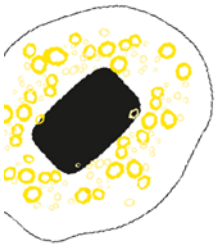


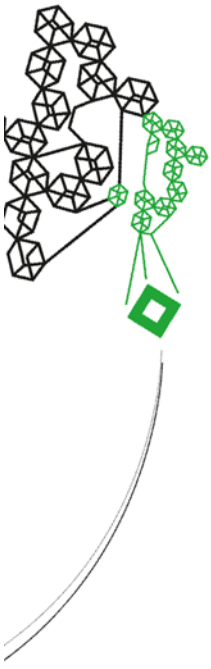
Data and Information 2023

‘Database report

Version 1.0
Date 26-05-2023
Status Final

Team EarnIT 4

Names (3077489) Gracjan Chmielnicki – BIT
(2993074) Tom Hansult - BIT
(2997894) Dirck Mulder - TCS
(2957566) Pepijn Meijer - TCS
(2920352) Thomas Brants - TCS
(2957868) Razvan Stefan - TCS



1 Security analysis

Ensuring the security of a website is of paramount importance to safeguard against malicious activities and unauthorized practices.

During user authentication, a JSON Web Token (JWT) is generated for each user upon sign-in. This JWT contains the user's unique identifier and additional information, such as the creation timestamp. To establish the authenticity of a JWT, the server signs it with a private key, enabling subsequent validation. Validating the JWT allows the server to confirm the authentication of the user. Subsequently, the token is returned to the user. If the user attempts to access any page or make requests without presenting this token, a 401 unauthorized error is returned.

All pages within the platform are categorized into three distinct groups: staff, student, and company. When a user requests access to a specific page, the server verifies the validity of the JWT and checks if the user is authorized to access the corresponding category. If authorization is denied, a 403 forbidden response is issued. This same authentication and authorization process applies to API requests as well.

In addition to authentication and authorization measures, the platform implements safeguards against potential malicious activities. Three identified problems have been addressed accordingly. Firstly, in the event of a database leak, the platform ensures the safety of user passwords by employing a hashing mechanism. The chosen hashing algorithm is bcrypt with 12 rounds, which is based on the blowfish cipher. The blowfish cipher is a block cipher that employs dynamic key changes, thereby mitigating the effectiveness of brute force attacks. Further information on this topic can be found at https://auth0.com/blog/hashing-in-action-understanding-bcrypt.

Secondly, to prevent SQL injection vulnerabilities, the platform utilizes prepared statements. Prepared statements serve to separate code and data, preventing unintended execution of SQL code by the server that may result from the mixing of code and data.

Lastly, the platform addresses cross-site scripting (XSS) risks by implementing server-side text escaping for user-provided inputs. Additionally, on the client side, the application employs the use of "innerText" instead of "innerHTML," thus mitigating the potential execution of HTML code even in the event of bypassing the server-side text escaping measures.