Risk Assessment Document

**Risk Identification**

|  |  |
| --- | --- |
| **Risk Description** | **Explanation** |
| Late Delivery | The project may not be completed by the due date if unforeseen delays occur in development or team members fall behind. |
| Software or Equipment Failure | Issues with tools like GitHub or Trello could lead to corrupted files or the inability to access essential project data. |
| Team Member Unavailability | A team member may be unable to contribute to the project or attend meetings due to unexpected commitments or emergencies. |
| Project Scope Changes | Midway through, the team may decide to alter the project scope, which could lead to setbacks or require redoing completed work. |
| Hardware Failure | A team member’s device could fail, making it difficult for them to participate fully in development tasks. |
| Presentation Completion Issues | If focus is primarily on coding, there may not be enough time to complete the final project presentation effectively. |
| Local Software Failure | The software used to build the project (e.g., IDE or libraries) might fail, causing potential file corruption or lost progress. |
| Limited Meeting Availability | The team may struggle to find times to meet outside of class due to scheduling conflicts. |
| Quality of Visuals | The final project might lack high-quality visuals, which could affect the presentation and overall appeal. |
| Excessive Coding Requirements | The coding workload could exceed the available time, leading to incomplete features by the project’s deadline. |
| Debugging Limitations | Some errors might remain unresolved due to time constraints, potentially impacting project functionality or presentation. |

**Risk Analysis**

|  |  |  |  |
| --- | --- | --- | --- |
| **Risk** | **Probability** | **Effect** | **Explanation** |
| Late Delivery | Low | Catastrophic: Missed deadline affects grades. | The due date has been clear from the start, making it unlikely we’ll miss it. |
| Software or Equipment Failure | Low | Catastrophic: File corruption or data loss could impact progress. | Reliable platforms like GitHub minimize the risk of losing project files. |
| Team Member Unavailability | Low | Serious: Limits productivity and delays milestones. | It’s unlikely a team member will be unavailable for an extended period. |
| Project Scope Changes | Low | Serious: Resetting progress can delay the project. | The theme has been firmly set, reducing the chance of major changes. |
| Hardware Failure | Moderate | Serious: Reduces a team member’s ability to contribute. | Personal devices could potentially malfunction during the project. |
| Presentation Completion Issues | Moderate | Serious: Incomplete presentation could affect grading. | Focus on coding might lead to rushing the presentation preparation. |
| Local Software Failure | Moderate | Tolerable: Files can be recovered, though minor delays occur. | Development tools may encounter issues, but backups can prevent major setbacks. |
| Limited Meeting Availability | Moderate | Tolerable: Rescheduling can catch up on missed discussions. | Scheduling conflicts could affect availability for some group meetings. |
| Quality of Visuals | Moderate | Tolerable: Lower visual quality is acceptable, but it affects appeal. | Time constraints may prevent creating high-quality visuals for the project. |
| Excessive Coding Requirements | High | Serious: Incomplete features could impact the final grade. | The functionality required is extensive, and time to complete it is limited. |
| Debugging Limitations | High | Tolerable: Minor bugs are acceptable if they don’t break the project. | Testing everything will be time-consuming, possibly leaving minor errors. |

**Risk Planning**

**Schedule Risk:**

To minimize scheduling issues, the team will develop a clear timeline with individual responsibilities and milestones. Weekly check-ins will help keep everyone aligned with the schedule. If delays occur, adjustments will be made to prioritize essential features.

**Technical Risk:**

For technical risks, the team will ensure backups are regularly saved to the cloud (GitHub) to prevent data loss. Each member will keep local copies of the project files to mitigate risks related to cloud or local software failures.

**Resource Availability:**

All team members’ roles and responsibilities are established early on. In the event that a team member cannot contribute, tasks will be reallocated temporarily to ensure the project remains on schedule. Tasks are tracked in Trello among the team.

**Scope Risk:**

The project scope will be clearly defined at the start, and any proposed changes will be discussed and approved by the team to avoid unnecessary deviations from the planned scope.

**Presentation Quality:**

While coding is a priority, the team will allocate time towards the end of the project to focus on presentation elements, including visuals and overall polish, ensuring the final product meets quality standards without rushing.