[CLA Hands-On] Github and HTML

Objective
This document is designed for a hands-on session to practice Git, a version control system, and HTML together. The focus is on how developers contribute to a shared repository collaboratively in a web application.

Context
Imagine you're working at a company called Intro-pal, where every developer adds their introduction to the repository. You notice that your introduction isn't included yet, so your task is to add it by creating your own HTML file in the repository.

Task 1: Clone the Repository

- Open your terminal.
- Clone the repository using the command **git clone https://github.com/abdullah-siddiqui-cla/git-intro.git**
- Navigate to the folder (git-intro) that was created as a result of cloning the repository.

Task 2: Create an Issue

In GitHub, create an issue to add your introduction to the repository.

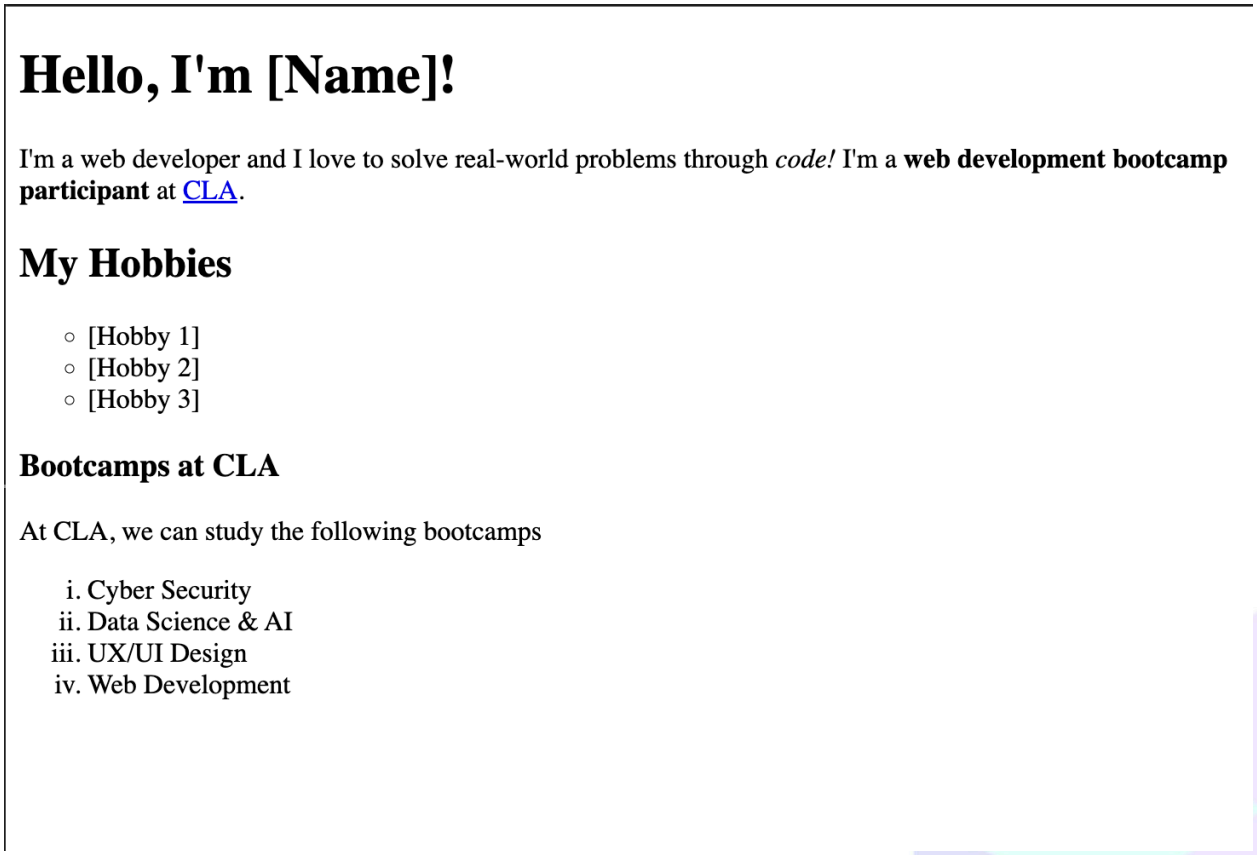
- Go to the “Issues tab” of the repository.
- Create a new issue with the title "[Your Name] - Add Introduction".
- Add a brief description: "This issue is to add my introduction to the Intro-Pal"

Task 3: Work on the Issue

Start working on the task of adding your introduction to the repository.

- Fetch the **dev** branch from the origin (Hint: Use **git fetch** command)
- Checkout to **dev** branch (Hint: Use **git checkout** command)
- Create a new branch with name like “[your name]-add-introduction” (Hint: Use **git checkout -b [branch name]** command)

- Create a new HTML file with your first name like "**abdullah.html**". Write HTML into the file to achieve an output similar to the following image. You can fill the content with your own name and hobbies.



Hello, I'm [Name]!

I'm a web developer and I love to solve real-world problems through *code!* I'm a **web development bootcamp participant** at CLA.

My Hobbies

- [Hobby 1]
- [Hobby 2]
- [Hobby 3]

Bootcamps at CLA

At CLA, we can study the following bootcamps

i. Cyber Security
ii. Data Science & AI
iii. UX/UI Design
iv. Web Development

- Stage, commit and push the changes to the new branch. (Hint: Use **git add**, **git commit**, and **git push** command).

Task 4: Create a Pull Request (PR)

Now that your changes are in GitHub, create a pull request (PR) to merge your branch into **dev**.

- Go to the "Pull Requests" tab in the GitHub repository.
- Click on "New Pull Request."
- Set the base branch to **dev** and compare it with your branch (<[your name]-add-introduction>).
- Add a descriptive title and summary, then submit the PR.

Task 5: Review a Teammate's Pull Request

You'll now review a teammate's PR to simulate peer review:

- Go to the "Pull Requests" tab and pick any of your team member's PRs.
- Check the file changes.
- If you have any feedback, leave a comment. Otherwise, approve the PR by going to "Review Changes > Approve".

Task 7: Merge dev into main

Once all PRs are merged into **dev**, it's time to update the main branch:

- Checkout to the main branch (Hint: Use **git checkout** command)
- Merge **dev** into main (Hint: Use **git merge** command)

The main branch is now updated with the latest changes from dev and ready for deployment to the production environment.