1. *Login, SignUp, Change Password, Edit Profile*

- As you're gonna implement login and sign up first, create a file in your app's directory with urls.py.

```
BasicApp/
├── coins/
│   ├── __init__.py
│   ├── admin.py
│   ├── apps.py
│   ├── migrations/
│   │   └── __init__.py
│   ├── models.py
│   ├── serializers.py
│   ├── tests.py
│   ├── urls.py                    <<urls file should be placed here
│   └── views.py
├── manage.py
└── BasicProject/
    ├── __init__.py
    ├── asgi.py
    ├── settings.py
    ├── urls.py
    └── wsgi.py
└── media/
```

First change your project's url to the link for app's urls.py:

```
from django.contrib import admin

from django.urls import path, include

from django.conf import settings

from django.conf.urls.static import static

from rest_framework import routers

from coins.views import CoinViewSet, CoinSearchView, coins_table, coin_details
```

```
# Create a router for registering viewsets
router = routers.DefaultRouter()

# Register CoinViewSet with the router
router.register(r'coins', CoinViewSet)

# Define URL patterns
urlpatterns = [
    # Admin site URL
    path('admin/', admin.site.urls),
    # API endpoints for coins using the router
    path('api/', include(router.urls)),
    path('coins/search/<path:path_params>/', CoinSearchView.as_view(), name='coin-search'),
    path("", include(("coins.urls", "coins"), "coins")),
]
# Serve media files in DEBUG mode
if settings.DEBUG:
    urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

After that , you can create the urls.py in app's directory well :

```
from django.urls import path, include
from django.contrib.auth import views as auth_views

app_name = 'coins'

urlpatterns = [
```

```
    path("", home, name="home"),
    path("signup/", authView, name="authView"),
    path("accounts/", include("django.contrib.auth.urls")),
]
```

This will be your urls.py which we set the url for login and signup which we imported django's in-built authentication views for authentication.

Next, we need to configure the views for it to get our customised template for login and sign process

You can modify the views.py as

```
from django.shortcuts import render, redirect, get_object_or_404  # Import render function from Django
from django.urls import reverse
from rest_framework import status  # Import status codes from Django REST Framework
from rest_framework.response import Response  # Import Response class from Django REST Framework
from rest_framework import viewsets  # Import viewsets from Django REST Framework
from .models import Coin  # Import the Coin model
from .serializers import CoinSerializer  # Import the CoinSerializer
from rest_framework.views import APIView
import re
from django.contrib.auth.forms import AuthenticationForm, UserCreationForm
from django.contrib.auth.decorators import login_required
from django.contrib.auth.mixins import LoginRequiredMixin
def home(request):
    return render(request, 'home.html', {})
def authView(request):
    if request.method == "POST":
        form = UserCreationForm(request.POST or None)
```

```
        if form.is_valid():
            form.save()
            return redirect(reverse("coins:login"))
    else:
        form = UserCreationForm()
    return render(request, "registration/signup.html", {"form": form})
```

And also place the redirect url in settings.py that,

```
LOGIN_REDIRECT_URL = "coins:home"
LOGOUT_REDIRECT_URL = "coins:login"
```

Next, you can place the custom view page for our redirects and home page

First place base.html in your templates/ directory for better view in all pages

Before that install bootstrap4 package in your root directory using the command,

```
pip install django-bootstrap4
```

And also place the “bootstrap4” in your settings.py in INSTALLED APPS

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login Registration Demo</title>
    <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet">
    <link rel="preconnect" href="https://fonts.googleapis.com">
```

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

    <link
href="https://fonts.googleapis.com/css2?family=Mulish:wght@400;700&disp
lay=swap" rel="stylesheet">

</head>

<body style="font-family: 'Mulish', sans-serif;">



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

 <div class="container">

    <a class="navbar-brand" href="{% url 'coins:home' %}">Django
Mine</a>

    <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbarNav" aria-controls="navbarNav"
aria-expanded="false" aria-label="Toggle navigation">

      <span class="navbar-toggler-icon"></span>

    </button>

    <div class="collapse navbar-collapse" id="navbarNav">

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

        <li class="nav-item">

          <a class="nav-link" href="{% url 'coins:home' %}">Home</a>

        </li>

        {% if user.is_authenticated %}

        <li class="nav-item">

          <a class="nav-link" href="{% url 'coins:view_profile' %}">My
Profile</a>

        </li>

        <li class="nav-item">

          <a class="nav-link" href="{% url 'coins:edit_profile' %}">Edit
Profile</a>

        </li>

        <li class="nav-item">

          <a class="nav-link" href="{% url 'coins:password_change'
%}">Change Password</a>
```

```
        </li>
        <li class="nav-item">
          <a class="nav-link" href="{% url 'coins:logout' %}">Logout</a>
        </li>
        {% else %}
        <li class="nav-item">
          <a class="nav-link" href="{% url 'coins:login' %}">Login</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="{% url 'coins:authView' %}">Sign Up</a>
        </li>
        {% endif %}
      </ul>
    </div>
  </div>
</nav>


<div class="container">
  {% block content %}{% endblock content %}
</div>


<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js">
</script>
</body>
</html>
```

(Note: comment the links like change password, profile, edit profile as of now since its not being implemented in this manual, you will be doing in upcoming instructions)

Now you can do the home.html in same templates/ directory itself :

```
{% extends "base.html" %}

{% load bootstrap4 %}

{% block content %}

<div class="container mt-5">

  <div class="row justify-content-center">

    <div class="col-md-8">

      <div class="card">

        <div class="card-header bg-primary text-white">

          <h5 class="mb-0">Welcome to Our Website</h5>

        </div>

        <div class="card-body">

          {% if user.is_authenticated %}

          <h3>Hello, {{ user.username }}!!</h3>

          {% else %}

          <h3>Hello, Guest!!</h3>

          {% endif %}

          <p class="mt-3">Thank you for visiting our site. We're glad to have you here!</p>

          <p>Explore our content and discover what we have to offer.</p>

          <div class="text-center mt-4">

            {% if user.is_authenticated %}

            <a href="{% url "coins:logout" %}" class="btn btn-danger btn-lg">Logout</a>

            <a href="{% url "coins:create_coin" %}" class="btn btn-success btn-lg ml-2">Create Coin</a>

            {% else %}

            <a href="{% url "coins:login" %}" class="btn btn-primary btn-lg">Login</a>

            {% endif %}

          </div>
```

```
            </div>
        </div>
    </div>
    <span> </span><span> </span>
    {% if user.is_authenticated and coins%}
    <h1 class="text-center mb-4"><strong>Coins Table</strong></h1>
    <div class="table-responsive">
        <table class="table table-striped table-hover border border-2 text-center align-middle">
            <thead class="table-dark">
                <tr>
                    <th>Name</th>
                    <th>Image</th>
                    <th>Description</th>
                    <th>Year</th>
                    <th>Country</th>
                    <th>Material</th>
                    <th>Weight</th>
                    <th>Starting Bid</th>
                    <th>Coin Status</th>
                    <th>Action</th>
                </tr>
            </thead>
            <tbody style="background: linear-gradient(90deg, white, #9bf8cd);">
                {% for coin in coins %}
                <tr>
                    <td>{{ coin.coin_name }}</td>
                    <td>
                        {% if coin.coin_image %}
```

```
                    <img src="{{ coin.coin_image.url }}" alt="{{ coin.coin_name }}" style="max-width: 100px;">
                    {% else %}
                        No Image
                    {% endif %}
                </td>
                <td>{{ coin.coin_desc }}</td>
                <td>{{ coin.coin_year }}</td>
                <td>{{ coin.coin_country }}</td>
                <td>{{ coin.coin_material }}</td>
                <td>{{ coin.coin_weight }}</td>
                <td>{{ coin.starting_bid }}</td>
                <td>{{ coin.coin_status }}</td>
                <td><a href="{% url 'coin-details' coin_id=coin.pk %}" class="btn btn-primary">View Details</a></td>
            </tr>
            {% endfor %}
        </tbody>
    </table>
    {% endif %}
</div>
</div>
{% endblock content %}
```

(Note : Comment the implementation of coin creation , coins table as of now since its not implemented in this manual yet, you will be going through upcoming instruction to do that.)

templates/registration/login.html :

```
{% extends "base.html" %}
```

```
{% load bootstrap4 %}

{% block content %}

 <div class="container">

   <div class="row justify-content-center">

     <div class="col-md-6">

       <div class="card mt-5">

         <div class="card-body">

           <h3 class="card-title" style="font-weight: bold;">Login</h3>

           <form method="POST">

             {% csrf_token %}

             {% bootstrap_form form %}

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

           </form>

           <p class="mt-3">Don't have an account? <a href="{% url
"coins:authView" %}" class="signup-link">Signup</a></p>


         </div>

       </div>

     </div>

   </div>

 </div>

{% endblock content %}
```

templates/registration/signup.html

```
{% extends "base.html" %}

{% load bootstrap4 %}

{% block content %}

 <div class="container">
```

```
    <div class="row justify-content-center">
      <div class="col-md-6">
        <div class="card mt-5">
          <div class="card-body">
            <h3 class="card-title" style="font-weight: bold;">Sign Up</h3>
            <form method="POST">
              {% csrf_token %}
              {% bootstrap_form form %}
              <button type="submit" class="btn btn-primary">Submit</button>
            </form>
            <p class="mt-3">Already have an account? <a href="{% url "coins:login" %}" class="signup-link">Login</a></p>
          </div>
        </div>
      </div>
    </div>
  </div>
{% endblock content %}
```

Now you can run the server command in the terminal with “python manage.py runserver”. You will able to get the url http://127.0.0.1:8000/, if you don't have any error logs in terminal.

Now you will be able to login and sign up facilities with django’s inbuilt form validation and handling the requests.

- Next we are gonna implement change password functionality,

  Go to your app’s urls.py and update the url as this :

```
from django.urls import path, include
```

```
from django.contrib.auth import views as auth_views

from coins.views import authView, home, custom_password_change, custom_password_change_done

from django.urls import reverse_lazy


app_name = 'coins'


urlpatterns = [

    path("", home, name="home"),

    path("signup/", authView, name="authView"),

    path("accounts/", include("django.contrib.auth.urls")),

    path("password_change/", custom_password_change, name="password_change"),

    path("password_change/done/", custom_password_change_done, name="password_change_done"),

]
```

Update your views.py as these :

```
from django.shortcuts import render, redirect, get_object_or_404  # Import render function from Django

from django.urls import reverse

from rest_framework import status  # Import status codes from Django REST Framework

from rest_framework.response import Response  # Import Response class from Django REST Framework

from rest_framework import viewsets  # Import viewsets from Django REST Framework

from .models import Coin  # Import the Coin model

from .serializers import CoinSerializer  # Import the CoinSerializer

from rest_framework.views import APIView

import re

from django.contrib.auth.forms import AuthenticationForm, UserCreationForm, PasswordChangeForm
```

```
from django.contrib.auth.decorators import login_required

from django.contrib.auth import update_session_auth_hash

from django.contrib.auth.views import PasswordChangeView,
PasswordChangeDoneView

from django.urls import reverse_lazy

from django.contrib.auth.mixins import LoginRequiredMixin

from django.contrib import messages

from django.urls import reverse

@login_required

def custom_password_change(request):

    if request.method == 'POST':

        form = PasswordChangeForm(request.user, request.POST)

        if form.is_valid():

            user = form.save()

            update_session_auth_hash(request, user)  # Important to keep the user
logged in

            messages.success(request, 'Your password was successfully updated!')

            return redirect(reverse('coins:password_change_done'))

        else:

            messages.error(request, 'Please correct the error below.')

    else:

        form = PasswordChangeForm(request.user)

    return render(request, 'registration/change-password.html', {'form': form})


@login_required

def custom_password_change_done(request):

    return render(request, 'registration/password-done.html')
```

Now we can move on to the templates which to be placed in templates/registration/change-password.html

```
{% extends "base.html" %}

{% load bootstrap4 %}

{% block content %}

<div class="container">

  <div class="row justify-content-center">

      <div class="col-md-6">

          <div class="card mt-5">

              <div class="card-body">

                  <h3 class="card-title" style="font-weight: bold;">Change Password</h3>

                  <form method="POST">

                      {% csrf_token %}

                      {% bootstrap_form form %}

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

                  </form>

              </div>

          </div>

      </div>

  </div>

</div>

{% endblock content %}
```

templates/registration/password-done/html

```
{% extends "base.html" %}

{% load bootstrap4 %}


{% block content %}

 <div class="container">
```

```
    <div class="row justify-content-center">
        <div class="col-md-6">
            <div class="card mt-5">
                <div class="card-body">
                    <h3 class="card-title" style="font-weight: bold;">Password Changed Successfully</h3>
                    <p>Your password has been changed successfully.</p>
                    <div class="text-center mt-3">
                        <a href="{% url 'coins:home' %}" class="btn btn-secondary">Home</a>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
{% endblock content %}
```

- Next, we are gonna implement the profile page and edit profile ,

  First update your app's urls.py

```
from django.urls import path, include
from django.contrib.auth import views as auth_views
from coins.views import authView, home, custom_password_change, custom_password_change_done
from django.urls import reverse_lazy
from .views import view_profile, edit_profile

app_name = 'coins'

urlpatterns = [
```

```
    path("", home, name="home"),

    path("signup/", authView, name="authView"),

    path("accounts/", include("django.contrib.auth.urls")),

    path("password_change/", custom_password_change,
name="password_change"),

    path("password_change/done/", custom_password_change_done,
name="password_change_done"),

    path("profile/", view_profile, name="view_profile"),

    path("profile/edit/", edit_profile, name="edit_profile"),

]
```

Update your views.py as like this :

```
from django.shortcuts import render, redirect, get_object_or_404  # Import
render function from Django

from django.urls import reverse

from rest_framework import status  # Import status codes from Django REST
Framework

from rest_framework.response import Response  # Import Response class from
Django REST Framework

from rest_framework import viewsets  # Import viewsets from Django REST
Framework

from .models import Coin, Profile  # Import the Coin model

from .serializers import CoinSerializer  # Import the CoinSerializer

from rest_framework.views import APIView

import re

from django.contrib.auth.forms import AuthenticationForm,
UserCreationForm, PasswordChangeForm

from django.contrib.auth.decorators import login_required

from django.contrib.auth import update_session_auth_hash

from django.contrib.auth.views import PasswordChangeView,
PasswordChangeDoneView

from django.urls import reverse_lazy

from django.contrib.auth.mixins import LoginRequiredMixin

from django.contrib import messages
```

```
from django.urls import reverse
from .forms import ProfileForm
@login_required
def view_profile(request):
    profile = Profile.objects.get_or_create(user=request.user)[0]
    return render(request, 'profile.html', {'profile': profile})

@login_required
def edit_profile(request):
    profile = Profile.objects.get_or_create(user=request.user)[0]
    if request.method == 'POST':
        form = ProfileForm(request.POST, request.FILES, instance=profile)
        if form.is_valid():
            form.save()
            messages.success(request, 'Your profile was successfully updated!')
            return redirect(reverse('coins:view_profile'))
        else:
            messages.error(request, 'Please correct the error below.')
    else:
        form = ProfileForm(instance=profile)
    return render(request, 'edit_profile.html', {'form': form})
```

You need to update your models also for this , as this needs another class model :

```
from django.db import models
from django.contrib.auth.models import User

class Coin(models.Model):
    # Define choices for coin status
```

```
    STATUS_CHOICES = (
        ('Select', 'Select'),  # Placeholder option
        ('available', 'Available'),
        ('sold', 'Sold'),
        ('pending', 'Pending'),
    )

    coin_id = models.AutoField(primary_key=True)  # Auto-incrementing primary key
    coin_image = models.ImageField(upload_to='coin_images/', null=True, blank=True)  # Image field to store coin image
    coin_name = models.CharField(max_length=100)  # Char field for coin name
    coin_desc = models.TextField()  # Text field for coin description
    coin_year = models.IntegerField()  # Integer field for coin year
    coin_country = models.CharField(max_length=50)  # Char field for coin country
    coin_material = models.CharField(max_length=50)  # Char field for coin material
    coin_weight = models.FloatField()  # Float field for coin weight
    starting_bid = models.FloatField()  # Float field for starting bid
    coin_status = models.CharField(max_length=50, choices=STATUS_CHOICES)  # Char field with choices for coin status
    created_by_id = models.IntegerField(null=True, blank=True)

    def __str__(self):
        return self.coin_name  # Return the coin name as its string representation

class Profile(models.Model):
    user = models.OneToOneField(User, on_delete=models.CASCADE)
    bio = models.TextField(max_length=500, blank=True)
    location = models.CharField(max_length=100, blank=True)
```

```
    website = models.URLField(max_length=200, blank=True)


    def __str__(self):
        return self.user.username  # Return the username as the string representation
```

Next, don't forget to make migrations and migrate it using "python manage.py makemigrations" and "python manage.py migrate"

Since, we are customising our profile form, we need to adjust our forms.py, so create it in your app's directory :

```
from django import forms
from django.contrib.auth.models import User
from .models import Profile


class ProfileForm(forms.ModelForm):
    username = forms.CharField(max_length=150, required=False)
    first_name = forms.CharField(max_length=30, required=False)
    last_name = forms.CharField(max_length=150, required=False)
    email = forms.EmailField(max_length=254, required=False)


    class Meta:
        model = Profile
        fields = ['username', 'first_name', 'last_name', 'email', 'bio', 'location', 'website']


    def __init__(self, *args, **kwargs):
        super(ProfileForm, self).__init__(*args, **kwargs)
        if self.instance.user:
            self.fields['username'].initial = self.instance.user.username
            self.fields['first_name'].initial = self.instance.user.first_name
```

```
        self.fields['last_name'].initial = self.instance.user.last_name
        self.fields['email'].initial = self.instance.user.email

    def save(self, commit=True):
        profile = super(ProfileForm, self).save(commit=False)
        if self.instance.user:
            self.instance.user.username = self.cleaned_data['username']
            self.instance.user.first_name = self.cleaned_data['first_name']
            self.instance.user.last_name = self.cleaned_data['last_name']
            self.instance.user.email = self.cleaned_data['email']
            if commit:
                self.instance.user.save()
        if commit:
            profile.save()
        return profile
```

Next, we can move on to the templates, place the templates in itself :

templates/profile.html :

```
{% extends "base.html" %}
{% load bootstrap4 %}
{% block content %}
<div class="container mt-5">
    <div class="row justify-content-center">
        <div class="col-md-12">
            <div class="card">
                <div class="card-header bg-primary text-white">
                    <div class="row">
                        <div class="col-md-6">
```

```
                    <h4 class="mb-0">My Profile</h4>
                </div>
                <div class="col-md-6 text-right">
                    <a href="{% url 'coins:edit_profile' %}" class="btn btn-outline-light">Edit Profile</a>
                </div>
            </div>
        </div>
        <div class="card-body">
            <div class="row">
                <div class="col-md-6">
                    <div class="mb-3">
                        <label class="form-label"><strong>Username:</strong></label>
                        <p class="form-control-static">{{ profile.user.username }}</p>
                    </div>
                    <div class="mb-3">
                        <label class="form-label"><strong>First Name:</strong></label>
                        <p class="form-control-static">{{ profile.user.first_name }}</p>
                    </div>
                    <div class="mb-3">
                        <label class="form-label"><strong>Last Name:</strong></label>
                        <p class="form-control-static">{{ profile.user.last_name }}</p>
                    </div>
                </div>
                <div class="col-md-6">
                    <div class="mb-3">
```

```
                        <label
class="form-label"><strong>Email:</strong></label>

                        <p class="form-control-static">{{ profile.user.email
}}</p>

                    </div>

                    <div class="mb-3">

                        <label class="form-label"><strong>Bio:</strong></label>

                        <p class="form-control-static">{{ profile.bio }}</p>

                    </div>

                    <div class="mb-3">

                        <label
class="form-label"><strong>Location:</strong></label>

                        <p class="form-control-static">{{ profile.location }}</p>

                    </div>

                    <div class="mb-3">

                        <label
class="form-label"><strong>Website:</strong></label>

                        <p class="form-control-static"><a href="{{ profile.website
}}">{{ profile.website }}</a></p>

                    </div>

                </div>

            </div>

        </div>

    </div>

</div>

{% endblock content %}
```

templates/edit_profile.html

```
{% extends "base.html" %}
```

```
{% load bootstrap4 %}

{% block content %}

 <div class="container">

   <div class="row justify-content-center">

     <div class="col-md-10">

       <div class="card mt-5">

         <div class="card-body">

           <h3 class="card-title" style="font-weight: bold;">Edit Your Details</h3>

           <form method="POST">

             {% csrf_token %}

             {% bootstrap_form form %}

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

           </form>



         </div>

       </div>

     </div>

   </div>

 </div>

{% endblock content %}
```

- Next, we are gonna implement creating the coin from the frontend for logged in users :

  First update the app's urls.py :

```
from django.urls import path, include

from django.contrib.auth import views as auth_views

from coins.views import authView, home, custom_password_change, custom_password_change_done
```

```
from django.urls import reverse_lazy

from .views import view_profile, edit_profile, create_coin


app_name = 'coins'


urlpatterns = [

    path("", home, name="home"),

    path("signup/", authView, name="authView"),

    path("accounts/", include("django.contrib.auth.urls")),

    path("password_change/", custom_password_change,
name="password_change"),

    path("password_change/done/", custom_password_change_done,
name="password_change_done"),

    path("profile/", view_profile, name="view_profile"),

    path("profile/edit/", edit_profile, name="edit_profile"),

    path("create-coin/", create_coin, name="create_coin"),

]
```

Your project's urls.py :

```
from django.contrib import admin

from django.urls import path, include

from django.conf import settings

from django.conf.urls.static import static

from rest_framework import routers

from coins.views import CoinViewSet, CoinSearchView, coins_table,
coin_details


# Create a router for registering viewsets

router = routers.DefaultRouter()
```

```
# Register CoinViewSet with the router

router.register(r'coins', CoinViewSet)


# Define URL patterns

urlpatterns = [

    # Admin site URL

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

    # API endpoints for coins using the router

    path('api/', include(router.urls)),

    path('coins/search/<path:path_params>/', CoinSearchView.as_view(), name='coin-search'),

    path('coins/', coins_table, name='coins-table'),  # URL for the coins table HTML page

    path('coin/<int:coin_id>/', coin_details, name='coin-details'),  # URL for the coin details page

    path("", include(("coins.urls", "coins"), "coins")),

]

# Serve media files in DEBUG mode

if settings.DEBUG:

    urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

Next update your views.py

This is a full code for views.py , so make sure all contents are there :

```
from django.shortcuts import render, redirect, get_object_or_404  # Import render function from Django

from django.urls import reverse

from rest_framework import status  # Import status codes from Django REST Framework

from rest_framework.response import Response  # Import Response class from Django REST Framework

from rest_framework import viewsets  # Import viewsets from Django REST Framework
```

```
from .models import Coin, Profile  # Import the Coin model

from .serializers import CoinSerializer  # Import the CoinSerializer

from rest_framework.views import APIView

import re

from django.contrib.auth.forms import AuthenticationForm,
UserCreationForm, PasswordChangeForm

from django.contrib.auth.decorators import login_required

from django.contrib.auth import update_session_auth_hash

from django.contrib.auth.views import PasswordChangeView,
PasswordChangeDoneView

from django.urls import reverse_lazy

from django.contrib.auth.mixins import LoginRequiredMixin

from django.contrib import messages

from django.urls import reverse

from .forms import ProfileForm, CoinForm



class CoinViewSet(viewsets.ModelViewSet):  # Define a viewset for the Coin
model

    queryset = Coin.objects.all()  # Define the queryset to fetch all coin objects

    serializer_class = CoinSerializer  # Specify the serializer class to use for the
Coin model




class CoinSearchView(APIView):

    def get(self, request, *args, **kwargs):

        # Extract search parameters from the URL path

        path_params = kwargs.get('path_params')



        # Parse the path_params string into field-value pairs

        search_params = {}

        if path_params:
```

```
        # Split the path_params string by '/'
        path_params_list = path_params.split('/')

        # Ensure there are an even number of elements (field-value pairs)
        if len(path_params_list) % 2 == 0:
            for i in range(0, len(path_params_list), 2):
                search_params[path_params_list[i]] = path_params_list[i+1]

    # Filter Coin objects based on the provided search parameters
    coins = Coin.objects.all()
    for field, value in search_params.items():
        coins = coins.filter(**{field: value})

    if not coins:
        return Response({"message": "No coins found for the provided search parameters"}, status=404)

    # Serialize the filtered queryset
    serializer = CoinSerializer(coins, many=True, context={'request': request})
    return Response(serializer.data)

def coins_table(request):
  # Retrieve data from the Coin model
  coins = Coin.objects.all()

  # Render the HTML template and pass the data to it
  return render(request, 'coins_table.html', {'coins': coins})
def coin_details(request, coin_id):
  # Retrieve the coin object with the specified ID
```

```
    coin = get_object_or_404(Coin, pk=coin_id)

    # Render the HTML template for coin details and pass the coin object to it
    return render(request, 'coin_details.html', {'coin': coin})

def home(request):
    # Get the logged-in user's ID
    user_id = request.user.id

    # Filter the coins based on the logged-in user's ID
    coins = Coin.objects.filter(created_by_id=user_id)

    return render(request, 'home.html', {'coins': coins})

@login_required
def create_coin(request):
    if request.method == 'POST':
        form = CoinForm(request.POST, request.FILES)
        if form.is_valid():
            # Get the logged-in user
            logged_in_user = request.user

            # Assign the logged-in user's ID to the created_by_id field of the coin
            coin = form.save(commit=False)
            coin.created_by_id = logged_in_user.id
            coin.save()

            messages.success(request, 'Coin created successfully!')
```

```
            return redirect(reverse("coins:home"))
    else:
        form = CoinForm()
    return render(request, 'create_coin.html', {'form': form})


def authView(request):
    if request.method == "POST":
        form = UserCreationForm(request.POST or None)
        if form.is_valid():
            form.save()
            return redirect(reverse("coins:login"))
    else:
        form = UserCreationForm()
    return render(request, "registration/signup.html", {"form": form})


@login_required
def view_profile(request):
    profile = Profile.objects.get_or_create(user=request.user)[0]
    return render(request, 'profile.html', {'profile': profile})


@login_required
def edit_profile(request):
    profile = Profile.objects.get_or_create(user=request.user)[0]
    if request.method == 'POST':
        form = ProfileForm(request.POST, request.FILES, instance=profile)
        if form.is_valid():
            form.save()
            messages.success(request, 'Your profile was successfully updated!')
```

```
            return redirect(reverse('coins:view_profile'))
        else:
            messages.error(request, 'Please correct the error below.')
    else:
        form = ProfileForm(instance=profile)
    return render(request, 'edit_profile.html', {'form': form})


@login_required
def custom_password_change(request):
    if request.method == 'POST':
        form = PasswordChangeForm(request.user, request.POST)
        if form.is_valid():
            user = form.save()
            update_session_auth_hash(request, user)  # Important to keep the user logged in
            messages.success(request, 'Your password was successfully updated!')
            return redirect(reverse('coins:password_change_done'))
        else:
            messages.error(request, 'Please correct the error below.')
    else:
        form = PasswordChangeForm(request.user)
    return render(request, 'registration/change-password.html', {'form': form})


@login_required
def custom_password_change_done(request):
    return render(request, 'registration/password-done.html')
```

Update your forms.py

```
from django import forms
```

```
from django.contrib.auth.models import User
from .models import Profile, Coin

class ProfileForm(forms.ModelForm):
    username = forms.CharField(max_length=150, required=False)
    first_name = forms.CharField(max_length=30, required=False)
    last_name = forms.CharField(max_length=150, required=False)
    email = forms.EmailField(max_length=254, required=False)

    class Meta:
        model = Profile
        fields = ['username', 'first_name', 'last_name', 'email', 'bio', 'location',
'website']

    def __init__(self, *args, **kwargs):
        super(ProfileForm, self).__init__(*args, **kwargs)
        if self.instance.user:
            self.fields['username'].initial = self.instance.user.username
            self.fields['first_name'].initial = self.instance.user.first_name
            self.fields['last_name'].initial = self.instance.user.last_name
            self.fields['email'].initial = self.instance.user.email

    def save(self, commit=True):
        profile = super(ProfileForm, self).save(commit=False)
        if self.instance.user:
            self.instance.user.username = self.cleaned_data['username']
            self.instance.user.first_name = self.cleaned_data['first_name']
            self.instance.user.last_name = self.cleaned_data['last_name']
            self.instance.user.email = self.cleaned_data['email']
```

```
        if commit:
            self.instance.user.save()
        if commit:
            profile.save()
        return profile


class CoinForm(forms.ModelForm):
    class Meta:
        model = Coin
        fields = ['coin_image', 'coin_name', 'coin_desc', 'coin_year', 'coin_country', 'coin_material', 'coin_weight', 'starting_bid', 'coin_status']
```

templates/create_coin.html :

```
{% extends "base.html" %}
{% load bootstrap4 %}
{% block content %}
 <div class="container">
   <div class="row justify-content-center">
     <div class="col-md-10">
       <div class="card mt-5">
         <div class="card-body">
           <h3 class="card-title" style="font-weight: bold;">Create Coin <a href="{% url "coins:home" %}" class="btn btn-warning">Return</a></h3>
           <form method="POST" enctype="multipart/form-data">
             {% csrf_token %}
             {% bootstrap_form form %}
             <button type="submit" class="btn btn-primary">Submit</button>
           </form>
```

```
        </div>
      </div>
    </div>
  </div>
</div>
{% endblock content %}
```

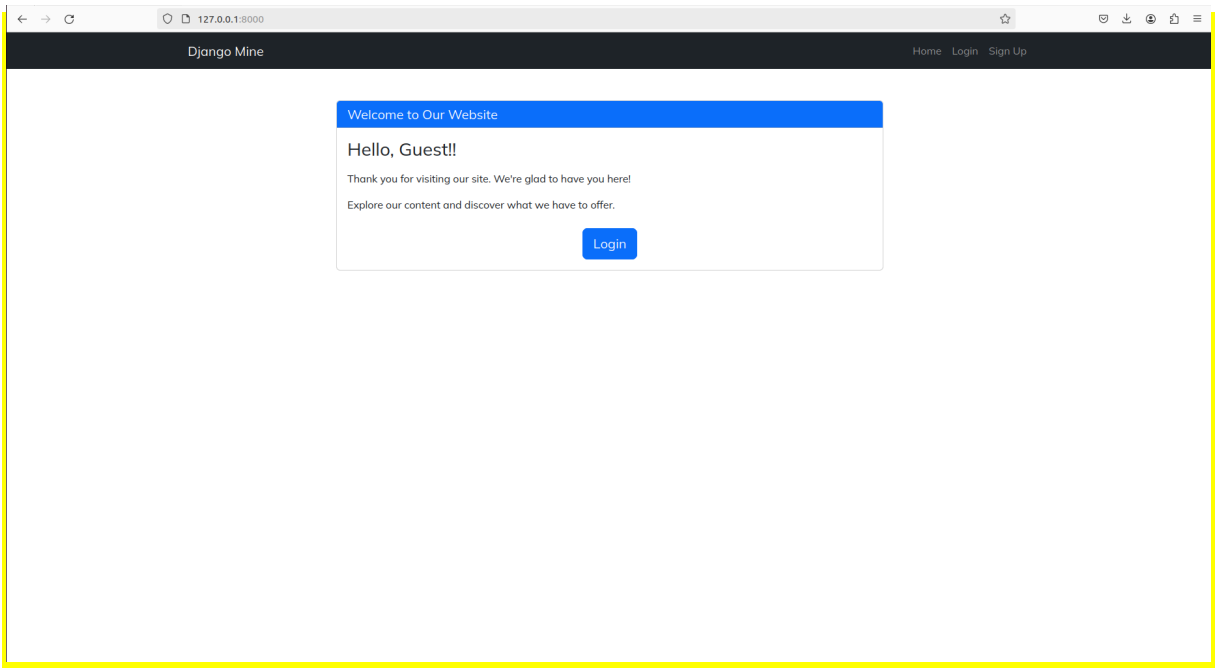
templates/coin_details.html

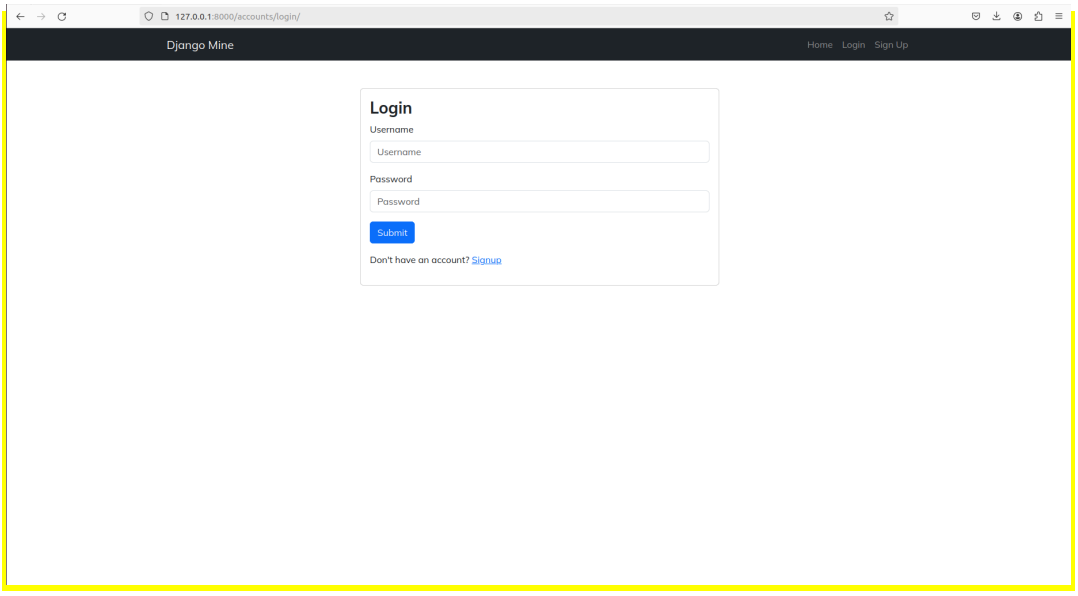
```
{% extends "base.html" %}
{% load bootstrap4 %}
{% block content %}
  <div class="container mt-5">
    <h1 class="text-center mb-4"><strong>COIN DETAILS</strong></h1>
    <div class="row justify-content-center">
      <div class="col-md-12">
        <div class="card border-0 shadow-lg" style="background-color: #c7e619;">
          <div class="card-body">
            <h3 class="card-title text-center mb-4">{{ coin.coin_name }}</h3>
            <div class="row">
              <div class="col-md-6">
                <p class="card-text"><strong>Year:</strong> {{ coin.coin_year }}</p>
                <p class="card-text"><strong>Country:</strong> {{ coin.coin_country }}</p>
                <p class="card-text"><strong>Material:</strong> {{ coin.coin_material }}</p>
                <p class="card-text"><strong>Weight:</strong> {{ coin.coin_weight }}</p>
                <p class="card-text"><strong>Starting Bid:</strong> {{ coin.starting_bid }}</p>
```

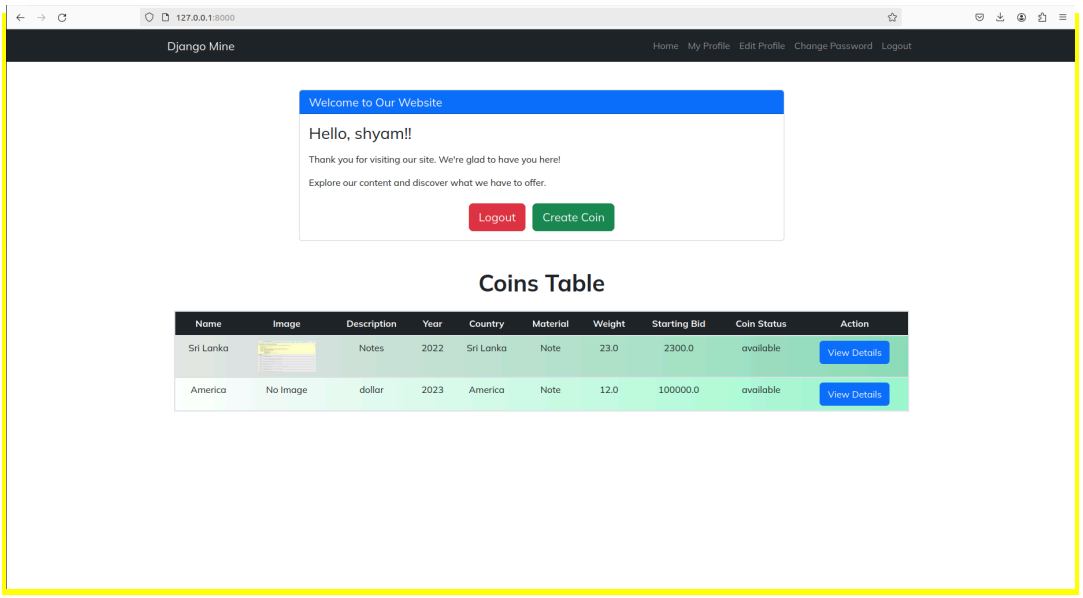
```
                        <p class="card-text"><strong>Coin Status:</strong> {{ coin.coin_status }}</p>

                        <a href="{% url 'coins:home' %}" class="btn btn-primary">Back to Home</a>

                    </div>

                    <div class="col-md-6">

                    {% if coin.coin_image %}

                        <p class="text-center"><img src="{{ coin.coin_image.url }}" alt="{{ coin.coin_name }}" class="img-fluid rounded"></p>

                    {% else %}

                        <p class="text-center">No Image</p>

                    {% endif %}

                        <p class="card-text"><strong>Description:</strong> {{ coin.coin_desc }}</p>

                    </div>

                </div>

            </div>

        </div>

    </div>

{% endblock content %}
```

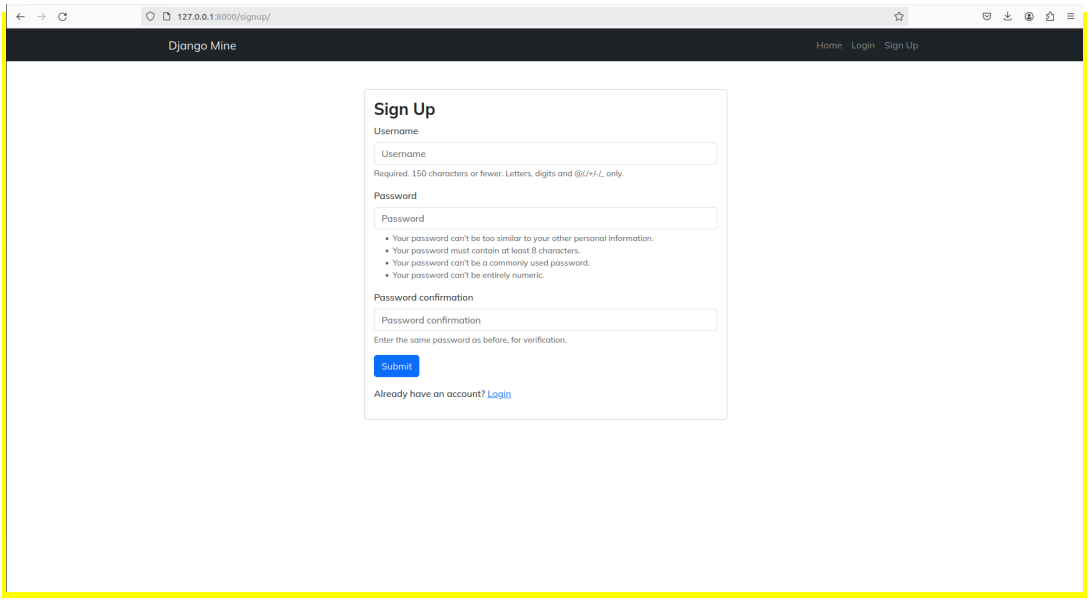
==========

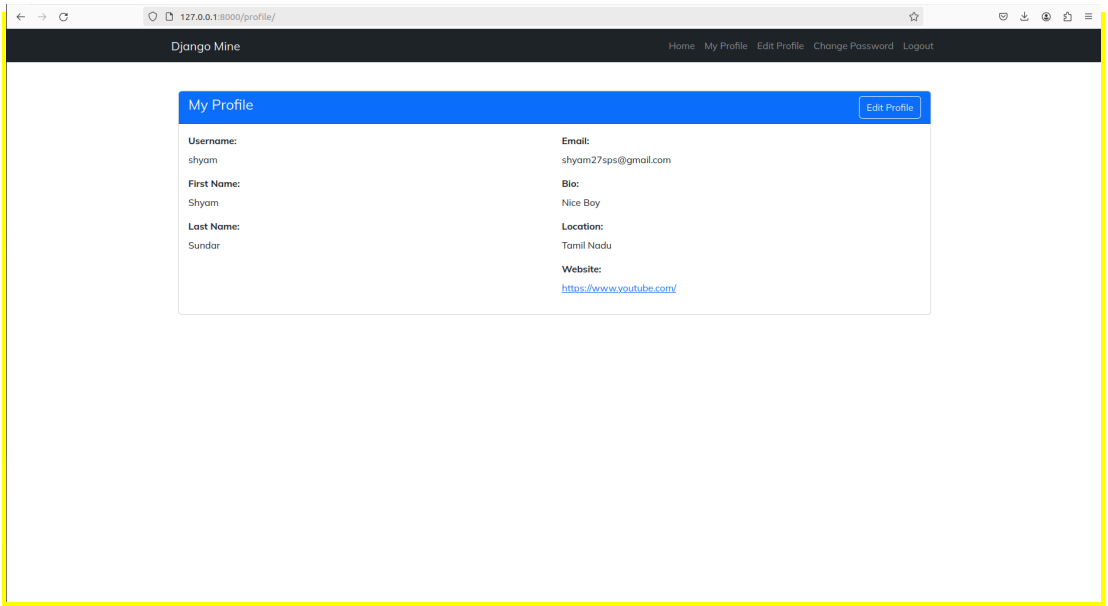
2. *Expected Images of Output :*

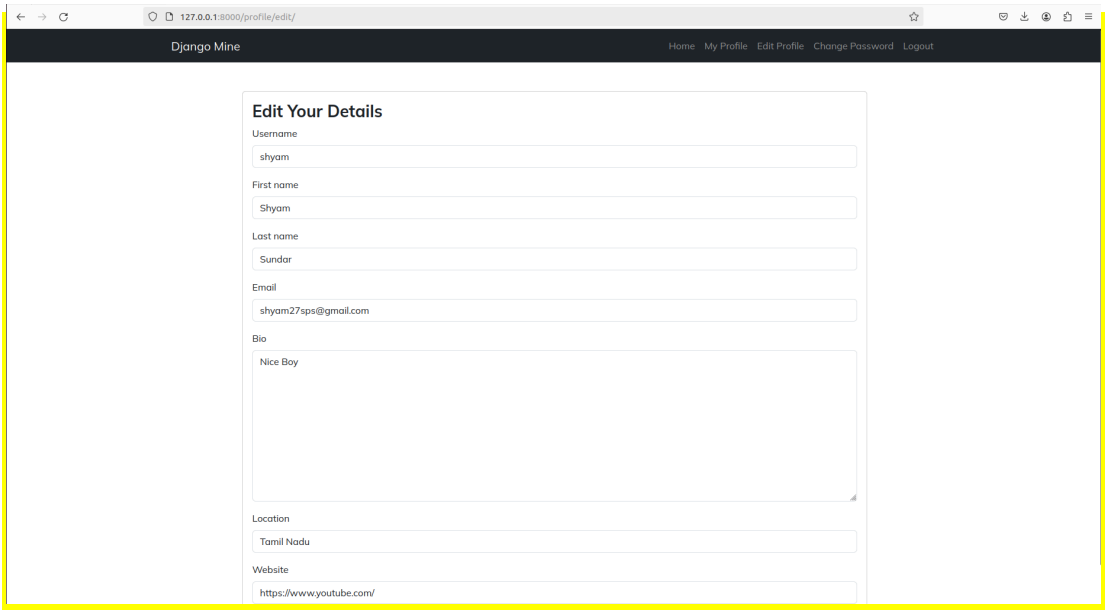


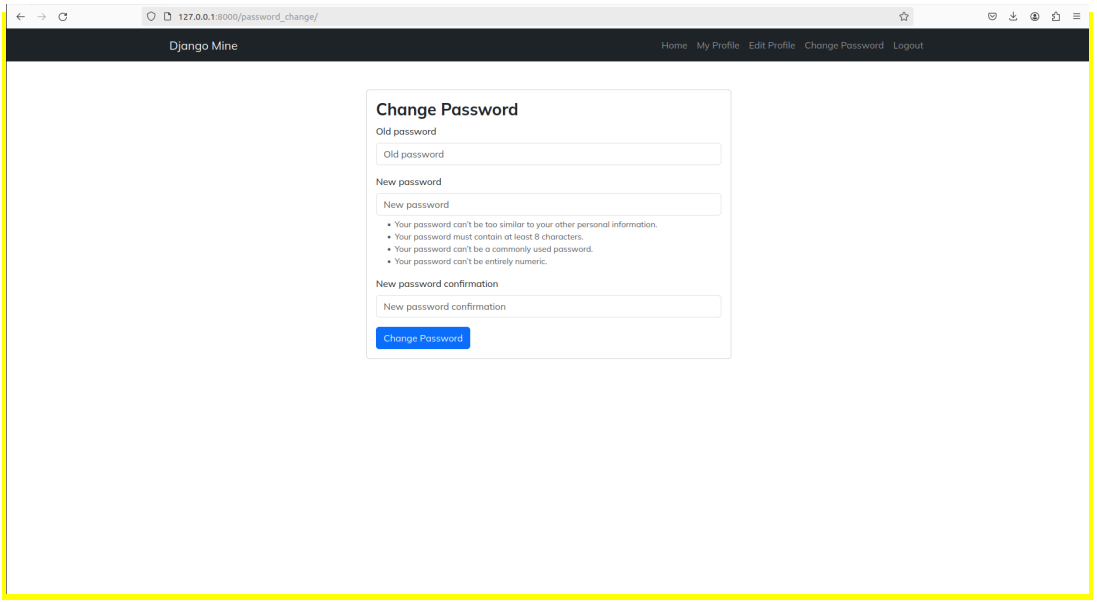


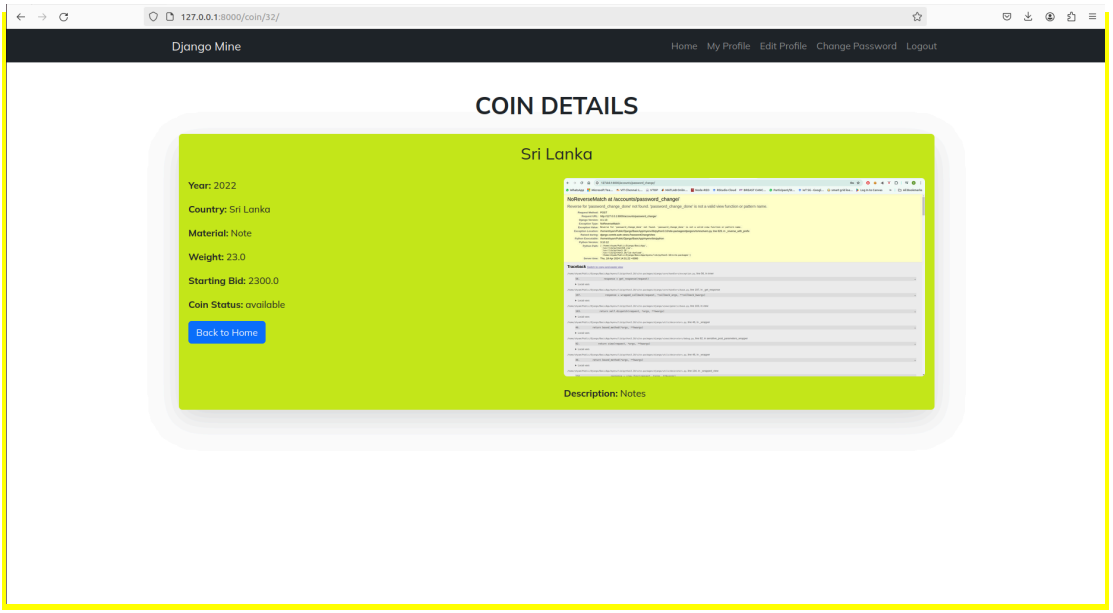

127.0.0.1:8000
Django Mine
Home My Profile Edit Profile Change Password Logout
Welcome to Our Website
Hello, shyam!!
Thank you for visiting our site. We're glad to have you here!
Explore our content and discover what we have to offer.
Logout
Create Coin
Coins Table
Name Image Description Year Country Material Weight Starting Bid Coin Status Action
Sri Lanka Notes 2022 Sri Lanka Note 23.0 2300.0 available View Details
America No Image dollar 2023 America Note 12.0 100000.0 available View Details


127.0.0.1:8000/signup/
Django Mine
Home Login Sign Up
Sign Up
Username
Username
Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.
Password
Password
Your password can't be too similar to your other personal information.
Your password must contain at least 8 characters.
Your password can't be a commonly used password.
Your password can't be entirely numeric.
Password confirmation
Password confirmation
Enter the same password as before, for verification.
Submit
Already have an account? Login


127.0.0.1:8000/profile/
Django Mine
Home My Profile Edit Profile Change Password Logout
My Profile
Edit Profile
Username:
shyam
Email:
shyam27sps@gmail.com
First Name:
Shyam
Bio:
Nice Boy
Last Name:
Sundar
Location:
Tamil Nadu
Website:
https://www.youtube.com/


127.0.0.1:8000/profile/edit/
Django Mine
Home My Profile Edit Profile Change Password Logout
Edit Your Details
Username
shyam
First name
Shyam
Last name
Sundar
Email
shyam27sps@gmail.com
Bio
Nice Boy
Location
Tamil Nadu
Website
https://www.youtube.com/


127.0.0.1:8000/password_change/
Django Mine
Home My Profile Edit Profile Change Password Logout
Change Password
Old password
Old password
New password
New password
Your password can't be too similar to your other personal information.
Your password must contain at least 8 characters.
Your password can't be a commonly used password.
Your password can't be entirely numeric.
New password confirmation
New password confirmation
Change Password


127.0.0.1:8000/coin/32/
Django Mine
Home My Profile Edit Profile Change Password Logout
COIN DETAILS
Sri Lanka
Year: 2022
Country: Sri Lanka
Material: Note
Weight: 23.0
Starting Bid: 2300.0
Coin Status: available
Back to Home
Description: Notes

==========