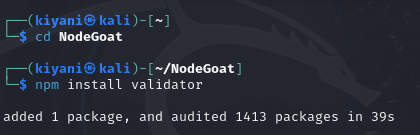
**DEVELOPERSHUB INTERNSHIP**

**WEEK 2: IMPLEMENTING SECURITY MEASURES**

**In this task, we are supposed to implement security measures in our web application, we have been provided with 4 security measures to implement.**

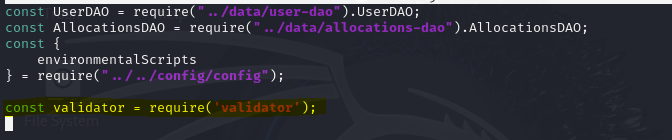
1. **Sanitize and Validate Inputs:**

**Installing the Validator library:**



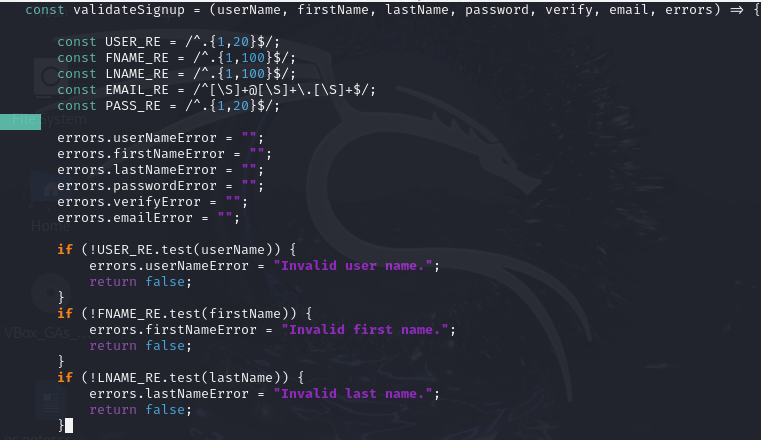
**Going to the session.js file in the app directory as it mainly contains all the input fields:**

**We start by entering the require statement at the top of session.js which loads the library into this file so that we can use the functions of it :**

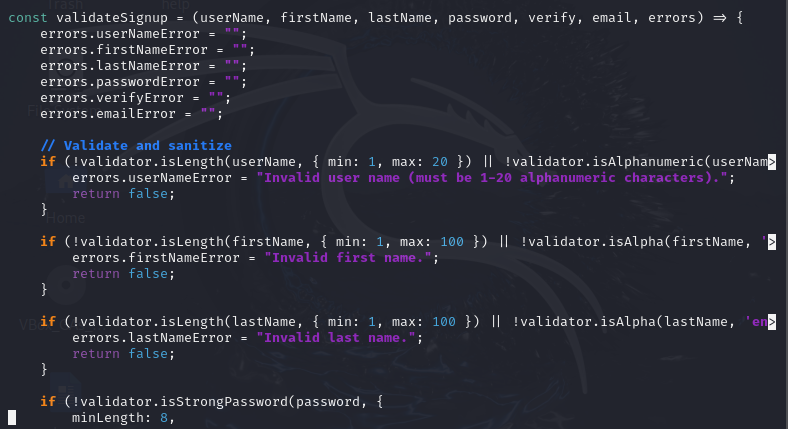


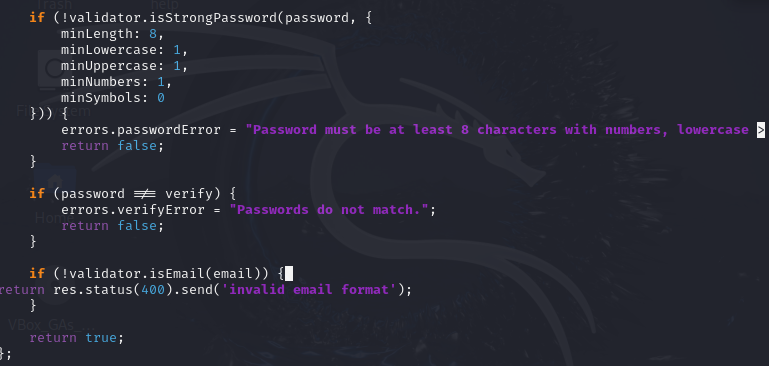
**To add the validation as we are asked to, we will change the validate signup block according to the requirement of our task:**

**Original:**

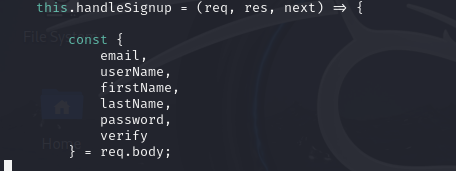


**After validation:**

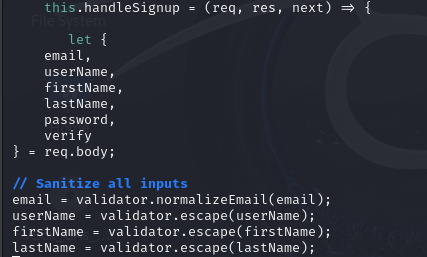




**Next we edit this block in handle sign up:**



**We validate and change to:**



**Explanation:**

**So as said in the task I first installed the validator and then I used the require line to add it in our session code. Now, in my session code I added the validation in my validate sign up block and I added sanitization in my handle sign up. I was given a prompt for email so I added that to validate email, and for others like username, passwords I added accordingly (as written in the code).**

1. **Password Hashing:**

**Installing bcrypt:**

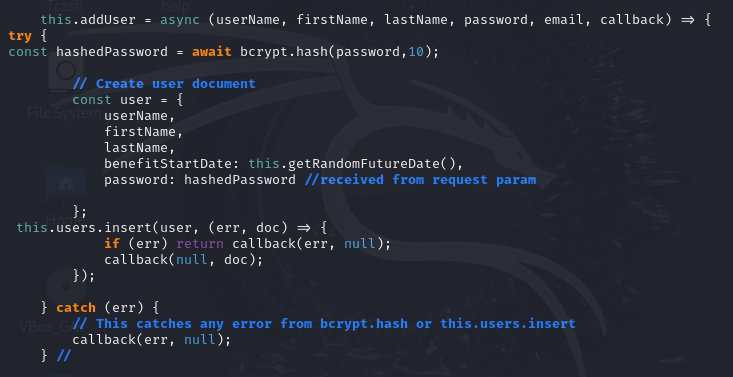


**Adding it in our code:**



**Now implementing the code according to the task:**

I added this in the user\_dao.js file where password management and storage is taking place.



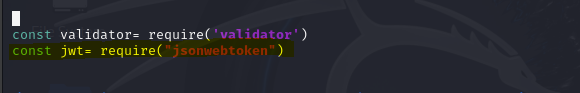
1. **Enhance Authentication:**

**The goal of this task was to enhance the security and flexibility of the application by adding JSON Web Token (JWT)-based authentication. My application was a session based application so I decided to add token based alongside the session based for the authentication.**

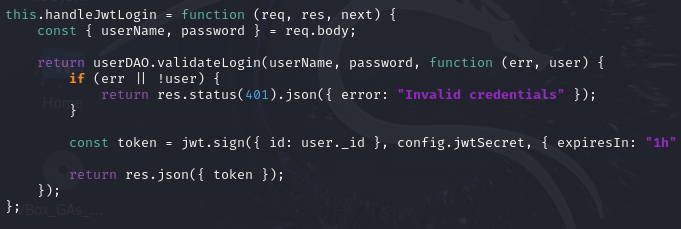
**Installing the required packages:**



**Requiring it in the session file of routes:**



**Added the login logic in my session.js file for JWT as instructed by the task:**



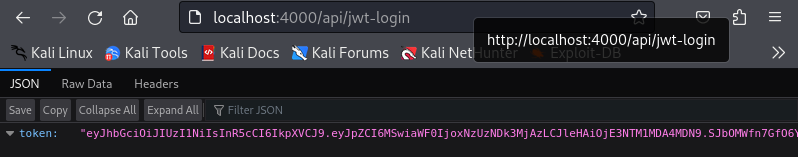
**I also updated the route in the index.js so we make a post request to this specific URL:**



* + You make a POST request to http://localhost:4000/api/jwt-login with the correct email and password.
  + If the credentials are correct, a JWT token is returned in the response.
  + If the credentials are wrong, the response is:

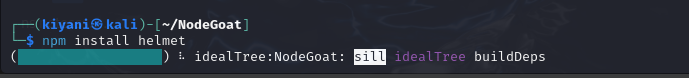
**{ "error": "Invalid credentials" }**

This JWT-based login works alongside the session-based login (/login route). The session authentication is still intact for browser-based flows, while the JWT version is useful for API or mobile clients.



1. **Secure Data Transmission:**

**Installing helmet in my system:**



**Requiring Helmet in the server.js :**



**Implement it in the code right at the start and under express app:**



**Headers shown in network tab:**