**Apps.py**

from django.apps import AppConfig

class MovieConfig(AppConfig):

    default\_auto\_field = "django.db.models.BigAutoField"

    name = "movie"

**models.py**

from django.db import models

# Create your models here.

class Movie(models.Model):

    movie\_name = models.CharField(max\_length=50)

    movie\_review = models.TextField(max\_length=50)

    movie\_img = models.FileField(upload\_to="movie/")

**views.py**

from django.shortcuts import render, redirect

from .models import \*

def movie(request):

    if request.method == "POST":

        data = request.POST

        movie\_name = data.get("movie\_name")

        movie\_review = data.get("movie\_review")

        movie\_img = request.FILES.get("movie\_img")

        data\_query = Movie.objects.create(movie\_name=movie\_name, movie\_review=movie\_review, movie\_img=movie\_img)

    query\_set = Movie.objects.all()

    return render(request, "movie.html", context = {"movie":query\_set})

def update\_movie(request, id):

    query\_set = Movie.objects.get(id = id)

    if request.method == "POST":

         data = request.POST

         movie\_name = data.get("movie\_name")

         movie\_review = data.get("movie\_review")

         movie\_img = request.FILES.get("movie\_img")

         query\_set.movie\_name = movie\_name

         query\_set.movie\_review = movie\_review

         if movie\_img:

             query\_set.movie\_img = movie\_img

         query\_set.save()

         return redirect('/movie/')

    context = {'movie': query\_set}

    return render(request, 'update\_movies.html', context)

def delete\_movie(request, id):

    query\_set = Movie.objects.get(id = id)

    query\_set.delete()

    return redirect('/movie/')

**movie.html**

<!doctype html>

<html lang="en">

  <head>

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">

    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script>

    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.14.7/dist/umd/popper.min.js" integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1" crossorigin="anonymous"></script>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/js/bootstrap.min.js" integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous"></script>

    <title>movie</title>

  </head>

  <body>

    <div>

        <form  class = "col-6 mx-auto card lg-shadow mt-5 p-5" method="post" enctype="multipart/form-data" >

            {%  csrf\_token  %}

            <div class="mb-3">

              <label  class="form-label">movie\_name</label>

              <input type="text" class="form-control"   name ="movie\_name" aria-describedby="emailHelp">

              <div id="emailHelp" class="form-text"> share your movie with ease.</div>

            </div>

            <div class="mb-3">

              <label  class="form-label">movie\_review</label>

              <textarea name="movie\_review"> </textarea>

            </div>

            <div class="mb-3 form-check">

              <label class="form-check-label" for="exampleCheck1">movie\_image</label>

              <input type="file" name ="movie\_img"class="form-check-input" id="exampleCheck1">

            </div><br>

            <button type="submit" class="btn btn-primary">Add Movie</button>

          </form>

    </div><br>

    <table class="table">

      <thead>

          <tr>

            <th scope = "column">S.No</th>

            <th scope = "column">Movie\_Name</th>

            <th scope = "column">Movie\_Review</th>

            <th scope = "column">Movie\_Image</th>

          </tr>

      </thead>

      <tbody>

        {% for each\_movie in movie %}

        <tr>

          <th scope = "row">{{forloop.counter}}</th>

          <td >{{each\_movie.movie\_name}}</td>

          <td >{{each\_movie.movie\_review}}</td>

          <td >

            {% if each\_movie.movie\_img %}

              <img src="{{each\_movie.movie\_img.url}}" style="width:100px; height:auto;">

            {% else %}

              No Image

            {% endif %}

          </td>

          <br>

          <td>

            <a href = "/delete-movie/{{each\_movie.id}}/ " class= "btn btn-danger">Delete</a>

            <a href = "/update-movie/{{each\_movie.id}}/ " class= "btn btn-success">Update</a>

          </td>

        </tr>

        {% endfor %}

      </tbody>

    </table>

  </body>

</html>

**Urls.py**

from django.contrib import admin

from django.urls import path,include

from movie import views

from django.conf import settings

from django.conf.urls.static import static

from django.contrib.staticfiles.urls import staticfiles\_urlpatterns

urlpatterns = [

    path("movie/", views.movie , name="movie"),

    path("admin/", admin.site.urls),

    path("delete\_movie/<id>/",views.delete\_movie, name= "delete\_movie" ),

    path("update\_movie/<id>/",views.update\_movie, name= "update\_movie" ),

]

if settings.DEBUG:

    urlpatterns += static(settings.MEDIA\_URL, document\_root = settings.MEDIA\_ROOT)

urlpatterns += staticfiles\_urlpatterns()

**settings.py**

"""

Django settings for project project.

Generated by 'django-admin startproject' using Django 5.1.3.

For more information on this file, see

https://docs.djangoproject.com/en/5.1/topics/settings/

For the full list of settings and their values, see

https://docs.djangoproject.com/en/5.1/ref/settings/

"""

import os

from pathlib import Path

# Build paths inside the project like this: BASE\_DIR / 'subdir'.

BASE\_DIR = Path(\_\_file\_\_).resolve().parent.parent

# Quick-start development settings - unsuitable for production

# See https://docs.djangoproject.com/en/5.1/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!

SECRET\_KEY = "django-insecure-^8(%tw7%nf9uoss$vw(+y99g$2z1%vznh&26i)c\_d!59bs)o+w"

# SECURITY WARNING: don't run with debug turned on in production!

DEBUG = True

ALLOWED\_HOSTS = []

# Application definition

INSTALLED\_APPS = [

    "django.contrib.admin",

    "django.contrib.auth",

    "django.contrib.contenttypes",

    "django.contrib.sessions",

    "django.contrib.messages",

    "django.contrib.staticfiles",

    "movie"

]

MIDDLEWARE = [

    "django.middleware.security.SecurityMiddleware",

    "django.contrib.sessions.middleware.SessionMiddleware",

    "django.middleware.common.CommonMiddleware",

    "django.middleware.csrf.CsrfViewMiddleware",

    "django.contrib.auth.middleware.AuthenticationMiddleware",

    "django.contrib.messages.middleware.MessageMiddleware",

    "django.middleware.clickjacking.XFrameOptionsMiddleware",

]

ROOT\_URLCONF = "project.urls"

TEMPLATES = [

    {

        "BACKEND": "django.template.backends.django.DjangoTemplates",

        "DIRS": [],

        "APP\_DIRS": True,

        "OPTIONS": {

            "context\_processors": [

                "django.template.context\_processors.debug",

                "django.template.context\_processors.request",

                "django.contrib.auth.context\_processors.auth",

                "django.contrib.messages.context\_processors.messages",

            ],

        },

    },

]

WSGI\_APPLICATION = "project.wsgi.application"

# Database

# https://docs.djangoproject.com/en/5.1/ref/settings/#databases

DATABASES = {

    "default": {

        "ENGINE": "django.db.backends.sqlite3",

        "NAME": BASE\_DIR / "db.sqlite3",

    }

}

# Password validation

# https://docs.djangoproject.com/en/5.1/ref/settings/#auth-password-validators

AUTH\_PASSWORD\_VALIDATORS = [

    {

        "NAME": "django.contrib.auth.password\_validation.UserAttributeSimilarityValidator",

    },

    {

        "NAME": "django.contrib.auth.password\_validation.MinimumLengthValidator",

    },

    {

        "NAME": "django.contrib.auth.password\_validation.CommonPasswordValidator",

    },

    {

        "NAME": "django.contrib.auth.password\_validation.NumericPasswordValidator",

    },

]

# Internationalization

# https://docs.djangoproject.com/en/5.1/topics/i18n/

LANGUAGE\_CODE = "en-us"

TIME\_ZONE = "UTC"

USE\_I18N = True

USE\_TZ = True

# Static files (CSS, JavaScript, Images)

# https://docs.djangoproject.com/en/5.1/howto/static-files/

STATIC\_URL = "static/"

STATICFILES\_DIRS = [

    os.path.join(BASE\_DIR, "static")

]

MEDIA\_URL = "media/"

MEDIA\_ROOT = os.path.join(BASE\_DIR, "media")

# Default primary key field type

# https://docs.djangoproject.com/en/5.1/ref/settings/#default-auto-field

DEFAULT\_AUTO\_FIELD = "django.db.models.BigAutoField"

**Asgi.py**

"""

ASGI config for project project.

It exposes the ASGI callable as a module-level variable named ``application``.

For more information on this file, see

https://docs.djangoproject.com/en/5.1/howto/deployment/asgi/

"""

import os

from django.core.asgi import get\_asgi\_application

os.environ.setdefault("DJANGO\_SETTINGS\_MODULE", "project.settings")

application = get\_asgi\_application()

**wsgi.py**

"""

WSGI config for project project.

It exposes the WSGI callable as a module-level variable named ``application``.

For more information on this file, see

https://docs.djangoproject.com/en/5.1/howto/deployment/wsgi/

"""

import os

from django.core.wsgi import get\_wsgi\_application

os.environ.setdefault("DJANGO\_SETTINGS\_MODULE", "project.settings")

application = get\_wsgi\_application()