CHƯƠNG 1: NHỮNG ĐIỀU CƠ BẢN VỀ WEB.

**1.Lập trình web là gì?**

-Lập trình web(**Web Development) là quá trình sử dụng các ngôn ngữ lập trình để xây dựng và phát triển các trang web hoặc ứng dụng web. Công việc này bao gồm việc tạo ra giao diện người dùng (frontend) và xử lý dữ liệu, logic phía máy chủ (backend).**

-Nếu làm cả 2 công việc frontend( giao diện người dùng) và backend(Lập trình phía máy chủ) thì gọi là Full Stack hay Full Stack Developer, Được ghép từ “full”(toàn bộ) và “stack”(nhóm).

+Phần Frontend là phần người dùng nhìn thấy và tương tác trực tiếp, bao gồm giao diện người dùng (User Interface-UI) và trải nghiệm người dùng (User Experience-UX).

+Phần Backend là phần nằm sau màn hình, xử lý các yêu cầu của người dùng và cung cấp dữ liệu cho Frontend.

**2.Học gì để code web?**

**-Frontend** sử dụng **HTML**, **CSS**, và **JavaScript** để xây dựng giao diện và tương tác với người dùng.

3 thành phần quan trọng của một trang web

- HTML (HyperText Markup Language): Đây là ngôn ngữ đánh dấu giúp xây dựng cấu trúc cơ bản của một trang web. HTML cho phép bạn tạo các phần tử như tiêu đề(Heading), đoạn văn(Paragraph), hình ảnh(image), liên kết(link), bảng (table)và biểu mẫu(Form).  
- CSS (Cascading Style Sheets): Đây là công cụ giúp bạn thiết kế và định dạng giao diện website, làm cho trang web trở nên chuyên nghiệp và thân thiện với người dùng. CSS cung cấp nhiều công cụ để tạo hiệu ứng động, responsive, giúp trang web hiển thị tốt trên nhiều thiết bị.  
- JavaScript: Đây là một ngôn ngữ lập trình giúp tạo hiệu ứng động, xử lý các sự kiện trên website và xây dựng các tính năng tương tác với người dùng. Với JavaScript, bạn có thể tạo ra các thanh trượt hình ảnh, hiệu ứng cuộn trang, form đăng nhập động, và nhiều tính năng nâng cao khác.  
Có thể nói rằng, HTML đóng vai trò như "bộ xương" cung cấp cấu trúc nội dung; CSS là "làn da" mang đến diện mạo về màu sắc và phong cách cho trang web; JavaScript là "bộ não" tạo sự tương tác và logic vận hành. Khi kết hợp, chúng tạo ra một trải nghiệm người dùng liền mạch và hấp dẫn.

-BackEnd là tất cả những phần hỗ trợ hoạt động cho một Website hoặc một ứng dụng. Đây là phần mà người dùng khi lướt Web hay trải nghiệm ứng dụng không thể nhìn thấy được. Tuy nhiên, BackEnd được ví như cơ quan đầu não thực hiện những yêu cầu, câu lệnh và lựa chọn thông tin phù hợp để hiển thị lên màn hình.

-BackEnd của một Website được xây dựng từ 3 thành phần là máy chủ, ứng dụng và cơ sở dữ liệu. Nhờ có các thành phần này mà Website hoạt động hiệu quả, các vấn đề như lưu trữ dữ liệu, truy xuất thông tin chính xác khớp với lệnh đưa ra, giúp người dùng nhanh chóng có được thông tin mình cần.

* **Ngôn ngữ lập trình (Programming Languages)**
* **Vai trò**: Xử lý logic nghiệp vụ, kết nối với cơ sở dữ liệu và thực hiện các thao tác xử lý yêu cầu của người dùng.
* **Ví dụ**: JavaScript (Node.js), Python, Java, PHP, Ruby, C#, Go, C++.
* **Framework Backend (Backend Frameworks)**
* **Vai trò**: Cung cấp cấu trúc và các công cụ hỗ trợ xây dựng ứng dụng nhanh chóng, xử lý các yêu cầu HTTP, quản lý cơ sở dữ liệu và bảo mật.
* **Ví dụ**:
  + **Node.js** (Express.js)
  + **Python** (Django, Flask)
  + **Ruby** (Ruby on Rails)
  + **Java** (Spring)
  + **PHP** (Laravel)
  + **C#** (ASP.NET Core)
* **Cơ sở dữ liệu (Databases)**
* **Vai trò**: Lưu trữ và quản lý dữ liệu của ứng dụng, cung cấp khả năng truy xuất, cập nhật, xóa và phân tích dữ liệu.
* **Ví dụ**:
  + **SQL**: MySQL, PostgreSQL, MS SQL Server
  + **NoSQL**: MongoDB, Cassandra, Redis
* **API (Application Programming Interface)**
* **Vai trò**: Cung cấp giao diện để frontend và các hệ thống khác có thể tương tác với backend. APIs thường được xây dựng theo chuẩn **RESTful** hoặc **GraphQL**.
* **Ví dụ**: Xây dựng các API RESTful với **Express.js**, **Django**, **Flask**.
* **Xử lý xác thực và bảo mật (Authentication & Security)**
* **Vai trò**: Đảm bảo rằng người dùng và các dịch vụ có quyền truy cập hợp lệ và bảo mật. Bao gồm xác thực người dùng, bảo vệ chống tấn công (XSS, CSRF, SQL Injection), mã hóa dữ liệu.
* **Ví dụ**:
  + **JWT** (JSON Web Token)
  + **OAuth** (Đăng nhập qua mạng xã hội)
  + **Bcrypt** (mã hóa mật khẩu)
* **Web Server (Máy chủ web)**
* **Vai trò**: Xử lý các yêu cầu HTTP/HTTPS, phân phối nội dung và ứng dụng đến client.
* **Ví dụ**:
  + **Nginx**
  + **Apache HTTP Server**
  + **Caddy**
* **Công cụ DevOps và Triển khai (DevOps & Deployment Tools)**
* **Vai trò**: Quản lý quá trình triển khai và duy trì ứng dụng trên các máy chủ, giám sát ứng dụng, đảm bảo tính sẵn sàng cao và mở rộng.
* **Ví dụ**:
  + **Docker** (containerization)
  + **Kubernetes** (quản lý container)
  + **CI/CD**: Jenkins, GitLab CI, Travis CI
  + **Cloud Platforms**: AWS, Google Cloud, Azure
* **Caching**
* **Vai trò**: Tăng tốc độ truy xuất dữ liệu và giảm tải cho cơ sở dữ liệu bằng cách lưu trữ dữ liệu tạm thời.
* **Ví dụ**:
  + **Redis**
  + **Memcached**
* **Message Brokers (Message Queues)**
* **Vai trò**: Xử lý các tác vụ nền và giao tiếp giữa các dịch vụ backend trong các hệ thống phân tán.
* **Ví dụ**:
  + **RabbitMQ**
  + **Apache Kafka**
  + **Amazon SQS**
* **Testing (Kiểm thử)**
* **Vai trò**: Đảm bảo rằng các phần của backend hoạt động chính xác, hiệu quả và không có lỗi.
* **Ví dụ**:
  + **Unit Testing**: Jest, Mocha
  + **Integration Testing**: Postman, Supertest
* **Logging & Monitoring**
* **Vai trò**: Giám sát hoạt động của ứng dụng, ghi lại lỗi và các sự kiện quan trọng để phân tích và xử lý sự cố.
* **Ví dụ**:
  + **Logging**: Winston (Node.js), Log4j (Java)
  + **Monitoring**: Prometheus, Grafana, ELK Stack (Elasticsearch, Logstash, Kibana)

**3.Công cụ hỗ trợ.**

Ngoài việc nắm vững các ngôn ngữ lập trình, ta cũng phải làm quen với cá công cụ hỗ trợ

- Trình soạn thảo mã nguồn (Editor): Các công cụ như Visual Studio Code, Sublime Text, Atom cung cấp môi trường làm việc thuận tiện, hỗ trợ gợi ý mã nguồn và tích hợp nhiều tính năng mạnh mẽ giúp lập trình viên viết code nhanh hơn.  
- Trình duyệt để kiểm tra (Browser): Khi viết code web, bạn cần kiểm tra trang web hiển thị như thế nào trên trình duyệt. Các trình duyệt như Google Chrome, Firefox có tích hợp công cụ DevTools giúp debug code và tối ưu hiệu suất web.  
- GitHub - Quản lý mã nguồn: GitHub là một nền tảng giúp lập trình viên quản lý mã nguồn, theo dõi các thay đổi và làm việc nhóm hiệu quả. Sử dụng Git và GitHub giúp bạn lưu trữ code, làm việc với các dự án mã nguồn mở, và hợp tác với các lập trình viên khác trên toàn cầu.

+**Lập trình**: Viết mã trong **VSCode**.

+ **Kiểm tra**: Mở trang web trong **Chrome** (hoặc Firefox) và dùng **DevTools** để debug.

+ **Lưu trữ và làm việc nhóm**: Đẩy mã lên **GitHub** để quản lý và hợp tác với các lập trình viên khác.