

Preparation for Project

CLASS ACTIVITY

✦ Generate a quiz about this unit

✦ Summarize this unit

✦ Have any doubt about this content? Ask!

Introduction

In this activity, you will kick-off the Module 1 project in your teams. The activity will serve as preparation for the project, that involves collaborative planning, creative ideation, and efficient project management - all essential skills that will serve you well in your professional career.

This activity is divided into three key parts:

1. Setting up your GitHub repository and practicing Git collaboration with your teammates.
2. Ideating your project's theme and topic.
3. Planning your project through the creation of a Kanban board.

Each part of this activity is designed to help you and your team organize and strategize your approach to the project. You'll start by setting up your GitHub repository and practicing Git collaboration. This will help you establish a solid foundation for efficient teamwork and version control as you progress through the project.

Next, you'll move on to project ideation, where you and your teammates will brainstorm and discuss potential themes and topics for your project. This is your chance to get creative and think about current events, social issues, or any other topics that you are passionate about.

Lastly, you'll dive into project management by creating a Kanban board. This visual tool will help you organize your tasks, assign responsibilities, and keep track of your progress throughout the project.

Let's get started!

Part 1: Git & GitHub (1 hour)

Git and GitHub are standard tools in the industry that are crucial for efficient collaboration and version control in software development and data analytics projects. Git allows you to track changes, revert to previous stages, and work on different branches simultaneously, while GitHub provides a platform for sharing code and collaborating with others. In this activity, and the project as a whole, you will have the opportunity to practice using these tools in a team context.

You will work together with your teammates to divide tasks, write code, and merge it into a shared repository. You will apply key Git and GitHub concepts such as branching, committing, creating pull requests, and resolving merge conflicts. Mastering these skills will not only help you collaborate more effectively on this project but will also be invaluable in your future professional endeavors.

Step by Step: Cloning, Branches, Committing, Pushing & Pulling

Here is a step by step breakdown of what you should achieve in this activity:

1. One person in your team should create a public GitHub repository:
 - Go to GitHub, click on 'New' to create a new repository, fill in the repository details and create the repository.
2. All members of the team should then be added to the repository as collaborators:
 - The person who created the repository can go to 'Settings' → 'Manage access' → 'Invite a collaborator' and then add the GitHub usernames of the other team members.
3. Each member should create a folder on their local machine where they will clone the repository then do the following:

```
1 git clone <repository-url> # Clone the repository
2 git checkout -b <branch-name> # Create and switch to a new branch
```

Copy

✦ Explain this code

4. Each team member creates a new branch and adds a `.txt` file with some placeholder text:

- Open your code editor, create a new file as a test with your name e.g. `sam.txt` and add some contents

5. Each member commits their code, and pushes it to their branch on GitHub:

```
1 git add sam.txt # Stage the changes
2 git commit -m "Add sam.txt" # Commit the changes
3 git push -u origin <branch-name> # Push the changes to the remote repository
```

Copy

✦ Explain this code

6. Each member makes a pull request:

- Go to the GitHub repository, click on 'Pull requests' → 'New pull request' → select your branch → 'Create pull request'.

7. Another team member reviews the pull request, and then merges it to the `main` branch.

- A team member can review the changes by going to the 'Files changed' tab in the pull request, and then approve and merge the pull request.

8. Carry this out for each team member and all of the pull requests have been merged

9. Everyone now pulls the latest code from the GitHub repository so they have the latest code on their machine.

Step by Step: Merge Conflict

Merge conflicts are something that is likely to happen. There are steps you can take (see guidelines below) to avoid this from happening, but in order to prepare you can create a merge conflict intentionally as practice. This is how you can do this:

1. All team members should switch to the `main` branch and pull the latest changes:

```
1 git checkout main
2 git pull
```

Copy

✦ Explain this code

2. Then, each team member should create a new branch.

```
1 git checkout -b <branch-name>
```

Copy

✦ Explain this code

3. Next, all team members should modify the **same line of the same file**, and then commit and push their changes to their branch on GitHub.

```
1 git add file.txt
2 git commit -m "Modify file.txt"
3 git push -u origin <branch-name>
```

Copy

✦ Explain this code

4. One team member should create a pull request and merge it into the `main` branch. Then, the other team members should create pull requests for their branches. Since they all modified the same line of the same file, there will be a merge conflict that needs to be resolved.

5. In order to resolve the merge conflict:

- The person who is responsible for merging the pull requests should go to the GitHub repository, click on the pull request with the merge conflict, and then click on 'Resolve conflicts'.
- GitHub will show the conflicting changes, and the person should choose which changes to keep or manually edit the file to resolve the conflict.
- After resolving the conflict, they should mark the conflict as resolved, commit the changes, and then merge the pull request.

6. Once the merge conflicts are resolved, you can delete the test .txt files created for the purposes of this activity to not have them in your repo.

Guidelines & Tips for Collaboration

Here are some guidelines for working with Git and GitHub:

1. **Descriptive Commit Messages:** Use descriptive commit messages to document the changes you made.
2. **Meaningful Names:** Use meaningful names for branches, files, and variables.
3. **Pull Latest Changes:** Always remember to `git pull` to have the latest changes of the project on your local machine before you start working.
4. **Avoid Merge Conflicts:** In the project planning, divide the work up so that two people are not required to work on the same file, to avoid merge conflicts.
5. **Communication:** When a merge conflict occurs, communication is important so that as a team you discuss how you plan to resolve the conflict, and agree on the final version of the code.
6. **Review Carefully:** Review merge conflicts carefully to ensure that work completed by someone is not accidentally removed by another team member.
7. **Branching Strategy:** It is common practice to create a "develop" branch and then different "feature" branches for the different features you are working on. The feature branches are then merged to the develop branch, and once everything is finished, the develop branch can be merged to the main branch.
8. **Comment Your Code:** Comment your code to explain the logic, especially in complex sections of the code. This will help your teammates understand the code better.
9. **Keep Main Branch Clean:** Keep the main branch clean and stable. Only merge code that is tested and working.
10. **Regular Communication:** Communicate with each other regularly and to discuss any issues or challenges you are facing.
11. **Project Management Tool:** Use a project management tool like [Trello](#) to keep track of tasks and their status.
12. **Refer to Previous Lessons:** Refer to previous lessons on Git, GitHub, and collaboration if necessary.

Part 2: Project Ideation (2 hours)

In this part of the activity, you will begin to generate ideas for your project and identify the dataset, API and/or website you will work with during the week. Here is a suggested outline of steps you can take as a team.

If you have already partnered with someone, chosen a topic, and researched and found the dataset, API and/or website you'll be using, you can skip this part. Just make sure you have business opportunities, hypotheses and problem statements defined before jumping into part 3.

1. **Individual Brainstorming (20 minutes):** Spend 20 minutes brainstorming ideas for topics and datasets you are interested in. Think about current events, social issues, or any other topics you are passionate about. Also consider the availability of data for your chosen topics. You should also think about potential business opportunities, hypotheses and problem statements that interest you as well.
2. **Share Ideas (15 minutes):** One by one, share your ideas with your partner. Explain why you are interested in each topic and what kind of analysis you think could be done with the data.
3. **Group Discussion (20 minutes):** As a group, discuss each idea and evaluate its feasibility, interest, and the availability of data. You can build on each other's ideas and combine them if you would like.
4. **Research Datasets (20 minutes):** Choose one or two topics and spend some time researching available data. Look for public data repositories, APIs, websites, or any other reliable sources of data.
5. **Present Findings (15 minutes):** Present the data found and explain how it could be used for the project. Discuss the pros and cons of each dataset, API and/or website, such as the amount of data cleaning required, the relevance of the data to the project idea, etc.
6. **Final Decision (20 minutes):** As a group, make a final decision on the project topic and dataset, API and/or website. Also decide on some business opportunities, hypotheses and problem statements you'll aim to answer and explore with your project. Make sure you are both on board with the decision and excited about the project!

Part 3: Project Management - Create a Kanban (1 hour)

In order to be successful in the project, being organized and managing tasks will be critical - this is especially true when you are working in teams. In this activity you will begin to create a Kanban board for the project.

The first step is to choose a tool to create the board and create an account. Trello is recommended, but you have some alternative options as well:

- Trello: <https://www.trello.com/>
- Asana: <https://www.asana.com/>
- Monday[.]com: <https://www.monday.com/>
- Notion: <https://www.notion.so/>
- ClickUp: <https://www.clickup.com/>

As a team, you should then starting thinking about the different tasks that you will be required to complete throughout the week. Here are some pointers to think about when creating your Kanban board:

1. **Define Tasks:** Start by listing all the tasks that need to be completed for the project. Think about all the different stages of the project, from data collection and cleaning to data analysis, visualization, and presentation. Also, consider any other tasks related to project management, communication, and documentation. Think back to the tasks completed during the week 2 mini project.
2. **Assign Tasks:** Assign each task to a team member. Make sure the workload is distributed evenly among all team members. If a task is too big, break it down into smaller, manageable sub-tasks.
3. **Set Deadlines:** Assign a deadline for each task. Be realistic about how much time each task will take and build in some buffer time for unexpected challenges.
4. **Prioritize Tasks:** Decide on the order in which the tasks need to be completed. Some tasks will be dependent on others, so make sure to take that into account when prioritizing.
5. **Update Regularly:** Make sure to update the Kanban board regularly. As soon as a task is completed, move it to the 'Done' column and start working on the next task. If a task is blocked for some reason, make a note of it on the Kanban board and discuss it with the team.
6. **Review Regularly:** Have regular team meetings to review the Kanban board and discuss any challenges or blockers. Make any necessary adjustments to the task assignments or deadlines.
7. **Keep It Visible:** Make sure the Kanban board is easily accessible to all team members - everyone should be able to see the status of the project at any time.
8. **Limit Work In Progress:** Limit the number of tasks that can be in progress at any one time. This will help to prevent bottlenecks and ensure that the team is focused on the most important tasks.

✨ Any doubt? Ask our AI Chatbot!



Lesson Completed

Mark as not completed

How would you rate the content of this unit?



PREVIOUS

← Agile

NEXT

Project Rubric →