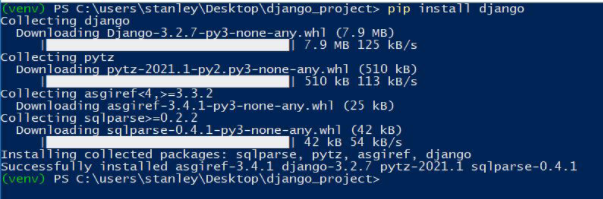
**CREATE WEBAPPLICATION USING DJANGO**

**Chapter 1 – Django setup**

**Step 1:** Install django using following the command

|  |
| --- |
| **pip install django** |

****

once the installation finishes,you need to verify Django

has beeninstalled, to do that following command

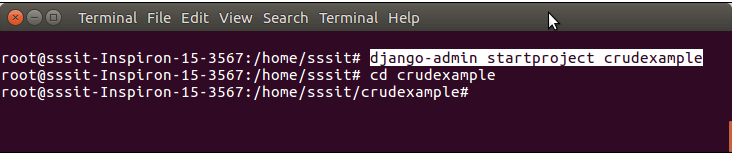


You will get output showing you the Django version installed on your system



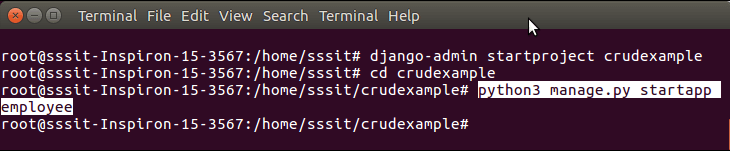
**Step 2:** create a Django project

|  |
| --- |
| **Command: django-admin startproject projectname** |

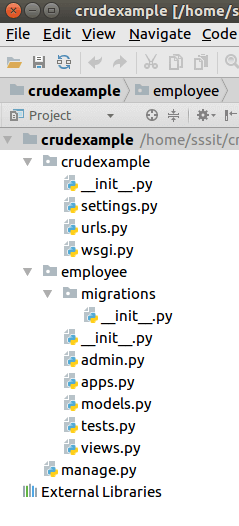


**Step 3:** create an app inside the project

|  |
| --- |
| **Command: django-admin startapp appname** |

****

**project structure**

****

**Step 4: migrate your project using following the command**

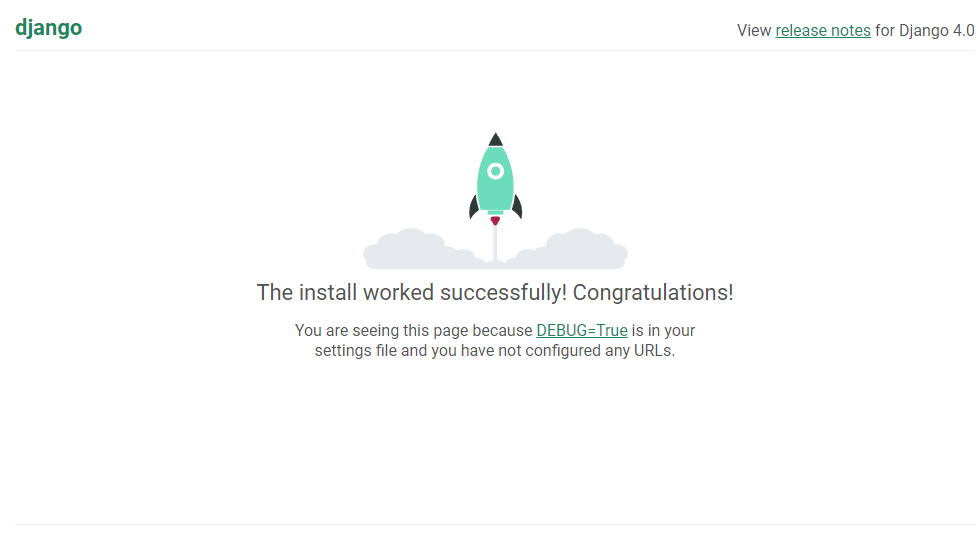
|  |
| --- |
| **python manage.py migrate** |

**Step 5: Runserver using following the command**

|  |
| --- |
| **python manage.py runserver** |

**Step 6:** Access the application by entering **localhost:8000**

**http://127.0.0.1:8000/**

****

**Chapter 2 – Link Html files in django**

**Step 1:** Create a **templates** folder inside the **employee** app and create(home,add,show)html files inside the directory.

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Home</title>

//bootstrap link //

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">

</head>

<body>

        <div class="container-fluid">

          <a class="navbar-brand" href="#">Navbar</a>

          <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">

            <span class="navbar-toggler-icon"></span>

          </button>

          <div class="collapse navbar-collapse" id="navbarNav">

            <ul class="navbar-nav">

              <li class="nav-item">

                <a class="nav-link active" aria-current="page" href="#">Home</a>

              </li>

              <li class="nav-item">

                <a class="nav-link" href="/add">ADD</a>

              </li>

              <li class="nav-item">

                <a class="nav-link" href="/show">SHOW</a>

              </li>

            </ul>

          </div>

        </div>

      </nav>

</body>

</html>

**Step 2:** **Provide Routing**

Provide URL patterns to map with views function.

**//urls.py**

from django.contrib import admin

from django.urls import path

from employee import views

urlpatterns = [

    path('admin/', admin.site.urls),

path(‘ ‘,views.home),

path(‘ ‘,views.add)’

path(‘ ‘,views.show),

**step 3: create a view function**

**//views.py**

***>>myproject/myapp/views.py***

from django.shortcuts import render

def home(request):

    return render(request,'home.html')

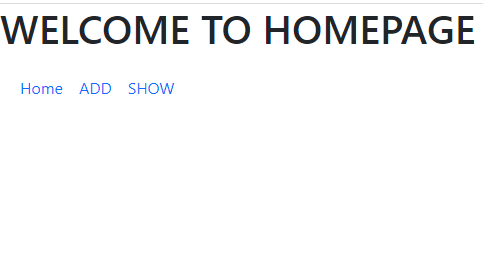
def add(request):

    return render(request,'add.html')

def show(request):

    return render(request,'show.html')

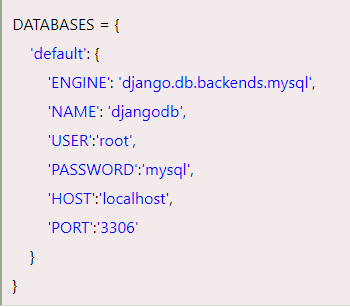
**step 4: Migrate and Run the server**

****

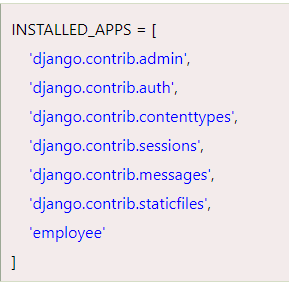
**Chapter 3 – Database setup**

**Step 1:** Create a database **djangodb** in mysql

**Step 2:** Configure into the **settings.py** file of django project. See the example.

****

Add your app name into installed app



**Chapter 4: Creating models and forms**

**Step 1:** put the following code in 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"

**Step 2:** create a forms.py inside the app

from django **import** forms

from employee.models **import** Employee

**class** EmployeeForm(forms.ModelForm):

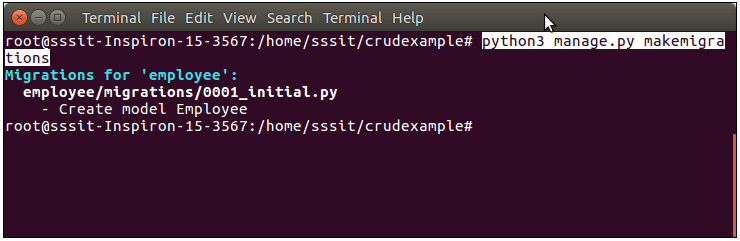
**class** Meta:

         model = Employee

         fields = "\_\_all\_\_"

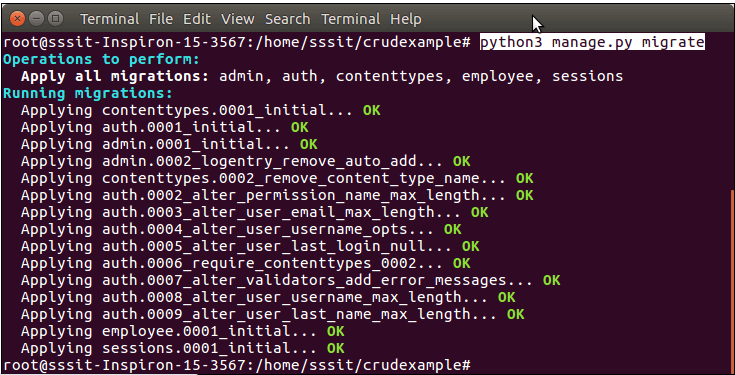
**Step 3:** Create migrations for the created model employee, use the following command.

**Command: python manage.py makemigrations**



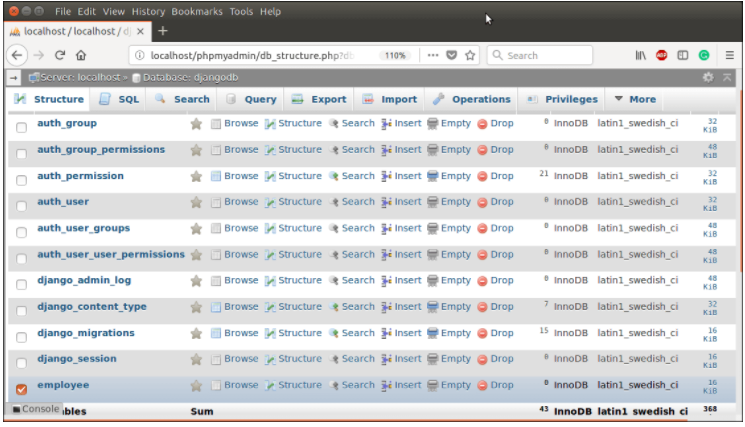
**Step 4:** Run the command to migrate the migrations.

**Command: python manage.py migrate**

****

Now, our application has successfully connected and created tables in database. It creates 10 default tables for handling project (session, authentication etc) and one table of our model that we created.

See list of tables created after migrate command.



**Step 14:** Runserver to following command

**Command:python manage.py runserver**

****

**Chapter 5 –Add a employee in database**

**step 1:** create view functions

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})

**step 2:** goto add.html and write following code

<!DOCTYPE html>

<html>

<head>

    <title>ADD</title>

//bootstrap link //

**<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/css/bootstrap.min.css" integrity="sha384-GJzZqFGwb1QTTN6wy59ffF1BuGJpLSa9DkKMp0DgiMDm4iYMj70gZWKYbI706tWS" crossorigin="anonymous">**

</head>

<body>

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

        {% csrf\_token %}

  <div class="container">

        <div class="form-group row">

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

            <div class="col-sm-4">

                <h3>Enter Details</h3>

            </div>

        </div>

        <div class="form-group row">

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

            <div class="col-sm-4">

                {{ form.eid }}

            </div>

        </div>

        <div class="form-group row">

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

            <div class="col-sm-4">

                {{ form.ename }}

            </div>

        </div>

        <div class="form-group row">

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

            <div class="col-sm-4">

                {{ form.eemail }}

            </div>

        </div>

<div class="form-group row">

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

<div class="col-sm-4">

                {{ form.econtact }}

            </div>

  </div>

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

    </div>

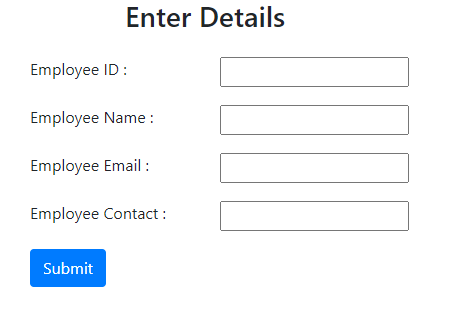
</form>

</body>

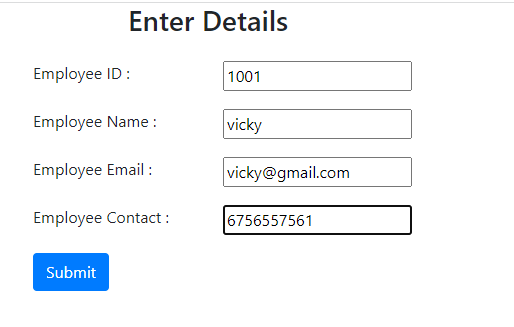
</html>

**step 3: Run the server**

output:



**Step 4:** Enter employee details and see the employee details in database



In database



**Chapter 6-Show a employee details in browser**

**Step 1:** create a view function

def show(request):

    employees = Employee.objects.all()

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

**step 2:** goto show.html and write following code

<!DOCTYPE html>

<html>

<head>

    <title>Employee Record</title>

//bootstrap link //

**<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/css/bootstrap.min.css" integrity="sha384-GJzZqFGwb1QTTN6wy59ffF1BuGJpLSa9DkKMp0DgiMDm4iYMj70gZWKYbI706tWS" crossorigin="anonymous">**

</head>

<body>

    <div class="container">

    <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>

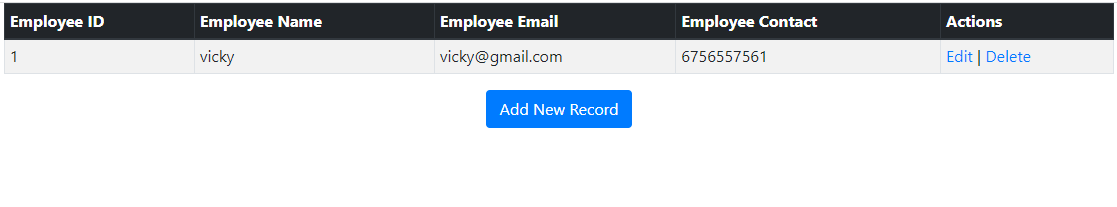
    <center><a href="/add" class="btn btn-primary">Add New Record</a></center>

</div>

</body>

</html>

Step 3: Run the server and see the employee details in browser



**Chapter 7: Edit the employee details**

**step 1:** create a url for edit page

from django.contrib import admin

from django.urls import path

from crud import views

urlpatterns = [

    path('admin/', admin.site.urls),

    path('',views.home),

    path('emp',views.emp),

    path('show',views.show),

    path('edit/<int:id>',views.edit),

**Step 2:** create a view function

def edit(request,id):

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

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

**Step 3:** goto edit .html and the following code

<!DOCTYPE html>

<html>

<head>

    <title>Edit page</title>

**<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/css/bootstrap.min.css" integrity="sha384-GJzZqFGwb1QTTN6wy59ffF1BuGJpLSa9DkKMp0DgiMDm4iYMj70gZWKYbI706tWS" crossorigin="anonymous">**

</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-lable"></label>

            <div class="col-sm-4">

                <h3>Update Details</h3>

            </div>

        </div>

        <div class="form-group row">

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

            <div class="col-sm-4">

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

            </div>

        </div>

        <div class="form-group row">

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

  <div class="col-sm-4">

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

            </div>

        </div>

        <div class="form-group row">

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

            <div class="col-sm-4">

      <input type="text" name="eemail" id="id\_eemail" value="{{ employee.eemail }}"/>

            </div>

        </div>

        <div class="form-group row">

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

            <div class="col-sm-4">

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

            </div>

        </div>

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

    </div>

</form>

</body>

</html>

**Step 4:** create a update url

from django.contrib import admin

from django.urls import path

from employee import views

urlpatterns = [

    path('admin/', admin.site.urls),

  path('',views.home),

    path('add',views.add),

    path('show',views.show),

    path('edit/<int:id>',views.edit),

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

**step 5: create a view function**

def update(request,id):

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

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

    if form.is\_valid():

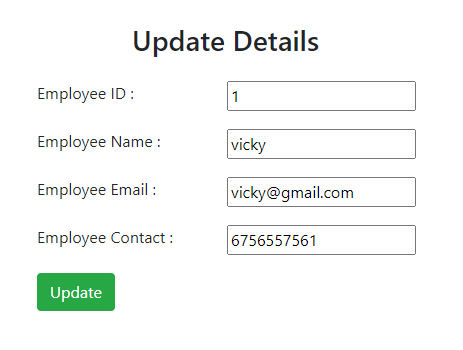
        form.save()

        return redirect('/show')

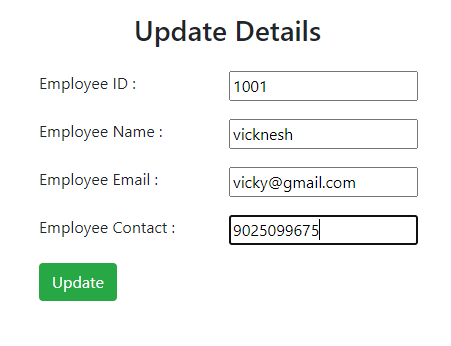
    return render(request,"edit.html",{'employee':employee})

**Step 6:** Migrate and Run the server

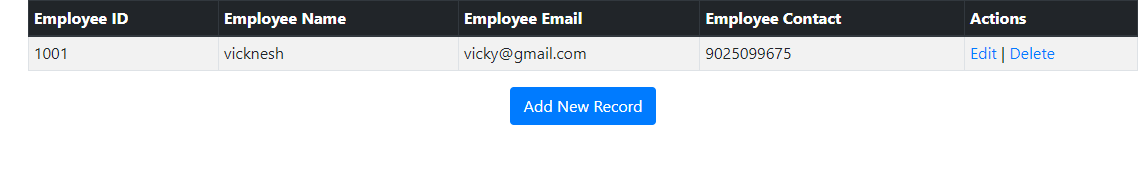
Output:



**After Updating:**

****

**View your Show page**

****

**Chapter 8 – Delete the employee details**

**Step 1:create a delete url**

from django.contrib import admin

from django.urls import path

from employee import views

urlpatterns = [

    path('admin/', admin.site.urls),

    path('',views.home),

    path('add',views.add),

    path('show',views.show),

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

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

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

]

**Step 2:** create a view function

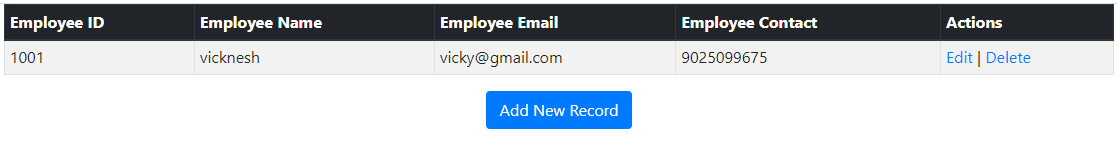
def delete(request,id):

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

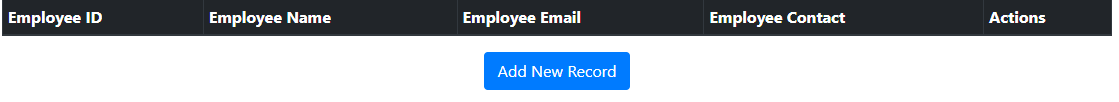
    employee.delete()

    return redirect("/show")

**Step 3:** Delete your employee details in show.html

****

**After Deleting employee details**

****