GitHub and GitHub Desktop: A Beginner's Tutorial

Introduction to GitHub

GitHub is a platform for version control and collaboration. It allows teams to work on projects together by tracking changes to the codebase over time. With GitHub, multiple people can work on the same files without overwriting each other's work, making it an essential tool for team projects.

Key Features:

- **Version Control**: Keeps a history of all changes.
- **Collaboration**: Allows teams to work on the same project efficiently.
- **Backup**: Ensures your code is safe and accessible.

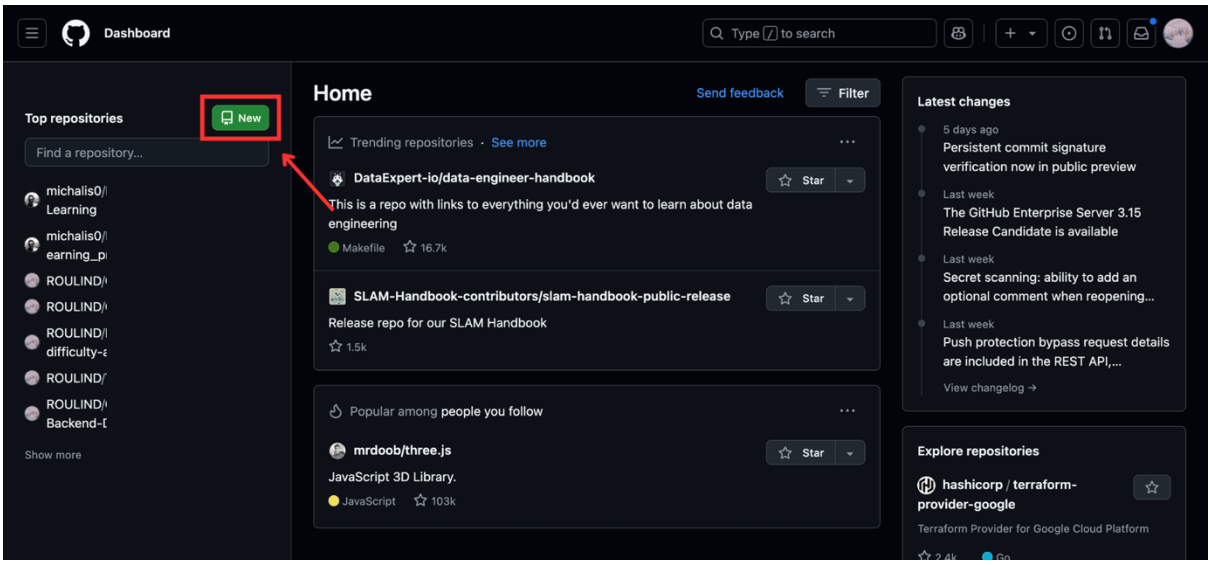
Setup: Creating a Private Repository and Adding Collaborators

Step 1: Create a GitHub Account

1. Go to GitHub.
2. Sign up for an account if you don't have one.

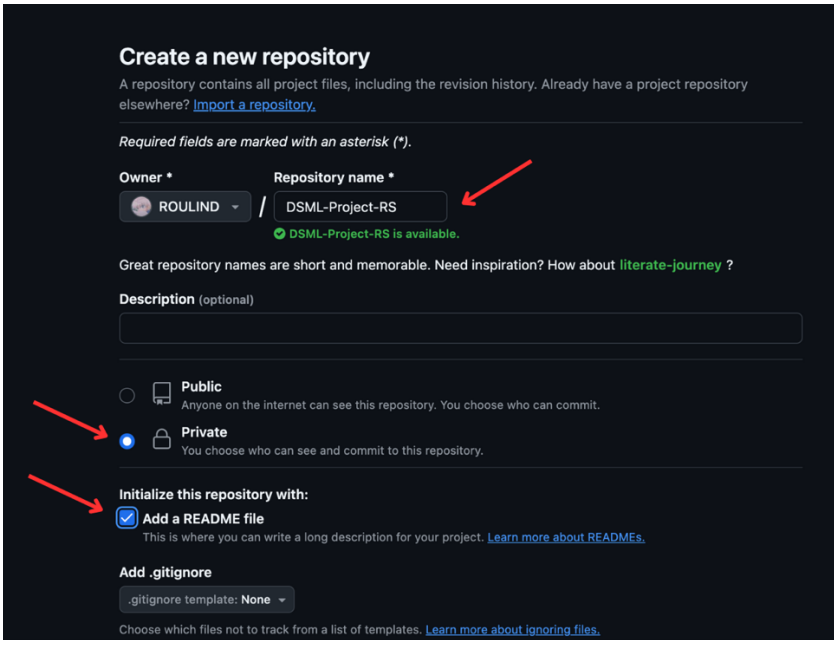
Step 2: Create a Repository

Once logged in, click the **+** button in the top-right corner and select **New repository**.



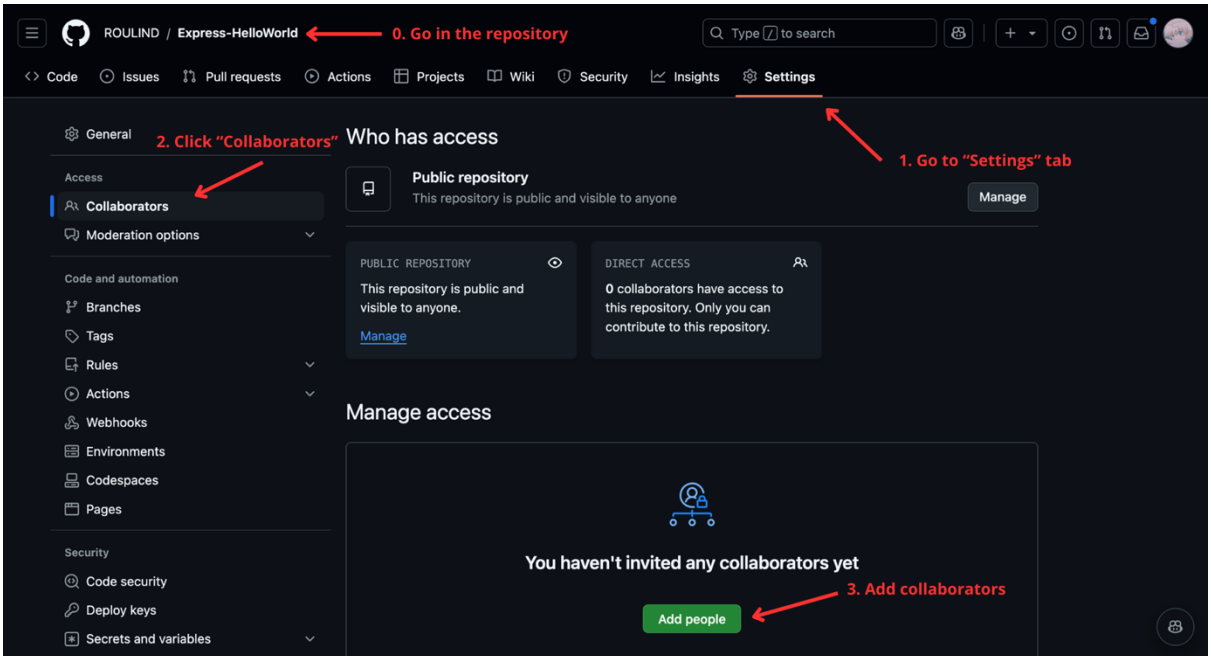
Fill in the repository details:

- **Repository name**: Choose a meaningful name for your project.
- **Description**: Briefly describe your project (optional but recommended).
- **Visibility**: Select **Private** to ensure only collaborators can see it.
- **Initialize repository with a README**: Check this box.



Step 3: Add Collaborators

1. Go to the **Settings** tab of your repository.
2. Click **Collaborators** in the left sidebar.
3. Add the following GitHub usernames:
    - **@roulind**
    - **@ahmadajal**
    - **...and people from your group**
4. Ensure they accept the invitations sent to their accounts.



GitHub Desktop: Introduction and Setup

What is GitHub Desktop?

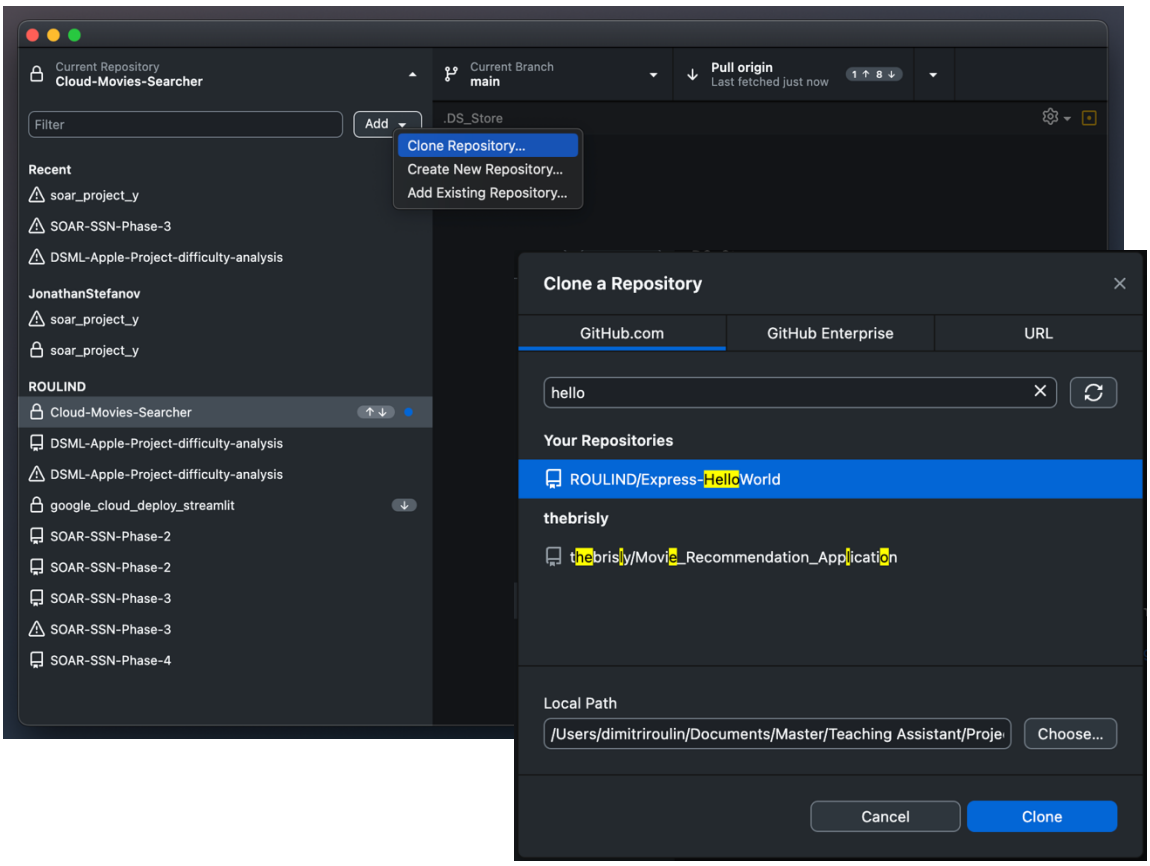
GitHub Desktop is a user-friendly application that simplifies the use of Git and GitHub. It provides a graphical interface to perform Git operations without needing to use the command line.

Installing GitHub Desktop

1. Download the app from GitHub Desktop.
2. Install the application following the on-screen instructions.

Setting up GitHub Desktop

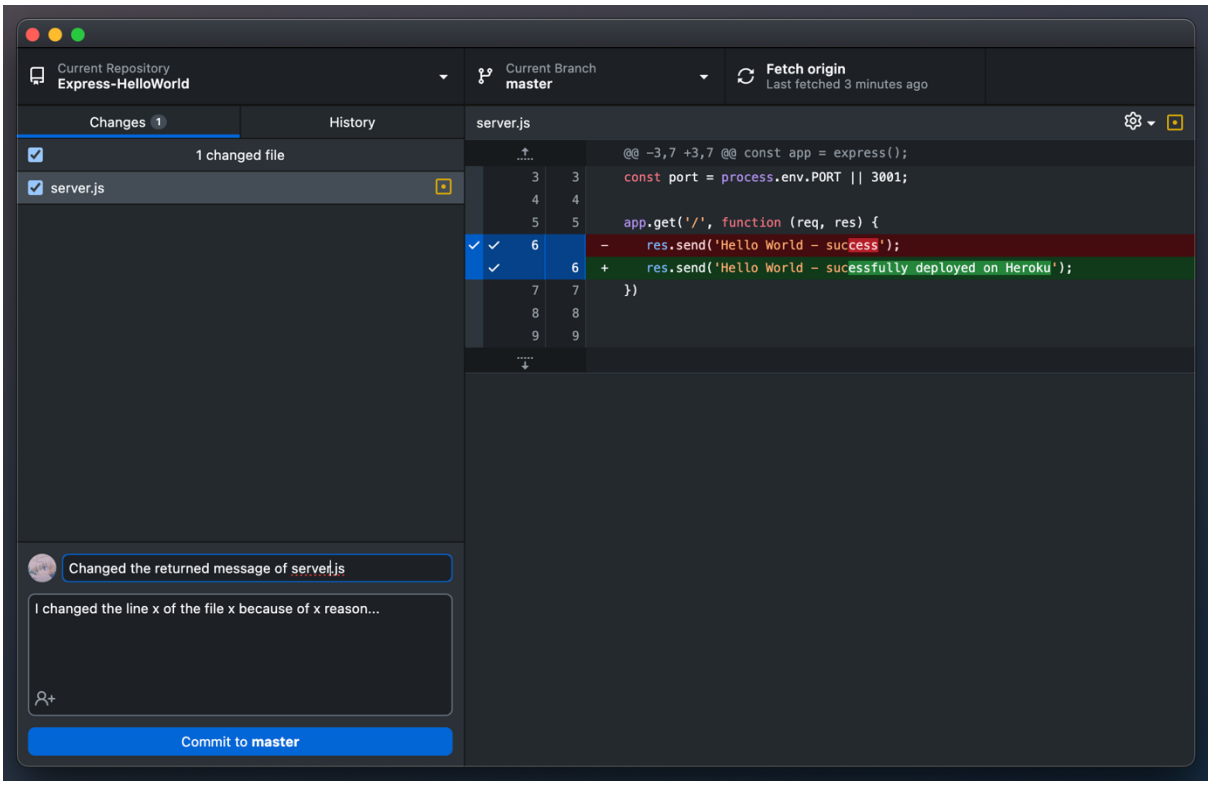
1. Open GitHub Desktop.
2. Log in with your GitHub credentials.
3. Clone your newly created repository:
    - Click **File** > **Clone Repository**.
    - Select the repository you created earlier.
    - Choose a local path to save the repository on your computer.



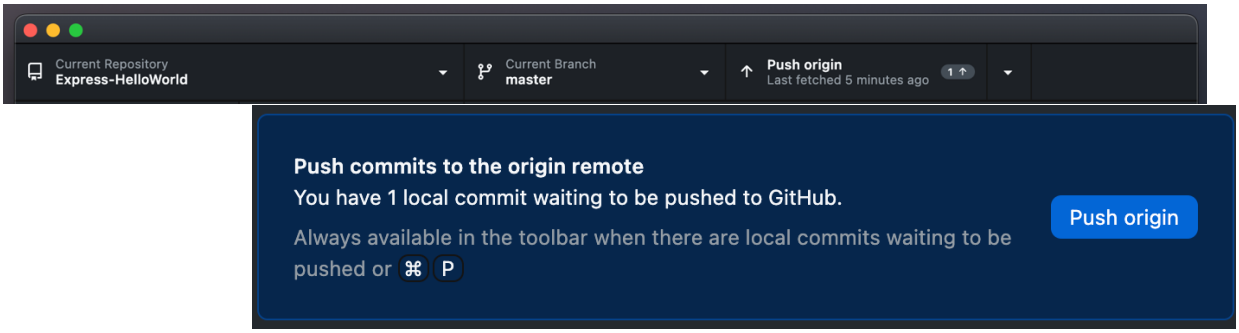
Basic Operations in GitHub Desktop

Commit Changes:

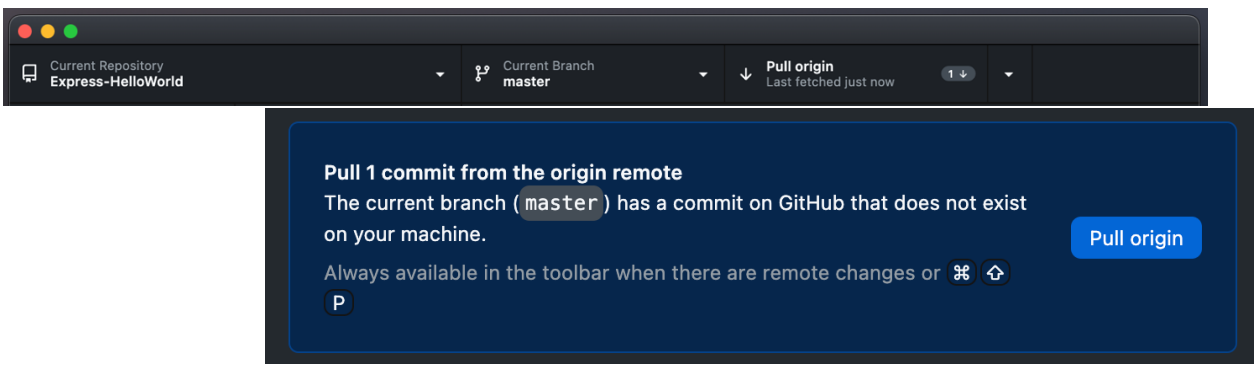
1. Edit files in your project folder.
2. Open GitHub Desktop. You'll see the changes listed under **Changes**.
3. Write a descriptive commit message.
4. Click **Commit to main (or master)**.



Push Changes: After committing, click **Push Origin** to upload changes to GitHub.



Pull Changes: Use the **Fetch Origin** button to update your local copy with changes from GitHub.



Basic Git Commands

While GitHub Desktop is convenient, it's important to understand the basic Git commands for flexibility.

Clone a Repository
Copies a remote repository to your local machine.

```
git clone <https://github.com/your-username/your-repository.git>
```

Add Changes
Adds modified files to the staging area :

```
git add filename
```

To add all changes.

```
git add .
```

Commit Changes
Saves your changes locally with a message describing what was done.

```
git commit -m "Your commit message"
```

Push Changes
Uploads your local commits to the remote repository.

```
git push origin main
```

Pull Changes
Updates your local copy with changes from the remote repository.

```
git pull origin main
```

Check Repository Status
Shows the status of your working directory and staging area.

```
git status
```

Conclusion

Remember:

- Write a clear and concise **README.md** file; it's part of your evaluation!
- Commit your work frequently to avoid losing changes.
- Communicate within your group to avoid conflicts.

Feel free to ask questions or reach out if you encounter any issues. Happy coding! 🚀

Additional resources

https://docs.github.com/fr/get-started/start-your-journey/hello-world

https://medium.com/@sachinsoni600517/complete-tutorial-of-git-and-github-for-basic-to-advanced-1dd34d12b90b