I am working on a django project I want your help:

This is the Project Structure :  
testwise/

├── media/

│ └── pdfs/

│ ├── abhishek\_madaan\_cc5.docx

│ ├── Alok\_Singh\_cc6\_dn3cAJL.docx

│ ├── Alok\_Singh\_cc6\_Pn77YiA.docx

│ ├── Alok\_Singh\_cc6.docx

│ ├── CC\_lab\_1.docx

│ ├── FUNDAMENTALS\_OF\_PSYCHOLOGY.docx

│ ├── FUNDAMENTALS\_OF\_PSYCHOLOGY.docx (duplicate?)

│ ├── HP\_Police\_Constable\_notification.pdf

│ └── Software\_engineering\_notes.pdf

├── project/

│ ├── \_\_pycache\_\_/

│ ├── \_\_init\_\_.py

│ ├── asgi.py

│ ├── settings.py

│ ├── urls.py

│ └── wsgi.py

├── static/

│ └── styles/

├── testwise\_main/

│ ├── \_\_pycache\_\_/

│ ├── migrations/

│ ├── templates/

│ │ └── testwise\_main/

│ │ ├── login.html

│ │ ├── pdf\_list.html

│ │ ├── signup.html

│ │ └── verify.html

│ ├── \_\_init\_\_.py

│ ├── admin.py

│ ├── apps.py

│ ├── forms.py

│ ├── models.py

│ ├── tests.py

│ ├── urls.py

│ └── views.py

├── db.sqlite3

└── manage.py

This is settings.py:

"""

Django settings for project project.

Generated by 'django-admin startproject' using Django 5.0.6.

For more information on this file, see

https://docs.djangoproject.com/en/5.0/topics/settings/

For the full list of settings and their values, see

https://docs.djangoproject.com/en/5.0/ref/settings/

"""

from pathlib import Path

import os

*# Build paths inside the project like this: BASE\_DIR / 'subdir'.*

BASE\_DIR = Path(\_\_file\_\_).resolve().parent.parent

*# Media files*

MEDIA\_URL = '/media/'

MEDIA\_ROOT = os.path.join(BASE\_DIR, 'media')

*# 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-yy^-xpe\_i=)lkwjtk5wh(@d)c!k0tb-qhvn&hzhz(mg=tdxrrg'

*# 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',

    'testwise\_main',

]

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 = 'project.urls'

TEMPLATES = [

    {

        'BACKEND': 'django.template.backends.django.DjangoTemplates',

        'DIRS': [os.path.join(BASE\_DIR, 'templates')],

        '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 = 'project.wsgi.application'

*# Database*

*# https://docs.djangoproject.com/en/5.0/ref/settings/#databases*

DATABASES = {

    'default': {

        'ENGINE': 'django.db.backends.sqlite3',

        'NAME': BASE\_DIR / 'db.sqlite3',

    }

}

*# 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 = [

    BASE\_DIR / "static",

]

*# Default primary key field type*

*# https://docs.djangoproject.com/en/5.0/ref/settings/#default-auto-field*

DEFAULT\_AUTO\_FIELD = 'django.db.models.BigAutoField'

EMAIL\_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'

EMAIL\_HOST = 'smtp.gmail.com'

EMAIL\_PORT = 587

EMAIL\_HOST\_USER = "dhruvtemura@gmail.com"

EMAIL\_HOST\_PASSWORD = "akin kwux tjjj ohyn"

EMAIL\_USE\_TLS = True

EMAIL\_USE\_SSL = False

DEFAULT\_FROM\_EMAIL = EMAIL\_HOST\_USER

This is the urls.py(project folder):

"""

URL configuration for project project.

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

    https://docs.djangoproject.com/en/5.0/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, include

from django.conf import settings

from django.conf.urls.static import static

urlpatterns = [

    path('admin/', admin.site.urls),

    path('',include('testwise\_main.urls'))

]

if settings.DEBUG:

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

For the App:

The admin.py:  
from django.contrib import admin

from .models import PDF

@admin.register(PDF)

class PDFAdmin(admin.ModelAdmin):

    list\_display = ('title', 'uploaded\_at')

    search\_fields = ('title',)

The Forms.py:

*# pdfapp/forms.py*

from django import forms

from .models import PDF

class PDFForm(forms.ModelForm):

    class Meta:

        model = PDF

        fields = ['title', 'pdf\_file']

        widgets = {

            'title': forms.TextInput(attrs={'class': 'form-control', 'placeholder': 'Enter PDF title'}),

            'pdf\_file': forms.FileInput(attrs={'class': 'form-control'}),

        }

The Models.py:

from django.db import models

from django.contrib.auth.models import User

*# Create your models here.*

class PDF(models.Model):

    user =  models.ForeignKey(User, on\_delete=models.CASCADE)

    title = models.CharField(max\_length=255)

    pdf\_file = models.FileField(upload\_to='pdfs/')

    uploaded\_at = models.DateTimeField(auto\_now\_add=True)

    def \_\_str\_\_(self):

        return *self*.title

The urls.py(app):

from django.urls import path

from . import views

urlpatterns = [

    path('pdf\_list/', views.pdf\_list, name='pdf\_list'),

    path('upload/', views.upload\_pdf, name='upload\_pdf'),

    path('delete/<int:pk>/', views.delete\_pdf, name='delete\_pdf'),

    path('', views.SignUp, name='signup'),

    path('login/',views.Login,name='login'),

    path('verify\_email/', views.verify\_email, name='verify\_email'),

]

The views.py :

from django.shortcuts import render, redirect, get\_object\_or\_404, HttpResponse

from .models import PDF

from .forms import PDFForm

from django.contrib import messages

from django.contrib.auth.models import User

from django.contrib.auth import authenticate,login

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

from django.core.mail import send\_mail

from project.settings import EMAIL\_HOST\_USER

import random

from django.http import JsonResponse

from django.views.decorators.csrf import csrf\_exempt

@login\_required(login\_url='login')

def pdf\_list(request):

    pdfs = PDF.objects.filter(user=request.user).order\_by('-uploaded\_at')

    form = PDFForm()

    return render(request, 'testwise\_main/pdf\_list.html', {'pdfs': pdfs, 'form': form})

def upload\_pdf(request):

    if request.method == 'POST':

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

        if form.is\_valid():

            PDF = form.save(commit=False)

            PDF.user = request.user

            PDF.save()

            messages.success(request, 'PDF uploaded successfully.')

        else:

            messages.error(request, 'Failed to upload PDF. Please ensure the file is a PDF.')

    else :

        form = PDFForm

    return redirect('pdf\_list')

def delete\_pdf(request, pk):

    pdf = get\_object\_or\_404(PDF, pk=pk)

    if request.method == 'POST':

        pdf.pdf\_file.delete()

        pdf.delete()

        messages.success(request, 'PDF deleted successfully.')

    return redirect('pdf\_list')

def SignUp(request):

    if request.method == 'POST':

        username = request.POST.get('username')

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

        pass1 = request.POST.get('password1')

        pass2 = request.POST.get('password2')

        if pass1 != pass2:

            messages.error(request, "Passwords don't match!")

            return render(request, 'testwise\_main/signup.html')

        try:

            my\_user = User.objects.create\_user(username, email, pass1)

            my\_user.is\_active = False

            my\_user.save()

            otp = random.randint(1000, 9999)

            request.session['signup\_otp'] = otp

            request.session['user\_email'] = email

            try:

                send\_mail(

                    'Email Verification OTP',

                    f'Your verification OTP is: {otp}',EMAIL\_HOST\_USER,[email],fail\_silently=False,

                )

                print(f"OTP {otp} sent to {email}")

                return render(request, 'testwise\_main/verify.html', {'email': email})

            except Exception as e:

                print(f"Email error: {str(e)}")

                messages.error(request, "Error sending email. Please try again.")

                my\_user.delete()

                return render(request, 'testwise\_main/signup.html')

        except Exception as e:

            print(f"User creation error: {str(e)}")

            messages.error(request, "Error creating user. Please try again.")

            return render(request, 'testwise\_main/signup.html')

    return render(request, 'testwise\_main/signup.html')

def verify\_email(request):

    if request.method == 'POST':

        user\_otp = request.POST.get('otp')

        stored\_otp = request.session.get('signup\_otp')

        user\_email = request.session.get('user\_email')

        if stored\_otp and str(stored\_otp) == str(user\_otp):

            user = User.objects.get(email=user\_email)

            user.is\_active = True

            user.save()

            del request.session['signup\_otp']

            del request.session['user\_email']

            messages.success(request, "Email verified successfully! Please login.")

            return redirect('login')

        else:

            messages.error(request, "Invalid OTP. Please try again.")

            return render(request, 'testwise\_main/verify.html', {'email': user\_email})

    return redirect('signup')

def Login(request):

    if request.method=='POST':

        username=request.POST.get('username')

        pass1=request.POST.get('pass')

        user=authenticate(request,username=username,password=pass1)

        if user is not None:

            login(request,user)

            return redirect('pdf\_list')

        else:

            return HttpResponse ('Username or Password is incorrect!')

    return render (request,'testwise\_main/login.html')

These are the HTML files :

<!DOCTYPE html>

<html lang="en">

<head>

*<!-- Design by foolishdeveloper.com -->*

    <title>Login Page</title>

    <link rel="preconnect" href="https://fonts.gstatic.com">

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css">

    <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;500;600&display=swap" rel="stylesheet">

*<!--Stylesheet-->*

    <style media="screen">

      \*,

\*:before,

\*:after{

    padding: 0;

    margin: 0;

    box-sizing: border-box;

}

body{

    background-color: #080710;

}

.background{

    width: 430px;

    height: 520px;

    position: absolute;

    transform: translate(-50%,-50%);

    left: 50%;

    top: 50%;

}

.background .shape{

    height: 200px;

    width: 200px;

    position: absolute;

    border-radius: 50%;

}

.shape:first-child{

    background: linear-gradient(

        #1845ad,

        #23a2f6

    );

    left: -80px;

    top: -80px;

}

.shape:last-child{

    background: linear-gradient(

        to right,

        #ff512f,

        #f09819

    );

    right: -30px;

    bottom: -80px;

}

form{

    height: 520px;

    width: 400px;

    background-color: rgba(255,255,255,0.13);

    position: absolute;

    transform: translate(-50%,-50%);

    top: 50%;

    left: 50%;

    border-radius: 10px;

    backdrop-filter: blur(10px);

    border: 2px solid rgba(255,255,255,0.1);

    box-shadow: 0 0 40px rgba(8,7,16,0.6);

    padding: 50px 35px;

}

form \*{

    font-family: 'Poppins',sans-serif;

    color: #ffffff;

    letter-spacing: 0.5px;

    outline: none;

    border: none;

}

form h3{

    font-size: 32px;

    font-weight: 500;

    line-height: 42px;

    text-align: center;

}

label{

    display: block;

    margin-top: 30px;

    font-size: 16px;

    font-weight: 500;

}

input{

    display: block;

    height: 50px;

    width: 100%;

    background-color: rgba(255,255,255,0.07);

    border-radius: 3px;

    padding: 0 10px;

    margin-top: 8px;

    font-size: 14px;

    font-weight: 300;

}

::placeholder{

    color: #e5e5e5;

}

button{

    margin-top: 50px;

    width: 100%;

    background-color: #ffffff;

    color: #080710;

    padding: 15px 0;

    font-size: 18px;

    font-weight: 600;

    border-radius: 5px;

    cursor: pointer;

}

.social{

  margin-top: 30px;

  display: flex;

}

.social div{

  background: red;

  width: 150px;

  border-radius: 3px;

  padding: 5px 10px 10px 5px;

  background-color: rgba(255,255,255,0.27);

  color: #eaf0fb;

  text-align: center;

}

.social div:hover{

  background-color: rgba(255,255,255,0.47);

}

.social .fb{

  margin-left: 25px;

}

.social i{

  margin-right: 4px;

}

    </style>

</head>

<body>

    <div class="background">

        <div class="shape"></div>

        <div class="shape"></div>

    </div>

    <form method="post">

        <h3>Login Here</h3>

        {% csrf\_token %}

        <label for="username">Username</label>

        <input type="text" placeholder="Enter Username" id="username" name="username">

        <label for="password">Password</label>

        <input type="password" placeholder="Password" id="password" name="pass">

        <button type="submit">Log In</button>

*<!-- <input type="button" value=""> -->*

*<!-- <div class="social">*

*<div class="go"><i class="fab fa-google"></i>  Google</div>*

*<div class="fb"><i class="fab fa-facebook"></i>  Facebook</div>*

*</div> -->*

        <a href="{% url 'signup' %}" >Create a account</a>

    </form>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>PDF Manager</title>

*<!-- Bootstrap CSS -->*

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">

</head>

<body>

<div class="container mt-5">

    <h1 class="mb-4">PDF Manager</h1>

*<!-- Display Messages -->*

    {% if messages %}

        {% for message in messages %}

            <div class="alert alert-{{ message.tags }} alert-dismissible fade show" role="alert">

                {{ message }}

                <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>

            </div>

        {% endfor %}

    {% endif %}

*<!-- Upload Form -->*

    <div class="card mb-4">

        <div class="card-header">

            Upload a New PDF

        </div>

        <div class="card-body">

            <form method="post" action="{% url 'upload\_pdf' %}" enctype="multipart/form-data">

                {% csrf\_token %}

                <div class="mb-3">

                    {{ form.title.label\_tag }}

                    {{ form.title }}

                </div>

                <div class="mb-3">

                    {{ form.pdf\_file.label\_tag }}

                    {{ form.pdf\_file }}

                </div>

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

            </form>

        </div>

    </div>

*<!-- PDF List -->*

    <h2>Uploaded PDFs</h2>

    {% if pdfs %}

        <table class="table table-striped">

            <thead>

                <tr>

                    <th>Title</th>

                    <th>Uploaded At</th>

                    <th>Actions</th>

                </tr>

            </thead>

            <tbody>

                {% for pdf in pdfs %}

                    <tr>

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

                        <td>{{ pdf.uploaded\_at|date:"Y-m-d H:i" }}</td>

                        <td>

                            <a href="{{ pdf.pdf\_file.url }}" class="btn btn-sm btn-success" target="\_blank">View</a>

                            <form action="{% url 'delete\_pdf' pdf.pk %}" method="post" style="display:inline;">

                                {% csrf\_token %}

                                <button type="submit" class="btn btn-sm btn-danger" onclick="return confirm('Are you sure you want to delete this PDF?');">Delete</button>

                            </form>

                        </td>

                    </tr>

                {% endfor %}

            </tbody>

        </table>

    {% else %}

        <p>No PDFs uploaded yet.</p>

    {% endif %}

</div>

*<!-- Bootstrap JS and dependencies -->*

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

    <title>Signup Page</title>

    <link rel="preconnect" href="https://fonts.gstatic.com">

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css">

    <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;500;600&display=swap" rel="stylesheet">

    <style media="screen">

      \*,

\*:before,

\*:after{

    padding: 0;

    margin: 0;

    box-sizing: border-box;

}

body{

    background-color: #080710;

}

.background{

    width: 430px;

    height: 520px;

    position: absolute;

    transform: translate(-50%,-50%);

    left: 50%;

    top: 50%;

}

.background .shape{

    height: 200px;

    width: 200px;

    position: absolute;

    border-radius: 50%;

}

.shape:first-child{

    background: linear-gradient(

        #1845ad,

        #23a2f6

    );

    left: -80px;

    top: -80px;

}

.shape:last-child{

    background: linear-gradient(

        to right,

        #ff512f,

        #f09819

    );

    right: -30px;

    bottom: -80px;

}

form{

    height: 600px;

    width: 450px;

    background-color: rgba(255,255,255,0.13);

    position: absolute;

    transform: translate(-50%,-50%);

    top: 50%;

    left: 50%;

    border-radius: 10px;

    backdrop-filter: blur(10px);

    border: 2px solid rgba(255,255,255,0.1);

    box-shadow: 0 0 40px rgba(8,7,16,0.6);

    padding: 50px 35px;

}

form \*{

    font-family: 'Poppins',sans-serif;

    color: #ffffff;

    letter-spacing: 0.5px;

    outline: none;

    border: none;

}

form h3{

    font-size: 32px;

    font-weight: 500;

    line-height: 42px;

    text-align: center;

}

label{

    display: block;

    margin-top: 30px;

    font-size: 16px;

    font-weight: 500;

}

input{

    display: block;

    height: 40px;

    width: 100%;

    background-color: rgba(255,255,255,0.07);

    border-radius: 3px;

    padding: 0 10px;

    margin-top: 2px;

    font-size: 14px;

    font-weight: 300;

}

::placeholder{

    color: #e5e5e5;

}

button{

    margin-top: 14px;

    width: 100%;

    background-color: #ffffff;

    color: #080710;

    padding: 15px 0;

    font-size: 18px;

    font-weight: 600;

    border-radius: 5px;

    cursor: pointer;

}

.social{

  margin-top: 30px;

  display: flex;

}

.social div{

  background: red;

  width: 150px;

  border-radius: 3px;

  padding: 5px 10px 10px 5px;

  background-color: rgba(255,255,255,0.27);

  color: #eaf0fb;

  text-align: center;

}

.social div:hover{

  background-color: rgba(255,255,255,0.47);

}

.social .fb{

  margin-left: 25px;

}

.social i{

  margin-right: 4px;

}

    </style>

</head>

<body>

    <div class="background">

        <div class="shape"></div>

        <div class="shape"></div>

    </div>

    <form action="" method="post">

        {% csrf\_token %}

        <h3>Signup Here</h3>

        <label for="username">Username</label>

        <input type="text" placeholder="Username" name="username" id="username">

        <label for="email">Email</label>

        <input type="email" placeholder="Email" name="email" id="email">

        <label for="password1">Password</label>

        <input type="password" placeholder="Password" id="password1" name="password1">

        <label for="password2">Confirm Password</label>

        <input type="password" placeholder="Confirm Password" id="password2" name="password2">

        <button type="submit">Signup</button>

        <a href="{% url 'login' %}" >i have already account</a>

    </form>

</body>

</html>

 <head>

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.1/jquery.min.js"></script>

 <head>

<form method="POST" action="{% url 'verify\_email' %}">

    {% csrf\_token %}

    <h2>Verify Your Email</h2>

    <p>We sent an OTP to {{ email }}</p>

    <input type="text" name="otp" id="otp" placeholder="Enter OTP" required>

    <button type="submit">Verify</button>

</form>