বর্তমানে ভার্সন কন্ট্রোল (Version control) এবং টিমওয়ার্ক করার জন্য গিট এবং গিটহাবে নিজেকে অভ্যস্ত করা খুবই জরুরি। তাই গিট এবং গিটহাব নিয়ে একটি সংক্ষিপ্ত আকারে লিখার চেষ্টা করেছি।

টুল কনফিগার করা:

git config --global user.name "[your name]"

git config --global user.email "[your email address]"

নাম এবং ইমেইল ঠিক করুন যেটা আপনি কমিট ট্রান্সেকশনের সাথে সংযুক্ত করতে চান।

রিপোজিটরি তৈরি করা

git init [project-name]

নির্দিষ্ট নাম দিয়ে একটি নতুন রিপোসিটোরি তৈরি করুন।

git clone [url]

পরিবর্তন করা

বর্তমান স্ট্যাটাস চেক করাঃ

git status

ফাইল স্টেজিং এরিয়াতে নেওয়াঃ

git add file name

সব আন-ট্র্যাকড ফাইলকে ট্র্যাক করতে একটা কমান্ড দিয়ে তাহলে এভাবে দিতে হবেঃ

git add --all

অথবা

git add .

এখন git status দিলে দেখবেন সব ট্র্যাক হয়ে গেছে, মানে স্টেজিং এরিয়াতে আছে।

ফাইনাল কমিট করাঃ

git commit -m "your commit here"

কমিট লগ চেক করাঃ

git log

এই লগ টা আরো সুন্দর করে কম্প্যাক ভার্শনে দেখতে চাইলে উপরের কমান্ডটা এভাবেও দেওয়া যাবেঃ

git log –oneline

পূর্বের ভার্শনে ফিরে যাওয়াঃ

git checkout 588418a

এখানে শেষেরটা হচ্ছে কমিট আইডি

এখন আপনার বর্তমান ওয়ার্কিং ডিরেক্টরি আগের একটা ভার্শনে রয়েছে। কিন্তু আপনি যদি মাস্টার ব্রাঞ্চে যেতে চান তাহলে আবার চেক-আউট দিতে হবে এভাবেঃ

git checkout main

ব্রাঞ্চ তৈরীঃ

git branch myCustomVersion

আপনি চাইলে আপনার প্রোজেক্টে থাকা সবগুলা ব্রাঞ্চ এর লিস্ট ও দেখতে পারবেনঃ

git branch

এখন নতুন ক্রিয়েট করা ব্রাঞ্চে চেক-আউট করা ঠিক আগের অন্য কোনো কমিটে চেক-আউট করার মতোই।

git checkout myCustomVersion

এখন দেখবেন আপনার ব্রাঞ্চ myCustomVersion এ চলে গেছেঃ

এখানেও একটা ছোট্ট শর্টকাট টেকনিক আছে। আপনি যদি চান নতুন ব্রাঞ্চ তৈরী করে সাথে সাথে সেই ব্রাঞ্চে চেক-আউট করতে, সেটা একলাইনের কমান্ডেই করতে পারবেনঃ

git checkout -b myCustomVersion-two

নতুন ব্রাঞ্চে মডিফিকেশনঃ

এখন আমরা আমাদের এই নতুন myCustomVersion ব্রাঞ্চে নতুন কিছু ট্রাই করবো।

এখন আমি এটা মাস্টার ব্রাঞ্চে বা মেইন প্রোজেক্টে নিয়ে যেতে চাই। কিন্তু তার আগে আপনার এই পরিবর্তনগুলো বর্তমান ব্রাঞ্চে কমিট করতে হবে। কারণ আপনি যতক্ষণ পর্যন্ত কোনো কিছু কমিট না করবেন, গিট সেগুলাকে কাউন্টই করবে না। কমিট করার জন্যেঃ

git add --all

git commit -m " myCustomVersion"

ব্যাস কমিট হয়ে গেলো আমার নতুন পরিবর্তনগুলো। এখন আমি এই myCustomVersion এ থাকা কাজগুলো মেইন main ব্রাঞ্চে নিতে চাচ্ছি। সেজন্যে প্রথমে main ব্রাঞ্চে চেক-আউট করতে হবেঃ

git checkout main

ব্রাঞ্চের নাম যেহেতু main, তাই এটা লিখেই চেক-আউট করতে পারবেন। এখন খেয়াল করুন আপনার main ব্রাঞ্চে যাওয়ার পর আপনার প্রোজেক্টের সেই আগের ভার্শনটাই রয়ে গেছে। নতুন myCustomVersion এ করা কাজ এখানে আসে নাই। আপনি যদি myCustomVersion এ করা কাজ ফেলে দিতে চাইতেন, তাহলে জাস্ট main চেক-আউট করে চলে আসলেই হচ্ছে, কোথাও কোনো লেখা বা কোডে হাত দিতে হবে না।

মনে করি নতুন ব্রাঞ্চে করা কাজ আমার ভালো লাগে নাই, বা এটা আমি রাখতে চাচ্ছি না। তাহলে জাস্ট সেই ব্রাঞ্চটাকে এভয়েড করে main এ চেক-আউট দিলেই চলবে বা চাইলে ব্রাঞ্চ ডিলেটও করে দিতে পারবেন। তবে আমরা myCustomVersion টা রাখবো। কিন্তু এর সাথে কিন্তু আমরা আরেকটা ব্রাঞ্চ তৈরী করেছিলাম myCustomVersion-two নামেঃ

ব্রাঞ্চ এর লিস্ট দেখতেঃ

git branch

main

myCustomVersion

myCustomVersion-two

আমরা myCustomVersion-two ব্রাঞ্চ ডিলেট করবো এখনঃ

git branch -D myCustomVersion-two

এখন এই ব্রাঞ্চ ডিলেট হয়ে যাবে, আর সেই সাথে ব্রাঞ্চে কোনো মডিফিকেশন থাকলে সেগুলোও ডিলেট হয়ে যাবে।

ব্রাঞ্চ main -এ মার্জ করাঃ

এখন মাস্টারে চেক-আউট করার পরে দেখবেন মাস্টার আগের ভার্শনেই আছে। এখন আমরা myCustomVersion এ করা মডিফিকেশনগুলো মাস্টারে আনতে চাচ্ছি। সেটা একদম ইজি। মাস্টার ব্রাঞ্চে থাকা অবস্থায় এই কমান্ড দিলেই অটোম্যাটিক মার্জ হয়ে যাবেঃ

git merge myCustomVersion

সেই সাথে myCustomVersion এর কমিটটাও গিট অটোম্যাটিক অ্যাড করবে। গিট লগ দেখলে সেটাই দেখতে পাবেনঃ

git log --oneline

একটা কমিটের সাথে আরেকটা কমিটের পার্থক্য দেখাঃ

তাহলে এই দুটোরই কমিট আইডি লাগবে। কমিট আইডি গিট লগ দিয়ে ইজিলিই বের করতে পারবেন। এখানে git diff এর সাথে উক্ত দুইটা কমিটের আইডি পাস করতে হবে এভাবেঃ

git diff fad1051 588418a

এখানে উক্ত দুইটা কমিটে কোন ফাইলে এবং ঠিক কি কি রিমুভ (লালগুলো) এবং অ্যাড(সবুজগুলো) করা হয়েছে সেগুলো দেখানো হবে ।

গিটহাব থেকে পুল করাঃ

গিটহাব থেকে এভাবেঃ

git pull origin main

বিঃদ্রঃ পুল করার আগে অবশ্যই আপনার কোনো কাজ কমিট করা বাকী থেকে গেলে সেটা কমিট করে নিতে হবে। আর এখানে মাঝেমধ্যে 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। গুগুল আছে সবসময় আপনার পাশে।

নিজের প্রোজেক্টে পুল রিকোয়েস্ট তৈরি করাঃ

গিটহাবে সাধারনত বাই ডিফল্ট মেইন ব্রাঞ্চ হয়ে থাকে। বাই কনভেনশন অন্য ব্রাঞ্চ অন্য কিছু টেস্ট করার উদ্দেশ্যে বানানো হয়ে থাকে। তো 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 git.txt ফাইলটা একটু মডিফাই করে।

প্রথমে নতুন একটা ব্রাঞ্চ বানিয়ে নেই hafiz নামেঃ

git checkout –b hafiz

এখনgit.txt ফাইলটা একটু মডিফাই করে নিই।

ব্যাস এখন এই মডিফিকেশনটা কমিট করে দেইঃ

git add --all

git commit -m "git.txt file added"

এখন এই ব্রাঞ্চ গিটহাবে পুশ করে দিবোঃ

git push origin hafiz

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

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

এবার hafiz এ দেখবেন লেখা রয়েছে This branch is 1 commit ahead of master. তো এখন এটা যাতে মাস্টারে অ্যাক্সেপ্ট 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।

গিটহাব থেকে প্রোজেক্ট ক্লোন করাঃ

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

এভাবে প্রথমে clone তারপরে গিটহাবের রিপোজটরির লিঙ্ক। তারপরে আপনার লোকাল ম্যাশিনে প্রোজেক্টটা কোন ডিরেক্টরির ভিতরে রাখতে চাচ্ছেন সেটার নামও দিতে পারবেন। লোকাল ডিরেক্টরির নাম দেওয়াটা অপশনাল, না দিলে বাই ডিফল্ট রিপোজটরির যে নাম সে নামের ডিরেক্টরিতেই ক্লোন হবে।

git clone https://github.com/hafizurrahmanomar/ gitCommand.git NewGit

এখন এন্টার দিলে প্রোজেক্ট আস্তে আস্তে ক্লোন হয়ে যাবে আপনার লোকাল ম্যাশিনে।

অন্যজনের প্রোজেক্টে পুল রিকোয়েস্ট তৈরী করাঃ

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