

## Experiment-1

**AIM:** Implement weather modeling\* using the quadratic solution in stages: hard-coding variables keyboard input, read from a file, for a single set of input, multiple sets of inputs. a. save all versions, debug, fix problems, create a Github account

### **What is GitHub?**

Ans:

- GitHub is a web-based platform for version control and collaborative software development using Git.
- It facilitates storing, sharing, and managing code repositories online.
- GitHub is widely used by individuals and organizations for open-source and private projects.
- GitHub facilitates a wide range of services in managing the code versions using version control systems which use git while working in command prompt.
- Some of the cmd's are: push , commit , pull , add , etc
- Each of the git commands has its own use in managing the code in the GitHub.

### **Introduction to Github**

Ans:

1. Foundation and Evolution:
  - a. GitHub was founded in 2008 by Tom Preston-Werner, Chris Wanstrath, PJ Hyett, and Scott Chacon.
  - b. It quickly became a leading platform for collaborative software development.
  - c. In 2018, Microsoft acquired GitHub, further enhancing its enterprise capabilities.
2. Purpose:
  - a. Designed as a platform for managing code repositories using Git, a distributed version control system.
  - b. Facilitates collaboration between developers, allowing for efficient code sharing, review, and integration.
3. Community and Ecosystem:
  - a. Hosts one of the largest developer communities globally, with millions of public and private repositories.
  - b. Offers a platform for open-source contributions, fostering innovation and shared learning.

4. Collaborative Environment:
  - a. Developers can work together on projects, regardless of location, through features like branches, pull requests, and issues.
  - b. Provides transparency in project workflows, enabling teams to track progress and contributions.
5. Integrated Tools and Features:
  - a. Combines Git's version control capabilities with additional features like wikis, CI/CD tools, and issue tracking.
  - b. Offers built-in automation tools through GitHub Actions for seamless workflow management.
6. Learning and Resources:
  - a. Offers GitHub Pages for hosting documentation and project showcases.
  - b. Provides educational resources like GitHub Campus and Codespaces for students and educators.
7. Accessibility:
  - a. Accessible via a web interface, command-line tools, and APIs for developers.
  - b. Integration with popular IDEs and text editors like Visual Studio Code enhances usability.
8. Business and Enterprise:
  - a. GitHub Enterprise offers advanced security, compliance, and team management tools for organizations.
  - b. Supports private repositories and secure development practices for enterprise-level projects.
9. Scalability:
  - a. Suitable for projects of all sizes, from personal projects to large-scale enterprise applications.
  - b. Allows seamless scalability for repositories, contributors, and integrations.

## **Some Key Features of GitHub**

1. Version Control
  - a. Tracks changes in code over time using Git.
  - b. Allows developers to manage and revert changes effectively.
2. Repositories
  - a. Centralized storage for code, documentation, and resources.

- b. Supports public and private repositories for secure or open collaboration.
- 3. Branches
  - a. Enables multiple developers to work on different features or fixes simultaneously.
  - b. Supports merging and conflict resolution for seamless integration into the main branch.
  - c. Can use multiple branches and various user's can work on various branches
  - d. At the end the branches can be combined into one for production purpose.
- 4. Pull Requests
  - a. Facilitates code reviews by allowing contributors to propose changes.
  - b. Includes discussion threads, inline comments, and approval mechanisms.
- 5. Issues and Project Management
  - a. Provides a system to track bugs, tasks, and feature requests.
  - b. Offers tools like labels, milestones, and project boards for organizing workflows.
- 6. Actions and Automation
  - a. Automates tasks like testing, building, and deploying code with GitHub Actions.
- 7. Documentation
  - a. Allows adding README files, wikis, and GitHub Pages for detailed project documentation.
- 8. Integration Ecosystem
  - a. Supports seamless integration with CI/CD tools, IDEs, and cloud services.
  - b. Includes GitHub Marketplace for adding additional features and tools.
- 9. Community Engagement
  - a. Features like starring, forking, and watching repositories encourage interaction.
  - b. Allows for open collaboration on public repositories.
- 10. Hosting and Scalability
  - a. Offers reliable hosting for repositories with global accessibility.
  - b. Scales to accommodate projects of any size, from individuals to enterprises.

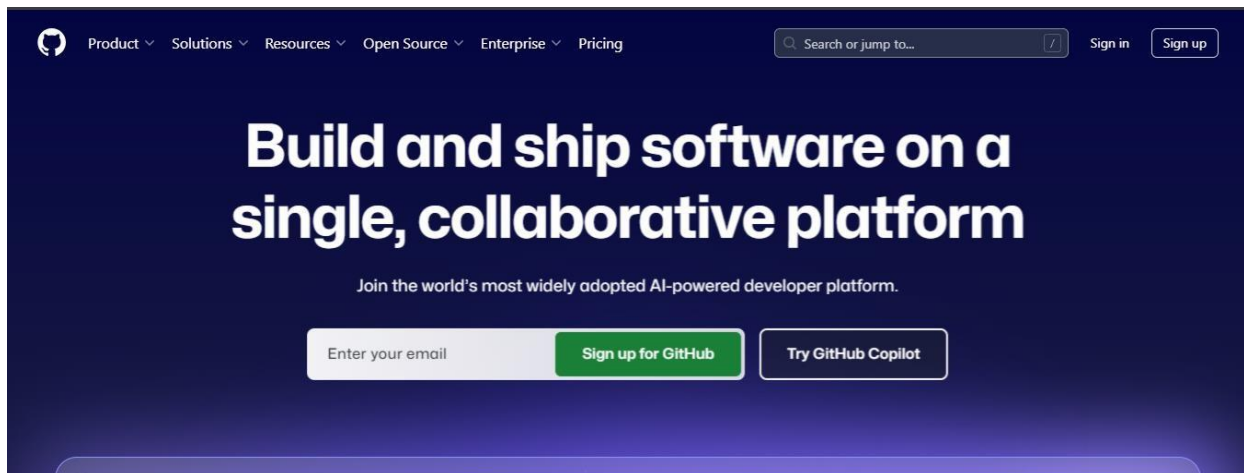
- 11.       GitHub Pages
  - a. Enables users to create static websites directly from a repository.
- 12.       Enterprise Features
  - a. Provides advanced tools for team and enterprise management, including GitHub Enterprise.

### How Git and GitHub are different

<b>GIT</b>	<b>GITHUB</b>
A distributed version control system for tracking changes in source code.	A web-based platform for hosting Git repositories and enabling collaboration
Command line tool	Web-based hosting service.
Manages version control locally or on any server.	Provides a centralized location for managing and sharing Git repositories.
Facilitates collaboration via cloning and merging repositories manually	Enhances collaboration with features like pull requests, issues, and discussions.
Does not host repositories; it works locally or with any server	Hosts repositories online, supporting both public and private repositories
Tracks changes and manages branches locally.	Uses Git for version control, providing a platform for centralized collaboration
Works with any code editor or development environment.	Offers integrations with third-party services and tools like GitHub Actions and CI/CD pipelines.

## Getting Started with GITHUB

1. Creating / Login to your GitHub account.
  - a. Open any browser like Chrome and go to github <https://github.com/>
  - b. If you don't have an account, create one or login if you already have an account .



Sign in to GitHub

Username or email address

Password

[Forgot password?](#)

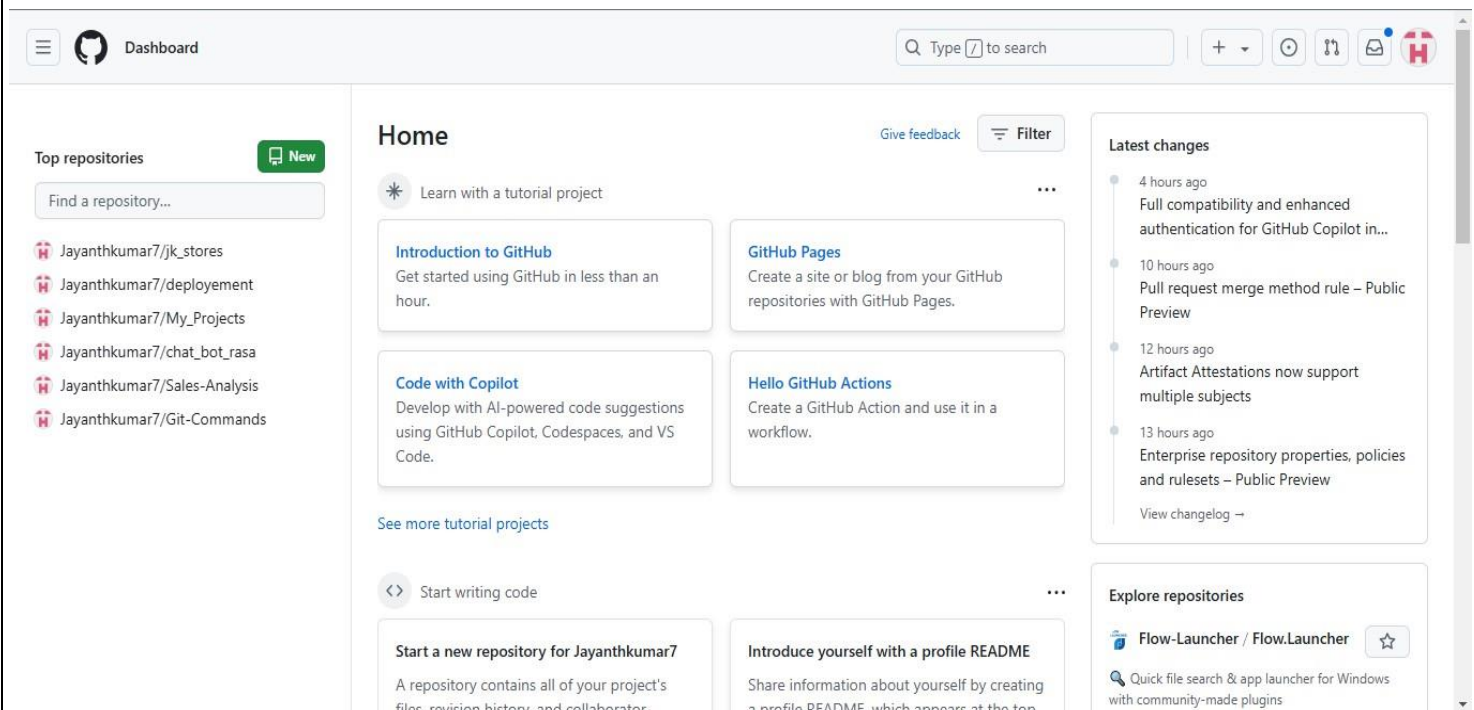
Sign in

[Sign in with a passkey](#)

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

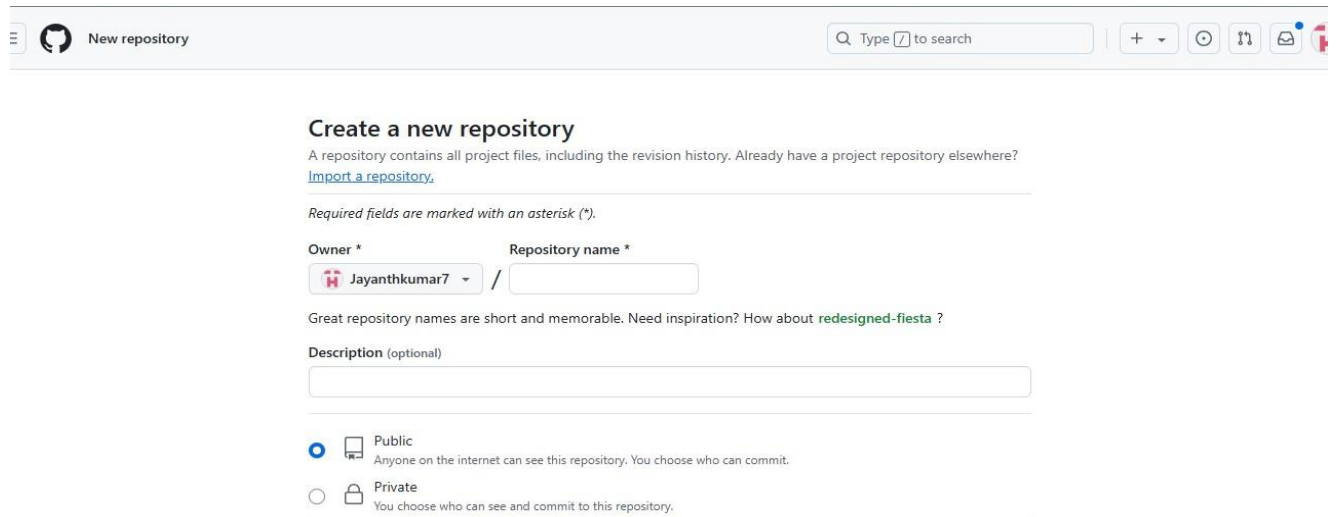
## 2. Dashboard management.

- a. On logging in you can now view the Dashboard which will consist of repositories you have already made
- b. Also one can change the settings, theme , and can also check other repositories which are not created by others by searching on the top-right corner.
- c. If you want to create a new repository you can create it by clicking on the new button which is on the right corner.



### 3. Create a new repository

- a. Click on the new button in the dashboard.
- b. Enter the name of the new repository
- c. Add the description if you need
- d. Create a readme file if you want to add
- e. Set the view to public/ private based on the priority
- f. If you want the repo to be visible to all the enable public
- g. Else set the visibility to private
- h. After all that scroll down and enter create repository.
- i. The repo is now created



The screenshot shows the GitHub 'Create a new repository' interface. At the top, there's a header with the GitHub logo, 'New repository', a search bar, and navigation icons. The main heading is 'Create a new repository', followed by a subtext: 'A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)'. Below this, a note states 'Required fields are marked with an asterisk (\*)'. The form has two required fields: 'Owner \*' (a dropdown menu showing 'Jayanthkumar7') and 'Repository name \*' (an empty text input). A suggestion text reads: 'Great repository names are short and memorable. Need inspiration? How about [redesigned-fiesta](#) ?'. There is an optional 'Description' text input field. At the bottom, there are two radio button options for visibility: 'Public' (selected) with the description 'Anyone on the internet can see this repository. You choose who can commit.', and 'Private' with the description '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.](#)

---

 You are creating a public repository in your personal account.

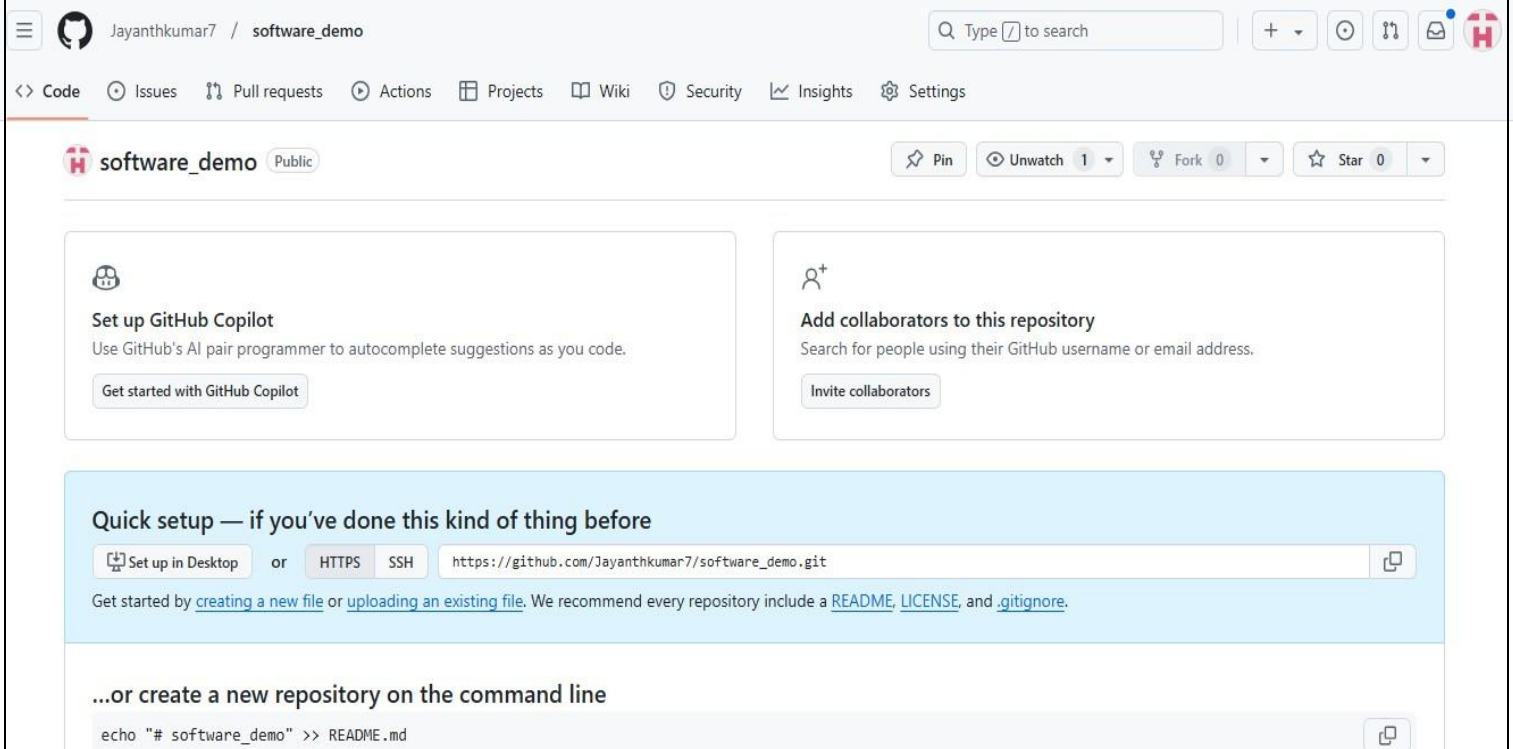
---

Create repository



#### 4. Managing the repository:

- a. After creation of the repository one can now able to add the code
- b. We can add the code by uploading / using git commands in the cmd in the local system of the user and push them to the repository.



#### 5. Add Files via upload

- a. A repo without the file's is use less so there are quite a few ways to add the files to the repo
- b. One is by uploading
- c. Under the quick setup there is an option "create a new file or uploading an existing file"
- d. Click on uploading an existing file.
- e. Choose the file you want to add the the repo
- f. After uploading the file you now have to give a commit message so that you can now know what and why you have uploaded the file.

- g. After adding the commit message click on commit changes
- h. You have now added the file's to the repo by uploading method and can now view the file in our repo

Jayanthkumar7 / software\_demo

Q Type to search

+ -

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

[software\\_demo /](#)

Drag files here to add them to your repository

Or [choose your files](#)

Commit changes

Add files via upload

Add an optional extended description...

tribute page.html

Uploading 1 of 1 files

Commit changes

Add files via upload


Add an optional extended description...

Commit changes

Cancel

© 2024 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Docs](#) [Contact](#) [Manage cookies](#) [Do not share my personal information](#)


 main  1 Branch  0 Tags

 Go to file

t

Add file

 Code

 Jayanthkumar7 Add file's via upload demo

0dfbe73 · now 1 Commit

 tribute page.html

Add file's via upload demo

now

 README





## Add a README






Help people interested in this repository understand your project by adding a README.






Link to the


Repo: [https://github.com/Jayanthkumar7/software\\_demo](https://github.com/Jayanthkumar7/software_demo)

Profile:

 Jayanthkumar7



 Overview  Repositories 7  Projects  Packages  Stars



Jayanthkumar7

Edit profile

Popular repositories

[jk\\_stores](#) Public

HTML

[deployment](#) Public

deploying the streamlit web app

Python

[chat\\_bot\\_rasa](#) Public

HTML

[Git-Commands](#) Public

Forked from [joshnh/Git-Commands](#)

A list of commonly used Git commands

[Sales-Analysis](#) Public

Jupyter Notebook

[software\\_demo](#) Public

HTML

Customize your pins