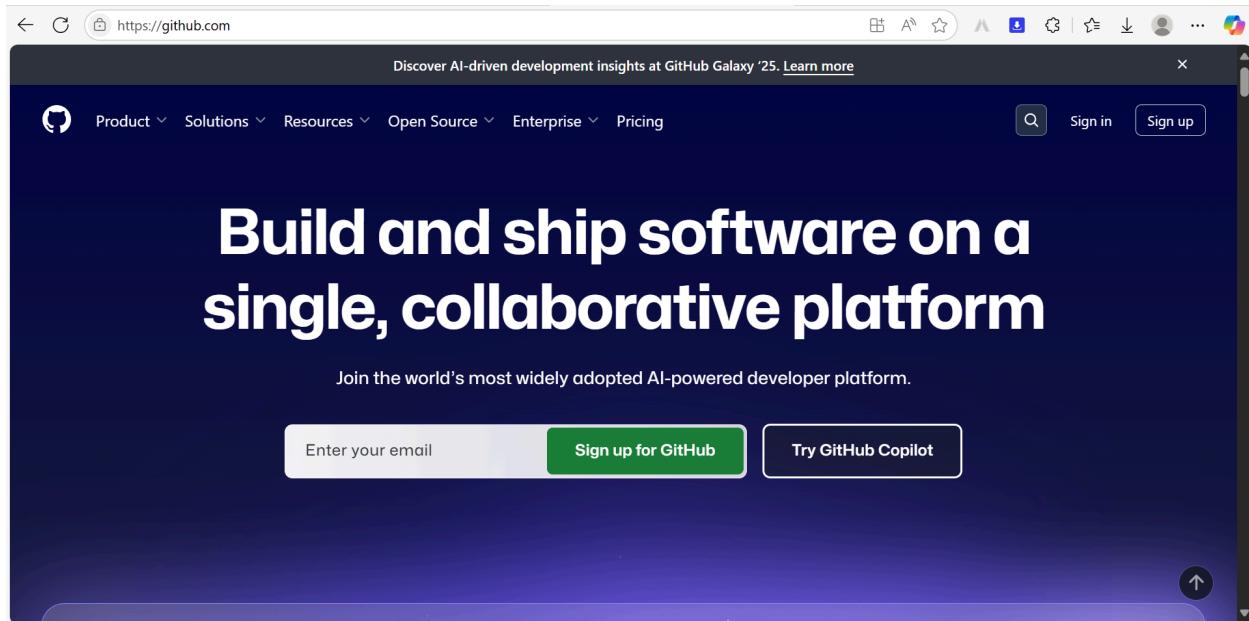


Jasmitha Chukka
jasmitha1217@gmail.com

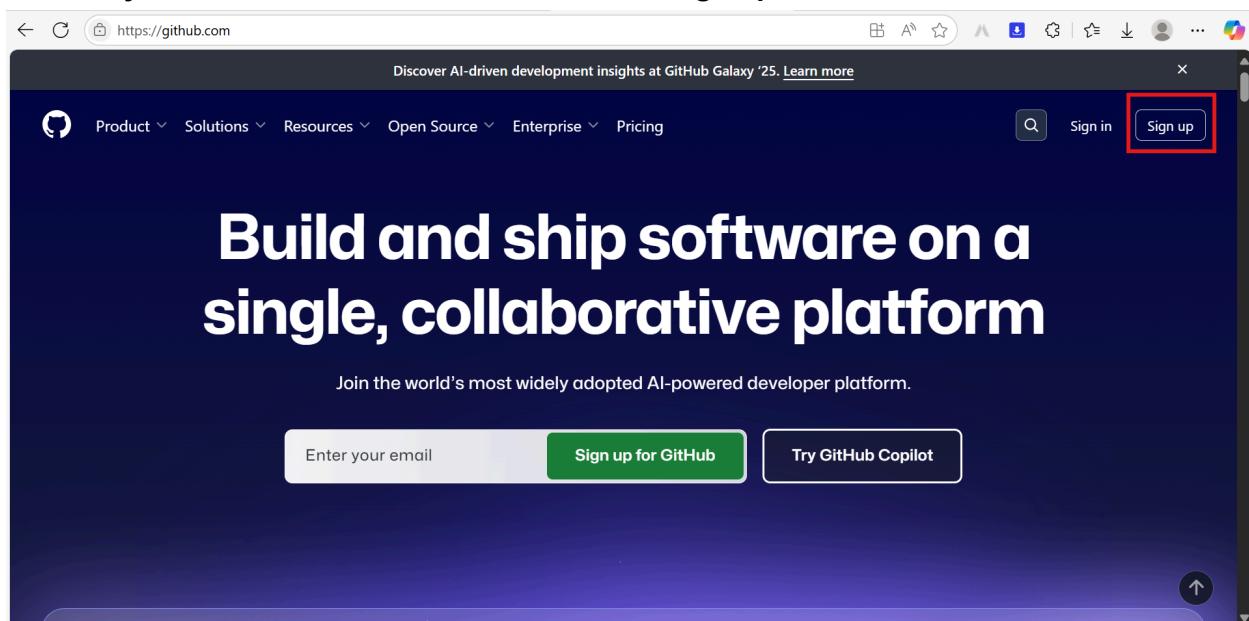
GITHUB ACCOUNT CREATION

By Jasmitha Chukka

- Go to GITHUB ([www.github.com](https://github.com)). You will get a page like this.

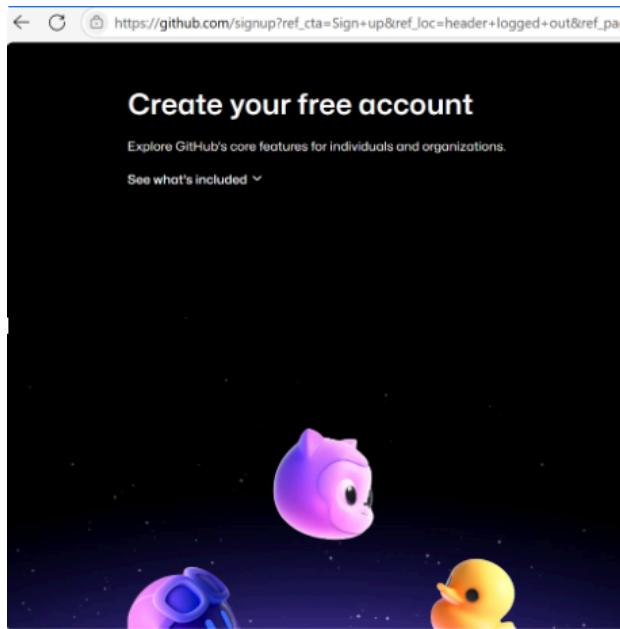


- If you have an account click on the sign in button.
- If you don't have an account click on the sign up button.



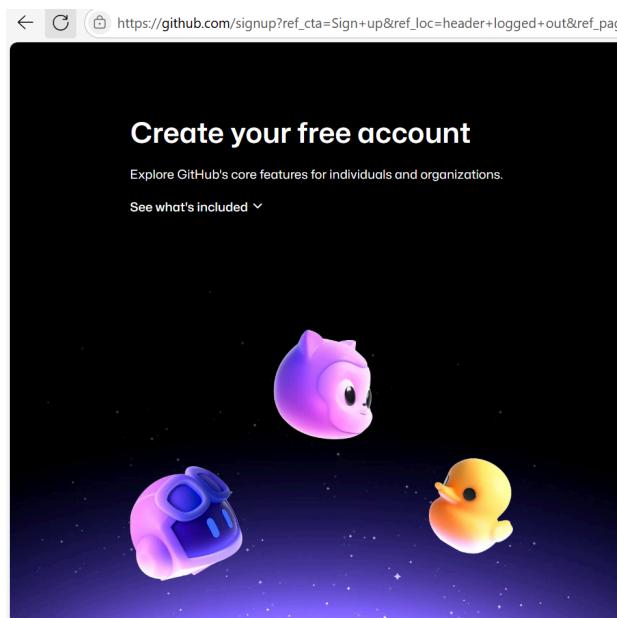
- After clicking on sign up you will get a sign up page.

Jasmitha Chukka
jasmitha1217@gmail.com



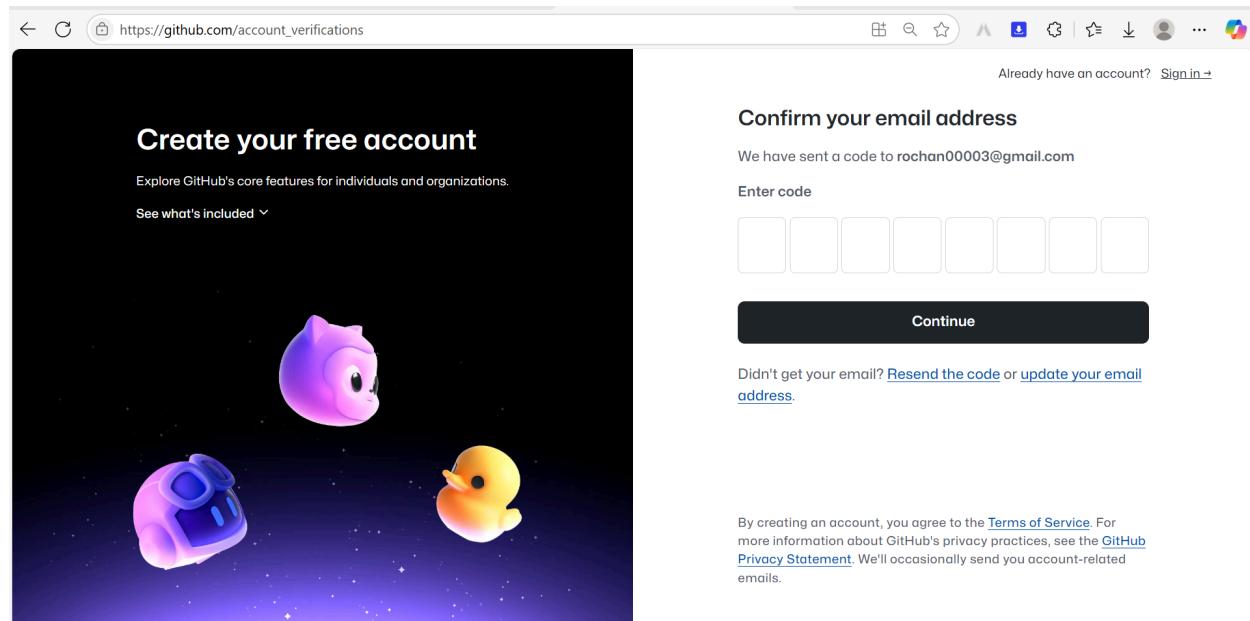
The screenshot shows the GitHub sign-up page at https://github.com/signup?ref_cta=Sign+up&ref_loc=header+logged+out&ref_page=%2F&source=header-home. The left side features a dark background with three floating 3D models: a purple cat head, a yellow duck, and a blue robot head. The right side contains a form titled "Sign up to GitHub". It includes fields for "Email" (redacted), "Password" (redacted), "Username" (redacted), "Your Country/Region" (set to India), and "Email preferences" (checkbox checked for "Receive occasional product updates and announcements"). A "Continue >" button is at the bottom.

- Please fill the details according to your requirements and click on Continue.

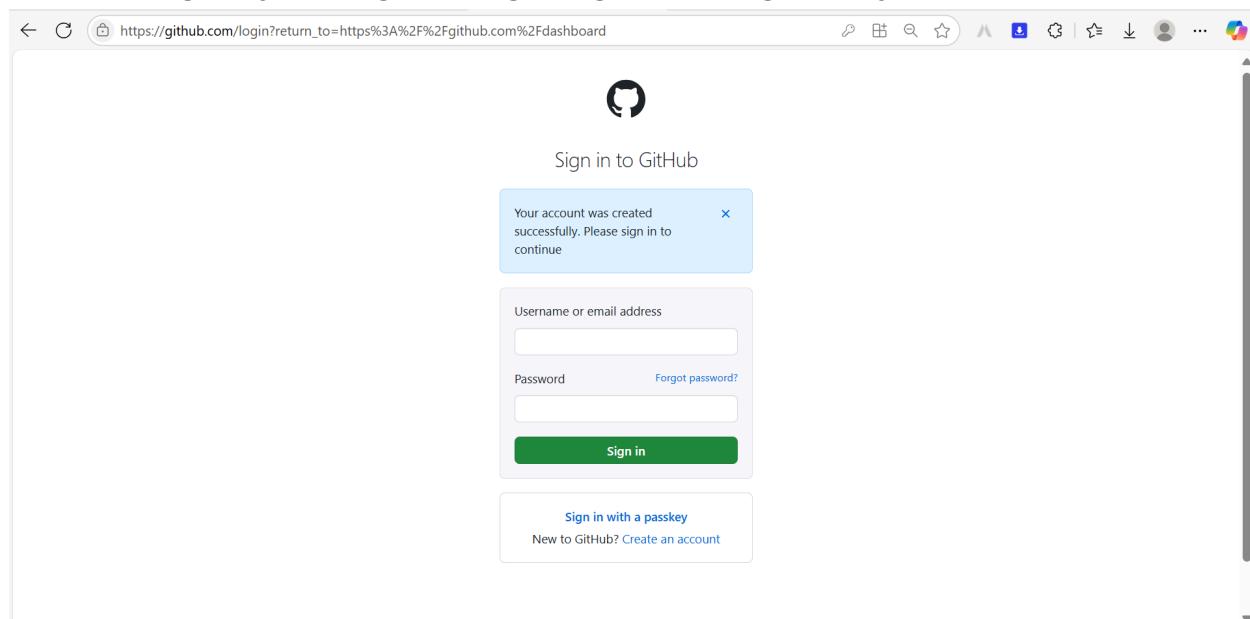


The screenshot shows the GitHub sign-up page after filling in the initial details. The right side now displays a "Verify your account" section with a message: "Please solve a puzzle so we can safely create your account." It offers two options: "Visual puzzle" (selected) and "Audio puzzle". At the top right, there is a link "Already have an account? Sign in ➔".

Verify your account.



After entering otp you will get the login page please login with your credentials.



After logging into your account you get a new dashboard.

The screenshot shows the GitHub dashboard at <https://github.com/dashboard>. The main area displays a "Home" section with various project suggestions like "Ask Copilot", "Python env variables guide", "What can I do here?", and "Create a profile README for me". It also features sections for "Learn with a tutorial project" (including "Introduction to GitHub" and "GitHub Pages"), "Code with Copilot" (with a note about AI-powered code suggestions), and "Hello GitHub Actions". A sidebar on the right titled "Explore repositories" lists popular projects such as "matplotlib", "kubernetes-client", and "scikit-learn", each with a star count and language information. At the bottom, there's a "Start writing code" button.

Jasmitha Chukka
jasmitha1217@gmail.com

Repository creation

Basically they are two kinds of repositories. 1. Public Repo 2. Private Repo

Public Repository

- Anyone can see it: The code and files are visible to everyone on the internet.
- Open-source: Others can fork, clone, and contribute (via pull requests) depending on your repo settings.
- Great for:
 - Open-source projects
 - Sharing work with the public
 - Portfolios

Example: A library you built and want others to use or contribute to.

Private Repository

- Only you (and collaborators you invite) can see it.
- Hidden from the public.
- Used when you want to:
 - Keep your code confidential
 - Work on projects before making them public
 - Store proprietary or sensitive information

Example: A company's internal project or your personal project still in development.

Now we'll how to create both private and public Repos :

Creating a Public Repo:

Click on Create Repository

The screenshot shows the GitHub dashboard at <https://github.com/dashboard>. On the left sidebar, there's a section titled 'Create your first project' with buttons for 'Create repository' and 'Import repository', both of which are highlighted with a red box. The main area is titled 'Home' and contains several cards: 'Ask Copilot', '<> Python env variables guide', 'What can I do here?', 'Create a profile README for me', 'Learn with a tutorial project' (with 'Introduction to GitHub' and 'GitHub Pages' sub-sections), 'Code with Copilot' (with 'Hello GitHub Actions' sub-section), and 'See more tutorial projects'. On the right side, there's an 'Explore repositories' sidebar with links to 'matplotlib / matplotlib', 'kubernetes-client / javascript', and 'scikit-learn / scikit-learn', each with a star count and language information.

You'll get the new window like create a new repository.

The screenshot shows the 'Create a new repository' form at <https://github.com/new>. The form has fields for 'Owner' (set to 'jasmitha1217'), 'Repository name' (set to 'PublicRepo'), and 'Description (optional)' (set to 'This is same Repo creation'). It includes sections for initializing the repository ('Add a README file' checked), adding a .gitignore template ('None'), choosing a license ('License None'), and setting the default branch ('main'). A note at the bottom says 'You are creating a public repository in your personal account.' A green 'Create repository' button is at the bottom right.

Fill in Repository Details

Repository name

- Give your project a meaningful name, e.g., `weather-app`.

Description (Optional)

- Add a short explanation of what your project does.

Visibility

- Select: **Public** → This makes your code accessible to everyone.

Optional (but Recommended) Settings

Initialize this repository with a README

- What it is: A **README.md** file is the landing page of your project.
- Why use it:
 - Explains what your project does
 - Shows how to install or use it
 - Often includes badges, screenshots, and links

Add .gitignore

- What it is: A **.gitignore** file tells Git which files/folders to exclude from version control.
- Why use it:
 - Prevents uploading unnecessary or sensitive files (e.g., `node_modules`, `.env`, compiled files)
- How to choose:
 - GitHub provides templates for many languages.
 - Example: For a Node.js project, select **.gitignore template** → **Node**.

Add a License

- What it is: A **LICENSE** file defines the legal terms under which your code is shared.
- Why use it:

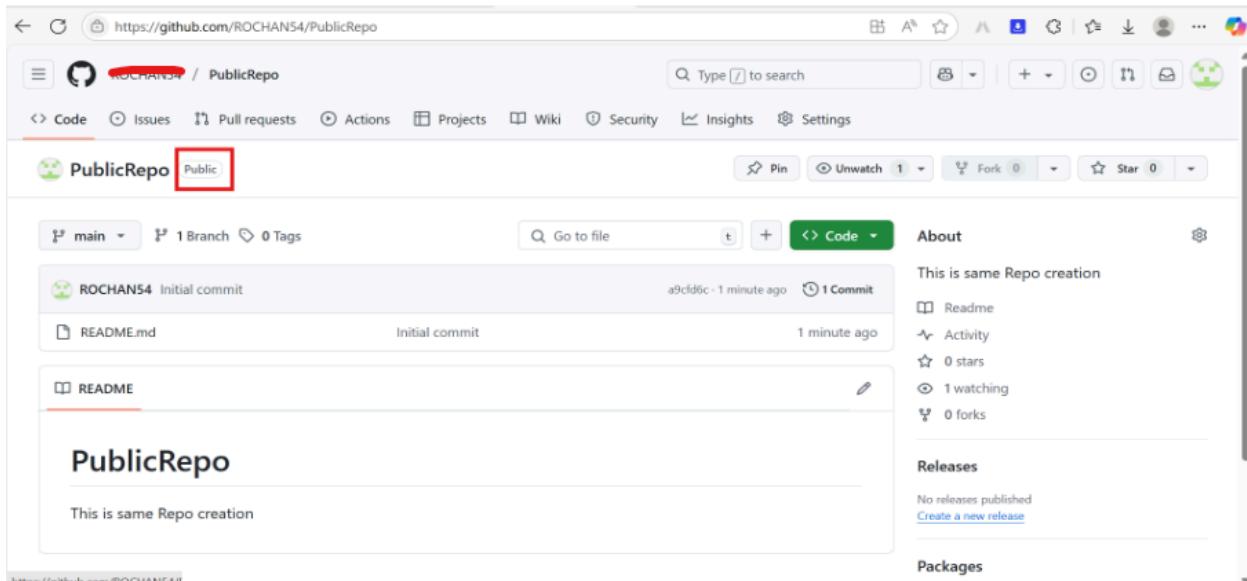
- Tells others what they can and cannot do with your code
- Encourages open-source collaboration
- Recommended license:
 - MIT License → Very permissive, allows reuse with few restrictions

Step 4: Click Create repository

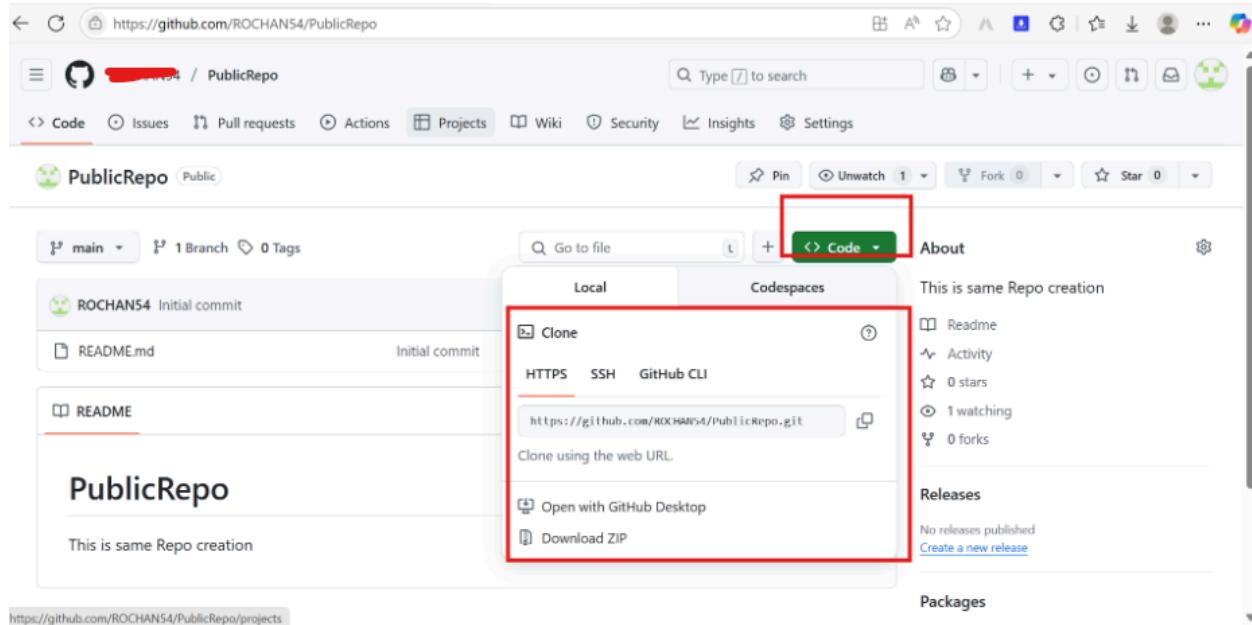
You're done!

Your new public repo now contains:

- A README.md to explain the project
- A .gitignore to exclude unnecessary files
- A LICENSE file for legal clarity



This public account any one can access this account.(ex:
<https://github.com/ROCHAN54/PublicRepo.git>) You can give or share the link via this link to access this repository.



Creating Private Repo:

Create a New Repository

- Click the “+” icon in the top-right corner.
- Select “New repository” from the dropdown.

Fill in Repository Details

- Repository name: Enter a name (e.g., **project-internal**).
- Description: (*optional*) Briefly describe what your project is.
- Visibility:
 - Select “Private” → Only you and people you explicitly invite can access it.

Optional (But Recommended)

Check the following options if needed:

- Initialize with a README (adds a default README file)

- Add `.gitignore` (choose one based on your language or framework, e.g., Python, Node)
- Add a license (optional — MIT is commonly used if you plan to make it public later)

Click “Create repository”

Your private repository is now created.

The screenshot shows two parts of the GitHub interface. The top part is a 'Create a new repository' form at <https://github.com/new>. It includes fields for 'Owner' (set to 'Redacted'), 'Repository name' ('PrivateRepo'), a note about the name being available, a 'Description' field ('This is Private Repo'), and a 'Visibility' section where 'Private' is selected. Below these are sections for 'Initialize this repository with:' (checkbox for 'Add a README file' is checked), 'Add .gitignore' (template dropdown set to 'None'), 'Choose a license' (dropdown set to 'None'), and a note about setting the default branch. At the bottom is a 'Create repository' button. The bottom part shows the resulting repository page for 'PrivateRepo'. The header shows the repository name and its status as 'Private'. The main area displays a single commit by 'ROCHAN54' with the message 'Initial commit'. The README file contains the text 'PrivateRepo' and 'This is Private Repo'. On the right side, there are sections for 'About' (describing it as a private repo with 1 commit, 1 branch, 0 tags, and 0 forks), 'Releases' (none published), and 'Packages' (none).

Now you want to give access to other users.

To access this account you need to give access to the user.

- You can invite collaborators to work with you:
 - Go to the repo → Settings → Collaborators and teams → Invite a collaborator
 - Enter their GitHub username or email

The screenshot shows the GitHub repository settings page for 'ROCHAN54/PrivateRepo'. The 'General' tab is selected. On the left, the 'Access' section is highlighted with a red box, containing the 'Collaborators' option. The main content area is titled 'Collaborators and teams' and shows a 'Private repository' status. It indicates '0 collaborators have access to this repository. Only you can contribute to this repository.' Below this, the 'Manage access' section shows a message: 'You haven't invited any collaborators yet' with a 'Add people' button. The entire 'Manage access' section is also highlighted with a red box.

If you have any queries
Please reach out to me
jasmitha1217@gmail.com