Final project

All the following without the docs

## **Part 2 Done**

## **Part 3 Done**

## **Part 4 Done**

* Two kinds of forms you can know from angular docs
* Import reactiveformsmodule inside the component that you want
* Formgroup, formcontrol, ( validtors is class and you import it and )
* ? -> is null safety , so we put it after the formname.get, search for it
* And inside validatots you write it inside the brancics and inside it you write your validtor inside a sqaure brackets as array
* And there is built in functions and if you want to write an specific way you can write it by pattern (search for it)
* Formgroup property in the form element in the html and take the binding square brackets
* Don't submit your form by clicking but link it when the form is completed
* (ngsubmit) -> search for it
* Sending the form value in the ngsubmit
* And then creating p cotain the error that could happen
* Then doing ngif for every error could happen and shown it to the client
* \*ngif="signupform.get('name')?.geterror('minlength')"
* Signupform.get('name')?.touched
* Search for it
* Invalid in the angulor

## **Part 5 Which Is 6 To Them Bec 5 Is Not Found Done**

* So, I import the httpclient module inside the module app and I import httpclient inside my auth service and import observable, this import from rxjs and we use it instead of promise and the other way
* Observable does not work until you do subscribe and this thing take three parameter and (next) and (error) and (complete) search for them
* So inside it, I wrote about next part and error part and inside the next part I write a condition if the signupform is done successfully or not and not will be the other condition in else
* Programmatic routing, search for it
* There is router class and you import it to use and inside router there is navigate which leads you to put your path that you want and you write it insed ( ) then inside it write [ ] and then inside / ( the path name )

## **Part 6 Which Is 7 To Them Bec 6 Is Not Found Done**

* About coping the part of sign up part to login part but changing few things and make sure that your url is ( part ( sign in ) )

## **Part 7 Which Is 8 To Them Bec 7 Is Not Found Done**

* So there is a thing is called token and this is very unique as like id but more speicifed so every user has his own token and from this token you can access so many things as your cart or payment or other and there is
* Jwt is library that backend developers use it to encode the code and there is two of the libaray one of encode and other for decode
* And we can use it to save it in the local storage to prove that the user is logged in
* B3d kda nfk el decode w nw7t el data gwa el user data
* And we see if the userdata is full so the user is logged and if userdata is null that means that the user is not logged