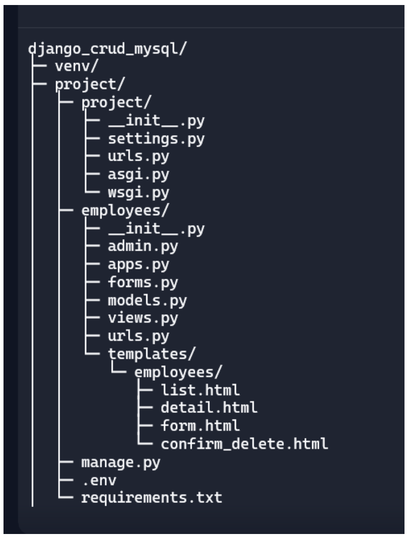
DJANGO CURD

Django CRUD with MySQL step by step with folder structure

Project overview and target folder structure



```
django_crud_mysql/
├─ venv/
├─ project/
│  ├─ project/
│  │  ├─ __init__.py
│  │  ├─ settings.py
│  │  ├─ urls.py
│  │  ├─ asgi.py
│  │  └─ wsgi.py
│  ├─ employees/
│  │  ├─ __init__.py
│  │  ├─ admin.py
│  │  ├─ apps.py
│  │  ├─ forms.py
│  │  ├─ models.py
│  │  ├─ views.py
│  │  ├─ urls.py
│  │  └─ templates/
│  │     └─ employees/
│  │        ├─ list.html
│  │        ├─ detail.html
│  │        ├─ form.html
│  │        └─ confirm_delete.html
│  ├─ manage.py
│  ├─ .env
│  └─ requirements.txt
```

1 **Create workspace and virtual environment:**

mkdir django_crud_mysql && cd django_crud_mysql
python -m venv venv // environment details should be there all in the project rood direct

? **Activate (Windows):**

venv\Scripts\activate

2 **Install dependencies:**

pip install django mysqlclient django-environ
pip freeze > requirements.txt // prod environment level

? **If mysqlclient fails on Windows:**

pip install pymysql

? In project/project/__init__.py add:

import pymysql
pymysql.install_as_MySQLdb()

3 **Create the Django project and app:**

django-admin startproject project
cd project
python manage.py startapp employees

MySQL database configuration

1. **Create database:**

CREATE TABLE orbcom.employeespython (
   id INT AUTO_INCREMENT PRIMARY KEY,
   firstname VARCHAR(100) NOT NULL,
   lastname VARCHAR(100) NOT NULL
);

```
INSERT INTO orbcom.employeespython (firstname, lastname)
VALUES
('Monika', 'Korukond');
commit;
select * from orbcom.employeespython;
```

2 **Add environment variables (.env at project root):**

```
DB_NAME= orbcom
DB_USER=root
DB_PASSWORD=admin
DB_HOST=localhost
DB_PORT=3306
DEBUG=True
```

3 **Configure settings (project/project/settings.py)**

```
import os
import environ

BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
env = environ.Env(DEBUG=(bool, True))
environ.Env.read_env(os.path.join(BASE_DIR, '.env'))

DEBUG = env('DEBUG')

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'employees',  // need to update
]

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': env('DB_NAME'),
        'USER': env('DB_USER'),
        'PASSWORD': env('DB_PASSWORD'),
        'HOST': env('DB_HOST'),
        'PORT': env('DB_PORT'),
        'OPTIONS': {'charset': 'utf8mb4'},
    }
}  // ned to include according to DB which we use

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',
        ],
    },
}]
```

Models, forms, views, urls, and templates

1 Register app (project/project/settings.py):

```
INSTALLED_APPS += ['employees']
```

2 Model (project/employees/models.py):

```
from django.db import models

class Employee(models.Model):
    name = models.CharField(max_length=100)
    role = models.CharField(max_length=50)
    salary = models.DecimalField(max_digits=10, decimal_places=2)
    created_at = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return f"{self.name} ({self.role})"
```

3 Admin (project/employees/admin.py):

```
from django.contrib import admin
from .models import Employee

@admin.register(Employee)
class EmployeeAdmin(admin.ModelAdmin):
    list_display = ('id', 'name', 'role', 'salary', 'created_at')
    search_fields = ('name', 'role')
    list_filter = ('role',)
```

4 Form (project/employees/forms.py):

```
from django import forms
from .models import Employee

class EmployeeForm(forms.ModelForm):
    class Meta:
        model = Employee
        fields = ['name', 'role', 'salary']
```

5 Views (project/employees/views.py):

```
from django.shortcuts import render, get_object_or_404, redirect
from .models import Employee
from .forms import EmployeeForm

def employee_list(request):
    employees = Employee.objects.order_by('id')
    return render(request, 'employees/list.html', {'employees': employees})

def employee_detail(request, pk):
    emp = get_object_or_404(Employee, pk=pk)
    return render(request, 'employees/detail.html', {'employee': emp})

def employee_create(request):
    if request.method == 'POST':
        form = EmployeeForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect('employee_list')
    else:
        form = EmployeeForm()
    return render(request, 'employees/form.html', {'form': form, 'title': 'Create Employee'})

def employee_update(request, pk):
    emp = get_object_or_404(Employee, pk=pk)
    if request.method == 'POST':
        form = EmployeeForm(request.POST, instance=emp)
        if form.is_valid():
            form.save()
            return redirect('employee_detail', pk=pk)
    else:
        form = EmployeeForm(instance=emp)
    return render(request, 'employees/form.html', {'form': form, 'title': 'Update Employee'})

def employee_delete(request, pk):
    emp = get_object_or_404(Employee, pk=pk)
    if request.method == 'POST':
        emp.delete()
        return redirect('employee_list')
    return render(request, 'employees/confirm_delete.html', {'employee': emp})
```

6 **App URLs (project/employees/urls.py):**

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

urlpatterns = [
    path('', views.employee_list, name='employee_list'),
    path('create/', views.employee_create, name='employee_create'),
    path('<int:pk>/', views.employee_detail, name='employee_detail'),
    path('<int:pk>/update/', views.employee_update, name='employee_update'),
    path('<int:pk>/delete/', views.employee_delete, name='employee_delete'),
]
```

7 **Project URLs (project/project/urls.py):**

```
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('employees/', include('employees.urls')),
]
```

8 **Templates (project/employees/templates/employees/):**

List.html

```
<!doctype html>
<h1>Employees</h1>
<a href="{% url 'employee_create' %}">Create</a>
<ul>
  {% for e in employees %}
    <li>
      <a href="{% url 'employee_detail' e.id %}">{{ e.name }} - {{ e.role }} - {{ e.salary }}</a>
      <a href="{% url 'employee_update' e.id %}">Edit</a>
      <form action="{% url 'employee_delete' e.id %}" method="post" style="display:inline">
        {% csrf_token %}<button type="submit">Delete</button>
      </form>
    </li>
  {% empty %}
    <li>No employees yet.</li>
```

```
 {% endfor %}
</ul>
```

Delete.html

```
<!doctype html>
<h1>{{ employee.name }}</h1>
<p>Role: {{ employee.role }}</p>
<p>Salary: {{ employee.salary }}</p>
<p>Created: {{ employee.created_at }}</p>
<a href="{% url 'employee_update' employee.id %}">Edit</a>
<a href="{% url 'employee_list' %}">Back</a>
```

form.html

```
<!doctype html>
<h1>{{ title }}</h1>
<form method="post">
  {% csrf_token %}
  {{ form.as_p }}
  <button type="submit">Save</button>
  <a href="{% url 'employee_list' %}">Cancel</a>
</form>
```

confirm_delete.html

```
<!doctype html>
<h1>Delete {{ employee.name }}?</h1>
<form method="post">
  {% csrf_token %}
  <button type="submit">Confirm</button>
  <a href="{% url 'employee_list' %}">Cancel</a>
</form>
```

Migrations, admin test, and run

1. **Make and apply migrations:**

```
python manage.py makemigrations
python manage.py migrate
```

2. **Create superuser and verify admin:**

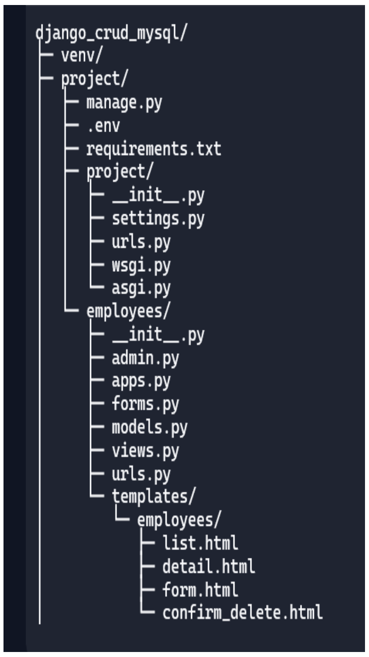
```
python manage.py createsuperuser
python manage.py runserver
```

->>> **Visit:** http://127.0.0.1:8000/admin and add a few employees.

3. **Test CRUD pages:**

List: http://127.0.0.1:8000/employees/
Create: http://127.0.0.1:8000/employees/create/
Detail/Update/Delete: via each employee’s links

Final folder structure snapshot



```
django_crud_mysql/
├─ venv/
├─ project/
│  ├─ manage.py
│  ├─ .env
│  ├─ requirements.txt
│  ├─ project/
│  │  ├─ __init__.py
│  │  ├─ settings.py
│  │  ├─ urls.py
│  │  ├─ wsgi.py
│  │  └─ asgi.py
│  └─ employees/
│     ├─ __init__.py
│     ├─ admin.py
│     ├─ apps.py
│     ├─ forms.py
│     ├─ models.py
│     ├─ views.py
│     ├─ urls.py
│     └─ templates/
│        └─ employees/
│           ├─ list.html
│           ├─ detail.html
│           ├─ form.html
│           └─ confirm_delete.html
```

```
CREATE TABLE orbcom.employeespython (
   id INT AUTO_INCREMENT PRIMARY KEY,
   firstname VARCHAR(100) NOT NULL,
   lastname VARCHAR(100) NOT NULL
);

INSERT INTO orbcom.employeespython (firstname, lastname)
VALUES
('Monika', 'Korukond');
commit;
select * from orbcom.employeespython;
```