|  |
| --- |
| /pics/views.py |

|  |
| --- |
| from django.shortcuts import render, redirect  from django.urls import reverse  from django.utils.http import urlencode  from .models import Customers  from .forms import UsersLoginForm, SignupForm  from django.contrib import messages  from django.shortcuts import render, redirect,HttpResponse,get\_object\_or\_404,HttpResponseRedirect  from django.contrib import messages  from .models import Customers,Albums,Photo,FavAlbums  from .forms import UsersLoginForm,AlbumCreations  import uuid,os  from django.conf import settings  from django.contrib.auth import authenticate, login as auth\_login  from django.contrib import messages  from .forms import UsersLoginForm,ImagesForm  import shutil  def index(request):  return render(request,'index.html')  def dashboard(request):  logined,email = verifyLogin(request)  print(logined)  if logined:    user = Customers.objects.get(email=email)  albums = Albums.objects.filter(user=user)  favorited\_albums = FavAlbums.objects.filter(user=user)  context ={  'albums' : albums,  'fav':favorited\_albums  }    return render(request, 'dashboard.html',context)  else:    return redirect('login')  def signup(request):  if request.method == 'POST':  form = SignupForm(request.POST)  if form.is\_valid():  print('else21')  user = form.save(commit=False)  user.set\_password(form.cleaned\_data['password'])  user.save()  return redirect('login')  else:  print('else')  return redirect('login')  else:  return redirect('login')    def login(request):  next\_url = request.POST.get('next','/dashboard')  if request.method == 'POST':  frm = UsersLoginForm(request.POST)  if frm.is\_valid():  email = frm.cleaned\_data.get('email')  password = frm.cleaned\_data.get('password')    if email and password:  try:  user = Customers.objects.get(email=email)  if user.check\_password(password):    print('logined')  if next\_url:  response = HttpResponseRedirect(next\_url)  response.set\_cookie('logined', [True,email])    return response  else:  response = redirect('dashboard')  response.set\_cookie('logined', [True,email])    return response  else:  messages.error(request, 'Incorrect password.')  except Customers.DoesNotExist:  messages.error(request, 'User does not exist.')  except Customers.MultipleObjectsReturned:  messages.error(request, 'Multiple users found with this email.')  else:  messages.error(request, 'Email or password not provided.')  else:  messages.error(request, 'Invalid form submission.')  else:  frm = UsersLoginForm()  next\_url = request.GET.get('next', '/dashboard')  return render(request, "login.html", {'form': frm,'next':next\_url})  def verifyLogin(request):  try:  response = eval(request.COOKIES.get('logined'))  logined = response[0]  if logined:  email = response[1]  else:  email = None  return logined ,email  except TypeError:  return False,None    def albumcreation(request):  logined, email = verifyLogin(request)  if logined:  if request.method == 'POST':  frm = AlbumCreations(request.POST)  if frm.is\_valid():  album\_name = request.POST.get('name')  album\_share = request.POST.get('share')  if album\_share == 'on' : album\_share = True  else: album\_share = False    code = str(uuid.uuid4())    try:    album = Albums.objects.create(  name=album\_name,  path=f'{code}\\',  user=Customers.objects.get(email=email),  share=album\_share,  code=code  )      media\_folder = os.path.join(settings.MEDIA\_ROOT, code)  os.makedirs(media\_folder, exist\_ok=True)    return redirect('dashboard')  except Exception as e:    print(f'Error creating album: {e}')  messages.error(request, 'An error occurred while creating the album.')  else:  frm = AlbumCreations()    return render(request, 'albumcreate.html', {'form': frm})  else:  return redirect('signup')  get\_object\_or\_404  def logout(request):    response = redirect('/') #  response.delete\_cookie('logined')  print('logout')  return response  def album\_view(request, id):  album = get\_object\_or\_404(Albums, code=id)  photos = album.photos.all()  form = ImagesForm()  logind, email = verifyLogin(request)  print('hi')  # Redirect to login if the user is not logged in  # if not logind:  # login\_url = reverse('login')  # next\_url = f"{login\_url}?{urlencode({'next': request.get\_full\_path()})}"  # return redirect(next\_url)  path = request.get\_full\_path()  login\_url = reverse('login')  next\_url = f"{login\_url}?{urlencode({'next': path})}"    def check\_ownership(album\_code, user\_email):  album\_details = get\_object\_or\_404(Albums, code=album\_code)  return album\_details.user.email == user\_email  # Check if the album is favorited by the user  try:  user = Customers.objects.get(email=email)  is\_favorited = FavAlbums.objects.filter(user=user, album=album).exists()  except:  user = None  is\_favorited = False  print(f'is\_favorited {is\_favorited}')  context = {  'album': album,  'photos': photos,  'form': form,  'logined': logind,  'owner': check\_ownership(id, email),  'user': user,  'is\_favorited': is\_favorited,  'path': next\_url  }    return render(request, 'albumview.html', context)  def upload\_photos(request, id):  album = get\_object\_or\_404(Albums, code=id)  photos = album.photos.all()  if request.method == 'POST':  form = ImagesForm(request.POST, request.FILES)  print("Form errors:", form.errors)  print("FILES data:", request.FILES)    if form.is\_valid():  images = request.FILES.getlist('images')  email = verifyLogin(request)[1]  user = Customers.objects.get(email=email)  for image in images:  Photo.objects.create(  album=album,  images=image,  uploaded\_by=user  )  return redirect('album\_view', id=id)  else:  form = ImagesForm()  context = {  'album': album,  'photos': photos,  'form': form  }  return render(request, 'albumview.html', context)  from django.shortcuts import get\_object\_or\_404, redirect  from .models import Albums, Photo  def delete\_album(request, id):  logined, email = verifyLogin(request)  if logined:  try:  user = Customers.objects.get(email=email)  album = get\_object\_or\_404(Albums, id=id, user=user)    album.delete()      media\_folder = os.path.join(settings.MEDIA\_ROOT, album.code)  print(media\_folder)  if os.path.exists(media\_folder):  shutil.rmtree(media\_folder)  messages.success(request, 'Album deleted successfully.')  return redirect('dashboard')  except Exception as e:  print(f'Error deleting albredirectdashboardum: {e}')  messages.error(request, 'An error occurred while deleting the album.')  return redirect('dashboard')  else:  return redirect('signup')  def delete\_photo(request, id, pid):  # Get the album based on the ID  album = get\_object\_or\_404(Albums, code=id)  photo = get\_object\_or\_404(Photo,id=pid)  filename = photo.images.name.split('/')[-1]  # Construct the full path of the file  file\_path = os.path.join(settings.MEDIA\_ROOT, f'{id}\{filename}')    # Get the photo record  photo = get\_object\_or\_404(Photo, album=album, images=photo.images.name)  print('file:',file\_path)  # Remove the file from the filesystem  if os.path.isfile(file\_path):  print(file\_path)  os.remove(file\_path)  print('Deleted')    # Delete the photo record from the database  photo.delete()    # Redirect to the album view page  return redirect('album\_view', id=id)  def imageupload(request,id):  form = ImagesForm(request.POST, request.FILES)  if request.method == 'POST':  album = Albums.objects.get(code=id)    user = album.user  images = request.FILES.getlist('images')  for image in images:  image\_ins = Photo.objects.create(  album = album,  uploaded\_by = user  )  image\_ins.save()  return redirect(f'album/{id}')  context = {'form': form}  return render(request, "upload.html", context)  def Fav\_albums(request, id):  logined, email = verifyLogin(request)  user = get\_object\_or\_404(Customers, email=email) # Get the user based on the email  album = get\_object\_or\_404(Albums, code=id) # Make sure you're fetching the album by the correct field  # Check if the album is already favorited  if FavAlbums.objects.filter(user=user, album=album).exists():  FavAlbums.objects.filter(user=user, album=album).delete() # Remove from favorites  else:  FavAlbums.objects.create(user=user, album=album) # Add to favorites  return redirect('album\_view', id=id) # Redirect to the album view    def EditAlbum(request,id):  album = Albums.objects.get(code=id)  if request.POST:  name = request.POST.get('name')  share = request.POST.get('share')  if share == 'on':  share = True  else:  share = False  Album = Albums.objects.get(code=id)  Album.name = name  Album.share = share  Album.save()  return redirect('album\_view',id=id)  context = {  'album' : album  }  return render(request, 'albumedit.html', context) |

|  |
| --- |
| /pics/urls.py |

|  |
| --- |
| """Pixora 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.urls import include, path  from . import views  from django.conf.urls.static import static  from django.conf import settings  from django.contrib.auth import views as auth\_views  urlpatterns = [  path('',views.index,name='index'),  path('dashboard/', views.dashboard, name='dashboard'),  path('accounts/login/', views.login, name='login'),  path('logout/', views.logout, name='logout'),  path('signup/', views.signup, name='signup'),  path('albumcreate/', views.albumcreation, name='albumcreate'),  path('album/<str:id>/', views.album\_view, name='album\_view'), # Use <str:id> for string ID  path('album/<str:id>/upload/', views.upload\_photos, name='upload\_photos'), # Use <str:id> for string ID  path('delete/<id>/<pid>/', views.delete\_photo, name='delete\_photo'),  path('album/delete/<int:id>/', views.delete\_album, name='delete\_album'),  path('album/<uuid:id>/favorite/', views.Fav\_albums, name='favorite\_album'),  path('album/<uuid:id>/edit/',views.EditAlbum,name='editalbum')  ] + static(settings.MEDIA\_URL, document\_root=settings.MEDIA\_ROOT) |

|  |
| --- |
| pics/models.py |

|  |
| --- |
| from django.db import models  from django.db import models  from django.contrib.auth.hashers import make\_password, check\_password  class Customers(models.Model):  name = models.CharField(max\_length=100, verbose\_name="Full Name")  password = models.CharField(max\_length=128, verbose\_name="Password")  email = models.EmailField(max\_length=255, verbose\_name="Email Address")  created\_at = models.DateTimeField(auto\_now\_add=True, verbose\_name="Date Created")  updated\_at = models.DateTimeField(auto\_now=True, verbose\_name="Last Updated")    def \_\_str\_\_(self):  return self.name  def set\_password(self, raw\_password):  self.password = make\_password(raw\_password)    def check\_password(self, raw\_password):  return check\_password(raw\_password, self.password)    # Create your models here.  class AlbumsManager(models.Manager):  def create\_with\_username(self, code, path, username):  user = Customers.objects.get(username=username)  return self.create(code=code, path=path, user=user)  class Albums(models.Model):  code = models.CharField(max\_length=255, unique=True)  name = models.CharField(max\_length=255 )  share = models.BooleanField(default=False)  path = models.CharField(max\_length=255,default=f'{code}\\')  user = models.ForeignKey(Customers, on\_delete=models.CASCADE, related\_name='albums')  created\_at = models.DateTimeField(auto\_now\_add=True)  objects = AlbumsManager() # Use the custom manager  def \_\_str\_\_(self):  return f"Album {self.code} by {self.user.username}"  import os  def uploadto(instance, filename):  return os.path.join(f'{instance.album.code}/',filename)  class Photo(models.Model):  album = models.ForeignKey(Albums, on\_delete=models.CASCADE, related\_name='photos')  images = models.FileField(upload\_to=uploadto,null=True , blank=True)  uploaded\_by = models.ForeignKey(Customers, on\_delete=models.SET\_NULL, null=True, blank=True)  uploaded\_at = models.DateTimeField(auto\_now\_add=True)  class FavAlbums(models.Model):  user = models.ForeignKey(Customers, on\_delete=models.CASCADE, related\_name='favorite\_albums')  album = models.ForeignKey(Albums, on\_delete=models.CASCADE, related\_name='favorited\_by')  favorited\_at = models.DateTimeField(auto\_now\_add=True)  class Meta:  unique\_together = ('user', 'album') # Ensures that a user can only favorite an album once  def \_\_str\_\_(self):  return f"{self.user.username}'s favorite: {self.album.name}" |

|  |
| --- |
| pics/forms.py |

|  |
| --- |
| from django.forms import ModelForm  from . models import Customers,Albums,Photo  from django import forms  from django.core.exceptions import ValidationError  class UsersLoginForm(ModelForm):  password = forms.CharField(  label='Password', # Custom label  widget=forms.PasswordInput(attrs={  'class': 'form-control',  'id': 'exampleFormControlTextarea1',  'placeholder': 'Enter your password',  'required': True  })  )  email = forms.EmailField(  label='Email ID', # Custom label  max\_length=200,  widget=forms.EmailInput(attrs={  'class': 'form-control',  'id': 'exampleFormControlTextarea1',  'placeholder': 'Enter your email',  'required': True  })  )  class Meta:  model = Customers  fields = ['email', 'password']  class SignupForm(ModelForm):    class Meta:  model = Customers  fields = '\_\_all\_\_'  def clean\_email(self):  email = self.cleaned\_data.get('email')  if Customers.objects.filter(email=email).exists():  raise ValidationError("A customer with this email already exists. signform ")  return email  class AlbumCreations(ModelForm):  class Meta:  model = Albums  fields = ['name','share']  class ImagesForm(forms.Form):  from django import forms  class ImagesForm(forms.Form):  images = forms.FileField(  widget=forms.ClearableFileInput(attrs={'multiple': True}),  label='Upload Images',  required=False  ) |

|  |
| --- |
| Pixora/settings.py |

|  |
| --- |
| """  Django settings for Pixora project.  Generated by 'django-admin startproject' using Django 2.2.  For more information on this file, see  https://docs.djangoproject.com/en/2.2/topics/settings/  For the full list of settings and their values, see  https://docs.djangoproject.com/en/2.2/ref/settings/  """  import os  from pathlib import Path  # Build paths inside the project like this: os.path.join(BASE\_DIR, ...)  BASE\_DIR = Path(\_\_file\_\_).resolve().parent.parent  # Quick-start development settings - unsuitable for production  # See https://docs.djangoproject.com/en/2.2/howto/deployment/checklist/  # SECURITY WARNING: keep the secret key used in production secret!  SECRET\_KEY = 'tyj(0o!b7yhw(ai%118sm6a7$zhqsm6^%r^^2m9j47^6s8)#fo'  # 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',  'pics',  ]  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 = 'Pixora.urls'  TEMPLATES = [  {  'BACKEND': 'django.template.backends.django.DjangoTemplates',  'DIRS': [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 = 'Pixora.wsgi.application'  # settings.py  # AUTHENTICATION\_BACKENDS = [  # 'picsu.backends.EmailBackend', # Replace 'yourapp' with the actual name of your app  # 'django.contrib.auth.backends.ModelBackend', # Keep the default backend if needed  # ]  # Database  # https://docs.djangoproject.com/en/2.2/ref/settings/#databases  from dotenv import load\_dotenv  import os  load\_dotenv()  DATABASES = {  'default': {  'ENGINE': 'django.db.backends.mysql',  'NAME': 'pixora', # Replace with your database name  'USER': os.getenv('DATABASE\_USERNAME'), # Replace with your MySQL username  'PASSWORD': os.getenv('DATABASE\_PASSWORD'), # Replace with your MySQL password  'HOST': os.getenv('DATABASE\_HOST'), # Set to your database host  'PORT': int(os.getenv('DATABASE\_PORT')),  }  }  # Password validation  # https://docs.djangoproject.com/en/2.2/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/2.2/topics/i18n/  LANGUAGE\_CODE = 'en-us'  TIME\_ZONE = 'Asia/Kolkata'  USE\_I18N = True  USE\_L10N = True  USE\_TZ = True  # settings.py  # Static files (CSS, JavaScript, Images)  # https://docs.djangoproject.com/en/2.2/howto/static-files/  STATIC\_URL = '/static/'  STATICFILES\_DIRS = [BASE\_DIR / 'static']  MEDIA\_URL = '/media/'  MEDIA\_ROOT = os.path.join(BASE\_DIR, 'media')  DEFAULT\_AUTO\_FIELD = 'django.db.models.BigAutoField' |

|  |
| --- |
| Pixora/urls.py |

|  |
| --- |
| """Pixora 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 include, path  from django.conf.urls.static import static  from django.conf import settings  urlpatterns = [  path('wp-admin/', admin.site.urls),  path('admin/', include(('wp\_admin.urls', 'wp\_admin'), namespace='wp\_admin')),  path('',include("pics.urls")),  ]  urlpatterns += static(settings.MEDIA\_URL, document\_root=settings.MEDIA\_ROOT) |

|  |
| --- |
| templates/albumcreate.html |

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Create Album</title>  <style>  body {  font-family: Arial, sans-serif;  background-color: #f0f0f0;  color: #333;  margin: 0;  padding: 0;  display: flex;  justify-content: center;  align-items: center;  height: 100vh;  }  .container {  background-color: white;  padding: 20px 40px;  border-radius: 10px;  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);  width: 100%;  max-width: 400px;  text-align: center;  }  h2 {  margin-bottom: 20px;  color: #444;  }  .messages {  list-style-type: none;  padding: 0;  margin-bottom: 20px;  }  .messages li {  background-color: #ffdddd;  color: #d8000c;  border-left: 4px solid #d8000c;  padding: 10px;  margin-bottom: 10px;  border-radius: 5px;  font-size: 14px;  text-align: left;  }  form {  display: flex;  flex-direction: column;  }  form p {  margin-bottom: 15px;  }  button {  background-color: #007bff;  color: white;  padding: 10px 15px;  border: none;  border-radius: 5px;  cursor: pointer;  font-size: 16px;  transition: background-color 0.3s ease;  }  button:hover {  background-color: #0056b3;  }  </style>  </head>  <body>  <div class="container">  <h2>Create Album</h2>  <!-- Display messages -->  {% if messages %}  <ul class="messages">  {% for message in messages %}  <li>{{ message }}</li>  {% endfor %}  </ul>  {% endif %}  <form method="post" action="{% url 'albumcreate' %}">  {% csrf\_token %}  {{ form.as\_p }}  <button type="submit">Create Album</button>  </form>  </div>  </body>  </html> |

|  |
| --- |
| templates/albumedit.html |

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Edit Album</title>  <style>  body {  font-family: Arial, sans-serif;  background-color: #f0f0f0;  color: #333;  margin: 0;  padding: 0;  display: flex;  justify-content: center;  align-items: center;  height: 100vh;  }  .container {  background-color: white;  padding: 20px 40px;  border-radius: 10px;  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);  width: 100%;  max-width: 400px;  text-align: center;  }  h2 {  margin-bottom: 20px;  color: #444;  }  .messages {  list-style-type: none;  padding: 0;  margin-bottom: 20px;  }  .messages li {  background-color: #ffdddd;  color: #d8000c;  border-left: 4px solid #d8000c;  padding: 10px;  margin-bottom: 10px;  border-radius: 5px;  font-size: 14px;  text-align: left;  }  form {  display: flex;  flex-direction: column;  }  form p {  margin-bottom: 15px;  }  button {  background-color: #007bff;  color: white;  padding: 10px 15px;  border: none;  border-radius: 5px;  cursor: pointer;  font-size: 16px;  transition: background-color 0.3s ease;  }  button:hover {  background-color: #0056b3;  }  </style>  </head>  <body>  <div class="container">  <h2>Edit Album</h2>  <!-- Display messages -->  {% if messages %}  <ul class="messages">  {% for message in messages %}  <li>{{ message }}</li>  {% endfor %}  </ul>  {% endif %}  <form method="post" action="{% url 'editalbum' album.code %}">  {% csrf\_token %}  <label>  <span>Album Name</span>  <input type="text" name="name" required value={{album.name}}>  </label>  <label>    </label>  <span>Share</span>  {% if album.share %}  <input type="checkbox" name="share" checked >  {% elif not album.share %}  <input type="checkbox" name="share" >  {% endif %}  </label>  <button type="submit">Finished</button>  </form>  </div>  </body>  </html> |

|  |
| --- |
| templates/albumview.html |

|  |
| --- |
| {% extends 'base.html' %}  {% block title %}  <title>Photos in {{ album.name }}</title>  <style type="text/css">  body {  font-family: Arial, sans-serif;  margin: 0;  padding: 0;  background-color: #f8f9fa;  }  .container {  max-width: 1200px;  margin: 2rem auto;  padding: 0 15px;  }  .favorite-form {  margin-bottom: 20px;  text-align: center;  }  h1 {  text-align: center;  margin-bottom: 2rem;  color: #333;  }  .photo-grid {  display: grid;  grid-template-columns: repeat(auto-fill, minmax(250px, 1fr));  gap: 15px;  }  .photo-item {  background-color: white;  border-radius: 8px;  overflow: hidden;  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);  transition: transform 0.3s;  }  .photo-item:hover {  transform: scale(1.05);  }  .photo-img {  width: 100%;  height: auto;  display: block;  }  .photo-info {  padding: 10px;  text-align: center;  }  .photo-info p {  margin: 5px 0;  font-size: 14px;  color: #555;  }  .delete-link {  color: #dc3545;  text-decoration: none;  font-weight: bold;  }  .delete-link:hover {  text-decoration: underline;  }  .upload-container {  margin-top: 2rem;  text-align: center;  }  .upload-container form {  display: inline-block;  text-align: left;  background-color: white;  padding: 15px;  border-radius: 8px;  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);  }  .upload-container .btn {  margin-top: 10px;  }  </style>  {% endblock %}  {% block content %}  <div class="container">  <h1>Photos in {{ album.name }}</h1>  {% if album.share %}  {% if logined %}  <div class="favorite-form">  <form method="post" action="{% url 'favorite\_album' album.code %}">  {% csrf\_token %}  {% if is\_favorited %}  <button type="submit" class="btn btn-danger">Remove from Favorites</button>  {% else %}  <button type="submit" class="btn btn-primary">Add to Favorites</button>  {% endif %}  </form>  </div>  {% endif %}  <div class="photo-grid">  {% if photos %}  {% for photo in photos %}  <div class="photo-item">  <img src="{{ photo.images.url }}" alt="Photo uploaded by {{ photo.uploaded\_by }}" class="photo-img">  <div class="photo-info">  <p><strong>Uploaded by:</strong> {{ photo.uploaded\_by }}</p>  <p><strong>Uploaded at:</strong> {{ photo.uploaded\_at }}</p>  {% if logined %}  <a href="{% url 'delete\_photo' album.code photo.id %}"  onclick="return confirm('Are you sure you want to delete this photo?');"  class="delete-link">Delete</a>  {% endif %}  </div>  </div>  {% endfor %}  {% else %}  <p>No photos available.</p>  {% endif %}  </div>  {% if logined %}  <!-- upload center -->  <div class="upload-container">  <form method="POST" action="{% url 'upload\_photos' album.code %}" enctype="multipart/form-data">  {% csrf\_token %}  <div class="mb-3">  <label for="formFile" class="form-label">Upload Files Here</label>  {{ form.as\_p }}  </div>  <button type="submit" class="btn btn-outline-primary">Submit</button>  </form>  </div>  {% elif not logined and owner %}  <!-- photo viewer -->  <div class="photo-grid">  {% if photos %}  {% for photo in photos %}  <div class="photo-item">  <img src="{{ photo.images.url }}" alt="Photo uploaded by {{ photo.uploaded\_by }}" class="photo-img">  <div class="photo-info">  <p><strong>Uploaded by:</strong> {{ photo.uploaded\_by }}</p>  <p><strong>Uploaded at:</strong> {{ photo.uploaded\_at }}</p>  {% if logined %}  <a href="{% url 'delete\_photo' album.code photo.id %}"  onclick="return confirm('Are you sure you want to delete this photo?');"  class="delete-link">Delete</a>  {% endif %}  </div>  </div>  {% endfor %}  {% else %}  <p>No photos available.</p>  {% endif %}  </div>  <div class="upload-container">  <form method="POST" action="{% url 'upload\_photos' album.code %}" enctype="multipart/form-data">  {% csrf\_token %}  <div class="mb-3">  <label for="formFile" class="form-label">Upload Files Here</label>  {{ form.as\_p }}  </div>  <button type="submit" class="btn btn-outline-primary">Submit</button>  </form>  </div>  {% else %}  <a href="{{ path }}" class="btn btn-primary">Login to UPLOAD</a>  {% endif %}  {% else %}  {% if owner%}    <div class="photo-grid">  {% if photos %}  {% for photo in photos %}  <div class="photo-item">  <img src="{{ photo.images.url }}" alt="Photo uploaded by {{ photo.uploaded\_by }}" class="photo-img">  <div class="photo-info">  <p><strong>Uploaded by:</strong> {{ photo.uploaded\_by }}</p>  <p><strong>Uploaded at:</strong> {{ photo.uploaded\_at }}</p>  {% if logined %}  <a href="{% url 'delete\_photo' album.code photo.id %}"  onclick="return confirm('Are you sure you want to delete this photo?');"  class="delete-link">Delete</a>  {% endif %}  </div>  </div>  {% endfor %}  {% else %}  <p>No photos available.</p>  {% endif %}  </div>  <div class="upload-container">  <form method="POST" action="{% url 'upload\_photos' album.code %}" enctype="multipart/form-data">  {% csrf\_token %}  <div class="mb-3">  <label for="formFile" class="form-label">Upload Files Here</label>  {{ form.as\_p }}  </div>  <button type="submit" class="btn btn-outline-primary">Submit</button>  </form>  </div>  {% else %}  <p>  This Album isn't shared.  </p>  {% endif %}  {% endif %}  </div>  {% endblock %} |

|  |
| --- |
| templates/base.html |

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <!-- Required Meta Tags -->  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <meta name="description" content="PIXORA - Manage your photo albums and collections with ease.">    <!-- Bootstrap CSS -->  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1RWuH61DGLwZJEdK2Kadq2F9CUG65" crossorigin="anonymous">    <!-- Favicon -->  {% load static %}  <link rel="icon" type="image/x-icon" href="{% static 'images/favicon.ico' %}">  <!-- Title Block with Default Fallback -->  {% block title %}    {% endblock %}  </head>  <body>  <!-- Navbar -->  <nav class="navbar navbar-expand-lg navbar-light bg-light">  <div class="container-fluid">  <a class="navbar-brand" href="#" aria-label="PIXORA Homepage">PIXORA</a>  <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNavAltMarkup" aria-controls="navbarNavAltMarkup" aria-expanded="false" aria-label="Toggle navigation">  <span class="navbar-toggler-icon"></span>  </button>  <div class="collapse navbar-collapse" id="navbarNavAltMarkup">  <div class="navbar-nav">  <a class="nav-link active" aria-current="page" href="/dashboard">Dashboard</a>  <a class="nav-link" href="/albumcreate">Create Album</a>  </div>  <div class="navbar-nav ms-auto">  {% if user.is\_authenticated %}  <span class="navbar-text me-2">Hello, {{ user.username }}!</span>  <a class="nav-link" href="{% url 'logout' %}">Logout</a>  {% else %}  <a class="nav-link" href="{% url 'login' %}">Login</a>  <a class="nav-link" href="{% url 'signup' %}">Sign Up</a>  {% endif %}  </div>  </div>  </div>  </nav>  <!-- Main Content -->  <div class="container mt-4">  {% block content %}  <!-- Content will be injected here -->  {% endblock %}  </div>  <!-- Footer -->  <footer class="bg-light text-center py-3 mt-5">  <div class="container">  <p class="mb-0">&copy; 2024 PIXORA. All rights reserved.</p>  <a href="/privacy-policy" class="text-muted">Privacy Policy</a> |  <a href="/terms-of-service" class="text-muted">Terms of Service</a>  </div>  </footer>  <!-- Bootstrap JS & Popper -->  <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js" integrity="sha384-oBqDVmMz9ATKxIep9tiCxS/Z9fNfEXiDAYTujMAeBAsjFuCZSmKbSSUnQlmh/jp3" crossorigin="anonymous"></script>  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.min.js" integrity="sha384-cuYeSxntonz0PPNlHhBs68uyIAVpIIOZZ5JqeqvYYIcEL727kskC66kF92t6Xl2V" crossorigin="anonymous"></script>  </body>  </html> |

|  |
| --- |
| templates/dashboard.html |

|  |
| --- |
| {% extends 'base.html' %}  {% block title %}    <title>Dashboard</title>  <style>  body {  font-family: Arial, sans-serif;  margin: 20px;  }  table {  width: 100%;  border-collapse: collapse;  }  table, th, td {  border: 1px solid #ddd;  }  th, td {  padding: 8px;  text-align: left;  }  th {  background-color: #f4f4f4;  }  </style>  <link rel="stylesheet" href="https://fonts.googleapis.com/css2?family=Material+Symbols+Outlined:opsz,wght,FILL,GRAD@20..48,100..700,0..1,-50..200" />  {% endblock %}  {% block content %}  <body>  <h1>Your Albums</h1>    <table>  <thead>  <tr>    <th>Name</th>  <th>Shared</th>  <th>Path</th>  <th>Created At</th>  <th>Go</th>  <th>Delete</th>  </tr>  </thead>  <tbody>  {% for album in albums %}  <tr>    <td>{{ album.name }}</td>  <td>{{ album.share|yesno:"Yes,No" }}</td>  <td>{{ album.path }}</td>  <td>{{ album.created\_at }}</td>  <td><a href="/album/{{album.code}}"><span class="material-symbols-outlined">  output  </span></td>  <td><a href="{% url 'delete\_album' album.id %}" onclick="return confirm('Are you sure you want to delete this album?');">  Delete Album  </a></td>  </tr>  {% empty %}  <tr>  <td colspan="6">You have no albums.</td>  </tr>  {% endfor %}  </tbody>  </table>  <table>  <h1>Your Favorites</h1>    <thead>  <tr>    <th>Name</th>  <th>Shared</th>  <th>Path</th>  <th>Created At</th>  <th>Go</th>  <th>Favorited At</th>  </tr>  </thead>  <tbody>  {% for fa in fav %}  <tr>    <td>{{ fa.album.name }}</td>  <td>{{ fa.album.share|yesno:"Yes,No" }}</td>  <td>{{ fa.album.path }}</td>  <td>{{ fa.album.created\_at }}</td>  <td><a href="/album/{{fa.album.code}}"><span class="material-symbols-outlined">  output  </span></a></td>  <td>{{ fa.favorited\_at }}</td>  </tr>  {% empty %}  <tr>  <td colspan="6">You have no Favorite albums.</td>  </tr>  {% endfor %}  </tbody>  </table>    </body>  {% endblock %} |

|  |
| --- |
| templates/index.html |

|  |
| --- |
| {% extends 'base.html' %}  {% block title %}  <title>Pixora</title>  {% endblock %}  {% block content %}  <!-- Title at the top -->  <div style="text-align: center; margin-top: 20px;">  <h1 style="font-family: 'Arial', sans-serif; font-weight: bold; font-size: 48px;">Pixora - Album Sharing App</h1>  </div>  <!-- Main content centered vertically with more upward adjustment -->  <div style="display: flex; justify-content: center; align-items: center; height: 70vh; text-align: center;">  <div>  <!-- Move the description text a bit higher with reduced margin -->  <p style="margin-bottom: 20px;">  Pixora is a user-friendly album-sharing web application that allows users to create, upload, and share photo albums with others.  Users can also mark albums as their favorites, making it easy to revisit memorable moments.  </p>  <h2 style="text-align: left; font-family: 'Berlin Sans FB', sans-serif; font-weight: normal;">Key Features:</h2>  <ul style="list-style-type: none; text-align: left;">  <li style="margin-bottom: 10px;">Easy sign-up and login with user authentication</li>  <li style="margin-bottom: 10px;">Google Drive integration for seamless album storage</li>  <li style="margin-bottom: 10px;">Create, upload, and share albums with a shareable link</li>  <li style="margin-bottom: 10px;">Mark albums as favorites to organize cherished moments</li>  <li style="margin-bottom: 10px;">View and upload photos through a shared link with restricted access</li>  <li style="margin-bottom: 10px;">Responsive design for both desktop and mobile users</li>  </ul>  <div>  <a href="#" style="display: inline-block; padding: 10px 20px; background-color: #007BFF; color: white; text-decoration: none; border-radius: 5px; font-size: 16px;">  Visit Pixora  </a>  </div>  </div>  </div>  <!-- Credits section moved up -->  <div style="text-align: center; padding: 10px 0; margin-top: 40px;">  <p style="font-family: 'Bell MT', monospace;">  <strong>CREDITS:</strong>  <a href="https://instagram.com/aaronshenny" target="\_blank" class="credit-link">Aaron Shenny</a>,  <a href="https://instagram.com/aaronjimmy\_207" target="\_blank" class="credit-link">Aaron Jimmy</a>,  <a href="https://instagram.com/aswinaravind27" target="\_blank" class="credit-link">Aswin Aravind</a>  </p>  </div>  <style>  .credit-link {  text-decoration: none;  color: inherit;  transition: color 0.3s ease, transform 0.3s ease;  }  .credit-link:hover {  color: #00fce3;  transform: scale(3.5);  }  </style>  {% endblock %} |

|  |
| --- |
| templates/login.html |

|  |
| --- |
| {% load static %}  <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Login & Registration</title>  <link rel="stylesheet" type="text/css" href="{% static 'css/style.css' %}">  <link href="https://fonts.googleapis.com/css?family=Nunito:400,600,700,800&display=swap" rel="stylesheet">  </head>  <body>  <div class="cont">  <!-- Login Form (Django) -->  <div class="form sign-in">  <h2>Sign In</h2>  <!-- Display Django messages -->  {% if messages %}  <ul>  {% for message in messages %}  <li>{{ message }}</li>  {% endfor %}  </ul>  {% endif %}  <form method="post" action="{% url 'login' %}">  <input type="hidden" name="next" value="{{ next }}">  {% csrf\_token %}  {{ form.as\_p }}  <button class="submit" type="submit">Sign In</button>  </form>        </div>  <div class="sub-cont">  <div class="img">  <div class="img-text m-up">  <h2>New here?</h2>  <p>Sign up and discover a great amount of new opportunities!</p>  </div>  <div class="img-text m-in">  <h2>One of us?</h2>  <p>If you already have an account, just sign in. We've missed you!</p>  </div>  <div class="img-btn">  <span class="m-up">Sign Up</span>  <span class="m-in">Sign In</span>  </div>  </div>  <!-- Registration Form -->  <div class="form sign-up">  <h2>Sign Up</h2>  <form method="post" action="{% url 'signup' %}">  {% csrf\_token %}  <label>  <span>Name</span>  <input type="text" name="name" required>  </label>  <label>  <span>Email</span>  <input type="email" name="email" required>  </label>  <label>  <span>Password</span>  <input type="password" name="password" required>  </label>    <button type="submit" class="submit">Sign Up Now</button>  </form>  </div>  </div>  </div>  <script type="text/javascript" src="{% static 'js/script.js' %}"></script>  </body>  </html> |

|  |
| --- |
| templates/sign-up.html |

|  |
| --- |
| <!-- templates/signup.html -->  <form method="post">  {% csrf\_token %}  {{ form.as\_p }}  <button type="submit">Sign Up</button>  </form> |