AGILE ASSIGNMENT

- **WHAT IS GIT?**

Git is a version control system used by developers to manage and track changes in source code during software development. Here's a simple explanation:

What Git Does:

Tracks Changes: Git records every change made to your files so you can go back to previous versions.

Collaborates: Multiple people can work on the same project at the same time without interfering with each other.

Branches and Merges: You can create separate branches to work on new features or fixes, then merge them back into the main code.

- **What is git repository?**

A Git repository (often just called a repo) is a directory that stores your project's files along with all the version history tracked by Git.

There are two types of Git repositories:

1.Local Repository:

This is on your computer. It includes:

Your working files (source code, etc.)

A hidden .git folder that contains all the version history and configuration for Git.

2.Remote Repository:

This is typically hosted online (e.g., on GitHub, GitLab, or Bitbucket).

Step:1-Create a local git repository-

1.Open terminal and command prompt.

2.Navigate to your project folder.

3.Example:

```bash
cd path/to/your/project
```

- **Initialize the Git repository**

```bash
git init
```

Step 2: Add and Commit Your Files

1.Add all files to the staging area

```bash
git add .
```

2.Commit the files

Bash Copy code

```bash
git commit -m "Initial commit"
```

Step 3: Create a Remote Repository on GitHub

Go to https://gitHub.com

Click “New repository “

Give your repo a name, choose public or private., click Create repository

GitHub will then show you a URL like:

arduino Copy Edit

```arduino
https://github.com/your-username/your-repo-name.git
```

Step 4:Connect local Repo to Git

Back in your terminal:

Step 4: Connect Local Repo to GitHub

Back in your terminal:

bash Copy Edit

```bash
git remote add origin https://github.com/your-username/your-repo-name.git
```

Step 5: Push your code to GitHub

🚀 **Step 5: Push Your Code to GitHub**

bash Copy Edit

```bash
git push -u origin master
```

> If you're using a newer Git version, replace `master` with `main`:

bash Copy Edit

```bash
git push -u origin main
```

- **3. Git command for Clone:**

bash Copy Edit

```bash
git clone <repository-url>
```

Example:

bash Copy Edit

```bash
git clone https://github.com/username/repository-name.git
```

This command:

Downloads a full copy of the remote repository (including all files and history)

Creates a folder named after the repository

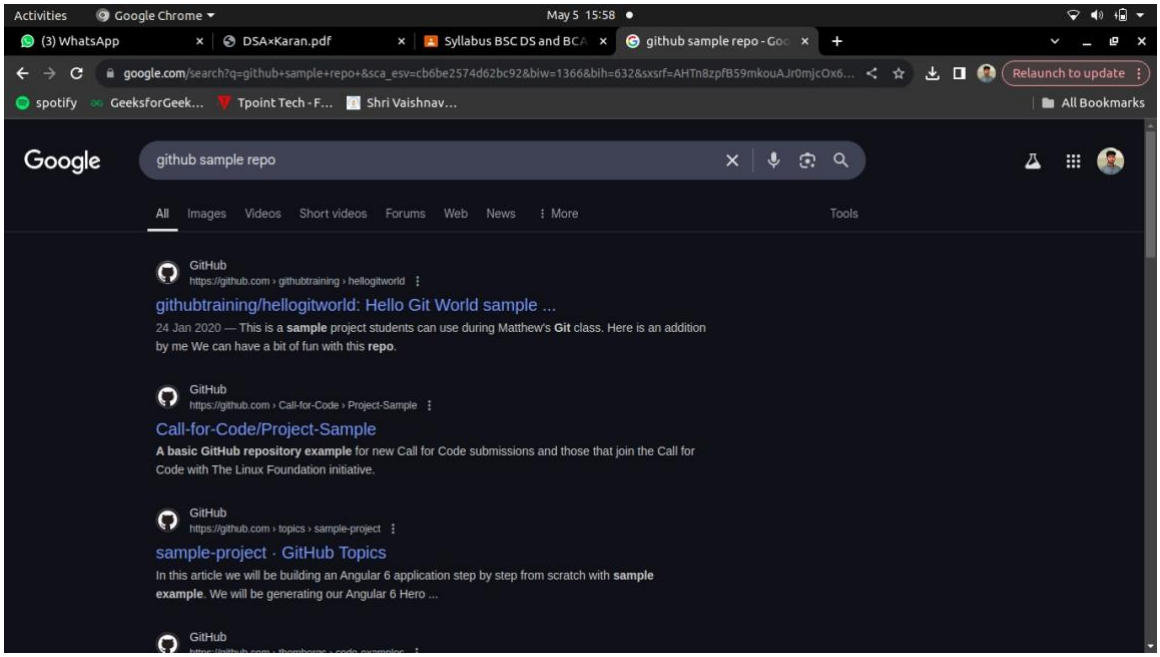
Automatically sets the remote URL as origin

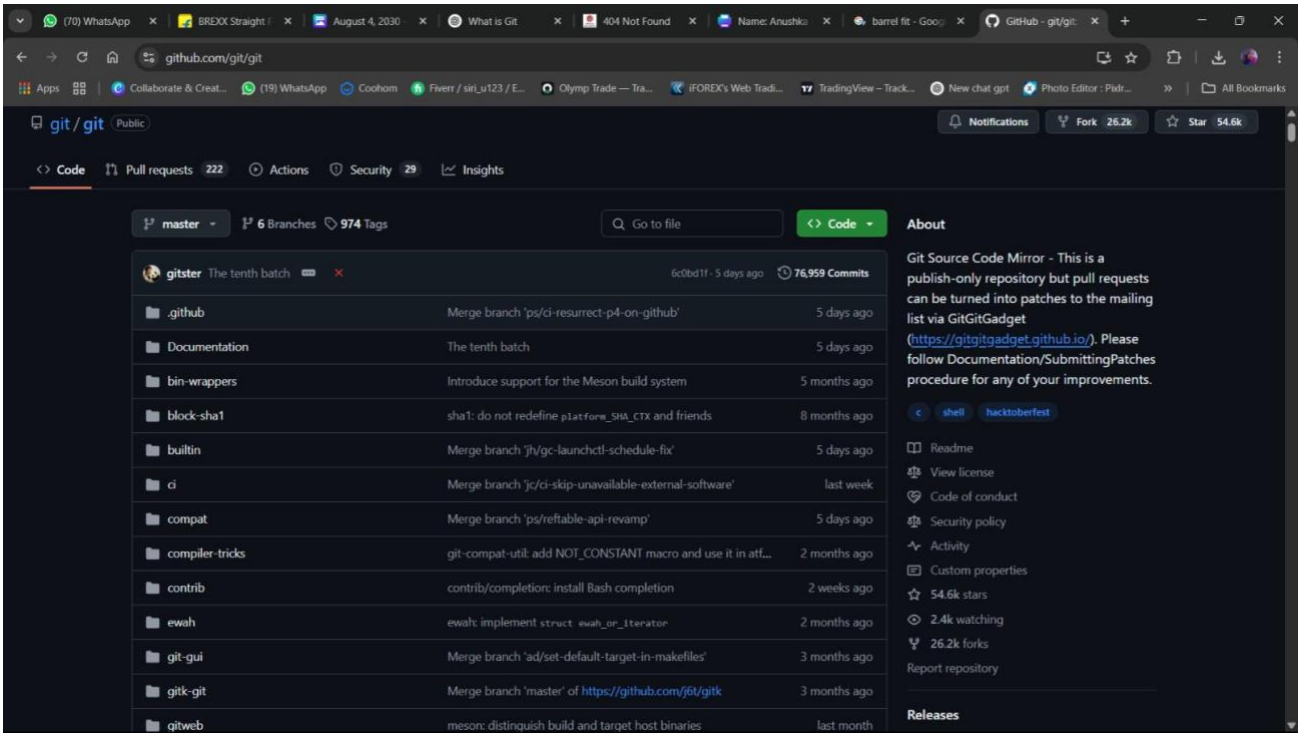
Optional: Clone into a specific folder name

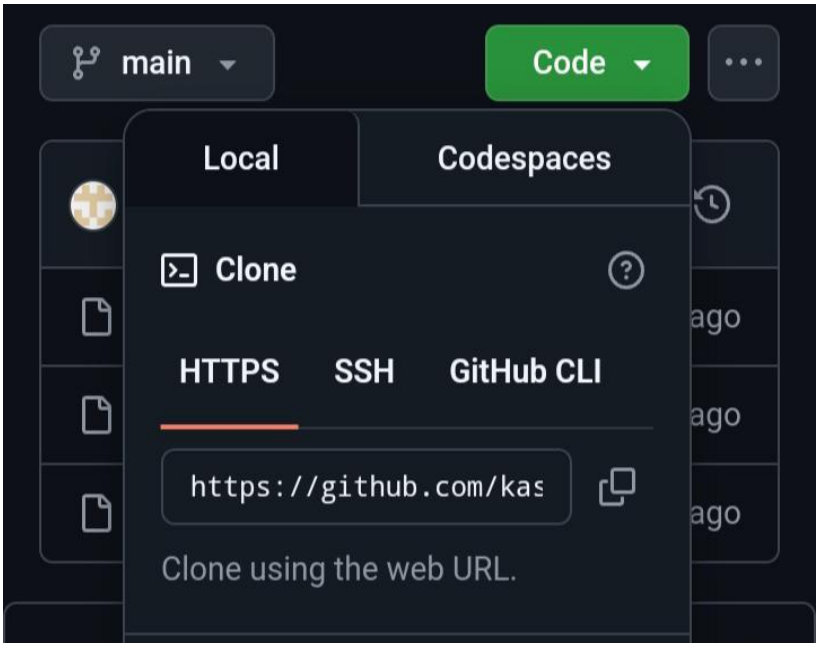
Optional: Clone into a specific folder name

bash Copy Edit

```bash
git clone https://github.com/username/repository-name.git my-folder-name
```

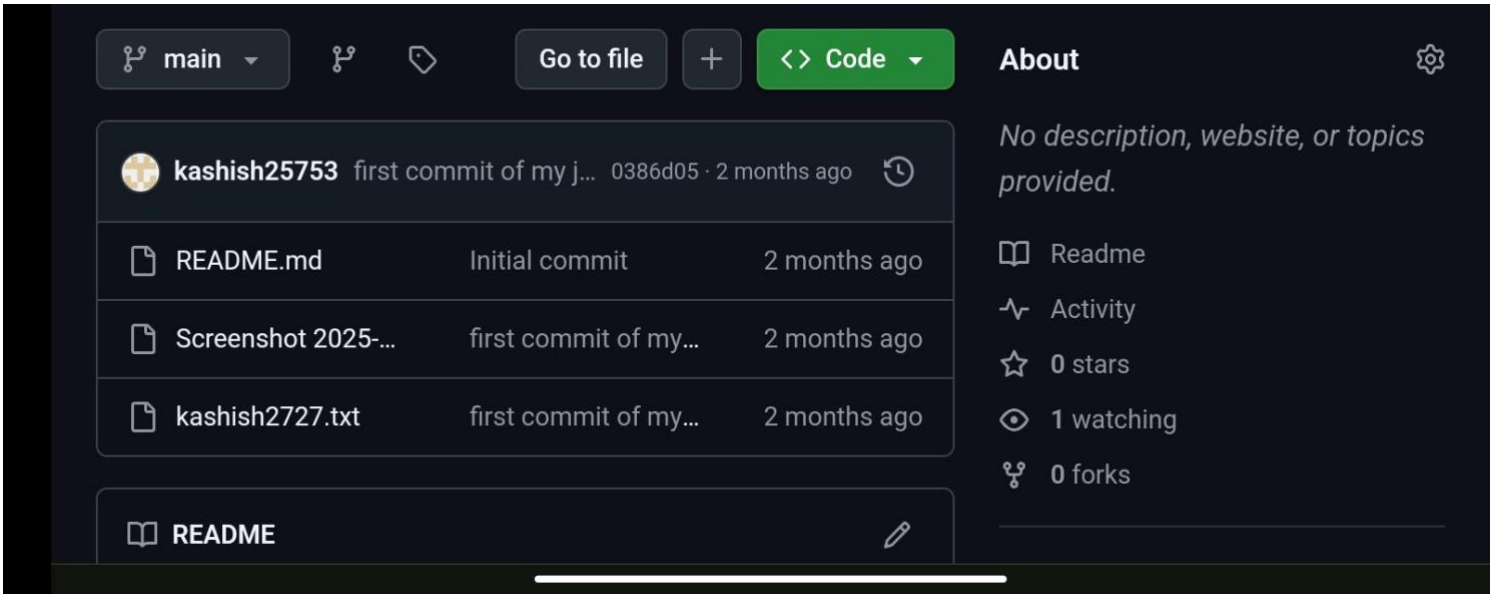





main
Code
Local
Codespaces
Clone
HTTPS
SSH
GitHub CLI
https://github.com/kas
Clone using the web URL.
ago
ago
ago


Repositories
kashish25753/MyFirstJava
kashish25753/Harshit-
kashish25753/MyFirstJavaProject


README
MyFirstJava



4. Git command for uploading a file

- **Git installed**
- **GitHub account**
- **GitHub repository created**
- **Your system is created to remote GitHub Repo(git remote repo origin…)**

STEP BY STEP -upload file to GitHub

>Step 1:Open terminal git /bash

Navigate to your project folder:

```bash
cd path/to/your/project
```

>Step 2: Initialize Git(if not already done)

```bash
git init
```

>Step 3:Add remote Repository (once)

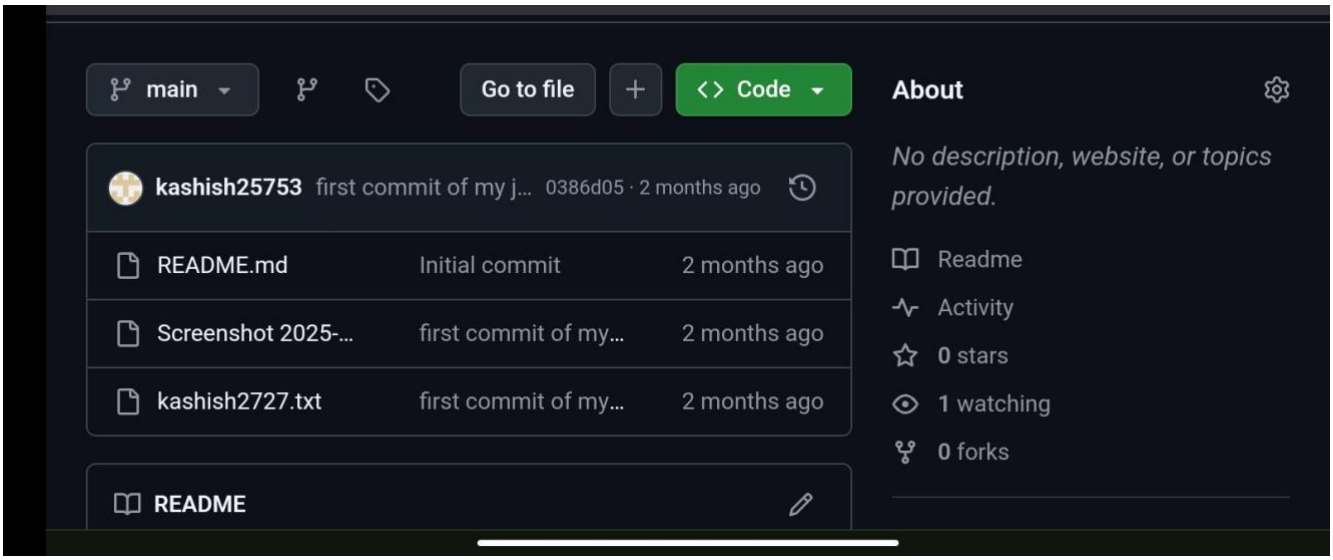
```bash
git remote add origin
https://github.com/yourusername/your
-repo.git
```

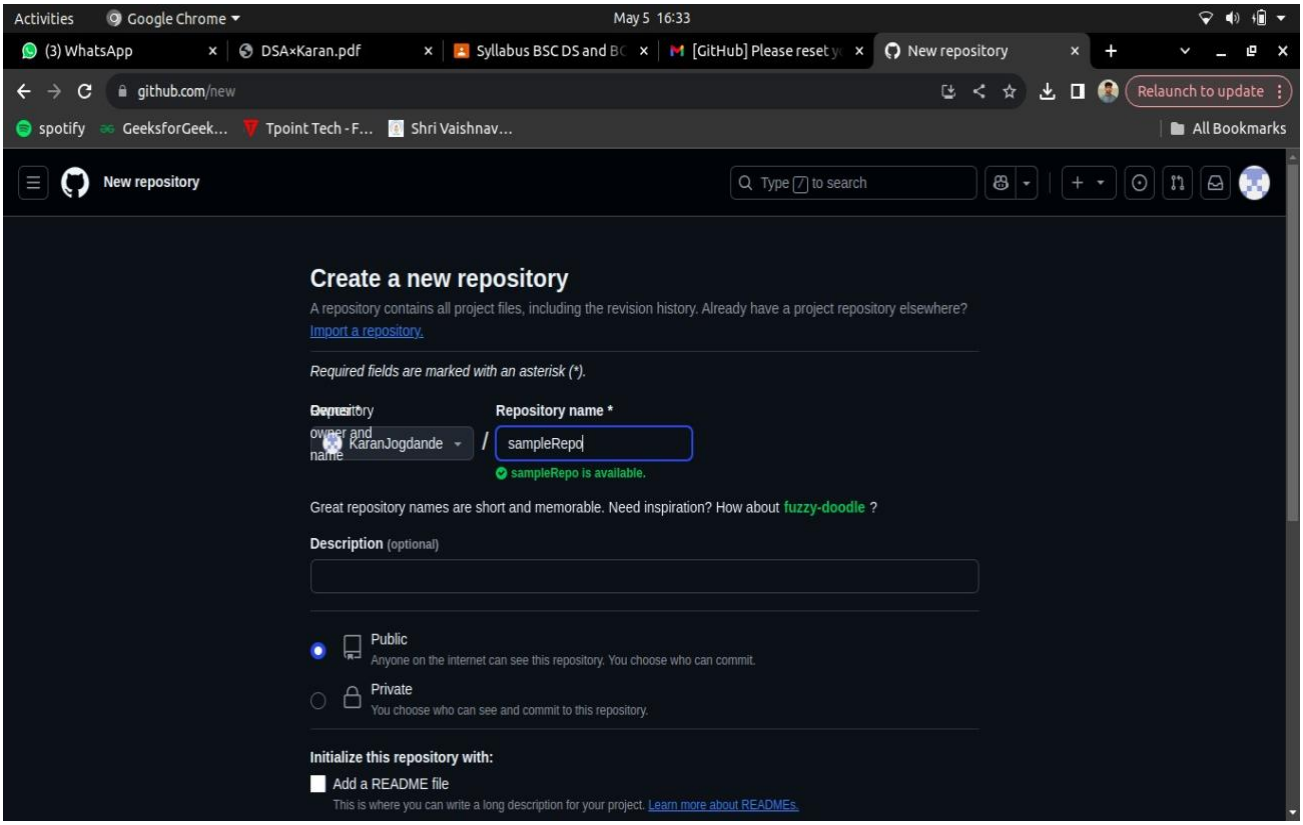
>Step 4: Add your file

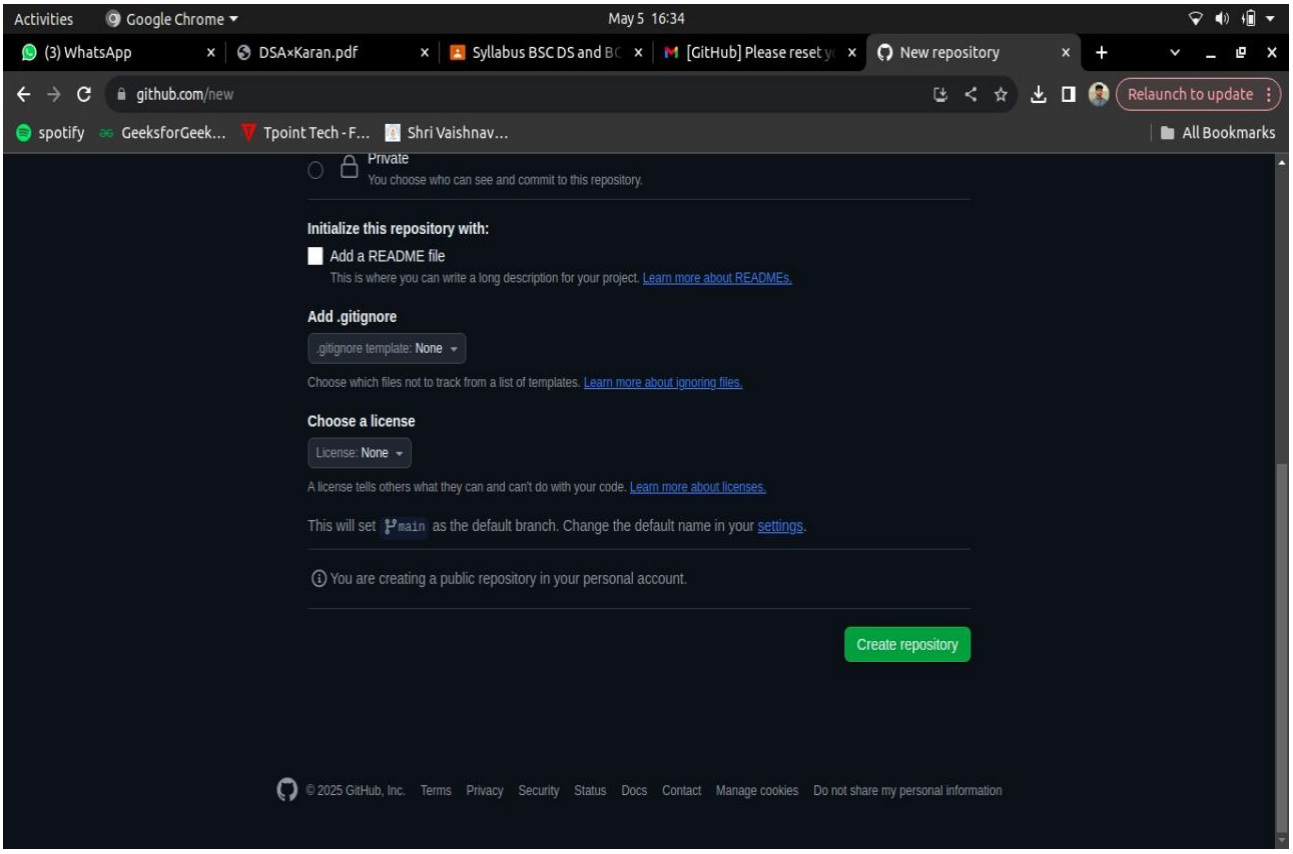
```bash
git add filename.ext
```

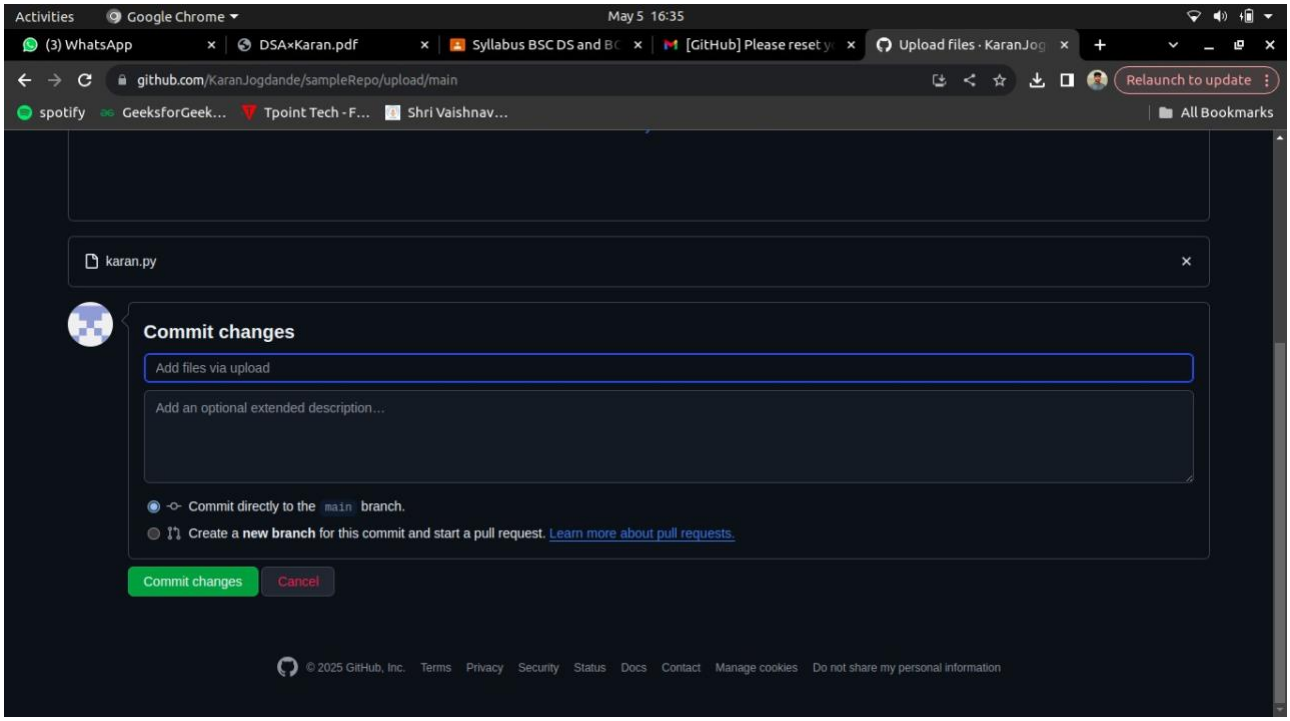
>Step 5:Commit your changes

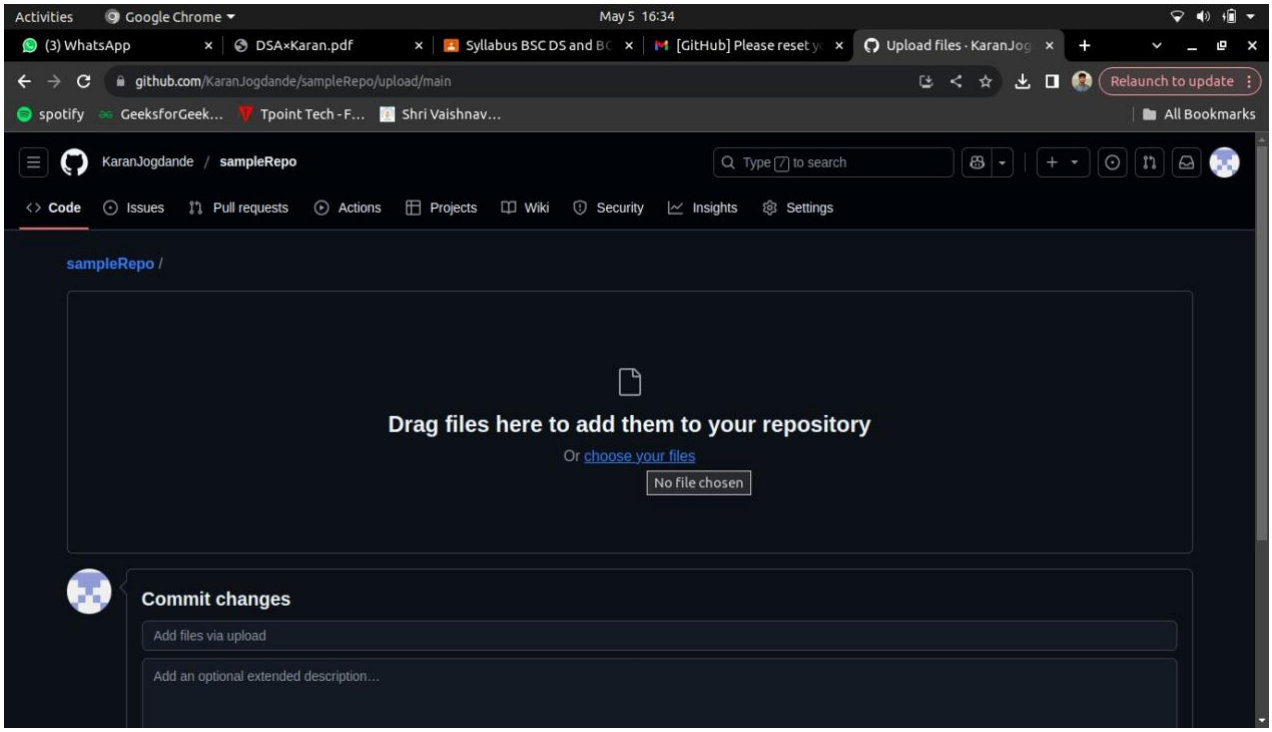
```bash
git commit -m "Add new file: your-new-file.ext"
```

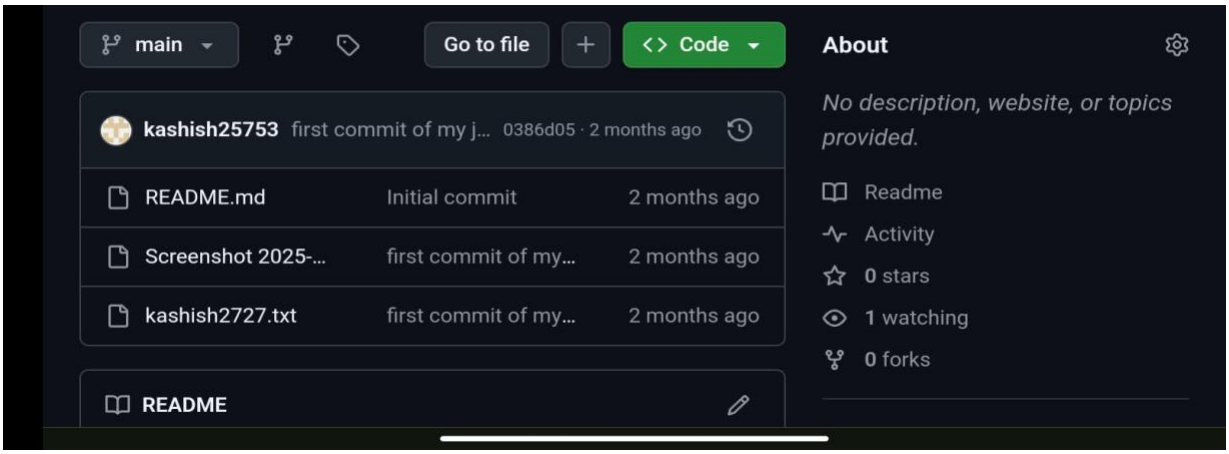



Activities
Google Chrome
May 5 16:33
(3) WhatsApp
DSA×Karan.pdf
Syllabus BSC DS and B
[GitHub] Please reset y
New repository
github.com/new
Relaunch to update
spotify
GeeksforGeek...
Tpoint Tech - F...
Shri Vaishnav...
All Bookmarks
New repository
Type / to search
Create a new repository
A repository contains all project files, including the revision history. Already have a project repository elsewhere?
Import a repository.
Required fields are marked with an asterisk (*).
Repository owner and name
KaranJogdande
Repository name *
sampleRepo
sampleRepo is available.
Great repository names are short and memorable. Need inspiration? How about fuzzy-doodle ?
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.

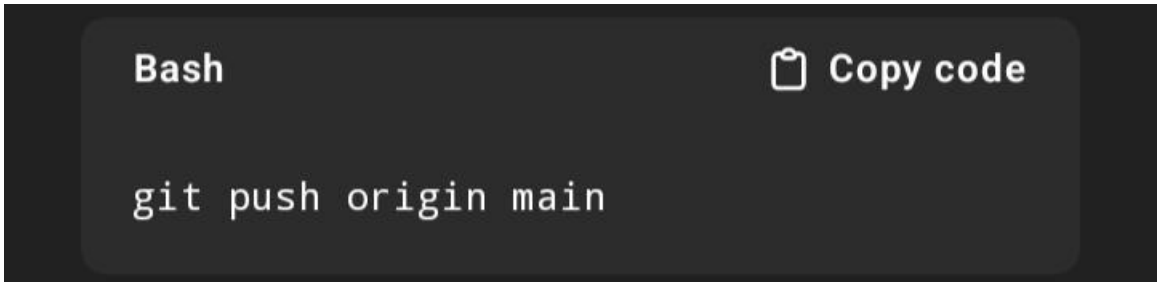

Activities
Google Chrome
May 5 16:34
(3) WhatsApp
DSA×Karan.pdf
Syllabus BSC DS and B
[GitHub] Please reset y
New repository
github.com/new
Relaunch to update
spotify
GeeksforGeek...
Tpoint Tech - F...
Shri Vaishnav...
All Bookmarks
Private
You choose who can see and commit to this repository.
Initialize 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
A license tells others what they can and can't do with your code. Learn more about licenses.
This will set main as the default branch. Change the default name in your settings.
You are creating a public repository in your personal account.
Create repository
© 2025 GitHub, Inc.
Terms
Privacy
Security
Status
Docs
Contact
Manage cookies
Do not share my personal information


Activities
Google Chrome
May 5 16:35
(3) WhatsApp
DSA×Karan.pdf
Syllabus BSC DS and B
[GitHub] Please reset y
Upload files · KaranJog
github.com/KaranJogdande/sampleRepo/upload/main
Relaunch to update
spotify
GeeksforGeek...
Tpoint Tech - F...
Shri Vaishnav...
All Bookmarks
karan.py
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
© 2025 GitHub, Inc. Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information


Activities
Google Chrome
May 5 16:34
(3) WhatsApp
DSA×Karan.pdf
Syllabus BSC DS and B
[GitHub] Please reset y
Upload files · KaranJog
github.com/KaranJogdande/sampleRepo/upload/main
Relaunch to update
spotify
GeeksforGeek...
Tpoint Tech - F...
Shri Vaishnav...
All Bookmarks
KaranJogdande / sampleRepo
Type / to search
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
sampleRepo /
Drag files here to add them to your repository
Or choose your files
No file chosen
Commit changes
Add files via upload
Add an optional extended description...



>Step 6:Push to GitHub



- **What is docker?**

Docker is an open-source platform that allows you to build, run, and

manage applications in lightweight, portable containers.

>Portability: Write code once and run it anywhere — on your PC, on a server, or in the cloud.

>Consistency: Avoid the "it works on my machine" problem.

>Isolation: Apps in different containers don’t interfere with each other.

>Efficiency: Uses less memory and starts faster than virtual machines.

🚀 **How Docker Works (In Simple Terms):**

1. You **write a Dockerfile** that describes your app and its environment.
2. You **build an image** from that Dockerfile using:

Bash Copy code

```bash
docker build -t my-app .
```

3. You **run a container** from that image using:

Bash Copy code

```bash
docker run my-app
```

- Commands:

Command	Description
docker build .	Builds an image from the current directory
docker images	Lists available images
docker run image-name	Runs a container from the image
docker ps	Lists running containers
docker stop container-id	Stops a running container

- **What is container?**

A container is a lightweight, standalone, and executable package of software that includes:

Application code
Runtime (e.g., Python, Node.js, Java)
Libraries and dependencies
System tools/configuration
Everything the application needs to run consistently across different environments is

Bundled inside the container.

- **Key features of container:**

•**Isolation**: Each container runs in its own environment, separate from other containers or the host System.

•**Portability**: Since all dependencies are included, a container can run anywhere — on your laptop, a Server, or in folder cloud — and it will behave the same.

•**Efficiency**: Containers share the host system's operating system kernel, so they are more efficient and Faster than virtual machines.

•**Lightweight**: Containers are smaller in size and start quickly.

Step-by-Step: Create Docker Image & Container

Step 1: Set up your project folder

Create a folder and move into it:

Create a folder and move into it:

Bash Copy code

```bash
mkdir my-docker-app
cd my-docker-app
```

Step 2: Create a Python app file

Create a file named app.py:

Create a file named app.py:

Python Copy code

```python
# app.py
print("Hello from Docker!")
```

Step 3: Create a Dockerfile

Create a file named Dockerfile (no extension) in the same folder:

Create a file named `Dockerfile` (no extension) in the same folder:

Dockerfile Copy code

```
# Use an official Python base image
FROM python:3.9

# Set working directory
WORKDIR /app

# Copy the current directory contents
into the container
COPY . .

# Set the default command to run the app
CMD ["python", "app.py"]
```

Step 4: Build the Docker Image

Run this command in the terminal from the project folder:

Run this command in the terminal from the project folder:

Bash Copy code

```
docker build -t my-python-app .
```

Step 5: Run a Container from the Image

Now run a container:

Now run a container:

Bash Copy code

```bash
docker run my-python-app
```

You should see the output:

Copy code

```
Hello from Docker!
```

•How to Create a Docker Image & Container Using GUI (Docker Desktop)

Step-by-Step (with GUI):

Step 1: Prepare Your App

Let's assume you have a folder named my-python-app with two files:

app.py (a simple Python script)

Dockerfile (contains instructions to build the image)

Example app.py

Python Copy code

```python
print("Hello from Docker GUI!")
```

Example Dockerfile

Dockerfile Copy code

```dockerfile
FROM python:3.9
WORKDIR /app
COPY . .
CMD ["python", "app.py"]
```

Step 2:Open Docker desktop with GUI

1.Launch Docker Desktop

2.Go to the “Containers” tab (you’ll manage running containers here)

3.Switch to the “Images” tab to manage and build images

Step 3: Build the Docker Image via GUI

1.Open the “Images” tab

2.Click “Build from Dockerfile”

3.Select the folder that contains your Dockerfile (my-python-app)

4.Name your image (e.g., my-python-app)

5.Click Build

Step 4: Run the Image as a Container

1.Go to the “Images” tab

2.Find your image (my-python-app)

3.Click “Run”

4.Optionally:

5.Set a container name

6.Configure ports, volumes