1. **Project Title**: The “EriTufJohMik’ Project re Mental Health Trends in Technology
2. **Team Members**: Erica Hoshino

Tuflika Putri

John Nasiakos

Mike Murphy

1. **Directory / File Structure for the Project:**

**Introduction**

Mike has just had a tutorial on best practice for managing Github file and directory structures to ensure smooth collaboration between team members and minimise the risk of conflicts with a demonstration of what best practice looks like.

Based on this he would recommend the following approach for the team.

1. **Recommended Approach**
   1. **The main branch on the Github account used by collaborators should be reserved exclusively for clean, working versions of code and supporting datasets / files that have been agreed by the team. This is currently “07-project-1-challenge”.**
   2. **Subsidiary branches should be established for each collaborator in the team ( Main/Erika, Main/John, Main/Tuflika & Main/Mike).**
   3. **Collaborators should do all their work in their own branch or any other sub branches they choose to set up.**
   4. **Collaborators should copy and backup any work in their local repository that they may want to keep then delete it from your local repository before you clone the main branch.**
   5. **Collaborators should then clone the main branch to their local repository on their desktop.**
   6. **Collaborators should then pull the main branch from github to their local desktop repository.**
   7. **At this point all collaborators should have exactly the same version of the main branch on their local desktop repository.**
   8. **Collaborators can then do all their required work in their local desktop repository.**
   9. **Once collaborators are happy with their work in their local desktop repository, they should push their work to their github branch ( Main/Erika, Main/John, Main/Tuflika & Main/Mike) - not to the main branch. Note: the sequence for pushing to github is as follows: -**
      1. **Git pull – to pull down any changes that may have been made to the main branch since the collaborator last did a git pull**
      2. **Once git bash says that your local repository is up to date with the github repository then you: -**
      3. **“git add . “**
      4. **“git commit -m ‘message re the purpose of the push’**
      5. **“git push”**
   10. **This will ensure that the collaborator’s local repository and their github repository are in sync. ( Main/Erika, Main/John, Main/Tuflika & Main/Mike).**
   11. **Once collaborators have each updated their github repositories the team should review the latest updates to each collaborator’s repository (local and github) and agree what needs to be consolidated into the team solution in the main branch.**
   12. **Any changes necessitated by the team review should be implemented in the github versions of the collaborators’ branches or in the main branch depending on what is required to achieve the best outcome.**
   13. **Once this has been done the github main branch and the collaborator’s branches should be merged in github.**
   14. **At this stage the github main branch contains the most up to date agreed version of the code, files and datasets necessary for the solution for the project.**
   15. **All collaborators can then resume the cycle of pulling the latest version of github main to their local repositories and continue their agreed individual tasks.**

**Note to all**

***I think that I have captured what my tutor and I went through. Please read, critique and provide feedback so that we can move forward in a best practice way to smooth the development of out project and avoid unnecessary pitfalls. We can then get Uday to cast his eye over it tomorrow***.