

এক পলকে গিট (Git) ও গিটহাব (GitHub)—পর্ব ৩/৩

M medium.com/প্রোগ্রামিং-পাতা/এক-পলকে-গিট-git-ও-গিটহাব-github-পর্ব-৩-৩-808862d945c7

June 5, 2018



আমি লিখা শেষ করার পর দেখলাম অনেক বড় হয়ে গেছে, তাই পুরোটাকে তিনটা পর্বে ভাগ করেছি।

medium.com

এক পলকে গিট (Git) ও গিটহাব (GitHub)—পর্ব ৩/৩

গিটহাবের সাথে লিঙ্ক করুন medium.com

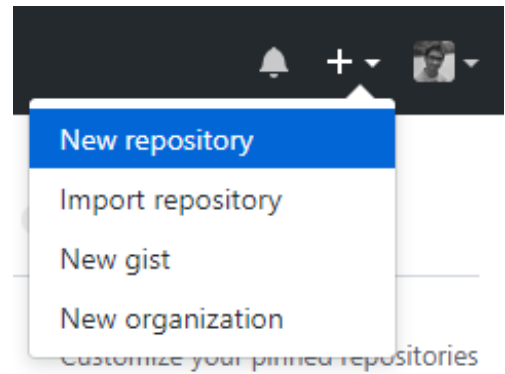
গত দুই পর্বে গিট নিয়ে আলোচনা করলেও এই পর্বে গিটহাবের উপরে বেশী আলোচনা হবে এবং বাকী ব্যাপারগুলো নিয়ে আলোচনা করবো।

গিটহাবের সাথে লিঙ্ক করাঃ

প্রথমে আপনি গিটহাবে একটা অ্যাকাউন্ট ক্রিয়েট করে নিন। এখান থেকে অ্যাকাউন্ট ক্রিয়েট করতে হবে। তারপর ইমেইল ভেরিফিকেশনসহ যাবতীয় প্রোফাইলের ইনফরমেশন দিয়ে নিজে নিজে বাকি কাজ করতে পারবেন আশা করি।

এখন আপনার গিটহাবের অ্যাকাউন্ট এ লগিন করলে উপরে ডান পাশে একটা প্লাস চিহ্ন দেখতে পাবেন। সেখানে ক্লিক করলে একটা মেনু ওপেন হবে এখানে **New repository** নামে লেখা দেখতে পাবেন। এখানে ক্লিক করলে আপনাকে নতুন রিপো তৈরী করার পেজে নিয়ে যাবে। মাইক্রোসফট গিটহাব নিয়ে নেওয়ার কারণে যদি ইন্টারফেসে কোনো পরিবর্তনও আসে তাহলে যে কোনো ভাবেই হউক এই নতুন রিপো তৈরীর লিঙ্কে ক্লিক করে পরের স্টেপে যাবেন।


তারপর এখানে প্রথমে আপনার রিপোজিটরির নাম দিবেন। রিপোজিটরির নাম ইউনিক এবং ইউ-আর-এল ফ্রেন্ডলি হতে হবে অবশ্যই। পরের ডেস্ক্রিপশন ফিল্ড অপশনাল, চাইলে কিছু দিতেও পারেন আবার খালিও রাখতে পারেন। এরপরের যে ফিল্ড আসবে সেখানে আপনি কি পাবলিক রিপোজিটরি করবেন নাকি প্রাইভেট করবেন সেটা জিজ্ঞাসা করা হয়েছে। আপনার ফ্রী অ্যাকাউন্ট হয়ে থাকলে শুধুমাত্র পাবলিক রিপোজিটরি করার অ্যাক্সেস পাবেন। তারপর বাকি ফিল্ডগুলো এভাবেই রেখে **Create repository** বাটনে ক্লিক করুন।



Create a new repository

A repository contains all the files for your project, including the revision history.

Owner

 zonayedpca

Repository name

learning-git

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

Description (optional)

☒ Public

Anyone can see this repository. You choose who can commit.

☐ Private

You choose who can see and commit to this repository.

☐ Initialize this repository with a README

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None

Add a license: None

Create repository

ব্যাস! আপনার গিটহাবে নতুন রিপোজিটরি ক্রিয়েট কমপ্লিট। এখন কিছু ইন্ট্রাকশন দেখবেন যেগুলো আসলেই অনেক দরকারী আপনার পরবর্তী স্টেপগুলোর জন্য। এখানে একদম নতুন গিট রিপোজিটরি বানিয়ে কিভাবে গিটহাবের সাথে কানেক্ট করবেন (প্রথমটা) বা অলরেডি গিট রিপোজিটরি আছে এমন প্রোজেক্টকে কিভাবে গিটহাবের সাথে কানেক্ট করবেন (দ্বিতীয়টা) সেই ইন্ট্রাকশন দেওয়া আছে। আমাদের যেহেতু গিট রিপোজিটরি অলরেডি আছে, তাই দ্বিতীয় ইন্ট্রাকশন অনুযায়ী কাজ করবো।



গিটহাবে পুশ করাঃ

এখন কমান্ড লাইন থেকে আপনার লোকাল ডিরেক্টরিতে থাকা প্রোজেক্টে চলে যান। `git status` দিয়ে শিউর হয়ে নেন সবকিছু কমিট করা আছে কিনা, নাকি কোনো কাজ কমিট করা বাকি আছে। তারপর এটাও চেক করে দেখুন কোন ব্রাঞ্চে আছেন। মাস্টারে থাকলে আপনি রেডী আপনার প্রোজেক্ট পুশ করার জন্যে।



এখন গিটহাবে দেওয়া ইন্সট্রাকশন অনুযায়ী আমাদের এই প্রোজেক্ট গিটহাবে পুশ করবো। তার জন্যে প্রথমে আপনাকে **remote origin** অ্যাড করতে হবে এভাবেঃ

```
git remote add origin https://github.com/zonayedpca/learning-git.git
```

এখানে লিঙ্কটা আপনার প্রোজেক্টের হবে, লিঙ্কটা গিটহাবে ক্রিয়েট করা প্রোজেক্টের লোকেশন থেকে খুঁজে পাবেন। এটা আপনার প্রোজেক্টে প্রথমবার অ্যাড করতে হবে, পরের বার পুশ করার সময় লাগবে না। কারণ হচ্ছে আপনার প্রোজেক্ট আপনি কোথায় হোস্ট করতে চাচ্ছেন সেটা আপনার প্রোজেক্টে প্রথমবারই বলে দিতে হবে। পরেরবার থেকে সে মনে রাখবে। তখন শুধু পুশ করলেই হবে। এখন প্রোজেক্ট পুশ করতে চাইলেঃ

```
git push origin master
```

ব্যাস এবার আপনার প্রোজেক্ট গিটহাবে চলে যাবে। এখানে আপনার গিটহাবের ইউজারনেম পাসওয়ার্ড দেওয়া লাগতে পারে। তবে এভাবে পুশ করার সময় বারবার ইউজারনেম পাসওয়ার্ড না দিয়ে আপনি চাইলে আপনার ম্যাশিনের জন্যে **SSH** কী সেটাপ করে নিতে পারবেন, তাহলে **SSH** ইউজ করলে আপনাকে আর ইউজারনেম আর পাসওয়ার্ড দিতে হবে না। **SSH** সেটাপ করতে চাইলে এখান থেকে গাইড নিতে পারবেন।



এখানে আমরা আমাদের প্রোজেক্টের শুধুমাত্র মাস্টার ব্রাঞ্চ পাঠিয়েছি। অন্য ব্রাঞ্চগুলো পুশ করতে চাইলে জাস্ট **master** এর জায়গায় সে ব্রাঞ্চ এর নাম লিখে পুশ করে দিলেই হবে। যেমন আমরা **table-version** টাও যদি পুশ করতে চাইঃ

```
git push origin table-version
```

গিটহাব থেকে পুল করাঃ

এখন ধরলাম আপনার এই প্রোজেক্টে আরো কয়েকজন ডেভেলপার আছে। এদেরকে **Collaborators** ও বলা হয়। আমি পরে এ ব্যাপারে লিখেছি কিভাবে **Collaborators** অ্যাড করবেন আপনার প্রোজেক্টে। এখন অন্য কোনো **Collaborator** আপনার প্রোজেক্টে নতুন কোনো কাজ করে সেটা পুশ করেছে গিটহাবে। এখন অটোম্যাটিক্যালিই কিন্তু সেই কাজের আপডেট গিটহাব থেকে আপনার লোকাল ম্যাশিনে চলে আসবে না। সেজন্যে আপনাকে সেটা পুল করতে হবে গিটহাব থেকে এভাবেঃ

```
git pull origin master
```

বিঃদ্রঃ পুল করার আগে অবশ্যই আপনার কোনো কাজ কমিট করা বাকী থেকে গেলে সেটা কমিট করে নিতে হবে। আর এখানে মাঝেমাঝে কনফ্লিক্ট হতে পারে। মানে দুইজন **Collaborator** একই ফাইল এডিট করার কারণে সেখানে গিট যতটুকু সম্ভব অটোম্যাটিক্যালি সেই কাজগুলো মার্জ করে দিবে, ঠিক ব্রাঞ্চ মার্জ করার মতোই। আর যদি কোনো কনফ্লিক্ট পায় যেটাতে গিট কনফিউজড, সেক্ষেত্রে গিট সেই লাইনের কোডগুলো স্পেশাল কিছু লেখা দিয়ে হাইলাইট করে দিবে। আপনার তখন ম্যানুয়ালী গিয়ে কোন লাইনটা রাখবেন আর কোনটা বাদ দিবেন সেটা বাছাই করে দিয়ে আবার সেই চেঞ্জগুলো কমিট করে দিতে হবে। আমি এই লেখা সিম্পল রাখতে যাচ্ছি তাই এগুলো নিয়ে বেশি গভীরে যাবো না। গুগল আছে সবসময় আপনার পাশে।

নিজের প্রোজেক্টে পুল রিকোয়েস্ট তৈরি করাঃ

গিটহাবে সাধারণত মাস্টার ব্রাঞ্চ বাই ডিফল্ট মেইন ব্রাঞ্চ হয়ে থাকে। বাই কনভেনশন অন্য ব্রাঞ্চ অন্য কিছু টেস্ট করার উদ্দেশ্যে বানানো হয়ে থাকে। তা আপনি কোনো প্রোজেক্টে কাজ করলে সেই প্রোজেক্টে অনেকজন **Collaborators** থাকতে পারে। তারমধ্যে হয়তো লীডেও কেউ থাকতে পারেন। এখন লীডের অনুমতি ছাড়া বা সিদ্ধান্ত ছাড়া নতুন কোনো ফিচার হয়তো মাস্টার ব্রাঞ্চে অ্যাড করা নাও যেতে পারে। সেক্ষেত্রে আপনার করা নতুন ফিচার অন্য **Collaborators** কিভাবে দেখবে? সিম্পল! আপনি আরেকটা ব্রাঞ্চে কাজ করে সেটা পুশ করে দিবেন গিটহাবে। ধরি আমাদের প্রোজেক্টে আমরা এখন নতুন ব্রাঞ্চ অ্যাড করতে চাচ্ছি আমাদের **friend-lists.txt** ফাইলটা একটু মডিফাই করে।

প্রথমে নতুন একটা ব্রাঞ্চ বানিয়ে নেই `location-version` নামেঃ

```
git checkout -b location-version
```

এখন `friend-lists.txt` ফাইলটা একটু মডিফাই করে নিই। মডিফাই করার পর ফাইলটা এরকম দেখাচ্ছেঃ

```
=====
|| Dibakar Sutradhar      || Cumilla, Bangladesh
=====
|| S M Shahadat Hossain  || Cumilla, Bangladesh
=====
|| Reduanul Houque Munna || Chattogram, Bangladesh
=====
|| Ar Rolin              || Dhaka, Bangladesh
=====
|| Niraj Paudel          || Pokhara, Nepal
=====
|| Tanvir Faisal Moon    || Cumilla, Bangladesh
=====
|| Sagar Neupane         || Kathmandu, Bangladesh
=====
|| Yadav Lamechane       || Kharar, Punjab, India
=====
```

ব্যাস এখন এই মডিফিকেশনটা কমিট করে দেইঃ

```
git add --all
git commit -m "Location added"
```



এখন এই ব্রাঞ্চ গিটহাবে পুশ করে দিবোঃ

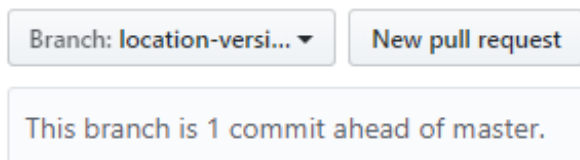
```
git push origin location-version
```

ব্যাস কোনো এরর না দেখালে আপনার পুশ হয়ে গেছে। এখন গিটহাবে গিয়ে আপনার করা নতুন `location-version` ব্রাঞ্চে চলে যান। এটা এখানে আপনার প্রোজেক্ট ফাইল লিস্টিং এর বাম পাশে উপরের দিকে পাবেন যেখান থেকে আপনি ব্রাঞ্চ সুইচ করতে পারবেন।



এবার `location-version` এ দেখবেন লেখা রয়েছে **This branch is 1 commit ahead of master.** তো এখন এটা যাতে মাস্টারে অ্যাক্সেস করা হয় সেজন্যে আপনি এখানে দেখবেন **New pull request** নামে একটা বাটন আছে। এখানে ক্লিক করলে পরের পেজে নিয়ে যাবে।

এখানে কি কি মডিফাই করা হয়েছে তার বিস্তারিত লিস্ট দেখতে পারবেন, আর পুল রিকোয়েস্ট এর জন্যে কোনো কমেন্ট করতে চাইলে সেটা লেখার সুযোগ পাবেন(অপশনাল)। পরে নিচে **Create pull request** বাটনে ক্লিক করলে ফাইনালী আপনার পুল রিকোয়েস্ট চলে যাবে।





এখানে আমরাই আমাদের নিজস্বের প্রোজেক্টে পুল রিকোয়েস্ট করেছি। তবে অনেকজন **Collaborators** থাকলে সেখানেও এভাবে করা যাবে। আর পুল রিকোয়েস্ট নামে ট্যাগে এই রিকোয়েস্ট টা থাকবে। যে প্রোজেক্টে এগুলো মার্জ করার দায়িত্ব থাকবে সে চাইলে সেখানে গিয়ে বিস্তারিত দেখে পুল রিকোয়েস্ট অ্যাক্সেস করতে পারবে আবার চাইলে বাদও দিয়ে দিতে পারবে। তবে এখানে যেহেতু সবাই **Collaborators** তাই যেকোনো চাইলেই এটা মার্জ করতে পারবে, কিন্তু প্রোজেক্টে সাধারণত ঠিকমতো ফ্লো বজায় রাখতে যার যেটা দায়িত্ব সে সেটা নিয়েই কাজ করে। গিটহাবে **Collaborators** ছাড়া আর কোনো পারমিশন লেভেল নাই তাই বাহ্যত সব **Collaborators**-ই সবকিছু করতে পারবে।

গিটহাব থেকে প্রোজেক্ট ক্লোন করা:

এখন গিটহাবে থাকা কোনো প্রোজেক্ট আমাদের লোকাল ম্যাশিনে আনতে চাইলে সেটাকে ক্লোন করতে হয়। অর্থাৎ গিটহাবে থাকা কোনো প্রোজেক্টের একটি ক্লোন কপি আপনার লোকাল ম্যাশিনে নামিয়ে আনতে চান। যে প্রোজেক্ট আপনার কাছে নাই, একদম নতুন। আপনি চাইলে গিটহাবের যে কোনো পাবলিক প্রোজেক্টই ক্লোন করতে পারবেন। সেক্ষেত্রে জাস্ট কমান্ড লাইন ওপেন করে ক্লোনের কমান্ড দিলেই পুরো প্রোজেক্ট আপনার লোকাল ম্যাশিনে চলে আসবে। এই জন্যে প্রথমে আপনার ক্লোন এর লিঙ্ক খুঁজতে হবে। ক্লোন করার লিঙ্ক আপনি গিটহাবের যেকোনো প্রোজেক্টের পাতায় গিয়ে প্রোজেক্ট ফাইল লিস্টিং এর ডান দিকের উপরের কোণায় পাবেন **Clone or download** নামের বাটনে।



এখান থেকে জিপ ফাইলও নামাতে পারবেন। তবে জিপে কারেন্ট ব্রাঞ্চের ফাইলগুলো থাকবে শুধু। গিট এর কোনো কনফিগারেশন থাকবে না। তাই আমরা এটা করবো না। আমরা কমান্ড লাইনের সাহায্যে ক্লোন করবো এভাবে:

```
git clone <GitHub Repo URL> <Local Directory Name(optional)>
```

এভাবে প্রথমে **clone** তারপরে গিটহাবের রিপোজিটরির লিঙ্ক। তারপরে আপনার লোকাল ম্যাশিনে প্রোজেক্টটা কোন ডিরেক্টরির ভিতরে রাখতে চাচ্ছেন সেটার নামও দিতে পারবেন। লোকাল ডিরেক্টরির নাম দেওয়াটা অপশনাল, না দিলে বাই ডিফল্ট রিপোজিটরির যে নাম সে নামের ডিরেক্টরিতেই ক্লোন হবে। ধরি, আমরা এই প্রোজেক্টটা ক্লোন করবো। তাই কমান্ড লাইন ওপেন করে কমান্ড লিখবো:

```
git clone https://github.com/zonayedpca/learning-git.git learning-git-with-zonayed
```

এখানে আমার লেখা ফলো করে থাকলে আপনিও হয়ত সেইম নামের রিপোজিট নিয়ে কাজ করছেন। তাই আমরা অপশনাল আর্গুমেন্টটাও দিলাম আমাদের মন মতো **learning-git-with-zonayed** নামক ডিরেক্টরিতে প্রোজেক্ট ক্লোন করার জন্যে।

এখন এন্টার দিলে প্রোজেক্ট আস্তে আস্তে ক্লোন হয়ে যাবে আপনার লোকাল ম্যাশিনে। ধরি, এটা আমি আমার ডেস্কটপে ক্লোন করেছি। এখন ডেস্কটপে দেখবেন ক্লোন করার পর **learning-git-with-zonayed** নামে একটি প্রোজেক্ট চলে আসছে। আর এটাই আপনার ক্লোন করা রিপোজিটরি। এখন এটার ভিতরে গিয়ে আপনি আবার চাইলে কাজ করে গিট ইউজ করতে পারবেন, লোকালি প্রোজেক্ট মডিফাই করতে পারবেন, সবই করতে পারবেন। আর এই প্রোজেক্টার **remote** আপনি যেখান থেকে ক্লোন করেছেন সেটাই সেট করা থাকবে অটোম্যাটিক্যালি। আপনি এই প্রোজেক্টের **Collaborator** না হয়ে থাকলে এই রিমোট পুশ করতে পারবেন না, আর **Collaborator** হয়ে থাকলে এই প্রোজেক্টেই পুশ করতে পারবেন আপনার করা নতুন কোনো কমিট।

অন্যজনের প্রোজেক্টে পুল রিকোয়েস্ট তৈরী করা:

এখন ধরলাম আপনি একটি প্রোজেক্টে কন্ট্রিবিউট করতে চাচ্ছেন। বা এখানে আপনি আমার ডেমো প্রোজেক্টে কন্ট্রিবিউট করতে চাচ্ছেন। তা সেক্ষেত্রে প্রথমে আমার গিটহাবের প্রোজেক্টে গিয়ে সেটা **fork** করতে হবে। এই **fork** বাটন গিটহাবের কাজিক্ত প্রোজেক্টের পেজে একদম উপরে ডান কোণায় পাবেন।



ফর্ক অনেকটা ক্লোনের মতই। ক্লোন করলে যেমন আপনার লোকাল ম্যাশিনে একটা কপি ডাউনলোড হয়। কিন্তু ফর্ক করলে লোকাল ম্যাশিনে কপি ডাউনলোড না হয়ে আপনার গিটহাব অ্যাকাউন্টে একটা কপি তৈরি হবে। সেই কপিটাকে আপনি আপনার রিপোজিটরি হিসাবে ব্যবহার করতে পারবেন। সেই ফর্ক করা রিপোজিটরিকে ক্লোন করে লোকাল ম্যাশিনেও নামিয়ে আনতে পারবেন।

ফর্ক এ ক্লিক করলে আপনি কোথায় সেটা ফর্ক করতে চাচ্ছেন সে অপশন আসবে। আপনার গিটহাবে কোনো অর্গানাইজেশন না থেকে থাকলে আপনার অ্যাকাউন্টই দেখাবে। আপনার অ্যাকাউন্টের উপরে ক্লিক করলে এটা ফর্ক হয়ে যাবে আপনার অ্যাকাউন্টে। এখন ফর্ক হচ্ছে এই প্রোজেক্টেরই বর্তমান ভার্সনটা আপনার অ্যাকাউন্টে কপি করে ফেলা, আর কোনোভাবে এটা মেইন প্রোজেক্টের সাথে লিঙ্কড থাকে যাতে পরে আরো কিছু জিনিস করতে পারেন। এখন এটা ক্লোন করে আপনি মডিফাই করে আবার আপনার ভার্সনেও পুশ করতে পারবেন। ফর্ক করার পর প্রজেক্টটা ক্লোন করে ফেলুন এভাবেঃ

```
git clone https://github.com/ThemeRed/learning-git
```

এখানে আপনার ফর্ক করা প্রোজেক্টের লিঙ্ক হবে। এবার প্রোজেক্টের ডিরেক্টরিতে গেলে দেখবেন আমি একটা ফাইল রেখেছি **QnA.txt** নামে। এখান আপনি আপনার নাম এবং সাথে আপনার তৈরী করা প্রথম গিটহাবের রিপোজিটরির লিঙ্ক টা জাস্ট পেস্ট করবেন। আমি আমার দুইটা লিঙ্ক দিয়েছি ডেমো হিসাবেঃ

```
-[Zonayed Ahmed] https://github.com/zonayedpca/learning-git
```

আগের থাকা লিঙ্কগুলো মুছবেন না। ওগুলোর নিচে আপনারটা পেস্ট করবেন। তারপর এটা কমিট করে পুশ করে দেন আপনার প্রোজেক্টেঃ

```
git add --all
git commit -m "Name and Repo Added"
git push origin master
```

ব্যাস এবার আপনার পুশ কমপ্লিট হয়ে গেলে আপনার গিটহাব প্রোফাইল থেকে আপনার ফর্ক করা প্রোজেক্টে গিয়ে দেখবেন নতুন এই কমিট পুশ হয়েছে।

এখন আপনার ফর্ক করা প্রোজেক্টে গিয়া ঠিক প্রোজেক্ট লিস্টিং এর উপরে বাম পাশে **New pull request** এর বাটন পাবেন একটা।



এটাতে ক্লিক করলে পরের পেজে আপনাকে মেইন প্রোজেক্ট এর সাথে আপনার ফর্ক করা প্রোজেক্ট মার্জ করার অপশন দেখাবে। এখানে হয়তো আপনি অটোম্যাটিক্যালি মার্জ করার অপশন পাবেন বা আপনার ফর্ক করার পরে মেইন প্রোজেক্টে কোনো চেঞ্জ হওয়ার কারণে আপনাকে আবার প্রোজেক্ট টা মিলিয়ে তারপর আপনার চেঞ্জ গুলো মার্জ করতে হবে। এখানে **Create pull request** নামে বাটন পাবেন।



এখানে ক্লিক করলে পরের পেজে কমেন্ট করার অপশন পাবেন। এখানে হয়তো একটু বর্ণনা দিয়ে দিবেন কি করেছেন আপনার এই কমিটে। আর নিচে আগের মতোই কি কি চেঞ্জ করেছেন সেগুলো দেখতে পারবেন।



ব্যাস এখন আপনি **Create pull request** বাটনে ক্লিক করে পুল রিকোয়েস্ট পাঠিয়ে দিতে পারবেন। এখন অরিজিনাল কন্ট্রিবিউটর আপনার চেঞ্জ মার্জ করলে আপনিও সে প্রোজেক্টের কন্ট্রিবিউটর লিস্টে অ্যাড হয়ে যাবেন। আমার এই প্রোজেক্টে আমার কথামতো নাম আর রিপোর ইউআরএল দিয়ে পুল রিকোয়েস্ট করলে আমি অ্যাক্সেপ্ট করে নিবো। তাহলে আপনিও আমার এই প্রোজেক্টের কন্ট্রিবিউটর লিস্টে অ্যাড হয়ে যাবেন।

আমি গিটহাব ইউজ করতে চাচ্ছি না...

হ্যাঁ এটা আপনার নিজের ইচ্ছা, আপনি চাইলে গিটল্যাব বা বিটবাকেট ইউজ করতে পারেন সেটা সম্পূর্ণ আপনার নিজের ইচ্ছা বা প্রয়োজন। মেইন ফাংশনালিটি একই, কারণ এখানে মেইন কাজটা করবে গিট। গিটের কমান্ড সবই এক থাকবে। শুধুমাত্র আপনার হোস্টিং প্রোভাইডার অনুযায়ী ইউআই একটু ডিফারেন্ট হতে পারে, রিমোট ইউআরএল আপনার হোস্টিং প্রোভাইডারের দেওয়া ইউআরএল অনুযায়ী অ্যাড করে নিবেন। বাট মেইন আইডিয়া/কনসেপ্ট সবারই মোটামুটি এক। তো আপনার নিজের ইচ্ছা মত আপনি যে কোনো একটা ইউজ করতে পারবেন।

স্টুডেন্টদের জন্যে গিটহাবের অফারঃ

গিটহাব স্টুডেন্টদের জন্যে ফ্রীতে প্রাইভেট রিপোজিটরির অ্যাক্সেস দেয়। আপনি যতদিন স্টুডেন্ট থাকবেন ততদিনের জন্যে ফ্রীতে ইউজ করতে পারবেন। আমার লেখা এই ফেসবুক পোস্ট ফলো করতে পারেন এজন্যে।

Zonayed Ahmed

আমরা সবাই গিটহাব সম্পর্কে জানি। গিটহাবের সুবিধা বলে শেষ করা যাবে না। ... গিটহাব স্টুডেন্টদের জন্য প্রিমিয়াম ফিচারগুলো...www.facebook.com

আরো কিছুঃ

১। **প্রোজেক্টে কাউকে কন্ট্রিবিউটর হিসাবে অ্যাড করাঃ** আপনার গিটহাবের প্রোজেক্টের পেজে গিয়ে উপরে দেখুন **Settings** নামে একটা ট্যাব আছে। এখানে ক্লিক করলে পরের পেজে একটা সাইডবার পাবেন। এখানে **Collaborators** নামে একটা মেনু আছে। এখানে ক্লিক করলে পরের পেজে **Collaborator** অ্যাড করার জন্যে ফর্ম আসবে। যাকে অ্যাড করতে চান তার ইউজারনেম দিলে তাকে দেখাবে ড্রপডাউনে। এখান থেকে সিলেক্ট করে দিলেই **Collaborator** হিসেবে অ্যাড হয়ে যাবে। এখানে যেহেতু অনেক গুরুত্বপূর্ণ কাজ করতেছেন তাই কোনো কোনো স্টেপে পাসওয়ার্ড দিতে হতে পারে।

২। **রিডমি ফাইল অ্যাড করাঃ** মাঝে মধ্যে গিটহাবের প্রোজেক্ট পেজে দেখবেন নিচে প্রোজেক্টের বিস্তারিত তথ্য দেওয়া থাকে। এজন্যে আপনার প্রোজেক্টের রুটে **README.md** নামে একটা ফাইল অ্যাড করতে হবে। আর এর ভিতরে মার্কডাউন সিনট্যাক্স ইউজ করে ভিতরে লিখতে হবে। গিটহাব অটোম্যাটিক্যালি এখানকার কন্টেন্টগুলো বিস্তারিত আকারে প্রোজেক্ট পেজে দেখাবে।

৩। **কোনো ডিরেক্টরি বা ফাইল ইগনোর করাঃ** মাঝেমাঝে আমাদের প্রোজেক্টে এমন কোনো ফাইল বা ডিরেক্টরি থাকতে পারে যেটা গিট ট্র্যাক করুক আমরা সেটা চাই না। সে জন্য **.gitignore** নামে একটা ফাইল বানাতে হবে রুট ডিরেক্টরিতে আর ভিতরে কোন কোন ফাইল বা ডিরেক্টরি ইগ্নোর করতে হবে সেগুলো মেনশন করে দিতে হবে। ভালো একটা উদাহরণ **node_modules** ডিরেক্টরি ইগ্নোর করা। গিটহাবে প্রোজেক্ট সেটাপ করার সময়ও লক্ষ্য করলে নিচে দেখবেন এটা সেটাপ করার একটা অপশন থাকে। আমি জাস্ট আইডিয়া দিলাম, বাকীটা গুগল করলে পেয়ে যাবেন।

আমি এই সিরিজে যা দেখালাম এগুলোর বাইরেও আরো অনেক কাজ রয়েছে গিট এবং গিটহাবের। তবে মেইন বেস এগুলোই। আপনি এই ব্যাসিকগুলো জানলে পরে বাকিগুলো গুগল করে করেও বের করে ফেলতে পারবেন। আসলে মেইন আইডিয়াটা ধরতে পারাই হলো আসল উদ্দেশ্য এই লেখার।