To start the project use this command

```
django-admin startproject crypto_portfolio_project
cd crypto_portfolio_project
```

To start the app use this command

```
python manage.py startapp portfolio
```

Now add this app to the **‘settings.py’**

```
INSTALLED_APPS = [
    "django.contrib.admin",
    "django.contrib.auth",
    "django.contrib.contenttypes",
    "django.contrib.sessions",
    "django.contrib.messages",
    "django.contrib.staticfiles",
    "portfolio",
]

STATICFILES_DIRS = [
    os.path.join(BASE_DIR, 'static'),
]
LOGOUT_REDIRECT_URL = 'holdings_list'  # Replace 'holdings_list'
with the appropriate URL name
```

```
# portfolio/models.py
from django.db import models
from django.contrib.auth.models import User



class CryptoCurrency(models.Model):
    name = models.CharField(max_length=100)
    symbol = models.CharField(max_length=10)

    def __str__(self):
        return self.name
```

```
class Holding(models.Model):
    cryptocurrency = models.CharField(max_length=100)
    quantity = models.DecimalField(max_digits=20, decimal_places=5)
    purchase_price = models.DecimalField(max_digits=20, decimal_places=2)
    purchase_date = models.DateField()

    def __str__(self):
        return f"{self.cryptocurrency} - {self.quantity}"



class Portfolio(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    image = models.CharField(max_length=100)
    cryptocurrency = models.CharField(max_length=100)
    symbol = models.CharField(max_length=10)
    quantity = models.DecimalField(max_digits=10, decimal_places=2)
    purchase_price = models.DecimalField(max_digits=10, decimal_places=2)
    purchase_date = models.DateField()

    def __str__(self):
        return f"{self.cryptocurrency} - {self.quantity}"

# portfolio/views.py
from django.contrib.auth.decorators import login_required
from django.contrib.auth.forms import AuthenticationForm
from django.contrib.auth import login, logout
from django.shortcuts import render, redirect
from django.shortcuts import render, redirect
from django.urls import reverse_lazy, reverse
from .utils import fetch_crypto_data

from .forms import RegistrationForm, AddCryptoForm

from .models import Portfolio

# register
def register(request):
    if request.method == 'POST':
        form = RegistrationForm(request.POST)
        if form.is_valid():
            user = form.save()
            login(request, user)
            # Redirect to home page after successful registration
            return redirect('holdings_list')
    else:
        form = RegistrationForm()
    return render(request, 'registration/register.html', {'form': form})

# login


def user_login(request):
    if request.method == 'POST':
        form = AuthenticationForm(request, data=request.POST)
        if form.is_valid():
            user = form.get_user()
            login(request, user)
            # Redirect to holdings list after successful login
            return redirect(reverse_lazy('holdings_list'))
    else:
        form = AuthenticationForm()
    return render(request, 'registration/login.html', {'form': form})

# logout
def user_logout(request):
    logout(request)
    # Redirect to the home page after logout
    return redirect(reverse('holdings_list'))

# homepage data
def holdings_list(request):
    crypto_data = fetch_crypto_data()
    holdings = [
        {
            "cryptocurrency": data["name"],
            "image": data["image"],
            "symbol": data["symbol"],
            "current_price": data["current_price"],
            "market_cap": data["market_cap"],
            "market_cap_rank": data["market_cap_rank"],
            "fully_diluted_valuation": data["fully_diluted_valuation"],
            "total_volume": data["total_volume"],
            "high_24h": data["high_24h"],
            "low_24h": data["low_24h"],
            "price_change_24h": data["price_change_24h"],
            "price_change_percentage_24h": data["price_change_percentage_24h"],
            "market_cap_change_24h": data["market_cap_change_24h"],
            "market_cap_change_percentage_24h":
data["market_cap_change_percentage_24h"],
            "circulating_supply": data["circulating_supply"],
            "total_supply": data["total_supply"],
            "max_supply": data["max_supply"],
            "ath": data["ath"],
            "ath_change_percentage": data["ath_change_percentage"],
            "ath_date": data["ath_date"],
            "atl": data["atl"],
            "atl_change_percentage": data["atl_change_percentage"],
            "atl_date": data["atl_date"],
            "roi": data["roi"],
            "last_updated": data["last_updated"],
            "purchase_price": 0,
            "purchase_date": None
        }
        for data in crypto_data
    ]
    return render(request, 'portfolio/holdings_list.html', {'holdings': holdings})

# add new crypto in their portfolio
@login_required
def add_to_portfolio(request):
    if request.method == 'POST':
        form = AddCryptoForm(request.POST)
        if form.is_valid():
            # Create a new Portfolio instance and associate it with the authenticated
user
            portfolio = Portfolio(
                user=request.user,
                image=form.cleaned_data['image'],
                cryptocurrency=form.cleaned_data['cryptocurrency'],
                symbol=form.cleaned_data['symbol'],
                quantity=form.cleaned_data['quantity'],
                purchase_price=form.cleaned_data['purchase_price'],
                purchase_date=form.cleaned_data['purchase_date']
            )
            # Save the portfolio instance
            portfolio.save()
            # Redirect to holdings list after adding the cryptocurrency
            return redirect('portfolio')
    else:
        form = AddCryptoForm()
    return render(request, 'portfolio/portfolio.html', {'form': form})

# own portfolio
@login_required
def portfolio(request):
    # Retrieve the user's portfolio data
    user_portfolio = Portfolio.objects.filter(user=request.user)
    return render(request, 'portfolio/portfolio.html', {'portfolio': user_portfolio})

import requests

def fetch_crypto_data():
    url = "https://api.coingecko.com/api/v3/coins/markets"
    params = {
        "vs_currency": "USD",
        "order": "market_cap_desc",
        "per_page": 100,
        "page": 1,
        "sparkline": False
    }
    response = requests.get(url, params=params)
    if response.status_code == 200:
        # print(response.json())
        return response.json()
    return None

# portfolio/forms.py

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

# forms.py
from django.contrib.auth.forms import UserCreationForm

class RegistrationForm(UserCreationForm):
    class Meta:
        model = User
        fields = ['username', 'password1', 'password2']

class AddCryptoForm(forms.Form):
    image = forms.CharField(label='image')
    cryptocurrency = forms.CharField(label='Cryptocurrency Name')
    symbol = forms.CharField(label='Symbol')
    quantity = forms.DecimalField(label='Quantity')
    purchase_price = forms.DecimalField(label='Purchase Price')
    purchase_date = forms.DateField(label='Purchase Date')

#crypto_portfolio_project\urls.py
from django.contrib import admin
from django.urls import path, include
from portfolio.views import holdings_list

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('portfolio.urls')),
    path('accounts/', include('django.contrib.auth.urls')),
    path('', holdings_list, name='holdings_list'),
]

# portfolio/urls.py
from django.urls import path
from . import views
from .views import register

urlpatterns = [
    path('', views.holdings_list, name='holdings_list'),
    path('add-to-portfolio/', views.add_to_portfolio, name='add_to_portfolio'),
    path('portfolio/', views.portfolio, name='portfolio'),
    path('register/', register, name='register'),
    path('logout/', views.user_logout, name='logout'),
    path('login/', views.user_login, name='login'),
    # Add URL patterns for other views
]

<!-- portfolio\templates\base.html -->
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>{% block title %}My Site{% endblock %}</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #1e1e1e; /* Dark background color */
            color: #fff; /* Light text color */
            margin: 0;
            padding: 0;
            text-align: center; /* Center align everything */
        }
        header, main, footer {
            display: inline-block; /* Display as inline-block for centering */
            text-align: left; /* Reset text-align for child elements */
        }
        header {
            background-color: #333; /* Darker background color for header */
            padding: 20px;
            width: 100%;
        }
        header h1 {
            margin: 0;
            font-size: 32px;
            color: #fff; /* Light text color for header */
        }
        nav ul {
            list-style-type: none;
            padding: 0;
            margin: 0;
        }
        nav ul li {
            display: inline;
            margin-right: 20px;
        }
        nav ul li a {
            text-decoration: none;
```

```
            color: #fff; /* Light text color for links */
            transition: color 0.3s;
        }
        nav ul li a:hover {
            color: #ddd; /* Lighter text color on hover */
        }
        main {
            padding: 20px;
            width: 100%;
        }
        footer {
            background-color: #333; /* Darker background color for footer */
            padding: 20px;
            width: 100%;
        }
        footer p {
            margin: 0;
            color: #fff; /* Light text color for footer */
        }
    </style>
</head>
<body>
    <header>
        <h1>My TRADING Portfolio</h1>
        <nav>
            <ul>
                <li><a href="{% url 'holdings_list' %}">Home</a></li>
                <li><a href="/portfolio/">My portfolio</a></li>
                <!-- Add more navigation links as needed -->
            </ul>
        </nav>
    </header>

    <main>
        {% block content %}
        {% endblock %}
    </main>

    <footer>
        <p>&copy; 2024 MySite. All rights reserved.</p>
    </footer>
</body>
</html>

<!-- portfolio\templates\portfolio\add_to_portfolio.html -->
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Add to Portfolio</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #222;
            color: #ddd;
            margin: 0;
            padding: 0;
        }
        .container {
            max-width: 500px;
            margin: 50px auto;
    background-color: #333;
    padding: 20px;
    border-radius: 5px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.5);
}
h1 {
    text-align: center;
    color: #fff;
}
form {
    margin-top: 20px;
}
label {
    display: block;
    margin-bottom: 5px;
    color: #bbb;
}
input[type="text"],
input[type="number"],
input[type="date"] {
    width: 100%;
    padding: 10px;
    margin-bottom: 10px;
    border: 1px solid #444;
    border-radius: 5px;
    background-color: #555;
    color: #ddd;
}
button[type="submit"] {
    background-color: #007bff;
    color: #fff;
```

```
            padding: 10px 20px;
            border: none;
            border-radius: 5px;
            cursor: pointer;
        }
        button[type="submit"]:hover {
            background-color: #0056b3;
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>Add Cryptocurrency to Portfolio</h1>
        <form method="post">
            {% csrf_token %}
            <label for="id_cryptocurrency">Cryptocurrency Name:</label>
            {{ form.cryptocurrency }}
            <label for="id_symbol">Symbol:</label>
            {{ form.symbol }}
            <label for="id_quantity">Quantity:</label>
            {{ form.quantity }}
            <label for="id_purchase_price">Purchase Price:</label>
            {{ form.purchase_price }}
            <label for="id_purchase_date">Purchase Date:</label>
            {{ form.purchase_date }}
            <button type="submit">Add to Portfolio</button>
        </form>
    </div>
</body>
</html>
```

```
<!-- portfolio\templates\portfolio\holdings_list.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>Cryptocurrency Holdings</title>
    <link rel="stylesheet" type="text/css" href="{% static 'portfolio/css/styles.css'
%}">
    <style>

/* Add dark theme */
body {
    background-color: #222; /* Dark background color for the body */
    color: #fff; /* Light text color */
}

.container {
    background-color: #222; /* Dark background color for the body */
    margin: 0 auto; /* Center align the container */
    max-width: 1200px; /* Set max width for better layout */
    padding: 20px; /* Add some padding for better spacing */
}

/* CSS styles to set the width and height of the image */
.crypto-container {
    display: flex;
    flex-wrap: wrap;
    justify-content: space-between;
    border-radius: 5px;
    box-shadow: 0 0 20px rgba(20, 87, 233, 0.5); /* Blue light shadow */
    background-color: #222; /* Dark background color for the container */
    padding: 20px; /* Add some padding for better spacing */
}

.crypto-card {
    width: calc(25% - 20px); /* Set the width of each card (25% of the container width with margins) */
    border: 1px solid #ccc;
    border-radius: 5px;
    padding: 10px;
    margin: 10px;
    cursor: pointer; /* Add cursor pointer for clickable effect */
    position: relative;
    z-index: 1;
    border-radius: 5px;
            box-shadow: 0 0 20px rgba(20, 87, 233, 0.5);
    transition: transform 0.3s ease-in-out;
    text-align: center; /* Align content to the center */
    background-color: #444; /* Dark background color for the card */
}

.crypto-image {
    width: 50px; /* Set the desired width */
    height: auto; /* Automatically adjust height to maintain aspect ratio */
}

.crypto-card.active {
    position: fixed;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    z-index: 999;
    background-color: rgba(0, 0, 0, 1); /* Semi-transparent black background */
    padding: 20px;
    box-sizing: border-box;
    overflow-y: auto; /* Allow scrolling if content exceeds screen height */
}

.crypto-card.active .crypto-image {
    width: 200px; /* Increase image size for modal */
}

.crypto-card.active .details {
    display: block; /* Show details section */
}

.details {
    display: none; /* Initially hide details section */
}

/* Style for the date input field */
input[type="date"],
input[type="number"] {
    width: 10%;
    padding: 8px;
    margin-top: 10px;
    margin-bottom: 20px;
    border: 1px solid #ccc;
    border-radius: 5px;
    box-sizing: border-box;
```

```
}

/* Style for the submit button */
button[type="submit"] {
    background-color: #446845;
    color: white;
    padding: 10px 20px;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    transition: background-color 0.3s;
}

button[type="submit"]:hover {
    background-color: #45a049;
}

/* Style for the close button */
.close-button {
    position: absolute;
    top: 10px;
    right: 10px;
    cursor: pointer;
    color: white;
    font-size: 24px;
}

/* Style for the navbar */
nav {
    background-color: #333; /* Dark background color for navbar */
    padding: 10px 20px;
    text-align: center;
}

nav ul {
    list-style-type: none;
    padding: 0;
    margin: 0;
}

nav ul li {
    display: inline;
    margin-right: 20px;
}

nav ul li a {
    text-decoration: none;
    color: #fff; /* Light text color for navbar links */
    transition: color 0.3s;
}

nav ul li a:hover {
    color: #ddd; /* Lighter text color on hover */
}
    </style>
    <script>
        // JavaScript function to toggle active class on card
        function toggleCard(cardId) {
            var card = document.getElementById(cardId);
            card.classList.toggle("active");
        }
        // JavaScript function to close the bulged-out screen
        function closeCard(cardId) {
            var card = document.getElementById(cardId);
            card.classList.remove("active");
        }

        // JavaScript function to prevent closing when clicking on the date field
        function stopPropagation(event) {
            event.stopPropagation();
        }
    </script>
</head>
<body>
    <nav>
        <ul>
            {% if user.is_authenticated %}
            <li><a href="{% url 'portfolio' %}">My Portfolio</a></li>
            <li><a href="{% url 'holdings_list' %}">Holdings</a></li>
            <li><a href="{% url 'logout' %}">Logout</a></li>
        {% else %}
            <li><a href="{% url 'login' %}">Login</a></li>
            <li><a href="{% url 'register' %}">Register</a></li>
        {% endif %}
        </ul>

    </nav>
    <div class="container">
        <h1>Cryptocurrency Holdings</h1>
        <div class="crypto-container">
            {% if holdings %}
                {% for holding in holdings %}
```

```
                    <div id="{{ holding.symbol }}" class="crypto-card"
onclick="toggleCard('{{ holding.symbol }}')">
                        <img src="{{ holding.image }}" alt="{{ holding.cryptocurrency }}
Image" class="crypto-image"> <!-- Cryptocurrency Image -->
                        <h2>{{ holding.cryptocurrency }}</h2> <!-- Cryptocurrency Name
-->
                        <p><strong>Symbol:</strong> {{ holding.symbol }}</p>
                        <p><strong>Current Price:</strong> ${{ holding.current_price
}}</p>
                        <div class="details">
                            <p><strong>Market Cap:</strong> ${{ holding.market_cap
}}</p>
                            <p><strong>Market Cap Rank:</strong> {{
holding.market_cap_rank }}</p>
                            <p><strong>Fully Diluted Valuation:</strong> ${{
holding.fully_diluted_valuation }}</p>
                            <p><strong>Total Volume:</strong> ${{ holding.total_volume
}}</p>
                            <form action="{% url 'add_to_portfolio' %}" method="post">
                                {% csrf_token %}
                                <input type="hidden" name="image" value="{{ holding.image
}}">
                                <input type="hidden" name="cryptocurrency" value="{{
holding.cryptocurrency }}">
                                <input type="hidden" name="symbol" value="{{
holding.symbol }}">
                                <input type="hidden" name="current_price" value="{{
holding.current_price }}">
                                <input type="date" name="purchase_date"
onclick="stopPropagation(event)">
```

```
                                <input type="number" name="quantity"
placeholder="Quantity" min="0" step="0.01" required
onclick="stopPropagation(event)">
                                <input type="number" name="purchase_price"
placeholder="Purchase Price" min="0" step="0.01" required
onclick="stopPropagation(event)">
                                <button type="submit">Add to Portfolio</button>
                            </form>
                            <!-- Close button for the bulged-out screen -->
                            <span class="close-button" onclick="closeCard('{{
holding.symbol }}')">&times;</span>
                            <!-- Add more details here as needed -->
                        </div>
                    </div>
                {% endfor %}
            {% else %}
                <p>No holdings found.</p>
            {% endif %}
        </div>
    </div>
</body>
</html>

<!-- portfolio\templates\portfolio\portfolio.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>User Portfolio</title>
    <link rel="stylesheet" type="text/css" href="{% static 'portfolio/css/styles.css'
%}">
    <style>
        /* CSS styles to set the width and height of the image */
        .crypto-image {
            width: 50px; /* Set the desired width */
            height: auto; /* Automatically adjust height to maintain aspect ratio */
        }

        /* Dark theme styles */
        body {
            background-color: #222; /* Dark background color for the body */
            color: #fff; /* Light text color */
            font-family: Arial, sans-serif; /* Specify a font for better readability */
        }

        h1 {
            text-align: center; /* Center align the heading */
            margin-top: 30px; /* Add some top margin for spacing */
        }

        .container {
            background-color: #222; /* Dark background color for the body */
            max-width: 800px; /* Limit the width of the container */
            margin: 0 auto; /* Center the container horizontally */
            padding: 20px; /* Add padding for content spacing */
            border-radius: 5px;
            box-shadow: 0 0 20px rgba(20, 87, 233, 0.5); /* Blue light shadow */
        }

        .crypto-container {
```

```
    display: flex;
    flex-wrap: wrap;
    justify-content: space-between;

}

.crypto-card {
    width: calc(25% - 20px); /* Set the width of each card (25% of the container width with margins) */
    border: 1px solid #444; /* Dark border color for the card */
    border-radius: 5px;
    padding: 10px;
    margin: 10px;
    cursor: pointer; /* Add cursor pointer for clickable effect */
    position: relative;
    z-index: 1;
    transition: transform 0.3s ease-in-out;
    border-radius: 5px;
    box-shadow: 0 0 20px rgba(20, 87, 233, 0.5);
    text-align: center; /* Align content to the center */
    background-color: #333; /* Dark background color for the card */
}

.crypto-card:hover {
    transform: translateY(-5px); /* Add a slight hover effect */
}

p {
    margin: 5px 0; /* Adjust paragraph margin for better spacing */
}
/* Style for the navbar */
```

```
nav {
    background-color: #333; /* Dark background color for navbar */
    padding: 10px 20px;
    text-align: center;
}

nav ul {
    list-style-type: none;
    padding: 0;
    margin: 0;
}

nav ul li {
    display: inline;
    margin-right: 20px;
}

nav ul li a {
    text-decoration: none;
    color: #fff; /* Light text color for navbar links */
    transition: color 0.3s;
}

nav ul li a:hover {
    color: #ddd; /* Lighter text color on hover */
}
    </style>
</head>
<body>
    <nav>
        <ul>
```

```
            {% if user.is_authenticated %}
            <li><a href="{% url 'portfolio' %}">My Portfolio</a></li>
            <li><a href="{% url 'holdings_list' %}">Holdings</a></li>
            <li><a href="{% url 'logout' %}">Logout</a></li>
        {% else %}
            <li><a href="{% url 'login' %}">Login</a></li>
            <li><a href="{% url 'register' %}">Register</a></li>
        {% endif %}
        </ul>

    </nav>
    <h1>My Portfolio</h1>
    {% if portfolio %}
        <div class="container">
            <div class="crypto-container">
                {% for item in portfolio %}
                    <div class="crypto-card">
                        <img src="{{item.image}}" alt="{{ item.cryptocurrency }} Image"
class="crypto-image"> <!-- Cryptocurrency Image -->
                        <h2>{{ item.cryptocurrency }}</h2> <!-- Cryptocurrency Name -->
                        <p><strong>Symbol:</strong> {{ item.symbol }}</p>
                        <p><strong>Quantity:</strong> {{ item.quantity }}</p>
                        <p><strong>Purchase Price:</strong> ${{ item.purchase_price
}}</p>
                        <p><strong>Purchase Date:</strong> {{ item.purchase_date
}}</p>
                    </div>
                {% endfor %}
            </div>
        </div>
    {% else %}
```

```
        <p>No holdings found.</p>
    {% endif %}
</body>
</html>

<!-- templates/registration/login.html -->
{% extends 'base.html' %}
{% block content %}
  <h2>Login</h2>
  <form method="post">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit">Login</button>
    <!-- Add a link/button to register -->
  </form>
  <a href="{% url 'register' %}">Register</a>
{% endblock %}

{% extends 'base.html' %}
{% block content %}
<h2>Register</h2>
<form method="post">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit">Register</button>
</form>
    <a href="{% url 'login' %}">Login</a>
{% endblock %}

/* portfolio/static/portfolio/css/styles.css */
```

```css
body {
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    background-color: #f8f9fa;
    margin: 0;
    padding: 0;
}

.container {
    max-width: 800px;
    margin: 20px auto;
    padding: 20px;
    background-color: #fff;
    border-radius: 10px;
    box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);
}

h1 {
    text-align: center;
    color: #007bff;
    margin-bottom: 20px;
}

table {
    width: 100%;
    border-collapse: collapse;
}

th, td {
    border: 1px solid #dee2e6;
    padding: 10px;
    text-align: left;
}

th {
    background-color: #f8f9fa;
    color: #6c757d;
}

tr:nth-child(even) {
    background-color: #f8f9fa;
}

tr:hover {
    background-color: #e9ecef;
}

p {
    text-align: center;
    color: #6c757d;
    font-style: italic;
    margin-top: 20px;
}
```

```
from django.contrib import admin
from .models import Holding
from .models import CryptoCurrency
# Register your models here.
admin.site.register(Holding)
admin.site.register(CryptoCurrency)
```

Running the Project

Run these commands to apply the migrations:

```
python3 manage.py makemigrations
python3 manage.py migrate
```

Createsuperuser using the below command

```
python3 manage.py createsuperuser
```

Run the server with the help of following command:

```
python3 manage.py runserver
```