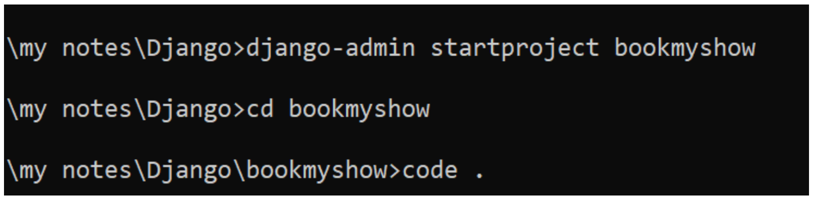
Model :

1. Prerequisite : engine, database name, username, password
    a. Ex : mysql, student, root, root
2. Model represents structure of table in django
3. To represent model, we have create a class in models.py
    a. File models.py is present in every application
4. Every model class that we create is must be subclass of Model class
    a. Model class is present inside models package (from django.db import models)
    b. models package has multiple useful componets like
        i. models.Model -> Model class
        ii. models.IntergerField() -> IntegerFiled() class to provide data type to column of table
5. After creating model class, register it into admin.py
    a. admin.py present in every application

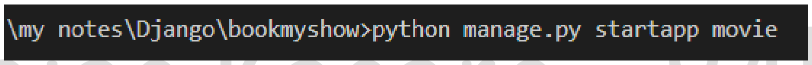
Creating a project

1. create project and open in vs code



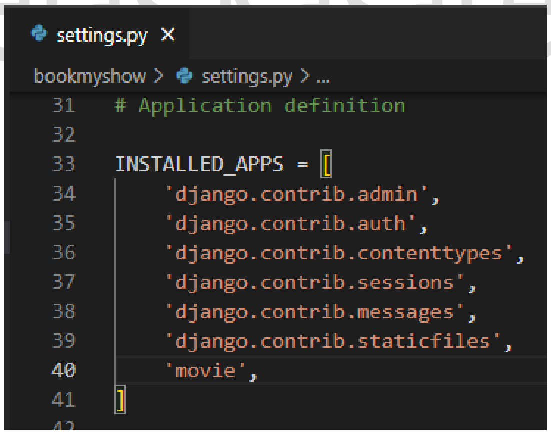
```
\my notes\Django>django-admin startproject bookmyshow

\my notes\Django>cd bookmyshow

\my notes\Django\bookmyshow>code .
```

2. create application : movie



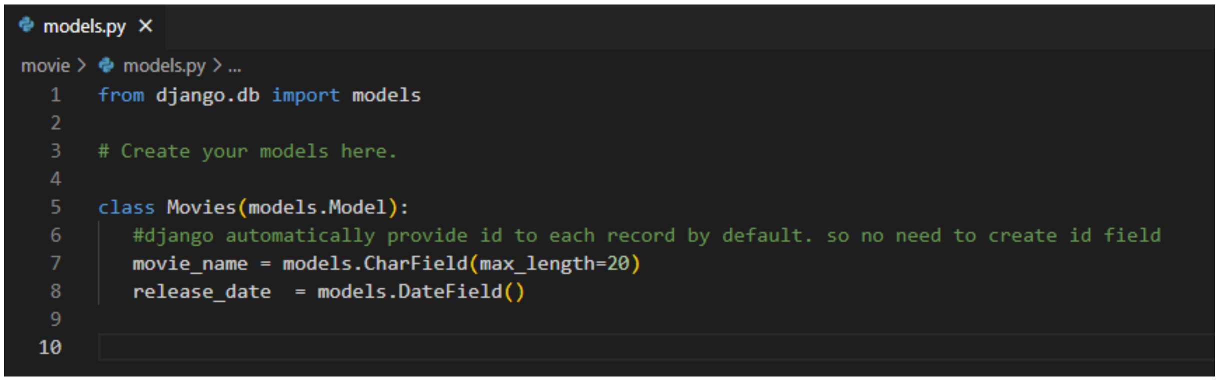
```
\my notes\Django\bookmyshow>python manage.py startapp movie
```

3. register application in settings.py



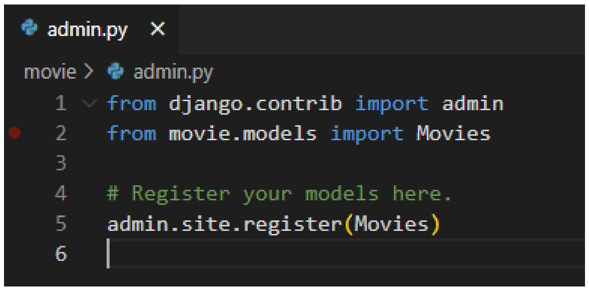
```
# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'movie',
]
```

4. open model.py from movie and create Movies model in it



```
from django.db import models

# Create your models here.

class Movies(models.Model):
    #django automatically provide id to each record by default. so no need to create id field
    movie_name = models.CharField(max_length=20)
    release_date  = models.DateField()

```

5. Register Movie model in admin.py

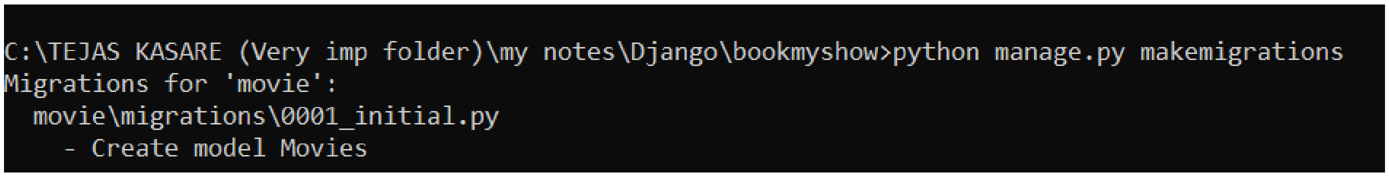


admin.py ×

movie > admin.py

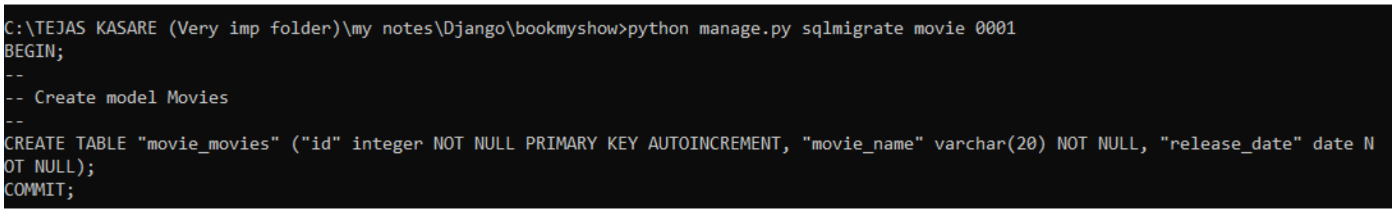
```python
from django.contrib import admin
from movie.models import Movies

# Register your models here.
admin.site.register(Movies)

```

6. Convert model class code into respective sql code
   >python manage.py makemigrations



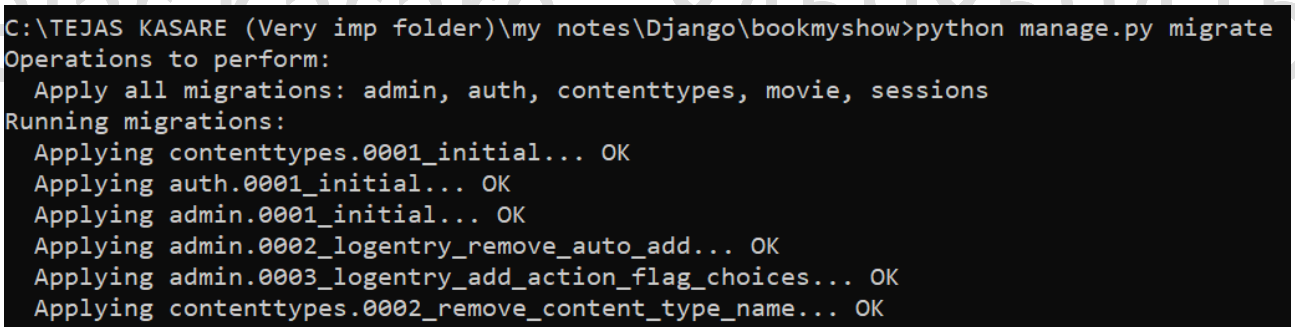
```
C:\TEJAS KASARE (Very imp folder)\my notes\Django\bookmyshow>python manage.py makemigrations
Migrations for 'movie':
  movie\migrations\0001_initial.py
    - Create model Movies
```

7. You can check respective sql code also by :
   >python manage.py sqlmigrate movie 0001



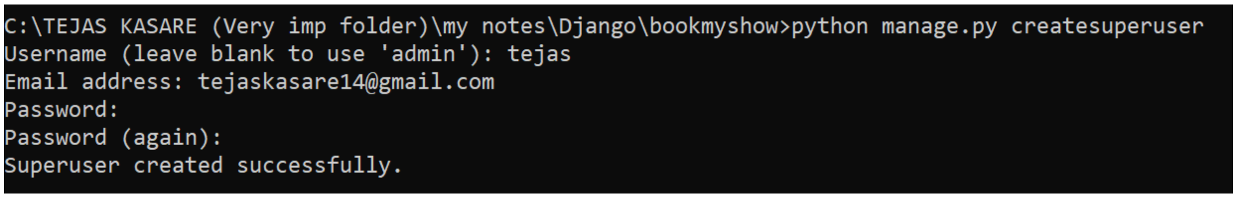
```
C:\TEJAS KASARE (Very imp folder)\my notes\Django\bookmyshow>python manage.py sqlmigrate movie 0001
BEGIN;
--
-- Create model Movies
--
CREATE TABLE "movie_movies" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "movie_name" varchar(20) NOT NULL, "release_date" date N
OT NULL);
COMMIT;
```

8. Executing sql code
   >python manage.py migrate



```
C:\TEJAS KASARE (Very imp folder)\my notes\Django\bookmyshow>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, movie, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
```

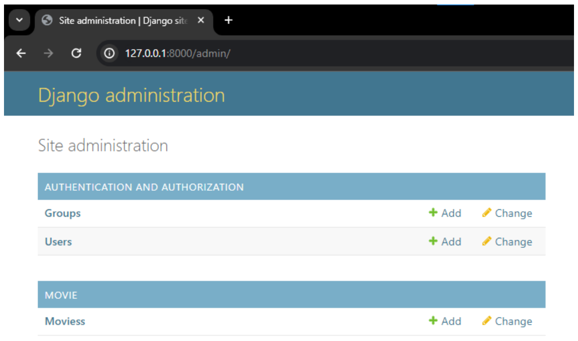
9. Creating super user for admin panel (tejas, django@123)
   >python manage.py createsuperuser



```
C:\TEJAS KASARE (Very imp folder)\my notes\Django\bookmyshow>python manage.py createsuperuser
Username (leave blank to use 'admin'): tejas
Email address: tejaskasare14@gmail.com
Password:
Password (again):
Superuser created successfully.
```

10. Start server

11. Go to admin url : http://127.0.0.1:8000/admin and add above credentials
    a. You will get following page with movie app



Site administration | Django site

127.0.0.1:8000/admin/

Django administration

Site administration

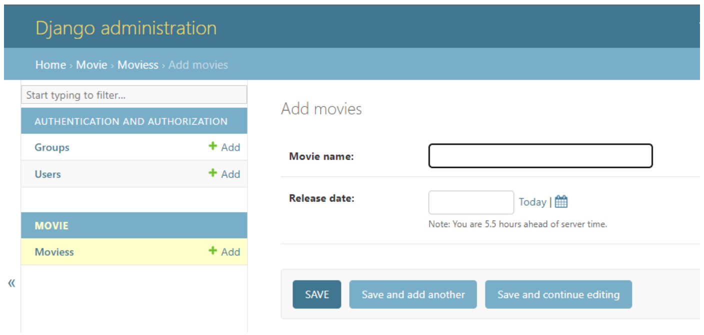
AUTHENTICATION AND AUTHORIZATION

Groups	+ Add	Change
Users	+ Add	Change

MOVIE

Moviess	+ Add	Change

    b. Click on add and add a movie



Django administration

Home › Movie › Moviess › Add movies

Add movies

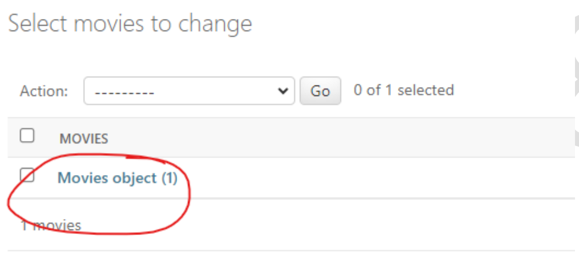
Movie name:

Release date: Today |

Note: You are 5.5 hours ahead of server time.

Start typing to filter...
AUTHENTICATION AND AUTHORIZATION
Groups + Add
Users + Add
MOVIE
Moviess + Add
SAVE
Save and add another
Save and continue editing

    c. After adding movie, you will get movie object, not columns



Select movies to change

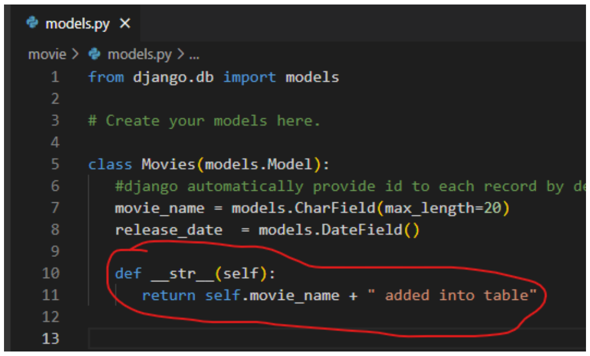
Action: --------- Go 0 of 1 selected

MOVIES

Movies object (1)

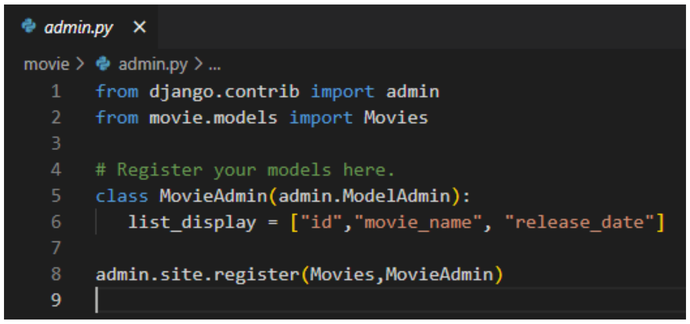
1 movies

    d. Sending response like “xyz movie added to table”. Add following in models.py



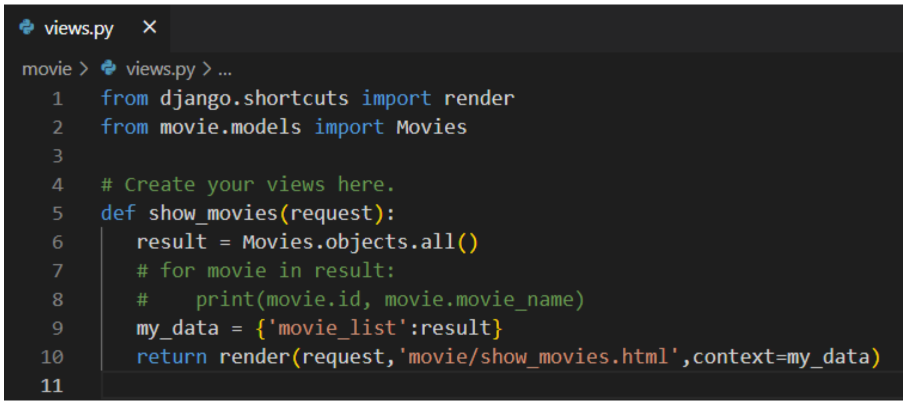
```python
from django.db import models

# Create your models here.

class Movies(models.Model):
    #django automatically provide id to each record by de
    movie_name = models.CharField(max_length=20)
    release_date  = models.DateField()

    def __str__(self):
        return self.movie_name + " added into table"

```

    e. Display column name instead on model object (above step c) in admin panel. Add following in admin.py



```python
from django.contrib import admin
from movie.models import Movies

# Register your models here.
class MovieAdmin(admin.ModelAdmin):
    list_display = ["id","movie_name", "release_date"]

admin.site.register(Movies,MovieAdmin)

```

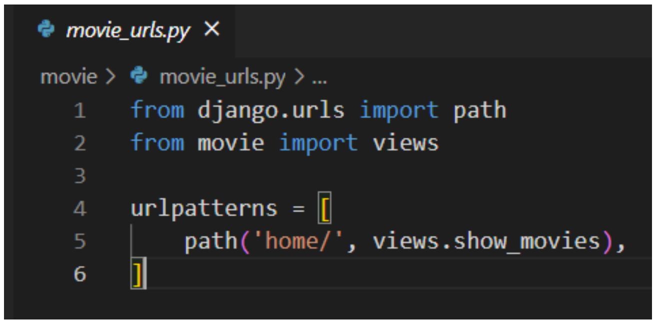
12. Fetch and Displaying data into html
    a. Create show_movie() view in movie -> views.py
        i.



views.py ×
movie > views.py > ...

```python
from django.shortcuts import render
from movie.models import Movies

# Create your views here.
def show_movies(request):
    result = Movies.objects.all()
    # for movie in result:
    #     print(movie.id, movie.movie_name)
    my_data = {'movie_list':result}
    return render(request,'movie/show_movies.html',context=my_data)

```

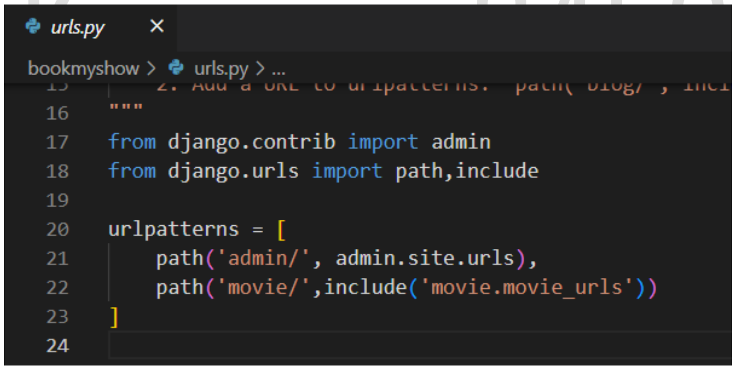
    b. Create application level url for above view
        i. Create movie_urls.py in movie folder



movie_urls.py ×
movie > movie_urls.py > ...

```python
from django.urls import path
from movie import views

urlpatterns = [
    path('home/', views.show_movies),
]
```

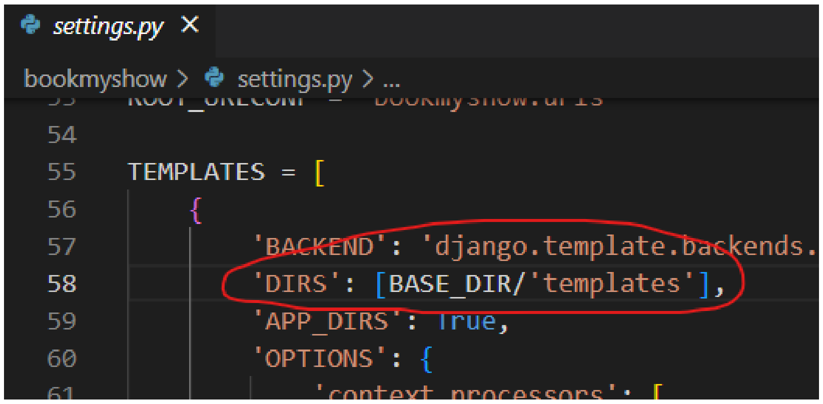
    c. Project level url for above movie application (urls.py)



urls.py ×
bookmyshow > urls.py > ...

```python
    2. Add a URL to urlpatterns:  path('blog/', incl
"""
from django.contrib import admin
from django.urls import path,include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('movie/',include('movie.movie_urls'))
]

```

    d. Create templates folder in bookmyshow folder (project level)
    e. Register templates folder in settings.py

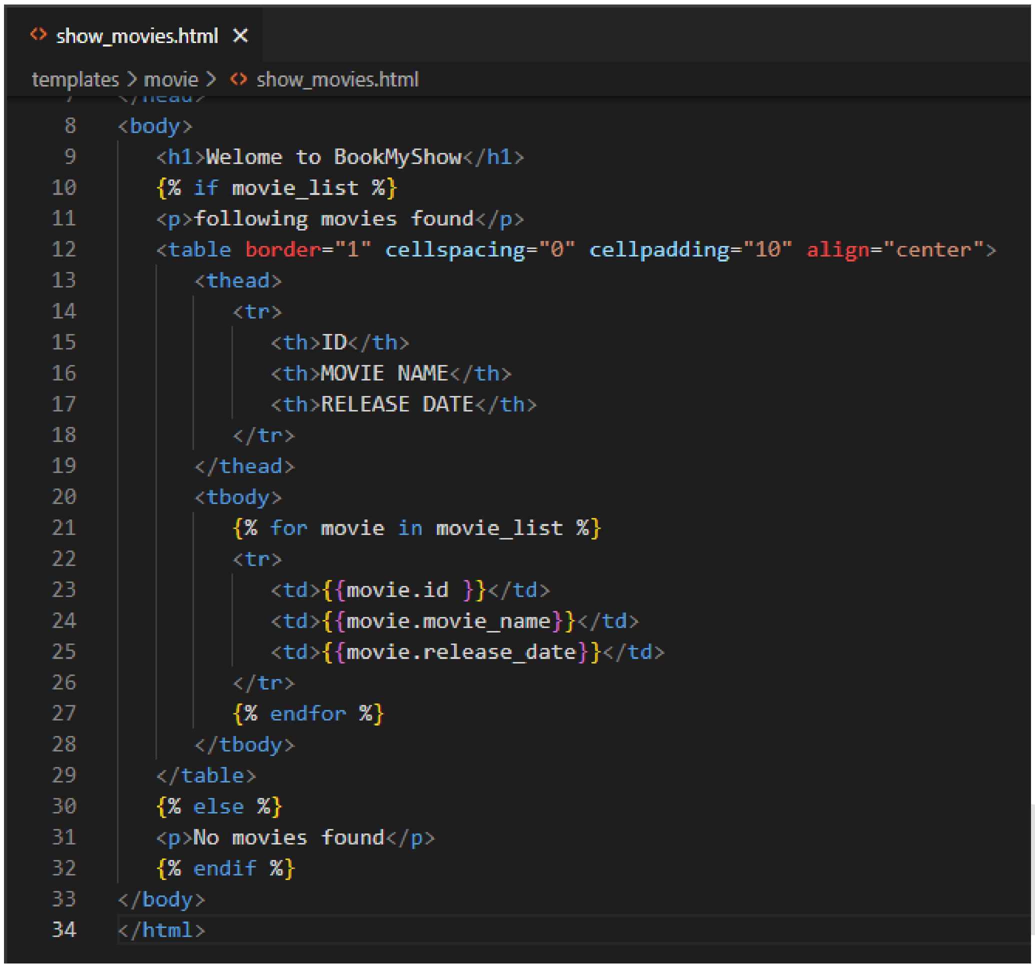


settings.py ×
bookmyshow > settings.py > ...

```python
ROOT_URLCONF = 'bookmyshow.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.
        'DIRS': [BASE_DIR/'templates'],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
```

f. Create movie folder in template folder

   i. Create show_movie.html in templates -> movie -> show_movie.html



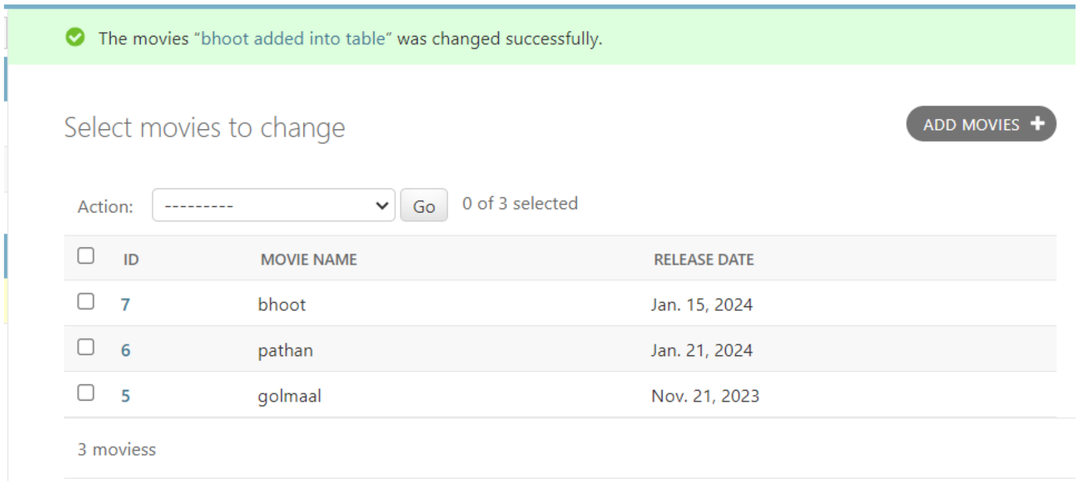
show_movies.html

templates > movie > show_movies.html

```html
<body>
    <h1>Welome to BookMyShow</h1>
    {% if movie_list %}
    <p>following movies found</p>
    <table border="1" cellspacing="0" cellpadding="10" align="center">
        <thead>
            <tr>
                <th>ID</th>
                <th>MOVIE NAME</th>
                <th>RELEASE DATE</th>
            </tr>
        </thead>
        <tbody>
            {% for movie in movie_list %}
            <tr>
                <td>{{movie.id }}</td>
                <td>{{movie.movie_name}}</td>
                <td>{{movie.release_date}}</td>
            </tr>
            {% endfor %}
        </tbody>
    </table>
    {% else %}
    <p>No movies found</p>
    {% endif %}
</body>
</html>
```

g. Runserver and check for output

   i. Admin panel : we are having 3 movies



The movies “bhoot added into table” was changed successfully.

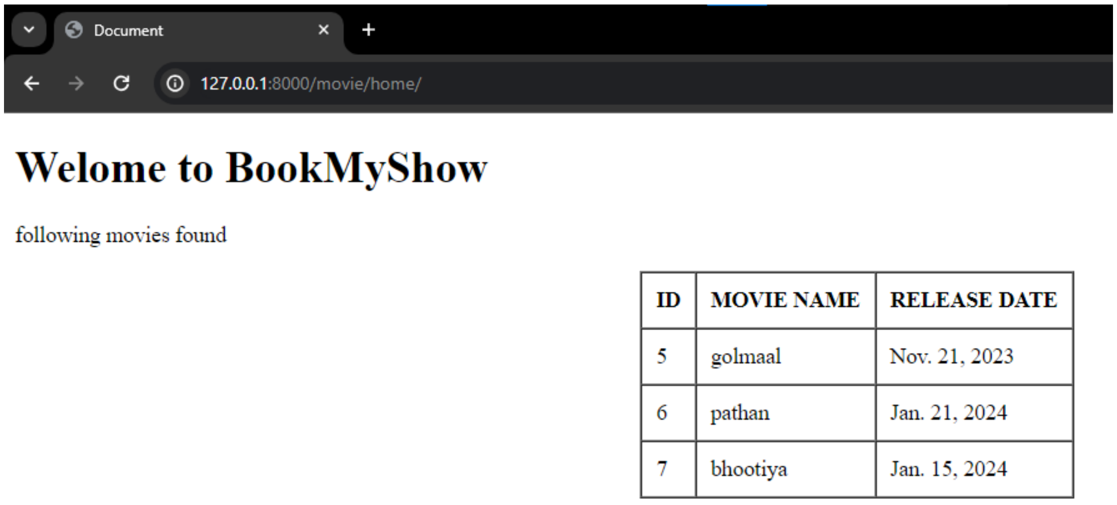
Select movies to change

0 of 3 selected

ID	MOVIE NAME	RELEASE DATE
7	bhoot	Jan. 15, 2024
6	pathan	Jan. 21, 2024
5	golmaal	Nov. 21, 2023

3 moviess

ii. Get all movies in our url : http://127.0.0.1:8000/movie/home/



Welome to BookMyShow

following movies found

ID	MOVIE NAME	RELEASE DATE
5	golmaal	Nov. 21, 2023
6	pathan	Jan. 21, 2024
7	bhootiya	Jan. 15, 2024

DONE !