GITHUB WORKFLOW

Revision 1.0

# Adding Features

1. In the Agile Process a project is divided into features that are rolled out iteratively.
2. GitHub Issues are used to highlight features.
3. To create an issue visit Project tab and select "Agile Project Management" Kanban board.

A screenshot of a computer

Description automatically generated

1. Create an item and give it a title (with an id AXXX, X is an integer) which would reflect the feature to be added.
2. Provide a short description of the item created, this could include describing the feature to be implemented.
3. Add an assignee to the item.
4. Lastly select "convert to issue" on the left nav bar, this will show the task in the Issues tab of the repository.

A screenshot of a computer

Description automatically generated

# Working on Features

1. In the Issues tab select the feature to be worked on.
2. Create a branch and give it the name of the feature to be added.
3. GitHub will provide you with a fetch and checkout commands to clone the branch that you would be working on.

A screenshot of a computer

Description automatically generated

1. Open vscode and in terminal paste the given commands.
2. Change the status of the project from ‘Todo’ to ‘In Progress’.
3. As you work on the feature add comments for easier documentation.
4. Comments section also supports Markdown allowing to write documentation which GitHub will display similarly to readme files.

# Committing

1. During the process of developing the Feature, it is best practise to commit each change done to a file for easier documentation.
2. The commit should answer the following questions: What was the change? Why?

# Pull Request & Merging

1. Once feature has been implemented in it’s own branch and passed tests, it is ready to be merged with the main branch.
2. The collaborator will perform a pull request, where other collaborators will review the code and provide feedback/comments. Before placing another pull request ensure that you have answered all comments provided from previous rejected pull request.
3. If everything is good then the feature is merged with the main branch and given a tag id named after the feature implemented.
4. Tag ID’s are human readable id’s that make finding versions easier than using the commit id’s (which are in hash).

# Merge Conflicts

To be added soon

# Bug Report or Questions

1. You can also provide questions and bug report as an Item in "Agile Project Management" Kanban board.
2. Adding labels can help classify the Item by labelling them as reports, questions, and other categories.
3. If the Item becomes part of implementing a feature, it could be “converted into issue” through the left nav bar (similar process as with adding features normally).

* E.g.: Question is how to sort comments left by users on other user’s Posts? This could turn into a feature where comments are sorted by (a) whether the owner of the Post finds the comment relevant to their post (approval from post owner) and (b) the upvote the comment has received from other user’s (like Reddit/Instagram comment likes).

1. To give the Item an ID, take the superset Feature’s ID that caused the bug or raised the question and place the Item after the major feature but before another major feature.
   * E.g.: To implement Issue A100 – Sign Up/Login Features, we have a question about how to connect login info to database, however the next feature is A110 - Add Post Feature, therefore we can give the Item the ID A105 – LoginDB as a good ID and title.