

অধ্যায়ঃ Git কি? Git বা ভার্সন কন্ট্রোল সিস্টেম বলতে কি বোঝায়?

আজকে আমরা দেখব git কি? git বা ভার্সন কন্ট্রোল বলতে কি বোঝায়। git কি, এই প্রশ্নের উত্তরে আমরা সাধারণত বলে থাকি git হচ্ছে ভার্সন কন্ট্রোল সিস্টেম। তাহলে ভার্সন কন্ট্রোল সিস্টেম বলতে কি বোঝায়? ভার্সন কন্ট্রোল সিস্টেম বলতে কি বোঝায় তা জানার আগে আমরা দেখি, যদি ভার্সন কন্ট্রোল সিস্টেম ব্যবহার না করলে কি কি সমস্যার সম্মুখীন হতে হয়। বা কি কি সমস্যার সম্মুখীন হতে পারি?

এখন ধরুন আপনি একটা কম্পিউটার সফটওয়্যার ডেভেলপমেন্টের কাজ করছেন। সফটওয়্যার ডেভেলপমেন্টের জন্য যে সমস্ত কোড লিখছেন, তা ওই কম্পিউটারের লোকাল হার্ড ড্রাইভে সেভ করেছেন। এখন কোন কারণে ওই কম্পিউটারের হার্ড ড্রাইভ ক্র্যাশ করলো। বা ওই কম্পিউটারে অন্য কোন সমস্যা হলো। এবং এর ফলে আপনার লেখা কোড নষ্ট হয়ে গেল।

আবার ধরুন আপনার কম্পিউটারের লোকাল হার্ড ড্রাইভে একটি প্রজেক্ট আছে এবং ওই প্রজেক্টে আপনি সহ আরও অনেকে কাজ করতে চাই। কিন্তু প্রজেক্টটি আপনার কম্পিউটারের লোকাল হার্ড ড্রাইভে থাকায় তা সম্ভব হবে না। আবার দুইজন প্রোগ্রামার একই প্রজেক্ট, দুটি ভিন্ন কম্পিউটারে কাজ করতে চায়, তাহলে কিভাবে করবেন? কোন প্রজেক্টে এভাবে কাজ করতে চাইলে তাদেরকে পেন ড্রাইভের সাহায্যে কোড ট্রান্সফার করে নিতে হবে। এবং কাজ শেষে ম্যানুয়ালী নিজ দায়িত্বে কোড মার্জ করতে হবে। যা বেশ কষ্টকর।

এরপর আরও সমস্যা আছে, যেমন এইভাবে সফটওয়্যার ডেভেলপমেন্টের কোন প্রজেক্টে কাজ করলে তার ভার্সন ট্র্যাকিং অপশন থাকে না। এখন ধরুন আপনার একটি সফটওয়্যার আছে এবং ওই সফটওয়্যারের কোড সুন্দর ভাবে কাজ করছে। কিন্তু আপনি আপনার সফটওয়্যারের কোন ফাংশনালিটি বাড়াতে চাচ্ছেন বা কোন কোডের পরিবর্তন করতে চাচ্ছেন, বা কোন bug ফিক্স করলেন। সব কাজ শেষ করে আপনি দেখলেন, যেটাকে আপনি bug হিসেবে মনে করেছিলেন তা আসলে কোন bug না, সফটওয়্যারের একটি ফিচার। আপনি যা bug ভেবেছিলেন তা যে আসলে সফটওয়্যারের একটি ফিচার, এটি বোঝার আগেই আপনি আপনার কোড পরিবর্তন করে ফেলেছেন।

এখন আপনি আপনার কোডের পূর্ববর্তী অবস্থায় ফিরে যেতে চাচ্ছেন, যে অবস্থায় আপনার কোডের সমস্ত ফিচার সুন্দরভাবে কাজ করত। কিন্তু তার কোন উপায় নেই। কারণ কম্পিউটারের নরমাল ফাইল সিস্টেম, ভার্সন কন্ট্রোলের সুবিধা দেয় না। কিন্তু আপনি যখন বড় কোন প্রজেক্টে কাজ করবেন, তখন কোডবেজে বারবার এই ধরনের সমস্যায় পড়বেন। আর এই সকল সমস্যার সমাধান দিতে পারে git বা ভার্সন কন্ট্রোল সিস্টেম। এখন তাহলে দেখা যাক ভার্সন কন্ট্রোল সিস্টেম এই সমস্যা গুলোর সমাধান দিতে পারে কিনা।

লোকাল কম্পিউটারে কাজ করার ক্ষেত্রে আমাদের প্রথম সমস্যা ছিল, লোকাল হার্ডডিস্ক ক্র্যাশ করা, বা লোকাল কম্পিউটারে অন্য কোন সমস্যা হওয়া। তবে আমরা ভার্সন কন্ট্রোল সিস্টেম ব্যবহার করে কাজ করলে, ভার্সন কন্ট্রোল সিস্টেম আমাদের লেখা কোডগুলো ক্লাউডে রাখা থাকে। আর এই ক্লাউড হল, রিমোট মেশিন বা রিমোট কম্পিউটার। যেটা আমাদের লোকাল কম্পিউটারের চাইতে অনেক বেশি স্ট্যাবল। শুধু তাই না, ভার্সন কন্ট্রোল সিস্টেম ব্যবহার করলে ভার্সন কন্ট্রোল সিস্টেমের ক্লাউড আমাদেরকে গ্যারান্টি দেয়, এখন থেকে কখনও আমাদের কোড হারাবে না বা নষ্ট হবে না। সুতরাং আমাদের কোড নির্ভয়ে ভার্সন কন্ট্রোল সিস্টেমের ক্লাউডে আপলোড করতে পারি, এবং সেখান থেকে আমাদের কোড যেকোন সময় সিল্ক করে ডাউনলোড করে নিতে পারি। অর্থাৎ ভার্সন কন্ট্রোল সিস্টেম ব্যবহার করে কোড ক্লাউডে আপলোড করলে, হার্ডডিস্কের যেকোন সমস্যা বা লোকাল কম্পিউটারের যেখান সমস্যা হলেও ভার্সন কন্ট্রোল সিস্টেমের ক্লাউড থেকে আমাদের কোড যেকোন মুহূর্তে ফেরত পেতে পারি। তাহলে দেখা গেল ভার্সন কন্ট্রোল সিস্টেম ব্যবহার করে আমাদের প্রথম সমস্যা সমাধান করতে পারি।

এখন দ্বিতীয় সমস্যাটি দেখা যাক। দ্বিতীয় সমস্যাটি ছিল, কোন ঝামেলা ছাড়াই একাধিক ব্যক্তি যেন একই কোডবেসে কাজ করতে পারে। ভার্সন কন্ট্রোল সিস্টেম ব্যবহার করলে আপনার মূল্যবান কোড ভার্সন কন্ট্রোল সিস্টেমের ক্লাউডে রাখা থাকে, তাই এখানে এক বা একাধিক ডেভেলপার একই কোডবেসে একই কম্পিউটারে বা আলাদা আলাদা কম্পিউটারে কাজ করতে পারে। এবং কাজ শেষে তারা ভার্সন কন্ট্রোল সিস্টেমের ক্লাউডে

Git And Github For Beginner

কোড পুশ দিয়ে, কোডবেস আপটুডেট রাখতে পারে।

এখন ধরুন দুইজন ডেভেলপার, কোন প্রজেক্টের কোডবেসে একই মেথডে কাজ করেছে। যেখানে sum নামে একটা মেথড আছে, যে মেথড দুইটা number value নেয় এবং সেই দুইটা number এর যোগফল return করে। কিন্তু দুজন ডেভেলপার একই মেথডে কাজ করাই, তারা দুজনেই sum মেথডের বেশ কিছু change করেছে। এবং কাজ শেষ তারা তাদের কোড ক্লাউডে পুশ দিয়েছে, বা লোকালে merge করতে চাইছে। এখন তাহলে কি হবে? দুইজন ডেভেলপার একই কোডবেসের একই মেথডে কাজ করাই তাদের কোড conflict করবে।

ভার্সন কন্ট্রোল সিস্টেম git এ merge tool নামে একটা tool আছে। এই merge tool এর মাধ্যমে আমরা এধরনে conflict - এর সমাধান করতে পারি। git merge tool এর মাধ্যমে merge করা manual merge করার চাইতে অনেক সহজ। ভার্সন কন্ট্রোল সিস্টেম git এ merge tool থাকাই conflict সমস্যা বা অন্য কোন সমস্যা ছাড়াই একই প্রজেক্টে এক বা একাধিক ডেভেলপার খুব সহজে কাজ করতে পারে। সুতরাং দেখা গেল ভার্সন কন্ট্রোল সিস্টেম git একই প্রজেক্টের কোডবেসে এক সাথে এক বা একাধিক ডেভেলপার কাজ করতে না পারার সমস্যার খুব ভাল সমাধান দিতে পারে।

এখন আসুন দেখি ভার্সন কন্ট্রোল সিস্টেম git বা github পরের সমস্যার সমাধান দিতে পারে কিনা দেখি। তৃতীয় সমস্যাটা ছিল ভার্সন ট্র্যাকিং সিস্টেম বা ভার্সন কন্ট্রোল সিস্টেম। ভার্সন কন্ট্রোল সিস্টেম বা VCS সঠিক ভাবে ব্যবহার করলে আপনার কোডের সমস্ত চেঞ্জ ট্র্যাক করা যায়। এতে একই ফাইল বিভিন্ন ভার্সন আকারে ক্লাউডে সেভ থাকে।

এখন ধরুন আপনার একটা ক্লাস ফাইল আছে। এবং এই ক্লাসে sum নামের একটি ফাংশন আছে। এখন ধরুন আপনার একটা ক্লাস ফাইল আছে এবং এই ক্লাসে sum নামের একটি ফাংশন আছে। এই ফাংশন দুইটা নাম্বার নেয় a ও b এবং a + b কে sum করে return করে। কিন্তু পরবর্তীতে আপনি আপনার প্রয়োজন মতো ফাংশনটিকে চেঞ্জ করেছেন।

এবং এই নতুন চেঞ্জ গুলোকে version number 2 দিয়ে ক্লাউডে আপলোড করেছেন। একই ভাবে version number 3, version number 4 ভিন্ন পয়েন্টে ক্লাউডে সেভ হবে।

এখন আপনি যেকোন পয়েন্টে যদি মনে করে আপনার প্রিভিয়াস ওয়ার্কিং ভার্সনে ফিরে যেতে চান, তা সহজেই যেতে পারবেন।

এখন বিভিন্ন ধরনে ভার্সন কন্ট্রোল সিস্টেম বা VCS আছে। উদাহরণ স্বরূপ git, mercurial, subversion সহ আরও অনেক ভার্সন কন্ট্রোল সিস্টেম বিদ্যমান।

কিন্তু git এখন পর্যন্ত সব থেকে ভালো ভার্সন কন্ট্রোল সিস্টেম। প্রধানত যে সমস্ত কারণে সবাই git পছন্দ করে তার ভিতর একটা হলো - এটা ডিস্ট্রিবিউটেড ভার্সন কন্ট্রোল সিস্টেম। অর্থাৎ এখানে ভার্সন হিস্ট্রি লোকাল কম্পিউটারে সেভ রাখে। এবং একই সাথে সেই হিস্ট্রি ক্লাউডেও সেভ রাখে। এর ফলে সুবিধা হলো, ধরুন আপনি এমন এক জায়গায় গিয়েছেন যেখানে ইন্টারনেটের সুবিধা নেই। ইন্টারনেট না থাকা স্বত্বেও আপনি ভার্সন কন্ট্রোল সিস্টেম ব্যবহার করতে পারবেন এবং ভার্সন ট্রাক করতে পারবেন।

এরপর আপনি আবার যখন ইন্টারনেটের সাথে কানেক্টেড হবেন তখন আপনি আপনার কোড রিমোট ক্লাউডে সিঙ্ক করে নিতে পারবেন। দ্বিতীয়ত github এর মতো অত্যন্ত জনপ্রিয় সার্ভিসও git ব্যবহার করে।

অধ্যায়ঃ Github কি?

আজকে আমরা দেখবো github কি এবং কেন github এতো জনপ্রিয়। github হলো git সার্ভিস বেস ওয়েবসাইট, github ওয়েবসাইটের মাধ্যমে আমরা আমাদের কোড ক্লাউডে রাখার সুবিধা দিয়ে থাকে। এর

Git And Github For Beginner

অর্থ এই না যে, github শুধুমাত্র ক্লাউডে আমাদের কোড রাখার জন্য ক্লাউড স্টোরেজ দিয়ে থাকে।

বরং github পুরোপুরি ভার্সন কন্ট্রোল সিস্টেম, যা powered by git। এখন দেখা যাক কিভাবে github ব্যবহার করতে হয়। github ব্যবহার করার জন্য প্রথমে আমাদেরকে github dot com ওয়েবসাইটে যেতে হবে। github আমাদেরকে আমাদের কোড আপলোড এবং আমাদের আপলোড করা কোড অন্যকারো সাথে contribute করার সুযোগ দিয়ে থাকে।

github থেকে এই সমস্ত সুবিধা পেতে হলে, সর্বপ্রথমে আমাদেরকে github dot com ওয়েবসাইটে রেজিস্ট্রেশন বা সাইন আপ করে নিতে হবে। ইউজারনেম, ইমেইল ও পাসওয়ার্ড দিয়েই github এ সাইন আপ করা যায়। আগের থেকেই github এ আমার একাউন্ট থাকাই github এ আমি সরাসরি লগইন করছি। লগইন করলে github ড্যাশবোর্ডে এ নিয়ে যাবে। এখানে বিভিন্ন সেকশন দেখতে পাবেন। যেমন topbar menu, top repository, কোন organization এর সাথে যুক্ত থাকলে organization, work list, changes list দেখতে পাবেন।

টপবার মেনুর উপরে ডানপাশে প্লাস সাইন দেয়া মেনুতে ক্লিক করলে সবমেনু ওপেন হবে। ওপেন হওয়া সব মেনু থেকে new repository মেনুতে ক্লিক করলে আমরা নতুন git repository তৈরী করতে পারব। টপবারের মেনুতে প্লাস মেনুর ডানপাশে থাকা প্রোফাইল আইকনের উপর ক্লিক করলে সাবমেনু ওপেন হবে। সাবমেনু থেকে Your profile মেনুতে গেলে আপনার github প্রোফাইল ওপেন হবে। Your profile মেনুর নিচে থাকা Your repositories মেনুতে ক্লিক করলে, আপনার সমস্ত repository list আকারে দেখাবে। এখন থেকে যেকোন repository ওপেন করতে পারবেন। আমি আমার github repository list এ আগেই তৈরী করা rzasel-git-tutorial নামের repository দেখতে পাচ্ছি। যেখানে tutorial হিসেবে দেখানোর জন্য আমি আগে থেকেই কিছু কোড ফাইল উপলোড করে রেখেছি। rzasel-git-tutorial নামের repository তে গেল repository এ থাকা আমাদের কোড ফাইল readme ফাইল, লাইসেন্স ফাইল এবং ডিরেক্টরি গুলো দেখতে পাচ্ছি।

repository তে থাকা ফাইল এবং ডিরেকটরী লিস্টের উপরে থাকা commits লিঙ্কে ক্লিক করলে এই repository র সমস্ত commit হিস্টোরি ডেট অনুসারে দেখতে পাবো। অর্থাৎ commit করা সকল ভার্সনের সমস্ত কোড github এর ক্লাউডে স্টোর করা থাকবে। এখন আমরা যেকোন সময় যদি মনে করি, আমরা পূর্ববর্তী যেকোন কমিটে ফিরে যেতে চাই; তাহলে যেকোন সময় খুব সহজে পূর্ববর্তী কমিটে ফিরে যেতে পারবো। rzasel-git-tutorial নামের repository থাকা একই কোড বেস আমার লোকাল কম্পিউটারে সিঙ্ক করে কাজ করতে পারব।

তাহলে github ভার্সন কন্ট্রলের সাথে আমাদেরকে ক্লাউড স্টোরেজ সুবিধাও দিয়ে থাকে। ক্লাউড স্টোরেজ সুবিধা দেওয়ায় যে কেউ যেকোন সময়, যেকোন জায়গা থেকে একসেস করতে পারি।

একই সময়ে অন্য যেকোন কম্পিউটার থেকে লগইন করে আপনি আপনার মতো কোড করে github এ আপনার কোড push দিতে পারেন। এবং github powered by git। Repository বলতে আমরা এমন একটা স্থান বা container কে বুঝি যেখানে আমরা কোন কিছু store করতে পারি।

এখানে মূলত আমাদের প্রজেক্টের ফাইল এবং ফোল্ডারগুলো থাকে। এখন আমরা দেখব কিভাবে github এ নতুন রিপোজিটরি তৈরী করা যায় এবং কিভাবে আমাদের কোড github এ push দিতে হয়।

আমরা github dot com এ লগইন করলে উপরের মেনুর ডানপাশ plus sign দেয়া মেনু দেখতে পাব। এই plus sign দেয়া মেনুতে ক্লিক করলে new repository নামের একটি মেনু দেখতে পাব। new repository মেনুতে ক্লিক করলে আমাদেরকে Create a new repository নামের page এ নিয়ে যাবে। Create a new repository page এ Repository নাম দিতে হবে। Repository নাম, যেকোন নাম দিতে পারেন। আমি এখানে Repository নাম দিয়েছি rzasel-tutorial, আপনি আপনার মতো যেকোন নাম দিতে পারেন। Repository নাম directory এর নামে যে সমস্ত character সাপোর্ট করে সেই সমস্ত character দিতে

Git And Github For Beginner

হবে। এবং Repository নাম unique হতে হবে। অর্থাৎ আপনার github এ একই নাম একাধিক repository হতে পারবে না। তবে আমি সাধারণত space এর পরিবর্তে dash ব্যবহার করি। এবং repository 'র' নামে lower case character ব্যবহার করি।

এর পর description অংশে description দিতে হবে, তবে description পাঁচটি optional, চাইলে নাও দিতে পারেন। তবে পরবর্তীতে বোঝার সুবিধার্থে description দেয়া ভালো। এরপর repository টি private হবে না public হবে সেইটা দিতে হবে। এখানে বলে রাখা ভালো কোন repository public হলে সেই সব repository publicly যেকোনো access করতে পারবে। কিন্তু private করলে, আপনার সেই সমস্ত repository আপনি লগইন করা ছাড়া access করতে পারবেন না। এরপর readme file থাকবে কিনা তা বলে দিতে হবে। শুরুতেই readme file না দিলে পরবর্তীতে manually readme file দেয়া যাবে।

এরপর dot gitignore file থাকবে কিনা, থাকলে কি ধরনে dot gitignore file তা সিলেক্ট করে দিতে হবে। আপনার প্রজেক্টটি যে language এর সেই ধরনে dot gitignore সিলেক্ট করে দেয়া ভালো। এবং সব শেষে আপনার কোড বেজের license কি ধরনের হবে তা সিলেক্ট করে দিতে হবে। আপনার কোড অন্যকেউ ব্যবহার করতে চাইলে আপনার কোড ব্যবহার করে সে modify বা commercially ব্যবহার করতে পারবে কিনা তা এই license দিয়ে বোঝা যায়। তবে এগুলো আপনি শুরুতেই না দিলেও হবে। পরবর্তীতে আপনি manually এগুলি এ্যাড করে দিতে পারবেন। Mandatory field গুলো দিয়ে create repository বাটনে ক্লিক করলে নতুন repository তৈরী হয়ে যাবে।

Git And Github For Beginner

Git And Github For Beginner

Git And Github For Beginner