



TRƯỜNG CAO ĐẲNG VĂN LANG
KHOA: CÔNG NGHỆ THÔNG TIN



PHÁT TRIỂN DỰ ÁN PHẦN MỀM



Bài Giảng



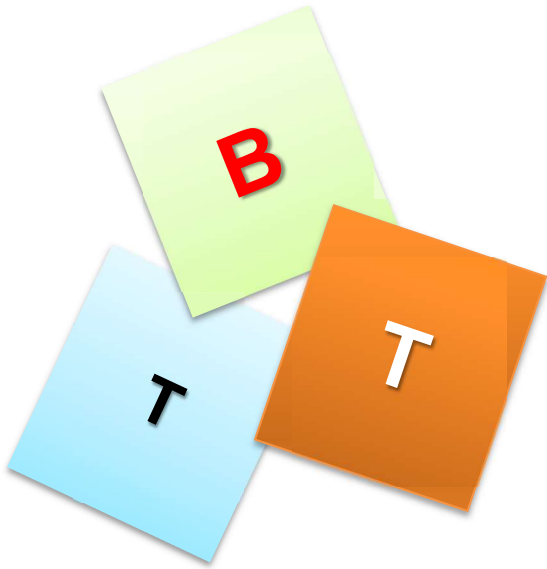
TRƯỜNG CAO ĐẲNG VĂN LANG
KHOA: CÔNG NGHỆ THÔNG TIN



Chương 4. Công nghệ xây dựng Front-end



Bài Giảng



4.1. Tổng quan về Phát triển Front-end

1. Khái niệm Front-end

- **Front-end** là phần **giao diện người dùng (User Interface - UI)** trong ứng dụng web, nơi người dùng trực tiếp tương tác.
- Gồm các thành phần như: bố cục trang, biểu mẫu, nút bấm, menu, bảng dữ liệu...
- Front-end kết nối với **Back-end** thông qua API (REST, GraphQL...).

2. Vai trò trong phát triển phần mềm

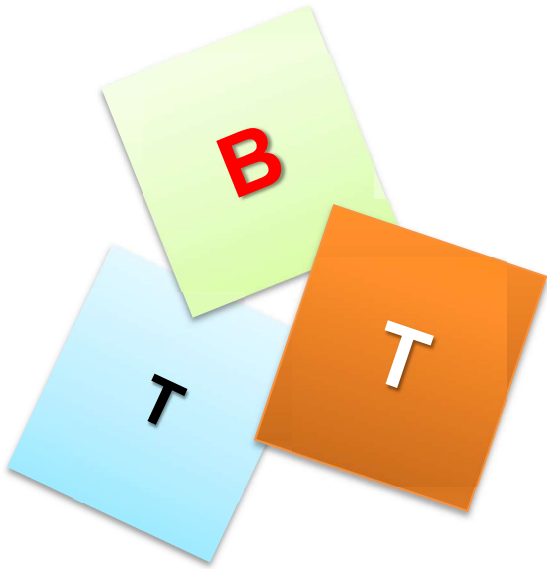
- Giao diện đẹp, dễ dùng giúp:
 - Tăng trải nghiệm người dùng (UX).
 - Giảm lỗi thao tác.
 - Tăng tính chuyên nghiệp và khả năng tiếp cận hệ thống.

3. Thành phần chính của Front-end

Thành phần	Mô tả
HTML	Khung xương của trang web – định nghĩa cấu trúc nội dung.
CSS	Tạo kiểu giao diện: màu sắc, font, bố cục, responsive.
JavaScript	Điều khiển hành vi, xử lý sự kiện, tương tác động với người dùng.

4. Công nghệ phổ biến trong Front-end

Công nghệ	Mục đích	Ghi chú
Bootstrap	Thư viện CSS giúp responsive	Dễ tích hợp, dùng nhanh.
ReactJS	Thư viện UI component-based	Hiện đại, do Meta phát triển.
VueJS	Framework thân thiện	Dễ học, nhẹ.
Angular	Framework toàn diện	Do Google phát triển.
Tailwind CSS	CSS utility-first	Tùy biến giao diện rất linh hoạt.



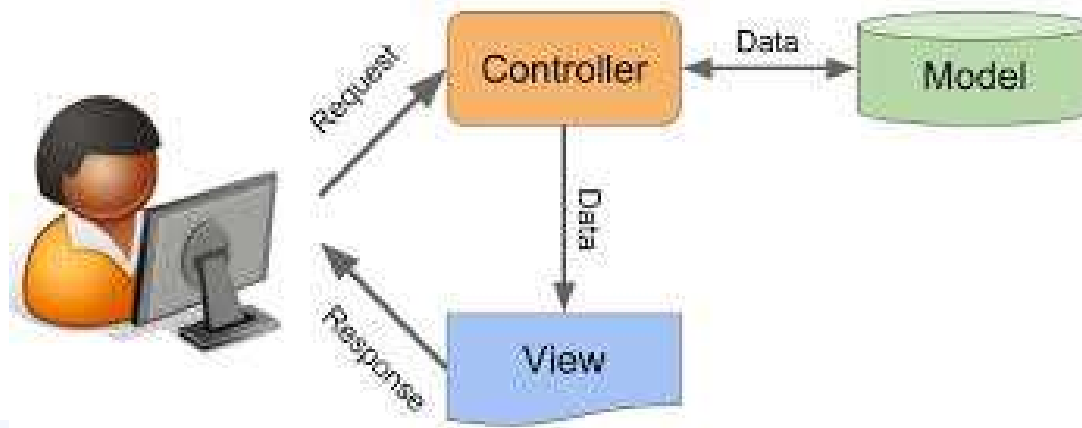
4.2. .NET MVC: Kiến trúc và Triển khai

1. Khái niệm về .NET MVC

- **MVC (Model – View – Controller)** là một mẫu kiến trúc phần mềm được Microsoft triển khai trong **ASP.NET MVC Framework**.
- Tách biệt ba thành phần chính:
 - **Model:** Quản lý dữ liệu và nghiệp vụ.
 - **View:** Hiển thị dữ liệu – giao diện người dùng.
 - **Controller:** Xử lý logic, điều hướng yêu cầu giữa Model và View.

2. Sơ đồ kiến trúc

- Người dùng gửi yêu cầu đến Controller.
- Controller gọi Model để xử lý nghiệp vụ và dữ liệu.
- Model trả dữ liệu → Controller truyền dữ liệu cho View.
- View hiển thị thông tin ra giao diện.



3. Cấu trúc thư mục dự án ASP.NET MVC

Thư mục	Chức năng
Controllers/	Chứa các file điều khiển xử lý logic
Models/	Chứa các lớp đại diện cho dữ liệu
Views/	Chứa giao diện (cshtml) của ứng dụng
wwwroot/	Tài nguyên tĩnh: CSS, JS, ảnh
appsettings.json	Cấu hình kết nối DB, môi trường

4. Quy trình xử lý trong ASP.NET MVC

- trình duyệt gửi yêu cầu đến URL: /Student/Details/1
- ASP.NET định tuyến đến StudentController.cs
- Trong Details(int id) gọi dữ liệu từ StudentModel
- Kết quả truyền vào View("Details", model)
- Giao diện Details.cshtml được hiển thị với dữ liệu

Ví dụ

```
public class StudentController : Controller
{
    public IActionResult Details(int id)
    {
        var student = new Student { Id = id, Name =
"Nguyễn Văn A" };
        return View(student);
    }
}
```

6. Ưu điểm của ASP.NET MVC

- Phân tách rõ ràng giữa UI, dữ liệu và xử lý.
- Hỗ trợ unit test dễ dàng.
- Tích hợp tốt với các công nghệ Microsoft (.NET, SQL Server).
- Hỗ trợ lập trình hướng đối tượng đầy đủ.

7. Nhược điểm

- Cần hiểu sâu về C# và .NET.
- Cấu hình nhiều hơn so với Razor Pages hoặc Web Forms.
- Chạy tốt nhất trên

8. Công cụ phát triển

- IDE: Visual Studio (hoặc Visual Studio Code cho .NET Core)



4.3. React – Xây dựng giao diện người dùng động

1. Giới thiệu về React

- **React** là thư viện JavaScript mã nguồn mở, phát triển bởi **Facebook**, dùng để xây dựng giao diện người dùng (UI).
- Tập trung vào phát triển **Single Page Applications (SPA)** với khả năng **render động** và **phản hồi nhanh**.

2. Các đặc điểm nổi bật của React

Đặc điểm	Giải thích
Component-Based	Giao diện chia nhỏ thành các "component" tái sử dụng được.
Virtual DOM	Cơ chế DOM ảo giúp cập nhật giao diện hiệu quả hơn DOM thật.
JSX (JavaScript XML)	Viết HTML trong JavaScript – dễ viết và trực quan hơn.
One-way Data Flow	Dữ liệu truyền một chiều giúp dễ kiểm soát trạng thái.

3. Cài đặt và khởi tạo dự án React

- **Cài đặt Node.js :**
 - <https://nodejs.org>
- **Khởi tạo dự án React:**
 - `npx create-react-app my-app`
 - `cd my-app`
 - `npm start`

4. Kiến trúc ứng dụng React

Thành phần	Vai trò
App.js	Thành phần chính (root component).
Component con	Các phần UI nhỏ hơn như Header, Footer, Form, Button,...
State & Props	Quản lý dữ liệu trong component (state) và truyền dữ liệu (props).
Event Handling	Xử lý các sự kiện người dùng như click, nhập liệu,...

Ví dụ

```
import React from 'react';
import Welcome from './Welcome';

function App() {
  return (
    <div>
      <h1>Chào mừng đến với React</h1>
      <Welcome name="Nguyễn Văn A" />
    </div>
  );
}

export default App;
```

```
function Welcome(props) {
  return <p>Xin chào, {props.name}!</p>;
}

export default Welcome;
```

Ưu – Nhược điểm của React

Ưu điểm	Nhược điểm
Nhanh, phản hồi thời gian thực	Học JSX ban đầu có thể lạ
Cộng đồng mạnh, thư viện phong phú	Phải quản lý nhiều thư viện phụ trợ
Tái sử dụng component hiệu quả	Cần hiểu kỹ về state, lifecycle

Khi nào nên chọn React?

- Dự án cần giao diện tương tác mạnh, thời gian thực (chat, dashboard, v.v).

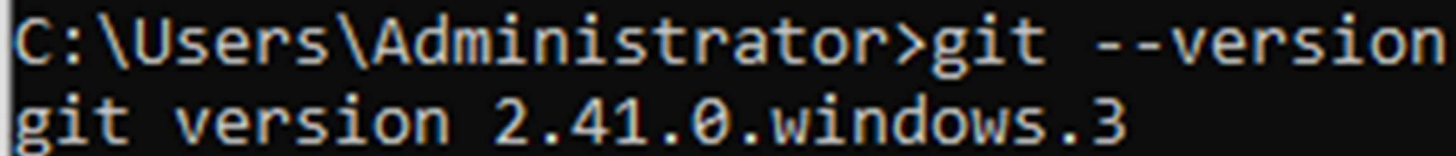


Tích hợp GitHub với Visual Studio Code (VS Code)

1. Cài đặt công cụ cần thiết

1.1. Cài đặt Git

- Tải tại: <https://git-scm.com/downloads>
- Cài xong, mở Terminal (hoặc CMD) gõ:
- `git --version`
- Nếu hiển thị phiên bản là đã cài thành công.

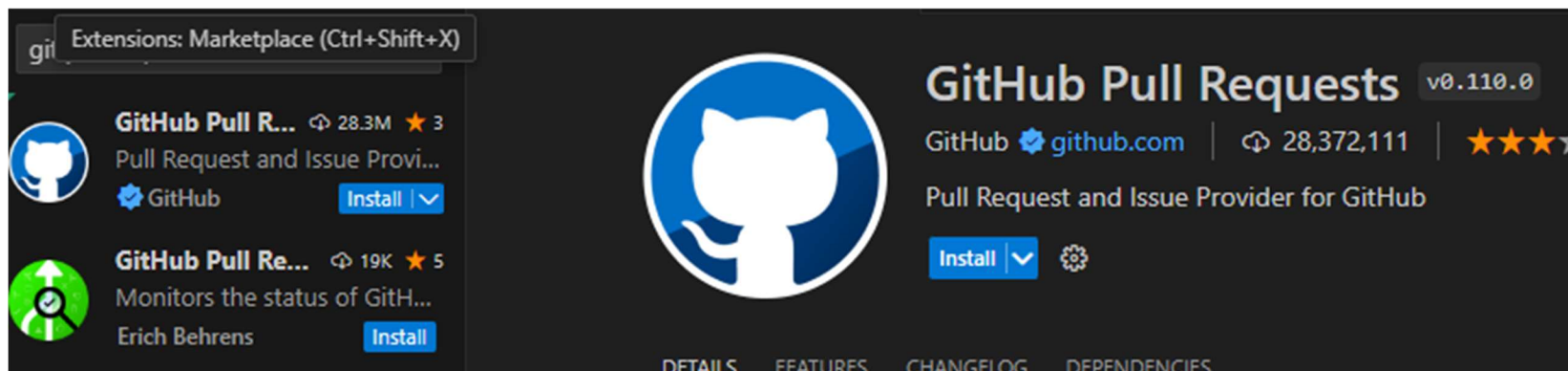


```
C:\Users\Administrator>git --version  
git version 2.41.0.windows.3
```

- **1.2. Cài đặt VS Code**
- Tải tại: <https://code.visualstudio.com>
- Cài xong, mở VS Code.

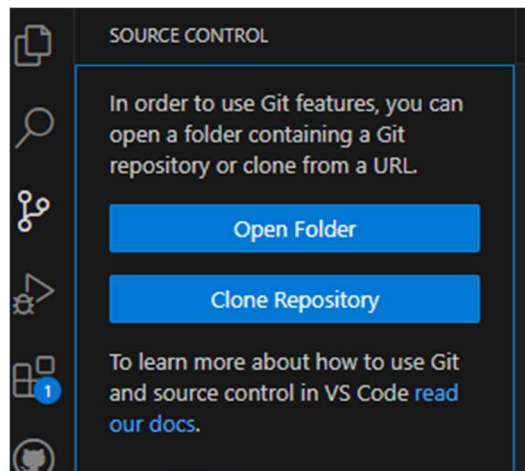
1.3. Cài đặt tiện ích GitHub trong VS Code

- Vào VS Code → Extensions (Ctrl+Shift+X)
- Tìm: GitHub Pull Requests and Issues
- Nhấn Install



2. Kết nối VS Code với tài khoản GitHub

- Mở VS Code → Bấm vào **Source Control** (**Ctrl+Shift+G**).
- Bấm vào nút **"Clone Repository"** hoặc mở terminal nhập:
 - `git config --global user.name "Tên"`
 - `git config --global user.email "email@gmail.com"`



3. Các thao tác Git cơ bản trong VS Code

Thao tác	Mô tả
git init	Khởi tạo repo Git mới trong thư mục
git clone [url]	Tải về repo từ GitHub
git add .	Thêm tất cả thay đổi vào staging
git commit -m "ghi chú"	Commit thay đổi
git push origin main	Đẩy code lên GitHub
git pull origin main	Kéo code mới từ GitHub về máy

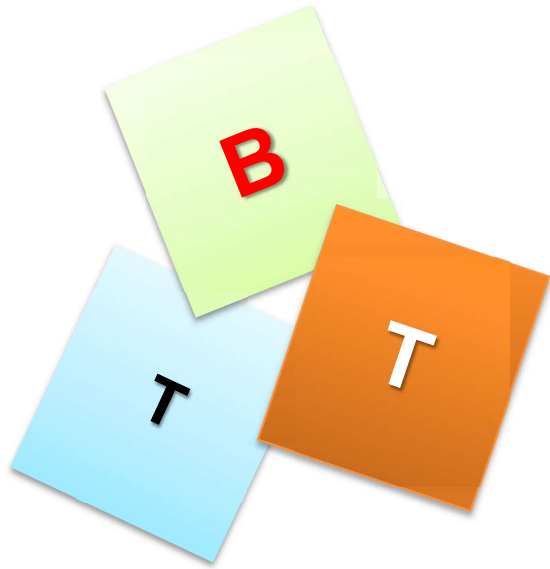
4. Thao tác trực tiếp với GitHub trong VS Code

4.1. Clone một repo GitHub:

- Vào GitHub, chọn một repo → Copy link HTTPS.
- Trong VS Code:
 - Ctrl+Shift+P → Git: Clone
 - Dán link → Chọn thư mục lưu → VS Code mở repo.

4.2. Thêm, commit và push:

- Thêm file hoặc sửa.
- Mở tab Source Control → "+" để stage thay đổi.
- Nhập nội dung commit → Click ✓ để commit.
- Click nút "..." → **Push** để đẩy lên GitHub.



quản lý nhóm dự án sử dụng GitHub

1. Tạo Repository dùng chung trên GitHub

1. Một thành viên (Leader) tạo repo tại:
<https://github.com/new>
2. Cấu trúc repo đề xuất:
 - /do-an-phan-mem/
 - frontend/
 - backend/
 - database/
 - docs/
 - báo cáo, phân tích yêu cầu
 - README.md
 - .gitignore
3. Vào tab **Settings** → **Collaborators** → Mời các thành viên khác bằng email hoặc tên GitHub.

2. Phân công công việc bằng nhánh (branches)

Thành viên	Nhánh gợi ý	Nội dung phụ trách
Sinh viên A	frontend-react	Giao diện người dùng bằng React
Sinh viên B	backend-node	Xây dựng API với Node.js
Sinh viên C	database-modeling	Thiết kế & viết script CSDL
Sinh viên D	docs-report	Viết báo cáo, tài liệu mô tả

Mỗi người **làm việc trên nhánh riêng** tránh đụng code nhau.

3. Quy trình làm việc chuẩn CI/CD mini

■ Mỗi người:

1. Clone repo:

- git clone <https://github.com/ten-nhom/ten-do-an.git>

2. Tạo nhánh mới từ main:

- git checkout -b backend-node

3. Code, commit:

1. git add .
2. git commit -m "Tạo API đăng ký người dùng"
3. git push origin backend-node

- Leader:
 - Mở GitHub → Tab Pull Requests
 - Review code các nhánh trước khi merge vào main
 - Kiểm tra xung đột, test chạy
 - Merge → Gửi thông báo cho nhóm






4. Theo dõi công việc với Issues và Projects

■ Issues:

- Ghi nhận task nhỏ (ví dụ: "Thiết kế giao diện trang đăng nhập")
- Có thể gán người thực hiện, thời hạn

- Projects (Kanban board):
- Giống như Trello ngay trong GitHub
- Tạo các cột:
 - To do
 - In progress
 - Review
 - Done
- Giao task, kéo thả để theo dõi tiến độ

5. Một số quy tắc làm việc nhóm hiệu quả

Quy tắc	Mô tả
 Giao tiếp	Dùng GitHub Discussions hoặc Zalo nhóm
 Pull thường xuyên	Tránh xung đột khi merge
 Commit rõ ràng	Ghi nội dung ngắn gọn: "Thêm validate form đăng ký"
 Đặt tên folder chuẩn	Không viết hoa, không dấu, ví dụ sinhvien-list/
Test code	Chạy code trước khi đẩy lên main
 Cập nhật README	Luôn ghi chú hướng dẫn chạy dự án



**Thực hành xây dựng giao diện với
các công nghệ khác nhau**