**OWASP Report**

Table of Contents:

* Injection
* Broken Authentication
* Sensitive data
* XML External entities
* Broken Access Control
* Security Misconfiguration
* Cross-Site Scripting(XSS)
* Insecure Deserialization
* Components with known vulnerabilities
* Insufficient logging and monitoring

1. Injection
   1. Definition:

* SQL injection, also known as SQLI, is a common attack vector that uses malicious SQL code for backend database manipulation to access information that was not intended to be displayed
  1. Relevance and risks

- The application is generally safe against injection because the incoming data goes through JPArepository and not directly in the SQL query. Therefore there is not risk

1. Broken Authentication
   1. Definition:

* Several vulnerabilities exploited by attackers using information such as sessionId or user credentials
  1. Relevance and risks
* The application uses JWT authentication.
* Potential risk. Only password validation check is password length.
* How to fix: implement weak password checking according to the list of [top worst passwords](https://github.com/danielmiessler/SecLists/tree/master/Passwords).

1. Sensitive data
   1. Definition

* This security threat occurs when the web application doesn't adequately protect sensitive information like session tokens, passwords, banking information, location, health data etc…
  1. Relevance and risks
* User token is destroyed after a few hours in order protect the user account
* Passwords are hashed
* No other sensitive information is stored

1. XML External entities
   1. Definition

* XML external entity injection (also known as XXE) is a web security vulnerability that allows an attacker to interfere with an application's processing of XML data.
  1. Relevance and risks
* Application does not use XML processing and therefore no risks exit

1. Broken Access Control
   1. Definition

* Access control enforces policy such that users cannot act outside of their intended permissions. Failures typically lead to unauthorized information disclosure, modification or destruction of all data, or performing a business function outside of the limits of the use
  1. Relevance and risks
* The application has CORS configuration enabled.
* No POST, PUT or DELETE is allowed for non-authorized users.
* No data is modified by unauthenticated user

1. Security Misconfiguration
   1. Definition

* Security misconfigurations are security controls that are inaccurately configured or left insecure, putting your systems and data at risk.
* Attackers will often attempt to exploit unpatched flaws or access default accounts, unused pages, unprotected files and directories, etc to gain unauthorized access or knowledge of the system.
  1. Relevance and risks
* Security configuration is properly set which minimizes the risk of security breaches

1. Cross-Site Scripting(XSS)
   1. Definition

* Cross-site scripting is a type of security vulnerability typically found in web applications. XSS attacks enable attackers to inject client-side scripts into web pages viewed by other users. A cross-site scripting vulnerability may be used by attackers to bypass access controls such as the same-origin policy
  1. Relevance and risks
* Web application is written using ReactJs, which automatically escapes by design and XSS and eliminates the risk

1. Insecure Deserialization
   1. Definition

* Insecure Deserialization is a vulnerability which occurs when untrusted data is used to abuse the logic of an application, inflict a denial of service (DoS) attack, or even execute arbitrary code upon it being deserialized
  1. Relevance and risks
* No data is deserialized in the website which
* No risk

1. Components with known vulnerabilities
   1. Definition

* This kind of threat occurs when the components such as libraries and frameworks used within the app almost always execute with full privileges. If a vulnerable component is exploited, it makes the hacker's job easier to cause a serious data loss or server takeover.
  1. Relevance and risks
* The application avoids using open source libraries
* Components are up-to-date
* Little to no risk

1. Insufficient logging and monitoring
   1. Definition

* What is Insufficient Logging and Monitoring? According to OWASP, “Exploitation of insufficient logging and monitoring is the bedrock of nearly every major incident. Attackers rely on the lack of monitoring and timely response to achieve their goals without being detected.”
  1. Relevance and risks
* Unit test are implemented for controllers
* System test with Selenium are implemented for some of the functionality
* Not all components are tested and monitored on the frontend.