**PYTHON ASSIGNMENT -5**

**Ticket Booking System**

1. Control structure

Task 1: Conditional Statements

In a BookingSystem, you have been given the task is to create a program to book tickets. if available tickets more than no Of Ticket to book then display the remaining tickets or ticket unavailable:

Tasks: 1. Write a program that takes the availableTicket and noOfBookingTicket as input. 2. Use conditional statements (if-else) to determine if the ticket is available or not. 3. Display an appropriate message based on ticket availability.

Task 2: Nested Conditional Statements Create a program that simulates a Ticket booking and calculating cost of tickets. Display tickets options such as "Silver", "Gold", "Dimond". Based on ticket category fix the base ticket price and get the user input for ticket type and no of tickets need and calculate the total cost of tickets booked.

Task 3: Looping From the above task book the tickets for repeatedly until user type "Exit"

**#TASK 1 - (IF-ELSE)**

Available\_ticket = input("Enter available ticket:")

No\_of\_Bookings = input("Enter No.of.Bookings:")

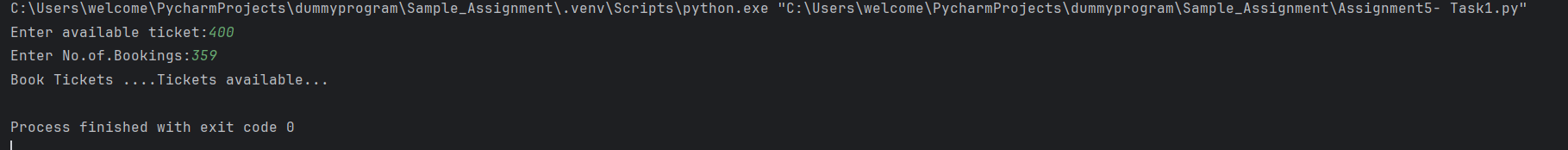
if (Available\_ticket > No\_of\_Bookings):

print("Book Tickets ....Tickets available...")

else:

print("Bookings closed ...AS... No tickets available...")

**Output:**

****

**#TASK2 - (NESTED CONDITIONS)**

while True:

print("Ticket categories: \n Silver \n Gold \n Diamond")

type = input("Enter Input:")

if type.lower() == "silver":

base\_price = 50

elif type.lower() == "gold":

base\_price = 100

elif type.lower() == "diamond":

base\_price = 150

elif type == "Exit":

print("....Exit....")

continue

else:

print("Invalid Entry..\n Try again")

break

try:

num\_of\_tickets = int(input("Enter No.Of Tickets: "))

if num\_of\_tickets <= 0:

print("Invalid no.of tickets")

else:

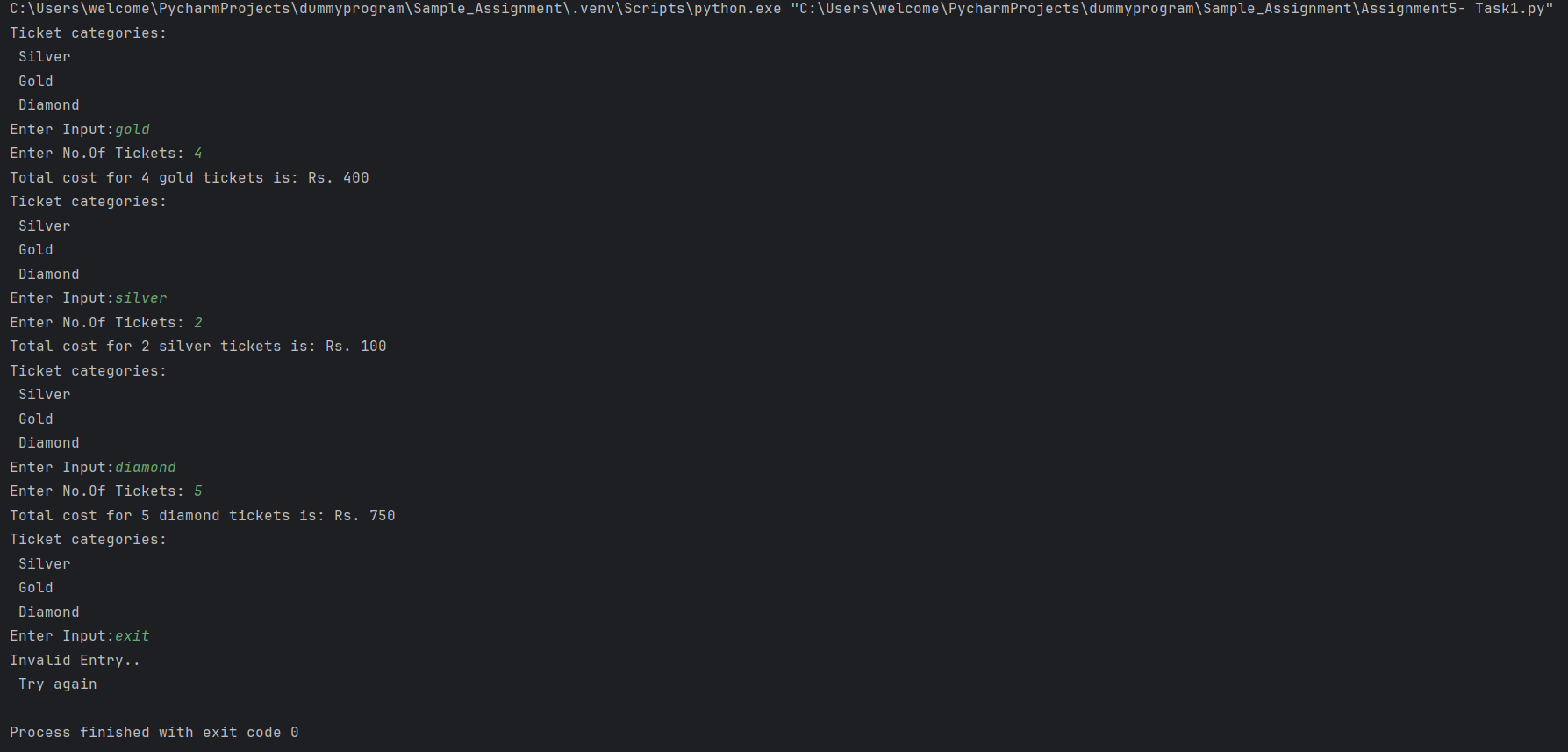
total\_cost = base\_price \* num\_of\_tickets

print(f"Total cost for {num\_of\_tickets} {type} tickets is: Rs. {total\_cost}")

except ValueError:

print("Invalid number...\n Try again")

**Output:**



**Ticket booking System**

Create TicketBookingSystem class and perform following operations: • Create a simple user interface in a main method that allows users to interact with the ticket booking system by entering commands such as "create\_event", "book\_tickets", "cancel\_tickets", "get\_available\_seats,", "get\_event\_details," and "exit”.

**CODE**

# Connecting Database

import mysql.connector as con

password = str(input("Database Password: "))

connection = con.connect(host="localhost", user="root", port="3306", password=password)

# create database

c = connection.cursor()

c.execute("show databases;")

list1 = c.fetchall()

list2 = []

for i in list1:

list2.append(i[0])

if "booktickets2" in list2:

c.execute("use booktickets2;")

else:

c.execute("create database booktickets2;")

c.execute(" use booktickets2;")

c.execute("create table events(event\_name varchar(20),event\_date varchar(20),event\_time varchar(20),total\_seats varchar(100),available\_seats varchar(100),ticket\_price int,event\_type varchar(20));")

c.execute("create table venue (venue\_name varchar(20),address varchar(20));")

c.execute("create table customer (customer\_name varchar(20),Event\_name varchar(20),phone\_number varchar(20),ticket varchar(20),payment varchar(20),Show\_date varchar(20));")

connection.commit()

#Login page

def signin():

print("\n")

print("....Welcome to Ticket Booking System...")

print("\n")

passwrd = input("Enter password to signin: ")

if passwrd == "12345":

print("Signin successful...")

menu()

else:

print("Password Incorrect\n Check your password \n ")

signin()

#creating Events

def create\_event():

event = input("Enter Event Name: ")

date = input("Enter Event Date: ")

time = input("Enter Event Time: ")

total\_seats = input("Enter Total Seats: ")

available\_seats = input("Enter Available Seats: ")

cost = input("Enter Event Price: ")

type = input("Enter Event Type: ")

data = (event, date, time, total\_seats, available\_seats, cost, type)

sql = "insert into events values(%s,%s,%s,%s,%s,%s,%s);"

c = connection.cursor()

c.execute(sql, data)

connection.commit()

print(".......Event Created Successfully.....")

menu()

#book tickets

def book\_tickets():

name = input("Enter Name: ")

event = input("Enter Event Name:")

phno = input("Enter Phone number:")

ticket = int(input(" No.Of. Ticket: "))

payment = int(input("Cost per ticket: "))

date = input("Enter show date and time: ")

data = (name, event, phno, ticket, payment\*ticket,date)

# data1 = (ticket,)

sql = "insert into customer values(%s,%s,%s,%s,%s,%s);"

# sql1 = "select payment from customer where ticket = %s;"

c = connection.cursor()

c.execute(sql, data)

connection.commit()

# connection.commit()

print("Tickets Booked....")

menu()

#cancel tickets

def cancel\_tickets():

name = input("Enter Your Name:")

ticket = input("Enter No.of Tickets:")

data = (name,ticket)

sql = "delete from customer where customer\_name = %s and ticket = %s;"

c = connection.cursor()

c.execute(sql, data)

connection.commit()

print(" Your Tickets has been cancelled.... ")

menu()

#get available seats

def get\_available\_seats():

Event = input("Enter Event Name: ")

data = (Event,)

sql = "select available\_seats from events where event\_name = %s;"

c = connection.cursor()

c.execute(sql, data)

list1 = c.fetchall()

Available\_Seats = list1[0]

for i in list1:

print("Available seats: ",i[0])

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

menu()

#get booking details

def get\_booking\_details():

sql = "select \* from customer;"

c = connection.cursor()

c.execute(sql)

list1 = c.fetchall()

for i in list1:

print("Customer Name: ", i[0])

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

print(" Phone number: ", i[2])

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

print("Payment:", i[4])

print("Event Detail: ", i[5])

print("---------------------\

-----------------------------\

------------------------------\

---------------------")

menu()

#get event details

def get\_event\_details():

date = input("Enter event Date:")

c = connection.cursor()

c.execute("select \* from events;")

list = c.fetchall()

for i in list:

if i[1] == date:

print("Event Name: ", i[0])

#print("Event Date: ", i[1])

print("Event Time: ", i[2])

print("Total seats: ", i[3])

print("Available Seats:", i[4])

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

print("Event Type: ", i[6])

print("---------------------\

-----------------------------\

---------------------")

menu()

#menu()

def menu():

print("\n")

print("Select an option:")

print("1.Create event")

print("2.Book Tickets")

print("3.Cancel Tickets")

print("4.Get Available Seats")

print("5.Get Event Details")

print("6.Get Booking Details")

print("7.Exit")

print("\n")

option = input("Enter option: ")

if option == '1':

create\_event()

elif option == '2':

book\_tickets()

elif option == '3':

cancel\_tickets()

elif option == '4':

get\_available\_seats()

elif option == '5':

get\_event\_details()

elif option == '6':

get\_booking\_details()

elif option == '7':

print("....Exit....")

else:

print("Invalid option...\n Try again...")

menu()

signin()

**OUTPUT:**

