**Steps to Create a Project**

This document summarizes the key steps to set up and create a project, as described in the lecture.

**1. Introduction to the Course and Tools**

* The course involves programming concepts for data platform development.
* Core technologies include Python, SQL, Docker, Kafka, and Git/GitHub.
* The course focuses heavily on Python for most tasks.

**2. Setting Up the Project**

**2.1 Create a Repository in GitHub**

1. Log in to your GitHub account.
2. Create a new repository:
   * Use an appropriate name, such as data\_platform\_project.
   * Add a description (optional).
   * Make the repository public or private as needed.
   * Initialize with a README file.
   * Set the repository to use Python as the default language.

**2.2 Clone the Repository Locally**

1. Copy the repository URL from GitHub.
2. Open a terminal and navigate to your desired local directory.
3. Run the following command to clone the repository:
4. git clone <repository\_url>

**3. Setting Up a Virtual Environment**

**3.1 Create a Virtual Environment**

* Use the following command to create a virtual environment:
* python -m venv <env\_name>

Example:

python -m venv venv

**3.2 Activate the Virtual Environment**

* On **Windows**:
* venv\Scripts\activate
* On **Mac/Linux**:
* source venv/bin/activate

**3.3 Install Required Packages**

* Install dependencies within the virtual environment:
* pip install <package\_name>
* For example, install Jupyter Notebook:
* pip install ipykernel
* Use the command below to confirm the installation:
* pip list

**4. Organizing the Project**

**4.1 Set Up Project Structure**

* Create the following folders within the repository:
  + code/: For source code.
  + video\_lectures/: For lecture-related materials.
  + exercises/: For practice files and exercises.

**4.2 Rename Files or Folders in Terminal (Optional)**

* Use the mv command to rename files or folders:
* mv <old\_name> <new\_name>

**4.3 Add a .gitignore File**

* Create a .gitignore file to exclude unnecessary files (e.g., Mac’s .DS\_Store):
* echo ".DS\_Store" >> .gitignore

**5. Git Workflow**

**5.1 Track Changes**

* Add files to the staging area:
* git add .
* Commit the changes:
* git commit -m "Initial project setup"

**5.2 Push Changes to GitHub**

* Push local changes to the remote repository:
* git push

**6. Best Practices**

* Use a virtual environment to avoid package conflicts.
* Structure your project into well-organized directories.
* Regularly commit and push changes to GitHub.
* Use .gitignore to exclude files you don’t want in the repository.
* Follow Python coding standards for consistency.

This document serves as a guide for setting up and managing a project, ensuring smooth development and collaboration throughout the course.