

# GitHub & Git Introduction

## Eli Nimy



**Connect with Me:**  
[LinkedIn](#)  
[GitHub](#)

# Outline

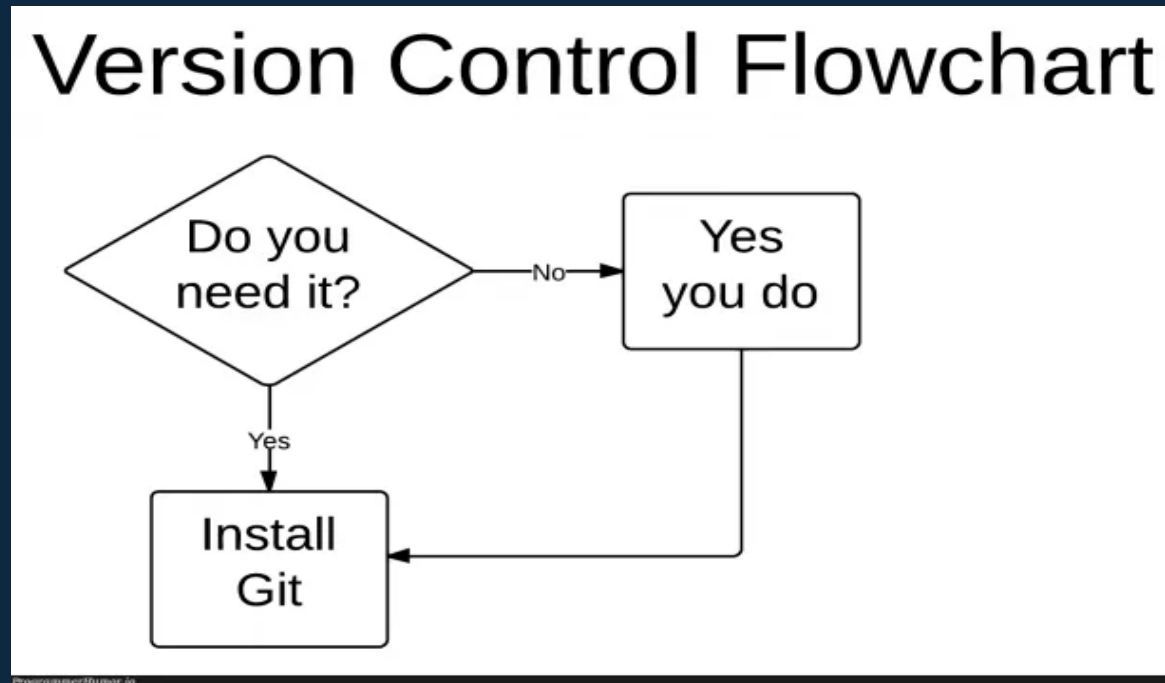
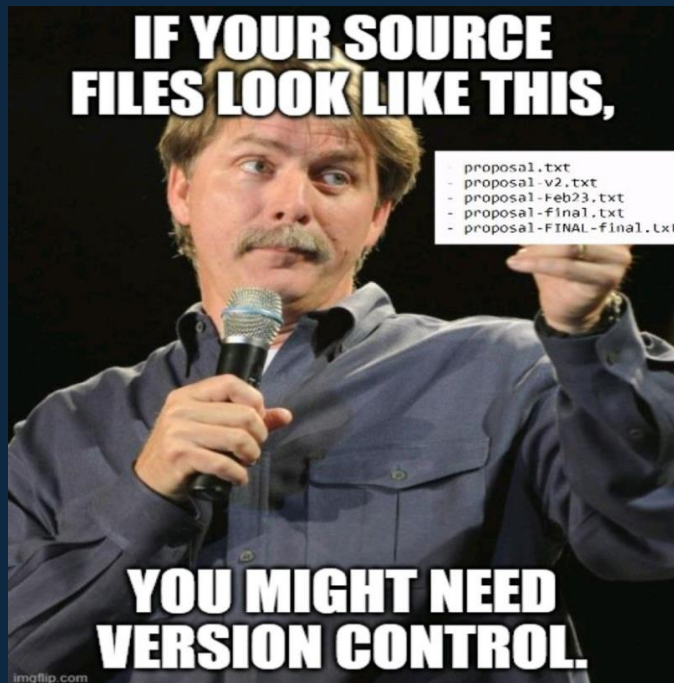
- Questions
- Why version control?
- What is Git?
- What is Git Bash?
- What is GitHub?
- Installing Git and creating a GitHub account
- Working with GitHub – Demo and examples

# Questions

1. What do you like about programming?
2. Where do you store your programming work?
3. How do you collaborate on programming projects?
4. How do you share your programming projects?
5. What do you use to document your coding projects?

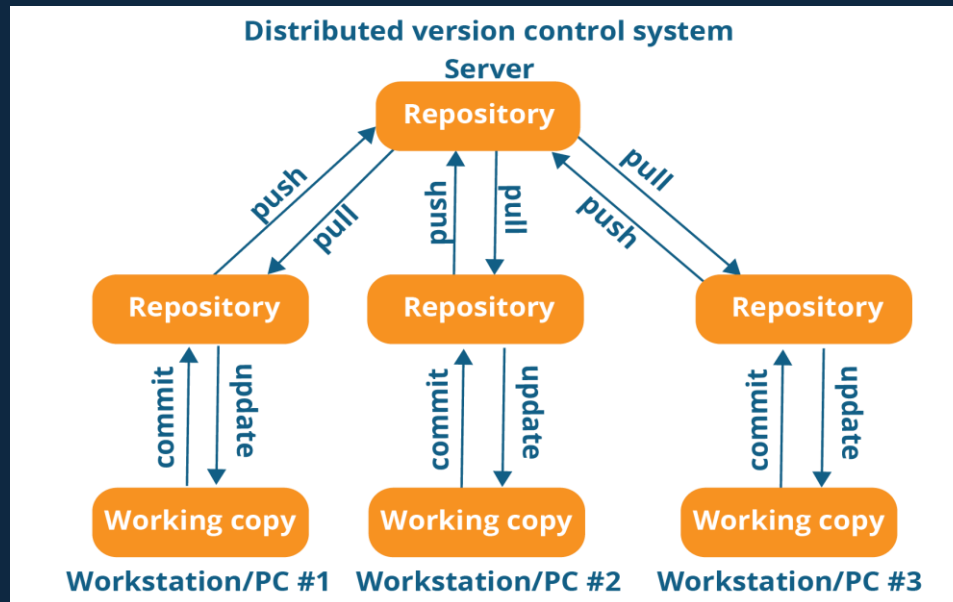
# Version Control

- **Version control:** a system that tracks changes in software development, enables collaboration, and provides a history of modifications



# What is Git?

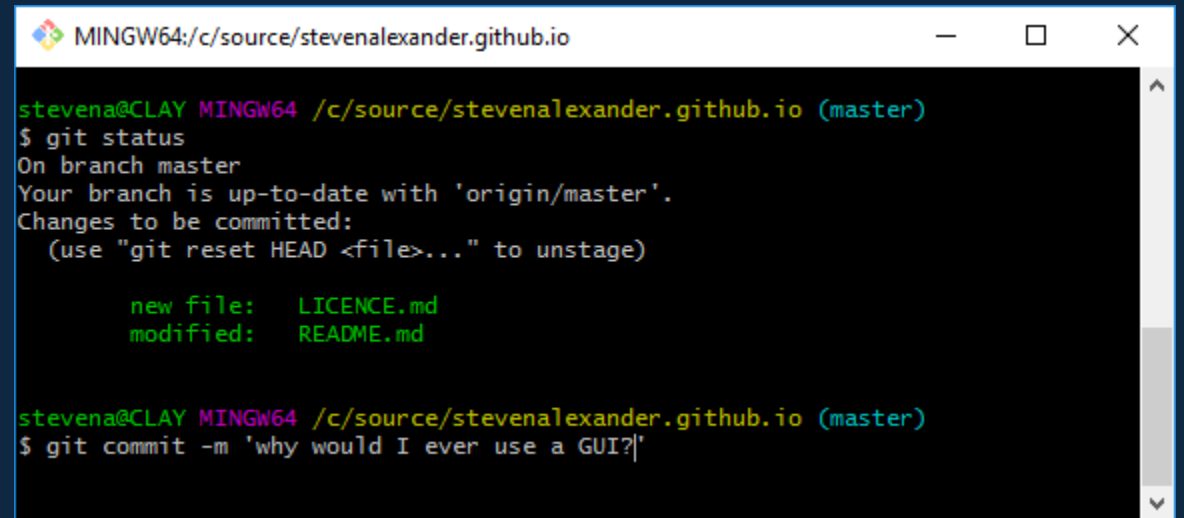
- **Git:** A distributed version control system:
  - Each user has a complete copy of the repository, including its full history.
  - Users can work offline and sync changes with others when connected.



Will demo how this works

# Gitbash – How we interact with Git

- **Git Bash:** A command-line interface for Git on Windows
- **Uses:**
  - Git operations – interacting with Git
  - Command-line operations – Unix-like commands: `cd`, `ls`, `mkdir`, `cd ..`
  - Integration with other tools – used alongside other tools and utilities in the software development ecosystem.

A screenshot of a Git Bash terminal window. The title bar shows the path "MINGW64:/c/source/stevenalexander.github.io". The terminal content shows a user named "stevena" on a "CLAY" machine, in a "MINGW64" environment, at the path "/c/source/stevenalexander.github.io" on the "master" branch. The user runs the command "\$ git status". The output indicates they are on the master branch, which is up-to-date with 'origin/master'. It lists changes to be committed: a new file "LICENCE.md" and a modified file "README.md". The user then runs the command "\$ git commit -m 'why would I ever use a GUI?'".

```
MINGW64:/c/source/stevenalexander.github.io

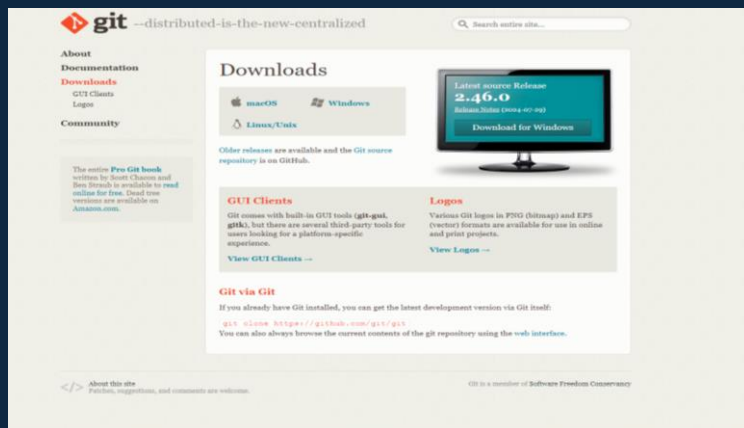
stevena@CLAY MINGW64 /c/source/stevenalexander.github.io (master)
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   LICENCE.md
        modified:  README.md

stevena@CLAY MINGW64 /c/source/stevenalexander.github.io (master)
$ git commit -m 'why would I ever use a GUI?'
```

# Download and Install Git | Video Guide: [link](#)

1. Navigate to <https://git-scm.com/downloads>
2. Select your operating system under the **Downloads** section.
3. For Windows users:
  - Determine your system type (32-bit or 64-bit) by searching for **System Information** on your PC.
  - Download the appropriate **standalone installer**:
    - Most Windows PCs are 64-bit, so click on **64-bit Git for Windows Setup**.
    - If your system is 32-bit, select the 32-bit Git for Windows Setup.



## Installation prompts:

1. Click **Next** to proceed through each step of the installation.
2. Use the default settings provided by the installer by clicking **Next** at each prompt.
3. Finally, click **Install** to begin the installation.
4. Once the installation is complete, click **Finish**.

# What is GitHub

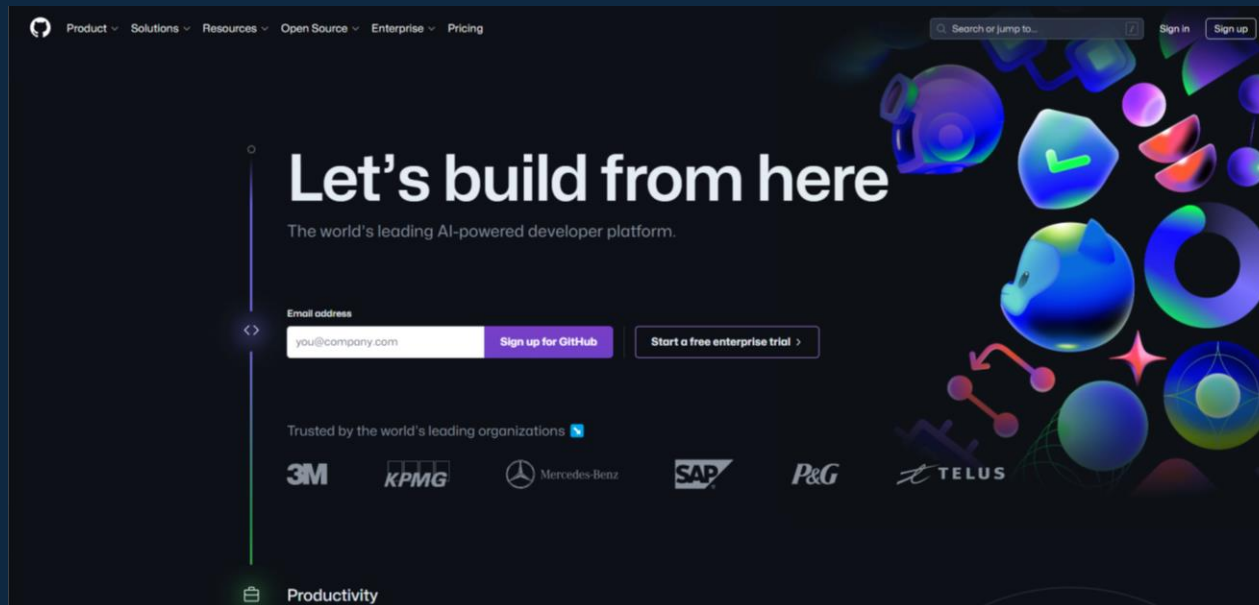
- **GitHub** – A developer platform that allows developers to create, store, manage and share their code.
- **Uses:**
  - Version control
  - Code collaboration
  - Documentation
  - Sharing code and open source
  - Community engagement
  - Portfolio and showcase
  - Education and learning
  - Data hosting





# Create GitHub Account | Video Guide: [link](#)

1. Navigate to <https://github.com/>
2. Click **Sign up**.
3. Follow the prompts to create your personal account



**IMPORTANT:** If you're having problems verifying your email address, there are some troubleshooting steps you can take. For more information, see ["Verifying your email address."](#)

# Demo 1 – Open ended

- Create a public GitHub repository named: random-things
  - Discuss repository naming conventions
  - Add a README, licenses, gitignore
  - Discuss licenses and how to edit the gitignore and README
- Copy all the resource files to the repo
- Edit repository README
  - Have a description section
  - Have git clone section
  - Add image and link to how to generate image
- How to add collaborators
- How to delete a repository or edit repository rights
- Add a local folder or code to GitHub
- How to clone a GitHub repository
- Make changes to files and commit the changes.
- Run python script on Git

# Activity 1

- Create a public repository
- Have a gitignore for all word documents and power points
- Document your repository
- Add any code or files to your repository that you would like to opensource
  - You can use the following files if you do not have any files to add: **index.html**, **styles.css** and **hello\_git.py**
- Share a link to your repository on the chat

# Demo 2

- Create a repository named: debug-python-functions
- Make the repository public
- Create a branch for each debug function
- The branch should be named after the function being debugged
- Write clear commit messages for what was debugged
- Merge all changes to the main branch once done
- Cover:
  - How to create branches

# Activity 2 – Groups of 3 or 5

- Create a repository named: debug-python-functions
- Make the repository public
- Add collaborators
- Each collaborator should debug a function in a separate branch
- The branch should be named after the function being debugged
- Write clear commit messages for what was debugged
- Merge all changes to the main branch once done

# GitHub Use: Documentation

- Source link: [git-and-github-sources](#)

[README](#) [MIT license](#)

## git-and-github-sources

Learning materials on Git and GitHub

### Introduction

- Version control and collaboration are critical in software development.
- GitHub and Git Bash are fundamental tools for version control.

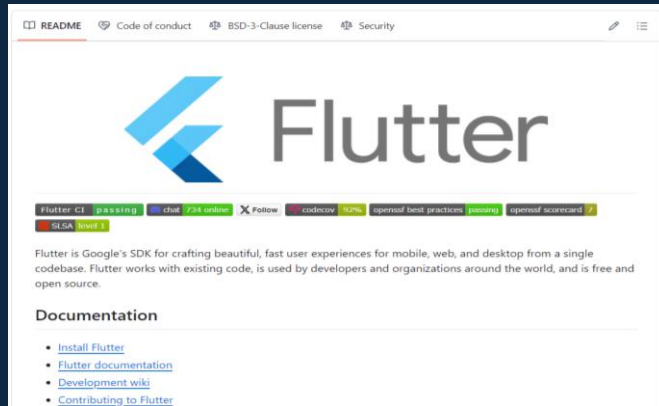
### Version control basics

- **Version control:** Tracks changes in software development, enables collaboration, and provides a history of modifications
- **Challenges**
  - Conflicting code changes
  - Difficulty tracking changes
  - Potential loss of data
- **Solution:** Git and GitHub

## Common Unix-like Commands

Command	Description	How to Use
<code>cd</code>	Change directory	<code>cd &lt;directory&gt;</code>
<code>pwd</code>	Print working directory (show current directory)	<code>pwd</code>
<code>ls</code>	List files and directories in the current directory	<code>ls [options]</code>
<code>mkdir</code>	Create a new directory	<code>mkdir &lt;directory&gt;</code>
<code>rm</code>	Remove files or directories	<code>rm [options] &lt;file/directory&gt;</code>
<code>cp</code>	Copy files or directories	<code>cp [options] &lt;source&gt; &lt;destination&gt;</code>
<code>mv</code>	Move or rename files or directories	<code>mv &lt;source&gt; &lt;destination&gt;</code>
<code>cd ..</code>	Move one directory back	<code>cd ..</code>
<code>cd</code> <code>../..</code>	To move multiple directories back, you can chain the <code>cd ..</code> command as needed	<code>cd ../../..</code>

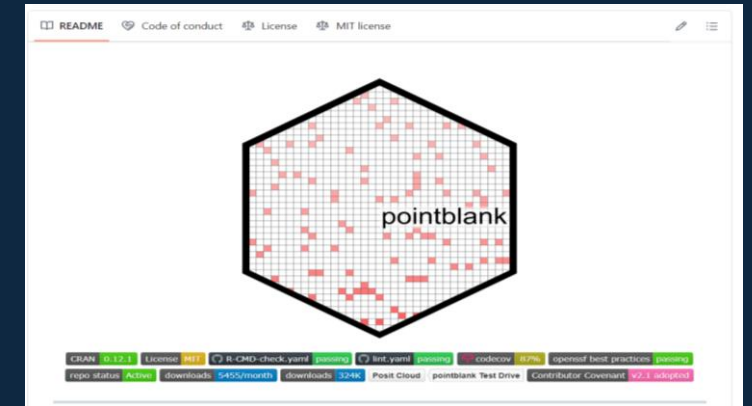
# GitHub Use: Sharing Code and Open Source



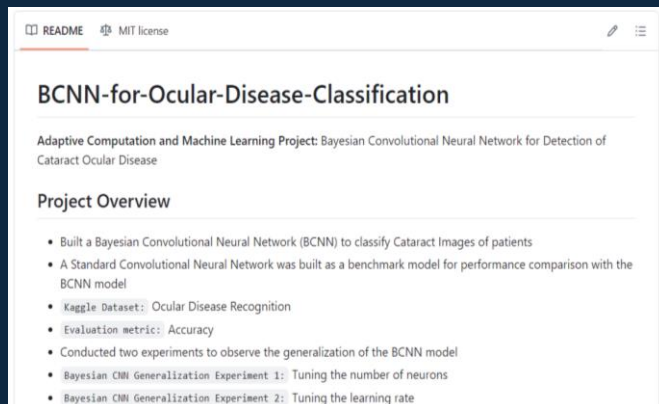
Flutter



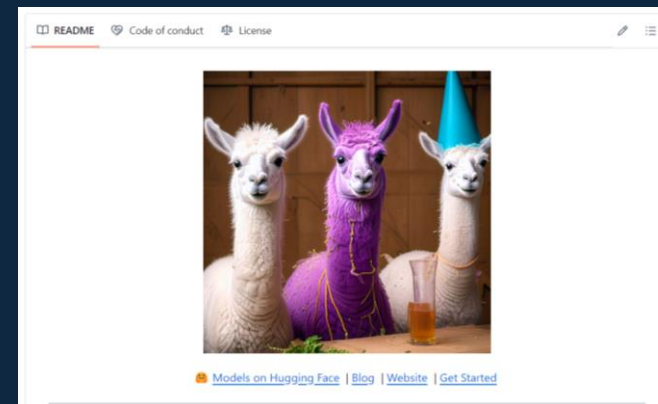
Pandas



Pointblank



BCNN Model



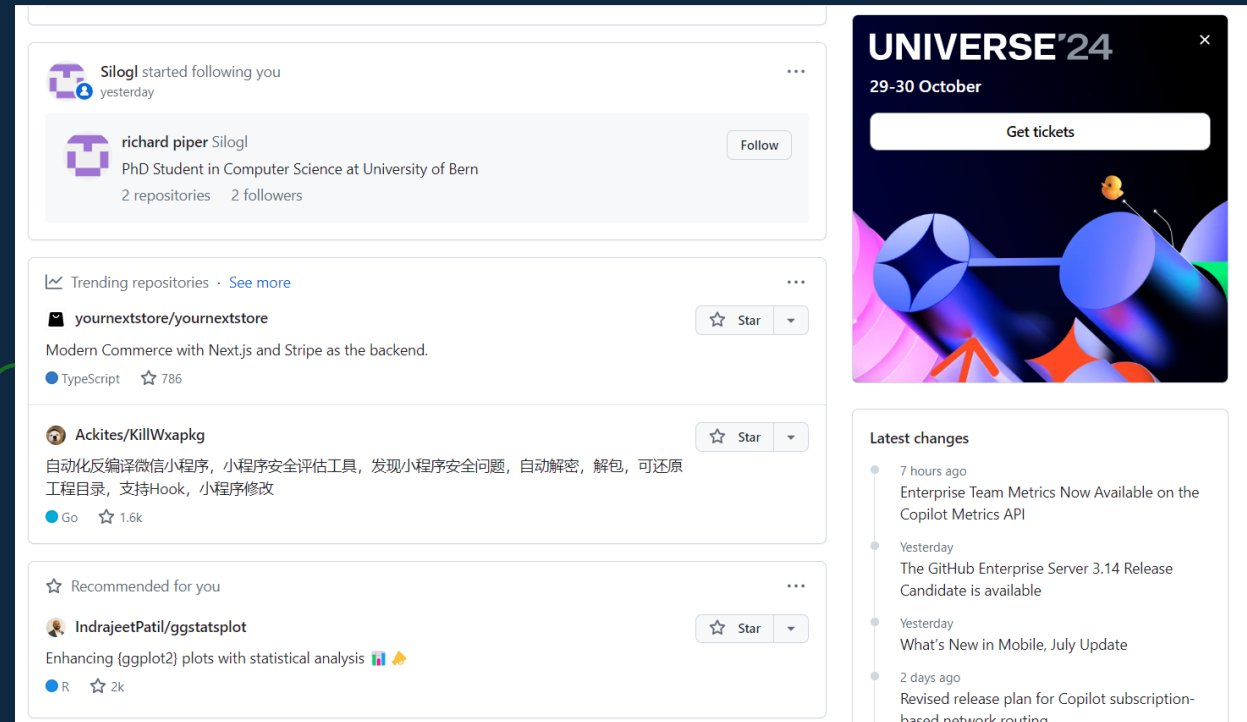
Llma Models



# GitHub Use: Community Engagement

- Star repositories
- Follow people
- Raise issues
- Contribute to discussions
- Etc.

Short discussion example:



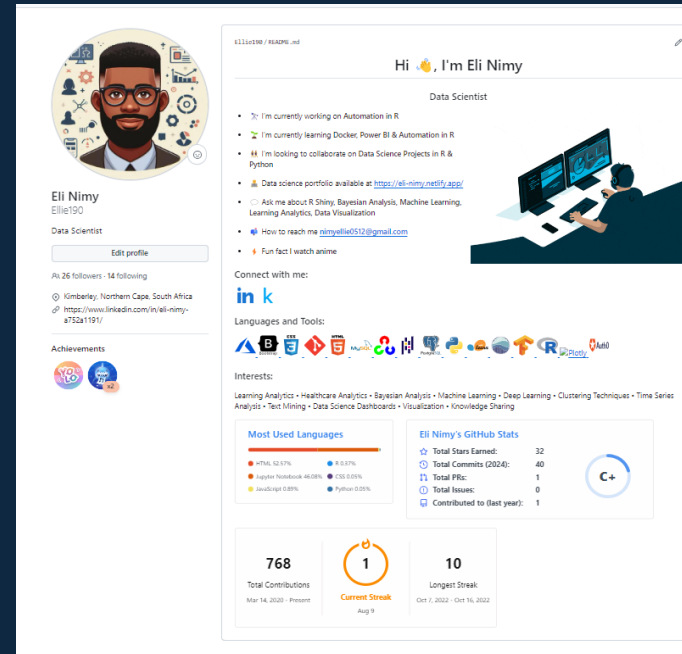
<https://github.com/>



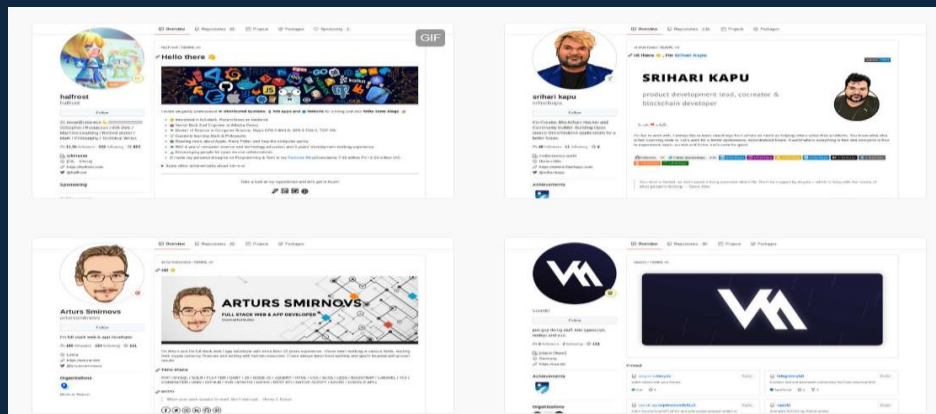
# GitHub Use: Portfolio and Showcase

## Profile creation

1. Click **New repository**
2. Name your repository as your GitHub account name
3. Click on **Public** and **Add a README file**
4. Click **create repository**
5. Click **Edit Read**
6. Navigate to [README generator](#)
7. Edit the README Generator and click **Generate**
8. Copy and paste the generated README to your GitHub profile README
9. Preview and click **commit changes**



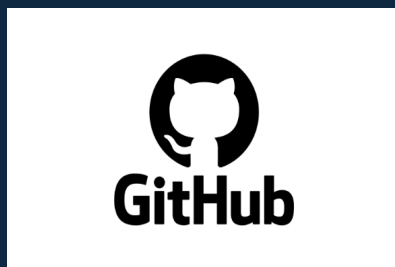
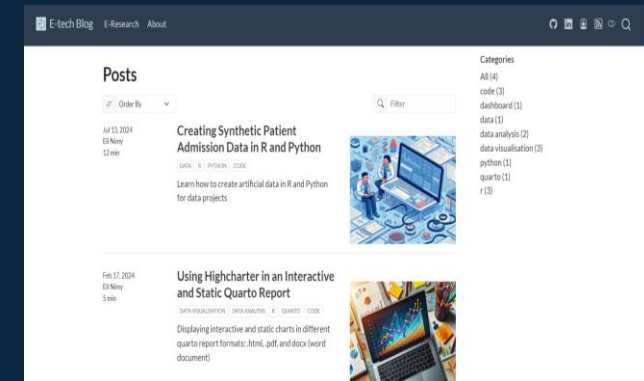
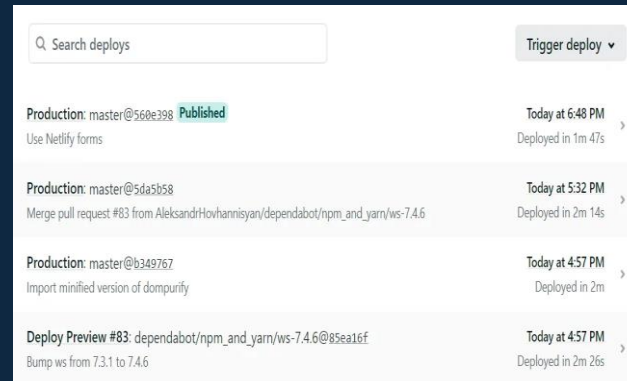
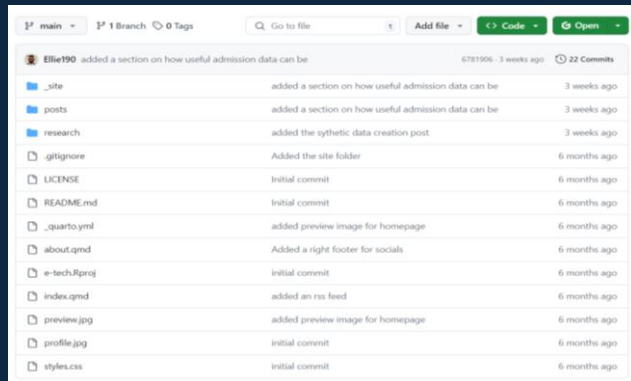
<https://github.com/Ellie190>



Awesome GitHub Profile READMEs

# GitHub Use: Data Hosting

**Data Hosting:** Code, Data, Models, Websites etc.



[elitechblog GitHub Repo](#)



[elitechblog Website](#)



Demo how to update the  
blog website using Git