Hello how are you

my name is augstin

I'm studying information systems engineering, which is like computer science

I have 3 years of experience with Django and 2 with javascript using react

I can tell you a bit about the last 2 projects I did recently

the first one is a social media app i made it using django rest framework for backend and react for frontend

Basically in the backend I overwrote the default user model to add more properties like user photo, biography and other things.

After that I put the AUTH_USER_MODEL in settings.py to point to my custom user model and also set the media folder to handle the images

Use SimpleJWT for authentication where you configure it in settings, basically specify things like the accees token, refresh tokens and algorithm

Then also set lso corsheaders to allow origins,

basically it serves so that django allows to connect to third applications. So I put the react server which was the localhost on port 3000

and create 7 endpoints just for the users app which were:

Login,

Register,

Update the current user authenticated

upload profile picture

Get user current user profile authenticated

Get all the users

get only one user with an id

And I also created a blog app where I had 2 models that one was the blogs and the other the comments, which have a parent-child relationship

In this app there were 6 endpoints which were:

get all the users

gett only one blog with the id

post to blog

update or put a blog with the id

delete a blog

and comment on blog

All the endpoints of this app are protected with the permission_classes, which I put IsAuthenticated,

so that to do anything in the app they need to be authenticated

excluding by login and register

then with the relation to the blog model, I put some extra permissions so that only the user who created the publication can manipulate it like editing or deleting a blog

After that, as always, I tested the API with Postman and I saw that everything was ok, so I continued with the forntend

create an app with react, install tailwindcss, axios and redux

Use Redux because I think the app can be more scalable and organized

Basically I structured the app where I had a folder for the constants where I basically specified the request, success and fail in the state of the app, then I had another folder where the

reducers were where I specified the state in more detail such as the payload, if it has an error and if it is loading

that if you look at it well it fits the request, that is if it is loading, if it fails, that is if there is an error and if it is satisfactory it will have a payload

Then the other folder was the actions, where it called the endpoint with axios, where basically in all of them it had a constant called config where it passed the token so that it had permissions to send a request to the backend

Then I had the last folder which is where the components of the app were

where I used the useSelector hook to grab that state from the store and there I could create different scenarios that were if there was an error, I would show it on the screen as an alert, if I was loading I would show a loader and I also used the Usedispatch hook where I would dispatch the actions

Then something was also using protectedroutes where to access all routes except for login, register and landing, you need to have an access token, this gave the app more security so that unauthenticated users could somehow access

Basically that is the social network with Django rest Framework and React, which I also have a video on YouTube that explains how to build it

Then I also have an E-Commerce with django Rest FrameWork, React and Solidity, which is basically an app that has dynamic payments in the blockchain with Etherium

It is built with the same pattern of the social network

I created 3 apps in Dajngo, which were the Orders, Products, and Users

For the orders I had 3 models that were the Orders themselves, I had a children that was the Order Item, where each item of Orders was that contained product, name and other relevant information

And I also had another model that was a shipping information

where I had a OneToOne relationship with order

In the serializers.py I made a logic to be able to access the Order Items of the order and also the shipping information... All this was inside a class called OrderSerializer

I had x endpoints where the first one was

add a new order where basically the order was going to be associated with a user, at that endpoint the shipping information was also going to be created, each product was going to be registered with a total price and a product was also going to be subtracted from the stock

Then another endpoint would be to get the order by ID

where only the usaer adnim or the owner of the order would be able to see the information

the other endpoint was to get the commands from the currently authenticated user

Then there was the enpoint to access all commands, which is basically an ednpoint only for an admin user

Then there was the Products app where I had a Product model that had a parent child relationship with Reviews

Endpoints were the first to get a search through a search bar

The other endpoint was to get all computer products on a first-come, first-served basis.

the other is to get a product with the id

And then the other endpoint was to create a review where you can only leave one review per user with a rating from 1 to 5 with a comment

The endpoints that I am going to comment on now all need admin permissions

Then another was to create a new product

The next one was to update a product

to remove a product

To upload the product image

For the users app, use the same logic as the previous project, which was basically authentication with a simple JWT with a custom user model.

in these endpoints there were 3 that needed admin user

what were they

Read, Delete, Update users without their consent

I had an endpoint to login, register

to see the profile with your data

another to update the current authenticated user and

to upload profile picture

In total there were 8 users, 8 products and 4 for orders, which makes a total of 20 endpoints

Then for the fontend I used Redux and tailwind css with the same structure as the previous project where I had a folder with actions, reducers and constants

The interesting part is in the smart contracts to be able to make dynamic payments with etherium

Basically I created another backend where all the logic was to be able to make transactions in the blockchain,

Create a folder called smart_conttract, install HardHat to run solidity locally and test smart contracts

Create a file called transaction.sol where all the logic and properties of the smart contract are, such as the ID of the transaction, from which wallet to which wallet it is transferred and the amount

Then create a new app in Alchamy in the Goerly test network, which is like fake etherium for developers. Once the app is created in Alchamy, create a key that uses it in hardhat.config.js where it is specified in which network it is going to use and also a private key from my metamask wallet

Then create a file called deploy.js which was basically going to deploy the smart contract

Once with all that I ran the deploy.js file

which basically will return me a number which is where the transaction is deployed and also a file called transaction.json which is basically an ABI, this file will take me to the frontend as well, which is basically all the information needed to interact with the etherium network once that was done had everything to make transactions in the frontend with React

Once all that was done, I created a file called Transactions that was going to unstructure the etherium object, verify that this metamask is installed, it will have the client's address dynamically, as well as the wallet that will receive the Ethers, and the amount to also turn off dynamically

Then associate the transaction number with the requested order in question, so that the user has access to their orders with the amount and transaction number in their profile, where they also have a link to see it in an ether scan.

Obviously the dmin user has access to all transactions made in the application

And basically that's the second project I've been working on,

I really want to make a tutorial on youtube on how to build this application