

Indigo Hackathon Prerequisites - GitHub Repositories

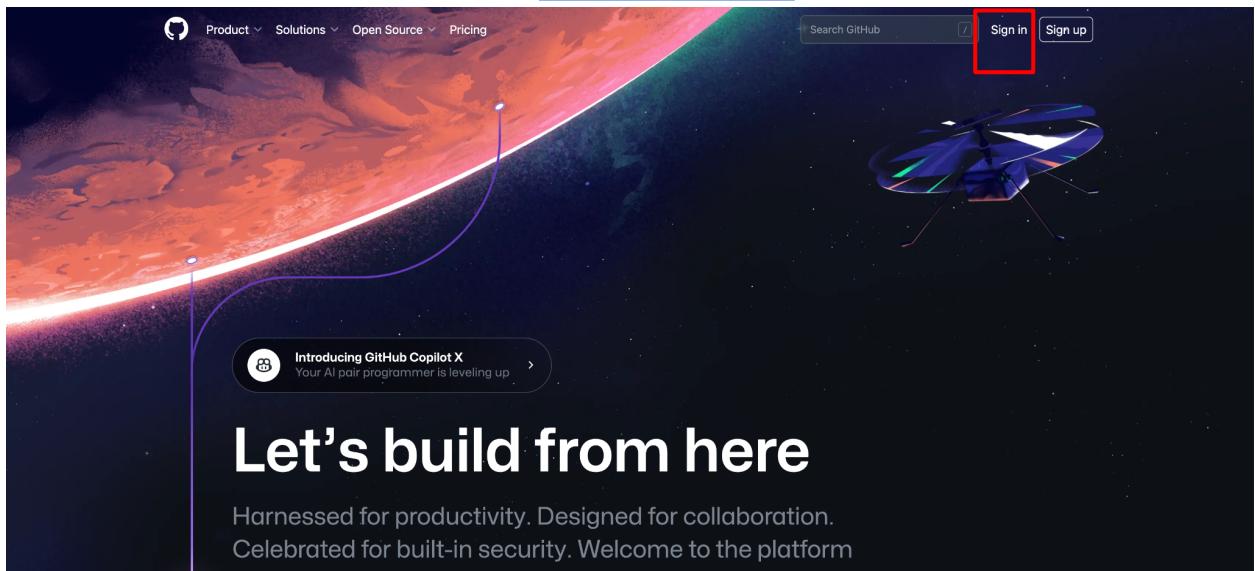
It is expected from the participants that they should have two empty public GitHub repositories on their personal GitHub account with push permissions on these repositories

High Level steps

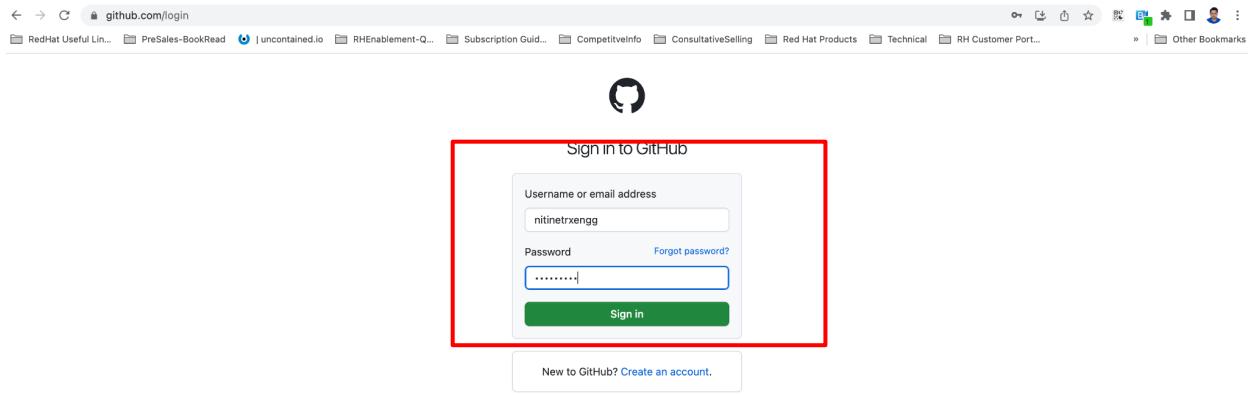
- Create two empty GitHub repositories on your personal GitHub Account
- Generate access token for these two repositories.
- Save the repositories URL & Access Token on your Laptops.

Below are the screenshots as a reference on how to create GitHub repositories & provide relevant permissions on these repositories to push the code using Fine Grained Access Tokens method.

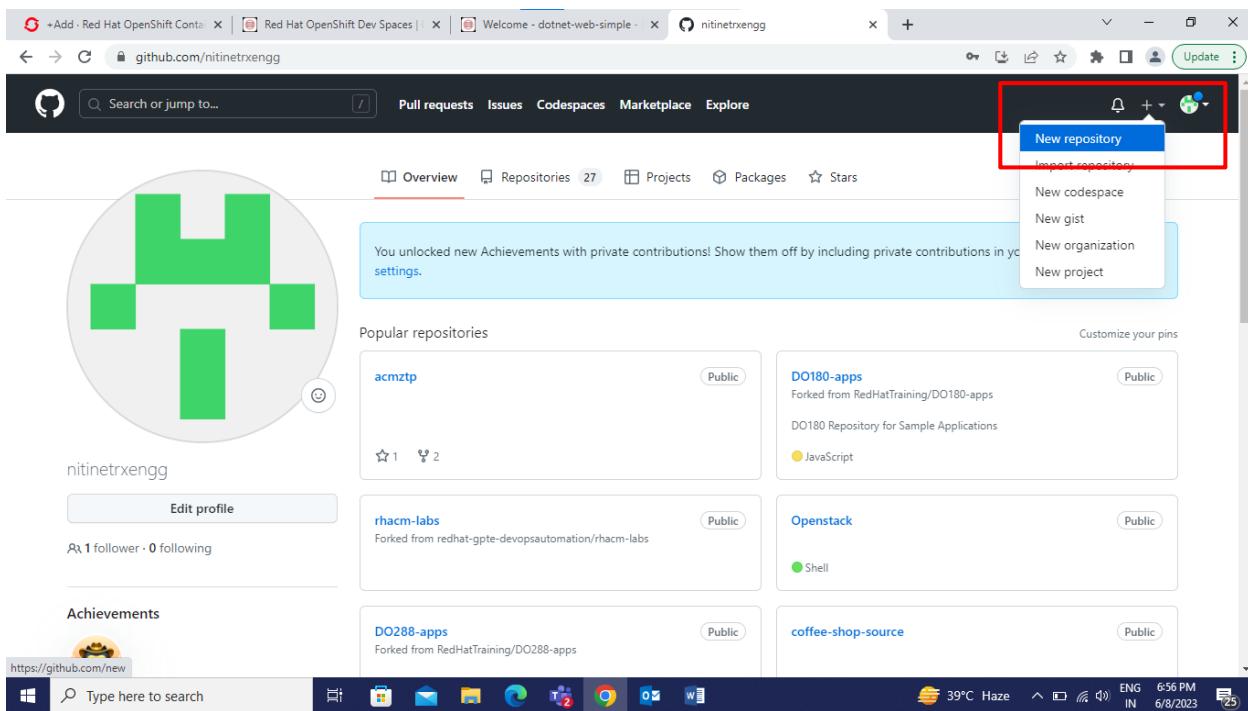
1. Login to your personal GitHub account - <https://github.com/> & click Sign in



2. Provide your GitHub Login credentials & Click on Sign in



3. Click on New repository



4. Provide the repository name as **indigohack-backend** & leave other options as default. Click on Create Repository at the bottom of the screen

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner * Repository name *

nitinetrxengg / indigohack-backend

indigohack-backend is available.

Great repository names are short and memorable. Need inspiration? How about [literate-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.](#)

① You are creating a public repository in your personal account.

5. It will display the successful page as shown below & **keep a note of this URL** of the repository on your laptop. This URL will be required later while committing the code using git commands.

[←](#) [→](#) [C](#) [github.com/nitinetrxengg/indigohack-backend](#) [RedHat Useful Lin...](#) [PreSales-BookRead](#) [uncontained.io](#) [RHEnabled-Q...](#) [Subscription Guid...](#) [CompetitiveInfo](#) [ConsultativeSelling](#) [Red Hat Products](#) [Technical](#) [RH Customer Port...](#) [Other Bookmarks](#)

 Search or jump to... Pull requests Issues Codespaces Marketplace Explore

[nitinetrxengg / indigohack-backend](#) Public

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

 Set up GitHub Copilot Use GitHub's AI pair programmer to autocomplete suggestions as you code.

 Invite collaborators Find people using their GitHub username or email address.

Quick setup — if you've done this kind of thing before

 Set up in Desktop or  HTTPS  <https://github.com/nitinetrxengg/indigohack-backend.git>

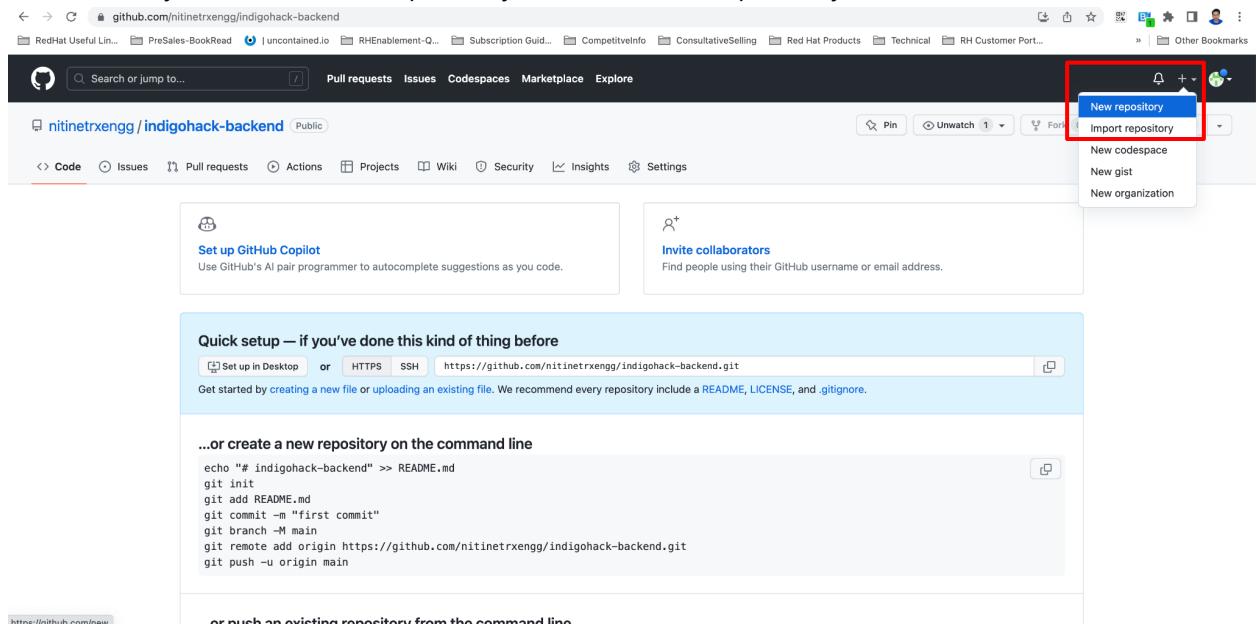
Get started by creating a new file or uploading an existing file. We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

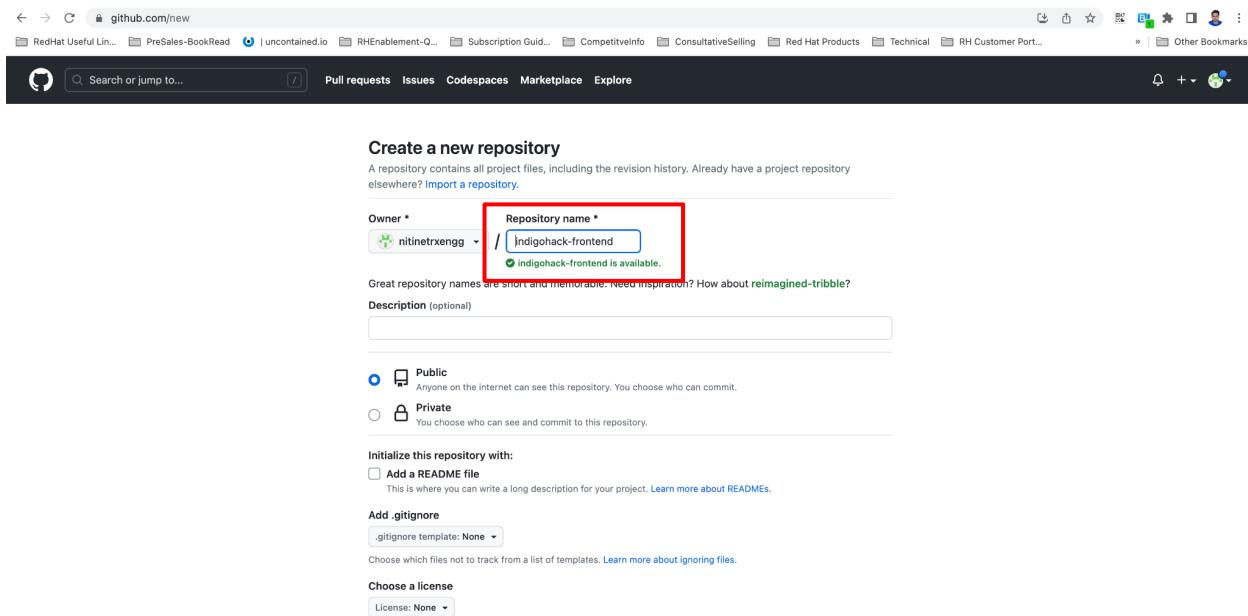
```
echo "# indigohack-backend" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/nitinetrxengg/indigohack-backend.git
git push -u origin main
```

...or push an existing repository from the command line

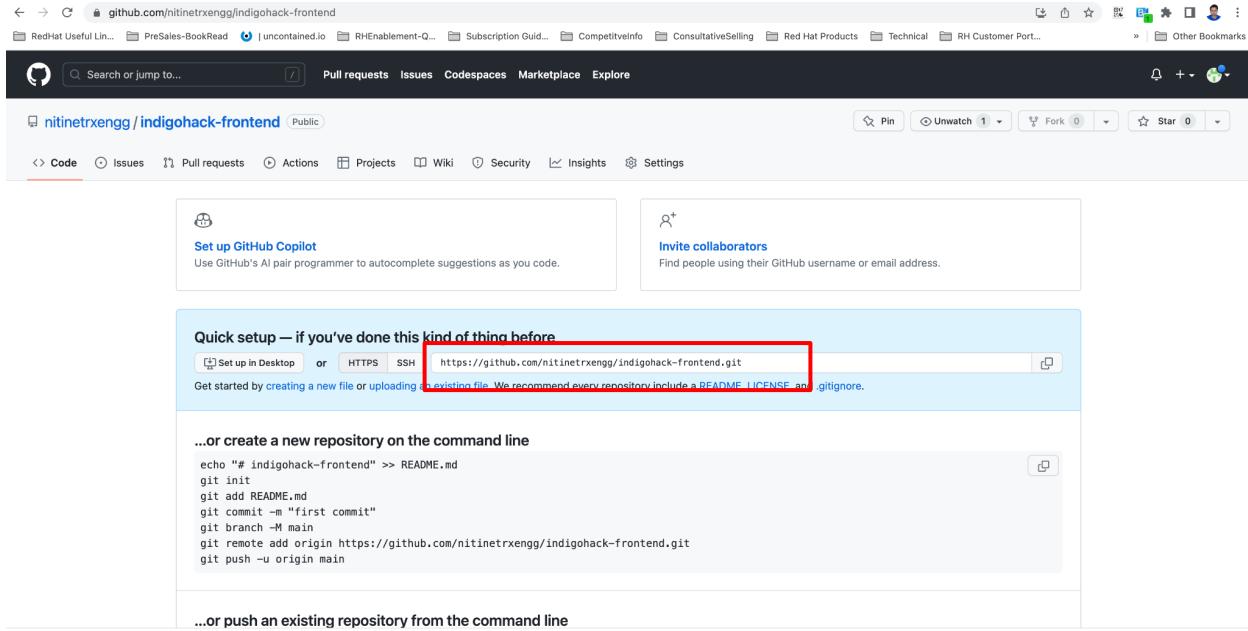
6. Now similarly create another repository. Click on New Repository



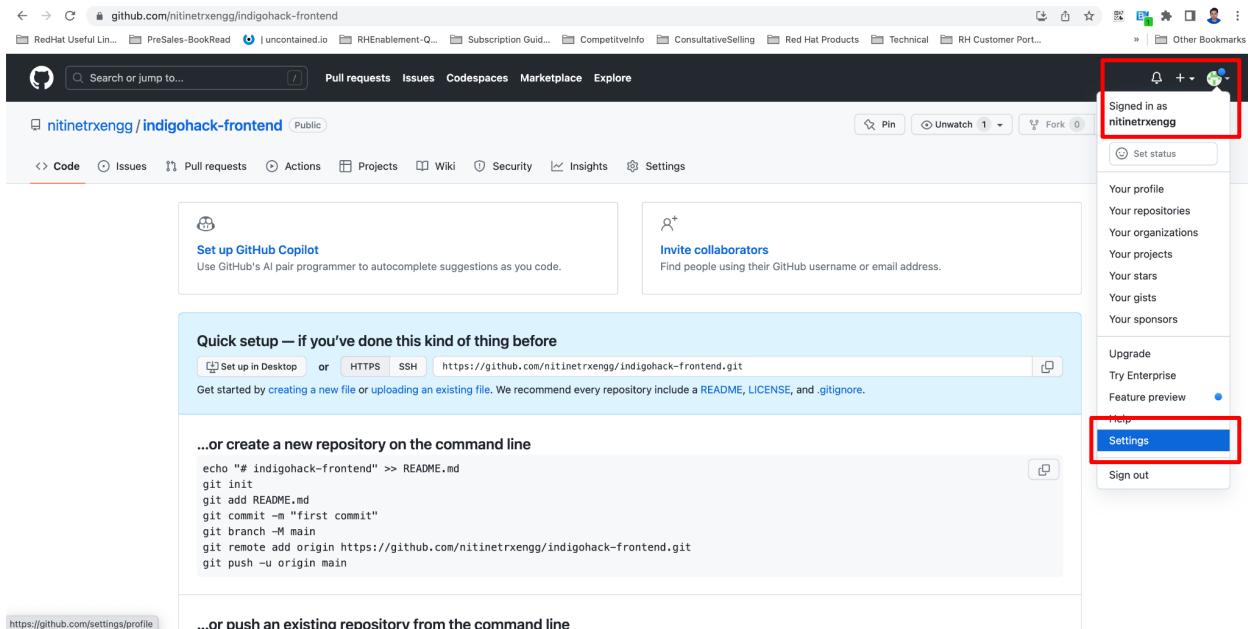
7. Provide the repository name as **indigohack-frontend** & leave other options as default & click on Create Repository at the bottom of the screen



8. It will display the successful page as shown below & **keep a note of this URL** of the repository on your laptop. This URL will be required later while committing the code using git commands.



9. After creating two repositories, go to your GitHub settings to create access token for these repositories. Click on Settings



10. Scroll down on the web page & click on Developer Settings

The screenshot shows the GitHub 'Your Profile' settings page. On the left, there's a sidebar with various profile management options like Security, Code security and analysis, Integrations, Applications, Scheduled reminders, Archives, Security log, and Sponsorship log. Below this is a large red box highlighting the 'Developer settings' section. This section contains fields for linking social profiles (Facebook and Twitter), company information (with a note about mentioning a GitHub organization), location, and a checkbox for displaying current local time. At the bottom right of this section is a green 'Update profile' button.

Contributions & Activity

Make profile private and hide activity
Enabling this will hide your contributions and activity from your GitHub profile and from social features like followers, stars, feeds, leaderboards and releases.

Include private contributions on my profile
Your contribution graph, achievements, and activity overview will show your private contributions without revealing any repository or organization information. [Read more](#).

https://github.com/settings/apps

11. Click on Fine-grained tokens under Personal access token using dropdown option

The screenshot shows the GitHub 'GitHub Apps' settings page. At the top, there's a navigation bar with links for Pull requests, Issues, Codespaces, Marketplace, and Explore. Below this is a search bar and a 'Settings / Developer settings' link. The main content area has a sidebar with GitHub Apps, OAuth Apps, and Personal access tokens. A dropdown menu is open over the 'Personal access tokens' item, showing two options: 'Fine-grained tokens' (which is highlighted with a red box) and 'Tokens (classic)'. At the top right of this dropdown is a '(Beta)' badge. To the right of the dropdown, there's a 'New GitHub App' button. Below the sidebar, there's a note about building GitHub Apps and a link to developer documentation.

The screenshot shows the GitHub 'tokens' settings page. The URL in the address bar includes '?type=beta'. The page lists several tokens, with the first one being 'dotnet-web-simple'. At the bottom right, there's a note about the beta status of the tokens feature.

12. Click on Generate new token option on the web page

The screenshot shows the GitHub developer settings page for personal access tokens. The URL is `github.com/settings/tokens?type=beta`. The 'Fine-grained personal access tokens' section is visible, with a 'Beta' badge. A red box highlights the 'Generate new token' button. Below it, a note says 'Need an API token for scripts or testing? Generate a personal access token for quick access to the GitHub API.' The navigation bar includes links for GitHub Apps, OAuth Apps, Personal access tokens (with 'Fine-grained tokens' selected), and Tokens (classic). The bottom of the page shows standard GitHub footer links like Terms, Privacy, Security, Status, Docs, Contact GitHub, Pricing, API, Training, Blog, and About.

13. Provide the name of the token as **indigohack**. If required, change expiration to 7 days. Then click on the "Only select repositories" option.

The screenshot shows the 'Create a fine-grained, repository-scoped token suitable for personal API use and for using Git over HTTPS' form. The 'Token name' field is set to 'indigohack' (highlighted by a red box). The 'Expiration' dropdown is set to '7 days' (highlighted by a red box). The 'Description' field is empty. Under 'Resource owner', the user 'nitinetrxengg' is selected. In the 'Repository access' section, the 'Only select repositories' option is checked (highlighted by a red box). A note below it says 'Select at least one repository. Max 50 repositories. Also includes public repositories (read-only.)'. A 'Select repositories' button is also present. The entire form is enclosed in a red border.

14. Click on Select repositories option & Select the two repositories that you have created in above steps. It will be selected one by one as shown in the screenshots below.

The screenshot shows the 'Repository access' section of the GitHub personal access token creation interface. The 'Only select repositories' option is selected. A search bar contains 'indigo'. Below it, a dropdown menu lists two repositories: 'nitinetrenga/indigohack-backend' and 'nitinetrenga/indigohack-frontend'. Both repositories have a blue selection bar underneath them, indicating they are selected. A red box highlights this selection area.

The screenshot shows the 'Overview' section of the GitHub personal access token creation interface. It includes sections for 'Repository access', 'Permissions', and 'Account permissions'. The 'Repository access' section shows the same repository selection as the previous screenshot, with 'indigo' searched and both 'indigohack-backend' and 'indigohack-frontend' selected. A red box highlights the selected repositories in the dropdown menu.

15. Finally, this is how it should look after selecting both the repositories

Repository access

Public Repositories (read-only)

All repositories
This applies to all current and future repositories owned by the resource owner.
Also includes public repositories (read-only).

Only select repositories
Select at least one repository. Max 50 repositories.
Also includes public repositories (read-only).

Select repositories ▾

Selected 2 repositories.

nitinetrxengg/indigohack-frontend

nitinetrxengg/indigohack-backend

Permissions

Read our [permissions documentation](#) for information about specific permissions.

Repository permissions
Repository permissions permit access to repositories and related resources. >

Account permissions
User permissions permit access to resources under your personal GitHub account. >

16. Now Click on the arrow to expand Repository permissions

Only select repositories
Select at least one repository. Max 50 repositories.
Also includes public repositories (read-only).

Select repositories ▾

Selected 2 repositories.

nitinetrxengg/indigohack-frontend

nitinetrxengg/indigohack-backend

Permissions

Read our [permissions documentation](#) for information about specific permissions.

Repository permissions
Repository permissions permit access to repositories and related resources. >

Account permissions
User permissions permit access to resources under your personal GitHub account. >

Overview

0 permissions for 2 of your repositories >

0 Account permissions >

17. Please provide access as "**Read and write**" wherever applicable. These are test repositories so you can also provide almost all the options with "Read and Write" access. Select **Read Only** where Read & Write permission is not available.

Note:

- In the below screenshot all the options are not displayed.. But you can select “Read and Write” Access wherever applicable in almost all the options & select Read only in other options.

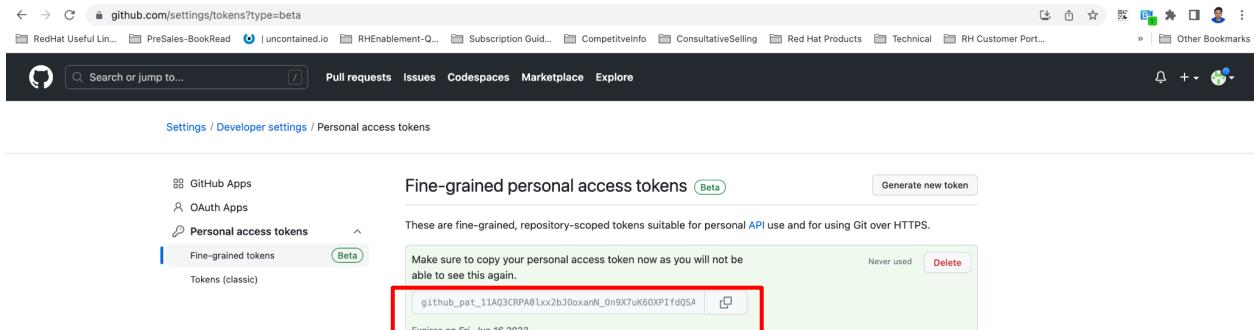
The screenshot shows the GitHub settings page for creating a personal access token. The URL is github.com/settings/personal-access-tokens/new. The page displays a list of repository permissions. Most dropdown menus for access levels are set to "Access: Read and write". Some dropdowns are set to "Access: No access". A red box highlights the dropdowns for "Actions", "Administration", "Code scanning alerts", "Codespaces", "Codespaces lifecycle admin", "Codespaces metadata", "Codespaces secrets", and "Commit statuses".

Permission	Access Level
Actions	Access: Read and write
Administration	Access: Read and write
Code scanning alerts	Access: Read and write
Codespaces	Access: Read and write
Codespaces lifecycle admin	Access: No access
Codespaces metadata	Access: No access
Codespaces secrets	Access: No access
Commit statuses	Access: Read and write

18. Now scroll down & Click on Generate token

The screenshot shows the GitHub settings page for generating a personal access token. The URL is github.com/settings/personal-access-tokens/new. The page displays sections for "Variables", "Webhooks", "Workflows", and "Account permissions". Below these is an "Overview" section showing "25 permissions for 2 of your repositories" and "0 Account permissions". At the bottom, a message states "This token will expire on Fri, Jun 16 2023." Two buttons are present: "Generate token" and "Cancel". A red box highlights the "Generate token" button.

19. This will create an access token with providing permissions to these repositories.. **Please copy the access token & save it on your laptop (This will not be displayed later if not copied now)**. This access token will be required later .



The screenshot shows the GitHub settings page for managing tokens. The URL is `github.com/settings/tokens?type=beta`. The main heading is "Fine-grained personal access tokens (Beta)". A note says, "These are fine-grained, repository-scoped tokens suitable for personal API use and for using Git over HTTPS." Below this, there's a message: "Make sure to copy your personal access token now as you will not be able to see this again." A token is listed: `github_pat_11AQ3CRPA0lxzbJ0oxanN_0n9X7uK60XP1fdQ5A`, with a copy icon next to it. The token has a green background and a red border around the copy icon. It also says "Never used" and "Delete". Below the token, it says "Expires on Fri, Jun 16 2023".

Outcome of the exercise

- Successful creation of two empty public git repositories
- Successfully created fine grained access token for the above two repositories
- Saved the repositories URLs & access token on your laptop