**CLASS BASED VIEW CRUD EXAMPLE**

**CREATE PROJECT, APP, FOLDERS**

PS C:\Users\kamal\django\_projects> django-admin startproject cbvEmployeeProject

PS C:\Users\kamal\django\_projects> cd .\cbvEmployeeProject\

PS C:\Users\kamal\django\_projects\cbvEmployeeProject> py -m manage startapp cbvEmpApp

PS C:\Users\kamal\django\_projects\cbvEmployeeProject> cd .

PS C:\Users\kamal\django\_projects\cbvEmployeeProject> mkdir templates

Directory: C:\Users\kamal\django\_projects\cbvEmployeeProject

PS C:\Users\kamal\django\_projects\cbvEmployeeProject> mkdir static

Directory: C:\Users\kamal\django\_projects\cbvEmployeeProject

PS C:\Users\kamal\django\_projects\cbvEmployeeProject> cd templates

PS C:\Users\kamal\django\_projects\cbvEmployeeProject\templates> mkdir cbvEmpApp

Directory: C:\Users\kamal\django\_projects\cbvEmployeeProject\templates

PS C:\Users\kamal\django\_projects\cbvEmployeeProject\templates> cd ..

PS C:\Users\kamal\django\_projects\cbvEmployeeProject> cd static

PS C:\Users\kamal\django\_projects\cbvEmployeeProject\static> mkdir css

Directory: C:\Users\kamal\django\_projects\cbvEmployeeProject\static

PS C:\Users\kamal\django\_projects\cbvEmployeeProject\static> mkdir images

Directory: C:\Users\kamal\django\_projects\cbvEmployeeProject\static

PS C:\Users\kamal\django\_projects\cbvEmployeeProject\static> cd..

PS C:\Users\kamal\django\_projects\cbvEmployeeProject>

**SETTINGS (settings.py)**

from pathlib import Path  
  
*# Build paths inside the project like this: BASE\_DIR / 'subdir'.*BASE\_DIR = Path(\_\_file\_\_).resolve().parent.parent  
STATIC\_DIR=Path(BASE\_DIR/'static')  
  
  
*# Quick-start development settings - unsuitable for production  
# See https://docs.djangoproject.com/en/5.0/howto/deployment/checklist/  
  
# SECURITY WARNING: keep the secret key used in production secret!*SECRET\_KEY = 'django-insecure-x4c\_szrexadtm(=we2+6z0m7d30)d\_\_j\_\_6q2h=&ry\_c!(u6+7'  
  
*# 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',  
 'cbvEmpApp'  
]  
  
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 = 'cbvEmployeeProject.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 = 'cbvEmployeeProject.wsgi.application'  
  
  
*# Database  
# https://docs.djangoproject.com/en/5.0/ref/settings/#databases*DATABASES={  
'default':{  
'ENGINE':'django.db.backends.mysql',  
'NAME':'employeeDb',  
'USER':'root',  
'PASSWORD':'Jerrick2911',  
'HOST':'localhost',  
'PORT':'3306'  
}  
}  
*# Password validation  
# https://docs.djangoproject.com/en/5.0/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/5.0/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/5.0/howto/static-files/*STATIC\_URL = 'static/'  
STATICFILES\_DIRS=[  
STATIC\_DIR,  
]  
  
*# Default primary key field type  
# https://docs.djangoproject.com/en/5.0/ref/settings/#default-auto-field*DEFAULT\_AUTO\_FIELD = 'django.db.models.BigAutoField'

**MODELS (models.py)**

from django.db import models  
from django.urls import reverse  
  
  
*# Create your models here.*class Employee(models.Model):  
 DEPARTMENT=[('Developer','DEVELOPER'),('Tester','TESTER'),('DBA','DBA'),('DevOps','DEVOPS'),('Cloud','Cloud'),  
 ('Security','SECURITY'),('Networking','NETWORKING'),('Support','SUPPORT')]  
 GENDER=[('Male','MALE'),('Female','FEMALE'),('Others','OTHERS')]  
 firstName=models.CharField(max\_length=20)  
 lastName=models.CharField(max\_length=20)  
 age=models.IntegerField(default=24)  
 gender = models.CharField(max\_length=10, choices=GENDER)  
 mobile=models.CharField(max\_length=25)  
 email=models.EmailField(max\_length=50)  
 department=models.CharField(max\_length=15,choices=DEPARTMENT)  
 salary=models.FloatField()  
 dateOfJoining=models.DateField(default='2024/01/01')  
  
 def get\_absolute\_url(self):  
 return reverse('detail',kwargs={'pk':self.pk})

**VIEWS (views.py)**

from django.shortcuts import render  
from django.views.generic import ListView, DetailView, CreateView, UpdateView, DeleteView  
from cbvEmpApp.models import Employee  
from django.urls import reverse\_lazy  
  
*# Create your views here.*class EmployeeListView(ListView):  
 model = Employee  
 *#default template\_name will be modelname\_list.html (employee\_list.html)  
 #default context\_object name is modelname\_list (employee\_list)*class EmployeeDetailedView(DetailView):  
 model = Employee  
 *#default template\_name will be modelname\_list.html (employee\_detail.html)  
 #default context\_object name is modelname (employee)*class EmployeeCreateView(CreateView):  
 model = Employee  
 fields = ('firstName','lastName','age','gender','mobile','email','department','salary','dateOfJoining')  
  
class EmployeeUpdateView(UpdateView):  
 model = Employee  
 *# FOR EDITING CERTAIN FIELDS* fields = ('mobile','email','department','salary',)  
 *#FOR EDITING ALL FIELDS  
 #fields = ('firstName','lastName','age','gender','mobile','email','department','salary','dateOfJoining')*class EmployeeDeleteView(DeleteView):  
 model=Employee  
 success\_url = reverse\_lazy('main')

**URLS (urls.py)**

from django.contrib import admin  
from django.urls import path  
from cbvEmpApp import views

urlpatterns = [  
 path('admin/', admin.site.urls),  
 path('employees/', views.EmployeeListView.as\_view(), name='main'),  
 path('<int:pk>/', views.EmployeeDetailedView.as\_view(), name='detail'),  
 path('create/', views.EmployeeCreateView.as\_view()),  
 path('update/<int:pk>/', views.EmployeeUpdateView.as\_view()),  
 path('delete/<int:pk>/', views.EmployeeDeleteView.as\_view())  
]

**MIGRAITON**

PS C:\Users\kamal\django\_projects\cbvEmployeeProject> py -m manage makemigrations

Migrations for 'cbvEmpApp':

cbvEmpApp\migrations\0001\_initial.py

- Create model Employee

PS C:\Users\kamal\django\_projects\cbvEmployeeProject> py -m manage migrate

Operations to perform:

Apply all migrations: admin, auth, cbvEmpApp, contenttypes, sessions

Running migrations:

Applying cbvEmpApp.0001\_initial... OK

**HTML(employee\_list.html)**

<!DOCTYPE html>  
{% load static %}  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Employee</title>  
 <link rel="stylesheet" href="{% static 'css/emp.css' %}" />  
</head>  
<body>  
 <div class="myDiv">  
 <h1> EMPLOYEES INFORMATION</h1>  
 <div>  
 <img src="{%static 'images/image2.jpg'%}" />  
 </div>  
 </div>  
 <div class="mycontainer">  
 {% if employee\_list %}  
 <table>  
 <thead>  
 <tr>  
 <th>Emp Id</th>  
 <th>First Name</th>  
 <th>Last Name</th>  
 <th>Age</th>  
 <th>Gender</th>  
 <th>Mobile</th>  
 <th>Email</th>  
 <th>Department</th>  
 <th>Salary</th>  
 <th>Date of Joining</th>  
 </tr>  
 </thead>  
 {% for emp in employee\_list %}  
 <tr>  
 <td><a href="/{{emp.id}}">{{emp.id}}</a></td>  
 <td>{{emp.firstName}}</td>  
 <td>{{emp.lastName}}</td>  
 <td>{{emp.age}}</td>  
 <td>{{emp.gender}}</td>  
 <td>{{emp.mobile}}</td>  
 <td>{{emp.email}}</td>  
 <td>{{emp.department}}</td>  
 <td>{{emp.salary}}</td>  
 <td>{{emp.dateOfJoining}}</td>  
 </tr>  
 {% endfor %}  
 </table>  
 {% else %}  
 <p>  
 Employee details not available... Shortly updated....  
 {% endif %}  
 <br> <br>  
 <div class="myDiv">  
 <button type="button" class="button button1"  
 onclick="window.location.href='/create';">  
 ADD EMPLOYEE  
 </button>  
 </div>  
 </div>  
</body>  
</html>

**HTML(employee\_form.html)**

<!DOCTYPE html>  
{% load static %}  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Employee</title>  
 <link rel="stylesheet" href="{% static 'css/emp.css' %}" />  
</head>  
<body>  
 <div class="myDiv">  
 <h1> ADD EMPLOYEES </h1>  
 <div>  
 <img src="{%static 'images/image2.jpg'%}" />  
 </div>  
 </div>  
 <div class="mycontainer">  
 <form method="post">  
 <table>  
 {{form.as\_table}}  
 {% csrf\_token %}  
 </table>  
 <br>  
 <input type="submit" value="Save">  
 <br> <br>  
 </form>  
 </div>  
</body>  
</html>

**HTML(employee\_detail.html)**

<!DOCTYPE html>  
{% load static %}  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Employee</title>  
 <link rel="stylesheet" href="{% static 'css/emp.css' %}" />  
</head>  
<body>  
 <div class="myDiv">  
 <h1> EMPLOYEES DETAILED INFORMATION</h1>  
 <div>  
 <img src="{%static 'images/image2.jpg'%}" />  
 </div>  
 </div>  
 <div class="mycontainerdisp">  
 <ol>  
 <h3><li>First Name : {{employee.firstName}}</li></h3>  
 <h3><li>Last Name : {{employee.lastName}}</li></h3>  
 <h3><li>Age : {{employee.age}}</li></h3>  
 <h3><li>Gender : {{employee.gender}}</li></h3>  
 <h3><li>Mobile : {{employee.mobile}}</li></h3>  
 <h3><li>Email : {{employee.email}}</li></h3>  
 <h3><li>Department : {{employee.department}}</li></h3>  
 <h3><li>Salary : {{employee.salary}}</li></h3>  
 <h3><li>Date Of Joining : {{employee.dateOfJoining}}</li></h3>  
  
 <button type="button" class="button button1" onclick="window.location.href='/employees';">  
 HOME  
 </button>  
 <button type="button" class="button button1" onclick="window.location.href='/update/{{employee.id}}';">  
 UPDATE  
 </button>  
 <button type="button" class="button button1" onclick="window.location.href='/delete/{{employee.id}}';">  
 DELETE  
 </button>  
 </ol>  
 </div>  
</body>  
</html>

**HTML(employee\_confirm\_delete.html)**

<!DOCTYPE html>  
{% load static %}  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Employee</title>  
 <link rel="stylesheet" href="{% static 'css/emp.css' %}" />  
</head>  
<body>  
 <div class="myDiv">  
 <h1> EMPLOYEES DELETE CONFIRMATION</h1>  
 <div>  
 <img src="{%static 'images/image2.jpg'%}" />  
 </div>  
 </div>  
 <div class="mycontainerdisp">  
 <table>  
 <thead>  
 <tr>  
 <th class="th1">  
 Do you want to delete the Employee ?  
 </th>  
 <th>  
 <form method="post">  
 <input type="submit" value="Confirm">  
 <a href="/{{student.id}}">Cancel</a>  
 {% csrf\_token %}  
 </form>  
 </th>  
 </tr>  
 </thead>  
 <tr>  
 <td>  
 First Name: {{employee.firstName}}  
 </td>  
 <td>  
 Last Name : {{employee.lastName}}  
 </td>  
 </tr>  
 </table>  
 </div>  
</body>  
</html>

**CSS FILE**

img{  
 weight: 100%;  
}  
  
h1{  
 color: blue;  
 font-size: 250%;  
 font-family:sans-serif;  
 text-align: center;  
 background-color: transparent;  
 border: 2px;  
}  
  
h2{  
 color: green;  
 font-size: 125%;  
 font-family: serif;  
 text-align: center;  
 background-color: transparent;  
 border: 2px;  
}  
  
h3{  
 color: blue;  
 font-size: 100%;  
 font-family: serif;  
 background-color: transparent;  
 border: 2px;  
}  
  
  
.mycontainer {  
 border: 10px green groove;  
 background-color: lightblue;  
 text-align: center;  
 display: flex;  
}  
  
.mycontainerdisp {  
 border: 10px green groove;  
 background-color: lightblue;  
 display: flex;  
}  
  
.myDiv {  
 border: 10px green groove;  
 background-color: lightblue;  
 text-align:center;  
}  
  
.myDiv1 {  
 border: 10px green groove;  
 background-color: lightblue;  
 text-align:justify;  
}  
  
table, th, td {  
 border: 2px solid white;  
 border-collapse: collapse;  
}  
  
th {  
 background-color: green;  
 text-align: center;  
 font-size: 16px;  
 font-weight: bolder;  
 font-family: serif;  
 margin: 4px 2px;  
}  
  
.th1 {  
 background-color: green;  
 text-align: left;  
 padding: 10px 10px;  
 font-size: 16px;  
 font-family: sans-serif;  
 margin: 4px 2px;  
}  
  
td{  
 background-color: cyan;  
 text-align: left;  
 padding: 10px 10px;  
 font-size: 16px;  
 font-family: sans-serif;  
 margin: 4px 2px;  
}  
  
td:hover {background-color: #D6EEEE;}  
  
p{  
 background-color: light;  
 text-align: left;  
 font-weight: 100;  
 font-family: sans-serif;  
 margin: 4px 2px;  
 color: green;  
}  
  
.button {  
 border: white;  
 color: green;  
 width:200px;  
 padding: 10px 10px;  
 text-align: center;  
 font-size: 16px;  
 margin: 2px 2px;  
 transition-duration: 0.5s;  
 cursor: pointer;  
}  
  
.button1 {  
 background-color: lightgreen;  
 color: blue;  
 border: 2px solid #04AA6D;  
 text-align: center;  
}

A group of people standing together

Description automatically generated

A group of people standing together

Description automatically generated

A group of people standing together

Description automatically generated

A group of people standing together

Description automatically generated

A group of people standing together

Description automatically generated

A group of people standing together

Description automatically generated

**DEFAULT AND CUSTOM FILTERS**

Create a directory called – templatetags and add an \_\_init\_\_.py blank file.

Next we can create a custom based filters and map it with the filed which we are passing

A screenshot of a computer screen

Description automatically generated

from django import template  
register=template.Library()  
  
*#register the filter using decorator*@register.filter(name='firstCapital')  
def firstUpper(value):  
 result=value.lower()  
 return result  
  
  
*#register the filter using decorator*@register.filter(name='addPrefix')  
def custPrefix(value, arg):  
 result=str(arg)+value.upper()  
 return result  
*#register the custom filer in the template library  
#register.filter("firstCapital",firstUpper)*

A computer screen shot of a course list

Description automatically generated