**PROJECT DESCRIPTION**

**Framework used:** Django

Django is an open source web application frame work written in Python.

**Command to start Project:** django-admin startproject skintone\_app

**Folder Structure**

In folder structure we have following html and css files listed below:

1. Templates
2. Accounts

* Confirm\_email.html
* Email.html
* Login.html
* Messages.html
* Register.html
* Verify.html

1. Pages

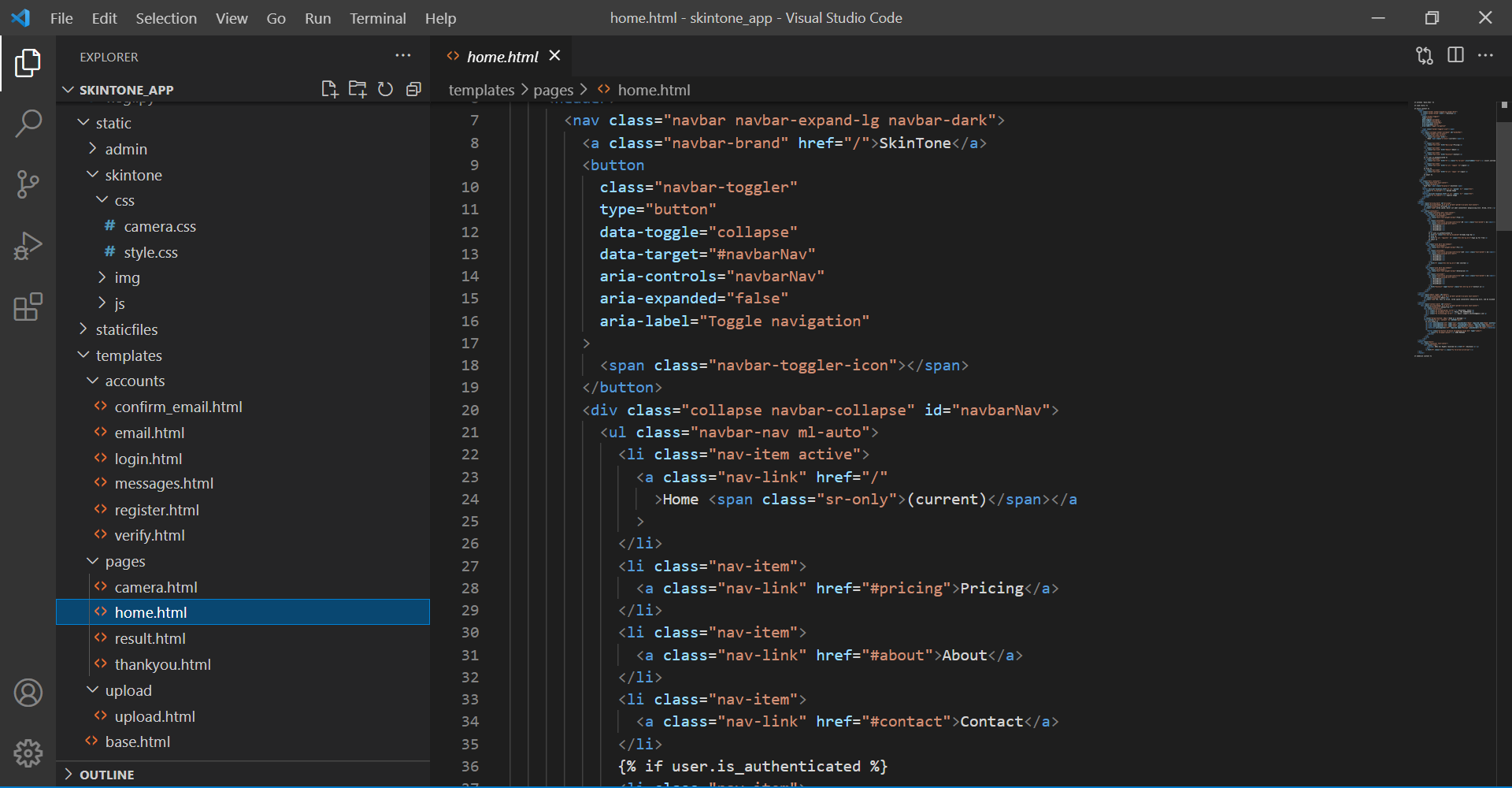
* Camera.html
* Home.html
* Result.html
* Thankyou.html

1. Upload

* Upload.html

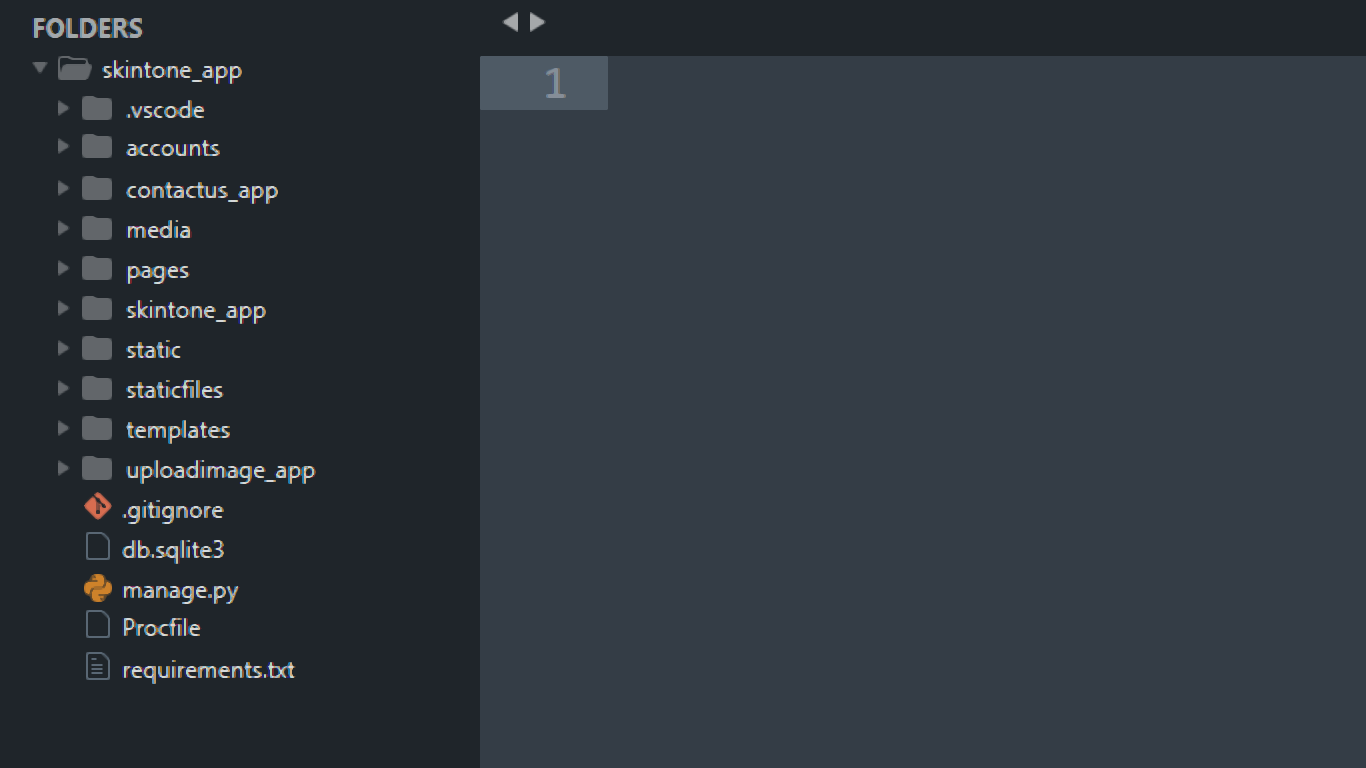
1. Base.html
2. Static
3. skintone

* css
* Style.css
* camera.css



In folder structure we have Four APPs that are listed below:

1. Accounts
2. Pages
3. Upload images app
4. Contact us app



1. **Accounts**

Any web app which will provide some sort of services and

work on user data, will be user specific, so we have to made records of the various task performed by user and also for security reasons and restricting access, app must contain a user authentication system. So, for that reason we have provided a secure registration system with email OTP verification that provides a better way for registering user with there real credentials. The logic for providing these features are listed billow:

**urls.py**

It handles all the Urls related to account sections.

*from* django.urls *import* path

*from* . *import* views

urlpatterns = [

path('register', views.register, *name*='register'),

path('verify', views.verify, *name*='verify'),

path('login', views.login, *name*='login'),

path('logout', views.logout, *name*='logout'),

path('dashboard', views.dashboard, *name*='dashboard'),

]

**views.py**

It handles all the views related to account sections.

*from* django.contrib *import* messages

*from* django.shortcuts *import* redirect, render

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

*from* django.core.mail *import* EmailMultiAlternatives

*from* django.template.loader *import* render\_to\_string

*from* django.utils *import* html

*from* django.utils.html *import* strip\_tags

*from* django.conf *import* settings

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

*from* .models *import* Profile

*import* random

*# Create your views here.*

*# register route*

def register(*request*):

*if* *request*.user.is\_authenticated:

*return* redirect('home')

*else*:

*if* *request*.method == 'POST':

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

email = *request*.POST.get('email')

password = *request*.POST.get('password')

confirm\_password = *request*.POST['confirm\_password']

*if* password == confirm\_password:

*if* User.objects.filter(*username*=username).exists():

messages.info(*request*, 'User Already exist')

*return* redirect('register')

*else*:

*if* User.objects.filter(*email*=email).exists():

messages.info(*request*, 'Email Already exist')

*return* redirect('register')

*else*:

*request*.session['username'] = username

*request*.session['email'] = email

*request*.session['password'] = password

otp = []

num = str(random.randint(1000, 9999))

*for* x *in* num:

otp.append(x)

context = {'name': username, 'auth\_otp': otp}

html\_content = render\_to\_string(

'accounts/email.html', context)

text\_content = strip\_tags(html\_content)

send\_mail = EmailMultiAlternatives(

"Account email verification",

text\_content,

settings.EMAIL\_HOST\_USER,

[email]

)

send\_mail.attach\_alternative(html\_content, 'text/html')

send\_mail.send()

*# user\_obj = User(username=username, email=email)*

*# user\_obj.set\_password(password)*

*# user\_obj.save()*

profile = Profile(

*user*=email, *auth\_otp*=num)

profile.save()

*return* redirect('verify')

*return* render(*request*, 'accounts/register.html')

*# login route*

def login(*request*):

*if* *request*.user.is\_authenticated:

*return* redirect('home')

*else*:

*if* *request*.method == 'POST':

username = *request*.POST['username']

password = *request*.POST['password']

user = auth.authenticate(*username*=username, *password*=password)

*if* user is not None:

auth.login(*request*, user)

*return* redirect('home')

*else*:

messages.info(*request*, "Invalid Username or Password")

*return* redirect('login')

*return* render(*request*, 'accounts/login.html')

*# verify route*

def verify(*request*):

*if* *request*.user.is\_authenticated:

*return* redirect('home')

*else*:

*if* *request*.method == 'POST':

count = Profile.objects.filter(

*user*=*request*.session['email']).count()

profile = Profile.objects.filter(

*user*=*request*.session['email'])

*if* *request*.POST.get('auth\_otp') == profile.values('auth\_otp')[count-1]['auth\_otp']:

user = User.objects.create\_user(

*username*=*request*.session['username'], *password*=*request*.session['password'], *email*=*request*.session['email'])

user.save()

auth.login(*request*, user)

profile.delete()

context = {'name': *request*.session['username']}

html\_content = render\_to\_string(

'accounts/confirm\_email.html', context)

text\_content = strip\_tags(html\_content)

send\_mail = EmailMultiAlternatives(

"Registration Successfull",

text\_content,

settings.EMAIL\_HOST\_USER,

[*request*.session['email']]

)

send\_mail.attach\_alternative(html\_content, 'text/html')

send\_mail.send()

*return* redirect('home')

profile.delete()

messages.info(*request*, "Invalid otp or email register again!")

*return* redirect('register')

*return* render(*request*, 'accounts/verify.html')

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

def logout(*request*):

auth.logout(*request*)

*return* redirect('login')

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

def dashboard(*request*):

*return* render(*request*, 'accounts/dashboard.html')

1. **Pages**

All the routs related to navigate inside web page is handled by this app.

**urls.py**

*from* django.urls *import* path

*from* . *import* views

urlpatterns = [

path('', views.home, *name*='home'),

path('camera', views.camera, *name*='camera'),

path('result', views.result, *name*='result'),

path('screenshot', views.screenshot, *name*='screenshot'),

]

**views.py**

*from* skintone\_app.settings *import* MEDIA\_ROOT, MEDIA\_URL

*from* django.conf *import* settings

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

*from* base64 *import* b64decode

*from* django.shortcuts *import* redirect, render

*from* .models *import* \*

*import* uuid

*from* .skintone *import* \*

def home(*request*):

*return* render(*request*, 'pages/home.html')

def camera(*request*):

*if* *request*.user.is\_authenticated:

*return* render(*request*, 'pages/camera.html')

*else*:

*return* redirect('login')

def upload\_webcam\_blob(*blob*, *id*):

*with* open(f'media/uploads/{*id*}.png', 'wb') *as* fh:

*# Get only revelant data, deleting "data:image/png;base64,"*

        data = *blob*.split(',', 1)[1]

        fh.write(b64decode(data))

*return* fh

def screenshot(*request*):

*if* *request*.method == "POST":

        id = str(uuid.uuid4())

        img = *request*.POST.get('image')

        upload\_webcam\_blob(img, id)

        link = f"uploads/{id}.png"

        image = Image\_upload(*img\_id*=id, *image*=link)

        image.save()

*request*.session['id'] = id

*request*.session['image'] = img

*return* redirect('result')

*else*:

*return* redirect('camera')

def result(*request*):

    id = *request*.session['id']

    img = Image\_upload.objects.filter(*img\_id*=id).first()

*# url = f"{settings.BASE\_DIR}/media/{img.image.url}"*

    url = *request*.session['image']

    result = skintone(url)[0]

*return* render(*request*, 'pages/result.html', {'image': img, 'result': result})

1. **Upload Image App**

It contains all the routes related to image uploads.

**urls.py**

*from* django.urls *import* path

*from* . *import* views

urlpatterns = [

    path('', views.upload, *name*='upload'),

]

**views.py**

*from* pages.skintone *import* skintone

*from* pages.views *import* result

*from* django.shortcuts *import* render

*from* .models *import* Image\_data

*from* .form *import* ImageForm

*# Create your views here.*

*import* base64

def image\_to\_data\_url(*filename*):

    ext = *filename*.split('.')[-1]

    prefix = f'data:image/{ext};base64,'

*with* open(*filename*, 'rb') *as* f:

        img = f.read()

*return* prefix + base64.b64encode(img).decode('utf-8')

def upload(*request*):

*if* *request*.method == "POST":

        form = ImageForm(*data*=*request*.POST, *files*=*request*.FILES)

*if* form.is\_valid():

            form.save()

            obj = form.instance

            link = f"./{obj.image.url}"

            url = image\_to\_data\_url(link)

            result = skintone(url)[0]

*return* render(*request*, "upload/upload.html", {"obj": obj, 'result': result})

*else*:

        form = ImageForm()

        img = Image\_data.objects.all()

*return* render(*request*, "upload/upload.html", {"img": img, "form": form})

1. **Contact Us App**

It contains all the routes related to contact us functionality.

**urls.py**

*from* django.urls *import* path

*from* . *import* views

urlpatterns = [

    path('', views.contact, *name*='contact'),

]

**views.py**

*from* django.shortcuts *import* redirect, render

*from* django.http *import* HttpResponse

*from* .models *import* Contactus

*# Create your views here.*

def contact(*request*):

*if* *request*.method == 'POST':

        name = *request*.POST.get('Name')

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

        subject = *request*.POST.get('Subject')

        message = *request*.POST.get('Message')

        contact = Contactus(*name*=name, *email*=email,

*subject*=subject, *message*=message)

        contact.save()

*return* render(*request*, 'pages/thankyou.html')

*return* redirect('home')

1. **Required Modules**

for using this app in you local environment you must have following modules installed in your system.

asgiref==3.3.4

astroid==2.4.2

autopep8==1.5.5

certifi==2020.12.5

colorama==0.4.4

cycler==0.10.0

dj-database-url==0.5.0

Django==3.2.2

django-heroku==0.3.1

gunicorn==20.1.0

imutils==0.5.4

isort==5.6.4

joblib==1.0.1

kiwisolver==1.3.1

lazy-object-proxy==1.4.3

matplotlib==3.4.2

mccabe==0.6.1

numpy==1.20.3

opencv-python==4.5.2.52

opencv-python-headless==4.5.2.52

Pillow==8.2.0

psycopg2==2.8.6

psycopg2-binary==2.8.6

pycodestyle==2.7.0

pylint==2.6.0

pyparsing==2.4.7

python-dateutil==2.8.1

pytz==2021.1

scikit-learn==0.24.2

scipy==1.6.3

six==1.15.0

skia-python==87.1

sqlparse==0.4.1

threadpoolctl==2.1.0

toml==0.10.2

whitenoise==5.2.0

wincertstore==0.2

wrapt==1.12.1

Also we have provided deployed it on Heroku server, us the following Url for test this out

[https://whispering-cliffs-35654.herokuapp.com](https://whispering-cliffs-35654.herokuapp.com/)

1. **How To Run**
   1. After installing all required modules you have to run following command from the project directory

python manage.py migrate (only for first time)

* 1. and then run following command

python manage.py runserve

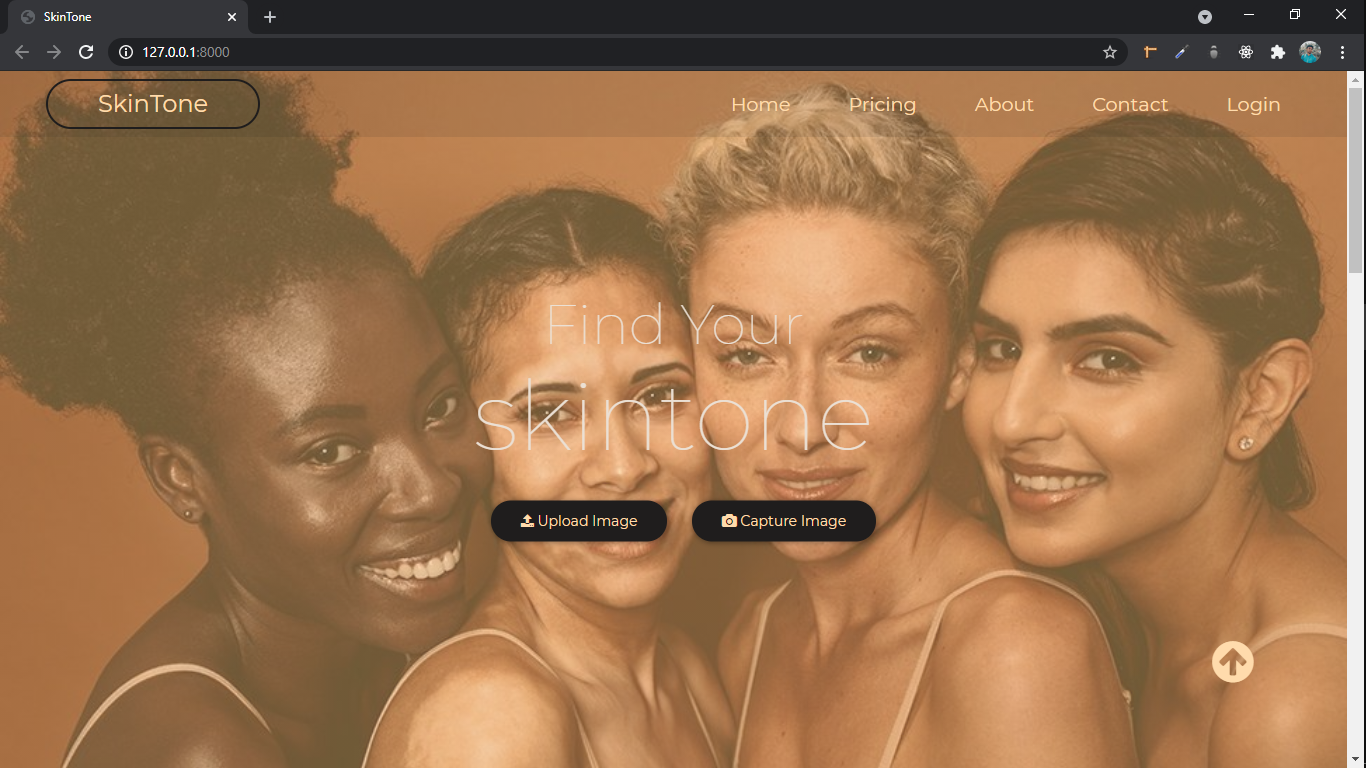
* 1. Just type on your browser

localhost:8000

or

<http://127.0.0.1:8000/>

Now you see a web page like this:



**For the admin panel**

Just type on your browser

localhost:8000/admin

or

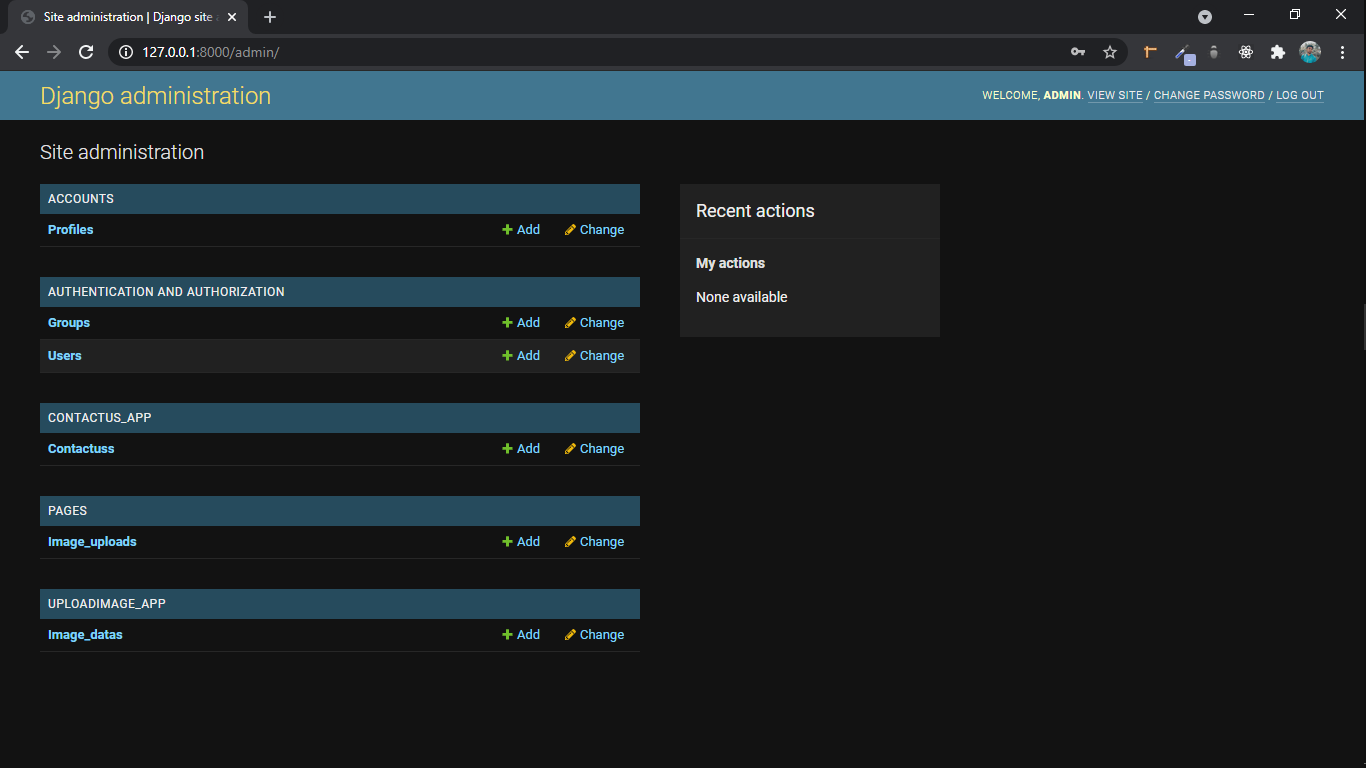
<http://127.0.0.1:8000/admin>

It will you for username and password use the following username and password

Username: admin

Password: admin@1234

Now you see following web page:



For live Testing

<https://whispering-cliffs-35654.herokuapp.com>

**Thank You**

Team B9