**SOURCE CODE**

**Urls.py**

"""PatentLegislation URL Configuration

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

https://docs.djangoproject.com/en/2.2/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 django.conf import settings

from django.conf.urls.static import static

from users import views as users

from admins import views as admins

from copyright import views as copyright

urlpatterns = [

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

path('', users.index, name='index'),

path('index/', users.index, name='index'),

path('logout/', users.logout, name='logout'),

path('userlogin/', users.userlogin, name='userlogin'),

path('userregister/', users.userregister, name='userregister'),

path('userlogincheck/', users.userlogincheck, name='userlogincheck'),

path('upload/', users.upload, name='upload'),

path('granted/', users.granted, name='granted'),

path('adminlogin/', admins.adminlogin, name='adminlogin'),

path('adminloginentered/', admins.adminloginentered, name='adminloginentered'),

path('viewusers/', admins.viewusers, name='viewusers'),

path('activateuser/', admins.activateuser, name='activateuser'),

path('viewdata/', admins.viewdata, name='viewdata'),

path('AdminAccept/', admins.AdminAccept, name='AdminAccept'),

path('AdminDelete/', admins.AdminDelete, name='AdminDelete'),

path('copyrightadministered/', copyright.copyrightadministered, name='copyrightadministered'),

path('copyrightloginentered/', copyright.copyrightloginentered, name='copyrightloginentered'),

path('permissions/', copyright.permissions, name='permissions'),

path('sendpermission/', copyright.sendpermission, name='sendpermission'),

path('sendreject/', copyright.sendreject, name='sendreject'),

]

if settings.DEBUG:

urlpatterns += static(settings.MEDIA\_URL, document\_root=settings.MEDIA\_ROOT)

user side views.py

from django.contrib import messages

from django.shortcuts import render, redirect

from django.http import HttpResponse

# Create your views here.

from users.forms import userForm, UploadForm

from users.models import UserModel, UploadModel

def index(request):

return render(request,'index.html')

def logout(request):

return render(request,'index.html')

def userlogin(request):

return render(request,"user/userlogin.html")

def userregister(request):

if request.method=='POST':

form1 = userForm(request.POST)

if form1.is\_valid():

form1.save()

return render(request, "user/userlogin.html")

#return HttpResponse("registreration succesfully completed")

else:

print("form not valied")

return HttpResponse("form not valied")

else:

form = userForm()

return render(request,"user/userregister.html",{"form":form})

def userlogincheck(request):

if request.method == "POST":

email = request.POST.get('uname')

pswd = request.POST.get('upasswd')

print("Email = ", email, ' Password = ', pswd)

try:

check = UserModel.objects.get(email=email,passwd=pswd)

status = check.status

print('Status is = ', status)

if status == "Activated":

request.session['id'] = check.id

#request.session['name'] = check.name

request.session['email'] = check.email

print("User id At", check.id, status,check.email)

return render(request, 'user/userpage.html', {})

else:

messages.success(request, 'Your Account Not at activated')

return render(request, 'user/userlogin.html')

# return render(request, 'user/userpage.html',{})

except Exception as e:

print('Exception is ', str(e))

pass

messages.success(request, 'Invalid Email id and password')

return render(request, 'user/userlogin.html')

def upload(request):

if request.method == 'POST':

form = UploadForm(request.POST, request.FILES)

if form.is\_valid():

form.save()

print("form saved")

files = UploadModel.objects.filter(email=request.session['email'])

return render(request, 'user/upload\_list.html',{'files': files})

else:

data = {'email': request.session['email'],'category':'Computer Technology','title':'title','description':'Description','file':'python library with versions.txt','status':'waiting','adminstatus':'waiting'}

form = UploadForm(data=data)

return render(request, 'user/uploadfile.html', {'form': form})

def upload\_list(request):

files = UploadModel.objects.all()

return render(request, 'user/upload\_list.html', {'files': files})

def granted(request):

email = request.session['email']

sts = 'granted'

dict = UploadModel.objects.filter(email=email)

return render(request,'user/granted.html',{'object':dict})

forms.py

from django import forms

from django.core import validators

from django.http import HttpRequest

from users.models import UserModel, UploadModel

def name\_check(value):

if value.isalpha()!=True:

raise forms.ValidationError("only string are allowed")

class userForm(forms.ModelForm):

firstname = forms.CharField(widget=forms.TextInput(), required=True, max\_length=100, validators=[name\_check])

lastname = forms.CharField(widget=forms.TextInput(), required=True, max\_length=100, validators=[name\_check])

passwd = forms.CharField(widget=forms.PasswordInput(), required=True, max\_length=100)

email = forms.CharField(widget=forms.TextInput(), required=True)

mobileno= forms.CharField(widget=forms.TextInput(), required=True, max\_length=10,validators=[validators.MaxLengthValidator(10),validators.MinLengthValidator(10)])

qualification = forms.CharField(widget=forms.TextInput(), required=True, max\_length=100)

city = forms.CharField(widget=forms.TextInput(), required=True, max\_length=100, validators=[name\_check])

status = forms.CharField(widget=forms.HiddenInput(), initial='waiting', max\_length=100)

def \_\_str\_\_(self):

return self.email

class Meta:

model = UserModel

fields=['firstname','lastname','passwd','email','mobileno','qualification','city','status']

# iterable

key\_technology =(

("Computer Technology", "Computer Technology"),

("Management Control", "Management Control"),

("Digital Communication Measurement", "Digital Communication Measurement"),

("Telecommunications", "Telecommunications"),

("Medical Technology", "Medical Technology"),

("Games", "Games"),

("Handling Furniture", "Handling Furniture"),

("Electrical Machinery", "Electrical Machinery"),

("Transport", "Transport"),

("Audio-visual technology", "Audio-visual technology"),

("Other Special Machines", "Other Special Machines"),

("Civil Engineering", "Civil Engineering"),

("Other Consumer Goods", "Other Consumer Goods"),

("Thermal processes and apparatus", "Thermal processes and apparatus"),

("Machine Tools", "Machine Tools"),

("Analysis of Biological Materials", "Analysis of Biological Materials"),

("Chemical Engineering", "Chemical Engineering"),

("Engines", "Engines"),

("Pumps", "Pumps"),

("Turbines", "Turbines"),

("Optics", "Optics"),

("Environmental Technology", "Environmental Technology"),

("Mechanical Elements", "Mechanical Elements"),

("Materials", "Materials"),

("Metallurgy", "Metallurgy"),

("Biotechnology", "Biotechnology"),

("Basic Communication Processes", "Basic Communication Processes"),

("Food Chemistry", "Food Chemistry"),

("Semiconductors", "Semiconductors"),

("Textile and Paper Machines", "Textile and Paper Machines"),

("Macro-Molecular Chemistry", "Macro-Molecular Chemistry"),

("Polymers", "Polymers"),

("Basic Materials Chemistry", "Basic Materials Chemistry"),

("Pharmaceuticals", "Pharmaceuticals"),

("Surface Technology", "Surface Technology"),

("Coating","Coating"),

("Organic fine Chemistry","Organic fine Chemistry"),

("Micro-structure and nano-technology","Micro-structure and nano-technology"),

)

class UploadForm(forms.ModelForm):

#email = forms.CharField(max\_length=100, initial=request.session['email'])

description = forms.CharField(widget=forms.Textarea(), required=True, max\_length=100)

category = forms.ChoiceField(choices=key\_technology)

class Meta:

model = UploadModel

fields = ('email','category','title','description','file','status','adminstatus')

models.py

from django.db import models

# Create your models here.

class UserModel(models.Model):

firstname = models.CharField(max\_length=50)

lastname = models.CharField(max\_length=50)

email = models.EmailField()

passwd = models.CharField(max\_length=40)

mobileno = models.CharField(max\_length=50, default="", editable=True)

qualification = models.CharField(max\_length=40)

city = models.CharField(max\_length=50)

status = models.CharField(max\_length=40,default="", editable=True)

def \_\_str\_\_(self):

return self.email

class Meta:

db\_table='userregister'

class UploadModel(models.Model):

#uuid = models.CharField(max\_length=30)

email = models.EmailField(default="", editable=True)

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

title = models.CharField(max\_length=400,unique=True)

description = models.CharField(max\_length=1000,blank=True)

file = models.FileField(upload\_to='files/pdfs/')

status = models.CharField(max\_length=600, default='waiting')

adminstatus = models.CharField(max\_length=600, default='waiting')

def \_\_str\_\_(self):

return self.category

class Meta:

db\_table='Uploadfile'

admin side views.py

from django.http import HttpResponse

from django.shortcuts import render

# Create your views here.

from users.models import UserModel, UploadModel

def adminlogin(request):

return render(request, "admins/adminlogin.html")

def adminloginentered(request):

if request.method == 'POST':

uname=request.POST['uname']

passwd=request.POST['upasswd']

if uname == 'admin' and passwd == 'admin@2020':

return render(request,"admins/adminloginentered.html")

else:

return HttpResponse("invalied credentials")

def viewusers(request):

object = UserModel.objects.all()

return render(request, "admins/viewusers.html", {"object": object})

def viewdata(request):

object = UploadModel.objects.all()

return render(request, "admins/viewdata.html", {"object": object})

def activateuser(request):

if request.method =='GET':

uname=request.GET.get('pid')

print(uname)

status='Activated'

print("pid=",uname,"status=",status)

UserModel.objects.filter(id=uname).update(status=status)

object=UserModel.objects.all()

return render(request,"admins/viewusers.html",{"object":object})

def AdminAccept(request):

id = request.GET.get('id')

UploadModel.objects.filter(id=id).update(adminstatus='granted')

object = UploadModel.objects.all()

return render(request, "admins/viewdata.html", {"object": object})

def AdminDelete(request):

id = request.GET.get('id')

UploadModel.objects.filter(id=id).delete()

object = UploadModel.objects.all()

return render(request, "admins/viewdata.html", {"object": object})

copyright Views.py

from django.http import HttpResponse

from django.shortcuts import render

import speech\_recognition as sr

import pyttsx3

# Create your views here.

from users.models import UploadModel

def copyrightadministered(request):

return render(request, "copyright/copyrightadministered.html")

def copyrightloginentered(request):

if request.method == 'POST':

uname=request.POST['uname']

passwd=request.POST['upasswd']

if uname == 'copyright' and passwd == 'copyright':

return render(request,"copyright/copyrightloginentered.html")

else:

return HttpResponse("invalied credentials")

def permissions(request):

object = UploadModel.objects.all()

return render(request, "copyright/userfiles.html",{'object':object})

import requests

def sendpermission(request):

if request.method == 'GET':

id = request.GET.get('id')

print(' ID is = ', id)

data=UploadModel.objects.get(id=id)

print(data.file,"%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%")

pd=UploadModel.objects.filter(status='granted')

for x in pd:

print(x.file,"^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^")

if pd:

cosine\_values = []

for x in pd:

# ================================================================

print(data.file,x.file,"===============================================================")

data1 = requests.get("http://localhost:8000/media/"+str(data.file)+"/")

data2 = requests.get("http://localhost:8000/media/"+str(x.file)+"/")

print(data.file,x.file,"=============================================@@@@@@@@@@@@@@@@@@@@@@@@@++++++")

# =============================================================

from nltk.corpus import stopwords

from nltk.tokenize import word\_tokenize

# X = input("Enter first string: ").lower()

# Y = input("Enter second string: ").lower()

X = str(data1.content)

Y = str(data2.content)

# tokenization

X\_list = word\_tokenize(X)

Y\_list = word\_tokenize(Y)

# print(X\_list,Y\_list,"===============================================================")

# sw contains the list of stopwords

sw = stopwords.words('english')

l1 = [];

l2 = []

# remove stop words from the string

X\_set = {w for w in X\_list if not w in sw}

Y\_set = {w for w in Y\_list if not w in sw}

# form a set containing keywords of both strings

rvector = X\_set.union(Y\_set)

for w in rvector:

if w in X\_set:

l1.append(1) # create a vector

else:

l1.append(0)

if w in Y\_set:

l2.append(1)

else:

l2.append(0)

c = 0

# cosine formula

for i in range(len(rvector)):

c += l1[i] \* l2[i]

cosine = c / float((sum(l1) \* sum(l2)) \*\* 0.5)

cosine\_values.append(cosine)

print(cosine\_values, "+++++++++++++++++++++++++++++++++++++++++++")

total = 0

for x in cosine\_values:

total += float(x)

avg = (total / len(cosine\_values))

print(avg, "@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@")

if avg >= 0.1:

#return render(request, 'user\_home.html',

# {"patent": "This ML Idea is Patented by someone else..try with your own idea..!"})

print("ml idea existed")

UploadModel.objects.filter(id=id).update(status='Existed')

else:

#UploadModel.objects.create(file=data.file)

qs = UploadModel.objects.filter(id=id)

print("sucessfully Patented,==========================================")

UploadModel.objects.filter(id=id).update(status='granted')

# qs.update(patent\_status="Already Patended")

#return render(request, 'user\_home.html', {"patent": "sucessfully Patented"})

else:

#UploadModel.objects.create(file=data.file)

qs = UploadModel.objects.filter(id=id)

#qs.update(patent\_status="Already Patended")

print("sucessfully Patented")

UploadModel.objects.filter(id=id).update(status='granted')

#return render(request, 'user\_home.html', {"patent": "sucessfully Patented"})

#UploadModel.objects.filter(id=id).update(status='granted')

object = UploadModel.objects.all()

return render(request, 'copyright/userfiles.html', {'object': object})

def sendreject(request):

if request.method == 'GET':

id = request.GET.get('id')

print(' ID is = ', id)

UploadModel.objects.filter(id=id).delete()

object = UploadModel.objects.all()

return render(request, 'copyright/userfiles.html', {'object': object})

base.html

{% load static %}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>PatentLegislation</title>

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

<meta content="" name="keywords">

<meta content="" name="description">

<!-- Favicons -->

<link href="{% static 'img/apple-touch-icon.png' %}" rel="apple-touch-icon">

<!-- Google Fonts -->

<link href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,700,700i|Raleway:300,400,500,700,800|Montserrat:300,400,700" rel="stylesheet">

<!-- Bootstrap CSS File -->

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

<!-- Libraries CSS Files -->

<link href="{% static 'lib/font-awesome/css/font-awesome.min.css' %}" rel="stylesheet">

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

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

<link href="{% static 'lib/owlcarousel/assets/owl.carousel.min.css' %}" rel="stylesheet">

<link href="{% static 'lib/magnific-popup/magnific-popup.css' %}" rel="stylesheet">

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

<!-- Main Stylesheet File -->

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

<!-- =======================================================

Theme Name: Reveal

Theme URL: https://bootstrapmade.com/reveal-bootstrap-corporate-template/

Author: BootstrapMade.com

License: https://bootstrapmade.com/license/

======================================================= -->

</head>

<body id="body">

<!--==========================

Top Bar

============================-->

<!--<section id="topbar" class="d-none d-lg-block">

<div class="container clearfix">

<div class="contact-info float-left">

<i class="fa fa-envelope-o"></i> <a href="mailto:contact@example.com">contact@example.com</a>

<i class="fa fa-phone"></i> +1 5589 55488 55

</div>

<div class="social-links float-right">

<a href="#" class="twitter"><i class="fa fa-twitter"></i></a>

<a href="#" class="facebook"><i class="fa fa-facebook"></i></a>

<a href="#" class="instagram"><i class="fa fa-instagram"></i></a>

<a href="#" class="google-plus"><i class="fa fa-google-plus"></i></a>

<a href="#" class="linkedin"><i class="fa fa-linkedin"></i></a>

</div>

</div>

</section>-->

<!--==========================

Header

============================-->

<header id="header">

<div class="container">

<div id="logo" class="pull-left">

<h1><a href="#body" class="scrollto">Patent<span>Legislation</span></a></h1>

<!-- Uncomment below if you prefer to use an image logo -->

<!-- <a href="#body"><img src="img/logo.png" alt="" title="" /></a>-->

</div>

<nav id="nav-menu-container">

<ul class="nav-menu">

<li class="menu-active"><a href="{% url 'index' %}">HOME</a></li>

<li><a href="{% url 'userlogin' %}">USERS</a></li>

<li><a href="{% url 'copyrightadministered' %}">ADMINISTERED</a></li>

<li><a href="{% url 'adminlogin' %}">ADMIN</a></li>

</ul>

</nav><!-- #nav-menu-container -->

</div>

</header><!-- #header -->

<!--==========================

Intro Section

============================-->

<!-- #intro -->

{% block contents %}

{% endblock %}

<!--==========================

Footer

============================-->

<!--<footer id="footer">

<div class="container">

<div class="copyright">

&copy; Copyright <strong>@lexa</strong>. All Rights Reserved

</div>

<div class="credits">

All the links in the footer should remain intact.

You can delete the links only if you purchased the pro version.

Licensing information: https://bootstrapmade.com/license/

Purchase the pro version with working PHP/AJAX contact form: https://bootstrapmade.com/buy/?theme=Reveal

</div>

</div>

</footer>--><!-- #footer -->

<a href="#" class="back-to-top"><i class="fa fa-chevron-up"></i></a>

<!-- JavaScript Libraries -->

<script src="{% static 'lib/jquery/jquery.min.js' %}"></script>

<script src="{% static 'lib/jquery/jquery-migrate.min.js' %}"></script>

<script src="{% static 'lib/bootstrap/js/bootstrap.bundle.min.js' %}"></script>

<script src="{% static 'lib/easing/easing.min.js' %}"></script>

<script src="{% static 'lib/superfish/hoverIntent.js' %}"></script>

<script src="{% static 'lib/superfish/superfish.min.js' %}"></script>

<script src="{% static 'lib/wow/wow.min.js' %}"></script>

<script src="{% static 'lib/owlcarousel/owl.carousel.min.js' %}"></script>

<script src="{% static 'lib/magnific-popup/magnific-popup.min.js' %}"></script>

<script src="{% static 'lib/sticky/sticky.js' %}"></script>

<!-- Contact Form JavaScript File -->

<script src="{% static 'contactform/contactform.js' %}"></script>

<!-- Template Main Javascript File -->

<script src="{% static 'js/main.js' %}"></script>

</body>

</html>

Index.html

{% extends 'base.html' %}

{% load static %}

{% block contents %}

<section id="intro">

<div class="intro-content">

<h2>The Influence of Artificial Intelligence <span>Development</span><br>on Patent Legislation</h2>

<!--<div>

<a href="#about" class="btn-get-started scrollto">Get Started</a>

<a href="#portfolio" class="btn-projects scrollto">Our Projects</a>

</div>-->

</div>

<div id="intro-carousel" class="owl-carousel" >

<div class="item" style="background-image: url('img/intro-carousel/1.jpg');"></div>

<div class="item" style="background-image: url('img/intro-carousel/2.jpg');"></div>

<div class="item" style="background-image: url('img/intro-carousel/3.jpg');"></div>

<div class="item" style="background-image: url('img/intro-carousel/4.jpg');"></div>

<div class="item" style="background-image: url('img/intro-carousel/5.jpg');"></div>

</div>

</section>

{% endblock %}

Userfiles.html

{% extends 'copyrightbase.html' %}

{% load static %}

{% block contents %}

<section id="intro">

<div class="intro-content">

<h2>View <span>users</span><br>details</h2>

<center><table border="2px solid blue" align="center">

<tr><th style="color:red">email</th>

<th style="color:red">category</th>

<th style="color:red">title</th>

<th style="color:red">description</th>

<th style="color:red">file</th>

<th style="color:red">Verification</th>

<th style="color:red">permission</th>

<th style="color:red">reject</th>

</tr>

{% for x in object %}

<tr style="color:black">

<td>{{x.email}}</td>

<td>{{x.category}}</td>

<td>{{x.title}}</td>

<td>{{x.description}}</td>

<td>{{x.file}}</td>

<td style="color:brown"><a href="{{x.file.url }}" class="btn btn-primary btn-sm" target="\_blank">verification</a></td>

{% if x.status == 'waiting' %}

<td> <a href="/sendpermission/?id={{ x.id }}&email={{ x.email }}" >permission</a></td>

{% elif x.status == 'Existed' %}

<td style="color:Skyblue">Existed</td>

{% else %}

<td style="color:Skyblue">Granted</td>

{% endif %}

{% if x.status == 'waiting' or x.status == 'Existed'%}

<td> <a href="/sendreject/?id={{ x.id }}&email={{ x.email }}" >Delete</a></td>

{% endif %}

</tr>

{% endfor %}

</table></center>

</div>

<div id="intro-carousel" class="owl-carousel" >

<div class="item" style="background-image: url('img/intro-carousel/1.jpg');"></div>

<div class="item" style="background-image: url('img/intro-carousel/2.jpg');"></div>

<div class="item" style="background-image: url('img/intro-carousel/3.jpg');"></div>

<div class="item" style="background-image: url('img/intro-carousel/4.jpg');"></div>

<div class="item" style="background-image: url('img/intro-carousel/5.jpg');"></div>

</div>

</section>

{% endblock %}

Activate users:

{% extends 'adminbase.html' %}

{% load static %}

{% block contents %}

<section id="intro">

<div class="intro-content">

<h2>View <span>users</span><br>details</h2>

<center><table border="2px solid blue" align="center">

<tr><th style="color:red">First Name</th>

<th style="color:red">Last Name</th>

<th style="color:red">Email</th>

<th style="color:red">Password</th>

<th style="color:red">Mobile</th>

<th style="color:red">qualification</th>

<th style="color:red">City</th>

<th style="color:red">status</th>

<th style="color:red">activate</th>

</tr>

{% for x in object %}

<tr style="color:black">

<td>{{x.firstname}}</td>

<td>{{x.lastname}}</td>

<td>{{x.email}}</td>

<td>{{x.passwd}}</td>

<td>{{x.mobileno}}</td>

<td>{{x.qualification}}</td>

<td>{{x.city}}</td>

<td>{{x.status}}</td>

{% if x.status == 'waiting' %}

<td> <a href="/activateuser/?pid={{ x.id }}" >Activate</a></td>

{% else %}

<td> Activated</td>

{% endif %}

</tr>

{% endfor %}

</table></center>

</div>

<div id="intro-carousel" class="owl-carousel" >

<div class="item" style="background-image: url('img/intro-carousel/1.jpg');"></div>

<div class="item" style="background-image: url('img/intro-carousel/2.jpg');"></div>

<div class="item" style="background-image: url('img/intro-carousel/3.jpg');"></div>

<div class="item" style="background-image: url('img/intro-carousel/4.jpg');"></div>

<div class="item" style="background-image: url('img/intro-carousel/5.jpg');"></div>

</div>

</section>

{% endblock %}