|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Risk | Risk Statement | Response strategy | Objectives | Likelihood | Impact | Risk Level |
| Hacking | During the early development of this project, my database was actually hacked and I was asked to send BTC to an address, so obviously hacking could be a big issue. Hackers could also use malicious scripts to mine for login credentials to edit the program and/or database. | Use a secure password instead of just using admin or root. | Ensure that the project stays secure for as long as feasibly possible. Avoid data leaks through brute force methods. | High | High | High |
| Repetitive Strain Injury or other computer related discomfort. | Spending long times at a keyboard can result in RSI or, in extreme cases, carpal tunnel injuries. This can be a major inconvenience and could cause delays in project delivery. Also having your monitor not at eye level and too bright will have a negative impact on your health. | Take regular breaks from not just coding but also computer usage to allow your hands to relax, as well as keeping your screen relatively dark and at eye level. | Lower the potential for computer related injuries or discomfort. This will aid in timely project delivery. | Low | Medium | Low-Medium |
| SQL injections | As myself and others attempted, SQL injections could cause major problems with the project, injecting misinformation and the potential to drop tables that really shouldn’t be dropped. | Ensure that all SQL statements executed by the program have a limit in terms of characters enabled to be entered in the program. Also prohibit the use of special characters such as the semi-colon. | Limit the probable attempted SQL injections by users with malicious intent or users trying to test the project. | High | High | High |
| Data Leaks | If you were to use incorrect Data Access Modifiers, this could lead to potentially personal data being leaked. This leaked information would not only be a GDPR violation, but could also be used for nefarious purposes by people with malicious intent. | Ensure that all access modifiers are set to the appropriate level of security, using private modifiers when appropriate. | Limit the amount of information available to users with malicious intent. Keep all details private where possible. | Medium | Low | Medium-Low |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Risk | Risk Statement | Response strategy | Objectives | Likelihood | Impact | Risk Level |
| Jenkins | Users with malicious intent could hack into the Jenkins server to potentially remove any vital files from the build. For example they could remove the pom file with all the dependencies and plugins in it which would stop the project from running properly. | Make sure Jenkins is updated to version 2.133, in order to implement SonicWall to provide protection against this exploit. | Lower the potential for malicious edits to the Maven build on Jenkins, preventing loss of vital project credentials. | Low | High | Med |
| GitHub | Any source code pushed to GitHub could potentially contain information that hackers would find useful when trying to a maliciously alter the project. The source files could potentially contain hard-coded login credentials which could allow for data leaks. | Use stronger passwords and usernames than just “admin” or “root”, then update those passwords every two weeks in accordance with the GDPR. | Reduce the likelihood of hacking and data leaks. | Medium | High | Medium-High |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |