1.Forms.py

from django import forms  
from studrecord.models import detail  
class detailform(forms.ModelForm):  
 class Meta:  
 model = detail  
 fields= '\_\_all\_\_'

2.create html

<html >  
{% extends 'Html file/base.html' %}  
{% block content %}  
<h1> Add New student Information here</h1>  
<form method ='POST'>  
 {{form.as\_p}}  
 {%csrf\_token %}  
 <input type='submit' value="Add student">  
</form>  
{% endblock %}  
</html>

3.Index html

<html>  
{% extends 'Html file/base.html' %}  
{% block content %}  
<h1>Student details</h1>  
<table border="1" width="850" height="200">  
<tr>  
<th>Name</th>  
<th>Age</th>  
<th>grade</th>  
</tr>  
 {% if stud\_list %}  
 {% for st in stud\_list %}  
  
<tr>  
  
 <td>{{st.Name}}</td>  
 <td>{{st.age}}</td>  
 <td>{{st.grade}}</td>  
 <td><a href="/update/{{st.id}}">update</a> &nbsp; &nbsp; <a href="/delete/{{st.id}}"> Delete</a></td>  
 </tr>  
  
{% endfor %}  
{% endif %}  
</table>

4.Urls.py

from django.contrib import admin  
from django.urls import path  
from studrecord import views  
urlpatterns = [  
 path('admin/', admin.site.urls),  
 path('student/',views.studrecord),  
 path('create/',views.create\_view),  
 path('delete/<id>', views.delete\_view),  
 path('update/<id>', views.update\_view),  
 ]

5.base.html

<html >  
{% load static %}  
<head>  
 <title></title>  
</head>  
<body>  
{% block content %}  
{% endblock %}  
  
</body>  
</html>

6.Views.py

from django.shortcuts import render,redirect  
from studrecord.models import detail  
from studrecord.forms import detailform  
  
def studrecord(request):  
 name="jone"  
 stud\_data=detail.objects.all()  
 stud\_dict={"stud\_list":stud\_data}  
 return render(request,"Html file/index.html",context = stud\_dict)  
*# Create your views here.*def create\_view(request):  
 form=detailform()  
 if request.method =='POST':  
 form=detailform(request.POST)  
 if form.is\_valid():  
 form.save()  
 return redirect('/student')  
  
 return render(request,'Html file/create.html',{'form':form})  
def delete\_view(request ,id):  
 stud\_data = detail.objects.get(id=id)  
 stud\_data.delete()  
 return redirect('/student')  
def update\_view(request,id):  
 stud\_data = detail.objects.get(id=id)  
 if request.method =='POST':  
 form = detailform(request.POST,instance=stud\_data)  
 if form.is\_valid():  
 form.save()  
 return redirect('/student')  
  
  
 return render(request,"Html file/update.html" ,{"student" :stud\_data})

7.models.py

from django.shortcuts import render,redirect  
from studrecord.models import detail  
from studrecord.forms import detailform  
  
def studrecord(request):  
 name="jone"  
 stud\_data=detail.objects.all()  
 stud\_dict={"stud\_list":stud\_data}  
 return render(request,"Html file/index.html",context = stud\_dict)  
*# Create your views here.*def create\_view(request):  
 form=detailform()  
 if request.method =='POST':  
 form=detailform(request.POST)  
 if form.is\_valid():  
 form.save()  
 return redirect('/student')  
  
 return render(request,'Html file/create.html',{'form':form})  
def delete\_view(request ,id):  
 stud\_data = detail.objects.get(id=id)  
 stud\_data.delete()  
 return redirect('/student')  
def update\_view(request,id):  
 stud\_data = detail.objects.get(id=id)  
 if request.method =='POST':  
 form = detailform(request.POST,instance=stud\_data)  
 if form.is\_valid():  
 form.save()  
 return redirect('/student')  
  
  
 return render(request,"Html file/update.html" ,{"student" :stud\_data})

models.py

from django.db import models  
  
class detail(models.Model):  
 Name=models.CharField(max\_length=20)  
 age=models.IntegerField()  
 grade=models.CharField(max\_length=10)

views.py

from django.shortcuts import render,redirect  
from studrecord.models import detail  
from studrecord.forms import detailform  
  
def studrecord(request):  
 name="jone"  
 stud\_data=detail.objects.all()  
 stud\_dict={"stud\_list":stud\_data}  
 return render(request,"Html file/index.html",context = stud\_dict)  
*# Create your views here.*def create\_view(request):  
 form=detailform()  
 if request.method =='POST':  
 form=detailform(request.POST)  
 if form.is\_valid():  
 form.save()  
 return redirect('/student')  
  
 return render(request,'Html file/create.html',{'form':form})  
def delete\_view(request ,id):  
 stud\_data = detail.objects.get(id=id)  
 stud\_data.delete()  
 return redirect('/student')  
def update\_view(request,id):  
 stud\_data = detail.objects.get(id=id)  
 if request.method =='POST':  
 form = detailform(request.POST,instance=stud\_data)  
 if form.is\_valid():  
 form.save()  
 return redirect('/student')  
  
  
 return render(request,"Html file/update.html" ,{"student" :stud\_data})

8.views.py

from django.shortcuts import render,redirect  
from studrecord.models import detail  
from studrecord.forms import detailform  
  
def studrecord(request):  
 name="jone"  
 stud\_data=detail.objects.all()  
 stud\_dict={"stud\_list":stud\_data}  
 return render(request,"Html file/index.html",context = stud\_dict)  
*# Create your views here.*def create\_view(request):  
 form=detailform()  
 if request.method =='POST':  
 form=detailform(request.POST)  
 if form.is\_valid():  
 form.save()  
 return redirect('/student')  
  
 return render(request,'Html file/create.html',{'form':form})  
def delete\_view(request ,id):  
 stud\_data = detail.objects.get(id=id)  
 stud\_data.delete()  
 return redirect('/student')  
def update\_view(request,id):  
 stud\_data = detail.objects.get(id=id)  
 if request.method =='POST':  
 form = detailform(request.POST,instance=stud\_data)  
 if form.is\_valid():  
 form.save()  
 return redirect('/student')  
  
  
 return render(request,"Html file/update.html" ,{"student" :stud\_data})

9.update.html

<html>  
{% extends 'Html file/base.html' %}  
{% block content %}  
<h1> update Student details</h1>  
<form method = 'POST'>  
Name : <input type ="text" name="Name" value ="{{st.Name}}">  
age : <input type ="text" name="age" value ="{{st.age}}">  
grade: <input type ="text" name="grade" value ="{{st.grade}}">  
 {% csrf\_token %}  
 <input type='submit' value="Update student">  
</form>  
{% endblock %}  
</html>

Settings .py

from pathlib import Path  
import os  
*# Build paths inside the project like this: BASE\_DIR / 'subdir'.*BASE\_DIR = Path(\_\_file\_\_).resolve().parent.parent  
BASE\_DIR2=os.path.dirname(os.path.dirname(os.path.abspath(\_\_file\_\_)))  
TEMPLATE\_DIR=os.path.join(BASE\_DIR2,'Template')  
STATIC\_DIR=os.path.join(BASE\_DIR2,'static')  
*# Quick-start development settings - unsuitable for production  
# See https://docs.djangoproject.com/en/4.1/howto/deployment/checklist/  
  
# SECURITY WARNING: keep the secret key used in production secret!*SECRET\_KEY = 'django-insecure-@%4f8i0c3z+9=x1kojnlaq2ar4^5ag4fjp=\_mgygk\_v(+spu3w'  
  
*# 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',  
 'studrecord',  
]  
  
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 = 'class9.urls'  
  
TEMPLATES = [  
 {  
 'BACKEND': 'django.template.backends.django.DjangoTemplates',  
 'DIRS': [TEMPLATE\_DIR],  
 '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 = 'class9.wsgi.application'  
  
  
*# Database  
# https://docs.djangoproject.com/en/4.1/ref/settings/#databases*DATABASES={  
 'default':{  
 'ENGINE':'django.db.backends.postgresql\_psycopg2',  
 'NAME':'database2',  
 'USER':'udaya',  
 'PASSWORD':'udaya',  
 'HOST':'localhost',  
 }  
}  
  
  
*# Password validation  
# https://docs.djangoproject.com/en/4.1/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/4.1/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/4.1/howto/static-files/*STATIC\_URL = 'static/'  
STATICFILE\_DIRS=['STATIC\_DIR']  
  
*# Default primary key field type  
# https://docs.djangoproject.com/en/4.1/ref/settings/#default-auto-field*DEFAULT\_AUTO\_FIELD = 'django.db.models.BigAutoField'

10.Delete.html

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title></title>  
</head>  
<body>  
  
</body>  
</html>