

To: Dr. Allison Hutchison

From: Star Command [Aryaa Pai , Chris Hales , Emily Parker, Eva Zhang, Saranya Sivaram]

Subject: Project Proposal for Jupyter Notebooks and Command Line Interfaces User Guide

Date: March 20, 2022

The purpose of this memo is to present our intentions and plans for a Jupyter Notebooks and Command-Line Interfaces user document. They are both heavily used in Cornell Engineering and in industry, and we will provide introductory knowledge as described in this project proposal.

I. Problem Statement

Many introductory programming courses in Cornell's College of Engineering require the use of command-line interfaces (CLIs) to download programs such as Jupyter Notebook. Jupyter Notebooks allow users to neatly present code and documentation; proper understanding of CLIs can give users more control over their operating system and execute tasks faster [1,2]. For students new to programming, these tools are essential but can be very intimidating without experience. Unfortunately, there is a significant lack of reference documentation for students to review, especially from Cornell.

II. Background and History of Jupyter Notebooks and CLI

In Computer Science and other programming related fields, Jupyter Notebooks and CLIs are popular and versatile tools for anyone to use [3]. Not only are they used in many Cornell Engineering courses, but they are also the industry standard for Data Science [4 - 7]. Jupyter Notebooks are incredibly useful for bridging the gap between displaying technical code and the logical explanations of its results [8]. Further, CLIs connect a user to their computer's operating system (OS), presenting prompts for the user to respond to with requests [9].

Our goal is to provide students of the Cornell College of Engineering with a resource that can help guide them to properly set up and use Jupyter Notebook and CLIs effectively. We will first introduce users to Jupyter Notebooks as it is a simpler platform. Then, we plan to integrate Jupyter with CLIs.

III. Current User Documents and Important Areas to Improve

There is an ever growing importance for understanding how to properly code and analyze results [10]. The current resources available are lacking in some areas, and we plan on improving the user experience.

Jupyter Notebook allows embedded multimedia for illustrative explanations; furthermore, it contains HTML and LaTeX components in its markdown blocks [8]. As such, formatting these cells to show exact formulas, links, or other content can enhance the learning experience. However, the only resources available are random websites that list information without proper organization.

Jupyter Notebook has access to countless useful Python packages [11, 12]. There are limited guides focused on how to import and install packages, especially for MacOS. For each individual package, the official documentation can also be intimidating for someone unfamiliar with programming.

Similarly for CLI, a lot of improvements are needed for documentation. There is no detailed guide that teaches users how to complete basic operations such as download packages and running code, especially for MacOS [13, 14]. Finally, resources that integrate both of these important tools are scarce.

IV. Tasks

The user document will be in the form of a website which contains a range of step-by-step tutorials. Each tutorial will be focus on a key topic:

- 1) downloading Jupyter Notebooks
- 2) performing advanced tasks in Jupyter Notebooks
- 3) using CLIs

The website will serve as the entry point of the user document and provide the user with a journey map for navigating the tutorials. We plan to create instructional documentation in Jupyter Notebooks as it will familiarize the user with the software itself. In the following section, we have described the specific tasks required to complete the project.

A. Tutorial 1: Installation of Jupyter Notebook using Anaconda (**Chris**)

The first tutorial will be hosted on a web page and outline the steps to download Jupyter Notebook using Anaconda, a package manager [15]. We chose Anaconda because of its simple graphical user interface, and CS 1110: Introduction to Computing Using Python and other introductory engineering classes recommend using it to download Python [16].

B. Tutorial 2: Writing Python Code in Jupyter Notebooks (**Emily**)

After the installation, our guide will teach users how to perform basic operations, manipulate strings, write functions, create objects, etc. Users without any background knowledge in computing should be able to follow this tutorial.

C. Tutorial 3: Using Cells and Markdown in Jupyter Notebooks (**Saranya**)

Unlike other popular coding environments, Jupyter gives users the ability to create and use cells. Cells are especially useful to beginner programmers because it allows the user to break code into simpler parts, making testing and debugging easier.

Code serves as a technical document, and good programmers comment their code. Jupyter integrates Markdown in cells, allowing software engineers to create more effective technical documents. We aim to teach users the Markdown language, which will help them to better communicate their ideas.

D. Tutorial 4: Plotting in Jupyter Notebooks (**Eva**)

Visualization is an important part in data science and computing fields. Jupyter supports the Matplotlib package, which allows users to plot data using Python. There are many commands to add titles or adjust layouts, and they are hard to master without proper tutorials. Matplotlib is also notorious for having

hard-to-understand documentation [source?]. We will provide a detailed guide for frequently used commands.

E. Tutorial 5: Navigating Packages in Jupyter Notebooks (**Chris and Eva**)

Jupyter supports many packages, which are commonly used by programmers and engineers alike, including Statistics, SciPy, ORTools, and more. Often the official documentation is difficult to grasp, using technical jargon that is confusing for beginners. We intend to create a more beginner-friendly document that steps through package download and uses less technical language. Perhaps more importantly, when the official documents are insufficient, our guide will teach users how to utilize search terms and navigate resource-based sites, like stackexchange.com and stackoverflow.com.

F. Tutorial 6: Introduction to the Unix Commands (**Aryaa**)

Unix commands are built-in programs which can be invoked to perform tasks or get information about the operating system. This tutorial will be hosted on a separate Jupyter notebook and introduce basic Unix commands. It will also allow users to practice within the Mac Terminal. We have chosen to focus on the Mac Terminal because most college students use MacOS [17]. The tutorial will include best practice for CLIs and highlight some more vulnerable commands to avoid.

G. Tutorial 7: Text Editing with CLI (**Saranya**)

Many courses at Cornell require students to run code on remote servers or electronics like Raspberry Pi, which can only be accessed using CLIs [18]. In such cases editing files using CLI based text editors can be a useful skill. This tutorial will focus on creating, viewing and editing files using Vim, a commonly used text editor within the command-line.

H. Website (**Aryaa**)

The website will be hosted on GitHub and will contain a basic landing page along with a road map to complete the tutorials. If time permits, we will add functionality for users to track their progress.

V. Task Schedule and Workflow

As per our original team workflow agreement, we are continuing to use a Trello board to manage our workflow and tasks schedule. The Trello board helps us designate tasks within each deliverable to specific team members, and keep track of the progress of the assignment being worked on.

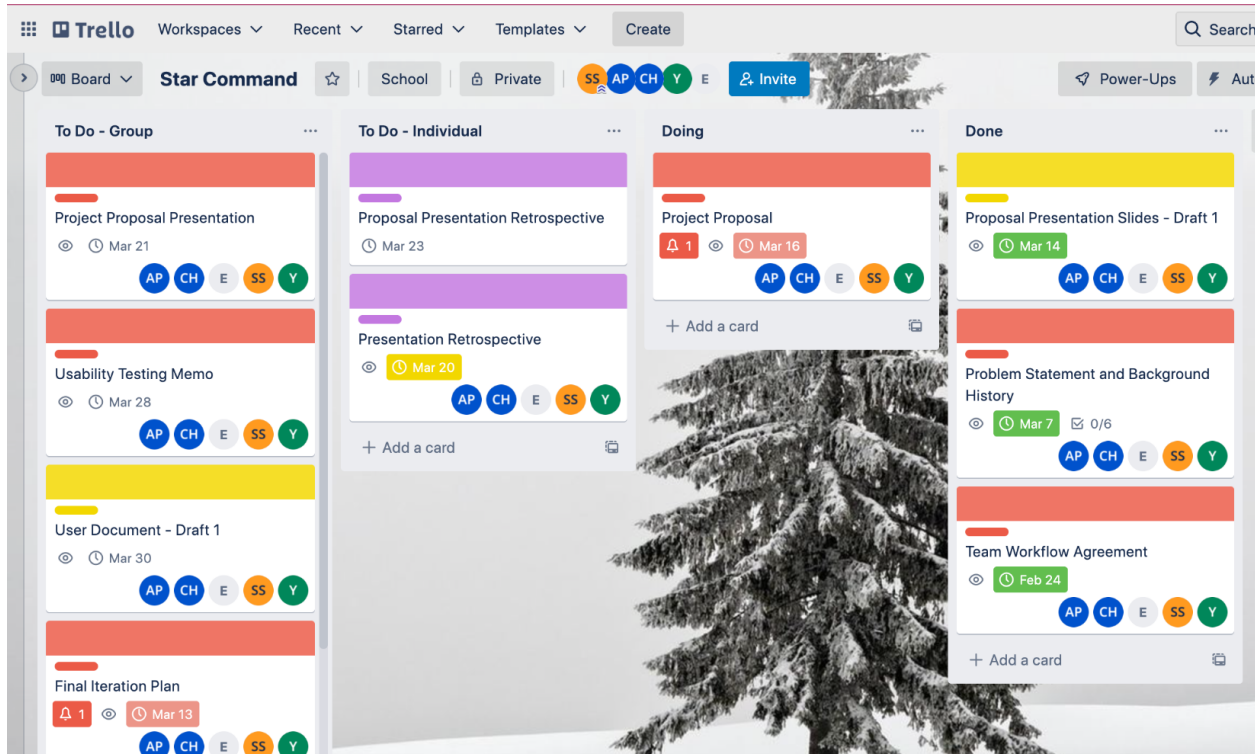


Fig. 1. Star Command's Trello Board

We categorized the upcoming deliverables for Project 2 into group (marked red) and individual (marked purple) to-do lists on our Trello board. Each deliverable has its own card that includes details of the assignment such as the submission deadline and owners of specific sub-tasks within the assignment. Our team has agreed to set assignment submission dates on Trello a day before the actual deadline, to account for unexpected problems.

As seen in Figure 2, each card contains a brief description of the assignment and the sub-tasks. These sub-tasks are divided amongst the team members which are agreed upon through a team meeting. A check mark indicates the completion of a sub-task, and we are able to view the progress of the assignment through the progress bar feature that Trello provides.

Project Proposal

in list [Doing](#)

Members: AP, CH, E, SS, Y, +

Labels: Submission, +

Due date: Mar 16 at 11:59 PM overdue ▼

Description Edit

The team will describe its topic and the content that will be covered. Building upon the Problem Statement & Background/History, your team should now refine its ideas or add more aspects to create a fuller vision of your final team project deliverables. After your individual or team proposal presentations last week, you should have a better idea of the content areas and medium/format for the user document.

To Do's Hide checked items Delete

60%

- Purpose statement @evaz1121
- Problem Statement and Background history + Revisions @chris_hales
- Scope - tasks to be done @aryaapai
- Scope - team workflow @saranyasivaram
- Evidence for the existence of the problem @emw2321 🕒 👤 ⋮

Add an item

Add to card

- Members
- Labels
- Checklist
- Dates
- Attachment
- Custom Fields

Add dropdowns, text fields, dates, and more to your cards.

[Start free trial](#)

Power-Ups

- Add Power-Ups

Automation ⓘ

- Add button

Actions

- Move
- Copy
- Make template
- Watch ✓

Fig. 2. Example Card for the Assignment 'Project Proposal'

References

- [1] N. Nilakshi, “Mastering the terminal : All you need to know,” *Medium*, 21-Aug-2021. [Online]. Available: <https://medium.com/codex/mastering-the-terminal-all-you-need-to-know-cd9fda-d93c8b>. [Accessed: 20-Mar-2022].
- [2] J. Rocheleau, Nikita, A. Monus, A. KS, N. Young, H. Lim, and T. Firdaus, “5 reasons why developers should learn command line,” *Hongkiat*, 24-Nov-2020. [Online]. Available: <https://www.hongkiat.com/blog/developers-command-line/>. [Accessed: 20-Mar-2022].
- [3] J. W. M. University, J. Wang, M. University, L. L. M. University, L. Li, Andreas Zeller CISPA Helmholtz Center for Information Security, A. Zeller, CISPA Helmholtz Center for Information Security, N. C. S. University, Kaist, and O. M. V. A. Metrics, “Better code, better sharing: Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: New ideas and emerging results,” *ACM Conferences*, 01-Jun-2020. [Online]. Available: https://dl.acm.org/doi/abs/10.1145/3377816.3381724?casa_token=2Q5i_W28-18AAA%3A62Avbf1VOaofZrOvoWXnVJKrmYEi24pguvzf_ThrJg0B0J0xjyoeOkYiWk38f1s8Q6zISZzMhpULNQ. [Accessed: 20-Mar-2022].
- [4] “Toolbox by Jupyter, it all makes sense - media.nature.com.” [Online]. Available: <https://media.nature.com/original/magazine-assets/d41586-018-07196-1/d41586-018-07196-1.pdf>. [Accessed: 21-Mar-2022].
- [5] “Citing ipython¶,” *IPython*. [Online]. Available: <https://ipython.org/citing.html>. [Accessed: 20-Mar-2022].
- [6] A. Cardoso, J. Leitão, and C. Teixeira, “Using the Jupyter notebook as a tool to support the teaching and learning processes in engineering courses,” in *The Challenges of the Digital Transformation in Education*, 2019, vol. 917, no. pp. 227–236.
- [7] Ashwin Pajankar and Ashwin Pajankar1, “Practical python data visualization,” *SpringerLink*. [Online]. Available: <https://link.springer.com/book/10.1007/978-1-4842-6455-3>. [Accessed: 20-Mar-2022].

- [8] D. Toomey, “Learning jupyter: Paperback,” *Barnes & Noble*, 30-Nov-2016. [Online]. Available: <https://www.barnesandnoble.com/w/learning-jupyter-dan-toomey/1124015385>. [Accessed: 20-Mar-2022].
- [9] J. Ellis, “Command line interface,” *Comms InfoZone*, 01-Jan-1970. [Online]. Available: <https://www.comms-express.com/infozone/article/command-line-interface/>. [Accessed: 20-Mar-2022].
- [10] J. Wang, L. Li, and A. Zeller, “Better code, better sharing: on the need of analyzing Jupyter Notebooks,” *arXiv.org*, 12-Jun-2019. [Online]. Available: <https://arxiv.org/abs/1906.05234>. [Accessed: 20-Mar-2022].
- [11] Real Python, “Python modules and packages – an introduction,” *Real Python*, 19-Jun-2021. [Online]. Available: <https://realpython.com/python-modules-packages/>. [Accessed: 20-Mar-2022].
- [12] “Python deep learning: Practical machine learning ...” [Online]. Available: <https://www.wiley.com/en-us/Python+Deep+Learning%3A+Practical+Machine+Learning+Application+Frameworks+with+Tensorflow+and+Pytorch-p-00341432>. [Accessed: 21-Mar-2022].
- [13] M. McDonnell, *Pro vim*. Berkeley, CA: Apress, 2014. Available: <https://link-springer-com.proxy.library.cornell.edu/book/10.1007/978-1-4842-0250-0>
- [14] I. Goldstein, “What! no Gui? -- teaching a text based command line oriented introduction to Computer Science Course.,” *Information Systems Education Journal*, 31-Jan-2019. [Online]. Available: <https://eric.ed.gov/?id=EJ1206681>. [Accessed: 20-Mar-2022].
- [15] “Installing Jupyter Notebook,” *Installing Jupyter Notebook - Jupyter Documentation 4.1.1 alpha documentation*. [Online]. Available: <https://test-jupyter.readthedocs.io/en/latest/install.html>. [Accessed: 20-Mar-2022].

- [16] “Python in CS 1110,” *CS 1110: Python*. [Online]. Available: <https://www.cs.cornell.edu/courses/cs1110/2018fa/materials/python.php#install>. [Accessed: 20-Mar-2022].
- [17] B. Content, “Here is why 71 percent of college students prefer MacBooks over Windows pcs,” *The Collegian*, 22-Dec-2021. [Online]. Available: <https://www.kstatecollegian.com/2021/12/21/here-is-why-71-percent-of-college-students-prefer-macbooks-over-windows-pcs/>. [Accessed: 20-Mar-2022].
- [18] R. Barnes and R. runs R. P. Press, “How to open terminal and use the command line,” *The MagPi magazine*. [Online]. Available: <https://magpi.raspberrypi.com/articles/terminal-help>. [Accessed: 20-Mar-2022].