L01\_WillianeYarro\_ITAI2373

Course: ItAI2373

Assignement: Intro to Jupyter and Github

Date: May 8, 2025

Two Page Journal/Summary

Introduction:

In the contemporary digital and data-centric landscape, acquiring the ability to manage code and collaborate effectively through platforms such as GitHub is an essential competency. This laboratory exercise was crafted to familiarize students with the fundamental tools utilized in programming and data science—specifically, GitHub for version control and Jupyter Notebook for interactive computing. By engaging in the process of establishing a GitHub repository, configuring a local development environment, and utilizing Jupyter Notebook, I not only enhanced my technical skills but also faced and resolved authentic setup challenges. The assignment reflects the operational dynamics of professionals in collaborative, open-source, and academic environments, equipping me to contribute confidently to projects and analyze code within a structured framework. Introduction

In the contemporary digital and data-centric landscape, acquiring the ability to manage code and collaborate effectively through platforms such as GitHub is an essential competency. This laboratory exercise was crafted to familiarize students with the fundamental tools utilized in programming and data science—specifically, GitHub for version control and Jupyter Notebook for interactive computing. By engaging in the process of establishing a GitHub repository, configuring a local development environment, and utilizing Jupyter Notebook, I not only enhanced my technical skills but also faced and resolved authentic setup challenges. The assignment reflects the operational dynamics of professionals in collaborative, open-source, and academic environments, equipping me to contribute confidently to projects and analyze code within a structured framework.

In the contemporary digital and data-centric landscape, acquiring the ability to manage code and collaborate effectively through platforms such as GitHub is an essential competency. This laboratory exercise was crafted to familiarize students with the fundamental tools utilized in programming and data science—specifically, GitHub for version control and Jupyter Notebook for interactive computing. By engaging in the process of establishing a GitHub repository, configuring a local development environment, and utilizing Jupyter Notebook, I not only enhanced my technical skills but also faced and resolved authentic setup challenges. The assignment reflects the operational dynamics of professionals in collaborative, open-source, and academic environments, equipping me to contribute confidently to projects and analyze code within a structured framework.

Purpose of This Lab:

The objective of this laboratory session was to familiarize students with two essential instruments utilized in contemporary software development and data science: GitHub and Jupyter Notebook. GitHub functions as a version control system that facilitates collaborative programming, tracking of historical changes, and management of remote repositories, whereas Jupyter Notebook offers an interactive platform for composing and executing Python code in real-time. Through the completion of this lab, I intended to cultivate fundamental competencies in establishing a version-controlled environment, utilizing notebooks, and resolving technical setup issues. The activity was structured not merely to accomplish specific tasks but to foster a more profound comprehension of how these tools are integrated into a professional development workflow.

1. What I Accomplished

Setting Up and Exploring Github

Part 1: Establishing and Navigating GitHub (20 minutes)

1.1 Registering for a GitHub Account: I began by visiting https://github.com and registering for a complimentary GitHub account with my email address. After confirming my account, I logged in to view my dashboard.

* 1. Navigating the GitHub Interface: After logging in, I examined GitHub’s user interface. I assessed the dashboard, which displays recent activities and provides access to all repositories. I navigated through the tabs for Repositories, Issues, and Pull Requests to gain insight into how developers collaborate and manage code on GitHub.
  2. 1.3 Establishing a New Repository:
  3. I clicked on the “+” icon located in the upper-right corner and chose “New repository.” I designated the repository as jupyter-exploration and selected the option to initialize it with a README file. This README acts as an introductory guide to the contents of the repository.
  4. 1.4 Executing My Initial Commit:
  5. I selected “Edit” on the README file and included a brief overview of the lab session. Subsequently, I committed this modification directly to the main branch with the message: “Initial commit with lab description.” This represented my first version-controlled update.
  6. To verify the commit and monitor file history, I also executed the following commands using Git bash

git status

git commit -m "Added my First Notebook"

git push origin main

These commands enabled me to confirm the status of my working directory, create a new commit accompanied by a message, upload the modifications to GitHub, and examine the commit history.

Part 2: Engaging with Jupyter Notebooks (35 minutes)

2.1 Accessing Jupyter Notebooks: Rather than utilizing an online platform such as Google Colab or AWS SageMaker, I opted to install and set up Jupyter Notebook on my local machine. I proceeded with the installation:

* **Python 3.13.4** from <https://www.python.org> (ensuring to add it to my PATH)
* **Visual Studio Code** from <https://code.visualstudio.com>
* **Git Bash** from <https://gitforwindows.org>
* **Jupyter and Python extensions** in VS Code via the Extensions Marketplace

2.2 Initiating Jupyter Notebook within VS Code:

I opened Visual Studio Code, navigated to the local directory where my GitHub repository had been cloned, and created a new file titled My\_First\_Notebook.ipynb. I examined the notebook interface, which included the cell type switcher (Markdown versus Code) and the toolbar for executing cells.

Following the installation of Python 3.13.4, I ensured that it was set as my default Python interpreter (or 'kernel') in VS Code. This was accomplished by clicking on the kernel name located in the upper-right corner of the notebook interface and selecting the interpreter path that directed to Python 3.13.4. This step was essential, as choosing the appropriate kernel guarantees that the code cells run correctly within the environment where Jupyter and the necessary packages are installed.

2.3 Created My Frist Notebook and Inserted Mardown Cell and wrote:

My first markdown cell in Jupyter.

 I inserted a **Code cell** and wrote:

print("Hello, World!")

**2.4 Saving and Sharing My Notebook:**

I saved the notebook, then opened **GitHub Desktop**. The software automatically recognized the changes. I wrote a commit message: “Add my first Jupyter Notebook,” and clicked **Commit to main**. Finally, I clicked **Push origin** to upload the notebook to GitHub.

This laboratory experience provided me with the skills to:

Establish a GitHub repository and perform my initial commit

Utilize GitHub Desktop for cloning repositories, committing modifications, and pushing updates

Employ Git Bash to run commands such as git status, git commit -m, and git push origin main

Install Python version 3.13.4 and set up Visual Studio Code with the required extensions

Choose the appropriate Python kernel in Jupyter to correspond with the environment (Python 3.13.4)

Create, execute, and oversee Jupyter Notebooks

Address challenges like determining the correct repository path using Git Bash

Activate Long Path Support in Windows to mitigate installation complications

I came to understand that version control is a crucial aspect of programming, serving not only for collaboration but also for managing updates and reverting changes. Additionally, I acquired hands-on experience in establishing a comprehensive development environment from the ground up.

My github link: https://github.com/joelleyarro03/jupyter-exploration.git

**Citations**

* GitHub: <https://github.com>
* GitHub Desktop: https://desktop.github.com
* Python: <https://www.python.org>
* Jupyter Documentation: https://jupyter.org
* Git Bash: <https://gitforwindows.org>
* Visual Studio Code: <https://code.visualstudio.com>
* Long Path Support Info: https://learn.microsoft.com/en-us/windows/win32/fileio/maximum-file-path-limitation
* Git CLI Reference: https://git-scm.com/docs