|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Risk | Risk Statement | Response strategy | Objectives | Likelihood | Impact | Risk Level |
| Being hacked | To potential for either GitHub or MySQL to be hacked via malicious scrips created. The result could potentially cause data to be lost or stolen and in a professional environment this may breach GDPR. | Make sure the username and password is secure and not a standard obvious password. Also do not hard code the username or password into the java script you are running. | In order to protect the data and personal information that may be contained within the database. | low | High | Medium/high |
| Data loss through SQL injections | The use or certain SQL statements that could effect the database. Such as name being entered as Drop Database; | Having exceptions on certain statement being entered into the system. In addition, can limit the amount of characters that can be entered by the user. | Making sure the fundamental design structure of the database cannot be changed and is secure. | Medium | High | High |
| Cloud platform vulnerability | As the database is connected through a virtual instance there is potential that connection can be broken. | Have a backup of the information either stored locally onto a machine or memory stick. | To make sure that you could still run the program using a locally hosted service instead of the cloud platform. | low | medium | low |
| Data Leaks | Using the incorrect data access modifiers could cause private information to be made public, again this could cause GDPR breaches if personal information is made publicly available. | Making sure the private modifier is used | To establish an extra band of security for private information. | Medium | Medium | Medium |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Risk | Risk Statement | Response strategy | Objectives | Likelihood | Impact | Risk Level |
| GitHub merge conflicts | Potential merge conflict, push/pulling problems. Without rebasing information could become scrambled when uploading code to GitHub. | Make sure you use different branches such as developer and try to adopt the Feature branch model. | To make sure uploading of code is safe and regular. Keeping code organised and reduce the risk of lost work. | low | medium | low |
| Bad code structure | Potential for code to be unreadable for other developers | Try to minimise code length and delete unnecessary methods and statements. This can be achieved by planning beforehand. | To provide readable code that can be used and developed further by others. | low | low | low |
| Poor test coverage | Poor code coverage may result in the program not working or not working to the optimum level of efficiency. | Test code coverage as you are developing the code. Don’t just test at the end. | To create agile code that can be tested throughout the process of developing. | medium | medium | medium |