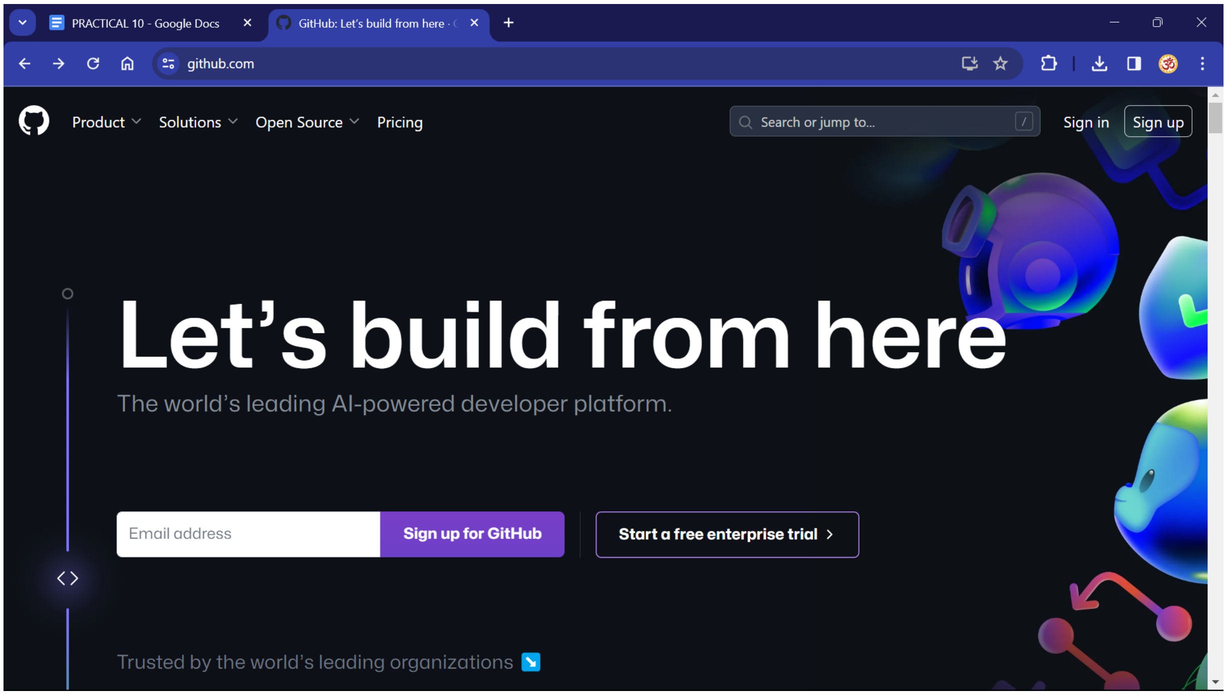
Practical 10

AIM: Demonstration GitHub facility

INTRODUCTION: GitHub is a popular web-based platform for hosting and collaborating on code repositories. It provides facilities for version control, issue tracking, code review, and team collaboration. Here's a demonstration of some basic GitHub facilities:

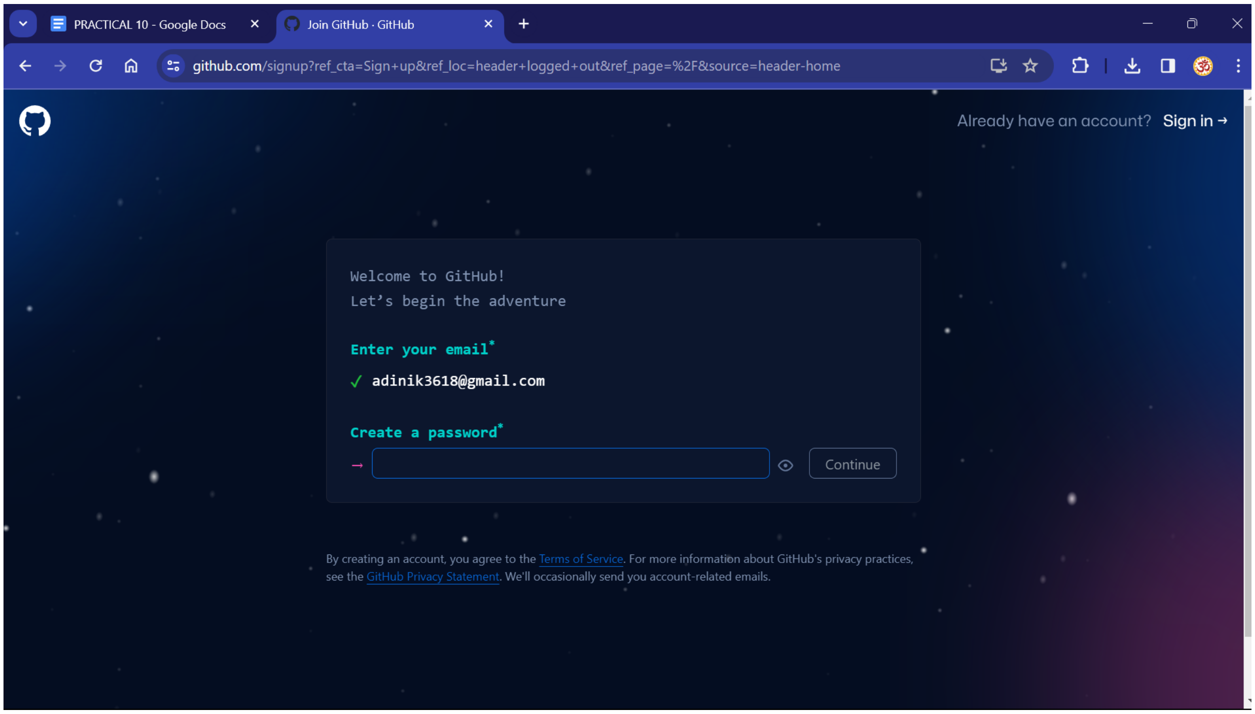
Step 1: Create a GitHub Account If you don't have a GitHub account, go to https://github.com/ and sign up for a new account.



Let's build from here

The world's leading AI-powered developer platform.

Trusted by the world's leading organizations


PRACTICAL 10 - Google Docs
Join GitHub · GitHub
github.com/signup?ref_cta=Sign+up&ref_loc=header+logged+out&ref_page=%2F&source=header-home
Already have an account? Sign in →

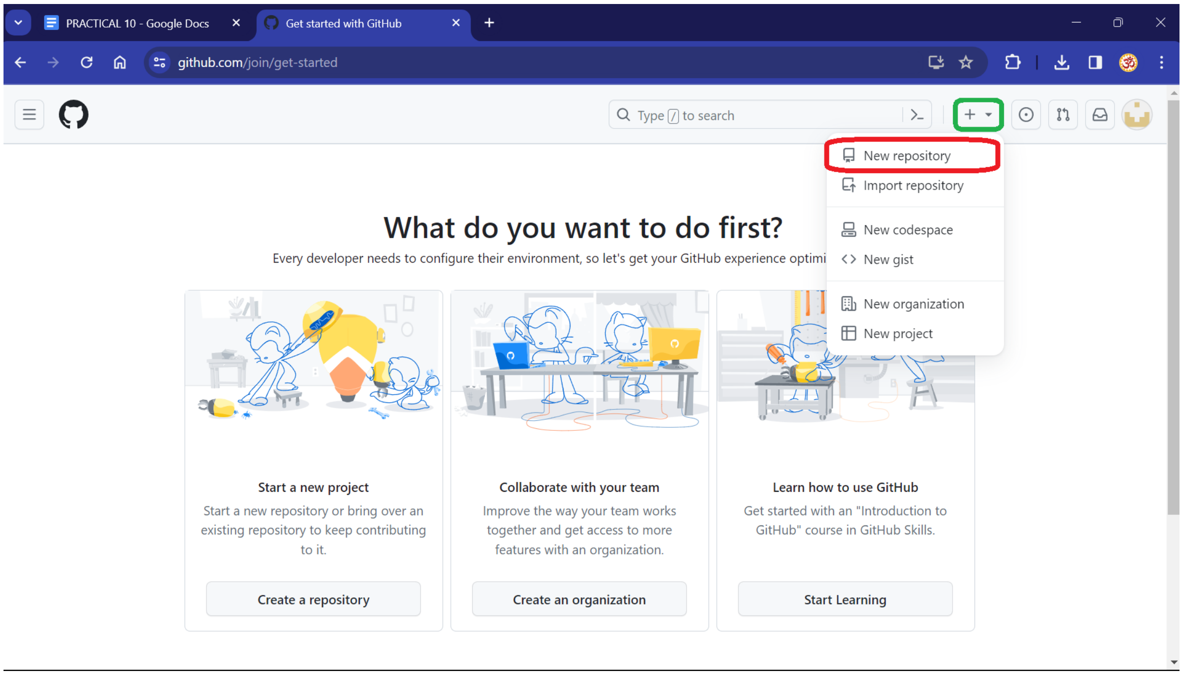
```
Welcome to GitHub!
Let's begin the adventure

Enter your email*
✓ adinik3618@gmail.com

Create a password*
→
```

By creating an account, you agree to the Terms of Service. For more information about GitHub's privacy practices, see the GitHub Privacy Statement. We'll occasionally send you account-related emails.

Step 2: *Create a New Repository Once you have a GitHub account, log in, and click on the "+" sign in the top right corner. Then, select "New repository." Fill in the repository details, such as the repository name, description, visibility (public or private), and other options. Click "Create repository" when you're done.*



What do you want to do first?

Every developer needs to configure their environment, so let's get your GitHub experience optimi

Start a new project

Start a new repository or bring over an existing repository to keep contributing to it.

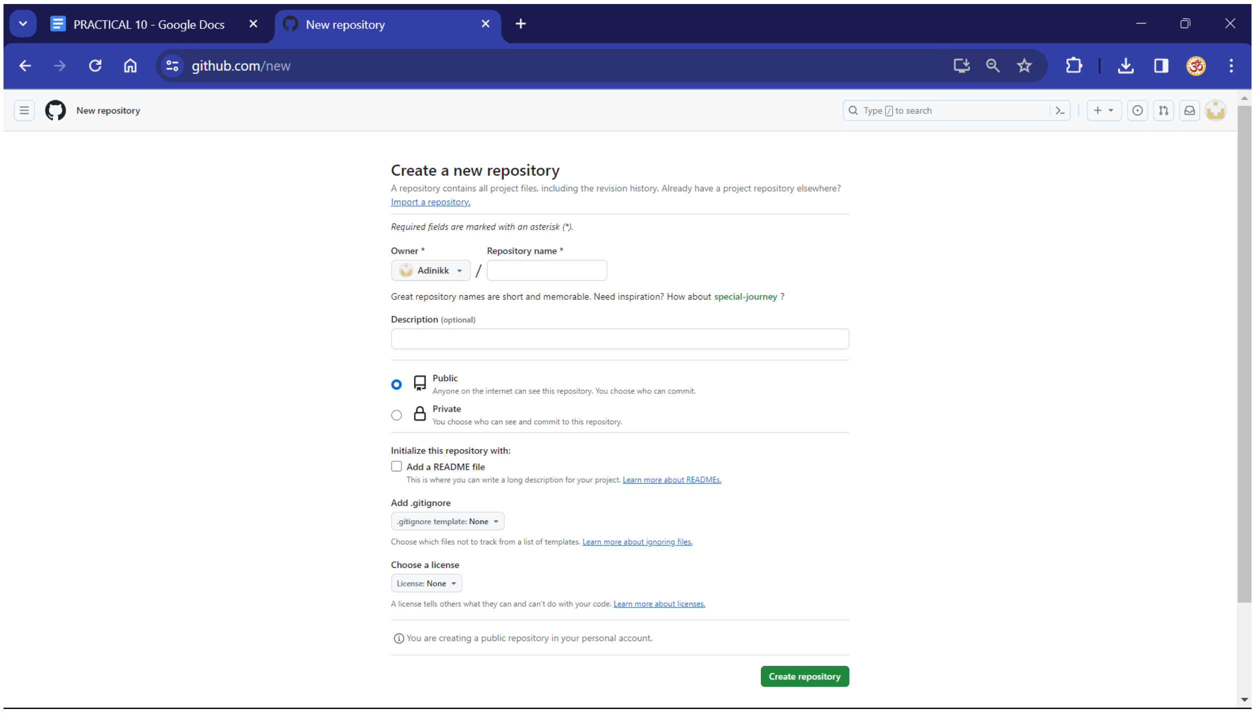
Collaborate with your team

Improve the way your team works together and get access to more features with an organization.

Learn how to use GitHub

Get started with an "Introduction to GitHub" course in GitHub Skills.

Step 3: Initializing the Repository: *After creating the repository, you have the option to initialize it with a README file, a .gitignore file (to specify which files to ignore), and an open-source license. The README file is useful for providing information about your project and instructions for anyone who visits the repository.*



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 (*).

Owner * Adinikk / Repository name *

Great repository names are short and memorable. Need inspiration? How about special-journey ?

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

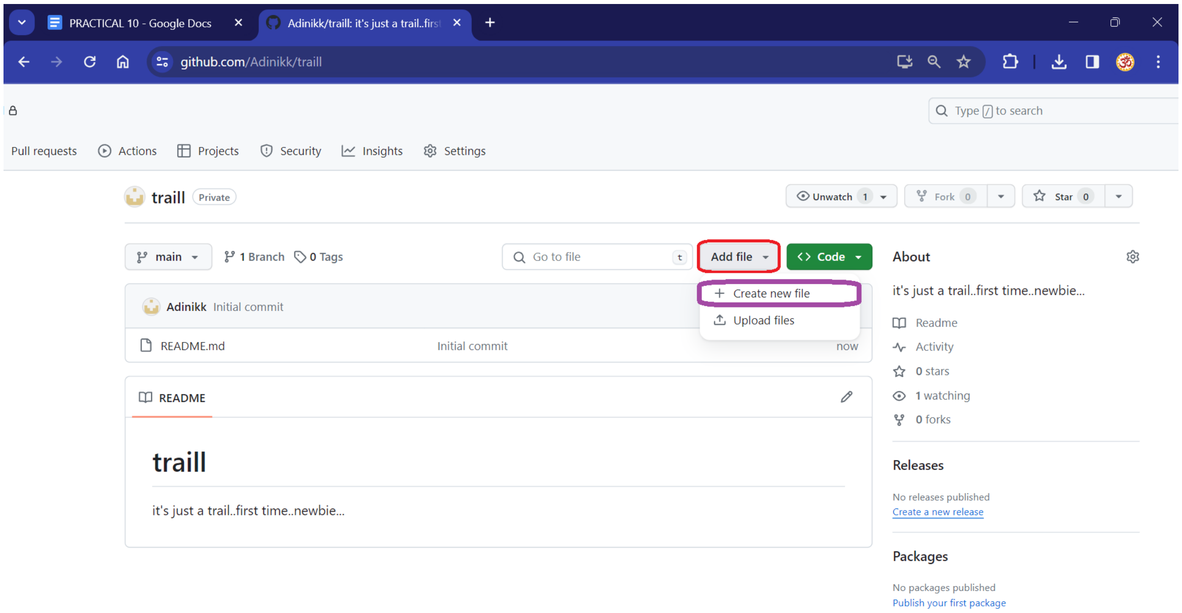
License: None

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

Step 4: Adding Files: You can demonstrate version control by adding files to your repository. Click the "Add file" button to create a new file or use the command-line interface to clone the repository to your local machine, make changes, and then push them back to the GitHub repository.



Step 5: Branching and Pull Requests: You can demonstrate the collaborative nature of GitHub by creating branches to work on specific features or fixes. When you're ready to merge your changes back into the main branch (usually "main" or "master"), you can create a pull request. Others can review the changes, provide feedback, and approve the merge.

Step 6: Issues and Discussions: GitHub also allows you to create issues to track tasks, bugs, or feature requests. You can use this feature to demonstrate how issues can be assigned to team members and resolved through code changes. Additionally, GitHub provides a "Discussions" feature where you can have more open-ended discussions with the community or contributors.

Step 7: Collaboration: For demonstration purposes, you can invite other GitHub users to collaborate on your repository. They can clone, branch, make changes, and push them back, just like you did earlier.

Remember, GitHub is a powerful tool, and there are many features and workflows to explore. This basic outline should give you a starting point to demonstrate the essential functionalities of GitHub for version control and collaborative software development.