**Project Documentation for Access Control System**

This document will contain explanations and steps used in creating the NestJs based access control system, as well as resources about the frameworks and technologies used in this project.

***What is the stack?***

Stack in this case refers to the collection of tools and technologies we’ll be using for the different aspects of the software. These aspects include: database layer, frontend framework/language, frontend render engine, server framework(s) etc.

Now that that’s cleared up, what is our stack?

***Backend***

**Database**: *PostgreSQL*

**ORM**: TypeORM. What is an ORM? Object Relational Mapper. This is a tool used to map native code (in our case typescript) to SQL queries, this way we can communicate with PostgreSQL without having to write a single line of SQL.

**Render Engine**: Nunjucks (Templating engine for NestJs), We used this to render html pages and send information from the backend to the frontend. A good use case is sending a username from the backend to frontend, so a page can display an individual’s username which is obviously dynamic as the page has to show a different username for each user on the platform. It can also be used to add html to other html files e.g. Adding a navigation bar to every page as something like a function. It’s very similar to Jinja2 for python (remember Flask?).

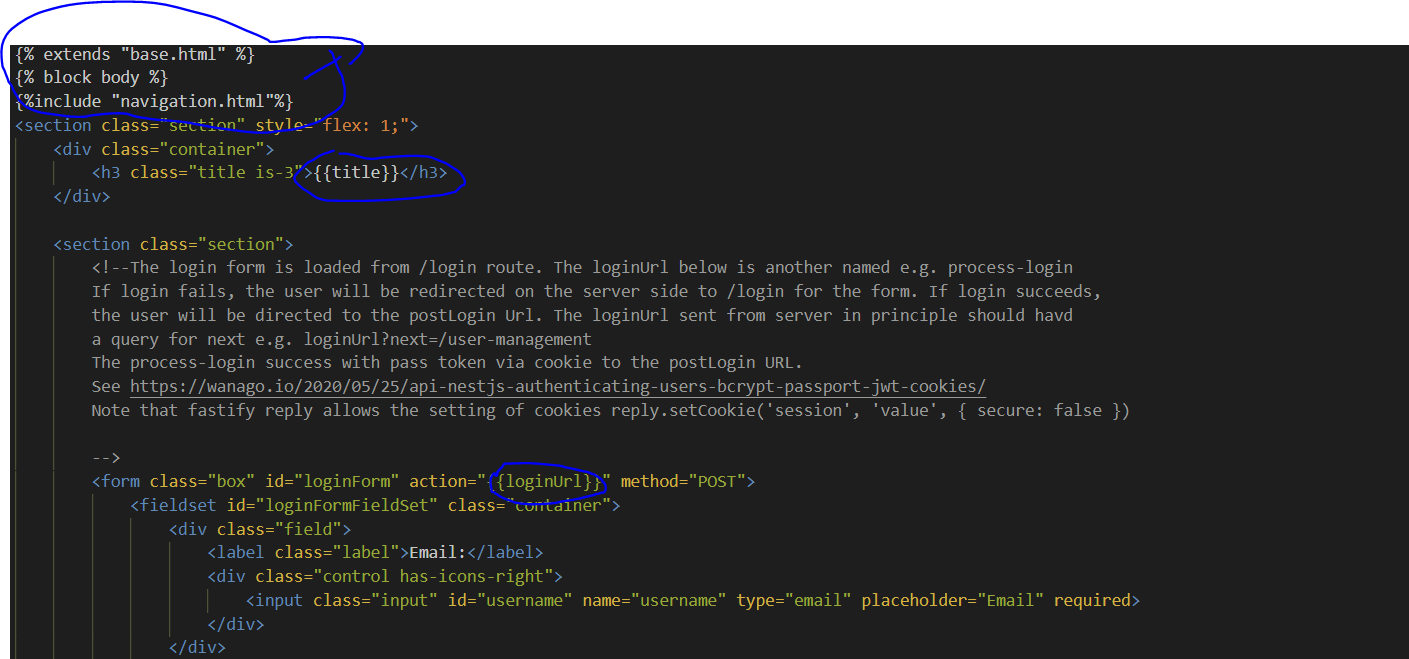


Figure : Example of nunjucks in action (Did not use this feature a lot in our project)

**Email Engine**: Nodemailer. This is used to send emails programmatically through a server. In our case our ‘SMTP: Simple Mail Transfer Protocol’ server was Gmail. Check *src/global/app.settings.ts* for how we configured it (which I’ll explain later in this document).

**Server Framework**: This is what will handle all our requests, responses and all the backend events that occur. Remember when I talked about Models, Controllers and Views? This is involved in the controller aspect (consisting of controller.ts and service.ts files) of the application. We went with **Fastify** (Nest itself is not a server framework, it is an application framework that makes use of tons of other frameworks to give you the full package of tools needed to build your application backend or API as well as connect to frontend).-