Description

Create API endpoints for a Brine like exchange which supports the following operations

1. Users should be able to sign-up.
2. Each user can have a unique wallet for every currency. That is, the user will have one separate wallet for `btc`, one separate wallet for `eth` and another wallet for `doge`. You'll have to create the following endpoints for wallets:
    a. list all the wallets.
    b. credit and debit the wallet balances.
3. Users should be able to create and cancel orders
    a. Users should be able to place orders only when the user holds enough funds in his wallet. For example: say a user holds 10 eth in his wallet and tries to place an order to sell 11 eth. This should not be allowed and should throw an error.
    b. Every order has a state associated with it. `state` could be one of the following:- pending, cancel, success
    c. Every order has a "sell_currency" which stores the name of a cryptocurrency coin (e.g. btc), and a "buy_currency" which stores the name of the fiat currency(e.g. usdc).
    d. The order has a "side" value for deciding which currency the user is trying to sell.

       For example, if the side is "sell", it means that the user wants to sell the "sell_currency" and buy the "buy_currency".

       Similarly, if the side is "buy", it means that the user wants to sell the "buy_currency" and buy the "sell_currency".
4. Bonus Task:
    a. Make sure all wallet operations keep the wallet balance consistent.
    b. Suppose let's say you have 2 eth coins in your eth wallet, and you placed a sell order for 2 eth. But this order is still in the "pending" state(not processed), and so the user still has the funds. And you again come on the exchange and try to place a sell order for 2 eth, this should not be allowed right. Try to implement in such a way that this should not be allowed.
    c. Admin users can view the dashboard which includes the following stats:
        i. Volume of orders placed in a particular time interval for every currency. Endpoint should take the following params: from_time, to_time

Tasks

Create the endpoints in the tasks as per the definition.

Task 1

Sign up

Register a user on the app.

```
endpoint: `/user/signup/` => POST
request_data: email, password
response:
{
   “status”: “success”,
   “message”: “User successfully signed up”
   “payload”: {
         “user_id”: “”
   }

}
```

Task 2

Deposit

Add currency to user’s wallet

```
endpoint: `/wallets/deposit/` => POST
request_data: currency, amount
response :
{
   “status”:”success”,
   “msg”: “successfully deposited funds”,
   “payload”: {
         “balance”:<>,
         “currency”:<>,
   }
}
```

Withdrawal

Remove currency from user’s wallet

```
endpoint: `/wallet/withdrawal/` => POST
request_data: currency, amount
```

```
response: {
   “status”:”success”,
   “msg”: “successfully withdrawn funds”,
   “payload”: {
         “balance”:<>,
         “currency”:<>,
   }
}
```

Balances

Return user’s wallet balance for all currencies.

```
endpoint: `/wallet/balances/` => GET
request_data: None
response: {
   “status”:”success”,
   “msg”: “successfully fetched funds”,
   “payload”: [
         {
                  “balance”:<>,
                  “currency”:<>
         },{
                  “balance”:<>,
                  “currency”:<>,
         }...,
         ]
   }
}
```

Task 3

Create order

```
endpoint - /order/create/ => POST
request_data - currency_buy, currency_sell, price, volume, side
response - {
         “Status”: “success”,
         “Message”: “Successfully created”,
         “Payload”: {
               "id": 904,
               "side": "sell",
               "price": "29580.51",
               “Volume”: 1,
               "state": "pending",
               "buy_currency": "btc",
               “sell_currency”: “usdt”,
               "created_at": "2022-05-27T12:36:57+02:00",
               "updated_at": "2022-05-27T12:36:57+02:00",
         }
}
```

Cancel Order

```
endpoint - /order/cancel/ => PUT
request_data - order_id
response: {
        “Status”: “success”,
        “Message”: “Successfully cancelled,
        “Payload”: “”
}
```

Bonus Task

Dashboard

```
endpoint: `/dashboard/` => GET
request_data:
        {
           “from_time”: “”,
           “to_time”: “”

        }
response:
        {
           “status”: “success”,
           “message”: “Data fetched successfully”,
           “payload”: {
               “btc”: “10”,
              “eth”: “2022”,
              “doge”: “1241241”
           }

        }
```