EXTREME PROGRAMMING(XP)

Extreme Programming (XP) is one of the numerous Agile frameworks applied by IT companies. But its key feature — emphasis on technical aspects of software development — distinguishes XP from the other approaches.

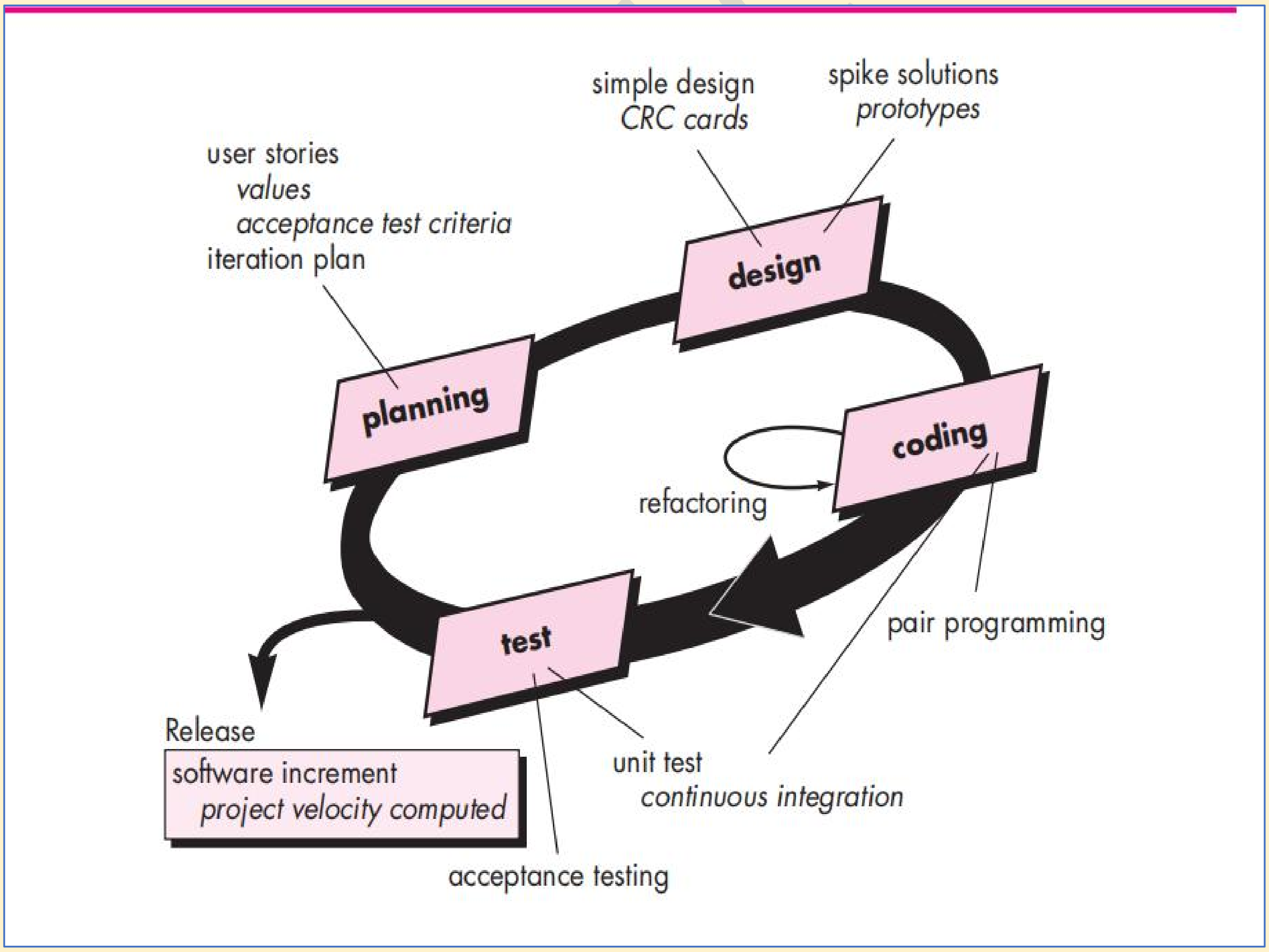
• XP is a lightweight, efficient, low-risk, flexible, predictable, scientific, and fun way to develop a software.

• Small to medium sized teams that work under vague and rapidly changing requirements.

• The five values of XP are communication, simplicity, feedback, courage, and respect

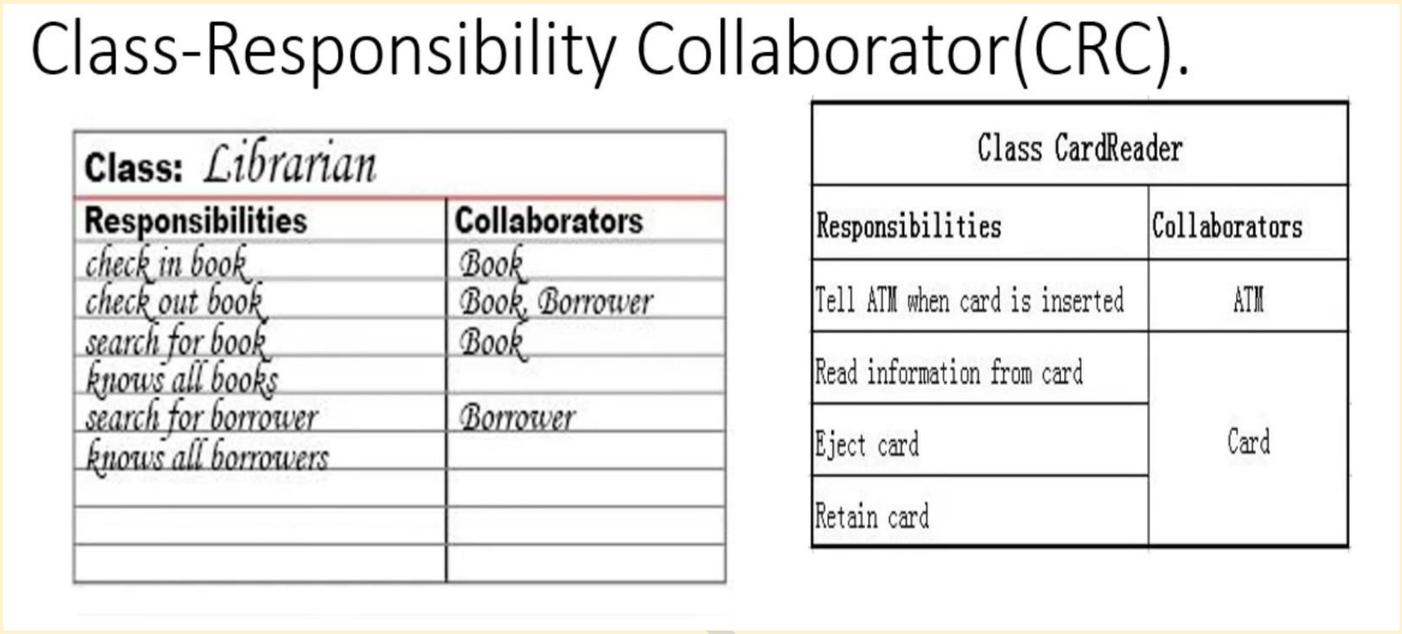
• Follows Object Oriented approach.

The Process & Roles of Extreme Programming



>>Keywords:

• **CRC cards:** CRC cards, or Class-Responsibility-Collaboration cards, are a brainstorming and design tool used in object-oriented software engineering. They consist of index cards divided into three sections, which help in identifying the classes within a system, along with their responsibilities and collaborators.



Class-Responsibility Collaborator(CRC).

Class: *Librarian*	
Responsibilities	**Collaborators**
check in book	*Book*
check out book	*Book, Borrower*
search for book	*Book*
knows all books	
search for borrower	*Borrower*
knows all borrowers	

Class CardReader	
Responsibilities	Collaborators
Tell ATM when card is inserted	ATM
Read information from card	Card
Eject card	
Retain card	

• **Spike solution Prototype:** A spike solution is a short-term experiment that helps developers explore complex or unknown issues. They can be useful for testing feasibility, reducing risks, and learning new skills or technologies.

• **Refactoring (Simplify):** Refactoring is about removing redundancy, eliminating unnecessary functions, increasing code coherency, and at the same time decoupling elements. *Keep your code clean and simple*, so you can easily understand and modify it when required would be the advice of any XP team member.

• **Pair Programming:** This practice requires two programmers to work jointly on the same code.

• **Project Velocity:** It is a metric used to measure how much work a team can complete within a specific time frame.

Values of extreme programming

XP has simple rules that are based on 5 values to guide the teamwork:

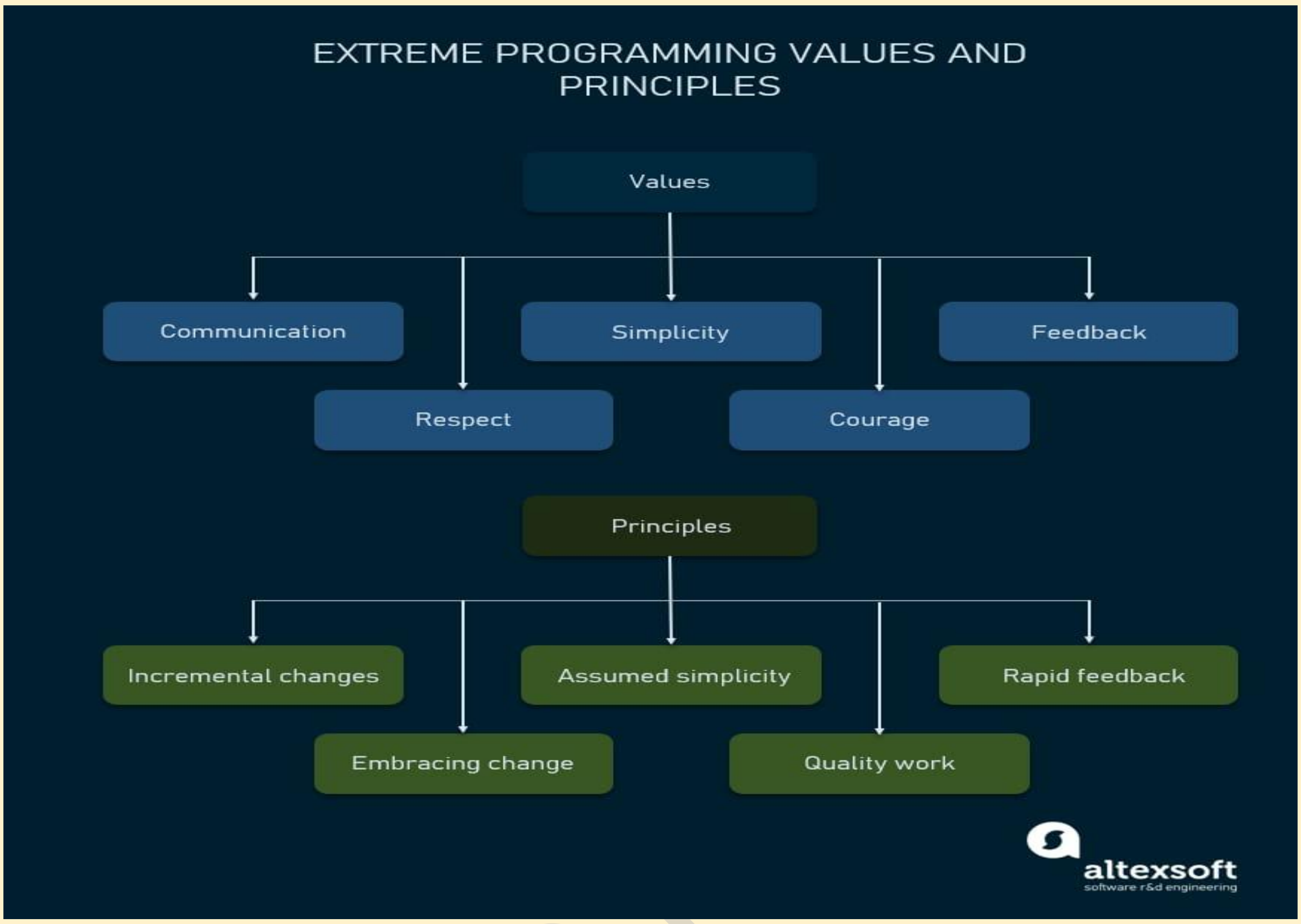
1. **Communication.** Everyone on a team works jointly at every stage of the project.
2. **Simplicity.** Developers strive to write simple code bringing more value to a product, as it saves time and effort.
3. **Feedback.** Team members deliver software frequently, get feedback about it, and improve a product according to the new requirements.
4. **Respect.** Every person assigned to a project contributes to a common goal.
5. **Courage.** Programmers objectively evaluate their own results without making excuses and are always ready to respond to changes.

These values represent a specific mindset of motivated team players who do their best on the way to achieving a common goal. XP principles derive from these values and reflect them in more concrete ways.

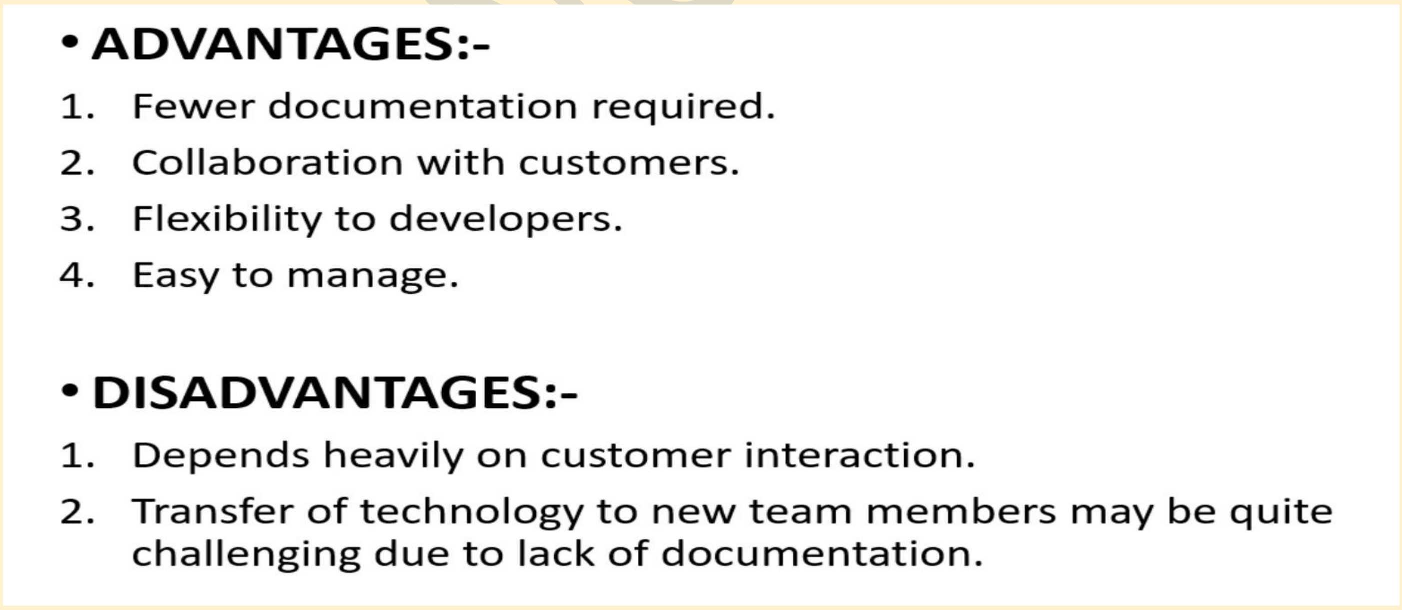
Principles of extreme programming

Most researchers denote 5 XP principles as:

1. **Rapid feedback.** Team members understand the given feedback and react to it right away.
2. **Assumed simplicity.** Developers need to focus on the job that is important at the moment and follow YAGNI (You Ain't Gonna Need It) and DRY (Don't Repeat Yourself) principles.
3. **Incremental changes.** Small changes made to a product step by step work better than big ones made at once.
4. **Embracing change.** If a client thinks a product needs to be changed, programmers should support this decision and plan how to implement new requirements.
5. **Quality work.** A team that works well, makes a valuable product and feels proud of it.



EXTREME PROGRAMMING(XP)



- **ADVANTAGES:-**

1. Fewer documentation required.
2. Collaboration with customers.
3. Flexibility to developers.
4. Easy to manage.

- **DISADVANTAGES:-**

1. Depends heavily on customer interaction.
2. Transfer of technology to new team members may be quite challenging due to lack of documentation.

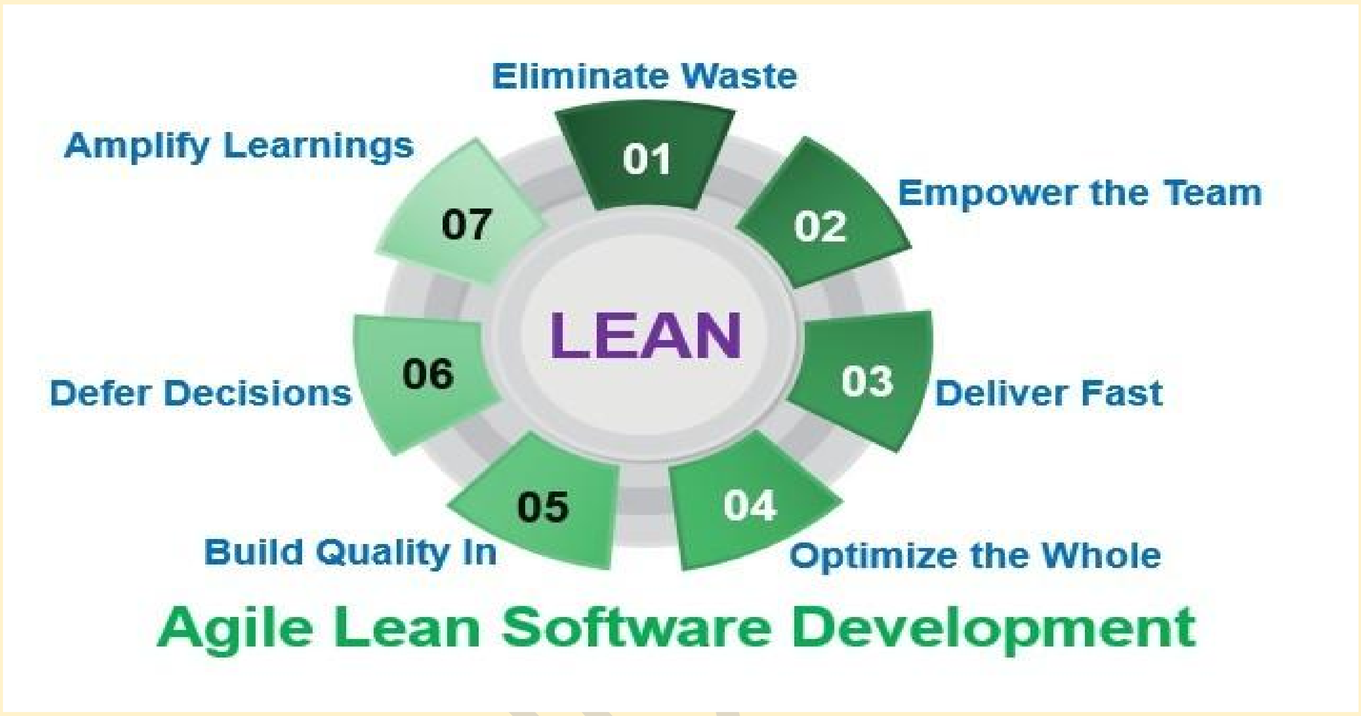
WEBSITE LINK- https://www.altexsoft.com/blog/extreme-programming-values-principles-and-practices/

Video Link- https://www.youtube.com/watch?v=7G4uN4S0jD4

Lean Software Development(LSD)

The **Poppendiecks** created **Lean Software Development (LSD)** by importing lean manufacturing ideas into software development. **LSD** focuses more on principles and, to me, feels less concrete compared to other agile methods.

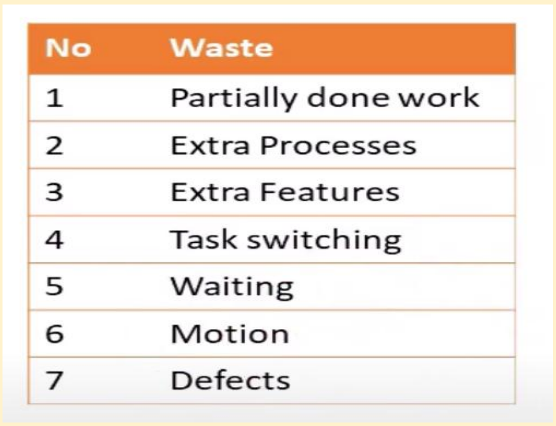
The Process Of LSD:



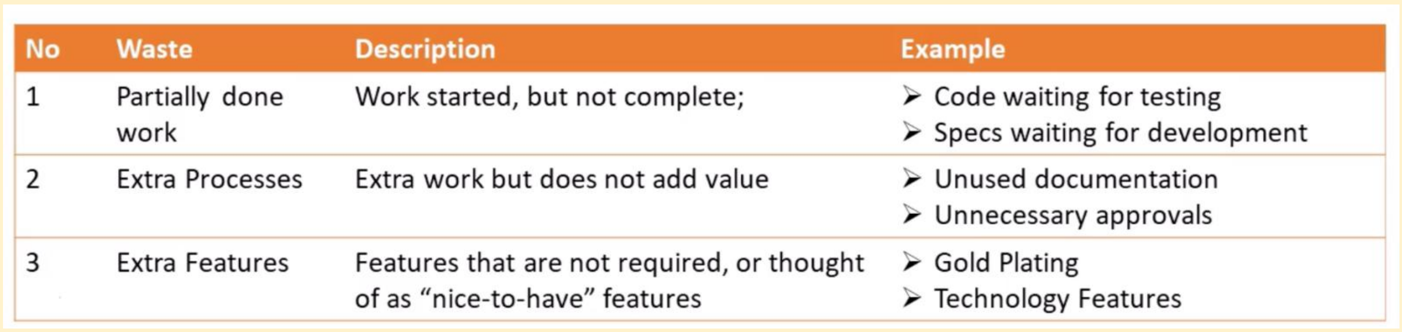
Now look at these 7 Core Concepts in more detail:

1. Eliminate Wastes: To maximize value, We must minimize Waste. For software systems, Waste can take the form of partially done work, delays, hand-offs, unnecessary features etc.. Therefore to increase the value we are getting from projects, we must develop ways to identify and then remove waste.

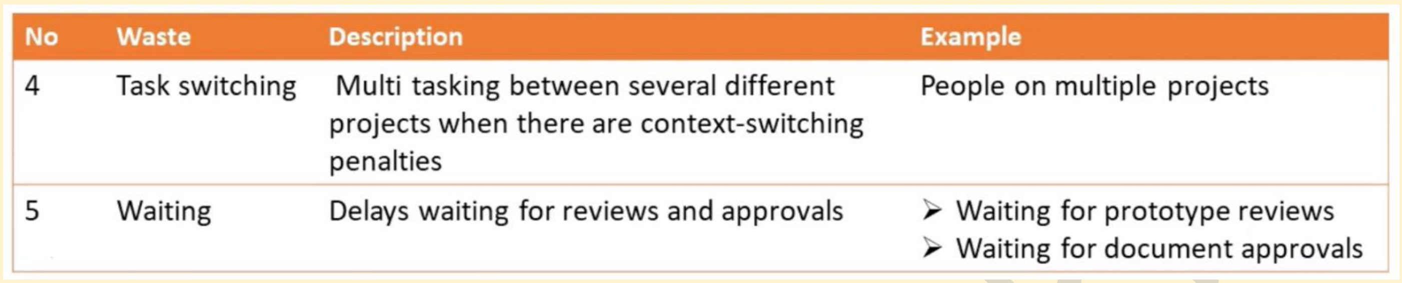
Example of wastes-



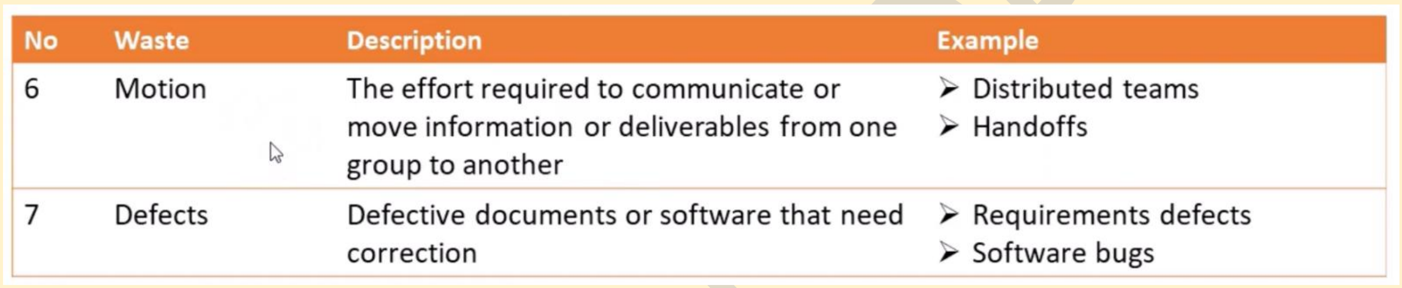
No	Waste
1	Partially done work
2	Extra Processes
3	Extra Features
4	Task switching
5	Waiting
6	Motion
7	Defects



No	Waste	Description	Example
1	Partially done work	Work started, but not complete;	➢ Code waiting for testing ➢ Specs waiting for development
2	Extra Processes	Extra work but does not add value	➢ Unused documentation ➢ Unnecessary approvals
3	Extra Features	Features that are not required, or thought of as "nice-to-have" features	➢ Gold Plating ➢ Technology Features



No	Waste	Description	Example
4	Task switching	Multi tasking between several different projects when there are context-switching penalties	People on multiple projects
5	Waiting	Delays waiting for reviews and approvals	➢ Waiting for prototype reviews ➢ Waiting for document approvals



No	Waste	Description	Example
6	Motion	The effort required to communicate or move information or deliverables from one group to another	➢ Distributed teams ➢ Handoffs
7	Defects	Defective documents or software that need correction	➢ Requirements defects ➢ Software bugs

2. Empower the team: Rather than taking a micromanagement approach, we should respect team members superior knowledge of the technical steps required on the project and let them make local decisions to be productive and successful.

3. Deliver Fast: We can maximize the project Return on investment (ROI) by quickly delivering valuable software and iterating through designs. We find the best solution through the Rapid Evolution of options. (It basically follows incremental model)

4. Optimize the Whole: We aim to see the system as more than the sum of its parts. We go beyond the pieces of the project and look for how it aligns with the organization. As part of optimizing the whole, we also focus on forming better inter-group relations.

5. Build quality in: Lean development doesn't try to "test-in" quality at the end; instead, we build quality into the product and continually assure quality throughout the development process, using techniques like refactoring, continuous integration and unit testing. (Quality over Quantity)

6. Defer Decisions: We balance early planning with making decisions and committing to things as late as possible. For example, this may mean re-prioritizing the backlog right up until it is time to plan an iteration, or avoiding being tide to an early technology-bounded solution.

Example- for example in 'US' they will not allow to choose the major stream at first like AI/DataScience/Machine Learning/Cyber security/Game-development/graphic design/web developmet/app development etc. Only they will focus on Common stream like CSE for first 2 or 3 years in B.Tech/B.E, after 2 or 3 years student will decide & get into their own Interested domain.

7. Amplify Learning: This concept involves facilitating communication early and often, getting feedback as soon as possible, and building on what we learn. Software projects are business and technology learning experiences, so we should start soon and keep learning. (It means we need to learn from the ground or base & we need to listen feedback from team members in both ways one is the Content point of view and another is the emotional point of view.)

WEBSITE LINK- https://hangoutagile.com/lean-software-development-wave-ii/

Video Link- https://www.youtube.com/watch?v=aoHDa3L2ngI

GitHub

>> What Is GitHub?

GitHub is an online platform for software development that allows users to store, track, and collaborate on software projects. It utilizes Git, an open-source version control system, enabling multiple developers to work on a project simultaneously.

>> Why We Use GitHub?

GitHub is used for storing, tracking, and collaborating on software projects. It allows developers to maintain code in repositories, manage version control, and work together with others on coding tasks, which enhances collaboration and project management. (mainly publish our project & for group work also)

>> What is Version Control?

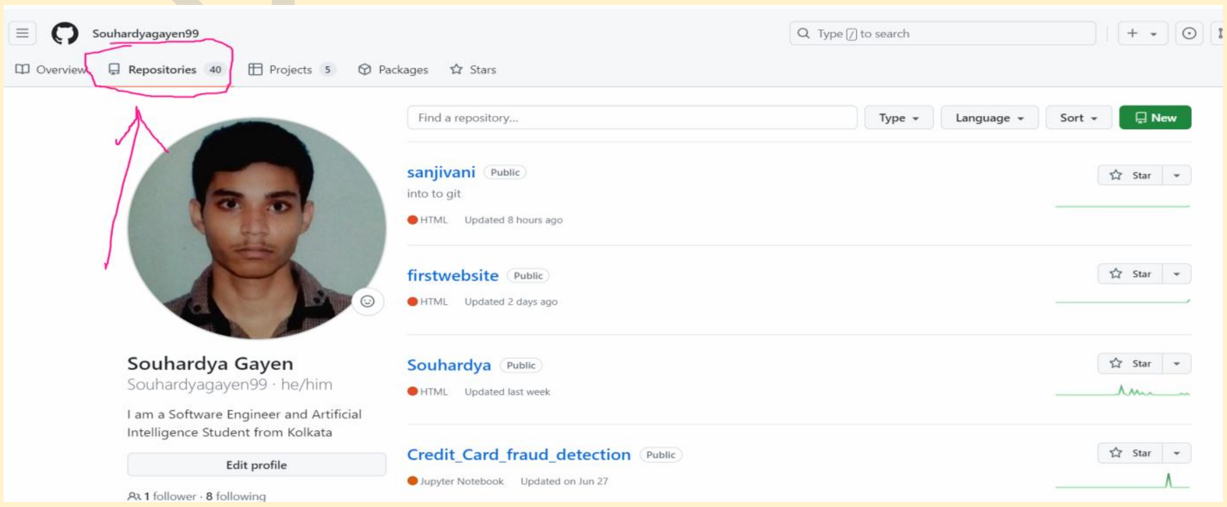
Version control is a system that helps manage changes to documents, programs, and other collections of information over time. It allows multiple users to collaborate on projects while keeping track of each change made. Here are some key aspects of version control. (for updating a version of any application/website/software)

>> How does version control work in GitHub?

Version control in GitHub is primarily managed through Git, a distributed version control system that allows multiple people to work on a project simultaneously without overwriting each other's changes.

>> What is a GitHub repository?

It's a place where you can store your code, your files, and each file's revision history. Repositories can have multiple collaborators and can be either public, internal, or private.



Souhardya Gayen
Souhardyagayen99 · he/him
I am a Software Engineer and Artificial Intelligence Student from Kolkata
1 follower · 8 following

sanjivani Public
into to git
HTML Updated 8 hours ago

firstwebsite Public
HTML Updated 2 days ago

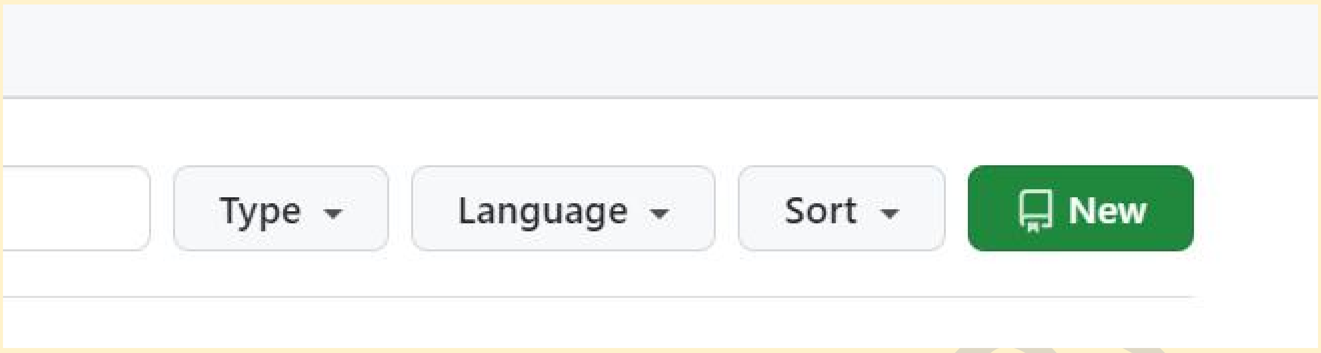
Souhardya Public
HTML Updated last week

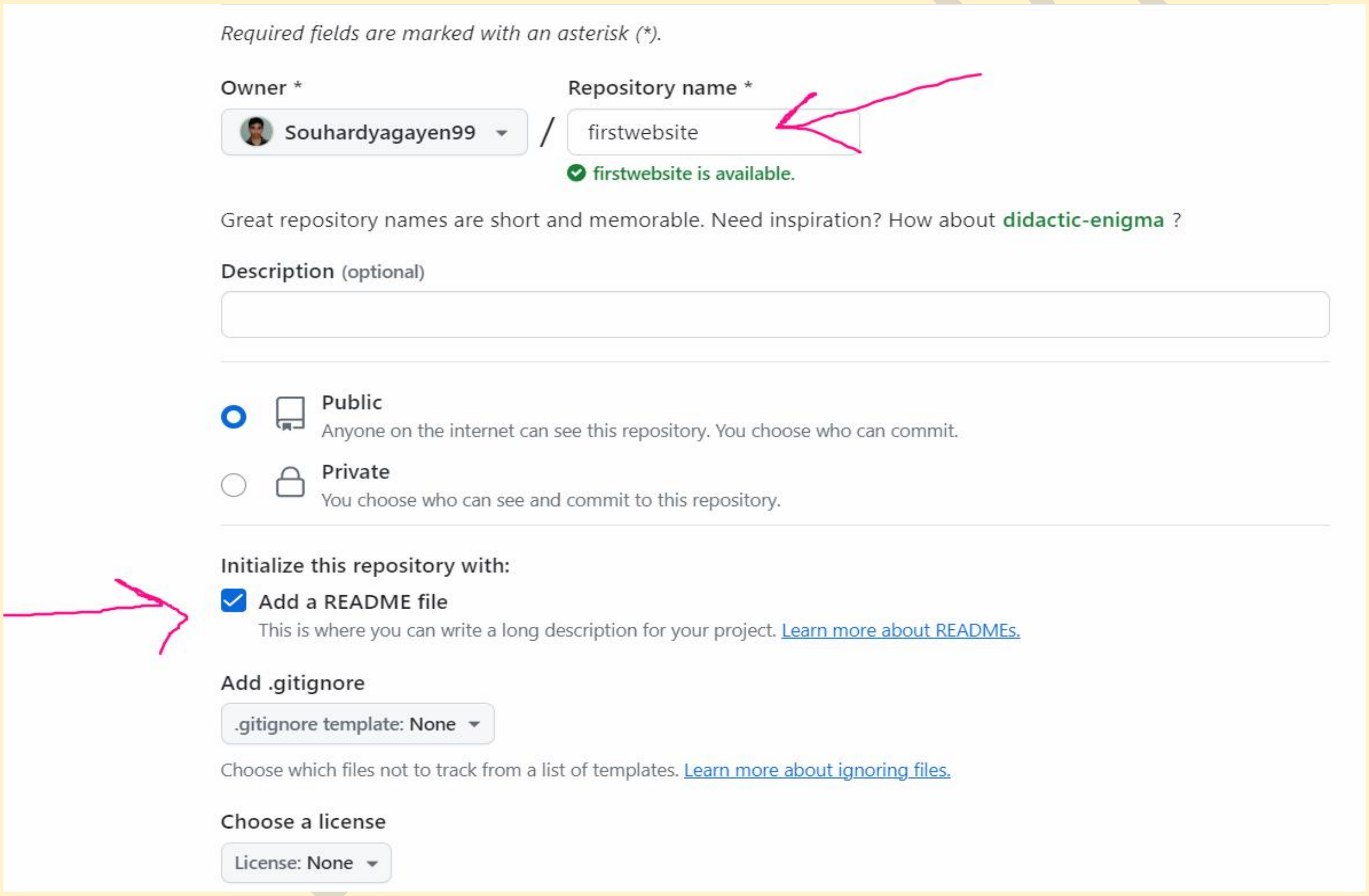
Credit_Card_fraud_detection Public
Jupyter Notebook Updated on Jun 27

First of all, Create an ID on GitHub [means Sign Up and Sign In]

(link- https://github.com/)

After that Create GitHub "New Repositories"




Required fields are marked with an asterisk (*).

Owner * Souhardyagayen99 / Repository name * firstwebsite

firstwebsite is available.

Great repository names are short and memorable. Need inspiration? How about didactic-enigma ?

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.

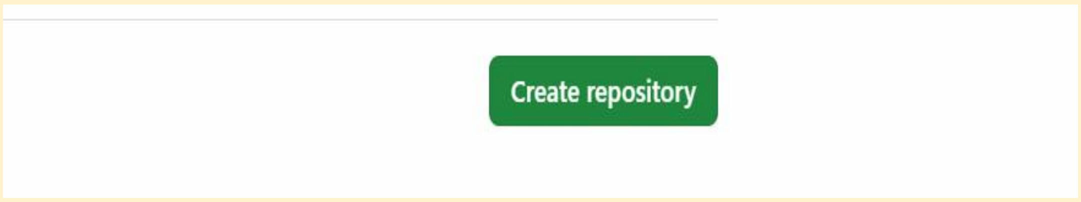
Add .gitignore

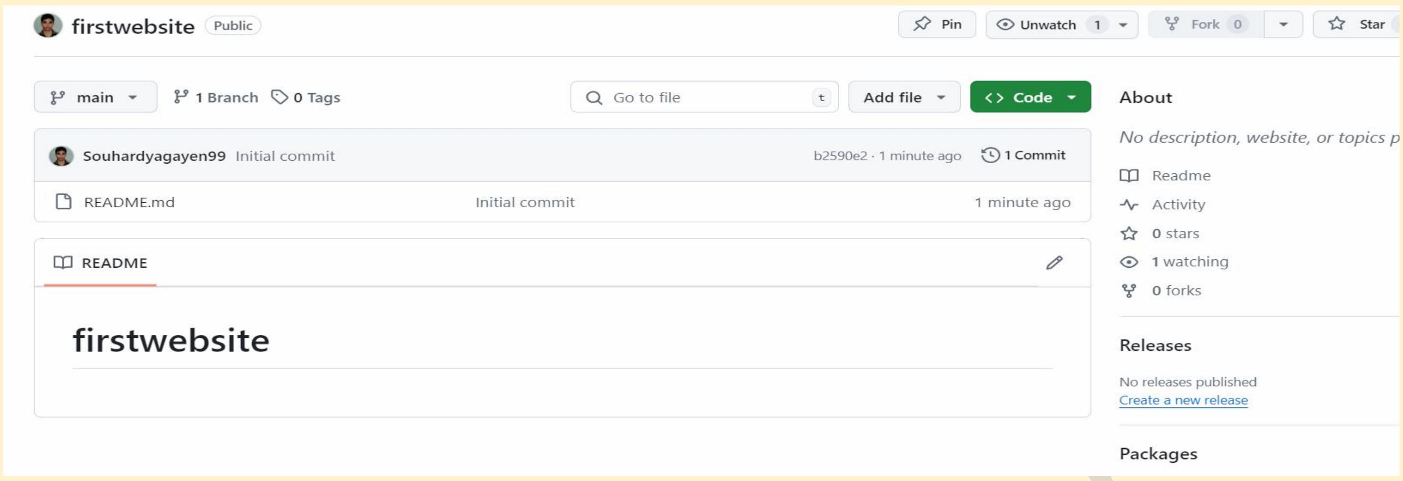
.gitignore template: None

Choose which files not to track from a list of templates. Learn more about ignoring files.

Choose a license

License: None




firstwebsite Public

Pin · Unwatch 1 · Fork 0 · Star

main · 1 Branch · 0 Tags · Go to file · Add file · Code

Souhardyagayen99 Initial commit b2590e2 · 1 minute ago 1 Commit

README.md Initial commit 1 minute ago

README

firstwebsite

About

No description, website, or topics p

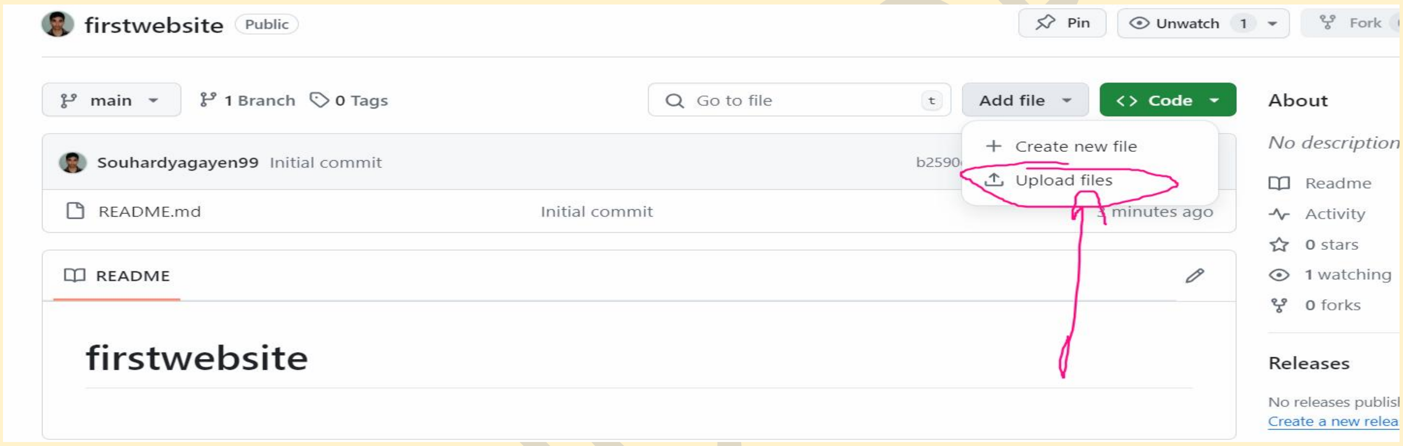
Readme · Activity · 0 stars · 1 watching · 0 forks

Releases

No releases published
Create a new release

Packages

>>Then Upload File


firstwebsite Public

main · 1 Branch · 0 Tags · Go to file · Add file · Code

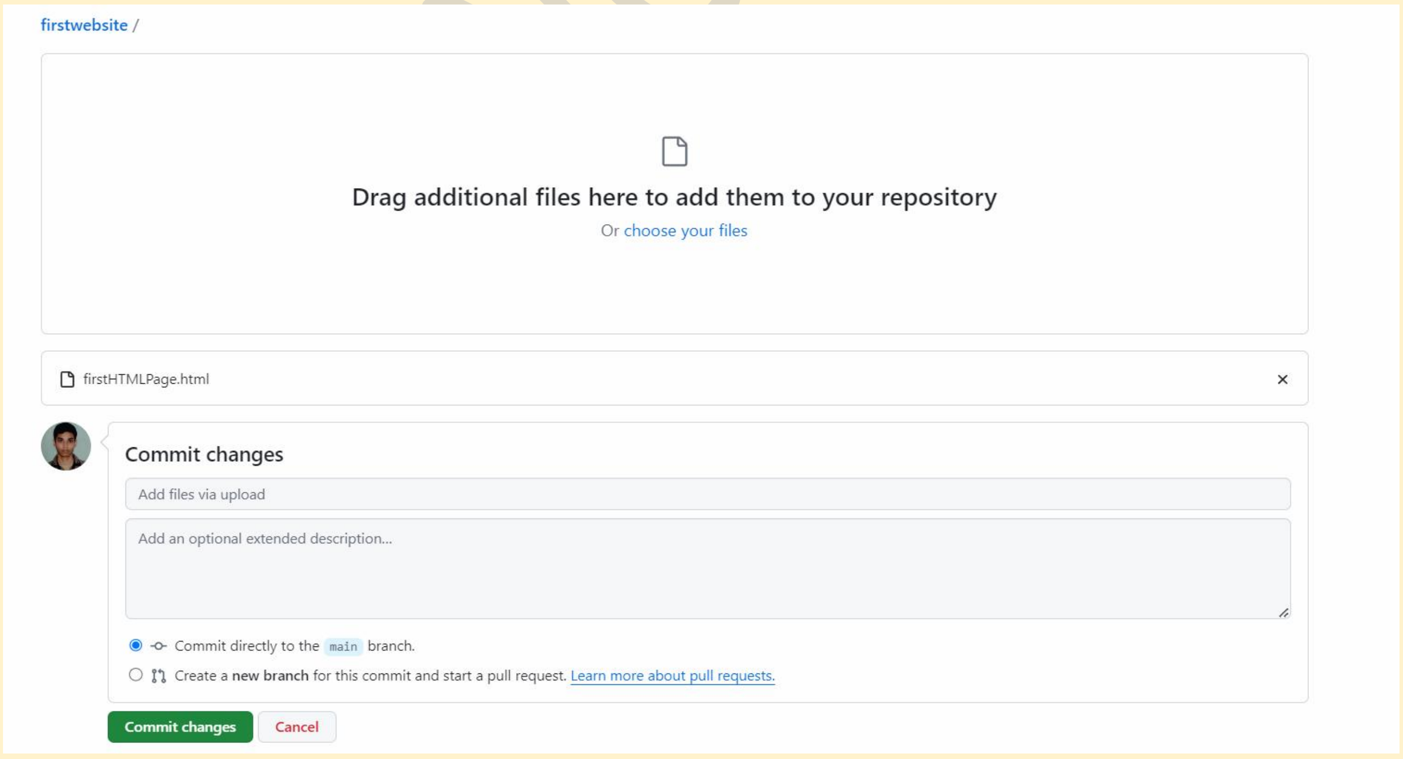
Add file: Create new file · Upload files

Souhardyagayen99 Initial commit b2590

README.md Initial commit minutes ago

README

firstwebsite


firstwebsite /

Drag additional files here to add them to your repository

Or choose your files

firstHTMLPage.html

Commit changes

Add files via upload

Add an optional extended description...

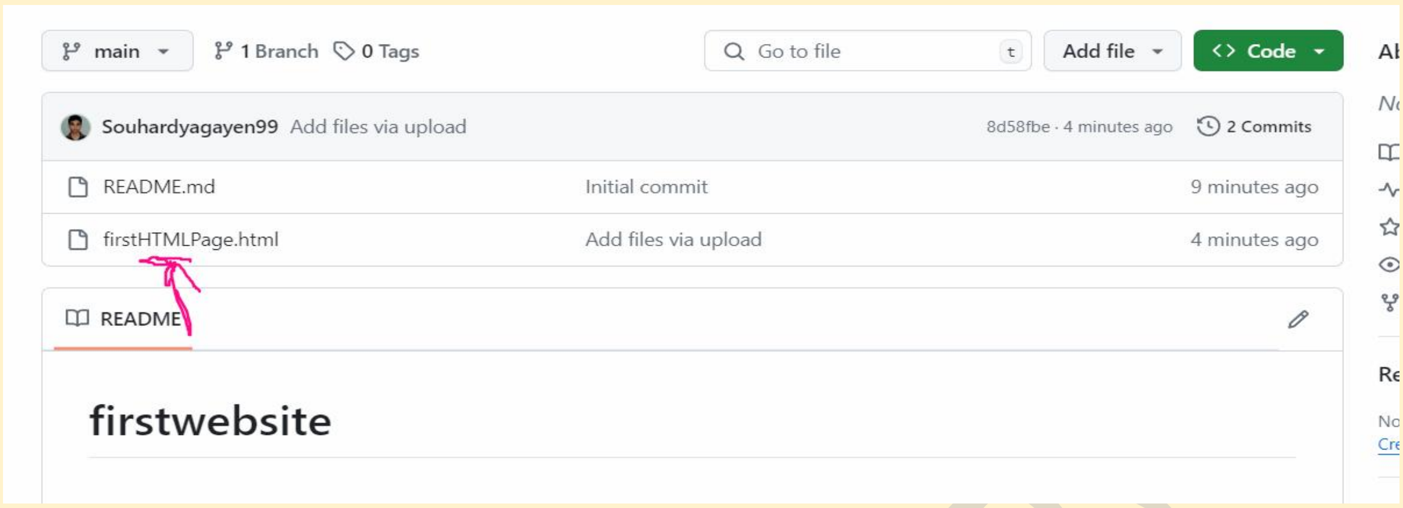
[✓ Commit directly to the `main` branch.]

Create a new branch for this commit and start a pull request. Learn more about pull requests.

Commit changes · Cancel

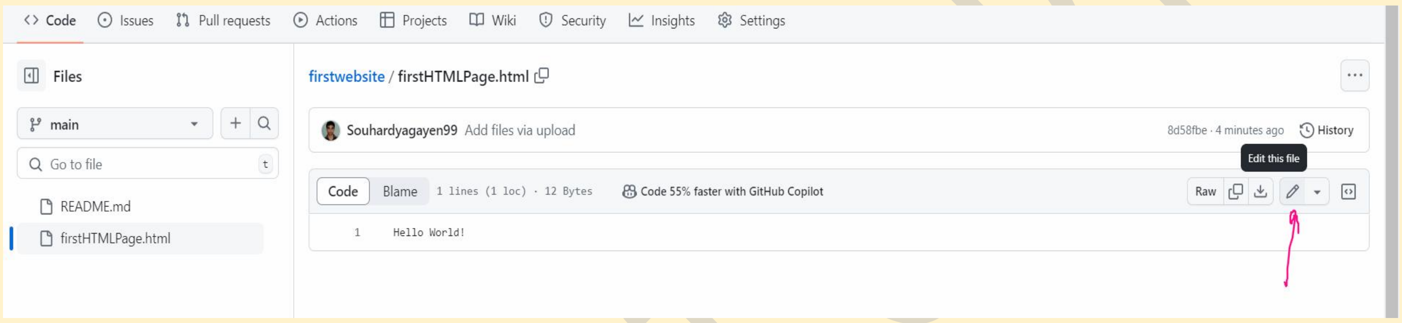
After Upload click “Commit Changes”

>>Then EDIT File



Souhardyagayen99 Add files via upload	8d58fbe · 4 minutes ago	2 Commits
README.md	Initial commit	9 minutes ago
firstHTMLPage.html	Add files via upload	4 minutes ago

firstwebsite

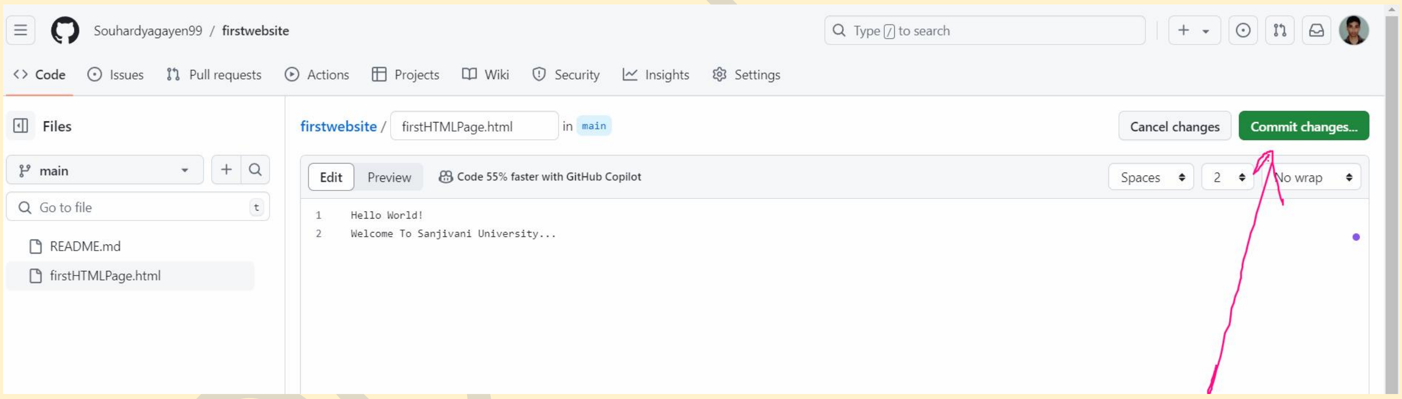


firstwebsite / firstHTMLPage.html

Souhardyagayen99 Add files via upload 8d58fbe · 4 minutes ago

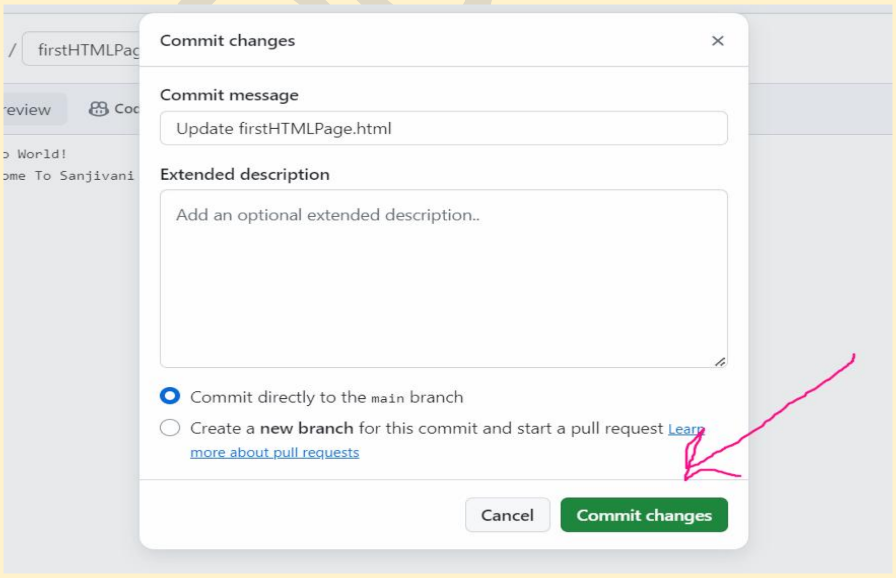
1 lines (1 loc) · 12 Bytes

```
Hello World!
```



firstwebsite / firstHTMLPage.html in main

```
Hello World!
Welcome To Sanjivani University...
```



Commit message

Update firstHTMLPage.html

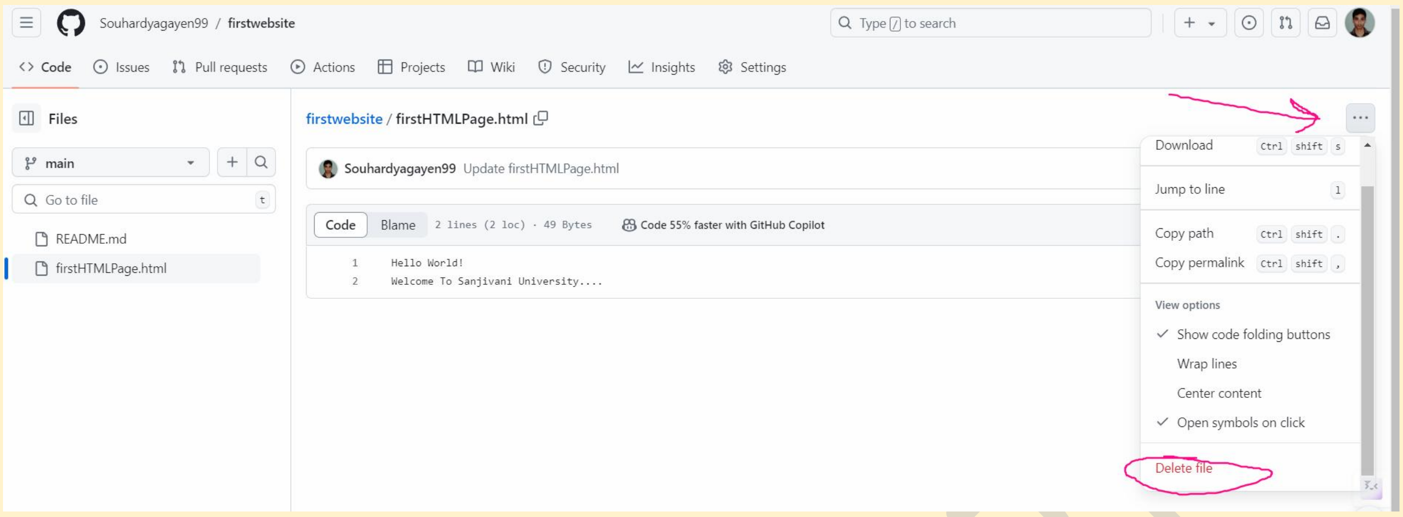
Extended description

Add an optional extended description..

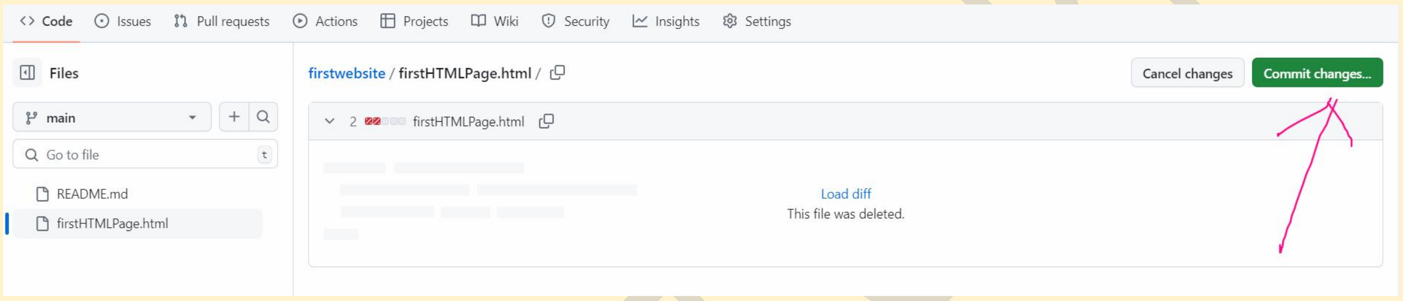
[✓ Commit directly to the main branch]

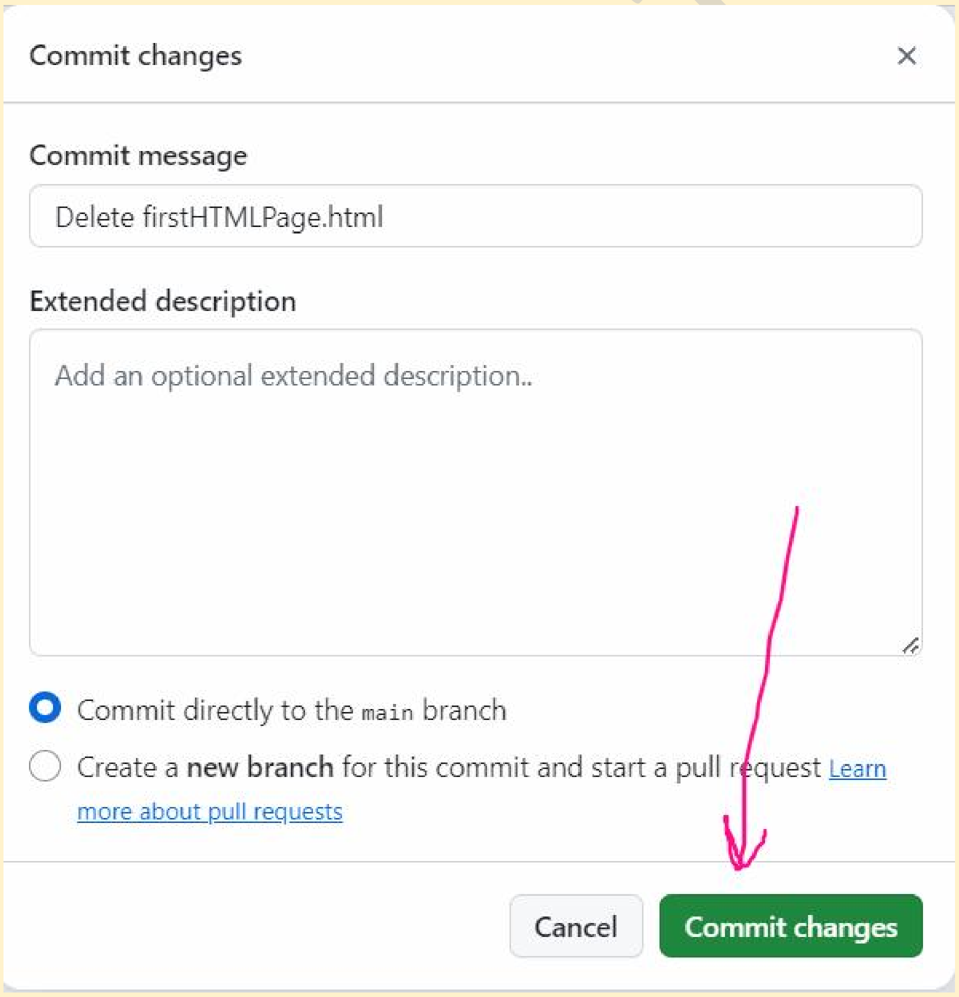
Create a **new branch** for this commit and start a pull request Learn more about pull requests

>>There-After Delete a Particular file



```
Hello World!
Welcome To Sanjivani University....
```




Commit changes

Commit message

Delete firstHTMLPage.html

Extended description

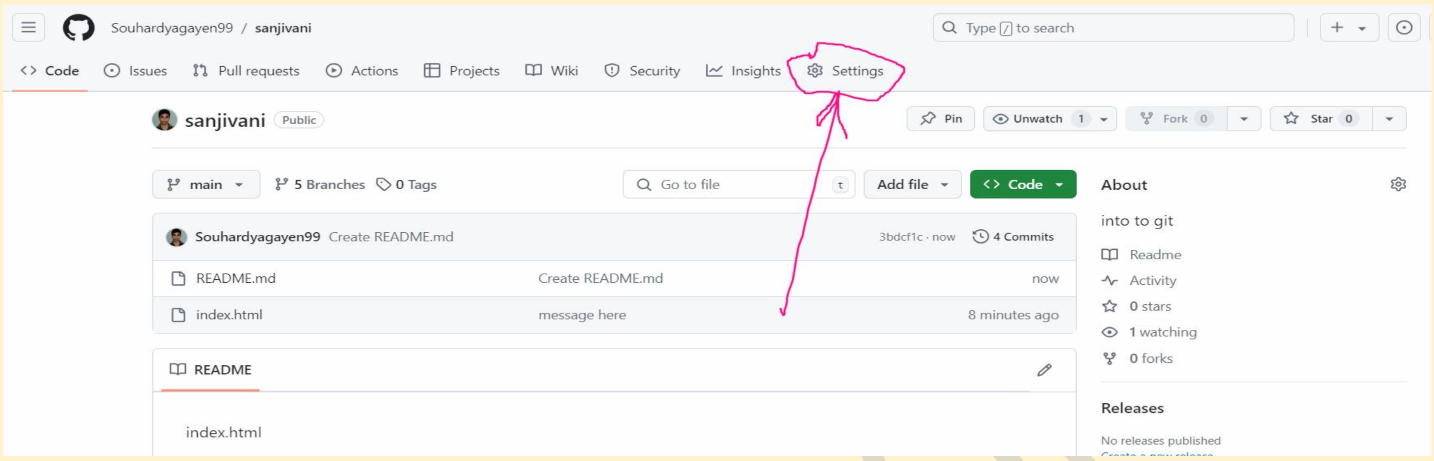
Add an optional extended description..

[✓ Commit directly to the `main` branch]

Create a **new branch** for this commit and start a pull request Learn more about pull requests

>>To Deploy/Publish/Host Our website

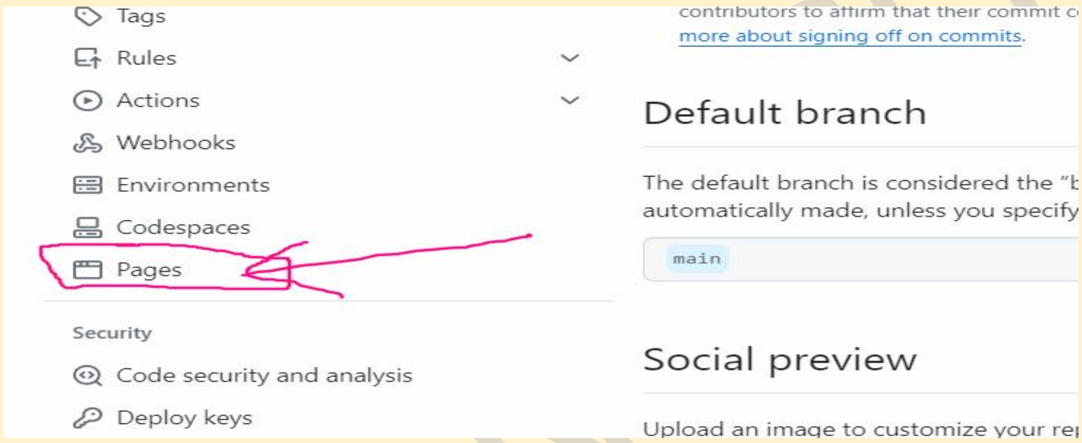
NOTE: For host in GitHub the name should be index.html otherwise it will not work Because github set this policy that Entry Page/Home Page/Landing Page Should be index.html



Souhardyagayen99 Create README.md		3bdcf1c · now 4 Commits
README.md	Create README.md	now
index.html	message here	8 minutes ago

README

index.html



contributors to affirm that their commit c
more about signing off on commits.

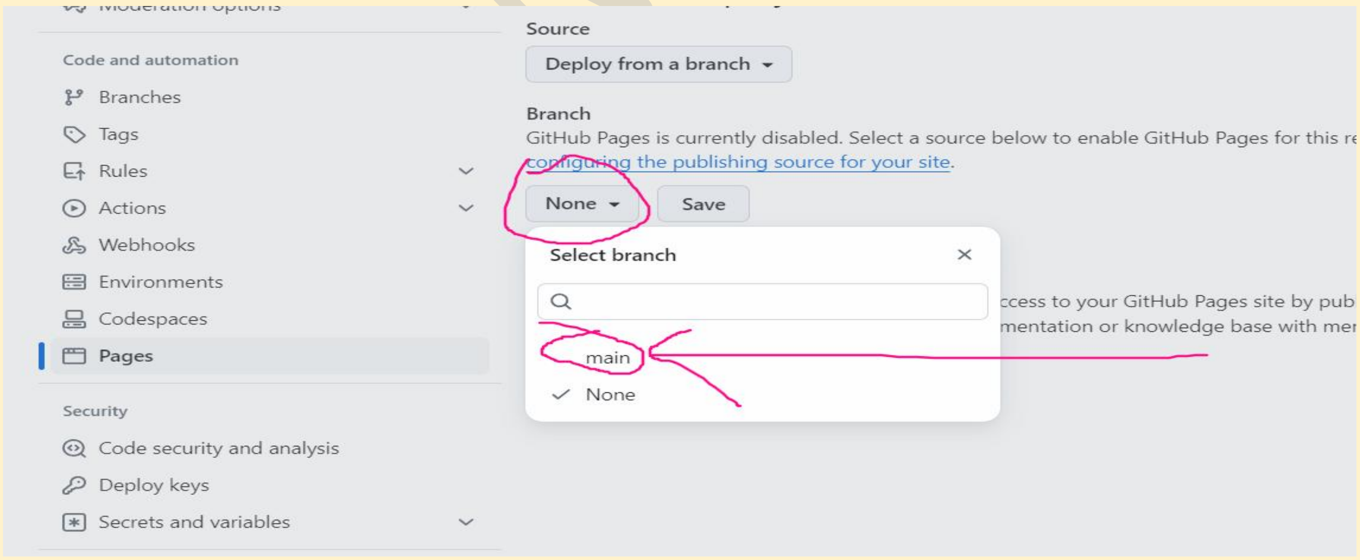
Default branch

The default branch is considered the "b
automatically made, unless you specify

main

Social preview

Upload an image to customize your re



Source

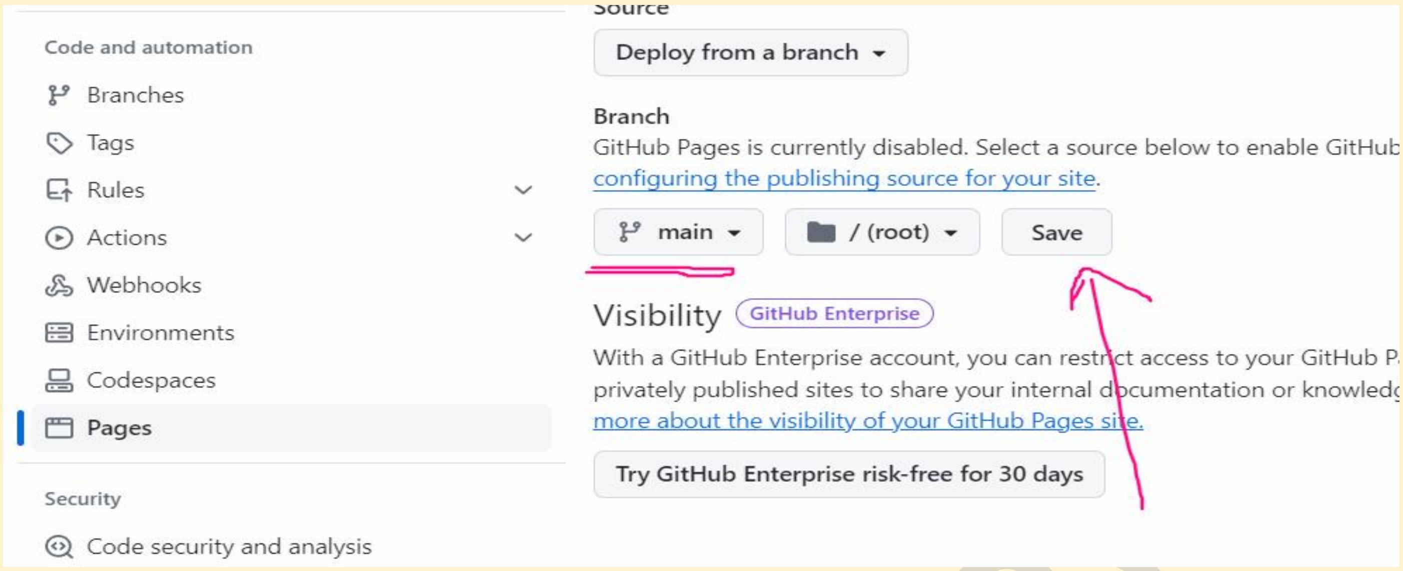
Deploy from a branch

Branch

GitHub Pages is currently disabled. Select a source below to enable GitHub Pages for this re
configuring the publishing source for your site.

ccess to your GitHub Pages site by pub
mentation or knowledge base with me

LINK- https://www.youtube.com/watch?v=4eMnz8VbgyM&t=305s (see the timestamp at 2:30 min)


Code and automation

Branches

Tags

Rules

Actions

Webhooks

Environments

Codespaces

Pages

Security

Code security and analysis

Source

Deploy from a branch

Branch

GitHub Pages is currently disabled. Select a source below to enable GitHub

configuring the publishing source for your site.

main / (root) Save

Visibility GitHub Enterprise

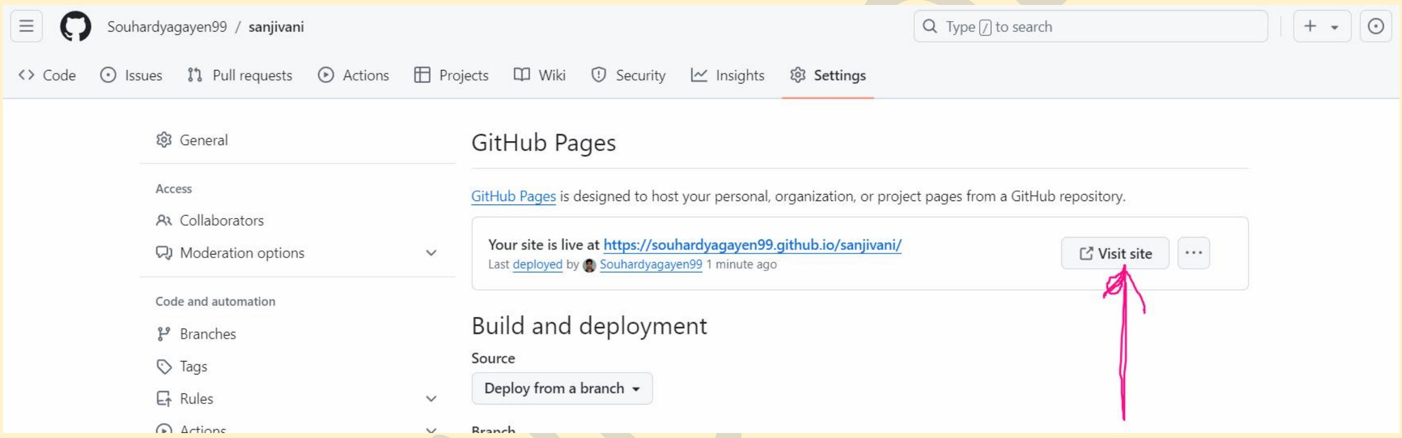
With a GitHub Enterprise account, you can restrict access to your GitHub P

privately published sites to share your internal documentation or knowled

more about the visibility of your GitHub Pages site.

Try GitHub Enterprise risk-free for 30 days

After 2-3 minutes Refresh/Reload the page and see…


Souhardyagayen99 / sanjivani

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Rules

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Your site is live at https://souhardyagayen99.github.io/sanjivani/

Last deployed by Souhardyagayen99 1 minute ago

Visit site

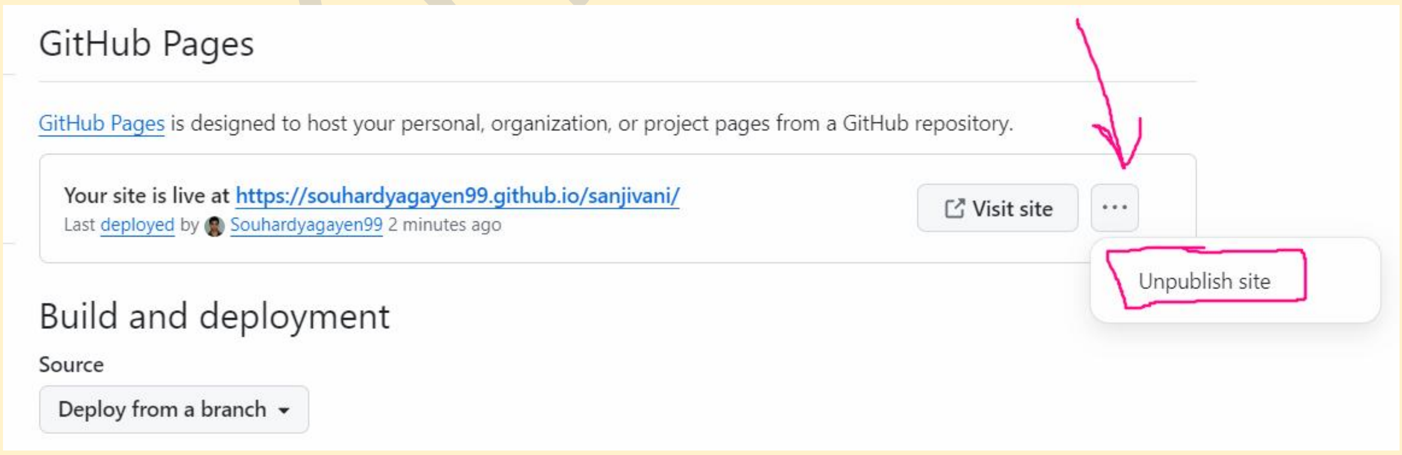
Build and deployment

Source

Deploy from a branch

You can share this link with your friends.

For Unpublish.


GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Your site is live at https://souhardyagayen99.github.io/sanjivani/

Last deployed by Souhardyagayen99 2 minutes ago

Visit site

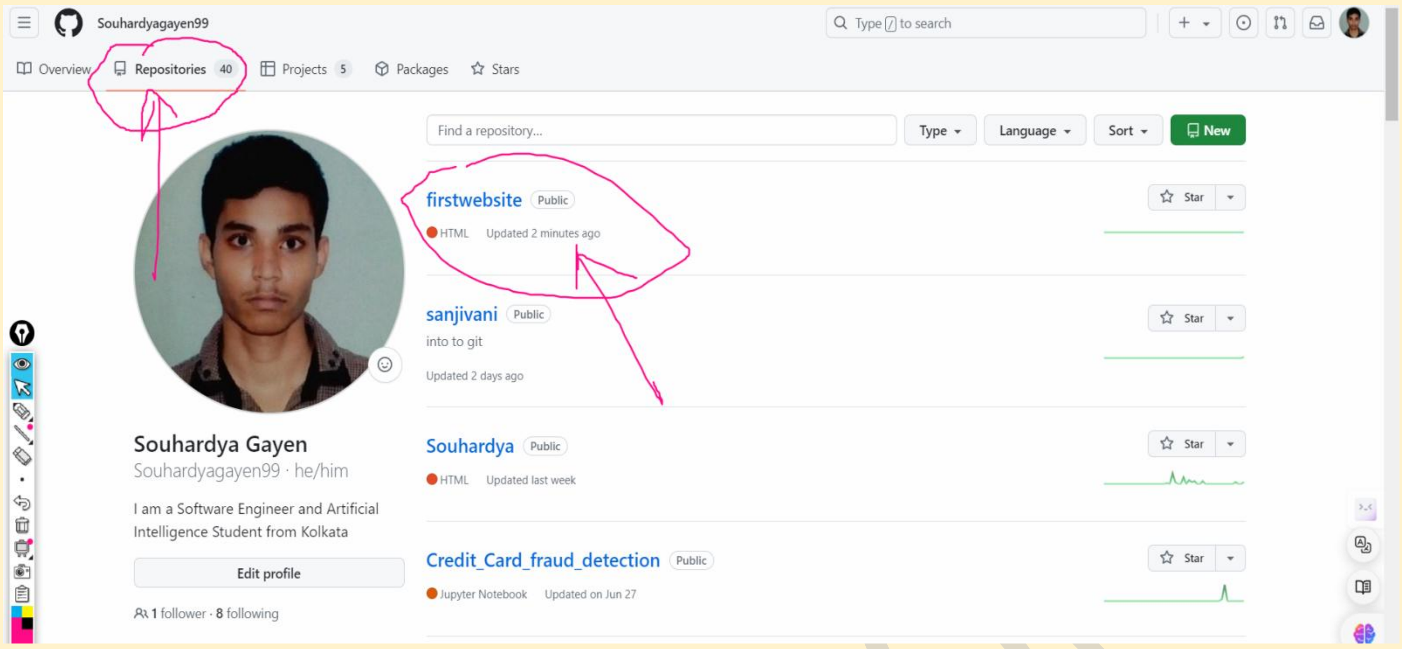
Unpublish site

Build and deployment

Source

Deploy from a branch

>>Now Delete A Full Repository


Souhardyagayen99

Overview | Repositories 40 | Projects 5 | Packages | Stars

Find a repository... | Type | Language | Sort | New

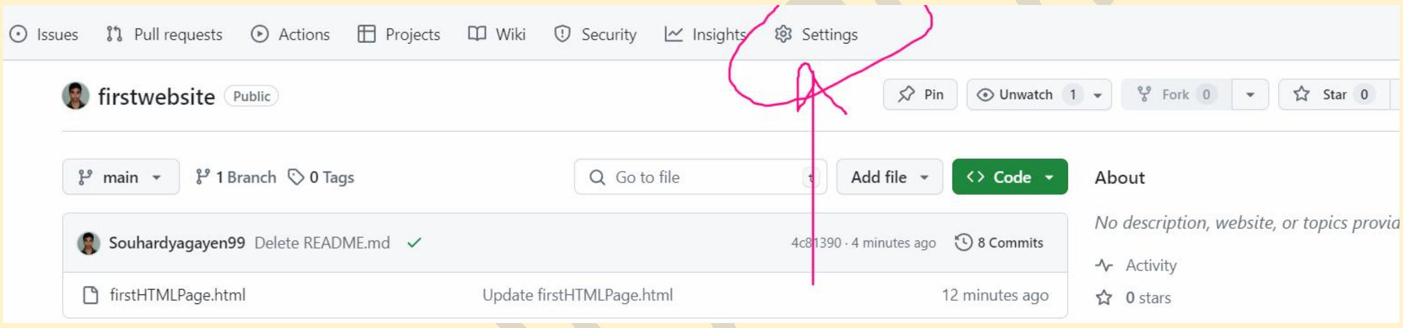
firstwebsite Public
HTML Updated 2 minutes ago

sanjivani Public
into to git
Updated 2 days ago

Souhardya Public
HTML Updated last week

Credit_Card_fraud_detection Public
Jupyter Notebook Updated on Jun 27

Souhardya Gayen
Souhardyagayen99 · he/him
I am a Software Engineer and Artificial Intelligence Student from Kolkata
Edit profile
1 follower · 8 following


Issues | Pull requests | Actions | Projects | Wiki | Security | Insights | Settings

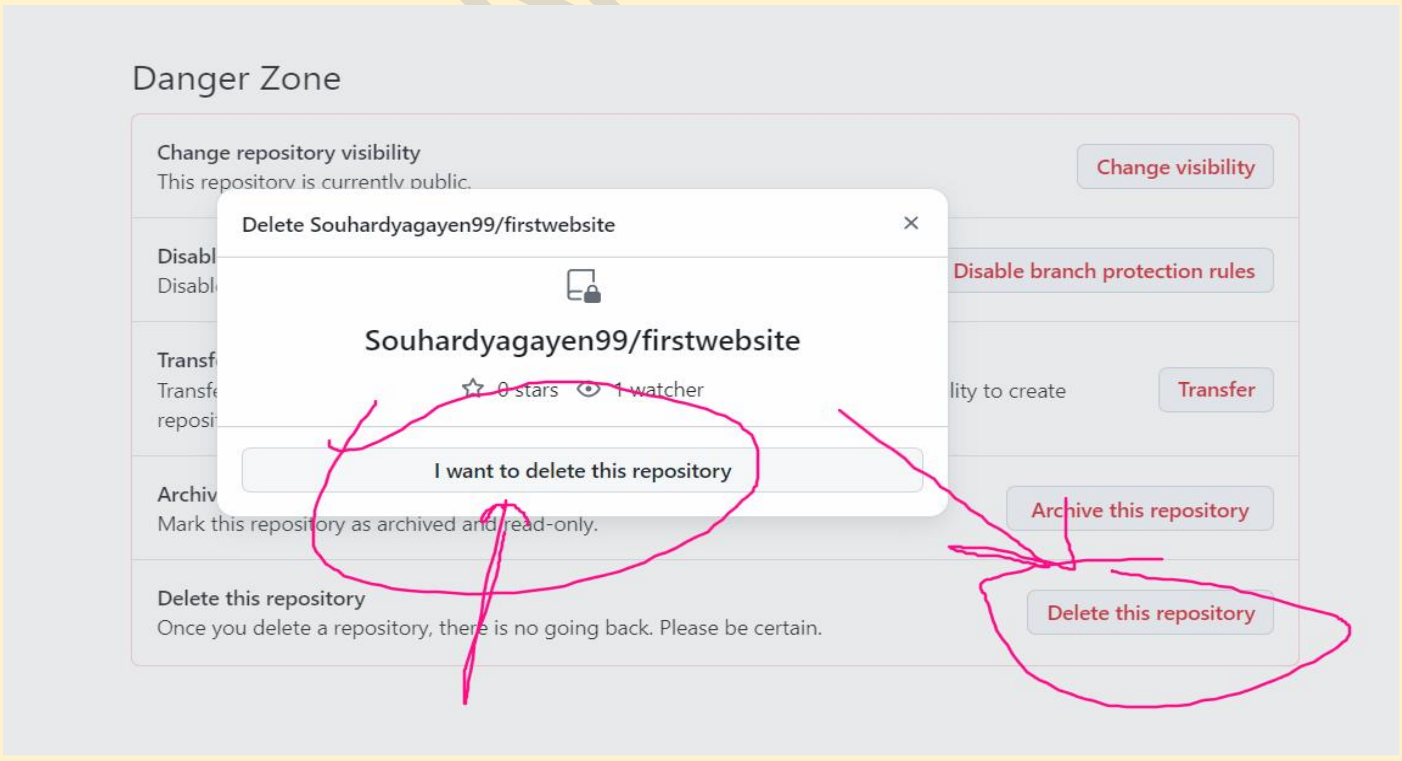
firstwebsite Public | Pin | Unwatch 1 | Fork 0 | Star 0

main | 1 Branch | 0 Tags | Go to file | Add file | Code

Souhardyagayen99 Delete README.md ✓ 4c81390 · 4 minutes ago 8 Commits
firstHTMLPage.html Update firstHTMLPage.html 12 minutes ago

About
No description, website, or topics provid
Activity
0 stars

Scroll Down and go to bottom…


Danger Zone

Change repository visibility
This repository is currently public.
Change visibility

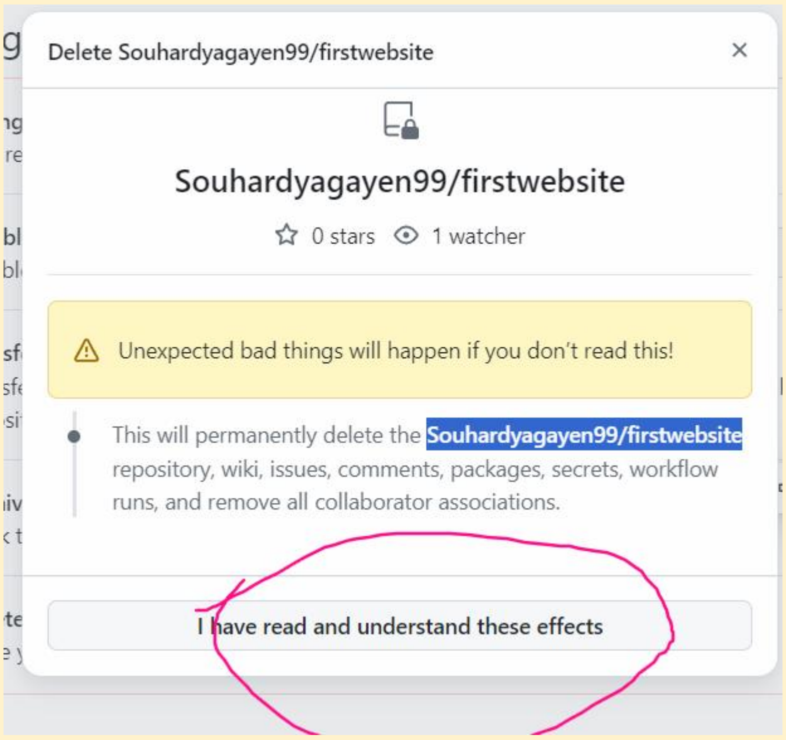
Disable branch protection rules

Transfer

Archive this repository
Mark this repository as archived and read-only.
Archive this repository

Delete this repository
Once you delete a repository, there is no going back. Please be certain.
Delete this repository

Delete Souhardyagayen99/firstwebsite
Souhardyagayen99/firstwebsite
0 stars 1 watcher
I want to delete this repository



Delete Souhardyagayen99/firstwebsite

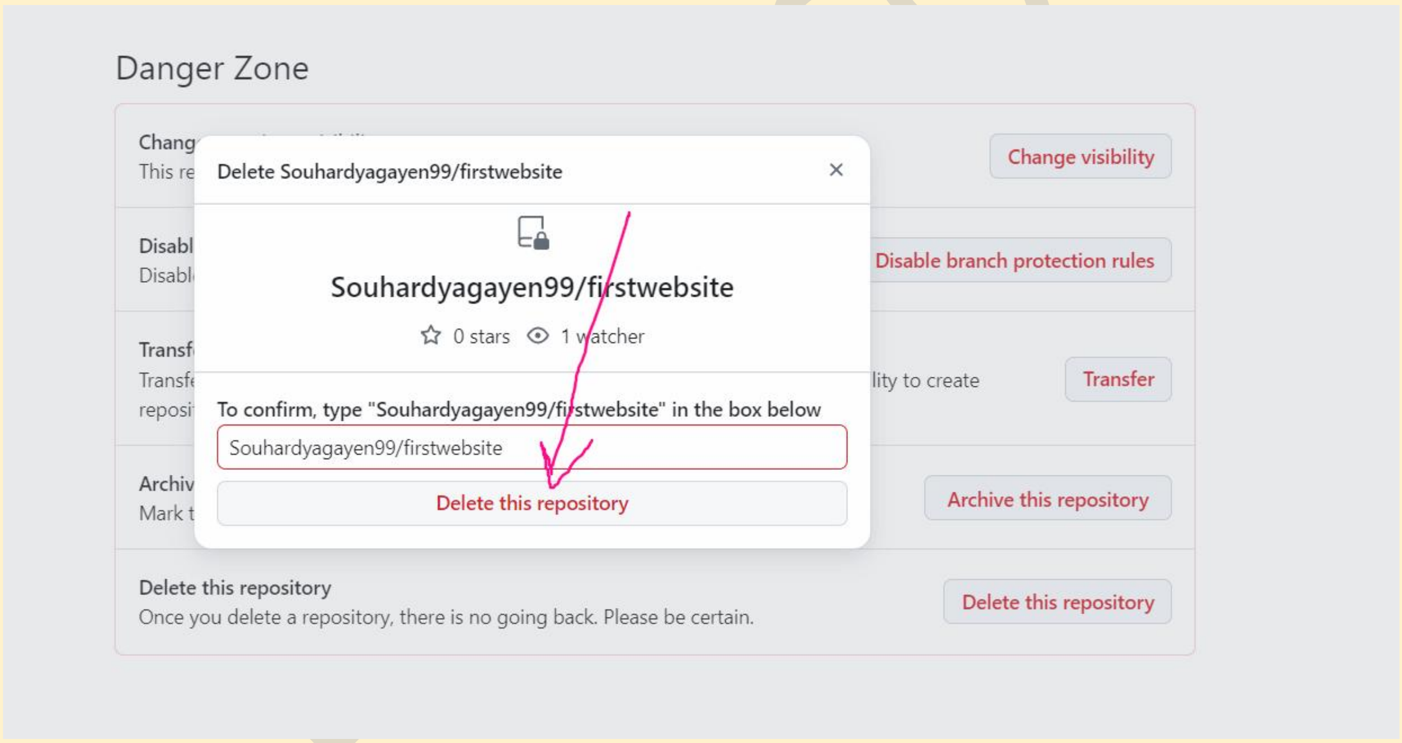
Souhardyagayen99/firstwebsite

0 stars 1 watcher

Unexpected bad things will happen if you don't read this!

This will permanently delete the Souhardyagayen99/firstwebsite repository, wiki, issues, comments, packages, secrets, workflow runs, and remove all collaborator associations.

I have read and understand these effects



Danger Zone

Delete Souhardyagayen99/firstwebsite

Souhardyagayen99/firstwebsite

0 stars 1 watcher

To confirm, type "Souhardyagayen99/firstwebsite" in the box below

Souhardyagayen99/firstwebsite

Delete this repository

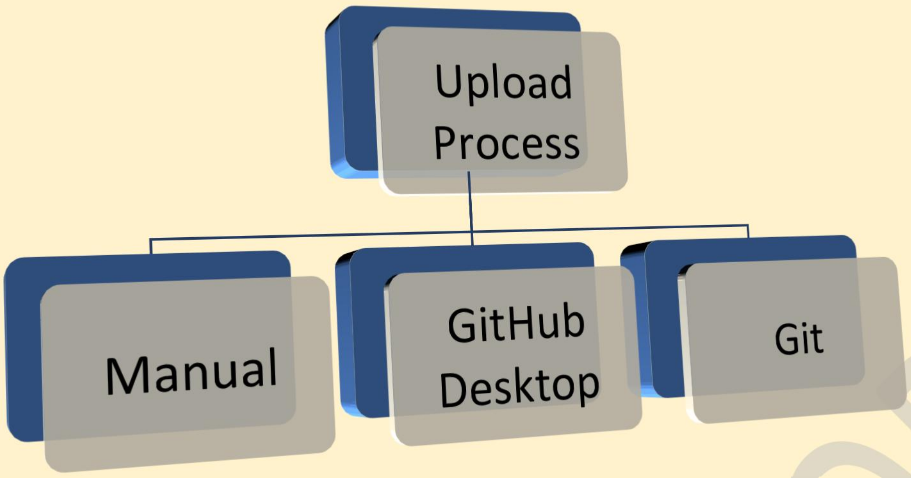
Delete this repository
Once you delete a repository, there is no going back. Please be certain.

*****

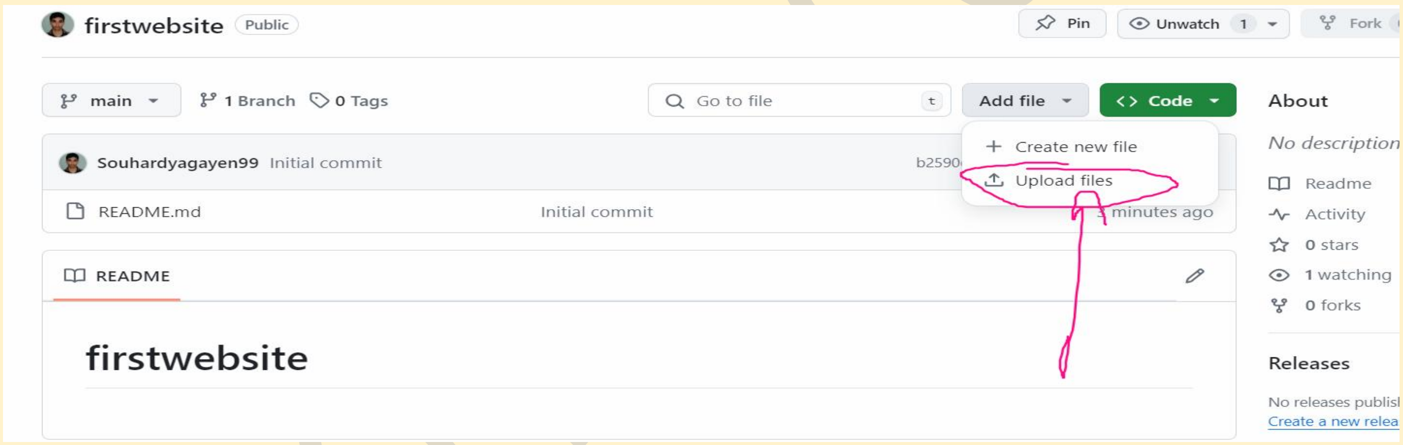
------→ P.T.O

>>TYPES OF UPLOADING FILE INTO GITHUB

- Upload file into GitHub Repositories through "Manual/Direct Drag & Drop Process" OR, "GitHub Desktop" OR, using "Git" you can upload.



Manual Process: It is a Drag & Drop Process It is actually for Small Projects.



Link-
https://www.youtube.com/watch?v=xmK1Q5uzH4w&list=PL0xmYJZV1DwDx8SlT70oug84M3PU1KEwb&index=13

GitHub Desktop: (It is Used for Large Projects)

At first, you have to create new repositories then add readme file then go to there where is your project file present.

Always Here We have to create New folder on Desktop & then clone the Repositories and then copy the code or file from your main original folder location.

Don't give the original file location where you have done your code otherwise it will replace your original files like readme...

Link-
https://www.youtube.com/watch?v=a28WGSNIEZc&list=PL0xmYJZV1DwDx8SlT70oug84M3PU1KEwb&index=8 (see that one)

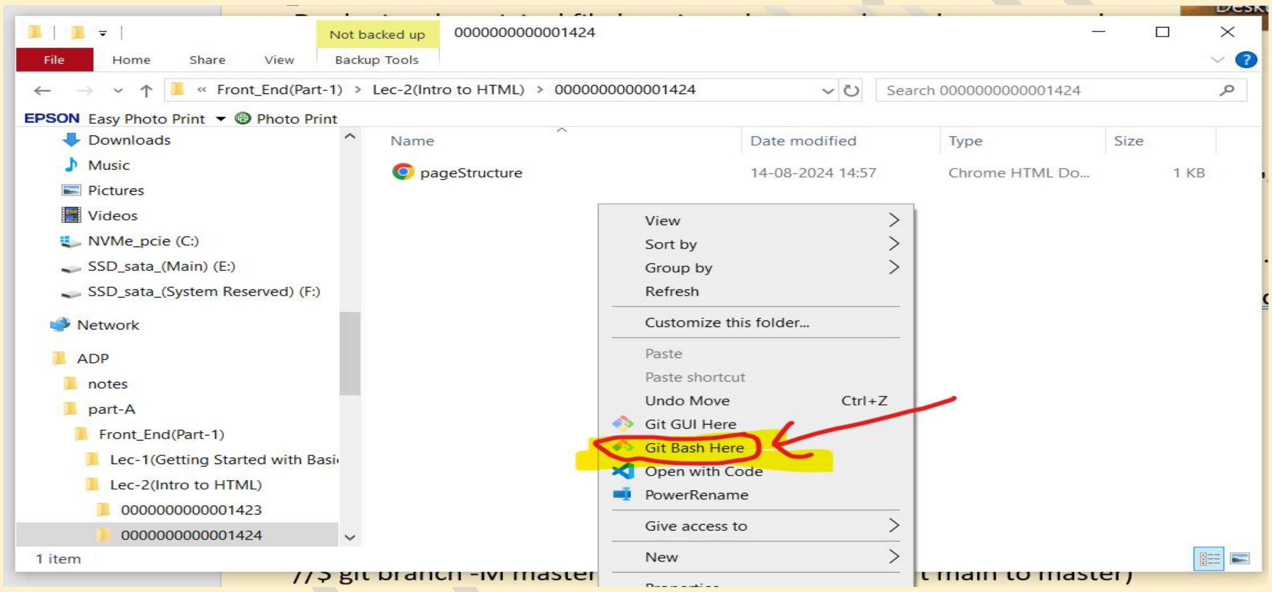
https://www.youtube.com/watch?v=eL7EOQ08-14&list=PL0xmYJZV1DwDx8SlT70oug84M3PU1KEwb&index=15

https://www.youtube.com/watch?v=A2R-CMMyp64&list=PL0xmYJZV1DwDx8SlT70oug84M3PU1KEwb&index=2&t=301s

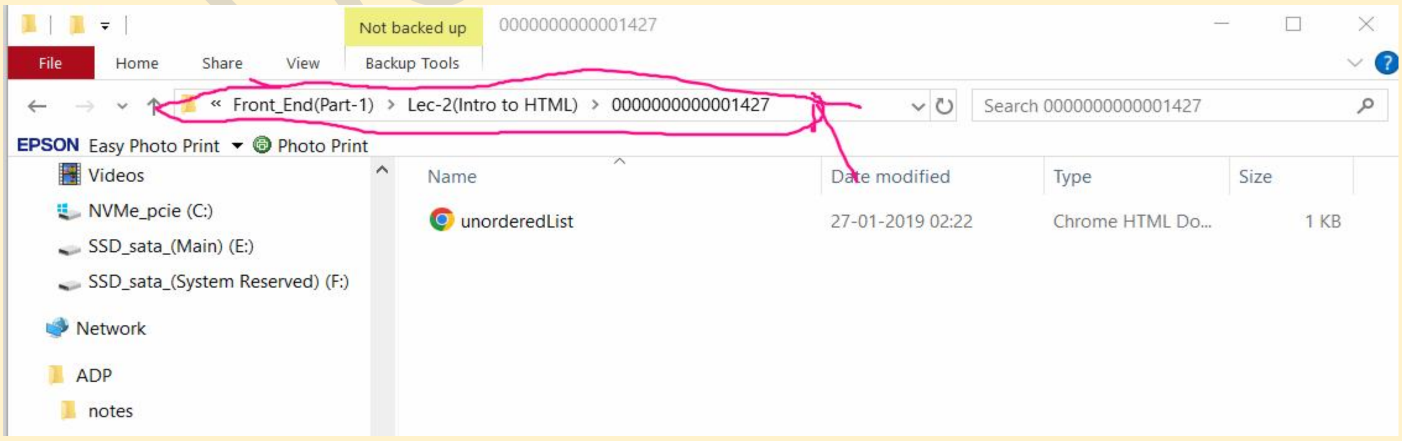
Git: (It is Used for Large Projects)

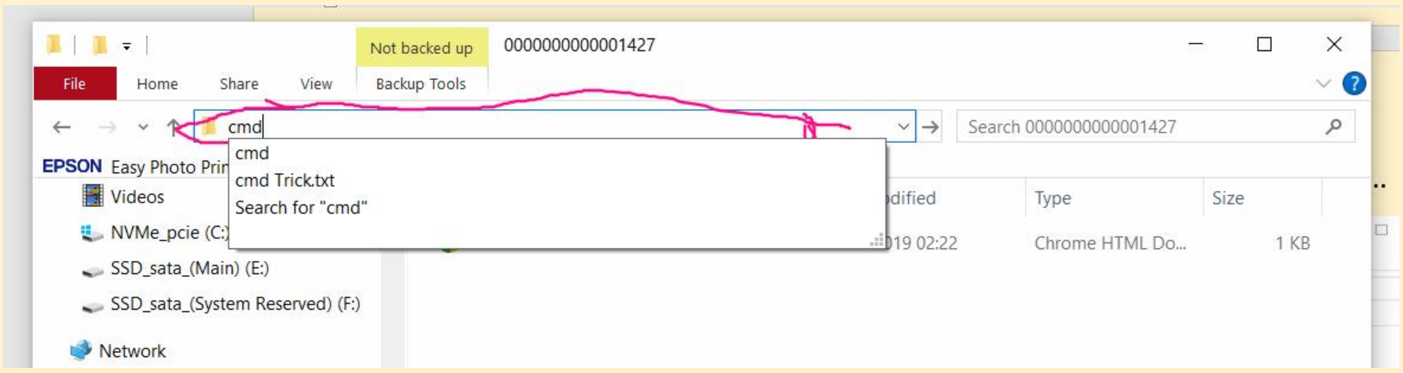
At first, you have to create new repositories then add readme file

Then go to there where is your project file present. Then right click and select "Git Bash here" and continue...

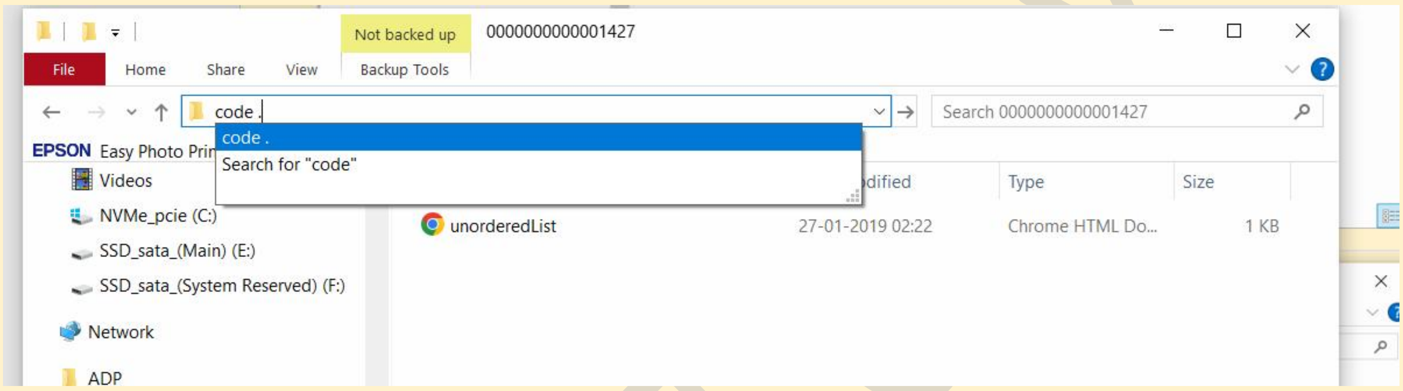


OR, select the address of your project file then type there “cmd” and continue…





OR, For VS Code(VS Code terminal) also select the address of your project file then type there “code .” and continue…



NOTE: Git Bash = CMD = VS Code Terminal (all are the same things)

Now Execute This Commands…

//$ echo "# ProjectChallenge" >> README1.md //(do not need) [it is for create README]

$ git init

$ git status //(to see status)

$ git branch -M main //you need to run(For convert master to main)

//$ git branch -M master //do not need(For convert main to master)

//$ git add README1.md // do not need(it is for add one file) OR

$ git add . // (to add all file)

$ git status

$ git commit -m "message here" // (Insteed of message here you can type any thing)

$ git remote add origin https://github.com/Souhardyagayen99/Online_Exam_System //(you need to copy the https code) OR

$ git remote add origin git@github.com:Souhardyagayen99/Online_Exam_System.git // (you need to copy the ssh code)

$ git status   //(to see status)

//$ git push -u origin main //OR

//$ git push -u origin master //OR

//$ git push origin master //OR

//$ git push origin main //OR

//$ git push origin master --force //OR

$ git push origin main --force // (USE THAT FOR PUSH)

NOTE: Only Run/Execute the Highlighted once's

It's Uploaded...

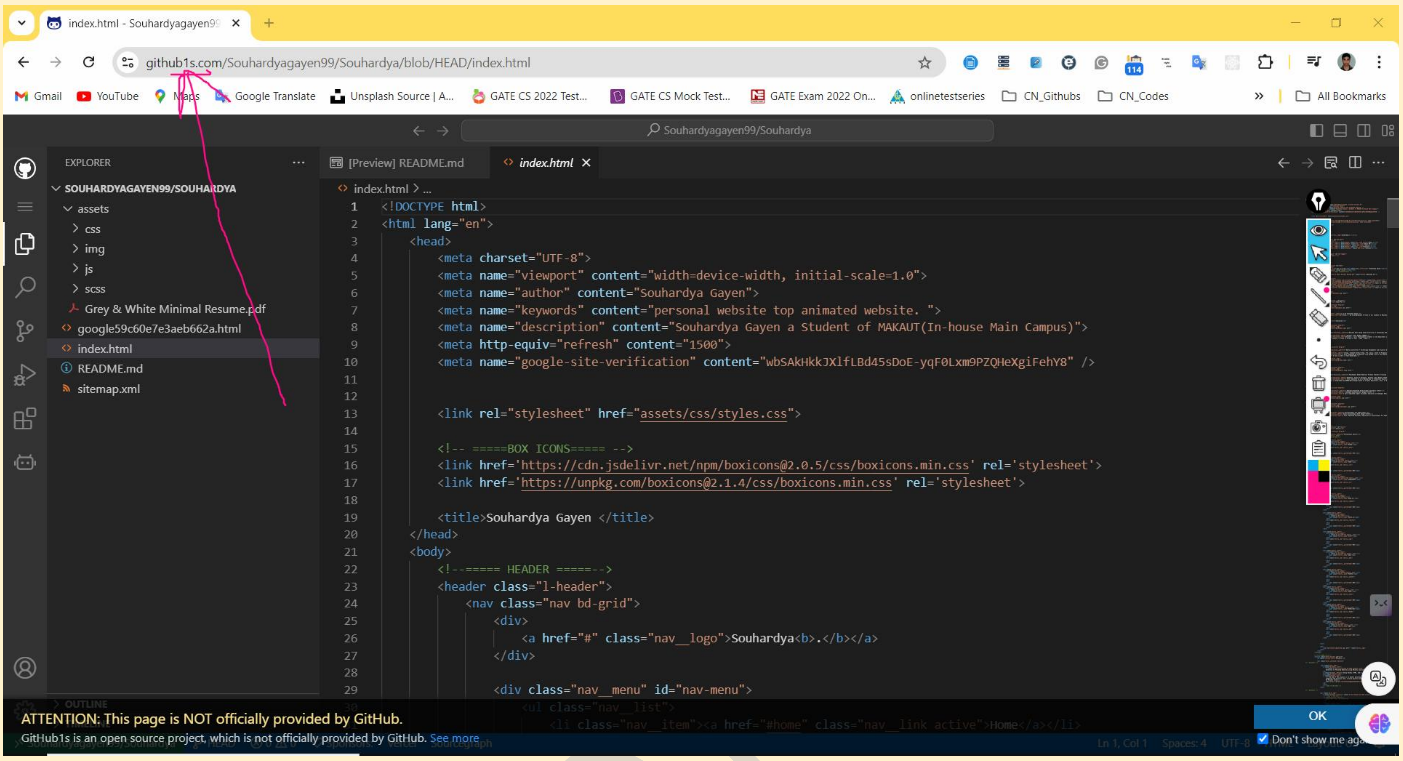
now refresh and see github.

---*****--

------→ P.T.O

>>TRICK(GitHub1s)

GitHub Repository in VS Code View Directly in Browser /OR, GitHub To VS Code Using GitHub1s:



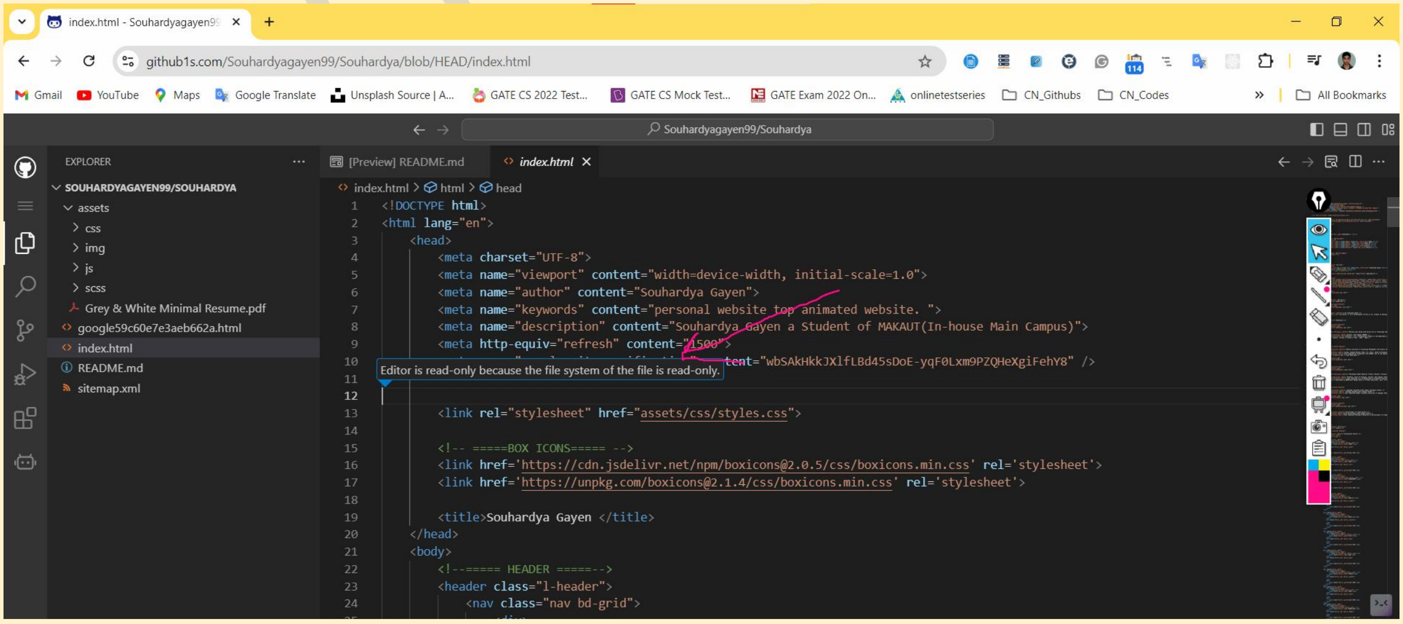
```html
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <meta name="author" content="Souhardya Gayen">
        <meta name="keywords" content="personal website top animated website. ">
        <meta name="description" content="Souhardya Gayen a Student of MAKAUT(In-house Main Campus)">
        <meta http-equiv="refresh" content="1500">
        <meta name="google-site-verification" content="wbSAkHkkJXlfLBd45sDoE-yqF0Lxm9PZQHeXgiFehY8" />


        <link rel="stylesheet" href="assets/css/styles.css">

        <!-- =====BOX ICONS===== -->
        <link href='https://cdn.jsdelivr.net/npm/boxicons@2.0.5/css/boxicons.min.css' rel='stylesheet'>
        <link href='https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css' rel='stylesheet'>

        <title>Souhardya Gayen </title>
    </head>
    <body>
        <!--===== HEADER =====-->
        <header class="l-header">
            <nav class="nav bd-grid">
                <div>
                    <a href="#" class="nav__logo">Souhardya<b>.</b></a>
                </div>

                <div class="nav__menu" id="nav-menu">
                    <ul class="nav__list">
                        <li class="nav__item"><a href="#home" class="nav__link active">Home</a></li>
```

ATTENTION: This page is NOT officially provided by GitHub.
GitHub1s is an open source project, which is not officially provided by GitHub. See more

Choose any of your Repository then Select the address of Github and write '1s' on that address like…

https://github1s.com/Souhardyagayen99/Souhardya/blob/HEAD/index.html

But this is only read only mode we can not write anything here we just see code here like VS Code format.


index.html - Souhardyagayen99
github1s.com/Souhardyagayen99/Souhardya/blob/HEAD/index.html
Souhardyagayen99/Souhardya
EXPLORER
SOUHARDYAGAYEN99/SOUHARDYA
[Preview] README.md
index.html
index.html > html > head

Editor is read-only because the file system of the file is read-only.

```html
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <meta name="author" content="Souhardya Gayen">
        <meta name="keywords" content="personal website top animated website. ">
        <meta name="description" content="Souhardya Gayen a Student of MAKAUT(In-house Main Campus)">
        <meta http-equiv="refresh" content="1500">
        ...tent="wbSAkHkkJXlfLBd45sDoE-yqF0Lxm9PZQHeXgiFehY8" />


        <link rel="stylesheet" href="assets/css/styles.css">

        <!-- =====BOX ICONS===== -->
        <link href='https://cdn.jsdelivr.net/npm/boxicons@2.0.5/css/boxicons.min.css' rel='stylesheet'>
        <link href='https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css' rel='stylesheet'>

        <title>Souhardya Gayen </title>
    </head>
    <body>
        <!--===== HEADER =====-->
        <header class="l-header">
            <nav class="nav bd-grid">
```

To Work on Multiple GitHub Branches...

```
$ git branch -M master //for create & switch on new branch OR,
```

```
$ git checkout -b master //for create new branch OR,
```

```
$ git branch master //for create new branch
```

Then add a folder/file in that branch using

```
$ git add . // (to add all files)
```

and for commit

```
$ git commit -m "message here"
```

and then push that folder/file to that repository using

```
$ git push origin feature-branch --force
```

To change the branch on GitHub repository

```
$ git branch -M master // for create & switch on new branch OR,
```

```
$ git checkout master
```

To merge the branches (we need to have the same no of content & all file name should be same)

```
$ git merge master // OR,
```

```
$ git merge --squash master
```

For Deleting a Branch

```
//$ git branch -d master // OR, (it is not working)
```

```
$ git push origin --delete master // (used this one)
```

git pull: This command is a combination of two Git commands: git fetch and git merge.

```
$ git pull origin main
```

A lot of things are here to explore….

GitHub will be Continue…..

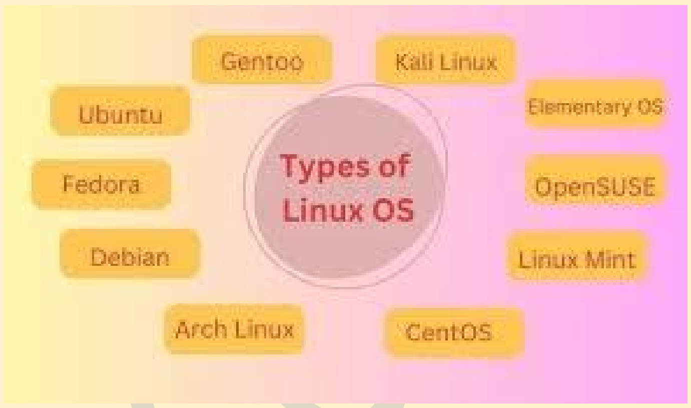
------→ P.T.O

Linux Basic Commands

What is Linux?

Linux is a free, open-source operating system that is similar to Windows but has many different versions. It is based on the Linux kernel, which was first released in 1991 by Linus Torvalds, a computer science student at the University of Helsinki.

- Linux comes in many varieties, which are called distributions: Ubuntu, Fedora, Debian, CentOS, and Arch Linux. Even android is a Linux distribution or distros.



Why We Should Use Linux?

Linux is a versatile operating system that can be used for many reasons, including:

1. Security

Linux has strong security features, like a permissions system that lets users control who can access their files.

2. Source

Linux is open-source software, which means that anyone can access and modify the source code. This allows for a large community of developers to constantly improve the operating system.

3. Customization

Linux can be tailored to meet the needs of different users, from home users to large corporations.

4. Education

Linux can be useful for students who want to learn how an operating system works, and can modify and extend the code to suit their needs.

5. Compatibility

Linux is compatible with almost every hardware device, making it a good option for repurposing older computers.

6. Updates

Most Linux distributions receive frequent and stable updates that include new features and security updates.

7. Automation

Automation can help minimize costs by reducing manual operations and accelerating deployments for cloud infrastructures.

------→ P.T.O

Using commands in the Terminal

1. Navigating the File System

- pwd (Print Working Directory):
  Displays the full path to the current directory you're in.
  Command: `pwd`

- ls (List):
  Lists the files and directories in the current directory.
  Command: `ls`
  - `ls -l`: Long format listing (shows detailed file information).
  - `ls -a`: Lists all files, including hidden files (those starting with a dot `.`).

- cd (Change Directory):
  Changes the current directory to another directory.
  Command: `cd /path/to/directory`
  - `cd ..`: Moves up one directory level.
  - `cd ~`: Moves to the home directory.

2. Managing Files and Directories

- touch:
  Creates a new empty file or updates the timestamp of an existing file.
  Command: `touch filename.txt`

- mkdir (Make Directory):
  Creates a new directory.
  Command: `mkdir new_directory`

- cp (Copy):
  Copies files or directories.
  Command: `cp source_file destination_file`
  - `cp -r source_directory destination_directory`: Recursively copies a directory.

- mv (Move/Rename):
  Moves or renames files or directories.
  Command: `mv old_name new_name`
  - Can also be used to move files to a different directory: `mv file.txt /path/to/destination/`

- rm (Remove):
  Deletes files or directories.
  Command: `rm filename.txt`
  - `rm -r directory_name`: Recursively deletes a directory and its contents.
  - **Caution:** `rm` is irreversible. Be careful, especially when using `rm -r`!

3. Viewing and Editing Files

- cat (Concatenate):
  Displays the content of a file.
  Command: `cat filename.txt`

- more and less:
  View file contents one page at a time.
  Command: `less filename.txt`
  - `more` is similar but less feature-rich than `less`.

- head and tail:
  Display the beginning or end of a file, respectively.
  Command: `head filename.txt` and `tail filename.txt`
  - `head -n 10 filename.txt`: Shows the first 10 lines of the file.
  - `tail -n 10 filename.txt`: Shows the last 10 lines of the file.

- nano, vim, vi:
  Text editors to view or edit files directly in the terminal.
  Command: `nano filename.txt`, `vim filename.txt`, or `vi filename.txt`

4. File Permissions

- chmod (Change Mode):
  Changes file or directory permissions.
  Command: `chmod 755 filename.sh`
  - `755` is a common permission setting (read, write, and execute for the owner; read and execute for others).

- chown (Change Ownership):
  Changes the ownership of a file or directory.
  Command: `chown user:group filename.txt`

5. System Information

- uname:
  Displays system information.
  Command: `uname -a`

- df (Disk Free):
  Shows disk space usage of file systems.
  Command: `df -h`

- du (Disk Usage):
  Shows the disk usage of files and directories.
  Command: `du -sh *`

- top:
  Displays the currently running processes.
  Command: `top`

- ps:
  Lists running processes.
  Command: `ps aux`

6. Networking

- ifconfig or ip addr:
  Displays network interfaces and IP addresses.
  Command: `ifconfig` or `ip addr`

- ping:
  Checks the connectivity to a remote host.
  Command: `ping google.com`

- wget:
  Downloads files from the internet.
  Command: `wget http://example.com/file.zip`

7. Process Management

- kill:
  Kills a process by its process ID (PID).
  Command: `kill 1234`

- killall:
  Kills all processes by name.
  Command: `killall processname`

8. Searching

- find:
  Searches for files and directories.
  Command: `find /path -name filename.txt`

- grep:
  Searches for patterns in files.
  Command: `grep "search_term" filename.txt`

9. Compression

- tar:
  Archives and compresses files.

Command: `tar -cvf archive.tar file1 file2` and `tar -xvf archive.tar`
- `tar -czvf archive.tar.gz directory`: Compresses a directory into a `.tar.gz` file.

- zip and unzip:
Compresses and extracts `.zip` files.
Command: `zip -r archive.zip directory` and `unzip archive.zip`

10. Miscellaneous

- echo:
Prints text to the terminal.
Command: `echo "Hello, World!"`

- history:
Displays the command history.
Command: `history`

- clear:
Clears the terminal screen.
Command: `clear`

Online Linux OS- https://cocalc.com/

Handbook Link- https://bjpcjp.github.io/pdfs/devops/linux-commands-handbook.pdf

website link- https://www.hostbillo.com/blog/50-linux-commands-with-screenshots/

Video link1- https://www.youtube.com/watch?v=IVquJh3DXUA

Video link2- https://www.youtube.com/watch?v=_tCY-c-sPZc&t=574s