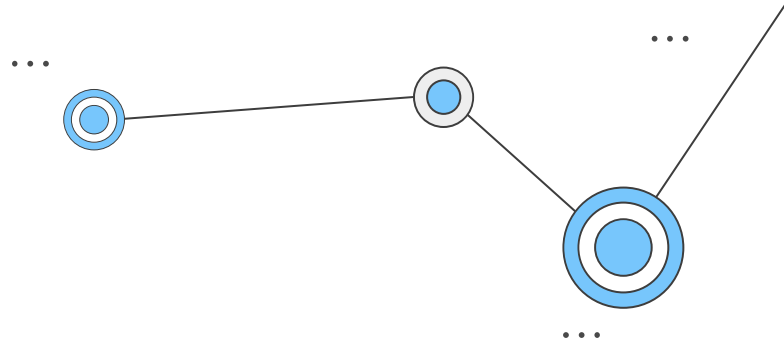


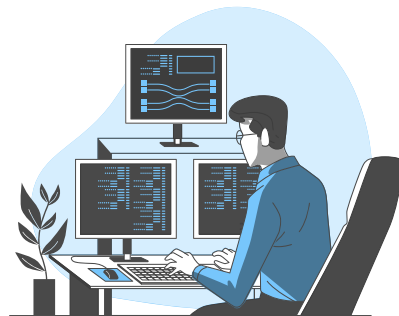


# GitHub



## Khóa học Backend

### Bài 04: Học GIT, GITHUB



# Nội dung

01

...

Học GIT

02

...

Sử dụng GITHUB

03

...

Deploy code lên Vercel





# 01

Học GIT



## 1.1. Giới thiệu chung



# git

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

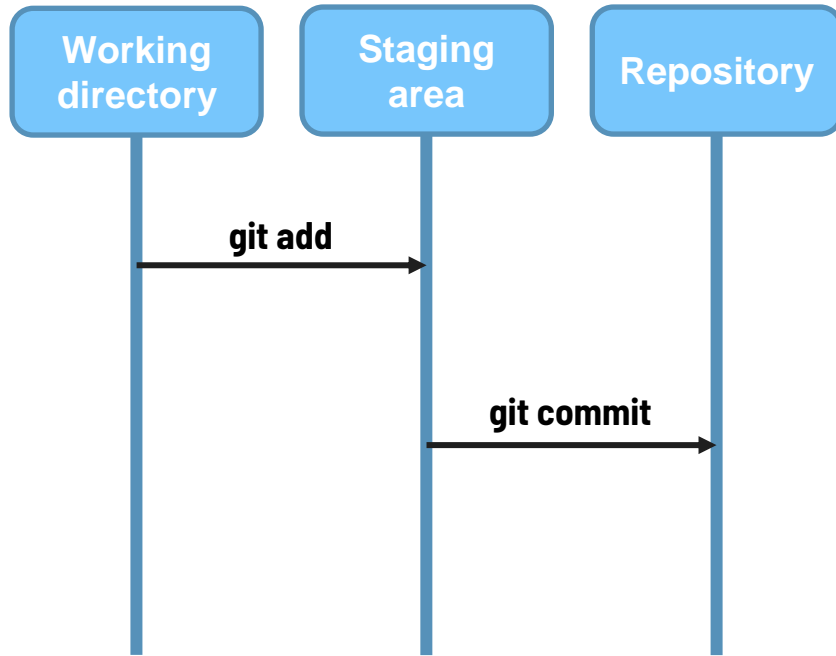
## 1.2. Cài đặt GIT



git

- Link cài đặt: <https://git-scm.com/downloads>

## 1.3. Các thuật ngữ



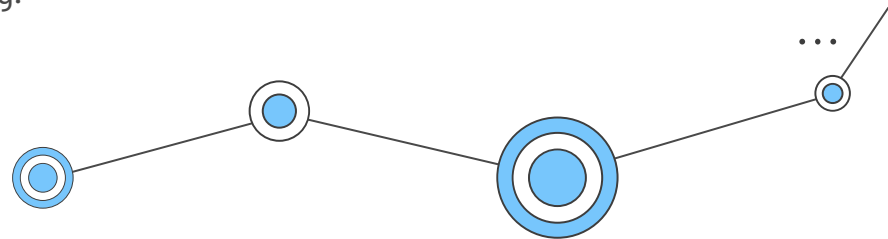
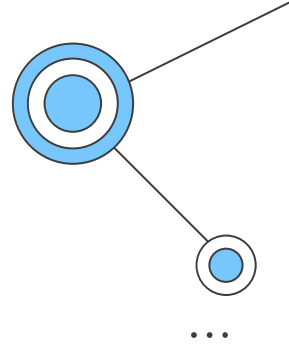
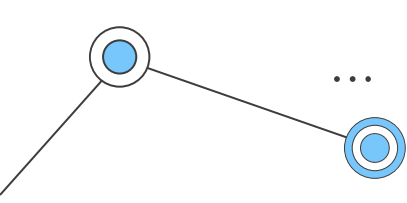
- **Working directory (Thư mục làm việc):** Khu vực chứa dự án mà chúng ta đang làm việc.
- **Staging area (Khu vực sắp xếp):** Khu vực chứa thông tin thay đổi của các file.
- **Repository (Kho lưu trữ):** Kho lưu trữ để lưu trữ dữ liệu, lịch sử các phiên bản.

## 1.4. Các câu lệnh GIT sử dụng phổ biến

- **git --version**
  - Xem phiên bản GIT hiện tại đang được cài trên máy tính.
- **git init**
  - Khởi tạo repository (kho lưu trữ) cho dự án.
  - Chạy câu lệnh 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**.
  - Staging area có tác dụng sắp xếp lại những file đã thêm vào.

## 1.4. Các câu lệnh GIT sử dụng phổ biến

- **git commit -m "Nội dung..."**
  - Chuyển các file từ vùng Staging sang vùng Repository.
  - Repository 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 một commit.
- **git diff**
  - Xem sự thay đổi của một file sau khi chỉnh sửa.
  - Điều kiện là file đó vẫn đang ở khu vực Working.





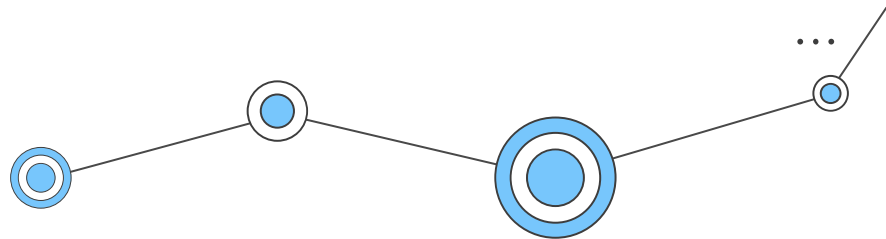
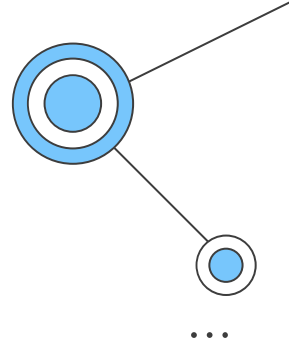
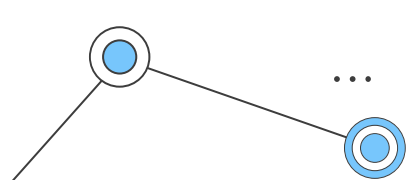
## 1.4. Các câu lệnh GIT sử dụng phổ biến

- **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 file đang ở vùng Working.
- **git reset HEAD ten\_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 là từ Repository về lại Staging.

## 1.4. Các câu lệnh GIT sử dụng phổ biến

- **git reset --mixed commit\_id**

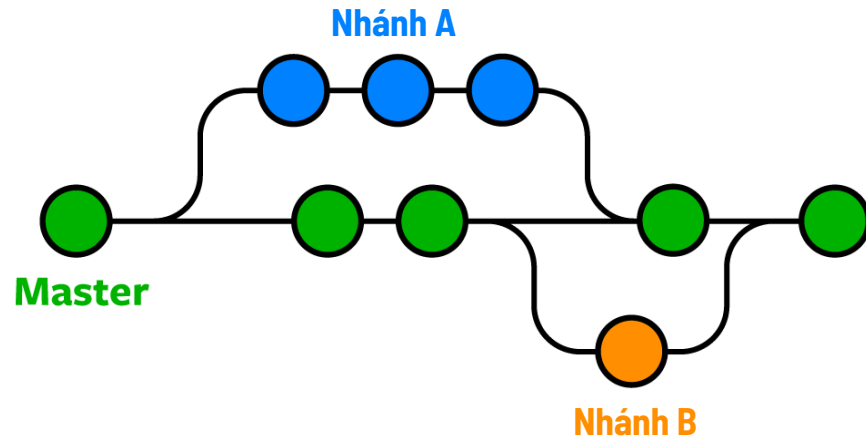
- Chuyển từ trạng thái đã commit về trạng thái trước lúc chạy lệnh git add.
- Tức là từ Repository về lại Working.



## 1.4. Các câu lệnh GIT sử dụng phổ biến

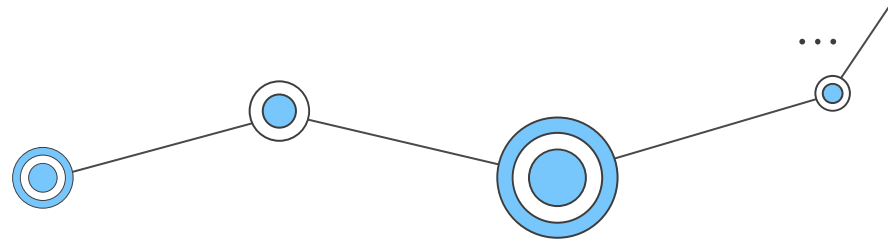
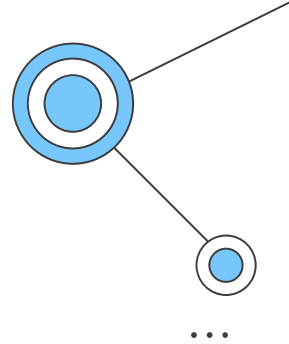
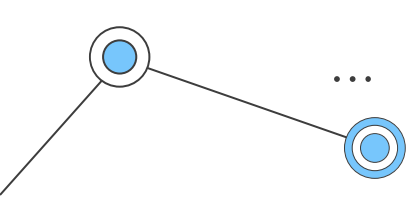
- **Nhánh trong GIT**

- Các nhánh đại diện cho các phiên bản của một kho lưu trữ tách ra từ dự án chính.
- Nhánh master là nhánh chính, để sau này deploy lên server.



## 1.4. Các câu lệnh GIT sử dụng phổ biến

- **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.



# 02

## Sử dụng GITHUB

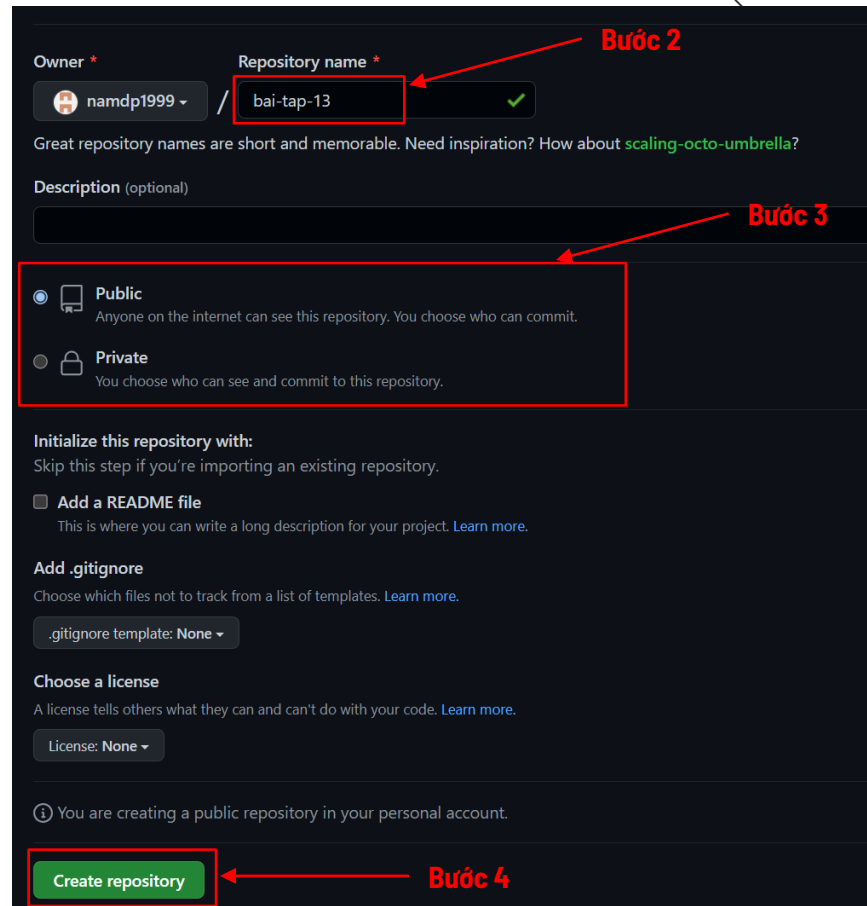
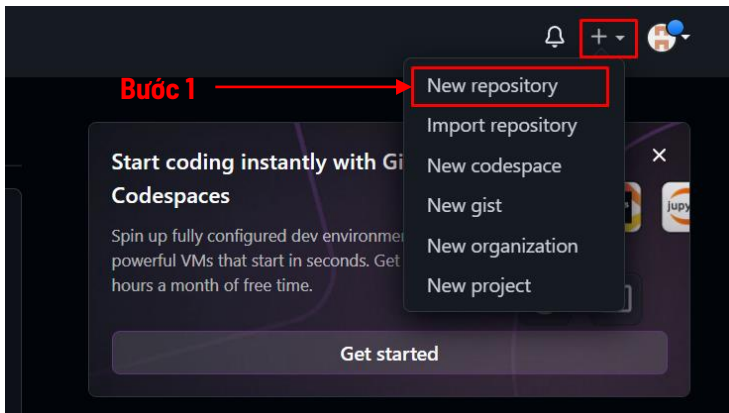
## 2.1. Giới thiệu chung



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

## 2.2. Tạo tài khoản và tạo Repository

- Link website: <https://github.com/>
- Cách tạo mới một Repository:



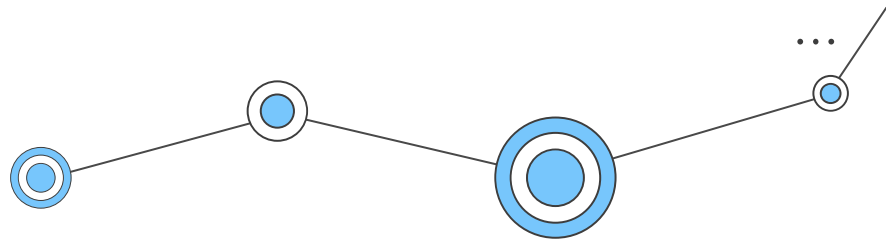
## .2.3. Đẩy code lần đầu lên GITHUB khi project đã có GIT

- **Bước 1:** git remote add origin url\_github\_https
- **Bước 2:** git branch -M main
- **Bước 3:** git push -u origin main



## 2.4. Đẩy code lần đầu lên GITHUB khi project chưa có GIT

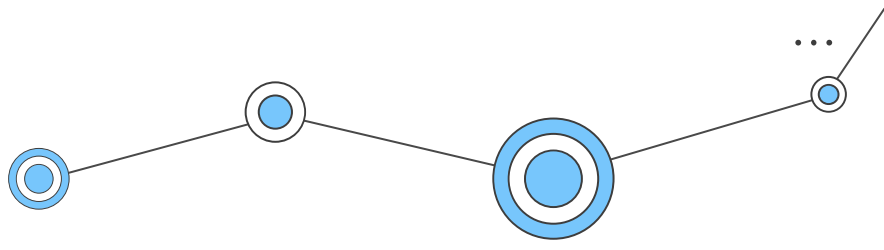
- **Bước 1:** git init
- **Bước 2:** git add .
- **Bước 3:** git commit -m "Nội dung commit"
- **Bước 4:** git branch -M main
- **Bước 5:** git remote add origin url\_github\_https
- **Bước 6:** git push -u origin main





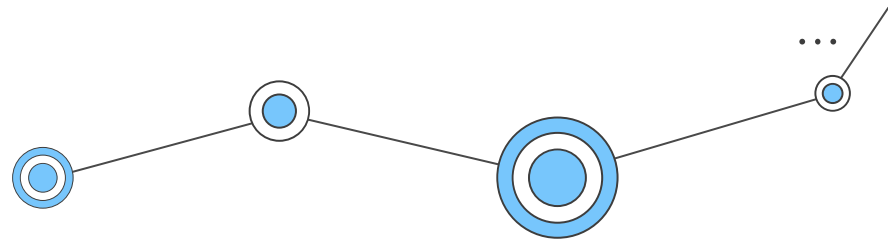
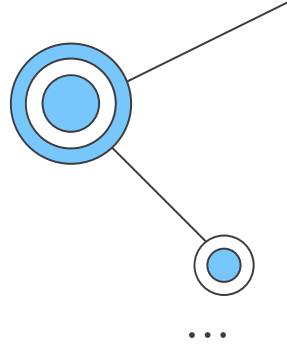
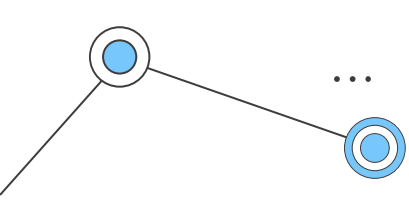
## ... 2.5. Đẩy code lên GITHUB cho các lần tiếp theo



- **Bước 1:** git add .
  - **Bước 2:** git commit -m "Nội dung commit"
  - **Bước 3:** git push
- 

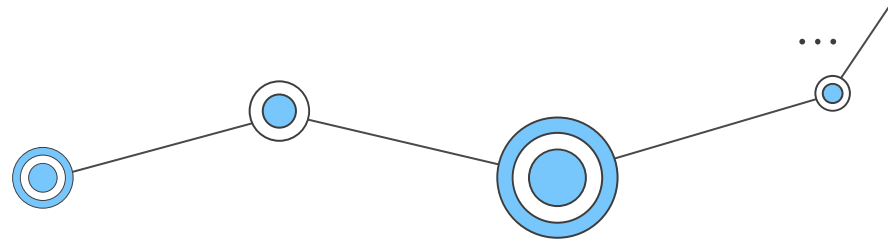
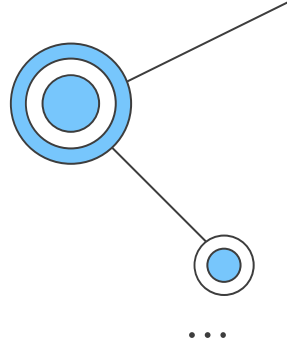
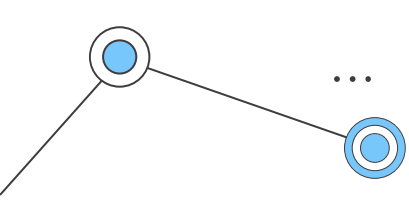
## 2.6. Kéo code từ GITHUB về máy

- **Bước 1:** git pull origin ten\_nhanh



## 2.6. Kéo code từ GITHUB về máy

- **Bước 1:** git pull origin ten\_nhanh

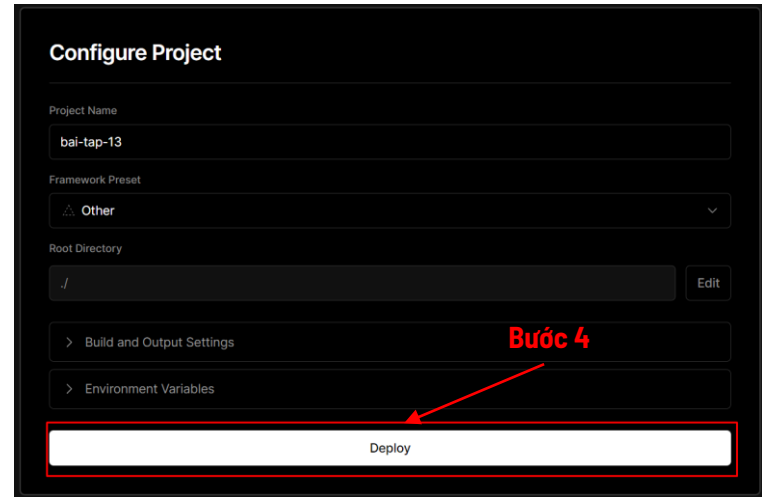
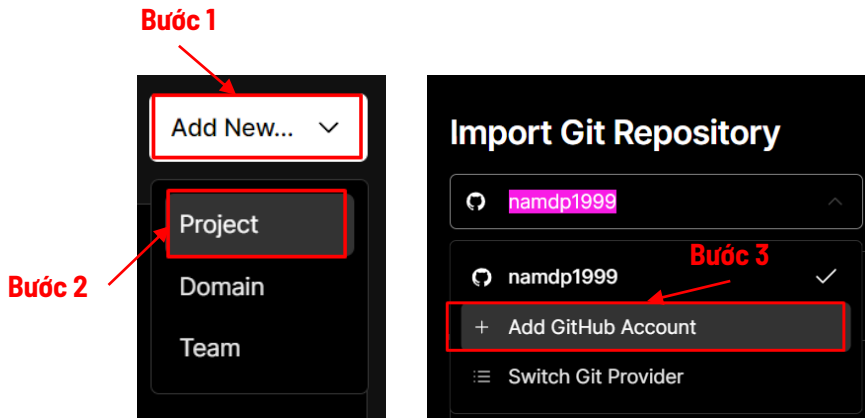


# 03

**Deploy code lên Vercel**

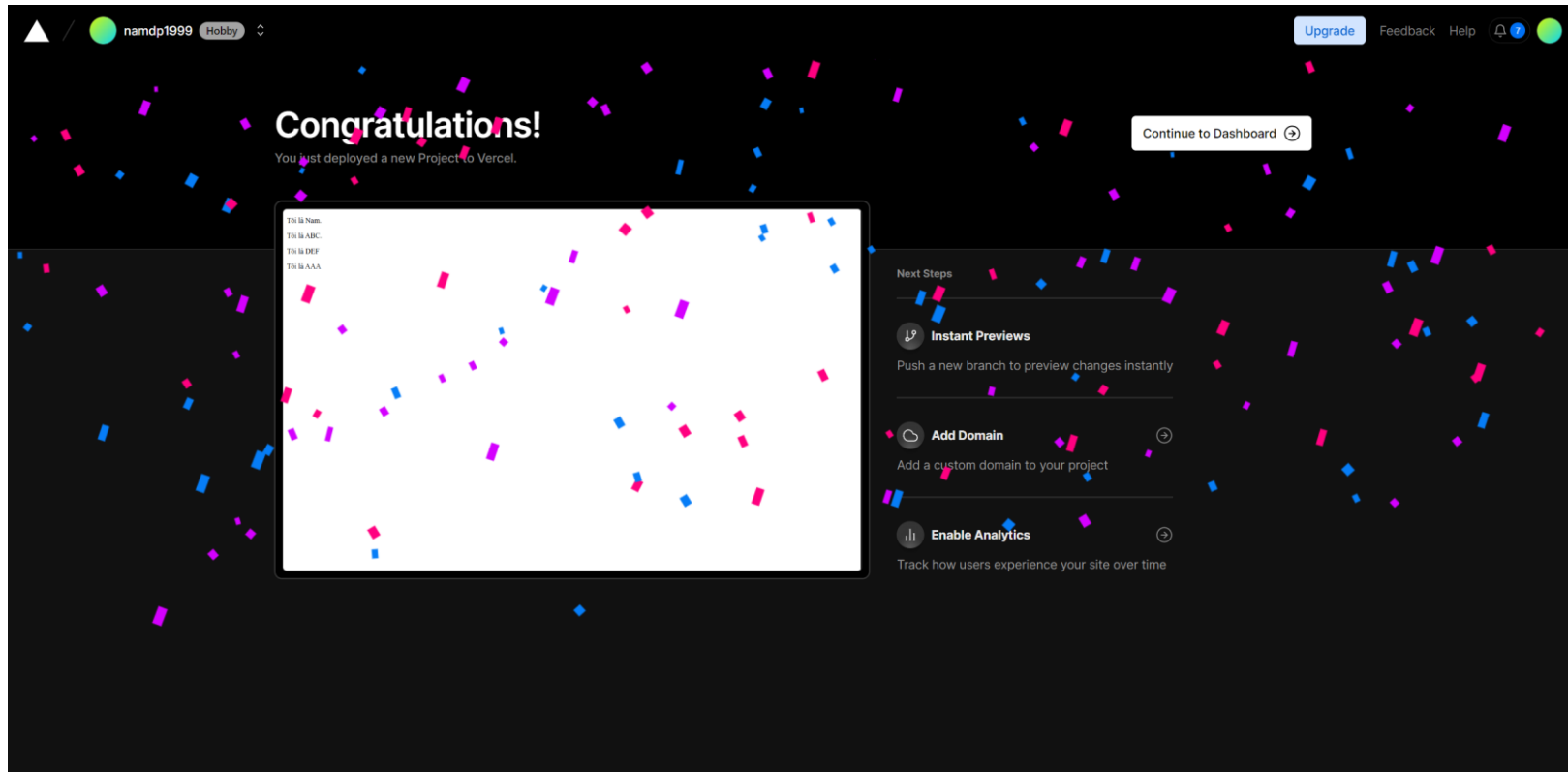
### 03. Deploy code lên Vercel

- Link website: <https://vercel.com/>
- Các bước deploy code lên Vercel:
  - **Bước 1:** Chọn "Add New..."
  - **Bước 2:** Chọn "Project"
  - **Bước 3:** Chọn "Add Github Account" (nếu chưa có)
  - **Bước 4:** Bấm Deploy



## 03. Deploy code lên Vercel

- Sau khi deploy thành công:



# Bài tập

- **Câu 1:** Đẩy hết các bài tập sau lên GITHUB, mỗi bài tập là một project riêng biệt (tức là mỗi bài là một Repository).
  - Bài 1
  - Bài 2
  - Bài 3
- **Câu 2:** Deploy các project đó lên trên Vercel.
- **Câu 3:** Chèn link bài tập (link Vercel) vào bảng Excel (gửi link sau).

