HAMDARD UNIVERSITY

IN PERSUIT OF EXCELLENCE

SEMESTER REPORT

2021

EVENT MANAGEMENT SYSTEM

SUBMITTED BY

S.M.SHAYAN HUSSAIN ZAIDI (946-2018)
WALEED AZHER KIDWAI (943-2018)
SADIQ-UL-AMIN (828-2018)

TEACHER NAME

MR.MOHSIN KHAN

HAMDARD UNIVERSITY

IN PERSUIT OF EXCELLENCE

TABLE OF CONTENT

HAMDARD UNIVERSITY

IN PERSUIT OF EXCELLENCE

ACKNOWLEDGEMENT

We thank **Mr. MOHSIN KHAN** who have been the great inspiration and who have provided sufficient background knowledge and understanding of the subject Programming fundamental.

Our humble prostration goes to our H.O.D. **Dr. FARHA ADEEBA** for providing all the necessary resources and environment, which have aided us to complete this project successfully.

We feel privileged to extend our deep sense of gratitude to our parents for their support and encouragement. Finally yet importantly, we would like to thanks our class friends for their support in completing the project.

HAMDARD UNIVERSITY

IN PERSUIT OF EXCELLENCE

ABSTRACT

This report is an introduction to the Event Management System in Python programming. Anybody, who does not know even the basics of Event Management System in Python, will be certainly able to understand and gain the great knowledge from this report. The core theme of the report focuses on the development of Event Management System in Python Programming language.

Event Management System is a well-prepared system to manage the details of events. It manages all the information about booking of events and of the peoples, booking the tickets for their desire events.

HAMDARD UNIVERSITY

IN PERSUIT OF EXCELLENCE

INTRODUCTION

This is a Python based Event Management project. This system provides various options like book an event, book tickets of the events, view all events, view total tickets that are sell out and their information. The project is developed using two important Pythons concepts that are classes and file handling.

The project is totally built at administrative end and thus only the administrator is guaranteed the access. The purpose of the project is to build an application program to reduce the manual work for managing the booking of events and collect the records of number of peoples coming to those events

HAMDARD UNIVERSITY

IN PERSUIT OF EXCELLENCE

Features of project

❖ CREAT AN EVENT:

❖ VIEW AN EVENT:

❖ BOOK A TICKET:

❖ VIEW TICKETS :

❖ TOTAL SUMMARY OF BOOKED TICKETS:

HAMDARD UNIVERSITY

IN PERSUIT OF EXCELLENCE

PROGRAM FOR EVENT MANAGEMENT SYSTEM

```
# Event Management System


# Features :

# 1. Create An Event

# 2. View Events

# 3. Book Ticket

# 4. View Ticket

# 5. Condition Check If Customer Already Buy Same Event Ticket

# 6. Condition Check if All Tickets are sold Out.

# 7. Show Overall Event Summary


import pickle   #Python pickle module is used for serializing and de-serializing a Python object structure

import os

import pathlib


################################## Book a Ticket Class


class Ticket:

    name = ''

    email = ''
```

```
event = ''
reference = 200000

def bookTicket(self):
    self.name= input("Enter Customer Name: ")
    self.email = input("Enter Customer Email: ")
    file = pathlib.Path("events.data")
    if file.exists():
        infile = open('events.data', 'rb')
        eventdetails = pickle.load(infile)

        self.reference = input("Enter Reference Code(10000 - 50000) : ")
        while True:
            if int(self.reference) <= 10000:
                print("Warning: Please Enter Valid Reference Code")
                self.reference = input("Enter Reference Code(10000 - 50000) : ")
            else:
                break

    for event in eventdetails:
        print("Available Event Code : " + event.eventcode + " Event Name : " + event.eventname)
    infile.close()
    self.event = input("Enter Event Code: ")
```

```
def check(self):
    file = pathlib.Path("tickets.data")
    if file.exists():
        infile = open('tickets.data', 'rb')
        ticketdetails = pickle.load(infile)
        for ticket in ticketdetails:
            if ticket.email == self.email and ticket.event == self.event:
                return True
        infile.close()


def gettotalticketcount(self):
    file = pathlib.Path("events.data")
    if file.exists():
        infile = open('events.data', 'rb')
        eventdetails = pickle.load(infile)
        for event in eventdetails:
            if event.eventcode == self.event:
                return int(event.eventTotalAvaibleSeat)
        infile.close
    else:
        return 0


def getBookedSeatCount(self):
    file = pathlib.Path("tickets.data")
    counter= 0
```

```
        if file.exists():
            infile = open('tickets.data', 'rb')
            ticketdetails = pickle.load(infile)
            for ticket in ticketdetails:
                if ticket.event == self.event:
                    counter = counter + 1
            return int(counter)
        return 0


############################## Create Event Class


class Event:
    eventname = ''
    eventcode = ''
    eventTotalAvaibleSeat = 10


    def createEvent(self):
        self.eventname= input("Enter Event Name: ")
        self.eventcode = input("Enter Event Code: ")
        self.eventTotalAvaibleSeat = input("Enter Event Total Availble Seats: ")
        print("\n\n ------> Event Created!")


################################################ Main Program Modules


# Book Ticket and Check Condition
```

```
def bookEventTicket():
    ticket = Ticket()
    ticket.bookTicket()
    if ticket.check():
        print("Warning : You Already Book A Seat")

    elif ticket.getBookedSeatCount() >= ticket.gettotalticketcount():
        print("Warning : All Ticket Sold Out")

    else:
        print("Sucess : Ticket Booked!")
        saveTicketDetiails(ticket)

# Save Ticket Detials to File

def saveTicketDetiails(ticket):
    file = pathlib.Path("tickets.data")
    if file.exists():
        infile = open('tickets.data', 'rb')
        oldlist = pickle.load(infile)
        oldlist.append(ticket)
        infile.close()
        os.remove('tickets.data')
    else:
        oldlist = [ticket]
```

```
    outfile = open('tempTicket.data', 'wb')
    pickle.dump(oldlist, outfile)
    outfile.close()
    os.rename('tempTicket.data', 'tickets.data')

# Display Saved Ticket Details

def getTicketDetails():
    file = pathlib.Path("tickets.data")
    if file.exists ():
        infile = open('tickets.data','rb')
        ticketdetails = pickle.load(infile)
        print("----------TICKET DETAILS----------")
        print("T-Ref    C-Name    C-Email    E-Code")
        for ticket in ticketdetails :
            print(ticket.reference,"\t",ticket.name,"\t", ticket.email, "\t",ticket.event)
        infile.close()
        print("--------------------------------------------------")
        input('Press Enter To Main Menu')
    else :
        print("NO TICKET RECORDS FOUND")

# Create Event Module

def createEvent():
```

```
    event = Event()
    event.createEvent()
    saveEventDetails(event)

# Save Event Details to File

def saveEventDetails(event):
    file = pathlib.Path("events.data")
    if file.exists():
        infile = open('events.data', 'rb')
        oldlist = pickle.load(infile)
        oldlist.append(event)
        infile.close()
        os.remove('events.data')
    else:
        oldlist = [event]
    outfile = open('tempevents.data', 'wb')
    pickle.dump(oldlist, outfile)
    outfile.close()
    os.rename('tempevents.data', 'events.data')

# Display All Event Details

def getEventsDetails():
    file = pathlib.Path("events.data")
```

```
    if file.exists ():
        infile = open('events.data','rb')
        eventdetails = pickle.load(infile)
        print("---------------EVENT DETAILS----------------------")
        print("E-Name    E-Code    E-Total-Seats")
        for event in eventdetails :
            print(event.eventname,"\t", event.eventcode, "\t",event.eventTotalAvaibleSeat)
        infile.close()
        print("--------------------------------------------------")
        input('Press Enter To Main Menu')
    else :
        print("NO EVENTS RECORDS FOUND")


# Display Reports About Events


def getEventsSummary():
    filetickets = pathlib.Path("tickets.data")
    if filetickets.exists():
        infiletickets = open('tickets.data', 'rb')
        ticketdetails = pickle.load(infiletickets)


    fileEvents = pathlib.Path("events.data")
    if fileEvents.exists ():
        infileEvents = open('events.data','rb')
        eventdetails = pickle.load(infileEvents)
```

```
        print("---------------REPORTS---------------------")
        for event in eventdetails :
            print("\n\nEvent Name : " + event.eventname + " | Total Seats : " + event.eventTotalAvaibleSeat +
" \n")
            for ticket in ticketdetails:
                if event.eventcode == ticket.event:
                    print(ticket.reference, "\t", ticket.name, "\t", ticket.email)


        infileEvents.close()
        infiletickets.close()


        print("-------------------------------------------------")
        input('Press Enter To Main Menu')
    else :
        print("NO EVENTS RECORDS FOUND")


############################################################ Start Program
ch=''
num=0
while ch != 8:
    print("\t\t\t\t----------------------")
    print("\t\t\t\tEVENT MANAGEMENT SYSTEM")
    print("\t\t\t\t----------------------")
    print("\tMAIN MENU")
    print("\t1. BOOK TICKET")
    print("\t2. VIEW TICKET")
```

```
print("\t3. CREATE EVENTS")
print("\t4. VIEW EVENTS")
print("\t5. SHOW SUMMARY")
print("\tSelect Your Option (1-5) ")
ch = input()

if ch == '1':
    bookEventTicket()
elif ch == '2':
    getTicketDetails()
elif ch == '3':
    createEvent()
elif ch == '4':
    getEventsDetails()
elif ch == '5':
    getEventsSummary()
```

HAMDARD UNIVERSITY

IN PERSUIT OF EXCELLENCE

CONCLUSION

At the beginning of this project, we want to make a Management System of an Event and at the end of this project we made this. After preparing the papers, we got the answers of the following questions:

- What is Event Management System?
- Is it possible to run the whole management system into Computer?
- How Events can be classified?
- What are the different types of events that can be arranged?
- What are the main responsibilities of an Event management professional?
- How will you handle large groups of ongoing events?