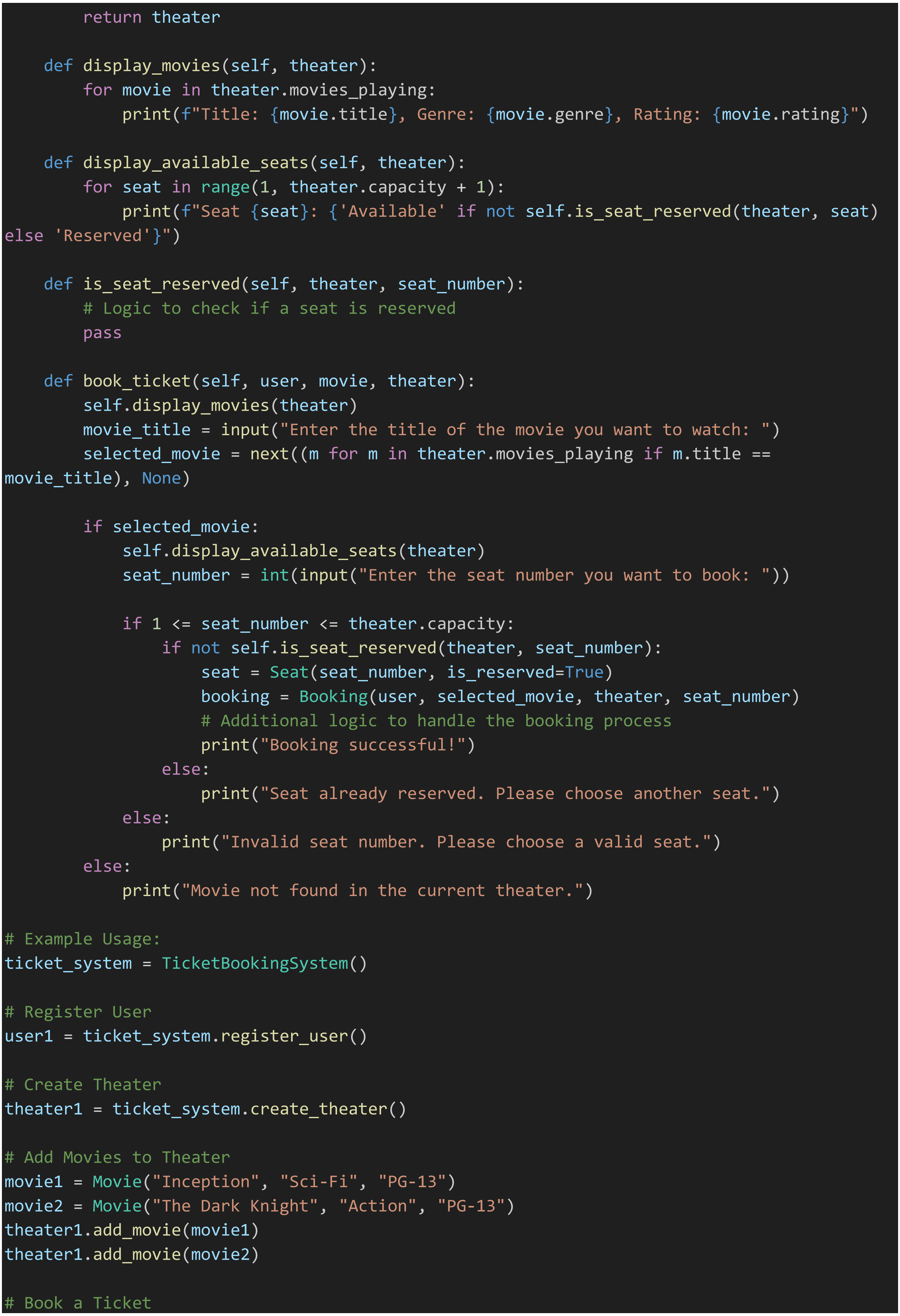
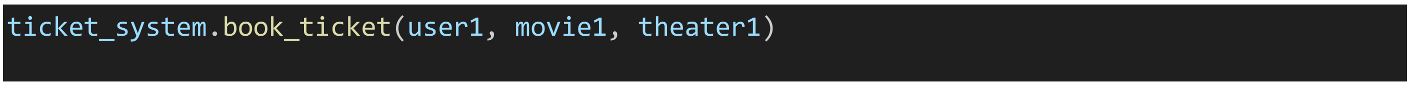
Source code

```
class Movie:
    def __init__(self, title, genre, rating):
        self.title = title
        self.genre = genre
        self.rating = rating

class Theater:
    def __init__(self, name, location, capacity):
        self.name = name
        self.location = location
        self.capacity = capacity
        self.movies_playing = []

    def add_movie(self, movie):
        self.movies_playing.append(movie)

class Seat:
    def __init__(self, seat_number, is_reserved=False):
        self.seat_number = seat_number
        self.is_reserved = is_reserved

class Booking:
    def __init__(self, user, movie, theater, seat_number):
        self.user = user
        self.movie = movie
        self.theater = theater
        self.seat_number = seat_number

class User:
    def __init__(self, username, email):
        self.username = username
        self.email = email

class TicketBookingSystem:
    def __init__(self):
        self.users = []
        self.theaters = []

    def register_user(self):
        username = input("Enter your username: ")
        email = input("Enter your email: ")
        user = User(username, email)
        self.users.append(user)
        return user

    def create_theater(self):
        name = input("Enter theater name: ")
        location = input("Enter theater location: ")
        capacity = int(input("Enter theater capacity: "))
        theater = Theater(name, location, capacity)
        self.theaters.append(theater)
```



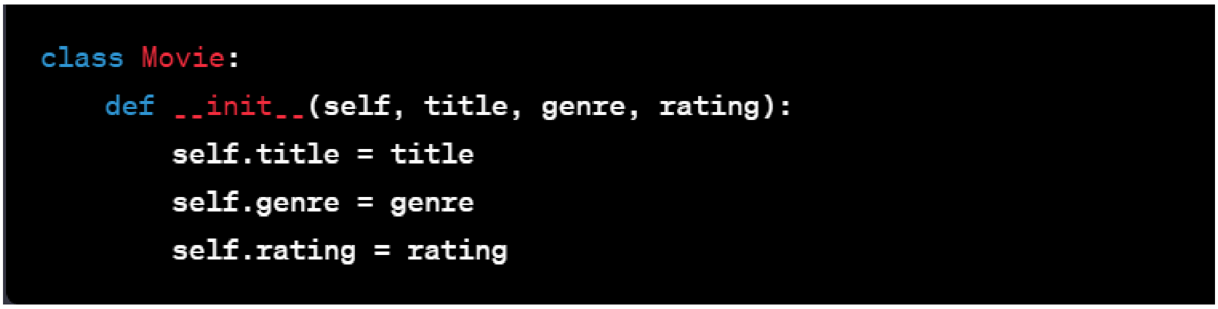
```
        return theater

    def display_movies(self, theater):
        for movie in theater.movies_playing:
            print(f"Title: {movie.title}, Genre: {movie.genre}, Rating: {movie.rating}")

    def display_available_seats(self, theater):
        for seat in range(1, theater.capacity + 1):
            print(f"Seat {seat}: {'Available' if not self.is_seat_reserved(theater, seat)
else 'Reserved'}")

    def is_seat_reserved(self, theater, seat_number):
        # Logic to check if a seat is reserved
        pass

    def book_ticket(self, user, movie, theater):
        self.display_movies(theater)
        movie_title = input("Enter the title of the movie you want to watch: ")
        selected_movie = next((m for m in theater.movies_playing if m.title ==
movie_title), None)

        if selected_movie:
            self.display_available_seats(theater)
            seat_number = int(input("Enter the seat number you want to book: "))

            if 1 <= seat_number <= theater.capacity:
                if not self.is_seat_reserved(theater, seat_number):
                    seat = Seat(seat_number, is_reserved=True)
                    booking = Booking(user, selected_movie, theater, seat_number)
                    # Additional logic to handle the booking process
                    print("Booking successful!")
                else:
                    print("Seat already reserved. Please choose another seat.")
            else:
                print("Invalid seat number. Please choose a valid seat.")
        else:
            print("Movie not found in the current theater.")

# Example Usage:
ticket_system = TicketBookingSystem()

# Register User
user1 = ticket_system.register_user()

# Create Theater
theater1 = ticket_system.create_theater()

# Add Movies to Theater
movie1 = Movie("Inception", "Sci-Fi", "PG-13")
movie2 = Movie("The Dark Knight", "Action", "PG-13")
theater1.add_movie(movie1)
theater1.add_movie(movie2)

# Book a Ticket
```



```
ticket_system.book_ticket(user1, movie1, theater1)
```

code end

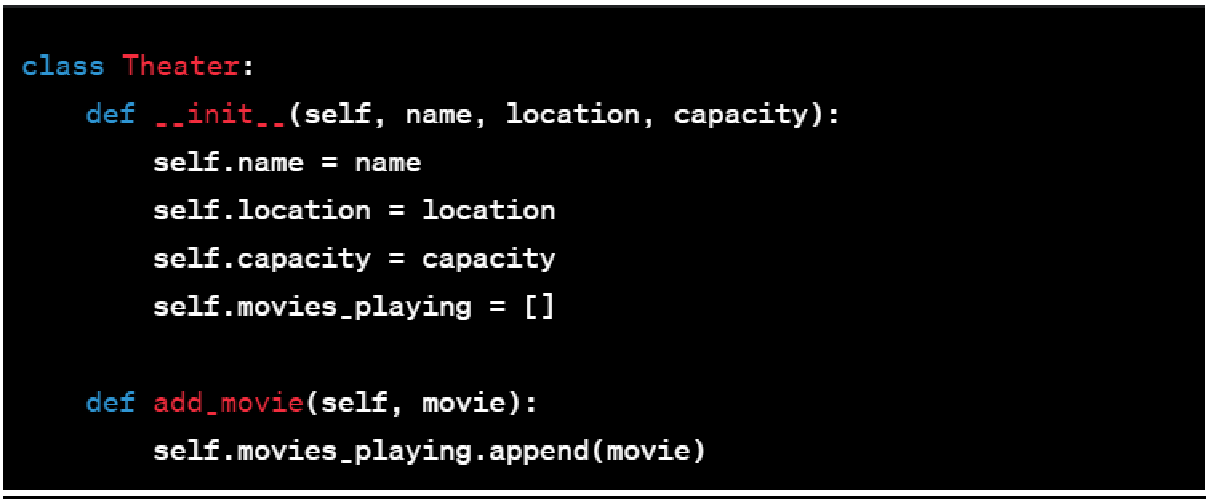
Movie Class



```
class Movie:
    def __init__(self, title, genre, rating):
        self.title = title
        self.genre = genre
        self.rating = rating
```

- This class represents a movie and has attributes for the movie's title, genre, and rating.

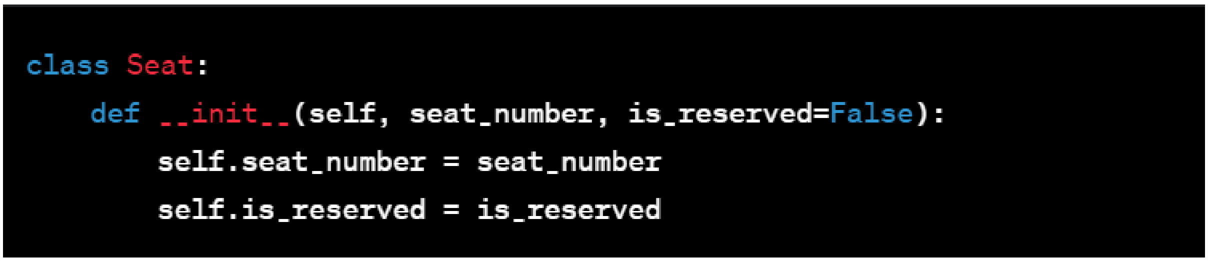
Theatre Class



```
class Theater:
    def __init__(self, name, location, capacity):
        self.name = name
        self.location = location
        self.capacity = capacity
        self.movies_playing = []

    def add_movie(self, movie):
        self.movies_playing.append(movie)
```

- The Theater class represents a movie theater and has attributes like name, location, and capacity.
- The add_movie method allows adding movies to the list of movies currently playing in the theater.

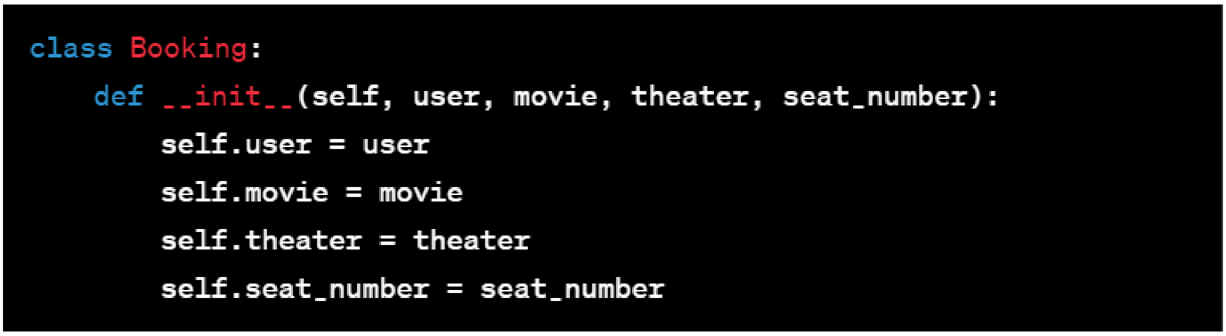
Seat Class



```
class Seat:
    def __init__(self, seat_number, is_reserved=False):
        self.seat_number = seat_number
        self.is_reserved = is_reserved
```

- The Seat class represents a seat in the theater, with attributes for the seat number and whether it is reserved.

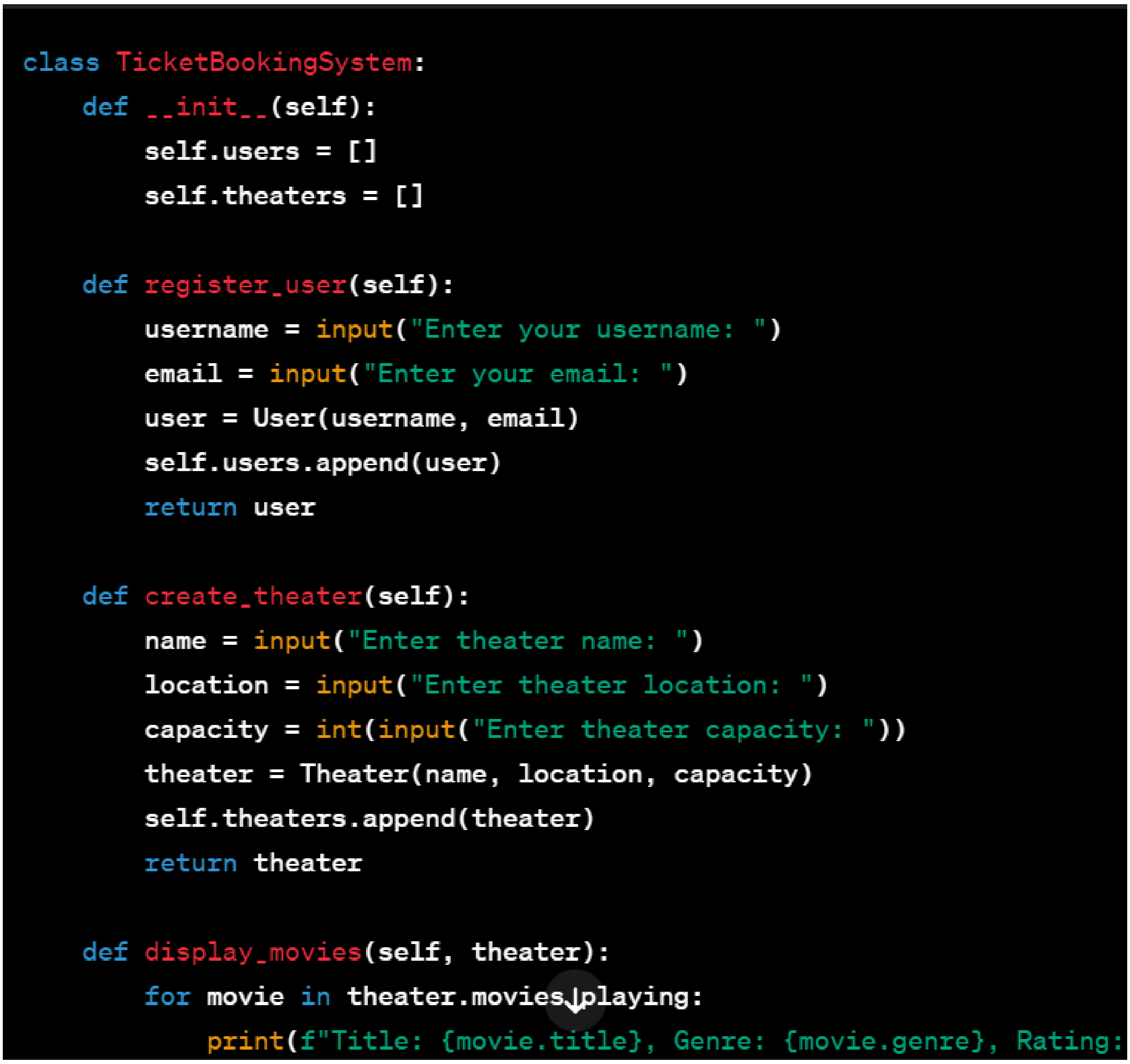
Booking Class



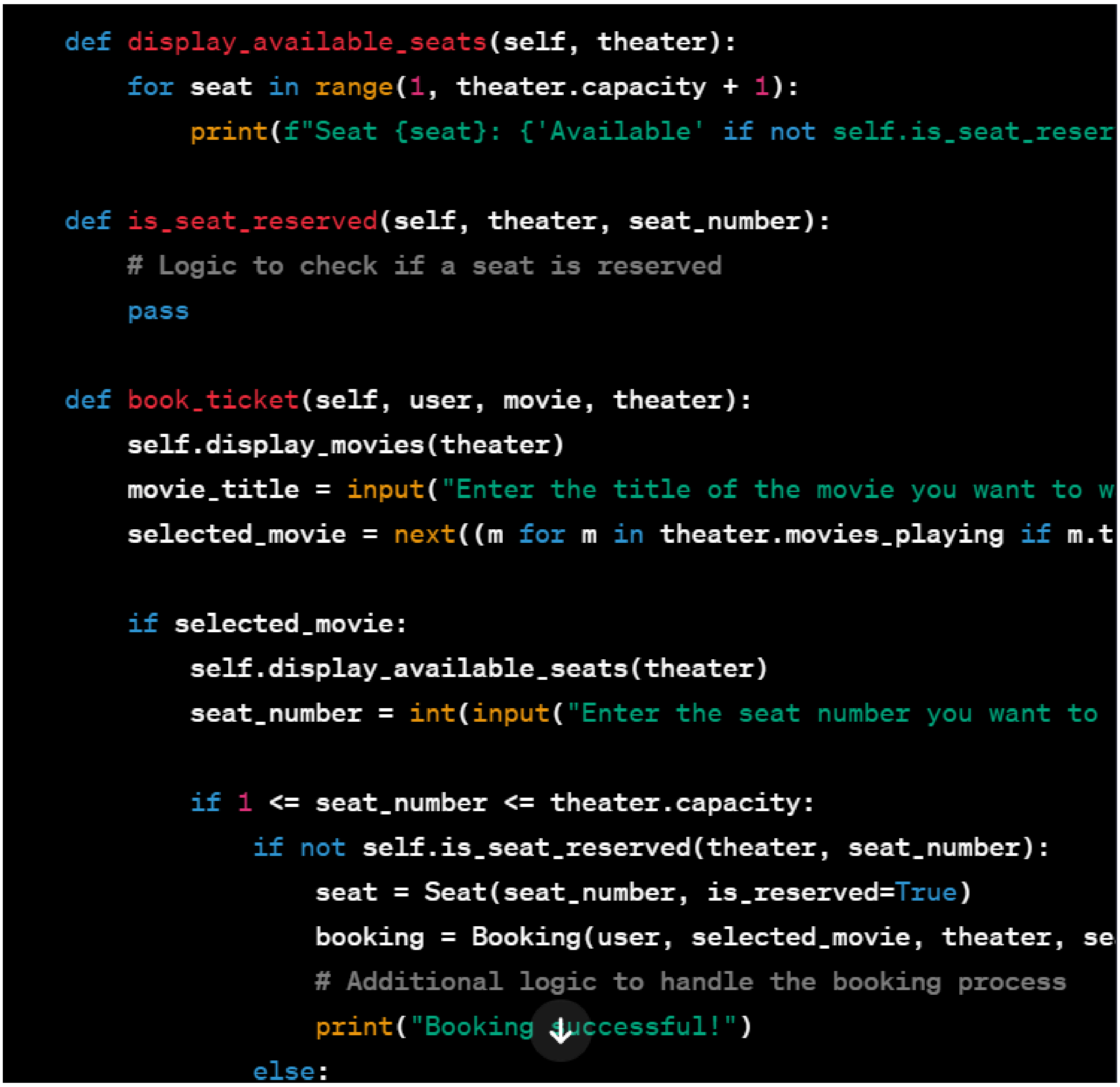
```python
class Booking:
    def __init__(self, user, movie, theater, seat_number):
        self.user = user
        self.movie = movie
        self.theater = theater
        self.seat_number = seat_number
```

- The Booking class represents a booking, connecting a user, a movie, a theater, and a seat number.

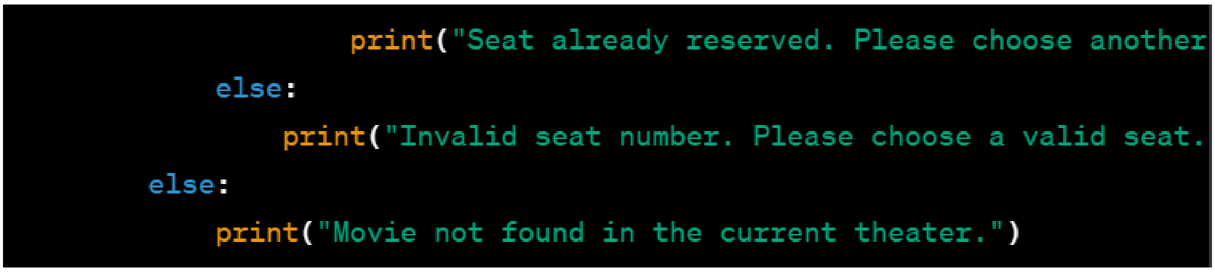
User Class



```python
class TicketBookingSystem:
    def __init__(self):
        self.users = []
        self.theaters = []

    def register_user(self):
        username = input("Enter your username: ")
        email = input("Enter your email: ")
        user = User(username, email)
        self.users.append(user)
        return user

    def create_theater(self):
        name = input("Enter theater name: ")
        location = input("Enter theater location: ")
        capacity = int(input("Enter theater capacity: "))
        theater = Theater(name, location, capacity)
        self.theaters.append(theater)
        return theater

    def display_movies(self, theater):
        for movie in theater.movies_playing:
            print(f"Title: {movie.title}, Genre: {movie.genre}, Rating:
```

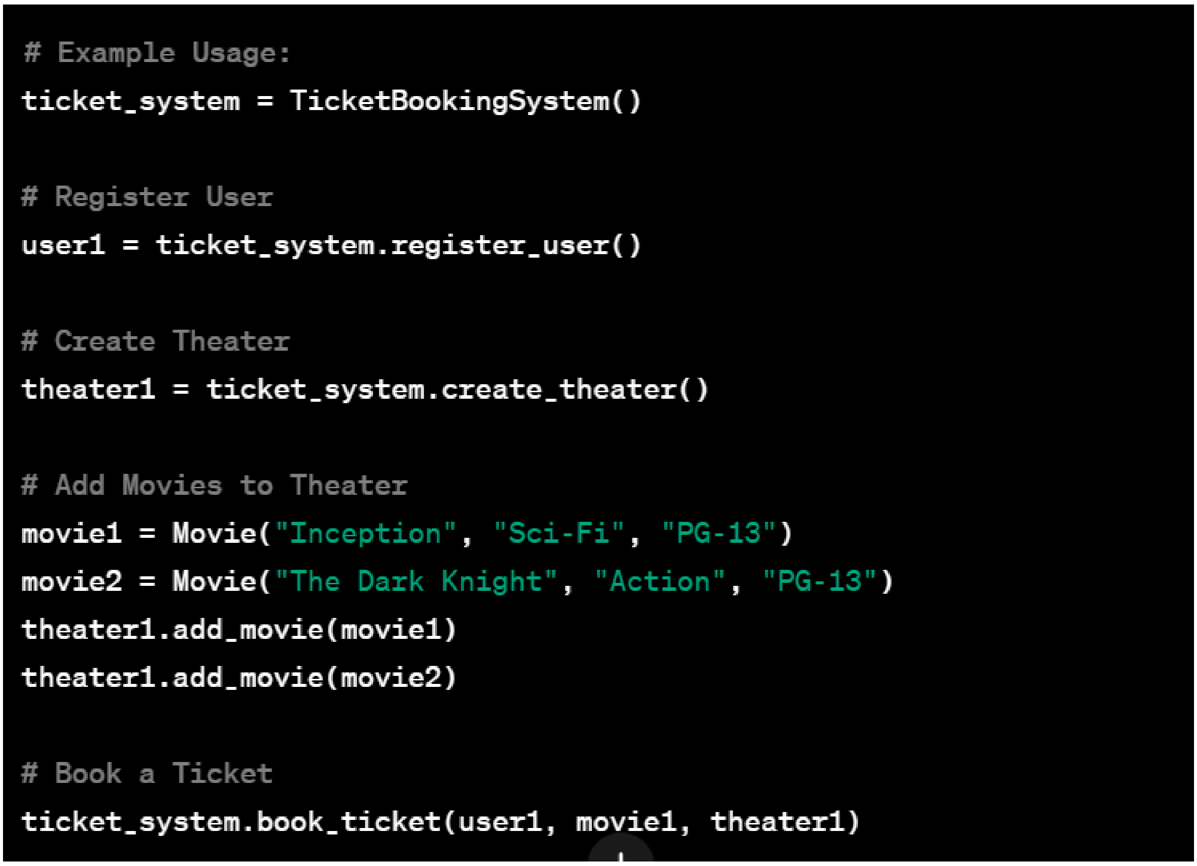


```
    def display_available_seats(self, theater):
        for seat in range(1, theater.capacity + 1):
            print(f"Seat {seat}: {'Available' if not self.is_seat_reser

    def is_seat_reserved(self, theater, seat_number):
        # Logic to check if a seat is reserved
        pass

    def book_ticket(self, user, movie, theater):
        self.display_movies(theater)
        movie_title = input("Enter the title of the movie you want to w
        selected_movie = next((m for m in theater.movies_playing if m.t

        if selected_movie:
            self.display_available_seats(theater)
            seat_number = int(input("Enter the seat number you want to

            if 1 <= seat_number <= theater.capacity:
                if not self.is_seat_reserved(theater, seat_number):
                    seat = Seat(seat_number, is_reserved=True)
                    booking = Booking(user, selected_movie, theater, se
                    # Additional logic to handle the booking process
                    print("Booking successful!")
                else:
```



```
                    print("Seat already reserved. Please choose another
            else:
                print("Invalid seat number. Please choose a valid seat.
        else:
            print("Movie not found in the current theater.")
```

Example Usage



```
# Example Usage:
ticket_system = TicketBookingSystem()

# Register User
user1 = ticket_system.register_user()

# Create Theater
theater1 = ticket_system.create_theater()

# Add Movies to Theater
movie1 = Movie("Inception", "Sci-Fi", "PG-13")
movie2 = Movie("The Dark Knight", "Action", "PG-13")
theater1.add_movie(movie1)
theater1.add_movie(movie2)

# Book a Ticket
ticket_system.book_ticket(user1, movie1, theater1)
```