**Project:** crudexample

**App:** employee

**Crudexample:**

**Crudexample/Settings.py:**

"""

Django settings for crudexample project.

Generated by 'django-admin startproject' using Django 4.2.6.

For more information on this file, see

https://docs.djangoproject.com/en/4.2/topics/settings/

For the full list of settings and their values, see

https://docs.djangoproject.com/en/4.2/ref/settings/

"""

from pathlib import Path

import os

# 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/4.2/howto/deployment/checklist/

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

SECRET\_KEY = 'django-insecure-6@in)(r@jz$sj%y7mzs9(sp6c\*i&u3nm)&%n\_&e4^i#(vus#!6'

# 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',

    'employee',

]

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 = 'crudexample.urls'

TEMPLATES = [

    {

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

        'DIRS': [os.path.join(BASE\_DIR, 'templates')],

        'APP\_DIRS': True,

        'OPTIONS': {

            'context\_processors': [

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

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

                'django.contrib.auth.context\_processors.auth',  # Add this line

                'django.contrib.messages.context\_processors.messages',  # Add this line

            ],

        },

    },

]

# crudexample/settings.py

WSGI\_APPLICATION = 'crudexample.wsgi.application'

# Database

# https://docs.djangoproject.com/en/4.2/ref/settings/#databases

DATABASES = {

    'default': {

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

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

    }

}

# Password validation

# https://docs.djangoproject.com/en/4.2/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/4.2/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/4.2/howto/static-files/

STATIC\_URL = '/static/'

# Default primary key field type

# https://docs.djangoproject.com/en/4.2/ref/settings/#default-auto-field

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

**Crudexample/urls.py:**

"""

URL configuration for crudexample project.

The `urlpatterns` list routes URLs to views. For more information please see:

    https://docs.djangoproject.com/en/4.2/topics/http/urls/

Examples:

Function views

    1. Add an import:  from my\_app import views

    2. Add a URL to urlpatterns:  path('', views.home, name='home')

Class-based views

    1. Add an import:  from other\_app.views import Home

    2. Add a URL to urlpatterns:  path('', Home.as\_view(), name='home')

Including another URLconf

    1. Import the include() function: from django.urls import include, path

    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))

"""

from django.contrib import admin

from django.urls import path

from employee import views

urlpatterns = [

    path('admin/', admin.site.urls),

    path('emp', views.emp),

    path('show',views.show),

    path('edit/<int:id>', views.edit),

    path('update/<int:id>', views.update),

    path('delete/<int:id>', views.destroy),

]

**employee:**

**employee/admin.py:**

from django.contrib import admin

# Register your models here.

**employee/apps.py:**

from django.apps import AppConfig

class EmployeeConfig(AppConfig):

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

    name = 'employee'

**employee/forms.py:**

from django import forms

from employee.models import Employee

class EmployeeForm(forms.ModelForm):

    class Meta:

        model = Employee

        fields = "\_\_all\_\_"

**employee/models.py:**

from django.db import models

class Employee(models.Model):

    eid = models.CharField(max\_length=20)

    ename = models.CharField(max\_length=100)

    eemail = models.EmailField()

    econtact = models.CharField(max\_length=15)

    class Meta:

        db\_table = "employee"

**employee/urls.py:**

from django.contrib import admin

from django.urls import path

from employee import views

urlpatterns = [

    path('admin/', admin.site.urls),

    path('emp', views.emp),

    path('show',views.show),

    path('edit/<int:id>', views.edit),

    path('update/<int:id>', views.update),

    path('delete/<int:id>', views.destroy),

]

**employee/views.py:**

from django.shortcuts import render, redirect

from employee.forms import EmployeeForm

from employee.models import Employee

# Create your views here.

def emp(request):

    if request.method == "POST":

        form = EmployeeForm(request.POST)

        if form.is\_valid():

            try:

                form.save()

                return redirect('/show')

            except:

                pass

    else:

        form = EmployeeForm()

    return render(request,'index.html',{'form':form})

def show(request):

    employees = Employee.objects.all()

    return render(request,"show.html",{'employees':employees})

def edit(request, id):

    employee = Employee.objects.get(id=id)

    return render(request,'edit.html', {'employee':employee})

def update(request, id):

    employee = Employee.objects.get(id=id)

    form = EmployeeForm(request.POST, instance = employee)

    if form.is\_valid():

        form.save()

        return redirect("/show")

    return render(request, 'edit.html', {'employee': employee})

def destroy(request, id):

    employee = Employee.objects.get(id=id)

    employee.delete()

    return redirect("/show")

**employee/templates/index.html:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Index</title>

    {% load static %}

    <link rel="stylesheet" href="{% static 'css/style.css' %}"/>

</head>

<body>

<form method="POST" class="post-form" action="/emp">

        {% csrf\_token %}

    <div class="container">

<br>

    <div class="form-group row">

    <label class="col-sm-1 col-form-label"></label>

    <div class="col-sm-4">

    <h3>Enter Details</h3>

    </div>

  </div>

    <div class="form-group row">

    <label class="col-sm-2 col-form-label">Employee Id:</label>

    <div class="col-sm-4">

      {{ form.eid }}

    </div>

  </div>

  <div class="form-group row">

    <label class="col-sm-2 col-form-label">Employee Name:</label>

    <div class="col-sm-4">

      {{ form.ename }}

    </div>

  </div>

    <div class="form-group row">

    <label class="col-sm-2 col-form-label">Employee Email:</label>

    <div class="col-sm-4">

      {{ form.eemail }}

    </div>

  </div>

    <div class="form-group row">

    <label class="col-sm-2 col-form-label">Employee Contact:</label>

    <div class="col-sm-4">

      {{ form.econtact }}

    </div>

  </div>

    <div class="form-group row">

    <label class="col-sm-1 col-form-label"></label>

    <div class="col-sm-4">

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

    </div>

  </div>

    </div>

</form>

</body>

</html>

**employee/templates/edit.html:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Index</title>

    {% load staticfiles %}

    <link rel="stylesheet" href="{% static 'css/style.css' %}"/>

</head>

<body>

<form method="POST" class="post-form" action="/update/{{employee.id}}">

        {% csrf\_token %}

    <div class="container">

<br>

    <div class="form-group row">

    <label class="col-sm-1 col-form-label"></label>

    <div class="col-sm-4">

    <h3>Update Details</h3>

    </div>

  </div>

    <div class="form-group row">

    <label class="col-sm-2 col-form-label">Employee Id:</label>

    <div class="col-sm-4">

        <input type="text" name="eid" id="id\_eid" required maxlength="20" value="{{ employee.eid }}"/>

    </div>

  </div>

  <div class="form-group row">

    <label class="col-sm-2 col-form-label">Employee Name:</label>

    <div class="col-sm-4">

        <input type="text" name="ename" id="id\_ename" required maxlength="100" value="{{ employee.ename }}" />

    </div>

  </div>

    <div class="form-group row">

    <label class="col-sm-2 col-form-label">Employee Email:</label>

    <div class="col-sm-4">

        <input type="email" name="eemail" id="id\_eemail" required maxlength="254" value="{{ employee.eemail }}" />

    </div>

  </div>

    <div class="form-group row">

    <label class="col-sm-2 col-form-label">Employee Contact:</label>

    <div class="col-sm-4">

        <input type="text" name="econtact" id="id\_econtact" required maxlength="15" value="{{ employee.econtact }}" />

    </div>

  </div>

    <div class="form-group row">

    <label class="col-sm-1 col-form-label"></label>

    <div class="col-sm-4">

    <button type="submit" class="btn btn-success">Update</button>

    </div>

  </div>

    </div>

</form>

</body>

</html>

**Employee/templates/show.html:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Employee Records</title>

    {% load static %}

    <link rel="stylesheet" href="{% static 'css/style.css' %}"/>

</head>

<body>

<table class="table table-striped table-bordered table-sm">

    <thead class="thead-dark">

    <tr>

        <th>Employee ID</th>

        <th>Employee Name</th>

        <th>Employee Email</th>

        <th>Employee Contact</th>

        <th>Actions</th>

    </tr>

    </thead>

    <tbody>

{% for employee in employees %}

    <tr>

        <td>{{ employee.eid }}</td>

        <td>{{ employee.ename }}</td>

        <td>{{ employee.eemail }}</td>

        <td>{{ employee.econtact }}</td>

        <td>

            <a href="/edit/{{ employee.id }}"><span class="glyphicon glyphicon-pencil" >Edit</span></a>

            <a href="/delete/{{ employee.id }}">Delete</a>

        </td>

    </tr>

{% endfor %}

    </tbody>

</table>

<br>

<br>

<center><a href="/emp" class="btn btn-primary">Add New Record</a></center>

</body>

</html>