Security report, Module 4, Team 41, Topicus Bank Transactions

Security breach	Covered (Yes/No)
Cross-Site Scripting XSS	Yes
SQL Injection	Yes
Authentication	Yes
Protection against malicious file uploads	Yes

Cross-Site Scripting XSS

The user inputs are minimized as much as possible (only search bar and mt940 file can be uploaded) the search bar has a maximum text length and sanitized from any scripts and the file content is sanitized from some possible injections

SQL Injection

SQL Injection occurs when the user input is improperly sanitized and ends up interpreted as a code. Therefore, in order to prevent it our application makes use of the prepared statements. Thanks to it, we have the control over the input of the user and we can avoid any manipulation of the application logic by the attacker

Authentication

When it comes to authentication we make use of the factor “Something the user knows”. In our case it means that we expect from the user that he knows the credentials in order to log in to the website.
Moreover, we do not store passwords as the plain text, we use the “SHA-256” hash function, so in the worst case scenario that the data is leaked somewhere it will be very difficult for the attacker to derive the password since hash functions are One-Way. It means that there is no one f1() function corresponding to f() function.

Protection against malicious file uploads

In order to protect our application against the malicious file uploads our team has created the method isSafe() which returns the boolean in the parser class where we check whether in the file there are keywords which could indicate that there is a

potential threat from the attacker. If so, we return false from this method and the file is not going to be uploaded.
Moreover, we check the extension of the file, so if the file is not in the expected format, it is not going to be uploaded. Therefore, only files which are in MT940 format are going to be uploaded.