**Học Git, sử dụng GitHub, Deploy code lên Vercel**

**A. GIT**

**1.1 Giới thiệu chung về Git.**

* *Git là một hệ thống* ***quản lý phiên bản*** *phân tán.*
* *Cung cấp* ***kho lưu trữ*** *(repository, có thể nói là repo) để chứa toàn bộ lịch sử phiên bản.*
* *Ưu điểm: tốc độ nhanh, đơn giản, phân tán, phù hợp với dụ án lớn nhỏ.*

**1.2 Các thuật ngữ.**

* ***Working directory (Thư mục làm việc):*** *Khu vực chứa dự án mà chúng ta đang làm việc.*
* ***Staging area (Khu vực sắp xếp):*** *Khu vực chứa thông tin thay đổi của các file.*
* ***Repository (Kho lưu trữ):*** *Kho lưu trữ để lưu trữ dữ liệu, lịch sử các phiên bản.*

***Quy trình:*** *Working directory 🡪git add 🡪 Staging area 🡪 Repo.*

**1.3 Các câu lệnh Git sử dụng phổ biến.**

* ***git --version:*** *Xem phiên bản hiện tại được cài trên máy tính.*
* ***git init:*** *Khởi tạo repo cho dự án, chạy câu lệch trong thư mục gốc của dự án.*
* ***git status:*** *Để xem trạng thái của những file đã được thay đổi trong dự án.*
* ***git add ten\_file hoặc git add .*** *: Chuyển các file đã thay đổi từ vùng Working sang vùng Staging, vùng Staging có tác dụng sắp xếp lại những file đã thêm vào.*
* ***Sự khác nhau giữa git add ten\_file và git add . :******git add .*** *thì sẽ add tất cả các file vào* ***vùng staging*** *còn* ***git add tên\_file*** *thì sẽ add file mà bạn cần add.*
* ***git commit -m”Nội dung” :*** *Chuyển các file từ vùng staging sang vùng repo, repo có tác dụng tạo ra 1 phiên bản mới.*
* ***git log:*** *Xem lại lịch sử các commit, commit mới sẽ hiện bên trên, commit cũ sẽ hiện bên dưới.*
* ***git show commit\_id :*** *dùng để xem chi tiết commit sẽ hiện bên dưới, để xem commit\_id thì dùng git status sẽ có commit\_id*
* ***git diff:*** *Xem sự thay đổi của một file sau khi chỉnh sửa, điều kiện là file đó vẫn đang ở khu vực Working.*
* ***gitk :*** *Mở dashboard xem trực quan hơn.*
* ***git checkout -- ten\_file:*** *Bỏ đi những thay đổi của file, để file đó trở về như lúc ban đầu, áp dụng cho filoe đó đang ở vùng Working.*
* ***git reset HEAD tên\_file hoặc git reset ten\_file :*** *Chuyển file đó từ vùng staging trở lại vùng Working.*
* ***git reset -- soft commit\_id :*** *Chuyển từ trạng thái đã commit về trạng thái trước lúc chạy lệnh git commit, tức từ repo trở lại staging.*
* ***git reset -- mixed commit\_id :*** *Chuyển từ trạng thái commit về trạng thái trước lúc chạy đến git add . , tức là từ Repo về lại Working.*

**1.4 Nhánh trong GIT.**

*Các nhánh đại diện cho các phiên bản của một kho lưu trữ tách ra từ dự án chính, nhánh master là nhánh chính.*

***Các lệnh về nhánh:***

* ***git branch :*** *Xem danh sách các nhánh.*
* ***git checkout -b ten\_nhanh :*** *Tạo một nhánh mới và chuyển sang nhánh đó.*
* ***git checkout ten\_nhanh :*** *Chuyển sang nhánh khác.*
* ***git merge ten\_nhanh :*** *Để merge nhánh khác vào trong nhánh hiện tại.*
* ***git branch -D ten\_nhanh :*** *Để xóa nhánh.*

**B. GITHUB**

* *GitHub là một hệ thống quản lý dự án và phiên bản trực tuyến.*
* *Giúp đồng bộ code cảu cả nhóm lên một kho lưu trữ chung.*

***Các bước đẩy code lần đầu lên GitHub khi project đã có Git:***

* *git remote add origin url\_github\_https*
* *git branch -M main*
* *git push -u origin main*

***Các bước đẩy code lần đầu lên GITHUB khi project chưa có GIT:***

* *git init*
* *git add .*
* *git commit -m “ Nội dung commit”*
* *git branch -M main*
* *git remote add origin url\_github\_https*
* *git push -u origin main*

***Kéo code từ GITHUB về local:***

* *git pull origin ten\_nhanh*

**C. Deploy code lên Vercel (Dành cho code web)**

**Lên website: Vercel.**

**Các bước deploy code lên Vercel:**

* *Chọn “Add new”*
* *Chọn “Project “*
* *Chọn “Add github account” (nếu chưa có)*
* *Bấm deploy*