**1. Introduction to GitLab and Project Creation**

- Begin by introducing GitLab and its role in version control and collaboration.

- Navigate to GitLab's homepage on your screen. Walk the team through the process of creating a new project.

- Click on "New project" -> "Create blank project".

- Explain the configurable options - Project Name, Project URL, Project Slug, Visibility Level, etc.

- Highlight the importance of the visibility level and the difference between Private, Internal, and Public.

- Finalize the creation process by clicking on "Create project".

**2. Adding Data and Exploratory Data Analysis (EDA)**

- Now, demonstrate how to add files to a GitLab repository. Navigate to the newly created project repository.

- Show how to drag and drop the Kaggle dataset CSV file into the designated folder on your local machine.

- Open this CSV file in Microsoft Visual Studio Code. Briefly walk through the data in the file, explaining what each column represents.

- Navigate to the terminal or command line interface. Demonstrate how to navigate to the directory containing the Kaggle dataset using the "cd" command.

- Here, take some time to explain some useful shell commands that can be used to navigate directories and manage files.

- Execute `git add .` to add the dataset to the staging area. Take a moment to explain the concept of the staging area, how it is a part of Git's two-step commit process, and why it's helpful (allows you to organize your commits).

- Follow this with `git commit -m "Add initial dataset"` to commit the changes. Explain the purpose of these Git commands and their importance in version control. Discuss the commit structure and how each commit represents a snapshot of the project at a certain point in time.

- Show them how to use `git status` to check the status of their repository.

**3. Adding Modeling Code**

- Create a new directory in your project folder named 'code' to hold your Python code file.

- Open the Python file in Visual Studio Code, briefly walk through the code, explaining the key parts of your model.

- Commit and push these changes to your GitLab repository using the terminal or command line with the commands `git add .`, `git commit -m "Add initial model code"`, and finally `git push`.

- You can use this step to introduce `git log` as a command to visualize the commit history and `git show` to see what changes were made in a specific commit.

**4. Presentation/Documentation**

- Introduce the concept of a README.md file, emphasizing its importance in providing a clear overview of the project, its purpose, and the contents of the repository.

- Open README.md in Visual Studio Code and demonstrate how to write a simple description of the repository, the project, the dataset, and the modeling code you've added.

- Commit and push the changes to GitLab (`git add .`, `git commit -m "Add README"`, `git push`).

**5. Cloning and Syncing the Repository**

- Now that you have a complete repository on GitLab, demonstrate how to clone it and sync it with your local machine.

- Navigate to a new folder directory on your local machine, open the terminal or command line, and use the command `git clone <repository\_url>` to clone the repository.

- Explain what cloning does and why it's necessary for collaboration.

- After cloning, show the contents of the new directory to confirm that all files from the GitLab repository are present.

- Discuss the concept of a Git workflow, explaining how changes move from your working directory, to the staging area, and then to the repository with each commit.