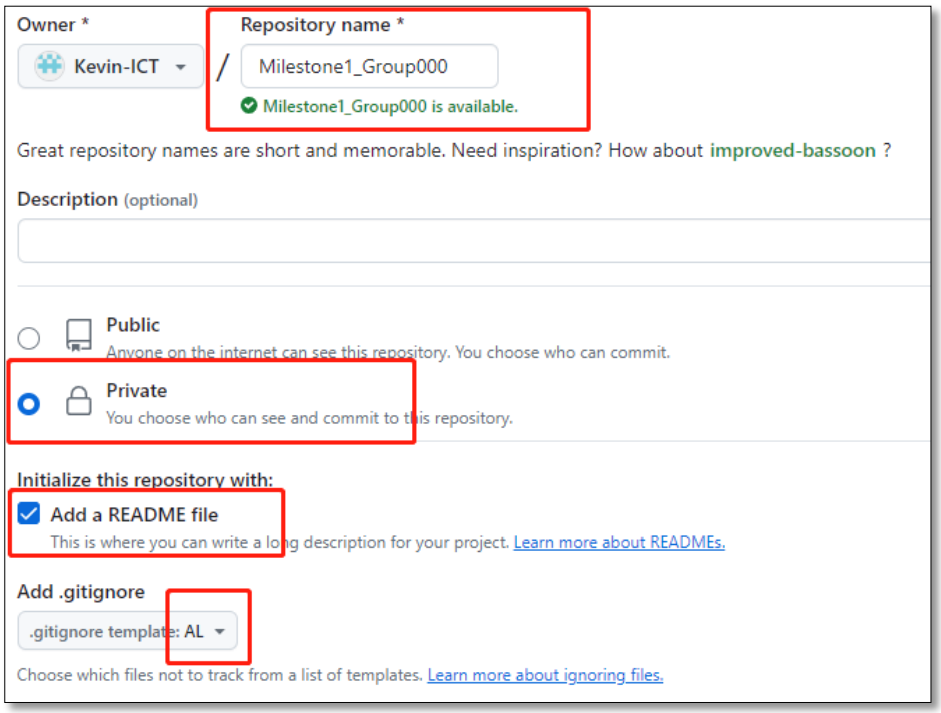
Workflow for Collaborating via Git & GitHub on Group Project

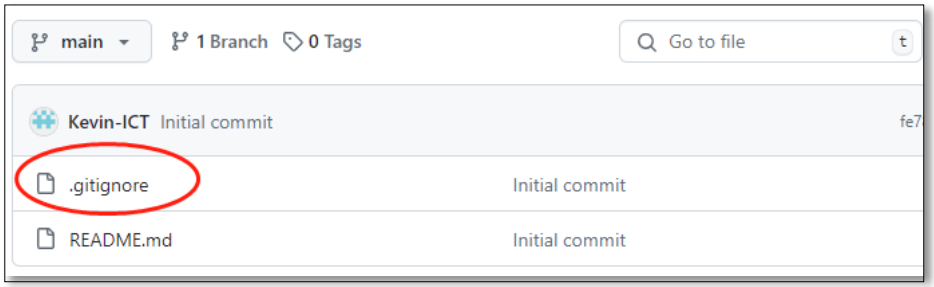
- Prerequisite: Git and Github have been successfully set via week2.
- **A0: Start your group project**

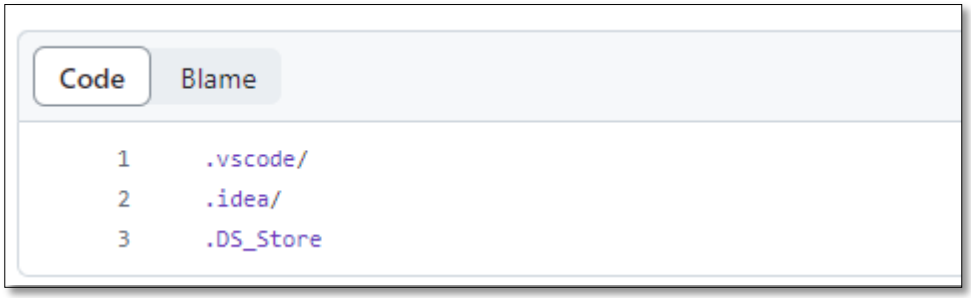
Step 1. **One of your teammates** (The others do nothing is this step. Wait and Go to **Step 2**)

1) creates a new private GitHub repository with a name **Milestion1_GroupXXX**, (XXX is replaced with your group number)

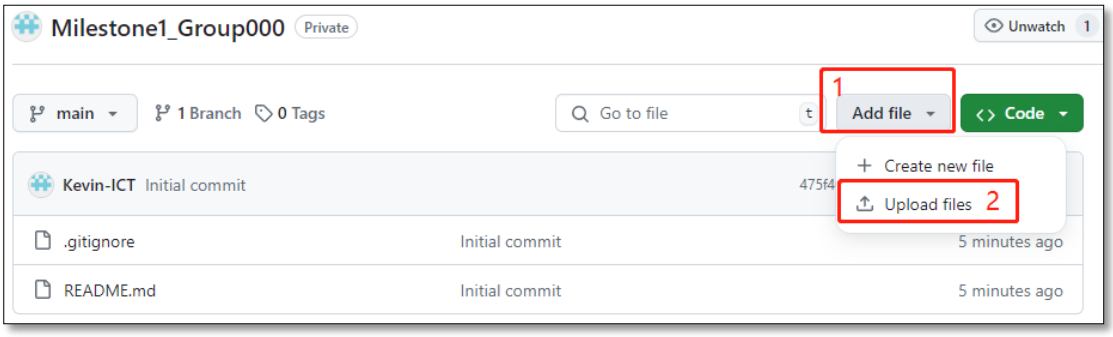


2) Click the .gitignore and edit like below

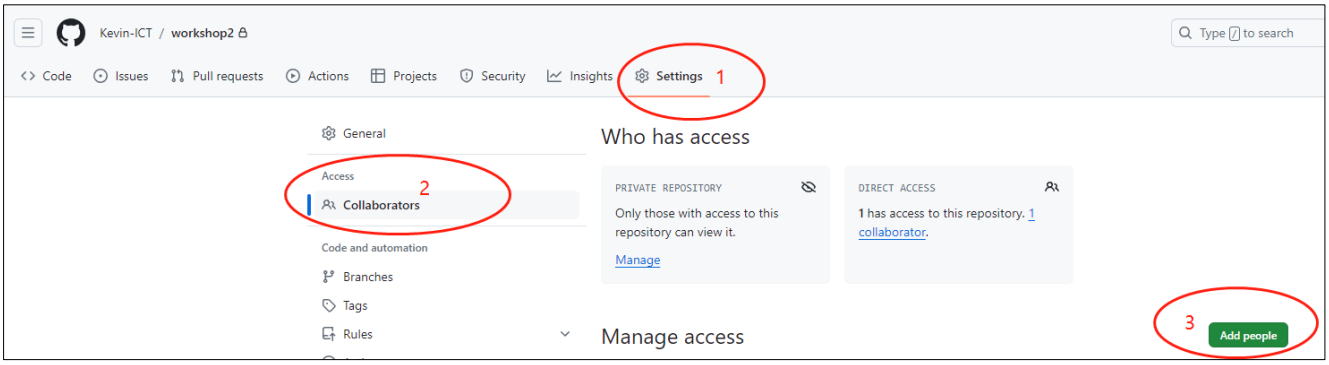




3) Download & unzip **Milestone1.zip**, and add these files to the GitHub repository.



4) Add other teammates to the repository



Step 2. **[The others]** will receive an invite email in your registered email box and need to accept this invite.

Step 3. [**Everyone**]

- **Windows User**: Open your git bash
- **MacOS User**: Open the Terminal
- [**Windows or MacOS**] If you installed git using conda, open a Terminal and activate your conda env.
  conda activate YOUR_CONDA_ENV
  (you can use conda env --list if you forget your conda env name.)

Important Note: All the commands below are executed in this Terminal window.

Step 4. **[Everyone]** Use **cd** to navigate to where you saved your workshop1, so that Milestone1_GroupXXX and workshop1 are in the same directory.

Step 5. **[Everyone]** Clone the repo using git clone https://xxx//zzzz.git
Then, **cd zzzz** (replace zzzz with your repository name)

Step 6. **[Everyone]** Create a local branch: git branch branch-name (don't contain space in the name)

• A1: Collaborate on your group project

Open the Terminal (Refer to A0.Step3) and Navigate to your local git directory in the Terminal, please follow the steps below to collaborate with your teammates and contribute to the project:

Q1. Switch to the local branch: git switch branch-name (use git branch to double check)

Q2. update your local repo with changes from the GitHub main branch: git pull origin main

1) If conflicts happen, open the conflicted file, fix conflicts manually. (Please refer to Step 5 and Step 6 in this Tutorial link)

Q3. Modify codes or documents in your local working directory
Q4. Add changes to staging: git add ./file_name
Q5. Commit changes: git commit -m "Write changes here."
Q6. Switch to main: git switch main (use git branch to double check)
Q7. Merge the previous working branch into the main: git merge branch-name
Q8. Push changes to GitHub repo: git push origin main
Q9. Repeat Q1 to Q8

1. Set soft-wrap for .md in PyCharm:
File -> Settings -> Editor -> General -> "Soft-wrap these files"

2. PyCharm converts markdown to pdf:
Tools > Markdown Converter > Export Markdown File To...

3. You have to use the cmd to produce the git log:
git log --oneline --graph --decorate --pretty=format:"%h %ad by [%an] | %s%d" --date=short > git_log.txt

Commonly used git cmd:

git help git status git clone <xxxx.git> git config --list --show-origin git add <file_name> git commit -m “xxxx” git remote -v	git branch git branch <branch_name> git switch <branch_name> git merge <branch_name> git pull origin main git push origin main