Git Branching Strategy for Capstone

Working as a Developer on the Repository

Below are the steps you should follow to make sure you use branching effectively when working on a task.

- Create a branch to do your work in for the task you have selected. To keep things simple, use a descriptive name of the task as the branch name

  ```
  git checkout -b add_carousel
  ```

- Make sure your branch is tracked by the remote repository as well

  ```
  git push -u origin add_carousel
  ```

- Open the project and work on your task. Add and commit your changes as often as possible. The holy trinity of git commands below are what you need when you are ready to commit any changes

  ```
  git add .
  git commit -m "description of my change"
  git push
  ```

- **(OPTIONAL but good practice)** When you are ready to merge your changes into the main branch, make sure your main branch is up to date with the `remote main` and deal with any merge conflicts that come up. Since you are working alone in this capstone, this should never be an issue. But it would be an issue if you had colleagues working on the same project.

  ```
  git checkout main
  git pull
  git checkout add_carousel
  git merge main
  ```

- Now, merge our changes into the main branch and push to the remote repo (origin)

  ```
  git checkout main
  git merge add_carousel
  git push
  ```

- Now we create a new branch for our next task and repeat the process

  ```
  git checkout -b NAME_OF_NEXT_BRANCH
  ```