এক পলকে গিট (Git) ও গিটহাব (GitHub)—পর্ব ১/৩

M medium.com/গ্রোগ্রামিং-পাতা/এক-পলকে-গিট-git-ও-গিটহাব-github-পর্ব-১-৩-90c0f6892d0f

June 5, 2018



আপনি যদি আমাকে জিজ্ঞাসা করেন ডেভেলপমেন্ট এর জগতে সবচেয়ে ইউজফুল টুল কোনটা, তাইলে আমি চোখ বন্ধ করে বলবো **গিট।** আমার সাথে আরো অনেকেই হয়তো একমত হবেন। তবে গিট আসলে কতটা শুরুত্বপূর্ন আর ইউজফুল তা বলার অপেক্ষাই রাখে না। আমি আজকে ব্যাসিকালি গিট নিয়েই আলোচনা করবো। আর গিটহাব নিয়েও একটু আলোচনা থাকবে। গিট আর গিটহাব নিয়ে যাদের কনফিউশান আছে, এই তুইটা আসলে ভিন্ন তুইটা জিনিস। এখানে মেইন কাজ **গিট** এর।

গিট শিখার সময় সবার মধ্যে কমন যে প্রশ্নটা প্রথমেই মাথায় আসে, গিট কেন ইউজ করবো? কেন বারবার আমাকে কমান্ডলাইনে কমান্ড দিতে হবে। কেন আমাকে এই এক্সট্রা আরেকটা জিনিস ঢুকাতে হবে আমার প্রোজেক্টে। এটা কি আসলে টাইম ওয়েস্ট না? প্রথমে সবার মাথায় এটাই আসে, কারন এটার গুরুত্ব আর কাজ সম্পর্কে ধারনা না থাকলে এটাকে অতিরিক্ত একটা টুল হিসাবেই মনে হবে। আমারও প্রথমে তাই মনে হয়েছিলো। আর তাই আমি আজকে যাতে সবাই এটার গুরুত্বটা অন্তত বুঝতে পারে সেরকম উদাহরণ দিয়ে লিখার চেষ্টা করবো।

গ্রাফিক্স ডিজাইনারদের যদিও গিটের সাথে পরিচিত থাকার কথা না, তবে উদাহরণ হিসাবে এটা বেশ ভালো যায়

প্রসঙ্গত উল্লেখ্য, গত ৪ জুন, ২০১৮ ইং তারিখে টেক জায়ান্ট মাইক্রোসফট ৭.৫ বিলিয়ন ডলারের বিনিময়ে কিনে নেয় গিটহাবকে।



গিট সেটাপঃ

গিট ইউজ করতে চাইলে অবশ্যই আপনাকে গিট <u>এখান</u> থেকে ডাউনলোড করে ইন্সটল করতে হবে আপনার সিস্টেমে। আপনার অপারেটিং সিস্টেম যেটাই হউক না কেন, সবার জন্যই গিট এভেইলেবল।

সেটাপ প্রসেসে আমি বিস্তারিত যাবো না। খুবই সিম্পল, যদি কিছু বুঝতেও না পারেন জাস্ট নেক্সট দিয়ে সেটাপ প্রসেস কমপ্লিট করুন।



সেটাপ করা শেষ হলে একটা গিট ব্যাশ (Git Bash) অ্যাপ্লিকেশন পাবেন। এটা কমান্ড লাইন। এটা ওপেন করলে কমান্ড দেওয়ার উইন্ডো পাবেন, এখানে আপনি ইউনিক্স-লাইক অপারেটিং সিস্টেমের কমান্ড ইউজ করতে পারবেন। এজন্যে আপনার আগের কিছু লিনাক্স/ইউনিক্স এর কমান্ডের সাথে পরিচয় থাকলে ইজিলিই এখানে ইউজ করতে

পারবেন। অথবা আজকে এখানে যে যে কমান্ডগুলো ইউজ করবো সেগুলো কোনটা কিভাবে কাজ করে সেগুলো শিখে ফেললেই আপাতত আপনি গিট ইউজ করতে পারবেন। এখন আপনি চাইলে আপনার কম্পিউটারে থাকা সব কমান্ড লাইন/টার্মিনাল থেকেই এখন গিট চালাতে পারবেন। আপনার পছন্দের কমান্ড লাইন/টার্মিনাল ওপেন করে নিচের কমান্ডটি লিখুনঃ

git --version

এটা এরকম কিছু আউটপুটে আপনার গিটের ভার্শন দেখাবেঃ



আমার পছন্দের টার্মিনাল cmder

গিট কনফিগার করাঃ

গিটের গ্লোবাল কিছু কনফিগারেশন করে নিতে হবে সবকিছু শুরু করার আগে। খুবই সিম্পল। জাস্ট আপনার কমান্ড লাইনটা ওপেন করে নিচের কমান্ডগুলো নিজের নাম এবং ইমেইল দিয়ে সেটাপ করে নিন। নিচের কমান্ডগুলো গিট এর গ্লোবাল কনফিগারেশন। অর্থাৎ আপনার সিসটেমে যত প্রজেক্টে গিট ইউজ করবেন তার সবগুলোতে ইউজারের নাম আর ইমেইল হিসাবে এগুলোই ইউজ করবে।

```
git config --global user.name "Zonayed Ahmed"
git config --global user.email "zonayedpca@yahoo.com"
```

ঠিক এভাবেই আপনি আপনার নাম আর আপনার ইমেইল দিবেন এখানে। ব্যাস কাজ কমপ্লিট।

কিন্তু আপনি যদি চান একাধিক প্রজেক্টের জন্য একাধিক নাম ও ইমেইল থাকবে তাহলে global কীওয়ার্ড ও তার আগের হাইফেন দুটো কেটে দিন। যেমন আপনার পিসিতে অফিসের একটা প্রজেক্ট আছে আবার আপনার পারসোনাল একটা প্রজেক্ট আছে। অফিসের প্রজেক্টটি রাখা আছে পিটহাবের আপনার অফিসের অ্যাকাউন্টে। আপনি সেই অ্যাকাউন্টে অ্যাক্সেস করেন আপনার অফিসের ইমেইল দিয়ে। আর আপনার পারসোনাল গিটহাব অ্যাকাউন্ট খোলা হয়েছিল আপনার পারসোনাল ইমেইল দিয়ে। তাহলে আপনার প্রজেক্টে যদি গ্লোবাল ইউজার নেম আর ইমেইল সেট করা থাকে তখন কিন্তু সব প্রজেক্টেই আপনার একই নাম ও একই ইমেইল দেখাবে। এজন্য গ্লোবাল ইউজার নেম, ইমেইল ইউজ না করা যেতে পারে।

```
git config user.name "Zonayed Ahmed"
git config user.email "zonayedpca@yahoo.com"

//
git config user.name "Zonayed Ahmed"
git config user.email "zonayed@awesomecompany.com"
```

নতুন গিট রিপোজিটরি সেটাপঃ

গিটে ডিরেক্টরিকেই ব্যাসিকালি রিপোজিটরি (Repository) বা শর্টকাটে অনেকে 'রিপো (Repo)' বলে। আপনার অলরেডি প্রোজেক্ট আছে এমন কোনো প্রোজেক্টে গিট স্টার্ট করতে চাইলে প্রথমে আপনার গিট ব্যাশ বা আপনার যেকোনো কমান্ড লাইন থেকেই সে প্রোজেক্টের ডিরেক্টরিতে যেতে হবে। সে ক্ষেত্রে আপনি যদি গিটের সেটাপের সময় কোনো অপশন পরিবর্তন করে না থাকেন তাহলে আপনার প্রোজেক্টের ভিতরে রাইট ক্লিক করলে দেখবেন Git Bash Here নামে একটা অপশন দেখাবে। এটাতে আপনার কাজ্কিত প্রোজেক্ট ডিরেক্টরির ভিতর থেকে ক্লিক করলে এই ডিরেক্টরিত গিট ব্যাশ ওপেন হবে যেখানে আপনি কমান্ড লিখতে পারবেন।

এখন ধরি আপনার Desktop এ একটা ডিরেক্টরি আছে learning-git নামে। ভিতরে কিছু ফাইল আছে। এটাই আপনার প্রোজেক্ট, এখানেই আমরা গিট স্টার্ট করতে চাই। তাহলে আমি আমার কমান্ড লাইন এই ডিরেক্টরি ওপেন করে নিচের এই কমান্ড লিখবোঃ

git init

```
C:\Users\Zonayed Ahmed\Desktop\learning-git

\( \) git init
Initialized empty Git repository in C:/Users/Zonayed Ahmed/Desktop/learning-git/.git/
```

আমি learning-git ডিরেক্টরির ভিতর থেকে git init কমান্ড রান করলাম যেহেতু আমি এটার ভিতরের সবকিছুই ট্র্যাক করতে চাই। ব্যাস এখন এই ডিরেক্টরির ভিতরে গিটের রিপো সেটাপ হয়ে গেলো। এখন থেকে গিট সব ট্র্যাক করা শুরু করতে পারবে, এই ডিরেক্টরির ভিতরে যতো ফাইল/ফোন্ডার আছে সব। তবে ট্র্যাক করলেও গিট সেগুলোকে ভার্সন হিসাবে স্টোর করবে না। তারজন্যে আপনাকেই স্পেসেফিকলি বলে দিতে হবে কোনটা কোনটা কখন কিভাবে সেইভ করতে হবে।

আমার এই ডিরেক্টরির ভিতরে তুইটা .txt ফাইল আছে। এগুলো এখন আমি চাচ্ছি গিট ভার্শন হিসাবে সেইভ করে রাখুক। তারজন্যে আমাদের তুইটা স্টেজ ক্রস করতে হবে। প্রথমে গিট আপনার উল্লেখিত ফাইলকে স্টেজিং এরিয়াতে নিবে, তারপর আবার আপনি চাইলে সেটা ফাইনাল হিসাবে আপনার গিট রিপোতে কমিট করতে পারবেন। এই তুই স্টেজের জন্য পৃথক পৃথক তুইটা কমাভ ইউজ করতে হবে।



বর্তমান স্ট্যাটাস চেক করাঃ

তার আগে আমরা গিটের বর্তমান অবস্থা দেখতে চাচ্ছি, যেটাকে বলে বর্তমান স্ট্যাটাস দেখতে চাচ্ছি কোন কোন ফাইল ট্র্যাক করা হয় নি বা কোন ফাইল স্টেজিং এ আছে। সেজন্যে নিচের এই কমান্ড ইউজ করতে হবেঃ

git status



আমার এই ডিরেক্টরিতে তুইটা ফাইল আছে QnA.txt আর friend-list.txt নামে। আমি যেহেতু মাত্রই গিট ইনিশিয়েট করলাম এই প্রোজেক্টে তাই তুইটা ফাইলই এখানে আন-ট্র্যাকড দেখাছে। আর সাথে কিছু হিন্টও দিয়ে দিছে কিভাবে ফাইলগুলো ট্র্যাক করতে হবে।

ফাইল স্টেজিং এরিয়াতে নেওয়াঃ

আমি প্রথম <code>QnA.txt</code> ফাইলটা ট্র্যাক করতে চাই বা যেটাকে বলে স্টেজিং এরিয়াতে নিতে চাই । সেজন্যে আমাকে এভাবে কমান্ড দিতে হবেঃ

git add QnA.txt

```
C:\Users\Zonayed Ahmed\Desktop\learning-git (master -> origin)
λ git add QnA.txt

C:\Users\Zonayed Ahmed\Desktop\learning-git (master -> origin)
λ |
```

এখন আপনার ফাইল কোনো ডিরেক্টরির ভিতরে হলে তাহলে সেভাবে ফাইলের রেফারেন্স দিতে হতো git add <Your file> এভাবে। এখন আবার git status দিলে দেখবেন কারেন্ট স্ট্যাটাসঃ

git status



এখানে এখন তুইটা সেকশন দেখাচ্ছে। যেটা ট্র্যাক করেছি সেটা এখন উপরে দেখাচ্ছে Changes to be committed সেকশন এ। আর নিচে আগের সেই আন-ট্র্যাকড ফাইলটাই দেখাচ্ছে। যাই হোক এই মুহূর্তে আপনার ফাইল <code>QnA.txt</code> স্টেজিং এরিয়াতে আছে, তাই এখন কমিট করলে গিট শুধুমাত্র এই ফাইলটাকেই ভার্শন হিস্টোরীতে রাখবে। আর যেটা এখনো ট্র্যাক করা হয় নাই সেটাকে

নিয়ে কিছু করবে না। এখন যদি আমরা চাই যে এই ডিরেক্টরির ভিতরের সব আন-ট্র্যাকড ফাইলকে ট্র্যাক করতে একটা কমান্ড দিয়ে তাহলে এভাবে দিতে হবেঃ

git add --all

git add .

এখন git status দিলে দেখবেন সৰ ট্র্যাক হয়ে গেছে, মানে স্টেজিং এরিয়াতে আছে। কোনো ফাইল আন-ট্র্যাকড নাই। আগের QnA.txt এখনো স্টেজিং এ আছে, যেহেতু এটা আমরা এখনো কমিট করি নাই। আর সাথে এখন friend-lists.txt ও চলে আসছে। এখন কমিট করলে তুইটা মিলেই পুরোটার একটা ভার্শন রাখবে গিট।

ফাইনাল কমিট করাঃ

কমিট হচ্ছে আপনি ফাইনাল সিদ্ধান্ত নিবেন আপনার ট্র্যাক করা চেঞ্জেসগুলোকে গিট রিপোতে রাখতে। এখন কমিট করতে চাইলে, প্রত্যেক কমিটের সাথে একটা ম্যাসেজও দিতে হয় যাতে পরবর্তিতে একদিন পরে বা এক বছর পরে বুঝতে সুবিধা হয় অমুক কমিটটা কি কারণে করা হয়েছিলো। সবকিছু এক লাইনে এভাবে হবেঃ

git commit -m "QnA and Friend Lists Added"

এখানে QnA and Friend Lists Added হচ্ছে আমাদের এই কমিটের ম্যাসেজ। কমান্ড দেওয়া হলে এরকম ম্যাসেজ দেখতে পাবেনঃ

***আপনি স্টেজিং এরিয়ারে নেওয়ার আগে কোনোকিছু কমিট করতে পারবেন না।