**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)

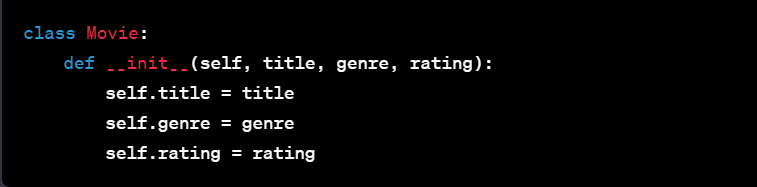
theater1.add\_movie(movie2)

# Book a Ticket

ticket\_system.book\_ticket(user1, movie1, theater1)

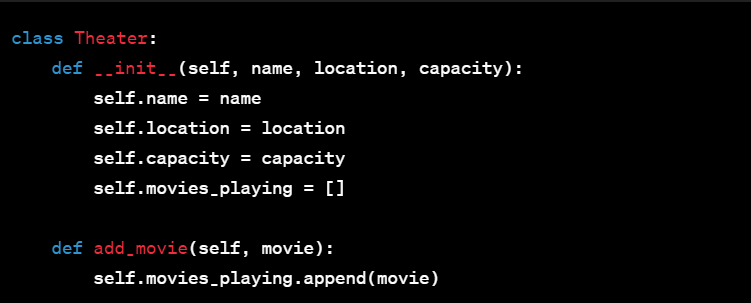
**code end**

**Movie Class**



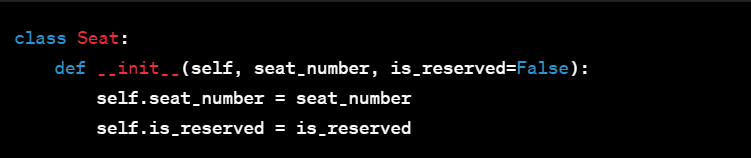
* This class represents a movie and has attributes for the movie's title, genre, and rating.

**Theatre Class**

****

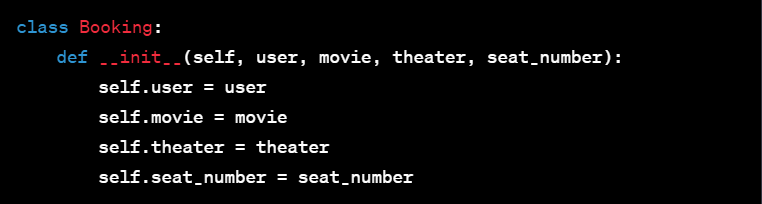
* The Theater class represents a movie theater and has attributes like name, location, and capacity.
* The add\_movie method allows adding movies to the list of movies currently playing in the theater.

**Seat Class**



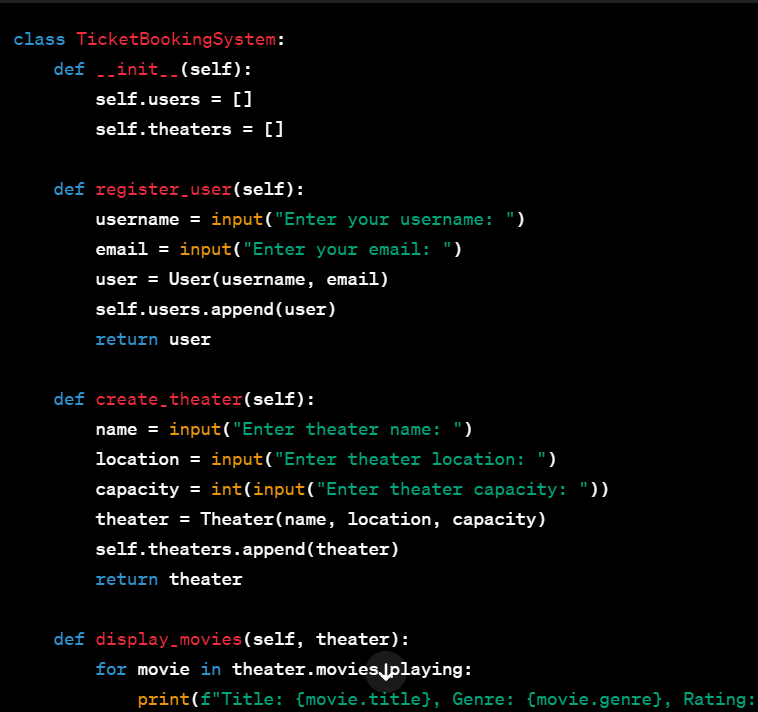
* The Seat class represents a seat in the theater, with attributes for the seat number and whether it is reserved.

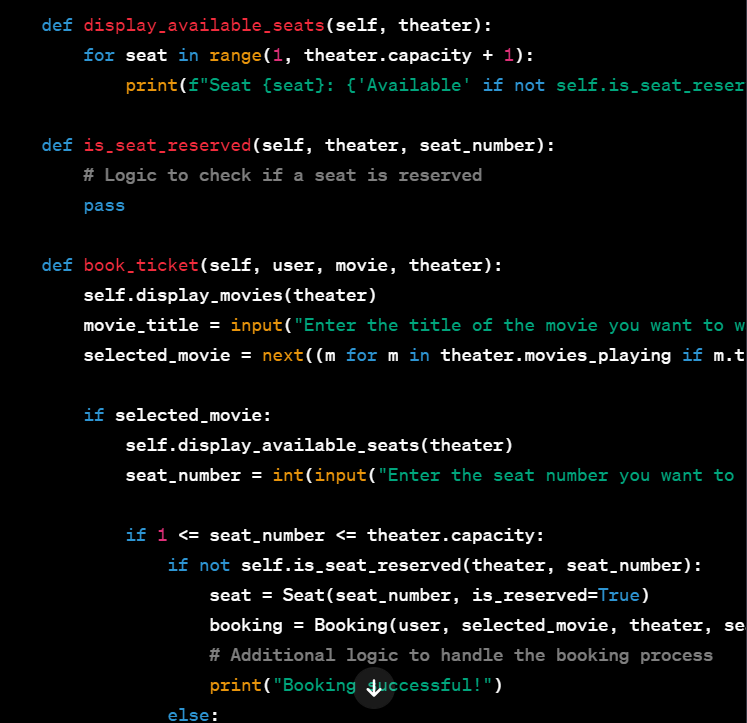
**Booking Class**

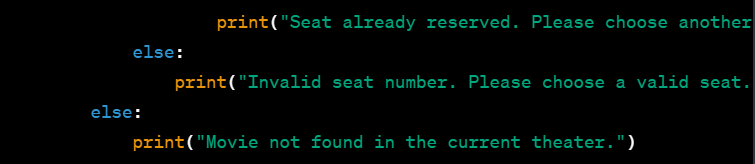


* The Booking class represents a booking, connecting a user, a movie, a theater, and a seat number.

**User Class**







**Example Usage**

