i guess, i better give u all the codes, so u can fix it. main issue is i want to add task with all the 6 components and one attachment, after hitting save (there needs to be file attachment button on add task), after hitting save it should go to tasks page, where it will show. also the 3 sorting option on all taks, u should fill them up with given sources incidents etc. now i am giving u codes: from main project folder: settings.py:"""

Django settings for a2i\_task\_management project.

Generated by 'django-admin startproject' using Django 5.1.1.

For more information on this file, see

https://docs.djangoproject.com/en/5.1/topics/settings/

For the full list of settings and their values, see

https://docs.djangoproject.com/en/5.1/ref/settings/

"""

from pathlib import Path

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

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

# Quick-start development settings - unsuitable for production

# See https://docs.djangoproject.com/en/5.1/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!

SECRET\_KEY = 'django-insecure-@hlrm@vvh2d&6hov4+zh7\_huy#$c9y+mwe7yzvb-lbea0v5c@\*'

# 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',

'tasks',

]

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 = 'a2i\_task\_management.urls'

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',

],

},

},

]

WSGI\_APPLICATION = 'a2i\_task\_management.wsgi.application'

# Database

# https://docs.djangoproject.com/en/5.1/ref/settings/#databases

DATABASES = {

'default': {

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

'NAME': 'a2i\_management\_db', # The correct database name

'USER': 'root', # Your MySQL username (likely 'root')

'PASSWORD': 'Diptowifi420', # Replace with your MySQL password

'HOST': 'localhost',

'PORT': '3306', # Default MySQL port, or '3307' if using phpMyAdmin

}

}

# Password validation

# https://docs.djangoproject.com/en/5.1/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',

},

]

# Authentication backends

# Add this section to explicitly define the authentication backends

AUTHENTICATION\_BACKENDS = [

'django.contrib.auth.backends.ModelBackend',

]

# Internationalization

# https://docs.djangoproject.com/en/5.1/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/5.1/howto/static-files/

STATIC\_URL = 'static/'

# Default primary key field type

# https://docs.djangoproject.com/en/5.1/ref/settings/#default-auto-field

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

urls.py :from django.contrib import admin

from django.urls import path, include

from tasks import views # Import the views from the tasks app

urlpatterns = [

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

path('', views.home, name='home'), # Redirect the root URL to the home view (which redirects to the dashboard)

path('tasks/', include('tasks.urls')), # Include the tasks app URLs

path('login/', views.user\_login, name='login'), # URL for user login

path('dashboard/', views.dashboard, name='dashboard'), # URL for the dashboard

]

from tasks folder: forms.py: from django import forms

from .models import Task, TaskAttachment

class TaskForm(forms.ModelForm):

class Meta:

model = Task

fields = ['title', 'description', 'status', 'source', 'incident', 'support\_category', 'office\_name']

widgets = {

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

'description': forms.Textarea(attrs={'class': 'form-control'}),

'status': forms.Select(attrs={'class': 'form-control'}),

'source': forms.Select(attrs={'class': 'form-control'}),

'incident': forms.Select(attrs={'class': 'form-control'}),

'support\_category': forms.Select(attrs={'class': 'form-control'}),

'office\_name': forms.TextInput(attrs={'class': 'form-control'}),

}

class TaskAttachmentForm(forms.ModelForm):

class Meta:

model = TaskAttachment

fields = ['attachment']

widgets = {

'attachment': forms.ClearableFileInput(attrs={'multiple': True}),

}

models.py: from django.db import models

# Predefined choices for Task fields

SOURCE\_CHOICES = [

('Phone Call', 'Phone Call'),

('Personal Email', 'Personal Email'),

('OTRS', 'OTRS'),

('SMS', 'SMS'),

('One 2 One', 'One 2 One'),

('Training', 'Training'),

('i-Top', 'i-Top'),

('AnyDesk', 'AnyDesk'),

('WhatsApp', 'WhatsApp'),

('Social Platforms', 'Social Platforms'),

('Productivity / Creativity', 'Productivity / Creativity'),

('Team Work', 'Team Work'),

('Personal', 'Personal'),

('Meeting', 'Meeting'),

('Nothi System', 'Nothi System'),

('Self Initiative', 'Self Initiative'),

('Super Admin', 'Super Admin'),

('Others', 'Others'),

('All', 'All'),

]

SUPPORT\_CATEGORY\_CHOICES = [

('User Management', 'User Management'),

('Organogram Management', 'Organogram Management'),

('Training Management', 'Training Management'),

('Incident Management', 'Incident Management'),

('Knowledge Sharing', 'Knowledge Sharing'),

('System Management', 'System Management'),

('Doc/Exl/Sheet Management', 'Doc/Exl/Sheet Management'),

('Productivity / Creativity', 'Productivity / Creativity'),

('Nothi / Note / File Management', 'Nothi / Note / File Management'),

('Team Management', 'Team Management'),

('Tech / Vendor Management', 'Tech / Vendor Management'),

('Migration Management', 'Migration Management'),

('myGov Management', 'myGov Management'),

('Mobile Application', 'Mobile Application'),

('Implementation', 'Implementation'),

('Query Management', 'Query Management'),

('Reporting', 'Reporting'),

('System Testing', 'System Testing'),

('Capacity Development', 'Capacity Development'),

('Tutorial Management', 'Tutorial Management'),

('Others', 'Others'),

('Office Management', 'Office Management'),

('Logistics / Apparatus Management', 'Logistics / Apparatus Management'),

('Request Management', 'Request Management'),

('All', 'All'),

]

INCIDENT\_CHOICES = [

('Potrojari', 'Potrojari'),

('Slowness', 'Slowness'),

('Note Pending', 'Note Pending'),

('Synchronize', 'Synchronize'),

('Template', 'Template'),

('Password', 'Password'),

('User Information', 'User Information'),

('OTP', 'OTP'),

('PDF Generation Error', 'PDF Generation Error'),

('Dak Upload', 'Dak Upload'),

('Office/Branch Information Correction', 'Office/Branch Information Correction'),

('Dak Forward', 'Dak Forward'),

('Nothi Permission', 'Nothi Permission'),

('Nothi Information Correction', 'Nothi Information Correction'),

('Archive', 'Archive'),

('Nothi Forward', 'Nothi Forward'),

('Dashboard', 'Dashboard'),

('Heading Correction', 'Heading Correction'),

('Substitute', 'Substitute'),

('Signature', 'Signature'),

('Doptor', 'Doptor'),

('Designation Align', 'Designation Align'),

('User Assign', 'User Assign'),

('User Release', 'User Release'),

('Dak Not Found', 'Dak Not Found'),

('Nothi Not Found', 'Nothi Not Found'),

('PDF Blank', 'PDF Blank'),

('Others', 'Others'),

]

STATUS\_CHOICES = [

('Dependency', 'Dependency'),

('Pending', 'Pending'),

('Done', 'Done'),

('Working', 'Working'),

('i-Top', 'i-Top'),

('Others', 'Others'),

]

class Task(models.Model):

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

description = models.TextField()

status = models.CharField(max\_length=50, choices=STATUS\_CHOICES)

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

source = models.CharField(max\_length=100, choices=SOURCE\_CHOICES)

incident = models.CharField(max\_length=100, choices=INCIDENT\_CHOICES)

support\_category = models.CharField(max\_length=100, choices=SUPPORT\_CATEGORY\_CHOICES)

office\_name = models.CharField(max\_length=255)

class TaskAttachment(models.Model):

task = models.ForeignKey(Task, on\_delete=models.CASCADE, related\_name='attachments')

attachment = models.FileField(upload\_to='uploads/')

urls.py: from django.urls import path

from . import views

urlpatterns = [

path('login/', views.user\_login, name='login'), # Login view

path('dashboard/', views.dashboard, name='dashboard'), # Dashboard view

path('tasks/', views.task\_list, name='task\_list'), # Task list view

path('tasks/add/', views.add\_task, name='add\_task'), # Add task view

path('tasks/<int:task\_id>/edit/', views.edit\_task, name='edit\_task'), # Edit task view

path('tasks/<int:task\_id>/delete/', views.delete\_task, name='delete\_task'), # Delete task view

]

views.py: from django.shortcuts import render, redirect, get\_object\_or\_404

from django.contrib.auth import authenticate, login

from django.contrib.auth.forms import AuthenticationForm

from django.shortcuts import render, redirect

from .models import Task, TaskAttachment

from .forms import TaskForm, TaskAttachmentForm

from django.shortcuts import redirect

from django.shortcuts import render

from django.utils.timezone import now

def user\_login(request):

if request.user.is\_authenticated:

return redirect('dashboard') # Redirect to the dashboard if the user is already logged in

if request.method == 'POST':

form = AuthenticationForm(request, data=request.POST)

if form.is\_valid():

username = form.cleaned\_data.get('username')

password = form.cleaned\_data.get('password')

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

if user is not None:

login(request, user)

return redirect('dashboard')

else:

form = AuthenticationForm()

return render(request, 'tasks/login.html', {'form': form})

def dashboard(request):

# You can customize this view as needed

current\_date = now().strftime('%Y-%m-%d') # Get the current date

return render(request, 'tasks/dashboard.html', {'current\_date': current\_date})

# View to display the list of tasks with filtering options

def task\_list(request):

tasks = Task.objects.all()

# Apply filtering based on sorting options if provided

source\_filter = request.GET.get('source')

category\_filter = request.GET.get('support\_category')

incident\_filter = request.GET.get('incident')

if source\_filter and source\_filter != 'All':

tasks = tasks.filter(source=source\_filter)

if category\_filter and category\_filter != 'All':

tasks = tasks.filter(support\_category=category\_filter)

if incident\_filter and incident\_filter != 'All':

tasks = tasks.filter(incident=incident\_filter)

return render(request, 'tasks/task\_list.html', {

'tasks': tasks,

'source\_choices': Task.\_meta.get\_field('source').choices,

'support\_category\_choices': Task.\_meta.get\_field('support\_category').choices,

'incident\_choices': Task.\_meta.get\_field('incident').choices,

})

# View to add a new task with file attachment

def add\_task(request):

if request.method == 'POST':

task\_form = TaskForm(request.POST)

if task\_form.is\_valid():

task = task\_form.save()

if request.FILES:

for f in request.FILES.getlist('attachments'):

TaskAttachment.objects.create(task=task, attachment=f)

return redirect('task\_list')

else:

task\_form = TaskForm()

return render(request, 'tasks/add\_task.html', {

'form': task\_form,

'attachment\_form': TaskAttachmentForm()

})

# View to edit an existing task with file attachment

def edit\_task(request, task\_id):

task = get\_object\_or\_404(Task, id=task\_id)

if request.method == 'POST':

task\_form = TaskForm(request.POST, instance=task)

if task\_form.is\_valid():

task\_form.save()

if request.FILES:

for f in request.FILES.getlist('attachments'):

TaskAttachment.objects.create(task=task, attachment=f)

return redirect('task\_list')

else:

task\_form = TaskForm(instance=task)

return render(request, 'tasks/edit\_task.html', {

'form': task\_form,

'task': task,

})

# View to delete a task

def delete\_task(request, task\_id):

task = get\_object\_or\_404(Task, id=task\_id)

task.delete()

return redirect('task\_list')

# View to handle the home page redirection

def home(request):

return redirect('dashboard') # Redirect to 'dashboard'; use 'task\_list' if you prefer to redirect to the task list page. tasks,templates,partials: sidebar: <div class="sidebar">

<ul class="list-group">

<li class="list-group-item">

<a href="{% url 'dashboard' %}" class="text-decoration-none">Dashboard</a>

</li>

<li class="list-group-item">

<a href="{% url 'task\_list' %}" class="text-decoration-none">Task List</a>

</li>

<li class="list-group-item">

<a href="{% url 'add\_task' %}" class="text-decoration-none">Add Task</a>

</li>

<!-- Add more links here -->

</ul>

</div>

templates,tasks, add\_task.html: {% extends 'base.html' %}

{% block title %}Add Task{% endblock %}

{% block content %}

<div class="container mt-5">

<h1 class="text-center">Add Task</h1>

<form method="post" enctype="multipart/form-data">

{% csrf\_token %}

<div class="form-group">

<label for="source">Source</label>

{{ form.source }}

</div>

<div class="form-group">

<label for="support\_category">Support Category</label>

{{ form.support\_category }}

</div>

<div class="form-group">

<label for="incident">Incident</label>

{{ form.incident }}

</div>

<div class="form-group">

<label for="status">Status</label>

{{ form.status }}

</div>

<div class="form-group">

{{ form.office\_name.label\_tag }}{{ form.office\_name }}

</div>

<div class="form-group">

{{ form.description.label\_tag }}{{ form.description }}

</div>

<div class="form-group">

<label for="attachments">Attachments (PDF/Word):</label>

<input type="file" name="attachments" multiple>

</div>

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

</form>

</div>

{% endblock %}

base.hrtml: <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>{% block title %}Task Manager{% endblock %}</title>

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

</head>

<body>

<nav class="navbar navbar-expand-lg navbar-light bg-light">

<a class="navbar-brand" href="{% url 'dashboard' %}">Task Manager</a>

<div class="collapse navbar-collapse">

<ul class="navbar-nav mr-auto">

<li class="nav-item"><a class="nav-link" href="{% url 'task\_list' %}">All Tasks</a></li>

<li class="nav-item"><a class="nav-link" href="{% url 'add\_task' %}">Add Task</a></li>

</ul>

</div>

</nav>

<div class="container">

{% block content %}{% endblock %}

</div>

</body>

</html>

dashboard.html: <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Dashboard</title>

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

<style>

body {

background-color: #f8f9fa;

}

.sidebar {

height: 100vh;

position: fixed;

top: 0;

left: 0;

width: 220px;

background-color: #343a40;

padding-top: 20px;

}

.sidebar a {

color: #fff;

padding: 15px;

text-decoration: none;

display: block;

}

.sidebar a:hover {

background-color: #495057;

}

.content {

margin-left: 220px;

padding: 20px;

}

.topbar {

display: flex;

justify-content: flex-end;

margin-bottom: 20px;

}

.topbar .current-date {

font-size: 18px;

font-weight: bold;

}

</style>

</head>

<body>

<div class="sidebar">

<h4 class="text-center text-white">Dashboard</h4>

<a href="{% url 'task\_list' %}">All Tasks</a>

<a href="{% url 'add\_task' %}">Add Task</a>

<a href="#">User Privileges</a>

<a href="#">Reports</a>

</div>

<div class="content">

<div class="topbar">

<span class="current-date">{{ current\_date }}</span>

</div>

<h1>Welcome to the Dashboard</h1>

</div>

</body>

</html>

edit task: <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Edit Task</title>

<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="text-center">Edit Task</h1>

<form method="post" enctype="multipart/form-data">

{% csrf\_token %}

{{ form.as\_p }}

<div class="form-group">

<label for="attachments">Attachments (PDF/Word):</label>

<input type="file" name="attachments" multiple class="form-control">

</div>

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

</form>

</div>

</body>

</html>

login.html: <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Login</title>

<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="text-center">Login</h1>

<form method="post">

{% csrf\_token %}

{{ form.as\_p }}

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

</form>

</div>

</body>

</html>

task\_list.html: {% extends 'base.html' %}

{% block title %}All Tasks{% endblock %}

{% block content %}

<div class="container mt-5">

<h1 class="text-center">All Tasks</h1>

<form method="get" class="mb-3">

<div class="form-group">

<label for="source">Filter by Source</label>

<select name="source" class="form-control">

<option value="All">All</option>

{% for key, value in source\_choices %}

<option value="{{ key }}">{{ value }}</option>

{% endfor %}

</select>

</div>

<div class="form-group">

<label for="support\_category">Filter by Support Category</label>

<select name="support\_category" class="form-control">

<option value="All">All</option>

{% for key, value in support\_category\_choices %}

<option value="{{ key }}">{{ value }}</option>

{% endfor %}

</select>

</div>

<div class="form-group">

<label for="incident">Filter by Incident</label>

<select name="incident" class="form-control">

<option value="All">All</option>

{% for key, value in incident\_choices %}

<option value="{{ key }}">{{ value }}</option>

{% endfor %}

</select>

</div>

<button type="submit" class="btn btn-primary mt-3">Filter</button>

</form>

<table class="table">

<thead>

<tr>

<th>Title</th>

<th>Source</th>

<th>Incident</th>

<th>Support Category</th>

<th>Status</th>

<th>Office</th>

<th>Actions</th>

</tr>

</thead>

<tbody>

{% for task in tasks %}

<tr>

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

<td>{{ task.get\_source\_display }}</td>

<td>{{ task.get\_incident\_display }}</td>

<td>{{ task.get\_support\_category\_display }}</td>

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

<td>{{ task.office\_name }}</td>

<td>

<a href="{% url 'edit\_task' task.id %}" class="btn btn-warning">Edit</a>

<a href="{% url 'delete\_task' task.id %}" class="btn btn-danger">Delete</a>

</td>

</tr>

{% endfor %}

</tbody>

</table>

</div>

{% endblock %}