Individual Assignment 1  
Apple Inc. Stock Price Visualization



25th February 2024

Ngoc Pham (25100660)

36104 - Data Visualization and Narratives

Master of Data Science and Innovation

University of Technology of Sydney

Table of Contents

[**1.Executive Summary 3**](#_3lf7g52hwelp)

[1.1. Overview of the Project 3](#_45obzpxeflc)

[1.2. Problem Statement 3](#_r2xncrft8uz5)

[1.3. Achieved Outcomes and Results 3](#_odor0e6erpn8)

[**2. Introduction 4**](#_sdsesyeiyz6a)

[2.1. Key Objectives or Goals of the Project 4](#_hsgkufnkil25)

[2.2. Stakeholders and Their Requirements 4](#_re7kvmgv11lh)

[2.3. Addressing the Requirements 4](#_jo4jyujxsay0)

[**3. Web App Presentation 5**](#_6h2c813v9872)

[3.1. Functionality 5](#_ukg31xpz1602)

[3.2. Instructions To Launch the App 8](#_45n0ez5etgvf)

[3.3. Users and Use Cases 9](#_7sate4zdf1fr)

[3.4. Commercial Potential 10](#_y2suvb77c0r7)

[3.5. Limitations 11](#_ckxdqrnc18jb)

[3.6. Improvements 11](#_qdd4xpi8qgl2)

[**4.Reflecting On Building Data Product 12**](#_l1zh5dq1fnet)

[**5.Collaboration 14**](#_834i11sn7ad7)

[5.1. Individual Contributions 14](#_5fycwkjr7bjp)

[5.2. Group Dynamic 16](#_4tp33skt3su4)

[5.3. Ways of Working Together 16](#_t9y1jnesjjm)

[5.4. Issues Faced 17](#_btkuk6bvcn5n)

[**6.Conclusion 19**](#_9hr11g79asdx)

[**7.References 20**](#_ma3y8ytjvp1y)

[**8.Appendix 21**](#_1uh2ul2g7q9w)

[Work Log 21](#_x6qgr72uhepk)

[VS Code GIT integration 23](#_z957j1jeecf2)

[Application screens 26](#_wusix723f6mf)

# 

# 1.Executive Summary

## 1.1. Overview of the Project

The primary objective of the project was to design and implement an interactive web application capable of conducting exploratory data analysis (EDA) on input datasets in the form of CSV files. The significance of the project lies in providing users, particularly data scientists and analysts, with an intuitive tool for gaining insights into their data without the need for complex programming or software. By focusing on a user-friendly interface developed through Streamlit in Python 3.9, the project aimed to streamline the EDA process, facilitating a quicker transition from data to insights.

## 1.2. Problem Statement

In the field of data science, a fundamental step in any data-related project is understanding the dataset at hand. This step often involves technical complexities and demands a considerable amount of time. The problem addressed by this project was the need for an accessible, interactive, and efficient web application that can assist users in performing EDA with minimal setup and learning curve.

## 1.3. Achieved Outcomes and Results

The project successfully delivered a fully functional web application that meets the following criteria:

* A user-friendly menu for uploading CSV files, making the application accessible to users with different technical backgrounds.
* An interactive container with four tabs enabling users to examine the overall dataset, and analyze specific numeric, text, and datetime columns.
* Provision of a comprehensive set of EDA functionalities, such as summary statistics, data visualizations, and trend analysis for the selected columns.
* Implementation of the application in Python 3.9 using Streamlit, ensuring a responsive and dynamic user experience.

# 2. Introduction

## 2.1. Key Objectives or Goals of the Project

1. To create an interactive web application that enables users to upload and analyze CSV files without extensive coding knowledge.
2. To allow dynamic exploration of datasets through a user-friendly interface, with functionalities to view overall information and delve into specific data types like numeric, text, and datetime.
3. To provide users with immediate, visual feedback on their data through charts and graphs that elucidate data distributions, trends, and statistical summaries.
4. To develop an application that is accessible to a range of users, from novice data enthusiasts to seasoned data scientists.
5. To ensure the application is built on a robust, scalable framework that can handle datasets of varying sizes and complexities.

## 2.2. Stakeholders and Their Requirements

The primary stakeholders for this project include:

1. Data Scientists: Require a tool that can quickly provide an overview and detailed analysis of data to assist in hypothesis formation and data preprocessing.
2. Data Analysts: Need to interpret data accurately and generate reports or insights that can influence business decisions.
3. Business Users: Want to extract data insights without delving into technical details or relying heavily on IT departments.

## 2.3. Addressing the Requirements

* By providing a streamlined process for uploading and analyzing data, the application reduces the complexity and time typically associated with EDA.
* The application focuses on a high degree of interactivity and visualization, which are crucial for a comprehensive understanding of the data, thus meeting the needs of less technical stakeholders.
* For business users, the application serves as a decision support tool, offering insights without the prerequisite of technical expertise.

# 3. Web App Presentation

## 3.1. Functionality

The web app consists of a main UI and has 4 tabs providing different information as detailed below:

**Main UI**

Users can upload their data in CSV format directly into the application. Validation is in place if any other file other than CSV is uploaded.

**DataFrame Tab**

This tab provides an overview of the uploaded file having the following specifications:

1. DataFrame (A Static Table) that displays:

* The number of rows
* The number of columns
* The number of duplicated rows
* The number of rows with missing values.
* The number of numeric columns
* The number of text columns
* The number of columns with date format
* The number of columns with boolean format

2. Columns (A Static Table) that displays:

* A list of columns present in the file.
* The data type (text, numeric, date) of each column.
* The memory usage of each column
* Index is also included as a new row along with the other column names

3. Explore Dataframe

This feature allows the user to further explore the data by displaying a table that is affected by the user inputs in the following fields

* A slider that has a range of ‘5’ to ‘no of rows of the dataset’. As the slider is moved, the display will dynamically update to show the selected number of rows.
* A radio button for selecting the method to display rows from the chosen table:
  + 'Head' to display the top rows of Selected dataset,
  + 'Tail' to display the bottom rows of Selected dataset,
  + 'Sample' to display the random Sample of Selected dataset.

**Numeric Serie Tab**

The second tab is designed to enhance the user's comprehension of the data within each numeric column of the dataset. A selection box is provided that displays a list of all the numeric columns in the datafile that a user can choose from. The data from the chosen column will be displayed in the following manner:

1. Numeric Column (A Static Table) that displays:

* The number of unique values
* The number of missing values
* The number of occurrence of 0 value
* The number of negative values
* The average value
* The standard deviation value
* The minimum value
* The maximum value
* The median value

2. A Histogram Chart will display the occurrences for each value.

3. Most Frequent Values (An Interactive Dataframe table) that shows occurrences and percentages of the top 20 most frequent values.

**Text Serie Tab**

The third tab functions similarly to the second tab but displays data of columns with text in them. A selection box is provided that displays a list of all the text columns in the datafile that a user can choose from. The data from the chosen column will be displayed in the following manner:

1. Text Column (A Static Table) that displays:

* The number of unique values
* The number of missing values
* The number of rows with empty string
* The number of rows with only whitespaces
* The number of rows with only lower case characters
* The number of rows with only upper case characters
* The number of rows with only alphabet characters
* The number of rows with only numbers as characters
* The mode value

1. A Bar Chart will display the occurrences for each value.
2. Most Frequent Values (An Interactive Dataframe table) that shows occurrences and percentages of the top 20 most frequent values.

**Datetime Serie Tab**

The fourth tab, very similar to both the above given tabs, displays any column which has date and time related information in it. . A selection box is provided that displays a list of all the date and time columns in the datafile that a user can choose from. The data from the chosen column will be displayed in the following manner:

1. Date Column (A Static Table) that displays:

* The number of unique values
* The number of missing values
* The minimum date
* The maximum date
* The number of occurrence of days falling during weekend (Saturday and Sunday)
* The number of weekday days (not Saturday or Sunday)
* The number of cases with future dates (after today)
* The number of occurrence of 1900-01-01 value
* The number of occurrence of 1970-01-01 value

1. A histogram Chart.
2. Most Frequent Values (An Interactive Dataframe table) that shows occurrences and percentages of the top 20 most frequent values.

## 3.2. Instructions To Launch the App

**Environment Setup**

The repository containing the project is at:

<https://github.com/Paiynthalir/94692_DSP_AT3_Group25.git>

1. First open Terminal/Command Prompt (depending on whether your machine’s operating system is MacOS or Windows) and type:

*cd [desired\_directory]*

where *desired\_directory* is your preferred location containing the folder of the project. By default, if you use *git clone* without *cd,* the app folder location will be set to your user profile folder.

1. Clone the repo using *git clone*, simply copy and paste the following command into your window of your Terminal/Command Prompt

*git clone https://github.com/Paiynthalir/94692\_DSP\_AT3\_Group25.git*

This step assumes that you have access to the private repository.

1. Install the necessary packages in your machine to ensure the application behaves as expected. The required packages are specified in the *requirements.txt* file. Copy and paste this command into your Terminal/Command Prompt. Ensure you have Python preinstalled to use *pip.*

*pip install -r requirements.txt*

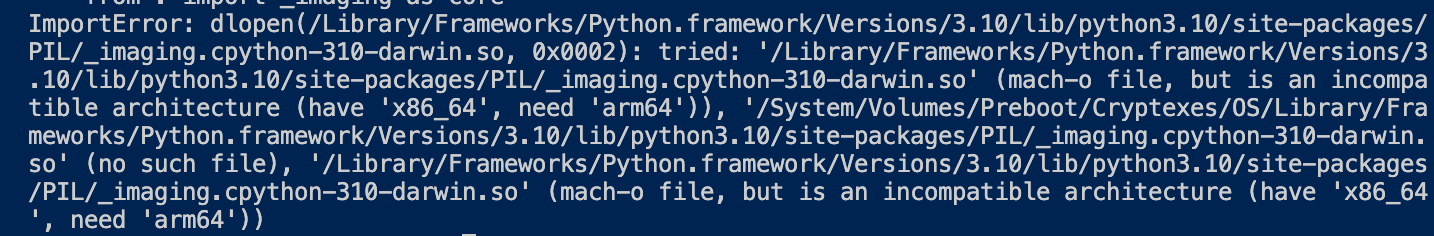
1. Open your Terminal/Command Prompt and paste the following code to run the application. After successful completion, the Streamlit application will open in a tab of your default browser.

*streamlit run /[desired\_directory]/94692\_DSP\_AT3\_Group25/app/streamlit\_app.py*

Alternatively, the app can be launched in any browser through this URL

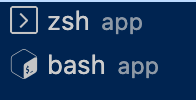
<https://dataexplorer-paiynthalir.streamlit.app/>

**Potential Error for Apple Chip Users**

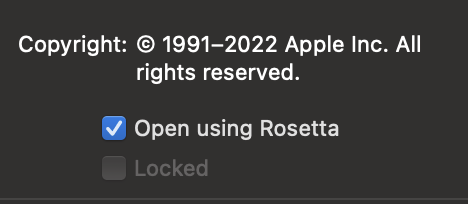
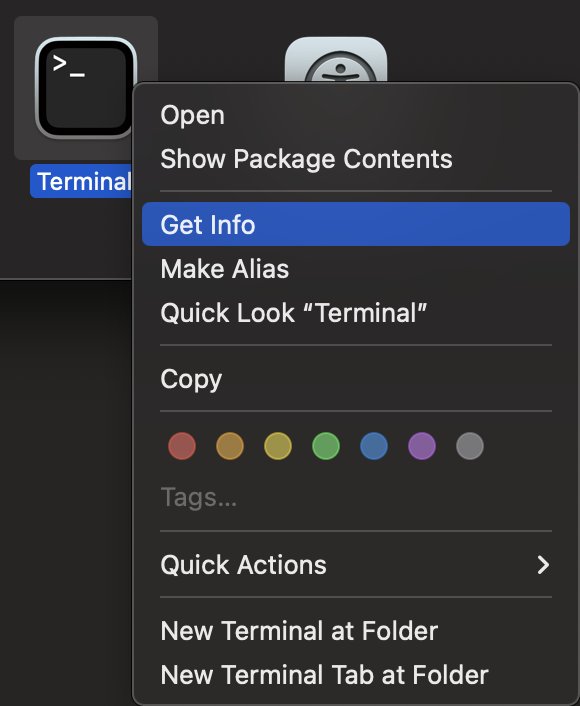


Solution:

1. Change the shell to bash



1. Get Info in Terminal and click on Open using Rosetta



## 3.3. Users and Use Cases

|  |  |
| --- | --- |
| Use Case 1 | EDA |
| Actor | Data Scientist, Data Analyst, Business Users |
| Flow | The users will open the provided link, log in, then upload their file. According to the data in the file, EDA can be performed with the given tabs |
| Pre Requisites | Logged into Streamlit with the website access.  The file should be in .csv format. |
| Benefits | * Provides a deeper understanding of data, highlighting structure, trends, and anomalies. * Quickly identifies data quality issues like missing values and outliers. * Significantly reduces the time spent on initial data analysis stages. * Makes complex data analysis accessible to non-experts with a user-friendly interface. * Offers interactive visualizations for revealing hidden patterns and correlations. * Allows customized analysis, including choosing data subsets and visualization types. * Improves decision-making for advanced data tasks and model building. * Enhances team collaboration with clear and intuitive data presentation. |

|  |  |
| --- | --- |
| Use Case 2 | Making code changes |
| Actor | Data Scientist, Data Analyst |
| Flow | Users can log into Git, access the repository where the code is present, modify the Python code on their local desktop, and then push these changes to the main repository, which will subsequently update the application. |
| Pre Requisites | Git log in id with approved repository access  VS code and python in local desktop with necessary libraries and packages installed |
| Benefits | * Enables robust version control for tracking and reverting changes over time. * Facilitates collaboration and peer code review, enhancing code quality. * Streamlines workflow with automatic testing and deployment through CI/CD. * Allows isolated testing of code changes in local environments, reducing bug risk. * Increases development speed and efficiency with simultaneous work on different features. * Provides easy access and backup of code, supporting remote and distributed work. * Supports scalability and organized code structure, essential for project growth. * Offers valuable documentation and traceability for each change via commit messages and Git history. |

## 

## 3.4. Commercial Potential

To commercialize this application successfully, consider the following strategies:

* Embedded Web App Development:
  + Develop gateways to enable interaction with other websites and platforms.
  + Make the application an integral part of various business processes, expanding our platform’s user base.
* Freemium Model:
  + Offer a basic version with certain limitations like file size or limit API calls and provide a paid version with enhanced file size support, automated data analysis pipeline or options of choosing machine learning models for specific datasets.
* Customer Support and Training:
  + Offer premium customer support, training, and consulting services and products to combat users' challenges.
  + Ensure customer satisfaction and resolve any issues promptly.

## 3.5. Limitations

While the application is well-designed and provide great quality data overview, it has certain limitations:

* File Size Limitation:
  + Large CSV files may not load or process efficiently as the file size allowed in Streamlit are only up to 200MB, leading to performance issues.
* Lack of Advanced Analysis Tools:
  + The application may not offer advanced statistical analysis or machine learning capabilities required for in-depth data analysis.
* Data Import Compatibility:
  + Handling all CSV variations, especially those with delimiter, encoding, or format issues, may pose challenges during importing operations.

## 3.6. Improvements

To enhance the application further, consider implementing the following improvements:

* Advanced Data Visualization:
  + Include advanced charting and graphing capabilities, supporting various chart types for more effective data trend visualization.
* Data Transformation Tools:
  + Add powerful data transformation features such as column merging, splitting, and pivoting, mathematical operations, and calculated field creation.
* Machine Learning Integration:
  + Incorporate machine learning and predictive analytics capabilities for advanced data analysis and pattern recognition.
* User Feedback Mechanism:
  + Implement a user feedback system within the application, allowing users to suggest improvements and report issues directly to the development team.
* Incorporate various file structures:
  + Currently the app only allows handling of .csv format of the files, future version will include the app to read different formats like .xsl, .xlsx, etc.

# 4.Reflecting On Building Data Product

As data scientists reflect on the development of the Data Explorer Web Application, the significance of data product development skills become more important. Working in the Data Science field, it is not sufficient to simply extract insights from data, you must be able to develop efficient tools to facilitate this process for others. By shifting their focus from data analysis to the development of data products, data scientists expand their influence by connecting technical analysis with business applications.

The development of data products like the Data Explorer Web App requires a lot of skills. In addition to a fundamental comprehension of programming languages like Python, a deep understanding of data structures and algorithms, and libraries utilized for data manipulation and visualization is an essential qualification. Knowledge of web frameworks, specifically Streamlit, is essential for the development of an interactive user interface. In addition to guaranteeing functionality, a comprehensive understanding of user experience design guarantees that the product is intuitive and easily navigable for users of diverse technical proficiencies.

In addition to these tools, data scientists have the ability to utilize their analytical capabilities to create an extensive collection of data products. These products may consist of interactive interfaces, machine learning models integrated into user-friendly applications, and automated reporting systems. Each of these products can generate value in a variety of contexts and fulfill distinct user requirements, such as streamlining internal operations or offering consumers novel services and insights.

As a result of the present trend of AI development, data products will inevitably become more complex. Complex decision-making processes can be automated by AI, which can also personalize user experiences and offer various predictive insights. Data scientists are confronted with the task and privilege of utilizing these technologies in order to construct inventive resolutions that tackle tangible issues.

The career progression of data scientists can be substantially enhanced through the development of data products. By transcending mere analysis, they exert a direct impact on decision-making and strategy, thereby demonstrating their capacity to convert technical proficiency into concrete business benefits. With the growing integration of data into various spheres of society and business, the capacity of data scientists to construct data products will emerge as a critical differentiating factor, establishing them as frontrunners in the domain of data-driven innovation.

short dash

# 5.Collaboration

## 5.1. Individual Contributions

***Student A - Paiynthalir Samyappan Nallamuthu (Pine)***

* Coding Tasks
  + - tab\_df/display.py
    - tab\_df/logics.py
    - test/
    - Streamlit\_app.py
* Testing Tasks
  + - Testing the app for my code
      * Using different CSV files
      * With missing and duplicate data
      * Other file extensions
      * Big, small and empty files
      * With date, text, numeric and boolean columns
      * With text and numeric columns alone
    - Testing the team’s code and giving feedbacks
* Documentation Tasks
  + - Readme.md
    - Requirements.txt
    - Draft of entire document
    - Web App Presentation
      * Functionality section - ‘Dataframe tab’
      * Instructions to Launch the application
    - Collaboration
      * Individual contributions
      * Group dynamic
      * Ways of working together
      * Issues faced
    - Reference
    - Appendix
      * Work Log
      * VS Code GIT Integration
      * Application screens
* Project coordination Tasks
  + - Git project setup and collaboration
    - Teams group updates and tracking progress

***Student B - Ngoc Quang Vinh Pham***

* Coding Tasks
  + - tab\_num/display.py
    - tab\_num/logics.py
* Testing Tasks
  + - Testing the app
      * Using different CSV files with different types of numerical values
      * Other file extensions
      * Continuous values
      * Discrete values
      * User interface rendering of numeric data with accuracy.
      * Accurate calculation and display of statistical data.
      * Effective operation of interactive components, including dropdown menus and sliders, if visible.
      * Performance and responsiveness, particularly when dealing with large datasets.
      * Management of extreme cases in numeric columns, including missing values, outliers, and non-numeric values.
  + Documentation Tasks
    - Reflecting On Building Data Product
    - Collaboration
      * Individual contributions
      * Issues faced
    - Conclusion
    - References
    - Appendix
      * Work Log
      * Application Screens

***Student C- Akanksha Kamath***

* Coding Tasks
  + - tab\_text/display.py
    - tab\_text/logics.py
* Testing Tasks
  + - Testing if the UI is met as per the requirement
    - Testing the functionality of the Text Tab
    - Testing the Web page functionality
    - Testing overall flow of the features.
* Documentation Tasks
  + - Executive Summary
      * Overview of the Project
      * Problem Statement
      * Achieved Outcomes and Results
    - Introduction
      * Key objectives or Goals
      * Stakeholders
      * Addressing the requirements
    - Web App Presentation
      * Functionality
      * Users and Use Cases

***Student D - Cong Tuan Minh Le***

* Coding Tasks
  + - tab\_date/display.py
    - tab\_date/logics.py
  + Documentation Tasks
    - Ideation for Future Works

Special mention to “Akanksha Kamath” for her eagerness to get into quick calls to discuss issues and also her idea of publishing the app to the public domain to enable other non technical stakeholders to access the application. Kudos to her!

## 5.2. Group Dynamic

* Team members were given equal opportunities to contribute and all of them contributed well.
* Planning well ahead of time and being proactive are the areas of improvement for group 25.
* Team members are willing to help each other when in need. The only challenge was to find a commonly suitable time for all of us to come together and work as a group.

## 5.3. Ways of Working Together

Microsoft teams and outlook

* Initial communication was through email and then we used Microsoft Teams group chat and calls/meetings option to communicate to each other.
* We did not have any planned meetings since the team had different workloads and timelines. Chat was a better way to communicate. That way each member can pick up the communication and contribute whenever they can find some time for this project.

VS Code and GIT

* A private shared repository was created and all the team members were added as collaborators.
* We worked on different code branches in the private shared repository and once code complete of their part of the tab, the code was merged into the main branch.
* VS Code was an easy to use tool to code, test and integrate to git.

## 5.4. Issues Faced

* Found it difficult to get responses from the team members and feel like a team to start with. In future, it will be better if one day is decided and all the team members can meet in person before or after one of the teaching sessions, to get more collaboration and also to network with peers.
* Git access issues. Initial invite was sent to all team members using their UTS emails. Some invites expired and one didn't work. We resolved this issue by sending out invites to emails other than uts email id and also resending invites again at different times.
* Till noon of the final submission day not all the code is merged into the main branch. Tried to share the git commands list to be run from the terminal in the teams chat. And the team members helped each other to debug and finally all the code is in. Lesson learnt is not to have the code merged on the last day, push to git as and when the coding is being done. This can help test if each other’s code is working as expected and also working together as a great app.
* Next time we need to implement a process of code reviews before commit to the main branch, to avoid issues during merging. And also on the very first day, we all should have tested and learnt how to use git using branches, pull, push, commit etc., This could have avoided last minute tensions
* Using M1 macbook for the code and get the "mach-o file, but is an incompatible architecture (have 'x86\_64', need 'arm64e')" error while running Streamlit indicates that the Streamlit installation or one of its dependencies is compiled for the x86\_64 architecture (used in Intel chips), while your M1 Mac uses the arm64e architecture. This architectural mismatch causes the incompatibility.
* There were gaps in communication among team members, leading to delays and confusion, which led to various technical issues, such as compatibility and performance, that arose during development. Thus, effective use of collaborative tools and consistent communication are essential for coordinating team endeavors and sustaining project progress. Moreover, the efficacy of individual tasks in dealing with technical challenges can certainly be passed when compared to collaborative problem-solving.

# short dash

# 

# 6.Conclusion

The successful completion of the Data Explorer Web Application Project signifies the integration of technical achievement and collaborative effort. Despite the fact that we have encountered some obstacles in the collaboration process, the endeavor has effectively manifested a tool that simplifies and enables anyone to conduct exploratory data analysis, thereby fulfilling the foreseen objectives and satisfying the varied requirements of its stakeholders. The accomplishment highlights the substantial progress achieved by every member of Group 25 in bridging the gap between user accessibility and complex data analysis.

By successfully combining a user-friendly interface with robust and scalable data processing capabilities, the team hasn’t only achieved the objectives of the project but also established a model for creation of intuitive data science tools in academic environments. The project’s prediction of validation that comprehensive data analysis tools can be developed to accommodate a diverse user base through the application of user-centered design, collaboration, and the integration of the appropriate set of skills. It's solid evidence that the app not only meets all the stakeholder requirements but in many respects exceeds expectations.

With an eye towards the future, the initiative has established a robust groundwork for additional enhancements and developments.

**Future Plans:**

* Explore opportunities to enhance interactivity and visualization features.
* Gather user feedback for iterative improvements and feature additions.
* Ensure compatibility with upcoming Python and Streamlit versions.
* Collaborate with the user community to address evolving data analysis needs.

The development process of the Data Explorer Web App serves as an affirmation to the dynamic nature of data science and the ongoing demand for tools that simplify the comprehension and implementation of data insights for more people.

short dash

# 7.References

* Collaborate on GitHub. https://code.visualstudio.com/docs/sourcecontrol/github. Accessed 14 Oct. 2023.
* St.Slider - Streamlit Docs. https://docs.streamlit.io/. Accessed 14 Oct. 2023.
* St.Radio - Streamlit Docs. https://docs.streamlit.io/. Accessed 14 Oct. 2023.
* St.Table - Streamlit Docs. https://docs.streamlit.io/. Accessed 5 Nov. 2023.
* St.Altair\_chart - Streamlit Docs. https://docs.streamlit.io/. Accessed 5 Nov. 2023.
* St.Dataframe - Streamlit Docs. https://docs.streamlit.io/. Accessed 5 Nov. 2023.
* Check the Versions of Python Packages and Libraries | Note.Nkmk.Me. 16 Aug. 2023, <https://note.nkmk.me/en/python-package-version/>.
* Pandas DataFrame Memory\_usage() Method. https://www.w3schools.com/python/pandas/ref\_df\_memory\_usage.asp#:~:text=The%20memory\_usage()%20method%20returns,memory%20usage%20of%20each%20column. Accessed 5 Nov. 2023.
* Converting Object Column in Pandas Dataframe to Datetime A Data Scientists Guide | Saturn Cloud Blog. 19 June 2023, <https://saturncloud.io/blog/converting-object-column-in-pandas-dataframe-to-datetime-a-data-scientists-guide/>.
* “Basic Writing and Formatting Syntax.” GitHub Docs, https://ghdocs-prod.azurewebsites.net/en/get-started/writing-on-github/getting-started-with-writing-and-formatting-on-github/basic-writing-and-formatting-syntax. Accessed 5 Nov. 2023.
* “Host Your Streamlit App for Free.” Streamlit, 24 Jan. 2023, <https://blog.streamlit.io/host-your-streamlit-app-for-free/>.
* Apply changes from one Git branch to another

<https://www.jetbrains.com/help/webstorm/apply-changes-from-one-branch-to-another.html>

short dash

# 8.Appendix

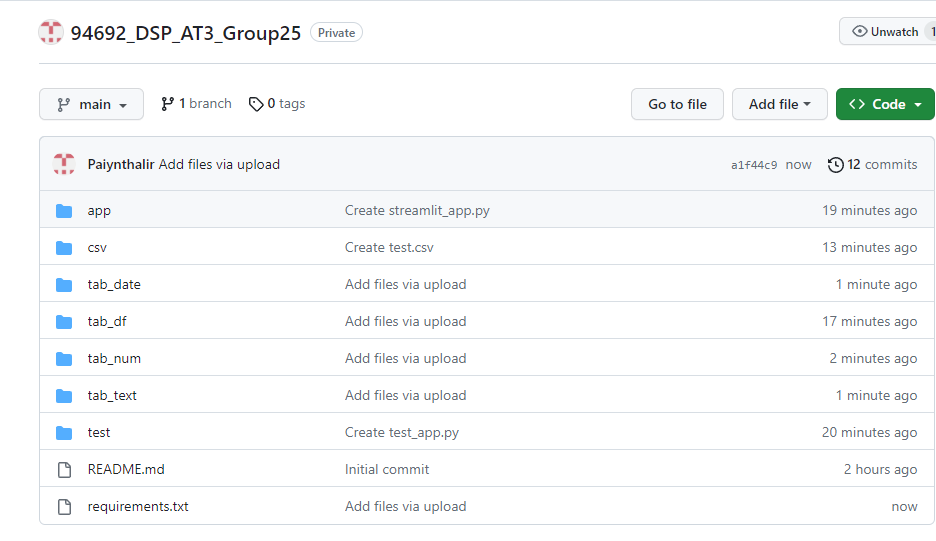
## Work Log

Work logs can be filled in by each team member about their progress and tasks. That way it will be easy to finally put points in the Collaboration topic in this document.

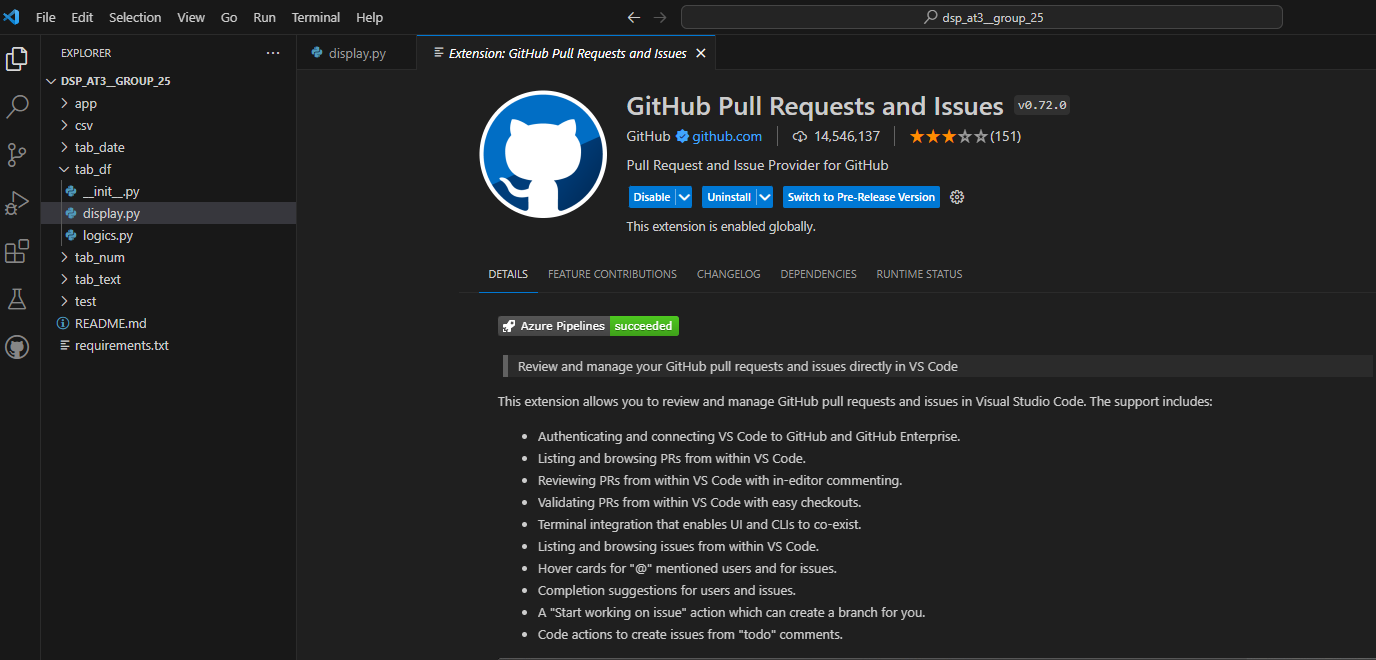
|  |  |
| --- | --- |
| **Date** | **Description** |
| 14 Oct 2023 | **Paiynthalir (Pine)** created teams group for collaboration with the team  And created this draft report and shared it with the team.  Private GIT repository created :<https://github.com/Paiynthalir/94692_DSP_AT3_Group25>  Added team members and teachers as collaborators and uploaded the project template file structure as well. I chose one set of tasks since I was planning to start work today.  ***Student A - Paiynthalir Samyappan Nallamuthu (Pine)***   * tab\_df/display.py * tab\_df/logics.py   Checked in some code under the tab\_df folder. Updated the VS Code GIT Integration appendix section.  **Ngoc Pham** received the team messages, still busy on other projects so chose the ‘tab\_num’ tasks and started working individually.  ***Student B - Ngoc Pham***   * tab\_num/display.py * tab\_num/logics.py |
| 19-Oct-2023 | **Paiynthalir (Pine)** Kept the key stakeholder in loop on the latest status of the team. Updated all the details needed for onboarding into this project in the Teams group chat. |
| 20-Oct-2023 | **Paiynthalir (Pine)** Onboarded ‘Akanksha’ through email and Teams group chat. |
| 4-Nov-2023 | **Paiynthalir(Pine)** Finished the coding part and uploaded in the git |
| 5-Nov-2023 | **Paiynthalir(Pine)** Finished fine tuning the code and the readme.md and requirements.txt and uploaded into git  Tried to each the team members individually to get the status and to know who wants to own which part of the documentation. Not able to reach everyone. Left some private messages along with the team's group message.    **Ngoc Pham** received the private message from Pine and still has some problems with the Altair version and incompatible architecture. |
| 6-Nov-2023 | **Ngoc Pham** pushed the code with some technical problems on the remote repository, trying to get on with the project report while dealing with the problems individually. |
| 7-Nov-2023 | **Paiynthalir(Pine)** Worked with Akanksha through MS teams call to connect to github and had discussion on remaining documentation work and shared the git commands and tool for APA reference format generation. |
| 8-Nov-2023 | **Ngoc Pham** commits the code, which the altair version has been solved and continues on the documentation report. |
| 9-Nov-2023 | **Paiynthalir(Pine)** Worked with Akanksha through MS teams call and had discussion on documentation work and bringing all the code together in Git  **Ngoc Pham** committed the final code with no other errors and fully focused on contributing to the document. |
| 10-Nov-2023 | **Paiynthalir(Pine)** Worked with Akanksha through MS teams call and had discussion on deploying the streamlit app so that it can be accessible to all stakeholders.  Now, I am trying the deployment steps using a streamlit cloud.  Late in the afternoon, I succeeded in deploying the app. Made some changes to the readme and in this document to reflect the deployed URL.  Tested the app on the whole after all the code was integrated and shared some feedback to the team.  Reviewed this document contents and tried to collaborate with feedback with the team. Added some details to fill in the gaps I see..  **Ngoc Pham** merged the code on thomas dev branch to the main branch so that we can launch the project (Get some useful help from Akanksha in order to do it).  Review the documentation report, finalize the conclusion and reflex the objectives and research questions as well as stakeholders’ requirements |

## VS Code GIT integration

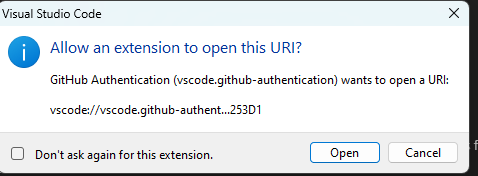
GIT repository is uploaded with the project template



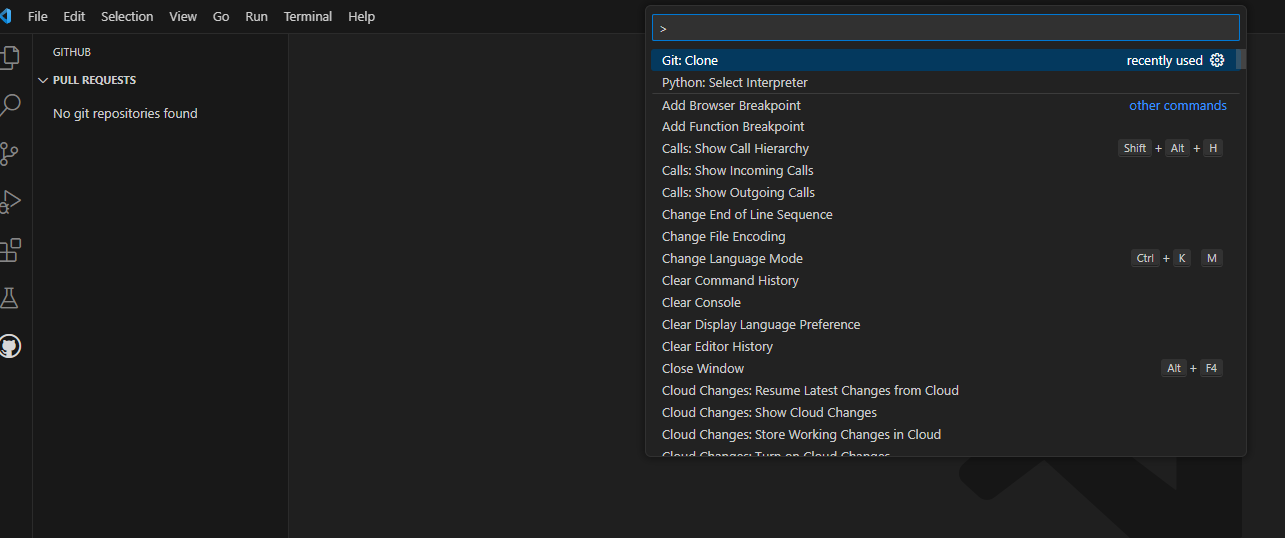
Install the extension, “GITHug Pull Requests and Issues”



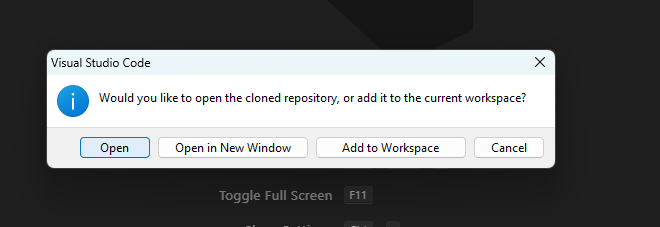
Login to the git account and integrate it with the VS code.



Clone the repository using git:clone command



Open the cloned repository



## Application screens

