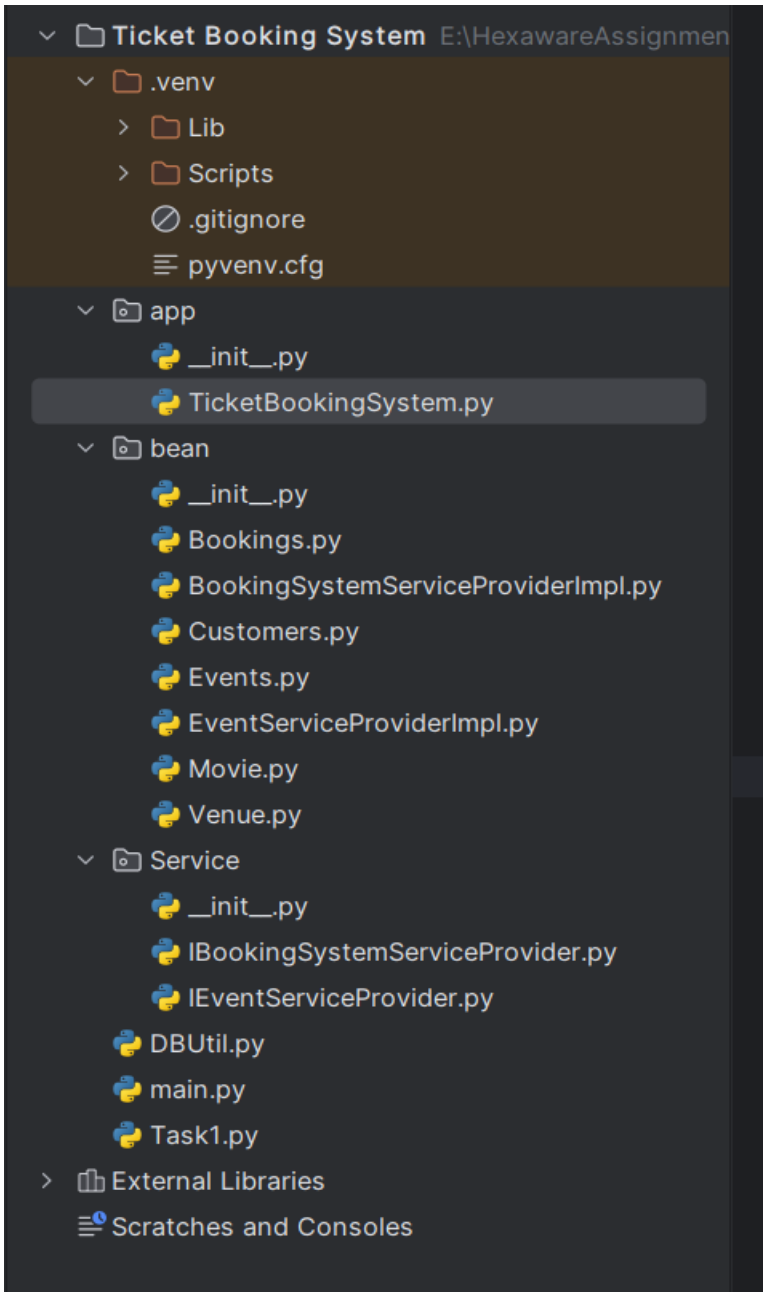
Name: Sarthak Shandilya

Assignment 5 (Ticket Booking System)

Submitted to: Karthika

Hexaware Technologies

Package maintenance according to assignment



This program starts when we run **Ticket Booking System file** in **app package**

This file asks user input to type few commands according to what stated in assignment.

Commands are:

1. create_event
2. get_event_details
3. get_available_seats
4. book_tickets
5. cancel_tickets
6. exit

Ticket Booking System file in **app package**

Program starts by running this file.

```python
from bean.BookingSystemServiceProviderImpl import BookingSystemServiceProviderImpl
from bean.EventServiceProviderImpl import EventServiceProviderImpl
from DBUtil import DBUtil


1 usage
class TicketBookingSystem(EventServiceProviderImpl, BookingSystemServiceProviderImpl):
    def __init__(self, dbUtil):
        super().__init__(dbUtil)

    1 usage
    def main(self):
        while True:
            print("Select one options from the options given below : ")
            print("1. Type create_event to Create a new event.")
            print("2. Type book_tickets to book tickets.")
            print("3. Type cancel_tickets Cancel Tickets.")
            print("4. Type get_available_seats to Know how many seats are Available.")
            print("5. Type get_event_details to see every event and it's details.")
            print("6. Type exit to Exit from the application.")
            choice = input("Enter your choice here : ")
            match choice:
                case "create_event":
                    row = self.create_event()
                    print(row)
                    print()
                case "get_event_details":
                    self.getEventDetails()
                    print()
                case "get_available_seats":
                    self.getAvailableNoOfTickets()
```

```python
                case "get_available_seats":
                    self.getAvailableNoOfTickets()
                    print()
                case "book_tickets":
                    num_tickets = int(input("Please enter the number of tickets you want to book : "))
                    self.book_tickets(num_tickets)
                case "cancel_tickets":
                    booking_id = int(input("Please enter your booking id here : "))
                    self.cancel_booking(booking_id)
                    print()
                case "exit":
                    break
                case _:
                    raise Exception("Unexpected input! Try again")
            print("Thanks for visiting our platform to book tickets. Hope to see you soon.")


    dbutil = DBUtil()
    events = TicketBookingSystem(dbutil)
    events.main()
```

```
C:\Users\hp\AppData\Local\Programs\Python\Python311\python.exe "E:\HexawareAssignments\Python\Ticket Booking System\app\TicketBookingSystem.py"
Select one options from the options given below :
1. Type create_event to Create a new event.
2. Type book_tickets to book tickets.
3. Type cancel_tickets Cancel Tickets.
4. Type get_available_seats to Know how many seats are Available.
5. Type get_event_details to see every event and it's details.
6. Type exit to Exit from the application.
Enter your choice here :
```

1. Create_event

Code:

```
1 usage
def create_event(self):
    event_name = input("Enter event name : ")
    date = input("Enter event Date : ")
    event_date = datetime.datetime.strptime(date, __format: "%Y-%m-%d").date()
    time = input("Enter event time in format HH:MM:SS : ")
    event_time = datetime.datetime.strptime(time, __format: "%H:%M:%S").time()
    venue = input("Enter venue name : ")
    total_seats = int(input("Enter total seats : "))
    available_seats = int(input("Enter available seats : "))
    ticket_price = float(input("Enter ticket price : "))
    event_type = input("Enter event type : ")
    cursor = self.dbUtil.getDBConnection()
    query = "insert into venue (venue_name) values (%s)"
    cursor.execute(query, (venue, ))
    self.dbUtil.con.commit()
    cursor.execute("select venue_id from venue where venue_name=%s", (venue, ))
    venue_id = cursor.fetchone()
    createEvent = ("insert into events (event_name,event_date,event_time,venue_id,total_seats,available_seats,ticket_price,event_type) values (%s,%s,%s,%s,%s,%s,%s,%s)")
    cursor.execute(createEvent, (event_name, event_date, event_time, venue_id[0], total_seats, available_seats, ticket_price, event_type, ))
    self.dbUtil.con.commit()
    cursor.execute("select * from events")
    rows = cursor.fetchall()
    print(rows)
    return rows
```

Output:

```
C:\Users\hp\AppData\Local\Programs\Python\Python311\python.exe "E:\HexawareAssignments\Python\Ticket Booking System\app\TicketBookingSystem.py"
Select one options from the options given below :
1. Type create_event to Create a new event.
2. Type book_tickets to book tickets.
3. Type cancel_tickets Cancel Tickets.
4. Type get_available_seats to Know how many seats are Available.
5. Type get_event_details to see every event and it's details.
6. Type exit to Exit from the application.
Enter your choice here : create_event
Enter event name : IPL semi
Enter event Date : 2024-05-23
Enter event time in format HH:MM:SS : 07:30:00
Enter venue name : Eden Gardens
Enter total seats : 1000
Enter available seats : 900
Enter ticket price : 2000.5
Enter event type : Sports
(101, 'Bollywood Night', datetime.date(2024, 2, 15), datetime.timedelta(seconds=68400), 1, 300, 250, Decimal('1300'), 'Concert')
(102, 'Cricket Championship', datetime.date(2024, 3, 20), datetime.timedelta(seconds=59400), 2, 1000, 800, Decimal('2500'), 'Sports')
(103, 'Tech Summit', datetime.date(2024, 4, 25), datetime.timedelta(seconds=36000), 3, 500, 400, Decimal('2000'), 'Conference')
(104, 'Comedy Show', datetime.date(2024, 5, 10), datetime.timedelta(seconds=73800), 4, 200, 180, Decimal('1500'), 'Comedy')
(105, 'Classical Music Gala', datetime.date(2024, 6, 5), datetime.timedelta(seconds=64800), 5, 400, 350, Decimal('1800'), 'Concert')
(106, 'IPL Final', datetime.date(2024, 6, 15), datetime.timedelta(seconds=70200), 2, 800, 700, Decimal('3000'), 'Sports')
(107, 'IT Conference', datetime.date(2024, 7, 12), datetime.timedelta(seconds=32400), 3, 600, 550, Decimal('2500'), 'Conference')
(108, 'Stand-up Comedy Night', datetime.date(2024, 8, 20), datetime.timedelta(seconds=75600), 4, 250, 230, Decimal('1200'), 'Comedy')
(109, 'Sufi Music Festival', datetime.date(2024, 9, 10), datetime.timedelta(seconds=63000), 5, 350, 300, Decimal('1500'), 'Concert')
(110, 'Football League', datetime.date(2024, 9, 25), datetime.timedelta(seconds=54000), 6, 700, 650, Decimal('2000'), 'Sports')
(111, 'Movie Premiere', datetime.date(2024, 10, 5), datetime.timedelta(seconds=72000), 7, 150, 120, Decimal('1000'), 'Movie')
(112, 'Action Blockbuster', datetime.date(2024, 10, 15), datetime.timedelta(seconds=66600), 8, 180, 150, Decimal('1100'), 'Movie')
```

```
(107, 'IT Conference', datetime.date(2024, 7, 12), datetime.timedelta(seconds=32400), 3, 600, 550, Decimal('2500'), 'Conference')
(108, 'Stand-up Comedy Night', datetime.date(2024, 8, 20), datetime.timedelta(seconds=75600), 4, 250, 230, Decimal('1200'), 'Comedy')
(109, 'Sufi Music Festival', datetime.date(2024, 9, 10), datetime.timedelta(seconds=63000), 5, 350, 300, Decimal('1500'), 'Concert')
(110, 'Football League', datetime.date(2024, 9, 25), datetime.timedelta(seconds=54000), 6, 700, 650, Decimal('2000'), 'Sports')
(111, 'Movie Premiere', datetime.date(2024, 10, 5), datetime.timedelta(seconds=72000), 7, 150, 120, Decimal('1000'), 'Movie')
(112, 'Action Blockbuster', datetime.date(2024, 10, 15), datetime.timedelta(seconds=66600), 8, 180, 150, Decimal('1100'), 'Movie')
(113, 'Romantic Movie Night', datetime.date(2024, 11, 1), datetime.timedelta(seconds=70200), 9, 200, 180, Decimal('1200'), 'Movie')
(114, 'FIFA CUP', datetime.date(2024, 5, 12), datetime.timedelta(seconds=75600), 6, 400, 120, Decimal('15000'), 'Sports')
(119, 'Wedding', datetime.date(2017, 12, 11), datetime.timedelta(seconds=21790), 17, 1000, 1, Decimal('1500'), 'Marraige')
(122, 'IPL semi', datetime.date(2024, 5, 23), datetime.timedelta(seconds=27000), 22, 1000, 900, Decimal('2001'), 'Sports')
Event id for this event is 122
Event created successfully you can see the details above.
```

Updates in database:

```
mysql> select * from events;
+----------+-----------------------+------------+------------+----------+-------------+-----------------+--------------+------------+
| event_id | event_name            | event_date | event_time | venue_id | total_seats | available_seats | ticket_price | event_type |
+----------+-----------------------+------------+------------+----------+-------------+-----------------+--------------+------------+
|      101 | Bollywood Night       | 2024-02-15 | 19:00:00   |        1 |         300 |             250 |         1300 | Concert    |
|      102 | Cricket Championship  | 2024-03-20 | 16:30:00   |        2 |        1000 |             800 |         2500 | Sports     |
|      103 | Tech Summit           | 2024-04-25 | 10:00:00   |        3 |         500 |             400 |         2000 | Conference |
|      104 | Comedy Show           | 2024-05-10 | 20:30:00   |        4 |         200 |             180 |         1500 | Comedy     |
|      105 | Classical Music Gala  | 2024-06-05 | 18:00:00   |        5 |         400 |             350 |         1800 | Concert    |
|      106 | IPL Final             | 2024-06-15 | 19:30:00   |        2 |         800 |             700 |         3000 | Sports     |
|      107 | IT Conference         | 2024-07-12 | 09:00:00   |        3 |         600 |             550 |         2500 | Conference |
|      108 | Stand-up Comedy Night | 2024-08-20 | 21:00:00   |        4 |         250 |             230 |         1200 | Comedy     |
|      109 | Sufi Music Festival   | 2024-09-10 | 17:30:00   |        5 |         350 |             300 |         1500 | Concert    |
|      110 | Football League       | 2024-09-25 | 15:00:00   |        6 |         700 |             650 |         2000 | Sports     |
|      111 | Movie Premiere        | 2024-10-05 | 20:00:00   |        7 |         150 |             120 |         1000 | Movie      |
|      112 | Action Blockbuster    | 2024-10-15 | 18:30:00   |        8 |         180 |             150 |         1100 | Movie      |
|      113 | Romantic Movie Night  | 2024-11-01 | 19:30:00   |        9 |         200 |             180 |         1200 | Movie      |
|      114 | FIFA CUP              | 2024-05-12 | 21:00:00   |        6 |         400 |             120 |        15000 | Sports     |
|      119 | Wedding               | 2017-12-11 | 06:03:10   |       17 |        1000 |               1 |         1500 | Marraige   |
|      122 | IPL semi              | 2024-05-23 | 07:30:00   |       22 |        1000 |             900 |         2001 | Sports     |
+----------+-----------------------+------------+------------+----------+-------------+-----------------+--------------+------------+
16 rows in set (0.00 sec)

mysql>
```

2. Book_tickets

Code:

```
    def book_tickets(self, num_tickets):
        print("Please enter customer details so we can assure booking : ")
        customer_name = input("First please enter your name : ")
        customer_email = input("Please enter you email : ")
        customer_phone = input("Please enter your phone number : ")
        cursor = self.dbUtil.getDBConnection()
        cursor.execute("insert into customer(customer_name,email,phone_number) values (%s,%s,%s)",
                       (customer_name, customer_email, customer_phone,))
        self.dbUtil.con.commit()
        cursor.execute("select customer_id from customer where customer_name=%s", (customer_name,))
        customer_row = cursor.fetchone()
        if customer_row:
            customer_id = customer_row[0]
        print("Please select one events from the events listed below : ")
        cursor.execute("select event_name from events")
        events = cursor.fetchall()
        for event in events:
            print(event[0])
        name_of_event = input("Enter your event here : ")
        cursor.execute("select event_id,ticket_price from events where event_name = %s", (name_of_event,))
        rows = cursor.fetchone()
        if rows:
            event_id, price = rows
        total_cost = price * num_tickets
```

```
        today = date.today()
        query = "insert into bookings (customer_id, event_id, num_tickets, total_cost, booking_date) values (%s,%s,%s,%s,%s)"
        cursor.execute(query, (customer_id, event_id, num_tickets, total_cost, today))
        self.dbUtil.con.commit()
        cursor.execute("select booking_id from bookings where customer_id = %s", (customer_id,))
        booking_id = cursor.fetchone()
        if booking_id:
            b_id = booking_id[0]
        print("Congratulations. Your booking is confirmed. Your booking id is ", b_id)
```

Output:

```
Enter your choice here : book_tickets
Please enter the number of tickets you want to book : 4
Please enter customer details so we can assure booking :
First please enter your name : Sudhanshu
Please enter you email : sudhanshu@yaoho.com
Please enter your phone number : 1346791346
Please select one events from the events listed below :
Bollywood Night
Cricket Championship
Tech Summit
Comedy Show
Classical Music Gala
IPL Final
IT Conference
Stand-up Comedy Night
Sufi Music Festival
Football League
Movie Premiere
Action Blockbuster
Romantic Movie Night
FIFA CUP
Wedding
IPL semi
Enter your event here : Bollywood Night
Congratulations. Your booking is confirmed. Your booking id is  5018
Select one options from the options given below :
```

Updates in database:

```
mysql> select * from bookings;
+------------+-------------+----------+-------------+------------+--------------+
| booking_id | customer_id | event_id | num_tickets | total_cost | booking_date |
+------------+-------------+----------+-------------+------------+--------------+
|       5001 |        1001 |      101 |           2 |       2500 | 2024-01-20   |
|       5002 |        1002 |      102 |           5 |      12500 | 2024-01-21   |
|       5003 |        1003 |      103 |           3 |       6000 | 2024-01-22   |
|       5004 |        1004 |      104 |           4 |       6000 | 2024-01-23   |
|       5005 |        1005 |      105 |           2 |       3600 | 2024-01-24   |
|       5006 |        1006 |      106 |           3 |       9000 | 2024-01-25   |
|       5007 |        1007 |      107 |           2 |       5000 | 2024-01-26   |
|       5008 |        1008 |      108 |           2 |       2400 | 2024-01-27   |
|       5009 |        1009 |      109 |           4 |       6000 | 2024-01-28   |
|       5011 |        1001 |      111 |           3 |       3000 | 2024-02-01   |
|       5014 |        1004 |      106 |           4 |       5200 | 2024-06-15   |
|       5015 |        1005 |      106 |           2 |       2500 | 2024-06-15   |
|       5016 |        1015 |      114 |          10 |     150000 | 2024-02-02   |
|       5018 |        1017 |      101 |           4 |       5200 | 2024-02-04   |
+------------+-------------+----------+-------------+------------+--------------+
14 rows in set (0.00 sec)
```

```
mysql> select * from customer;
+-------------+----------------+-------------------------+--------------+
| customer_id | customer_name  | email                   | phone_number |
+-------------+----------------+-------------------------+--------------+
|        1001 | Priya Sharma   | priya.sharma@email.com  | 9876543210   |
|        1002 | Rahul Verma    | rahul.verma@email.com   | 87654321091  |
|        1003 | Pooja Singh    | pooja.singh@email.com   | 7654321098   |
|        1004 | Aman Gupta     | aman.gupta@email.com    | 65432109879  |
|        1005 | Nisha Patel    | nisha.patel@email.com   | 5432109876   |
|        1006 | Sameer Shah    | sameer.shah@email.com   | 4321098765   |
|        1007 | Anjali Desai   | anjali.desai@email.com  | 3210987654   |
|        1008 | Rohan Malhotra | rohan.malhotra@email.com | 2109876543   |
|        1009 | Shreya Kapoor  | shreya.kapoor@email.com | 1098765432   |
|        1010 | Kartik Joshi   | kartik.joshi@email.com  | 9876543211   |
|        1012 | Aayushi Singh  | saayushi@outlook.com    | 1234567891   |
|        1013 | Aayushi Singh  | saayushi@outlook.com    | 1234554321   |
|        1014 | Aayushi Singh  | saayushi@yahoo.com      | 1298347650   |
|        1015 | Mayank Kumar   | mayank@yahoo.com        | 1234512345   |
|        1016 | Sarthak        | sarthak@gmail.com       | 1234545321   |
|        1017 | Sudhanshu      | sudhanshu@yaoho.com     | 1346791346   |
+-------------+----------------+-------------------------+--------------+
16 rows in set (0.00 sec)
```

3. Cancel_bookings

Code:

```
45
        1 usage
46      def cancel_booking(self, booking_id):
47          cursor = self.dbUtil.getDBConnection()
48          query = "delete from bookings where booking_id = %s"
49          cursor.execute(query, (booking_id,))
50          self.dbUtil.con.commit()
51          print("Your booking is cancelled successfully.")
52
```

Output:

```
Select one options from the options given below :
1. Type create_event to Create a new event.
2. Type book_tickets to book tickets.
3. Type cancel_tickets Cancel Tickets.
4. Type get_available_seats to Know how many seats are Available.
5. Type get_event_details to see every event and it's details.
6. Type exit to Exit from the application.
Enter your choice here : cancel_tickets
Please enter your booking id here : 5018
Your booking is cancelled successfully.
```

Updates in database:

```
mysql> select * from bookings;
+------------+-------------+----------+-------------+------------+--------------+
| booking_id | customer_id | event_id | num_tickets | total_cost | booking_date |
+------------+-------------+----------+-------------+------------+--------------+
|       5001 |        1001 |      101 |           2 |       2500 | 2024-01-20   |
|       5002 |        1002 |      102 |           5 |      12500 | 2024-01-21   |
|       5003 |        1003 |      103 |           3 |       6000 | 2024-01-22   |
|       5004 |        1004 |      104 |           4 |       6000 | 2024-01-23   |
|       5005 |        1005 |      105 |           2 |       3600 | 2024-01-24   |
|       5006 |        1006 |      106 |           3 |       9000 | 2024-01-25   |
|       5007 |        1007 |      107 |           2 |       5000 | 2024-01-26   |
|       5008 |        1008 |      108 |           2 |       2400 | 2024-01-27   |
|       5009 |        1009 |      109 |           4 |       6000 | 2024-01-28   |
|       5011 |        1001 |      111 |           3 |       3000 | 2024-02-01   |
|       5014 |        1004 |      106 |           4 |       5200 | 2024-06-15   |
|       5015 |        1005 |      106 |           2 |       2500 | 2024-06-15   |
|       5016 |        1015 |      114 |          10 |     150000 | 2024-02-02   |
+------------+-------------+----------+-------------+------------+--------------+
13 rows in set (0.00 sec)
```

4. Get_available_seats

Code:

```
1 usage
def getAvailableNoOfTickets(self):
    cursor = self.dbUtil.getDBConnection()
    query = "select event_name from events"
    cursor.execute(query)
    event_names = cursor.fetchall()
    print("Please select one events from below : ")
    for event in event_names:
        print(event)
    take_event = input("Please type your event name here correctly : ")
    query = "select available_seats from events where event_name=%s"
    cursor.execute(query, (take_event,))
    seats = cursor.fetchall()
    print(seats)
```

Output:

```
Enter your choice here : get_available_seats
Please select one events from below :
('Bollywood Night',)
('Cricket Championship',)
('Tech Summit',)
('Comedy Show',)
('Classical Music Gala',)
('IPL Final',)
('IT Conference',)
('Stand-up Comedy Night',)
('Sufi Music Festival',)
('Football League',)
('Movie Premiere',)
('Action Blockbuster',)
('Romantic Movie Night',)
('FIFA CUP',)
('Wedding',)
('IPL semi',)
Please type your event name here correctly : IPL semi
[(900,)]
```

5. Get_event_details

Code:

```
1 usage
def getEventDetails(self):
    cursor = self.dbUtil.getDBConnection()
    cursor.execute("select * from events")
    rows = cursor.fetchall()
    for row in rows:
        print(row)
```

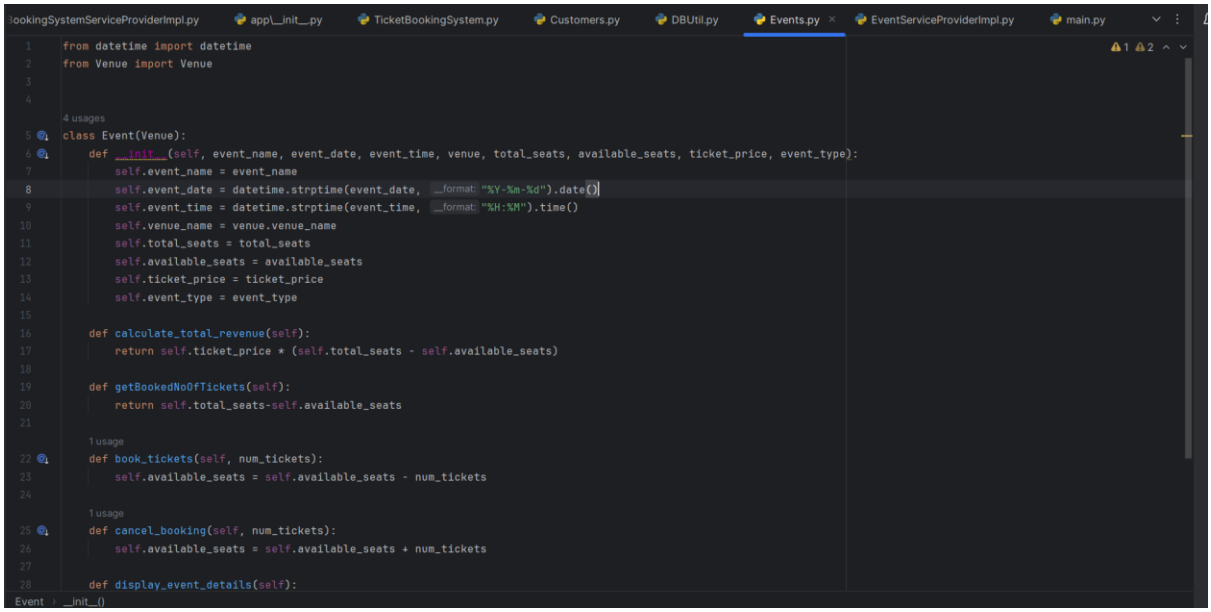
Output:

```
C:\Users\hp\AppData\Local\Programs\Python\Python311\python.exe "E:\HexawareAssignments\Python\Ticket Booking System\app\TicketBookingSystem.py"
Select one options from the options given below :
1. Type create_event to Create a new event.
2. Type book_tickets to book tickets.
3. Type cancel_tickets Cancel Tickets.
4. Type get_available_seats to Know how many seats are Available.
5. Type get_event_details to see every event and it's details.
6. Type exit to Exit from the application.
Enter your choice here : get_event_details
(101, 'Bollywood Night', datetime.date(2024, 2, 15), datetime.timedelta(seconds=68400), 1, 300, 250, Decimal('1300'), 'Concert')
(102, 'Cricket Championship', datetime.date(2024, 3, 20), datetime.timedelta(seconds=59400), 2, 1000, 800, Decimal('2500'), 'Sports')
(103, 'Tech Summit', datetime.date(2024, 4, 25), datetime.timedelta(seconds=36000), 3, 500, 400, Decimal('2000'), 'Conference')
(104, 'Comedy Show', datetime.date(2024, 5, 10), datetime.timedelta(seconds=73800), 4, 200, 180, Decimal('1500'), 'Comedy')
(105, 'Classical Music Gala', datetime.date(2024, 6, 5), datetime.timedelta(seconds=64800), 5, 400, 350, Decimal('1800'), 'Concert')
(106, 'IPL Final', datetime.date(2024, 6, 15), datetime.timedelta(seconds=70200), 2, 800, 700, Decimal('3000'), 'Sports')
(107, 'IT Conference', datetime.date(2024, 7, 12), datetime.timedelta(seconds=32400), 3, 600, 550, Decimal('2500'), 'Conference')
(108, 'Stand-up Comedy Night', datetime.date(2024, 8, 20), datetime.timedelta(seconds=75600), 4, 250, 230, Decimal('1200'), 'Comedy')
(109, 'Sufi Music Festival', datetime.date(2024, 9, 10), datetime.timedelta(seconds=63000), 5, 350, 300, Decimal('1500'), 'Concert')
(110, 'Football League', datetime.date(2024, 9, 25), datetime.timedelta(seconds=54000), 6, 700, 650, Decimal('2000'), 'Sports')
(111, 'Movie Premiere', datetime.date(2024, 10, 5), datetime.timedelta(seconds=72000), 7, 150, 120, Decimal('1000'), 'Movie')
(112, 'Action Blockbuster', datetime.date(2024, 10, 15), datetime.timedelta(seconds=66600), 8, 180, 150, Decimal('1100'), 'Movie')
(113, 'Romantic Movie Night', datetime.date(2024, 11, 1), datetime.timedelta(seconds=70200), 9, 200, 180, Decimal('1200'), 'Movie')
(114, 'FIFA CUP', datetime.date(2024, 5, 12), datetime.timedelta(seconds=75600), 6, 400, 120, Decimal('15000'), 'Sports')
(119, 'Wedding', datetime.date(2017, 12, 11), datetime.timedelta(seconds=21790), 17, 1000, 1, Decimal('1500'), 'Marraige')
(122, 'IPL semi', datetime.date(2024, 5, 23), datetime.timedelta(seconds=27000), 22, 1000, 900, Decimal('2001'), 'Sports')

Select one options from the options given below :
1. Type create_event to Create a new event.
```

Customers class:

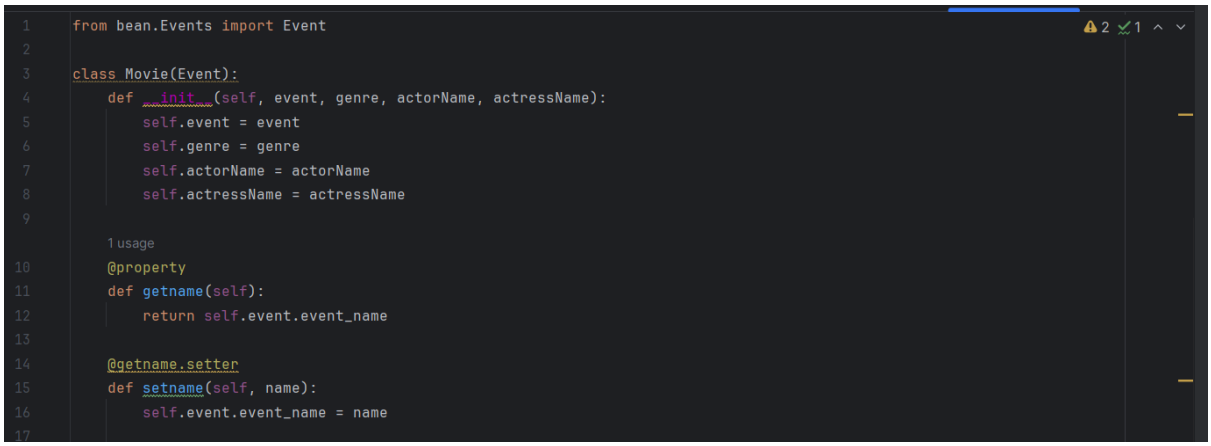
```python
class Customer:
    def __init__(self, customer_name, email, phone):
        self.customer_name = customer_name
        self.email = email
        self.phone = phone

    def display_customer_details(self):
        print(f"Customer Name : {self.customer_name}")
        print(f"Email : {self.email}")
        print(f"Phone Number : {self.phone}")
```

Events class:



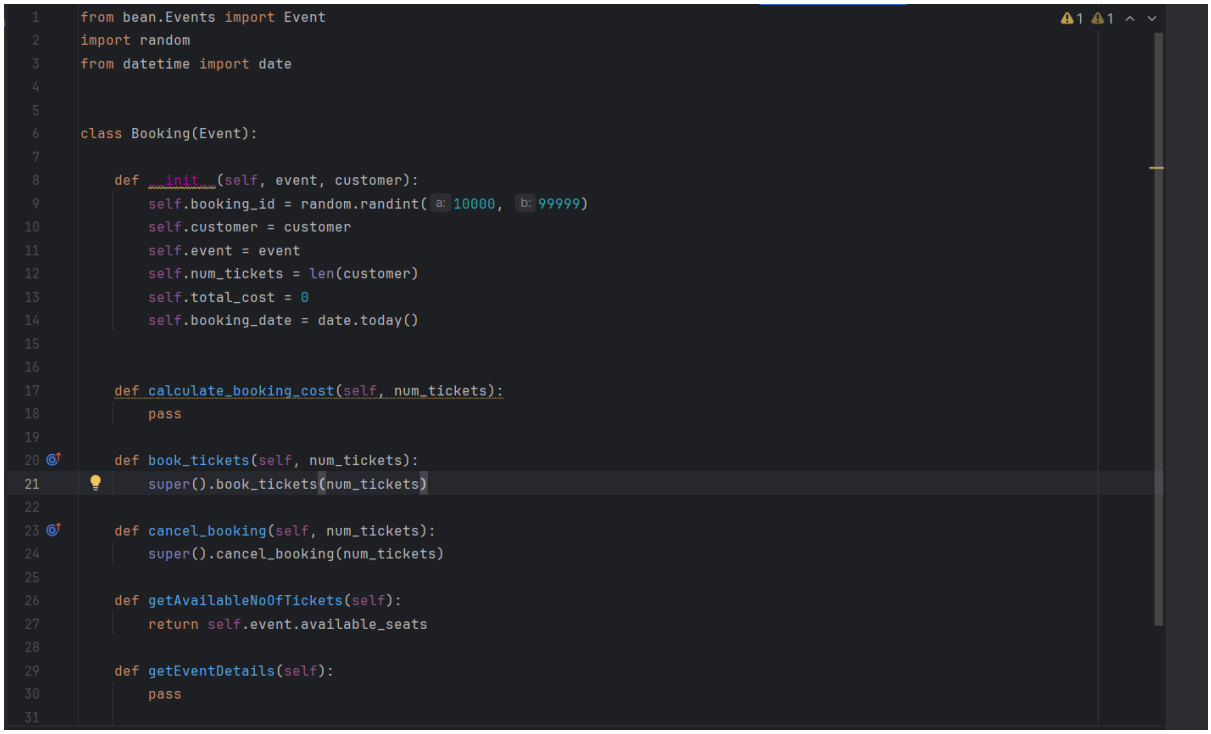

```python
def display_event_details(self):
    print(f"Event name = {self.event_name}")
    print(f"Date of event = {self.event_date}")
    print(f"Time of event = {self.event_time}")
    print(f"Venue name = {self.venue_name}")
    print(f"Available Seats = {self.available_seats}")
```

Movie class:



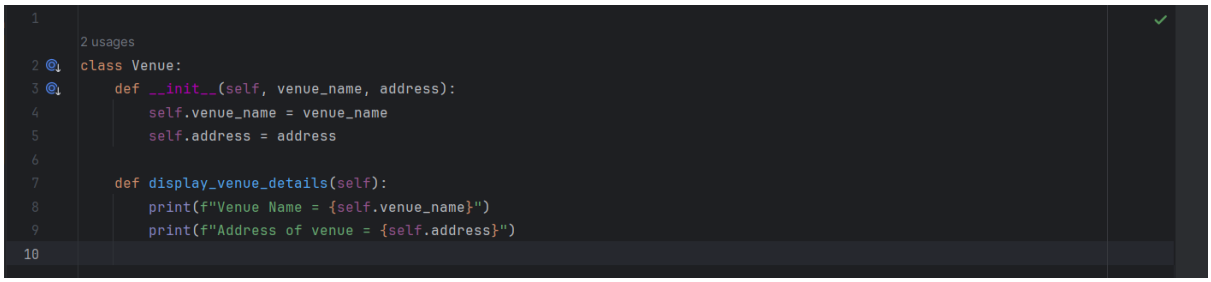
```python
from bean.Events import Event

class Movie(Event):
    def __init__(self, event, genre, actorName, actressName):
        self.event = event
        self.genre = genre
        self.actorName = actorName
        self.actressName = actressName

    1 usage
    @property
    def getname(self):
        return self.event.event_name

    @getname.setter
    def setname(self, name):
        self.event.event_name = name
```

Booking class:



```python
from bean.Events import Event
import random
from datetime import date


class Booking(Event):

    def __init__(self, event, customer):
        self.booking_id = random.randint( a: 10000,  b: 99999)
        self.customer = customer
        self.event = event
        self.num_tickets = len(customer)
        self.total_cost = 0
        self.booking_date = date.today()


    def calculate_booking_cost(self, num_tickets):
        pass

    def book_tickets(self, num_tickets):
        super().book_tickets(num_tickets)

    def cancel_booking(self, num_tickets):
        super().cancel_booking(num_tickets)

    def getAvailableNoOfTickets(self):
        return self.event.available_seats

    def getEventDetails(self):
        pass
```

Venue class:



```
2 usages
class Venue:
    def __init__(self, venue_name, address):
        self.venue_name = venue_name
        self.address = address

    def display_venue_details(self):
        print(f"Venue Name = {self.venue_name}")
        print(f"Address of venue = {self.address}")
```