



INTRODUCTION TO GITHUB

WHAT IS GITHUB???

Imagine you and your friends are working on a big project together, like creating a Website. Each of you writes different parts of the code. To make sure everything comes together nicely, you need a way to keep track of all the changes and see who made what. Here GitHub comes into picture.

GitHub is like a collaborative bookshelf for code and projects. It keeps track of changes, helps you work with others, and makes sure your (code) is safe and organized. It's an essential tool for developers and teams to work together and create amazing projects!

Where do we use GITHUB??

GitHub is a place on the internet where programmers/individual store their code and work together on software projects. It's like a virtual workspace where developers can collaborate and keep track of all the changes they make to the code.

It is used by developers, teams, open-source projects, companies, and individuals to manage, collaborate on, and share code, making it an integral part of modern software development workflows.

Some more use-case of GITHUB

- ▶ **Code Storage :** It acts as a storage , where we generally store our codes. It provides a secure and reliable platform to keep code safe and accessible from anywhere.
- ▶ **Learning and Collaboration:** GitHub fosters a strong developer community where individuals can learn from others, contribute to projects they find interesting, and collaborate with developers from around the world.
- ▶ **Portfolio and Showcasing Work:** For individual developers, GitHub can serve as a portfolio to showcase their coding skills and contributions to various projects.
- ▶ **Version Control:** It is primarily used for version control, which means tracking changes in code over time. It allows developers to work on the same project simultaneously, keeping track of who made what changes and when. This ensures that the codebase remains organized and enables easy collaboration.

COMPONENTS OF GITHUB

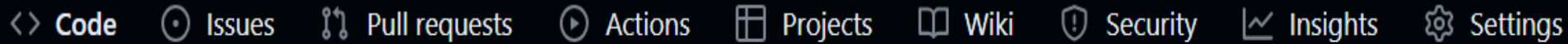
The main component of Github is **REPOSITORIES**.



A repository, often referred to as a "repo," is a central storage space on GitHub where you can keep all the code and related files for a specific project. It acts as a container that holds everything needed to build and maintain the software application or project.

This can be public, allowing anyone to see the code, or private, restricted to a specific group or organization.

Other Components are:-

A horizontal navigation bar with a black background and white text and icons. It contains the following items from left to right: '<> Code', a circle with a dot icon followed by 'Issues', a branch icon followed by 'Pull requests', a play button icon followed by 'Actions', a grid icon followed by 'Projects', a book icon followed by 'Wiki', a shield icon followed by 'Security', a line graph icon followed by 'Insights', and a gear icon followed by 'Settings'.

Code: IT is the core element of a repository that stores the actual source code and related files for a software project.

Issues : It is a feature that helps developers and teams track, manage, and discuss tasks, bugs, feature requests, and other project-related items. It serves as a centralized system for communication and collaboration around the development process.



Pull Request: It is a crucial feature that facilitates collaboration, code review, and the merging of changes from one branch to another within a repository. Pull requests are commonly used when developers want to propose their code changes to be integrated into the main branch of a project.

Project : Project component in GitHub is a powerful tool for task and project management. It helps teams visualize their work, organize tasks, and collaborate effectively, enhancing overall productivity and project success.

Security: It is a set of features and tools designed to help developers and repository owners enhance the security of their code and identify potential vulnerabilities in their projects. The Security component focuses on providing insights and actions to address security concerns in software development.



Wiki: It is a feature that allows developers and project contributors to create and maintain documentation, user guides, and other informative content related to a repository.

Insights: It offer detailed metrics and visualizations that enable better decision-making and understanding of how the project is progressing.

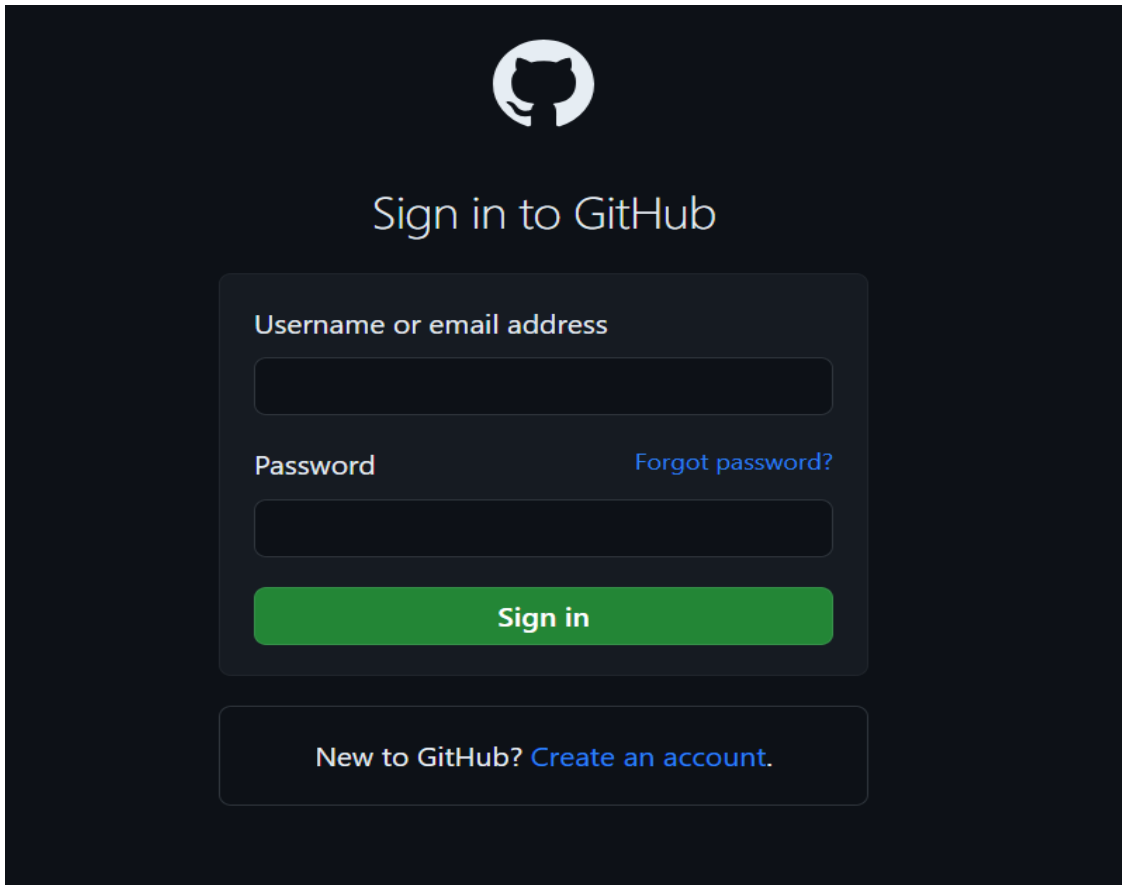
Settings: This allows repository owners and administrators to manage various aspects of the repository, set access permissions, configure integrations, and customize the repository's behavior.




CREATING GITHUB ACCOUNT

1. Login to <https://github.com/>

2. Click on Create Account.

A screenshot of the GitHub sign-in page. At the top center is the GitHub Octocat logo. Below it, the text "Sign in to GitHub" is displayed. The main form area contains two input fields: "Username or email address" and "Password". To the right of the password field is a link that says "Forgot password?". Below these fields is a green "Sign in" button. At the bottom of the form area, there is a link that says "New to GitHub? Create an account.".



Sign in to GitHub

Username or email address

Password [Forgot password?](#)

[Sign in](#)

New to GitHub? [Create an account.](#)



3. Enter Email ID , Password , Username and then click on continue.

```
Welcome to GitHub!
Let's begin the adventure

Enter your email*
✓ agrawalniharika250997@gmail.com

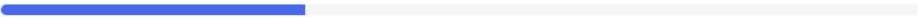
Create a password*
✓ .....

Enter a username*
✓ nihu25

Would you like to receive product updates and announcements via
email?
Type "y" for yes or "n" for no
→ y
```

Continue

4. Select the below option and click on continue.



This will help us guide you to the tools that are best suited for your projects.

How many team members will be working with you?

☒ Just me ☐ 2-5 ☐ 5-10

☐ 10-20 ☐ 20-50 ☐ 50+

Are you a student or teacher?





☐ N/A ☒ Student ☐ Teacher

Continue


5. Select the below option and click on Continue.

What specific features are you interested in using?

Select all that apply so we can point you to the right GitHub plan.

- ☒  Collaborative coding
Codespaces, Pull requests, Notifications, Code review, Code review assignments, Code owners, Draft pull requests, Protected branches, and more.
- ☐  Automation and CI/CD
Actions, Packages, APIs, GitHub Pages, GitHub Marketplace, Webhooks, Hosted runners, Self-hosted runners, Secrets management, and more.
- ☐  Security
Private repos, 2FA, Required reviews, Required status checks, Code scanning, Secret scanning, Dependency graph, Dependabot alerts, and more.
- ☐  Client Apps
GitHub Mobile, GitHub CLI, and GitHub Desktop.

6. Next choose the Free-Trial Option and Click on Continue for Free.



Learn to ship software like a pro.


GitHub gives students free access to the best developer tools so they can learn by doing.

Free

- > Unlimited public/private repositories
- > 2,000 CI/CD minutes/month
Free for public repositories
- > 500MB of Packages storage
Free for public repositories
- > 120 core-hours of Codespaces compute
- > 15GB of Codespaces storage
- > Community support

Get additional student benefits

GitHub Pro

-  **Protect your branches**
Ensure that collaborators on your repository cannot make irrevocable changes to branches.
- > Draft pull requests
- > Pages and Wikis
- > 3,000 CI/CD minutes/month
Free for public repositories
- > 2GB of Packages storage
Free for public repositories
- > 180 core-hours of Codespaces compute
- > 20GB of Codespaces storage
- > Web-based support

GitHub Student Developer Pack

Enrich your college technical community

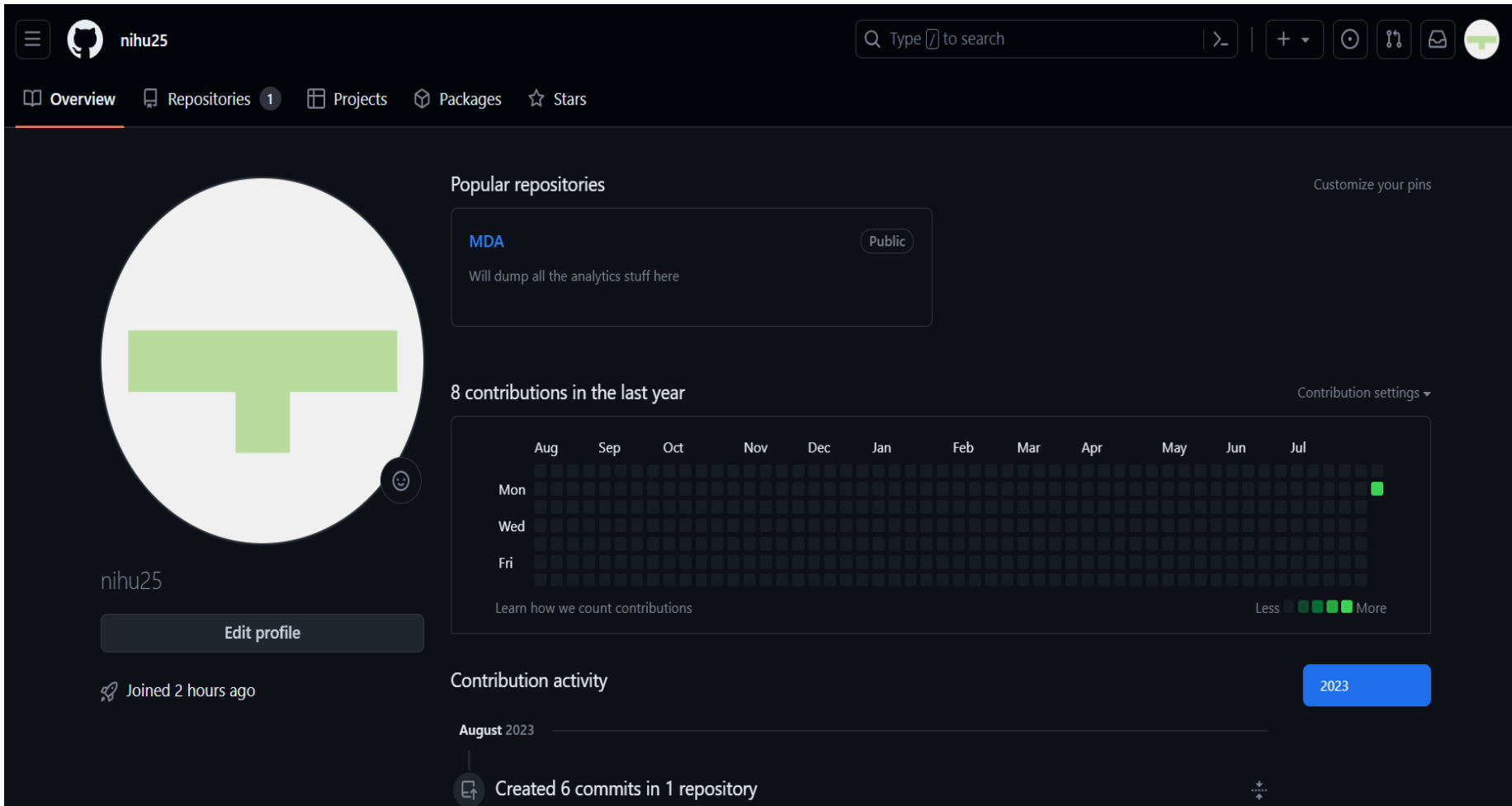
Learn the skills to build diverse tech communities on campus with training, mentorship, and support from GitHub.

Continue for free

Apply for your GitHub student benefits

Skip personalization

7. Once your get account created, you will land into your Github Profile.



The screenshot shows a GitHub profile page for the user **nihu25**. The profile is dark-themed. The user's avatar is a green 'T' on a white background. The profile name is **nihu25**, and it shows they joined 2 hours ago. The 'Overview' tab is selected, showing a list of popular repositories, including **MDA** (Public) with the description 'Will dump all the analytics stuff here'. Below this, it shows '8 contributions in the last year' with a calendar grid. The grid shows contributions for Monday, Wednesday, and Friday in August. The 'Contribution activity' section shows a bar chart for August 2023, indicating 6 commits in 1 repository.

nihu25

Joined 2 hours ago

Popular repositories

MDA Public

Will dump all the analytics stuff here

8 contributions in the last year

Contribution settings

Aug Sep Oct Nov Dec Jan Feb Mar Apr May Jun Jul

Mon

Wed

Fri

Learn how we count contributions

Less More

Contribution activity

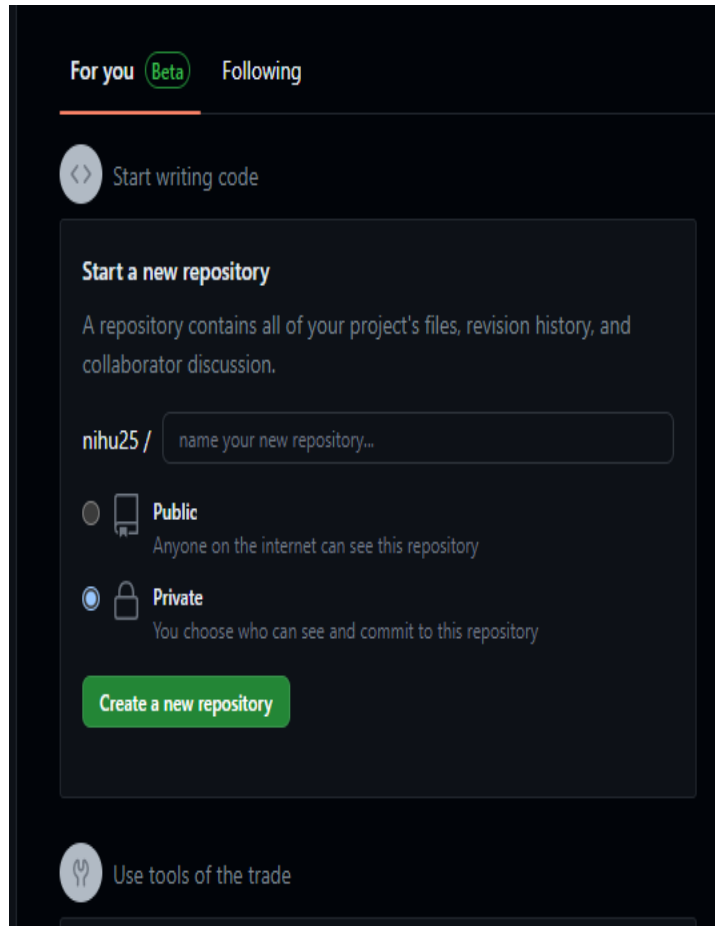
2023

August 2023

Created 6 commits in 1 repository

8. Now will start with creation of Repositories.

Click On Create a new repository (Fig .1). Enter the Repository name and then Click on Create Repository(Fig 2)



This screenshot shows the 'Start a new repository' page on GitHub. At the top, there are tabs for 'For you' (with a 'Beta' badge) and 'Following'. Below these is a 'Start writing code' button with a code icon. The main section is titled 'Start a new repository' and includes a description: 'A repository contains all of your project's files, revision history, and collaborator discussion.' Below this is a text input field for the repository name, preceded by the username 'nihu25 /'. There are two radio button options: 'Public' (selected) and 'Private'. At the bottom is a green 'Create a new repository' button. A footer section says 'Use tools of the trade' with a wrench icon.

For you **Beta** Following

<> Start writing code

Start a new repository

A repository contains all of your project's files, revision history, and collaborator discussion.

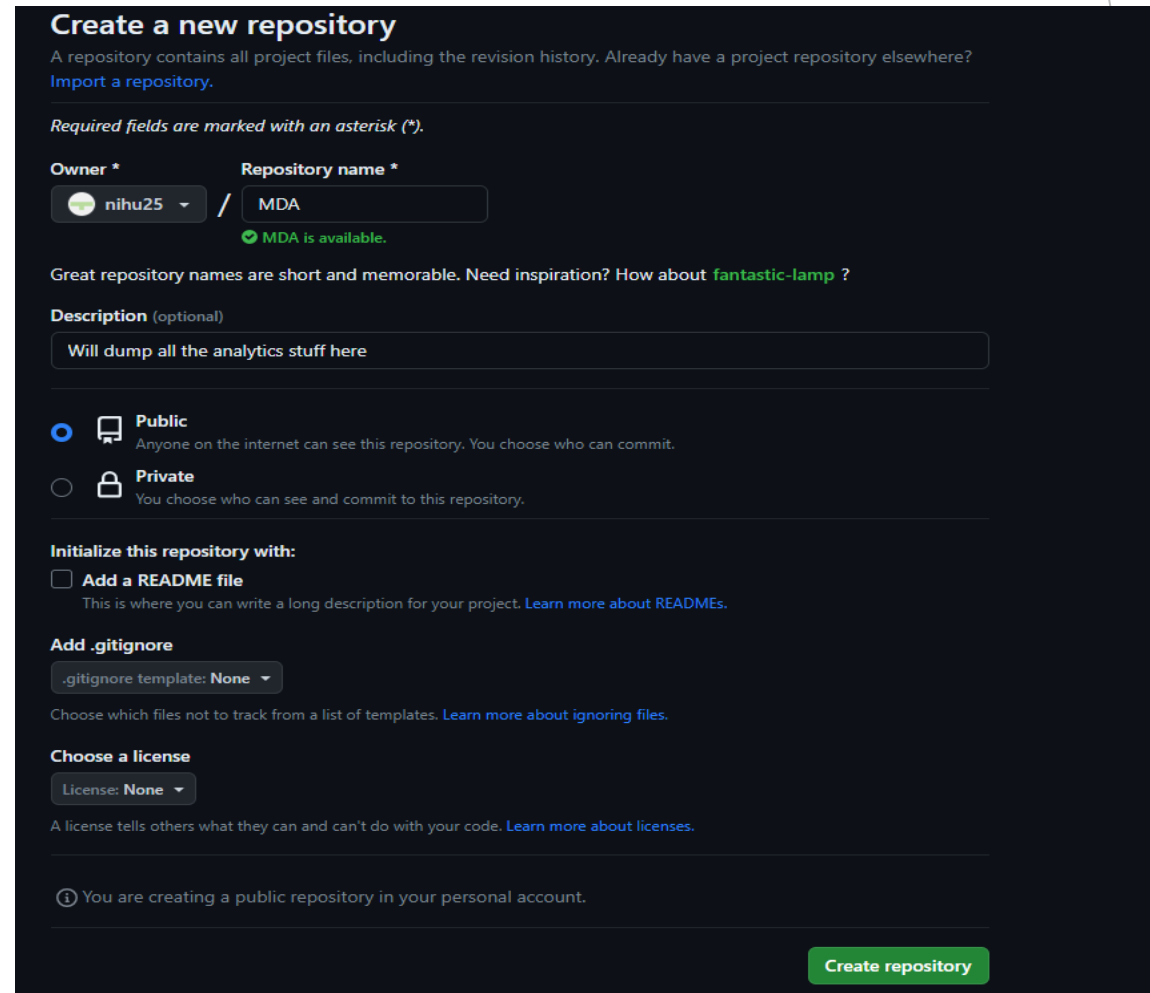
nihu25 /

☒ **Public**
Anyone on the internet can see this repository

☐ **Private**
You choose who can see and commit to this repository

Create a new repository

Use tools of the trade



This screenshot shows the 'Create a new repository' page on GitHub. It includes a title 'Create a new repository' and a subtitle 'A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.' Below this is a note 'Required fields are marked with an asterisk (*)'. The 'Owner' is set to 'nihu25' and the 'Repository name' is 'MDA', with a green checkmark indicating 'MDA is available.' A suggestion for repository names is provided: 'Great repository names are short and memorable. Need inspiration? How about fantastic-lamp?'. The 'Description' field contains 'Will dump all the analytics stuff here'. Under 'Initialize this repository with:', the 'Add a README file' checkbox is checked. The 'Add .gitignore' section has a dropdown set to 'None'. The 'Choose a license' section also has a dropdown set to 'None'. At the bottom, a note states 'You are creating a public repository in your personal account.' and a green 'Create repository' button is present.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner * Repository name *

/

MDA is available.

Great repository names are short and memorable. Need inspiration? How about [fantastic-lamp](#) ?

Description (optional)

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

☒ **Add a README file**
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template:

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

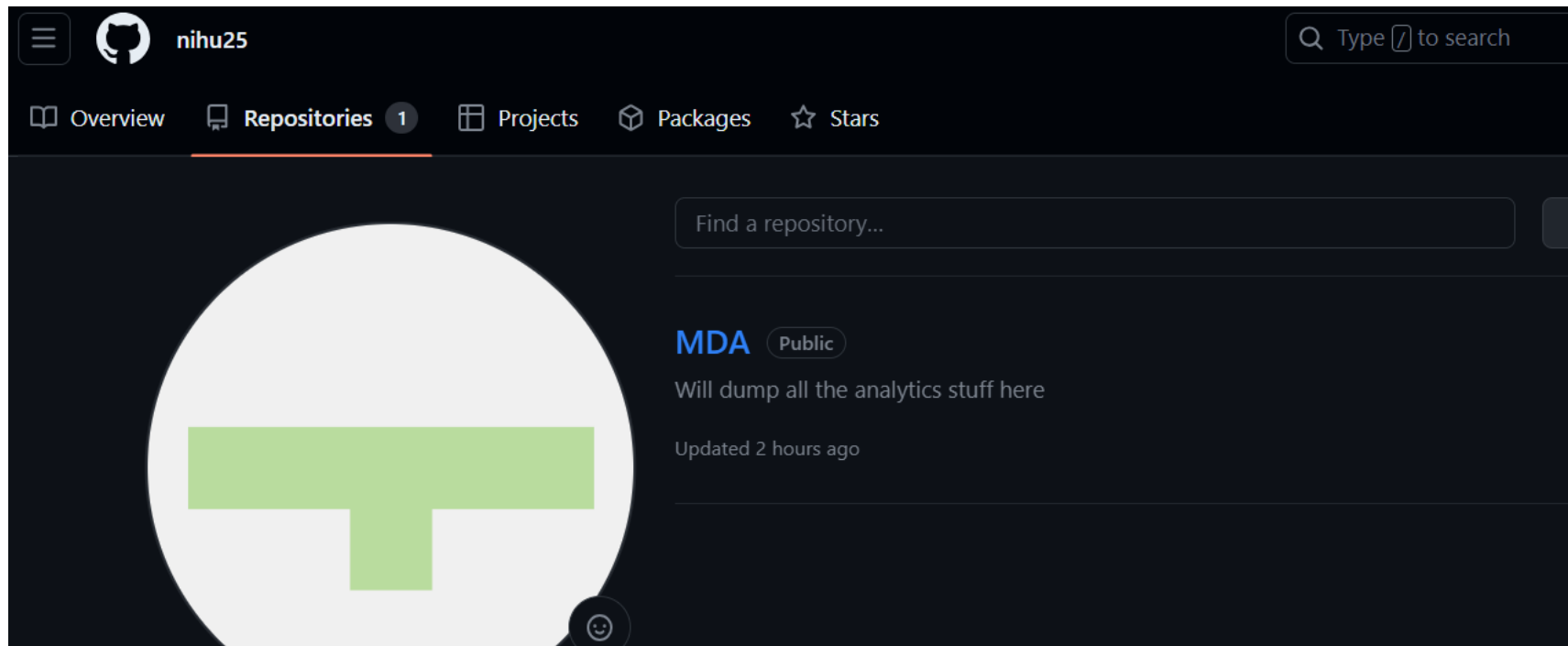
License:

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

You are creating a public repository in your personal account.

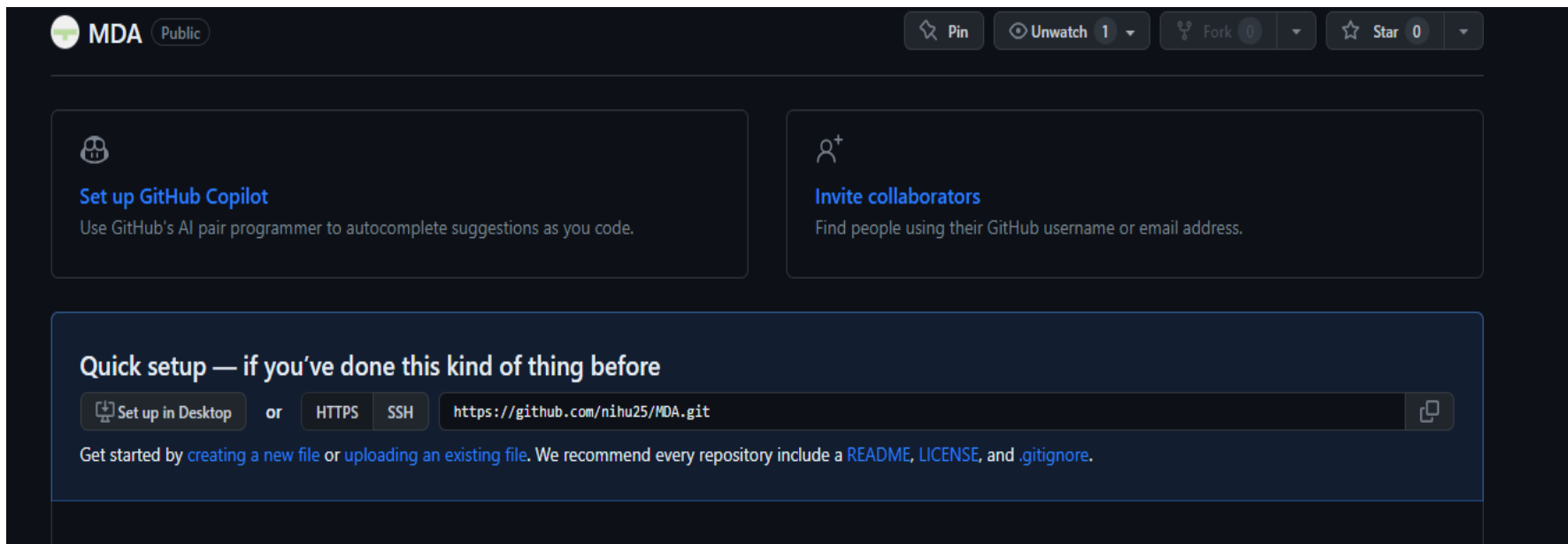
Create repository

9. Once it is created it will display on your profile.



Now the next step is to create sub folders inside that repository.

10. Click on the Repository you created and there you will get a option to create a new file.



11. There is no direct option to create folder inside repository , for creating folder will click on Create new file, will give file name with slash (/). Folder will get created. Here MY-SQL folder.

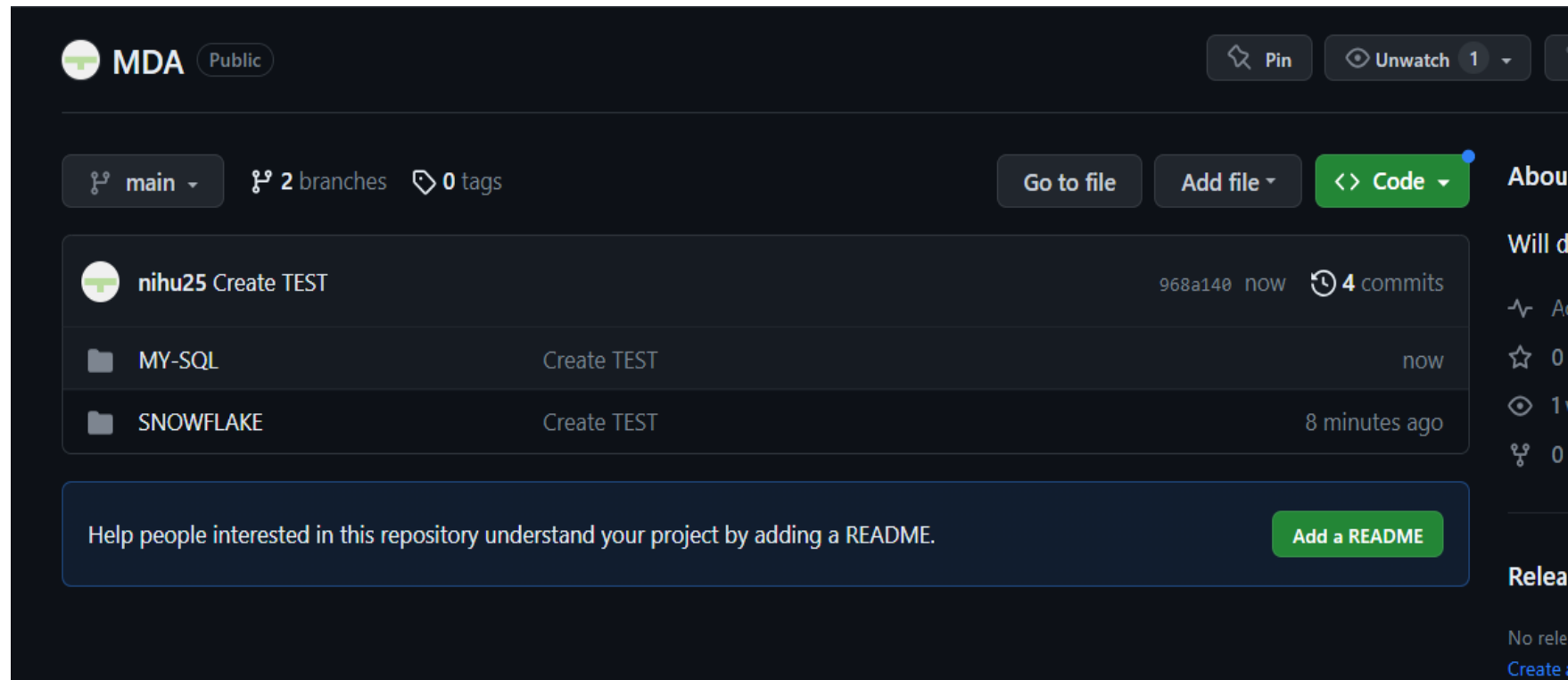
After that enter the file name you want to create inside that folder(Creating TEST File here) and click on COMMIT.



WHY COMMIT ??


Whenever any changes we make in repository (creation of file, uploading file etc) we commit those changes so that it get saved.

12. Two folders created in repository and inside every folder TEST file created.
Similarly you need to create folders.




13. Uploading existing file : Go inside the folder where you want to upload any file. Here ER.pdf. Choose the file and then click on Commit. It will get uploaded.

MDA / MY-SQL



Drag additional files here to add them to your repository
Or [choose your files](#)

ER.pdf ×



Commit changes

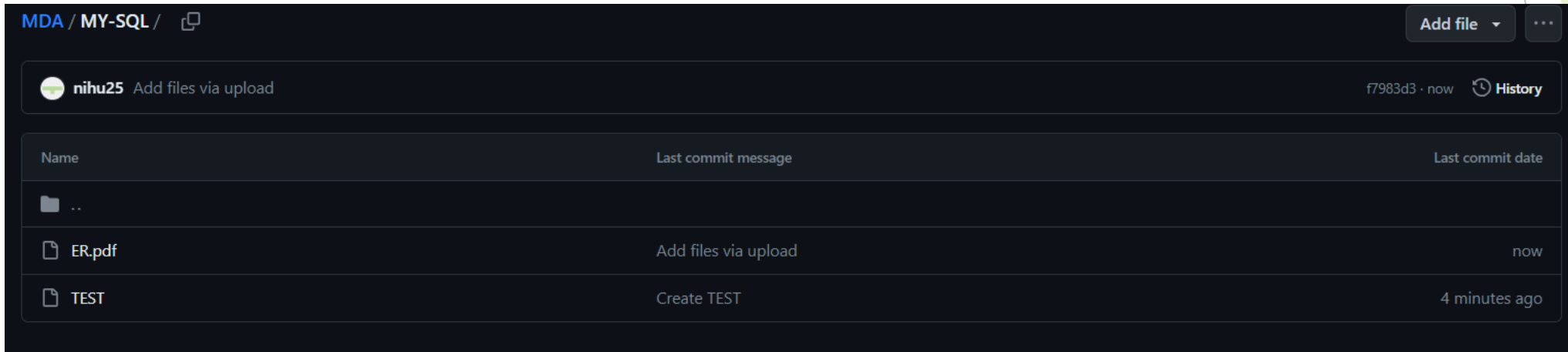
Add files via upload

Add an optional extended description...

☒ Commit directly to the `main` branch.
☐ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

Commit changes Cancel

Here we can see we have MDA repository , we have MY-SQL folder and then we have one TEST file and other we uploaded an existing ER.pdf file.



MDA / MY-SQL /			Add file	...
nihu25 Add files via upload			f7983d3 · now	History
Name	Last commit message	Last commit date		
..				
ER.pdf	Add files via upload	now		
TEST	Create TEST	4 minutes ago		

Similarly, you can create other folders and upload the file.

GIT CLI

**We can also perform all the github operation using command line.
For that we have Git CLI.**

Git CLI (Command Line Interface) is a command-line tool that provides access to the Git version control system. It allows developers to interact with Git repositories and manage their codebase using text commands typed into a terminal or command prompt. The Git CLI is available on various operating systems, including Windows, macOS, and Linux.

Some special Git Cli Command

- ▶ **git clone [repository_url]:** Clone (download) a repository from GitHub to your local machine. Replace [repository_url] with the URL of the repository you want to clone.
- ▶ **git commit :** Create a new commit .
- ▶ **git status:** Check the status of your working directory and see which files are modified, staged, or untracked.
- ▶ **git push:** Push your local commits to the remote repository on GitHub.
- ▶ **git pull:** Fetch changes from the remote repository and merge them into your local branch.
- ▶ **git init:** Initialize a new Git repository in the current directory.



THANK
YOU