_city)

plot\_bar=df\_city.plot.bar()

plt.show()#show bar chart as graph

return

def audit\_out():

conn=sqlite3.connect('SysTA.db')

c=conn.cursor()

print("\*\*\*open data base success\*\*\*")

print("List Customer Bookings")

cursor=c.execute("SELECT \* FROM booking\

LEFT JOIN loc\_profile ON loc\_profile.loc\_id=fk\_loc\_id\

LEFT JOIN cust\_profile ON\ cust\_profile.cust\_id=fk\_cust\_id")

colnames=cursor.description

header=[]

for head in colnames:

header.append(head[0])

print()

audit\_list=cursor.fetchall()

pd.set\_option('display.max\_columns',None)

df=pd.DataFrame(audit\_list,columns=header)

filename=datetime.now().strftime("%Y\_%m\_%d-%I\_%M\_%S\_%p")

df.to\_excel(f'{filename}.xlsx',header=True,index=True)

print("Download successful")

conn.close()

return df

Chart, bar chart

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface

Description automatically generated

A picture containing text

Description automatically generated

def audit\_out():#datastorage

print(t.audit\_out())

return

Text

Description automatically generated

Graphical user interface

Description automatically generated