**Contribution Guide**

Please follow the following code of conduct when contributing to the development of this project’s source code:

**Code of Conduct:**

**1a. Push Process:**

When pushing file updates to the repository, you should ensure that:

* Every push contains a commit message, describing the nature of the commit and push, in less than 50 characters.
* Any dependencies which are needed have been included in the file push or include instructions for dependency installation.
* The commit message does not contain any language which could be interpreted as:
  + Offensive
  + Misleading
  + Lacking in description

All pushes should be made to the “master branch”, however, if you decide that files will be pushed to a different branch than the “master branch”, then inform the Project Manager or the Project Maintainer so that they can plan when to merge the branches together.  
When you push files to a different branch, a pull request should be generated and the Project Manager should be informed so they can review the changes and decide whether to pull the contributions into the “master branch”, merging the branches together, or not. If the Project Manager is not available, then contact the Project Maintainer.

**1b. Pull Process:**

When pulling files from the repository, you should ensure that:

* A message is provided to inform the Project Manager and other contributors on the changes that have been made to the files by the contribution.
* To inform the Project Manager of the changes made, either contact them on one of the communication channels, or use GitHub’s “@ mention system” in the pull request.

Project Contributors can create Pull Requests on the GitHub repository, which gives the other contributors more information about the changes you are pushing to a branch. The instructions of creating Pull Requests are:

1. Click the green “New Pull Request” button in the Pull Request tab.
2. Compare the branch that you created with the branch you wish to merge your changes with.
3. Look over your changes and ensure that they are accurate.
4. Click the green “Create Pull Request” when you are satisfied with the changes.
5. Describe your Pull Request with a title and description.

After a Pull Request has been made, you should notify the Project Manager of the new request and they can read the proposed changes to the repository. The instructions to merge Pull Requests are:

1. Click the green “Merge Pull Request” button.
2. Click the “Confirm Merge” button, which will merge the branches together.
3. Delete the branch which the changes were made to before the merge, as its changes have been merged with the target branch, by clicking the “Delete Branch” button.

**1c. Scope of Contributions:**

These rules are valid for both contributing to and retrieving the files which will be used to create the final project. The circumstance in which a file will not be relevant to the project will be subject to a discussion during a meeting in which the entire group is present. If a stalemate is reached in relation to whether a file will no longer be relevant to the project, then the Project Manager will make the final decision.

**1d. Rights and Responsibilities:**

The Project Manager is responsible for ensuring that every team member is feeling confident with continuing with their respective contribution areas as well as upholding the project milestones.

The Project Maintainer is responsible for maintaining the structure and directory of the GitHub repository and that any files which have been pushed to the repository are in the correct folders.

The Project Contributors are responsible for developing the code until the final project has been completed and submitted. Their responsibilities also extend to both upholding good practises (*1ei*) and avoiding bad practises (*1eii*).

**1e. Examples of Practises:**

**1ei. Examples of Good Practises:**

Examples of behaviour that is considered good practise:

* + Following the Push (*1a*) and Pull (*1b*) processes when contributing to the project.
  + Taking constructive criticism into account when contributing to the project in the future.
  + Contributing to the project before the stated deadline.

**1eii.Examples of Bad Practises:**

Examples of behaviour that is considered good practise:

* + Writing unhelpful or vague commit messages or pull request messages.
  + Writing offensive messages in either the commit or pull request messages.
  + Arguing with other team members on any communication channel.
  + Not including any message while contributing to the project.