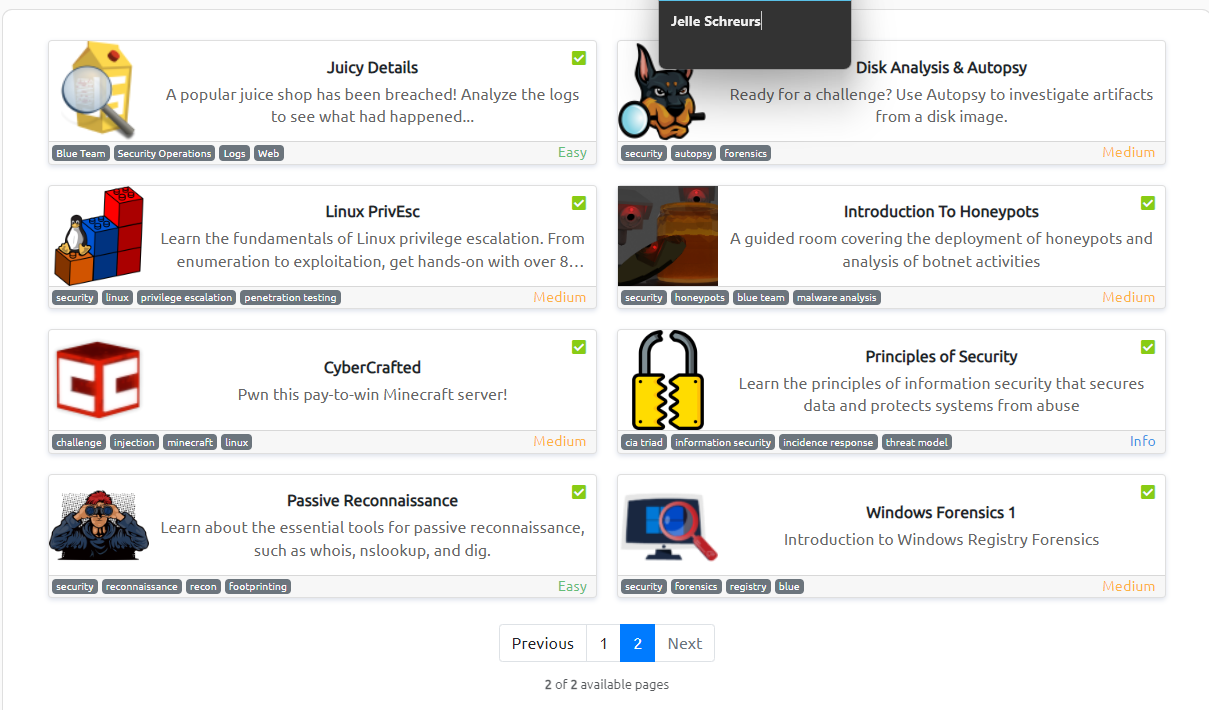
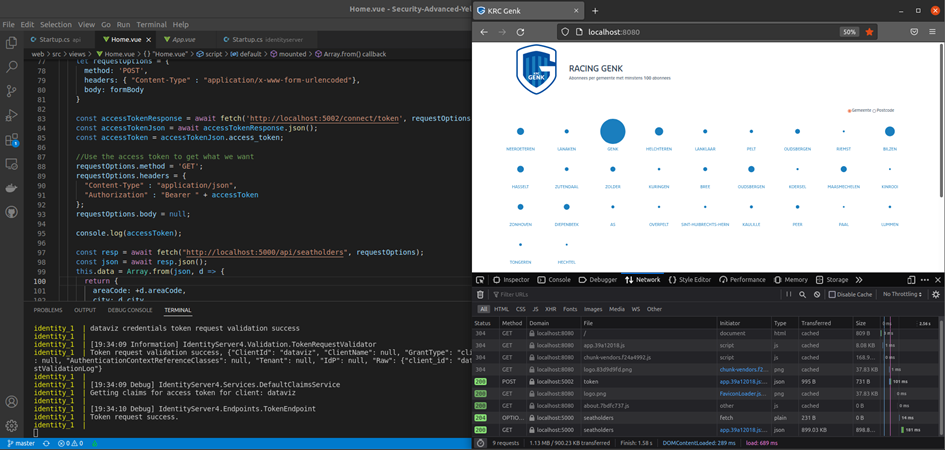
**Yellow Lab**   
  
White, Red en Blue oefeningen  
  
Hierbij de screenshots van de oefeningen op TryHackMe. Er zijn hier meerdere kamers aanwezig wegens het behalen van de nodige sterrren voor de verdere security opdracht.  
  
Graphical user interface, application

Description automatically generated  
  
Yellow Lab  
  
   
  
***- Zoek eens op wat CORS betekent. Probeer ook te achterhalen hoe we dat zouden kunnen oplossen.***  
 - CORS (Cross-Origin Resource Sharing) is een systeem, bestaande uit verzenden van HTTP headers, dit  
 bepaald of browsers frontend Javascript code blokkeren van de geaccepteerde respons voor  
 het cross-origin request. (docs, 2022)  
  
 - zoeken op de foutmelding: “Access-Control-Allow-Origin.NET”  
 - > <https://docs.microsoft.com/en-us/aspnet/core/security/cors?view=aspnetcore-6.0>

***Denk er aan dat de services draaien binnen een docker container. De web api draait binnen de docker container op `http://api:80`. De web app draait binnen de container op `http://web:80`. Betekent dit dat request eigenlijk gestuurd wordt van `http://web:80` naar `http://api:80`? Of van `http://localhost:8080` naar `http://api:80`? Of van `http://localhost:8080` naar `http://localhost:5000`?***  
 - <http://localhost:8080>  
  
***Lees de informatie over client credentials nog eens na op*** [***https://auth0.com/docs/get-started/authentication-and-authorization-flow/client-credentials-flow***](https://auth0.com/docs/get-started/authentication-and-authorization-flow/client-credentials-flow) ***.***

***Waarom hebben we deze signature nodig? Wat is deze signature?*** - Deze signature geeft ons toegang tot de data. Het is dus een soort van key.  
  
***Waarom staat het `client\_secret` niet in de token?*** - Omwille van veiligheid wordt dit hier niet meegegeven, het acces-token opvragen gebeurd  
 via de web-applicatie.

***Lost het CORS probleem tussen de identity server en de web app op.***  
 - dit baseert zich op de eerste wijzigingen doorgevoerd op de voorgaande cors.

***Zou het gebruik van HTTPS ons helpen om de client secret in de web app te verbergen?*** - Het verbruik dat tussen de cliënt en de server plaats vind zal worden geëncrypteerd door  
 HTTPS, het kan ons dus zeker helpen om de client secret te verbergen.

***Welke andere OAuth flow zou hier toepasselijk zijn? Is dit wel een correcte case om OAuth toe te passen?***  
 OAuth heeft meerdere flows, waarvan er 4 flows zijn om tokens te verkrijgen. Elke flow van   
 Oauth heeft dan ook zijn eigen sterktes en kenmerken. De Code Flos of Authorization Code   
 Grant lijkt mij in het geval van de opgave een goede degelijke keuze wegens het  
 aansprekingspunt van een database. Authenticatie om na te kijken of er autorisatie is voor de   
 gebruiker is volgens mijn in veel cases toe te passen en belangrijk om eventueel  
 gegevensverlies kunnen tegen te gaan.