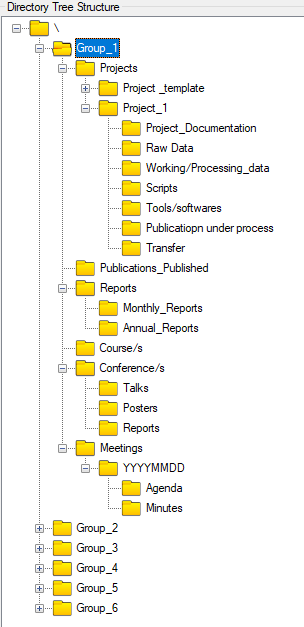
**README**

Welcome to the README file for the Directory Structure on our Data Server. This document serves as a comprehensive guide to understanding the organization and layout of directories on our data server. As data is a critical asset for our organization, structured and organized directory hierarchies are essential for maintaining order and accessibility in a data-intensive environment. Our directory structure is meticulously planned to ensure ease of navigation, clarity in file organization, and optimized storage allocation.

In this documentation, you will find detailed information about the purpose and contents of each directory, along with guidelines for naming conventions. We encourage all users to adhere to the guidelines outlined in this documentation to maintain consistency and integrity within our data management practices.

**Directory structure:**



**/Group\_1 (Group name)**

This top-level directory serves as the primary container for all content associated with Group 1 (Group name). Each group within our organization has its own dedicated directory within this structure, providing a centralized location for storing and accessing group-specific data and resources.

**/Projects**

Contains directories for various projects undertaken by Group members. Leveraging the "Template" application automates the creation of new project directories and their requisite subdirectories, eliminating manual intervention. This makes sure all projects have the same structure.

Steps to create a new project directory using the "Template" application:

* Click on the "Template" application.
* Enter the name of your new project as the directory name.
* The programme will automatically create all of the subdirectories needed for the new project.

**/Project\_1 (Project name)**

This directory serves as a project-specific repository with associated subfolders.

Subdirectories:

* **/Project\_Documentation:** This directory contains all documentation related to Project, including project plans, specifications, requirements documents, and any other relevant documentation.
* **/Raw Data:** The Raw Data directory houses all raw data files collected or generated for Project. No one is permitted to delete files from this directory.
* **/Working (Processing) data:** In this directory, users can store the data that has been processed or is currently being processed as part of Project. This may include cleaned data, intermediate results, or data undergoing analysis.
* **/Scripts:** The Scripts directory contains all code scripts developed for Project. This may include scripts for data processing, analysis, visualization, or any other project-related tasks.
* **/Tools (software’s):** This directory stores setup and license any tools or software applications used in Project.
* **/Publications under process:** Here, you can find any publications or manuscripts that are currently under process for Project. This may include drafts, outlines, or any other materials related to publication efforts.
* **/Transfer:** This directory is used for transferring files related to Project, such as sharing data with collaborators or archiving project files.

**/Publications\_Published**

Directory for storing published publications. You can create subdirectories based on the year to store publications by year (E.g. 2024\_Pub, 2023\_Pub etc).

**/Reports**

Include sub-directories for monthly and annual reports to store various group reports.

**/Course(s)**

This directory will be utilised for storing course(s) materials organised by or for the group. You can create subdirectories for different courses.

**/Conference(s)**

This directory serves as a repository for information and materials related to conferences attended by group members. It includes subdirectories for talks, posters, and reports presented or generated during these conferences.

* **/Talks**: Houses presentations delivered by group members at attended conferences.
* **/Posters**: Stores files of posters presented by group members at attended conferences.
* **/Reports**: Contains reports summarizing the conferences attended by group members.

**/Meetings**

This directory serves as a repository for group meeting data and information. It consists of date-wise subdirectories, each containing further subdirectories for agenda and minutes files.

**Naming Strategy Guidelines:**

It outlines a comprehensive guide to establish consistent and intuitive naming conventions across our organization's files and directories. A well-defined naming strategy is essential for efficient data management, retrieval, and collaboration. By adhering to the guidelines outlined here, users will ensure clarity, consistency, and ease of navigation when accessing files and directories. Additionally, a standardized naming convention promotes better organization, reduces ambiguity, and enhances overall productivity within our organization.

**Descriptive Elements:**

* Select terms that accurately describe the content of the folder/file (E.g.- Experiment conditions, Type of data, Researcher name/initials, Lab name/location, Project or experiment name or acronym, Date or date range of experiment, Experiment number or sample ID etc.).
* Choose terms that are easily understood by all users.
* Incorporate meaningful, commonly used terms within organization.

**Date Format:**

* Use a date format such as YYYYMMDD, YYMMDD, or YYYY Month DD.
* Ensure the date format is consistent across all folders/files.
* Example: “20240122\_MeetingNotes.docx” or “22 January 2024\_MeetingNotes.docx”

**Conciseness:**

* Keep file names concise by avoiding unnecessary words.
* Avoid using small words (like a, the, of, and) and use standard abbreviations.
* Example: ProjectProposal\_Q1\_2024.xlsx

**Avoid Special Characters:**

* Do not use special characters with special meanings to the operating system.
* Example: FinalPresentation\_2024.pptx (Avoid using characters like ! ? @ # $ ~ ^ & % \* ` ; < > , ' " | [ ] ( ))

**Spaces or Separators:**

* Decide if spaces are permitted and choose a consistent separator (hyphens, underscores, or no separation).
* Use separators that align with organization's conventions.
* Example: Financial-Report-Q4\_2023.xlsx or ProjectPlanV2.docx

**Title Case:**

* Capitalize the first letter of all principal words for better readability.
* Example: ProjectX\_MilestoneReport\_V3.docx

**Include Version Numbers:**

* If files will have multiple iterations, include version numbers (e.g., V1, V2, V3, V3.1).
* Clearly indicate the version in the file name.
* Example: ProjectX\_DesignProposal\_V2.1.pdf

**Git Versioning tool:**

Separate Git versioning tool is developed for effective way to monitor and save different versions of data files, reports, scripts, and other important project components. This user-friendly tool makes version management simple, and no need of deep understanding of Git or GitHub. Users may easily save versions of their work with a few clicks and minimum input, and they will be instantly archived on our institute's GitHub account.

Within the "Git\_Versioning" directory, individual directories are allocated for each group, encompassing two distinct Git utilities tailored to streamline version control processes:

1. **Git\_Mod:** Dedicated to preserving and managing modified file versions, facilitating efficient tracking of alterations made within projects.
2. **Git\_Add:** Designed for the seamless integration of newly added files, automatically pushing them to the institute's GitHub repository.

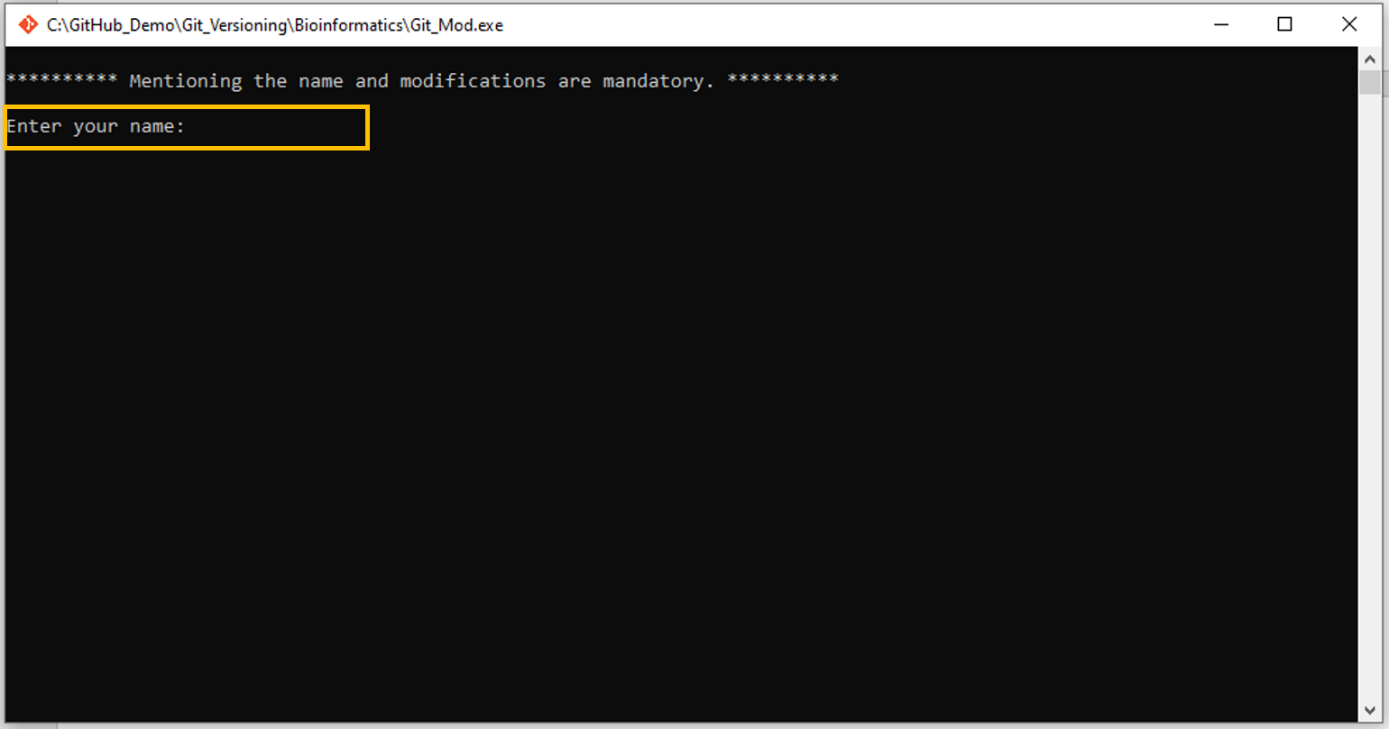
**Steps to use applications:**

**Git\_Mod:**

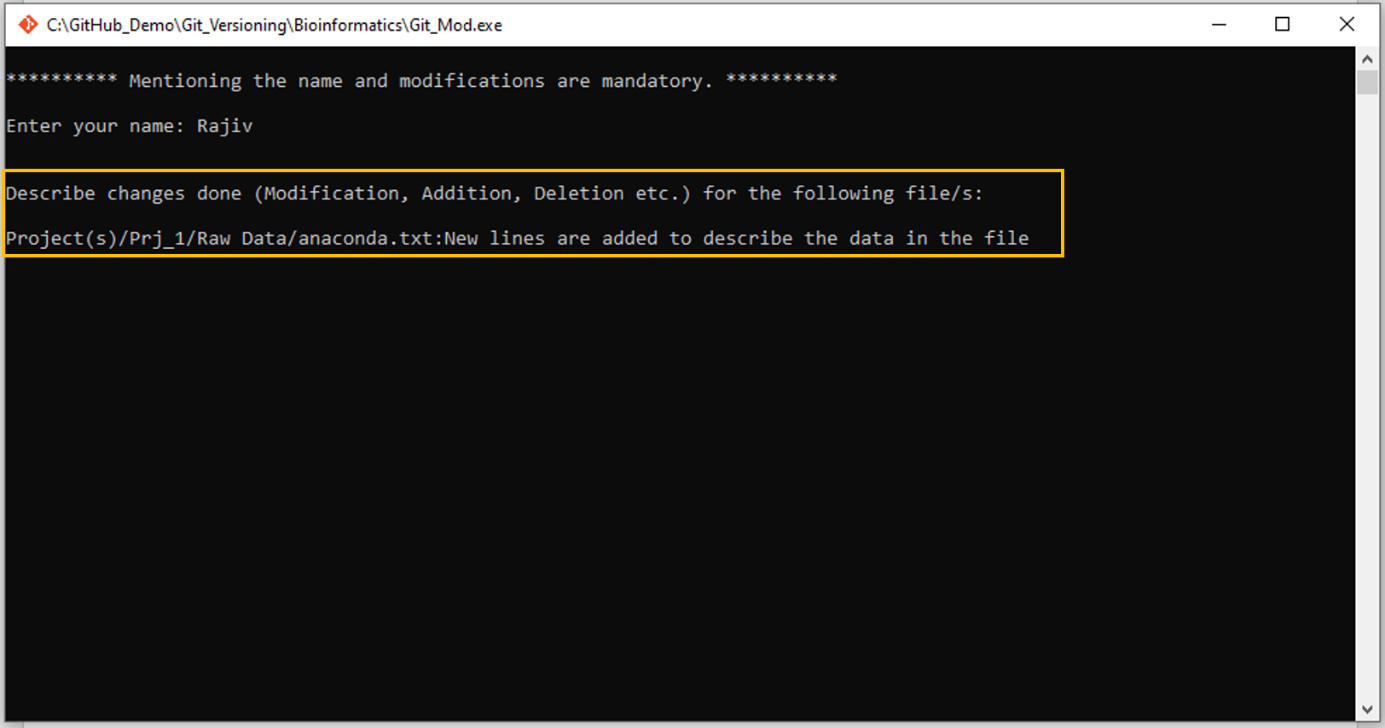
* Navigate to the "Git\_Versioning" directory after making any changes (additions or deletions) to any file(s) (Reports, Presentations, Raw data files, etc.).
* There is a separate directory for each group that contains the application named "Git\_Mod"



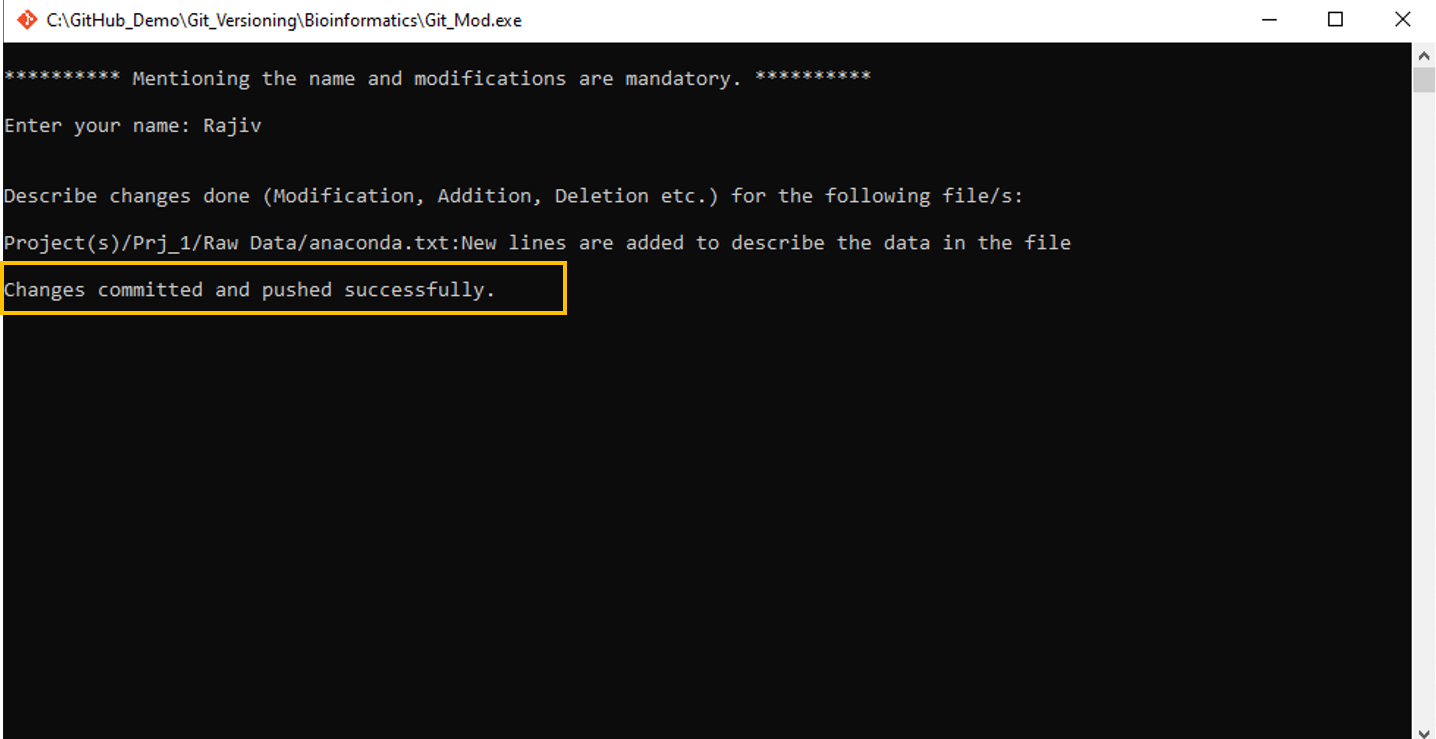
* Click on the application
* The application window will launch and prompt you to enter your name. Mentioning the name is mandatory to keep track who made the modification/s.



* After adding the name, the programme asks for the change that was made in the corresponding file.



* After detailing the change/s, the change/s are committed, and the file/s are automatically pushed into the corresponding repository on our institute's GitHub account.



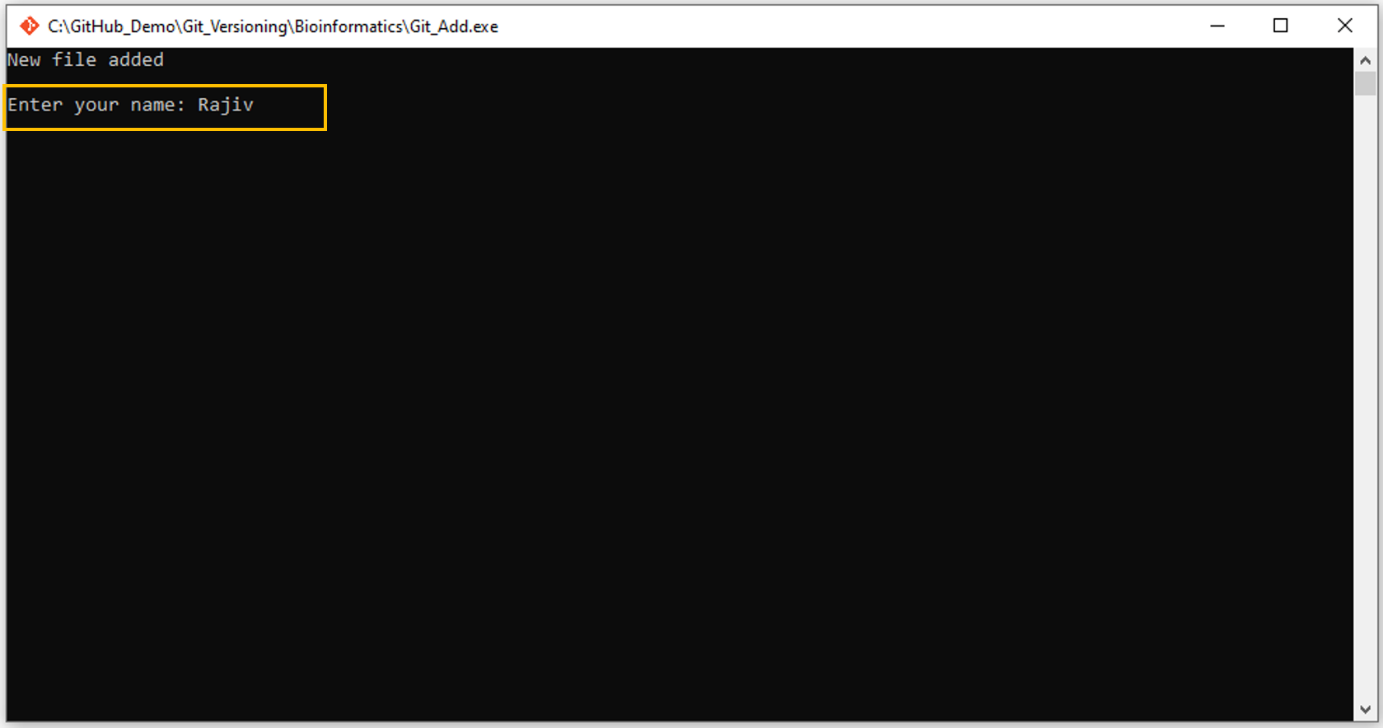
**Git\_Add:**

If you add a new file/s in any directory, you must execute this programme to update the appropriate repository on our institute's GitHub account.

* Navigate to the "Git\_Versioning" directory after adding the new file/s.
* There is a separate directory for each group that contains the application named "Git\_Add"



* Click on the application
* The application window will launch and prompt you to enter your name. Mentioning the name is mandatory to keep track who add the new file/s.



* After entering the name, new file/s are listed and automatically pushed into the corresponding repository on our institute's GitHub account

