Homework 2: Introduction to Django

1 Objective

You are tasked with building a RESTful Movie Theater Booking Application using Python and Django. The application should allow users to:

- View movie listings
- Book seats
- Check their booking history through a RESTful API

Additionally, you will create an attractive user interface using Django templates and Bootstrap.

2 Time Requirement

Approximately 4-5 hours.

3 Instructions

3.1 Project Setup (30 minutes)

1. Create a new Django project named `movie_theater_booking`.
2. Set up a virtual environment and install dependencies:

```
python3 -m venv myenv
source myenv/bin/activate
pip install django djangorestframework
```

3.2 Creating the Booking App (30 minutes)

Inside your project, create a Django app named `bookings` with the following models:

- **Movie**: title, description, release date, duration.
- **Seat**: seat number, booking status.
- **Booking**: movie, seat, user, booking date.

3.3 Implementing the MVT Architecture (1 hour)

- Create and run migrations.
- Develop RESTful views using Django REST Framework.
- Set up templates for movie listings, seat booking, and booking history.

3.4 Creating an Attractive User Interface (1 hour)

Use Bootstrap to create responsive templates:

```
{% extends 'bookings/base.html' %}
{% block title %}Movie List{% endblock %}
{% block content %}
<h2>Available Movies</h2>
<ul class="list-group">
    {% for movie in movies %}
        <li class="list-group-item">
            <h5>{{ movie.title }}</h5>
            <p>{{ movie.description }}</p>
            <a href="{% url 'book_seat' movie.id %}" class="btn btn-primary">
                Book Now</a>
        </li>
    {% endfor %}
</ul>
{% endblock %}
```

3.5 RESTful API Implementation (1 hour)

Create serializers and define API endpoints:

- `/api/movies/` : List and manage movies.
- `/api/seats/` : Check and book seats.
- `/api/bookings/` : View and create bookings.

3.6 Testing (30 minutes)

Write unit tests for models and test API responses.

3.7 Deployment (30 minutes)

Deploy using Django's development server:

```
python manage.py runserver 0.0.0.0:3000
```

4 Submission Requirements

Submit a zip file containing:

- Project source code with tests.
- README file with setup instructions.
- GitHub repository with frequent commits.

5 Assessment Criteria

- **Functionality (35 points)**: Meets all requirements.
- **User Experience (15 points)**: Clean UI with Bootstrap.
- **Code Quality (15 points)**: Follows Django conventions.
- **Testing (20 points)**: Unit and integration tests included.
- **Deployment (10 points)**: Accessible via DevEdu.
- **Documentation (5 points)**: Clear README included.