Risk Assessment Key

The Matrix interprets values for the severity of the risk (a) and likelihood of occurrence (b)  
(taking into account the frequency and duration of exposure) on a scale of 1 to 5,   
then multiply them together to give the rating band:

|  |  |
| --- | --- |
| Hazard Severity (a) | Likelihood of Occurrence (b) |
|
| 1 – Trivial (eg formatting issue, no impact on functionality or users) **2 – Minor** (eg may impede functionality)  **3 – Moderate** (eg minor security hazard or may cause build to break)  **4 – Serious** (eg project security vulnerability or build will break)  **5 – Fatal** (eg important sensitive information leaks, or multiple above) | **1 – Remote** (almost never)  **2 – Unlikely** (occurs rarely)  **3 – Possible** (could occur, but uncommon)  **4 – Likely** (will likely occur)  **5 – Very likely** (will occur) |

The risk rating (high, medium or low) indicates the level of   
response required to be taken when designing the action plan.

**Trivial**

**Minor**

**Moderate**

**Serious**

**Fatal**

|  |  |  |
| --- | --- | --- |
| Rating Bands (a x b) | | |
| **LOW RISK**  **(1 – 8)** | **MEDIUM RISK**  **(9 - 12)** | **HIGH RISK**  **(15 - 25)** |
|  |  |  |
| Continue, but review periodically to ensure controls remain effective | Continue, but, if possible, implement additional reasonably practicable controls where possible and monitor regularly | Implement additional controls wherever possible – handling this risk determines the quality of product. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** |
| **2** | **4** | **6** | **8** | **10** |
| **3** | **6** | **9** | **12** | **15** |
| **4** | **8** | **12** | **16** | **20** |
| **5** | **10** | **15** | **20** | **25** |

**Remote**

**Unlikely**

**Possible**

**Likely**

**Very likely**

| # | Risk Statement | Risk Description | Severity  A | Likelihood  B | A x B | Response Strategy |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | **GitHub** | Any source code might not get pushed regularly to ensure changes made to the code is saved | 2 | 5 | 10 | Set reminders to push code to GitHub |
| 2 | **SQL** | A user could type an incorrect command and delete the tables | 3 | 3 | 9 | Use user access to control who can do certain things |
| 3 | Not understanding the material | As there are many different technologies we are implementing and with the little time to master them, time pressure will be challenging. | 4 | 4 | 16 | Consult the trainers, recordings and external material to help |
| 4 | 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. | 5 | 2 | 10 | Use stronger passwords and usernames than just “admin” or “root”, and keep them regularly updated. |
| 5 | Covid-19 | During the current situation it may possible to get ill. | 3 | 1 | 3 | Stay indoors and go out as little as possible. |
| 6 | SpringBoot | Understanding something this new and a bit more complex is a little difficult. | 3 | 4 | 12 | Trying to use the time in between to learn and do as much as I can. |
| 7 | Computer Use | Spending long times working on a computer may cause RSI (repetitive Strain Injury), eye strain and back problems. | 3 | 3 | 9 | Reduce the amount of time spent in front of the computer to prevent eye straining and back pain. |
| 8 | Time | With the deadline creeping up and little time in between the learning time management will be the biggest key factor for this project. Not time managing properly result in lack of functionality or features, whether it is in the product or documentation. | 5 | 5 | 25 | Use the KANBAN board in Jira to help plan out a structure on running the project. It gives a good aid in how long you might need and how to get things going. |