OWASP

**A1 Injection**

Is the Application Vulnerable?:

In my application the user-data is validated, so it makes it not resistant to attacks. Hibernate Query Language (HQL) is not used in the code and the data is retrieved by using mapping of a whole object or path variable. Hostile data is not directly used or concatenated by using SQL or command that contains both structure and hostile data in dynamic queries, commands, or stored procedures. CRUD repositories are the ORM in my application. In my configuration of CI/CD I haven’t added the static source (SAST) and dynamic application test (DAST) tools, which are used to identify newly introduced injection flaws prior to production deployment.

In conclusion, my app is not that vulnerable and cannot be broken into by attacking the queries or by modifying the ‘id’ in the browser url, for example, because it is not displayed there and is safe.

**A2 Broken Authentication**

Is the Application Vulnerable?:

My software application doesn’t implement the of the user’s identity, authentication because the authentication is done automatically after logging in. In my application there is no forget password function, so the app is still safe at some level because the attackers cannot change get into the account and change the password of the user. The password of the users in are hashed both when registering and when updating a user. The only vulnerability here is that the token of the authorization is saved into the local storage rather than in a cookie (which is stated to be the safer way).

**A3 Sensitive Data Exposure**

Is the Application Vulnerable?:

None of the data in the website is transmitted in clear text (by HTTP or FTP). The only for which they are used are sending a request (OK, CREATED, etc.) in the backend to outline the result. I still don not have the functionalities in this case that can be attacked. The only thing is that password of the user uses simple hashes to store everyone’s passwords.

**A4 XML External Entities**

Is the Application Vulnerable?:

**A5 Broken Access Control**

Is the Application Vulnerable?:

Access control enforces policy such that users cannot act outside of their intended permissions. In my application, when implementing the authorization, I made the user to not be able access the actions that require an ADMIN role. There is a “@PreAuthorized(“hasrole(‘ADMIN’)”)”, which states that a certain functionality is only permitted to the admin. Also, the information of the user can be changed either by the admin or the user, and the password can be changed only by the user themselves, and not in the url. In order the website to be safer, there is used “@CrossOrigin(origins = “localhost:3000")”, where I state that only this url can receive information or send requests to the backend. Accessing API with missing access controls for POST, PUT and DELETE is not applied, which means there is no vulnerability.

**A6 Security Misconfiguration**

Is the Application Vulnerable?:

**A7 Cross-Site Scripting**

Is the Application Vulnerable?:

**A8 Insecure Deserialization**

Is the Application Vulnerable?:

**A9 Using Components with Known Vulnerabilities**

Is the Application Vulnerable?:

**A10 Insufficient Logging abd Monitoring**

Is the Application Vulnerable?:

Logs are only stored locally may happen to attract more attackers and in my application, only the JWT token is being stored in the local storage.