Django

Install django

* pip install django

Create project

* django-admin startproject college

Go into project directory

* cd college

Create applications in project level

* Python manage.py startapp students
* Python manage.py startapp branch
* Python manage.py startapp admins

Create *templates* & *static* folder in project level

* in settings.py (project level)

Register all applications in *settings.py* in project level under *INSTALLED\_APPS & rest\_framework* also.

Register *TEMPLATE\_DIR = Path.joinpath(BASE\_DIR, 'templates')*

Register *STATIC\_DIR = Path.joinpath(BASE\_DIR, 'static')*

Under *TEMPLATES*

Copy *TEMPLATE\_DIR* in *DIRS*

Below *STATIC\_URL*

*STATICFILES\_DIRS = [STATIC\_DIR]*

REST\_FRAMEWORK = {

'DEFAULT\_AUTHENTICATION\_CLASSES': (

'rest\_framework.authentication.SessionAuthentication',

),

}

Add in settings.py

LOGOUT\_REDIRECT\_URL = '/'

LOGIN\_REDIRECT\_URL = '/'

CONNECT MSSQL TO DJANGO

* In *settings*.*py* change-

DATABASES = {

'default': {

'ENGINE': 'mssql',

'NAME': 'college',

'USER': '',

'PASSWORD': '',

'HOST': 'DESKTOP-KIRAN',

'PORT': '',

'OPTIONS': {"driver": "ODBC Driver 17 for SQL Server",}}}

To check connection open command prompt in pycharm terminal window

* python manage.py shell
* from django.db import connection
* x = connection.cursor()
* exit()
* in urls.py (college project level)

from django.contrib import admin

from django.urls import path, include

from admins import views

urlpatterns = [

path('admin/', admin.site.urls),

path(**''**, views.home),

path('branch/', include('branch.urls')),

path('admins/', include('admins.urls')),

path('students/', include('students.urls')),

path("accounts/", include("django.contrib.auth.urls")),

path("signup/", views.signup),

]

* in models.py (branch app level)

from **django.db** import **models**

class Mechanical(**models.Model**):

gen = (("Male", 'Male'), ("Female", "Female"), ("Unspecified", "Unspecified"))

name = **models**.CharField(max\_length=50)

age = models.IntegerField()

phone = models.BigIntegerField()

email = models.EmailField()

gender = models.CharField(max\_length=20, **choices**=gen, default="Male")

* in views.py (branch app level)

from django.**shortcuts** import **render**

from .forms import MechForms

from django.http import HttpResponseRedirect, HttpResponse

from .models import Mechanical

from **django.contrib.auth.decorators** import **login\_required**

def Mechanical\_forms(r):

**mech\_user = MechForms**

if r.method == 'POST':

**forms = MechForms(r.POST)**

if forms.is\_valid():

forms.save()

return HttpResponse('<h1>Successful...</h1>')

return render(r, 'branch/mechanical.html',

{"mech\_user": mech\_user})

**@login\_required()**

def data(r):

table = Mechanical.objects.all()

return render(r, 'admins/admins.html', {"table": table})

def **update(r,id):**

obj = Mechanical.objects.get(**id=id**)

# fetch data from database for particular id

if r.method == 'POST':

**form\_u = MechForms(r.POST, instance=obj)**

# map data from database & data from user

if form\_u.is\_valid():

form\_u.save()

return HttpResponseRedirect('/admins/admins')

return render(r, 'branch/mech\_update.html', {'obj': obj})

def **delete**(r,id):

obj = Mechanical.objects.get(id=id)

obj.delete()

return HttpResponseRedirect('/admins/admins')

* in **forms.py** (branch app level)

from **django** import **forms**

from **.models** import **Mechanical**

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

class Meta:

model = Mechanical

fields = "\_\_all\_\_"

def clean\_age(self):

your\_age = self.**cleaned\_data**['age']

if your\_age >= 16:

return your\_age

else:

**raise** **forms.ValidationError**('Barkya Ghari ja')

* in **urls.py** (branch app level)

from django.urls import path

from branch import views

urlpatterns = [

path('mechanical/', views.Mechanical\_forms),

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

path('delete**/<id>/**', views.delete),]

To register model in admin panel

* in **admin.py** (branch app level)

from django.contrib import admin

from .models import Mechanical

class MechAdmin(admin.ModelAdmin):

list\_display= ["name", "age", "phone", "email", "gender"]

**admin.site.register(Mechanical, MechAdmin)**

* signup form in **forms.py** (admins app level)

from **django.contrib.auth.models** import **User**

from django import forms

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

password= forms.CharField(widget=forms.PasswordInput())

class Meta:

model = User

fields = ['username', 'password', 'email', 'first\_name', 'last\_name']

* in **views.py** (admins app level)

from django.shortcuts import render

from .forms import SignUp

from django.http import HttpResponseRedirect

def home(r):

return render(r, "home.html")

def signup(r):

form1 = SignUp # forms from forms.py

if r.method == 'POST':

forms\_s=SignUp(r.POST) # create new forms with new data

if forms\_s.is\_valid():

**user**=forms\_s.save() # save forms in var

**user.set\_password(user.password)** # pass the password to set\_password to encrypt

user.save()

return HttpResponseRedirect('/')

return render(r, 'registration/signup.html', {"form1": form1})

Django REST API

* in **views.py** (students app level)

from rest\_framework.**views** import **APIView** # to view good look

from rest\_framework.**response** import **Response** # tp create response in json

from branch.models import Mechanical

from rest\_framework import **status**

from .serializers import MechanicalSer

class StudentRest(APIView):

def get(self, r):

obj = Mechanical.objects.all()

# in obj data will receive in queryset

serobj = **MechanicalSer**(obj**, many=True**)

# pass queryset to serializer module created, it gives json object

return **Response**(**serobj.data**)

def post(self, r):

# in post client sends the data, that data we need to catch in post

# data come through post is in json format

# we need to serialize the data to queryset & dump to database

obj = MechanicalSer(**data=r.data**)

# we pass request+data came from POST to deserializer

if obj.is\_valid():

obj.save()

return **Response**(obj.data,

status = status.HTTP\_201\_CREATED)

return **Response**(obj.errors,

status = status.HTTP\_400\_BAD\_REQUEST)

class StudentUpdDel(APIView):

def put(self, r, pk):

obj = Mechanical.objects.get(**pk=pk**)

serobj = MechanicalSer(obj, **data=r.data**)

if serobj.is\_valid():

serobj.save()

return Response(**serobj.data**,

status=status.HTTP\_201\_CREATED)

return Response(**serobj.errors**,

status=status.HTTP\_400\_BAD\_REQUEST)

def delete(self, r, pk):

obj = Mechanical.objects.get(**pk=pk**)

obj.delete()

return Response(status=status.HTTP\_204\_NO\_CONTENT)

* in **serializers.py** (students app level)

from rest\_framework import serializers

from branch.models import Mechanical

class MechanicalSer(serializers.ModelSerializer):

class Meta:

model = Mechanical

fields = '\_ \_all\_ \_'

* in **urls.py** (students app level)

from django.urls import path

from students import views

urlpatterns = [

path('api/mechanical/', views.StudentRest.as\_view()),

path('api/mechanical/<pk>',views.StudentUpdDel.as\_view()]

* in **urls.py** (admins app level)

from django.urls import path

from branch import views

urlpatterns = [ path('admins/', views.data),]

* in **urls**.py (college project level)

from django.contrib import admin

from django.urls import path, include

from admins import views

urlpatterns = [ path(**''**, views.home),

path('admin/', admin.site.urls),

path('branch/', include('branch.urls')),

path('admins/', include('admins.urls')),

path('students/', include('students.urls')),

path("accounts/", include("django.contrib.auth.urls")),

path("signup/", views.signup),]

* in **urls**.py (branch level)

from django.urls import path

from branch import views

urlpatterns = [ path('update/<id>/', views.update),

path('mechanical/', views.Mechanical\_forms),

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

* in **admins.html** (templates/admins) to show data from db🡪 branch/views.data🡪”table”

{% if table %}

<table class="table table-hover">

<thead>

<tr>

<th scope="col">id</th>

. . .

<th scope="col"> </th>

</tr>

</thead>

{% for cad in table %}

<tr style="color:green">

<td>{{cad.id}}</td>

. . .

<td><a class="btn btn-warning"

href**="/branch/update/{{cad.id}}"**

role="button">Update</a></td>

<td><a class="btn btn-danger"

href**="/branch/delete/{{cad.id}}"**

role="button">! Delete</a></td>

</tr>

{% endfor %}

</table>

{% else %}

<h1>NO DATA</h1>

{% endif %}

* in **mech\_update.html** (templates/branch)

to update data 🡪 branch/views.update🡪”obj”

<**form** method = 'post'>

**{% csrf\_token %}**

**{{obj.as\_p}}**

<br>

Name : <input type="text" name="name" value="{{**obj.name**}}"><br><br>

Age : <input type="text" name="age" value="{{**obj.age**}}"><br><br>

Phone : <input type="text" name="phone" value="{{**obj.phone**}}"><br><br>

E-mail : <input type="text" name="email" value="{{**obj.email**}}"><br><br>

Gender : <select name="gender" id="id\_gender">

<option value="Male">Male</option>

<option value="Female">Female</option>

<option value="Unspecified">Unspecified</option>

</select><br><br><br>

<input class="btn btn-warning" type="submit" value="Update'{{**obj.name**}}'" role="button">

</**form**>

* in **mechanical.html** (templates/branch)

to input the data from user 🡪 branch/views.Mechanical\_forms 🡪 ”mech\_user”

{% extends 'home.html' %}

{% block baseblock %}

<body>

<h1 align="center"> Laptop Enquiry </h1>

<div class="container" align="center" >

<form method="POST">

{% csrf\_token %}

{{mech\_user.as\_p}}

<input value="submit" type="submit">

</form>

</div>

</body>

{% endblock %}

* in **home.html** (templates) base page 🡪 admins/views.home

<li class="nav-item">

<a class="nav-link" href="/admins/admins">Admin</a> </li>

<li class="nav-item">

<a class="nav-link" href="/students/api/mechanical/">Student Corner</a> </li>

<a class="dropdown-item" href="/branch/mechanical">Mechanical</a>

<a class="dropdown-item" href="/accounts/login">Login</a>

<a class="dropdown-item" href="/accounts/logout">Log out</a>

<a class="dropdown-item" href="/signup">SignUp</a>

* in **login.html** (templates/registration)

user login page 🡪 django.contrib.auth.urls 🡪 ”form”

{% extends 'home.html' %} {% block baseblock %}

<body>

<div class="container" align="center" >

<h1>Login</h1>

<form method="**POST**">

**{% csrf\_token %}**

**{{form.as\_p}}**

<input value="submit" type="submit">

</form>

</div>

</body>

{% endblock %}

* in **signup.html** (templates/registration)

user sign up page 🡪 admins/views.signup 🡪 ”form1”

{% extends 'home.html' %} {% block baseblock %}

<body>

<div class="container" align="center" >

<h1>SignUp</h1>

<form method="**POST**">

**{% csrf\_token %}**

**{{form1.as\_p}}**

<input value="submit" type="submit">

</form>

</div>

</body>

{% endblock %}