GIT WORKFLOW AND COMMAND USAGE

    SCREENSHOT 1:
1.) git remote add <remote repo url> : To access our remote repository
2.) git pull-: To pull data onto our local repo
3.) git init- to ensure git is initialised in the appropriate local folder
4.) git status: to check tracked and untracked files

    SCREENSHOT 2
5.) git checkout creating_api_endpoiints: to switch from main branch to api branch
6.) git add : To add the files to the staging area

    SCREENSHOT 3
7.) git commit -m "created third api endpoints": To commit the staged file.
8.) git push origin creating_api_endpoiints: to push to the appropriate branch

The commit history also shows individual branches that were created and has since merged into main - these branches were deleted to keep our repository tidy.


```
Last login: Mon Oct 23 16:13:58 on ttys000
[kemioluwasanmi@kemis-Air ~ % cd /Users/kemioluwasanmi/Desktop/4CodeGirls
[kemioluwasanmi@kemis-Air 4CodeGirls % git init
Reinitialized existing Git repository in /Users/kemioluwasanmi/Desktop/4CodeGirls/.git/
kemioluwasanmi@kemis-Air 4CodeGirls % git remote add origin https://github.com/charlotteswebdev/4CodeGirls.git
[error: remote origin already exists.
kemioluwasanmi@kemis-Air 4CodeGirls % git pull
Already up to date.
[kemioluwasanmi@kemis-Air 4CodeGirls % git status
On branch main
[Your branch is up to date with 'origin/main'.

Untracked files:
```

```
[kemioluwasanmi@kemis-Air 4CodeGirls % git checkout creating_api_endpoints
M       PawsitiveAdoptions/app.py
M       PawsitiveAdoptions/config.py
M       PawsitiveAdoptions/db_utils.py
M       PawsitiveAdoptions/main.py
Already on 'creating_api_endpoints'
Your branch is up to date with 'origin/creating_api_endpoints'.
[kemioluwasanmi@kemis-Air 4CodeGirls % git add PawsitiveAdoptions/app.py PawsitiveAdoptions/db_utils.py PawsitiveAdoptions/main.py
```

```
kemioluwasanmi@kemis-Air 4CodeGirls %
kemioluwasanmi@kemis-Air 4CodeGirls % git commit -m "created third endpoints"
[creating_api_endpoints 6807b00] created third endpoints
kemioluwasanmi@kemis-Air 4CodeGirls % git push origin creating_api_endpoints
Enumerating objects: 19, done.
Counting objects: 100% (16/16), done.
Delta compression using up to 8 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (9/9), 2.43 KiB | 2.43 MiB/s, done.
Total 9 (delta 5), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (5/5), completed with 4 local objects.
To https://github.com/charlotteswebdev/4CodeGirls.git
   9627f15..6807b00  creating_api_endpoints -> creating_api_endpoints
kemioluwasanmi@kemis-Air 4CodeGirls %
```

TEAM WORK ON GITHUB:
Project Description:
Our project was a collaborative endeavour to develop an API endpoint and client API requests for a programme designed to facilitate dog adoption, retrieve shelter information, and manage a mailing list.

Repository Setup:
- Sophia initiated the project by creating the GitHub repository.

Branching:
- We began by familiarising ourselves with Git branching.
- We established the 'main' branch as our primary platform.
- Each of us enriched the README file with our profiles, committing and merging these updates into 'main.'
- In Section 1 of the assignment, we created separate branches to manage our contributions.
- In the subsequent phase, we created the 'creating API endpoints' branch to develop the essential API for our dog adoption programme.

Development and Collaboration:
- Sophia and Charlotte focused on refining 'main.py,' structuring the programme's logic, configuring inputs, and creating necessary functions.
- Lauren and Ana Catarina concentrated on creating SQL tables and 'db_utils.py' to streamline database interactions.
- Kemi and Jess took charge of developing API endpoints using Flask and ensuring a seamless interface between the API and the client.

Collaboration and Development:
- Collaborative tools such as Slack and Zoom calls facilitated communication and progress updates.
- This collaborative approach mirrored daily stand-up meetings, promoting clarity regarding individual responsibilities.

Resolving Conflicts:
- During branch merges, occasional conflicts arose but were promptly addressed through the pull request mechanism, ensuring a smooth collaboration.