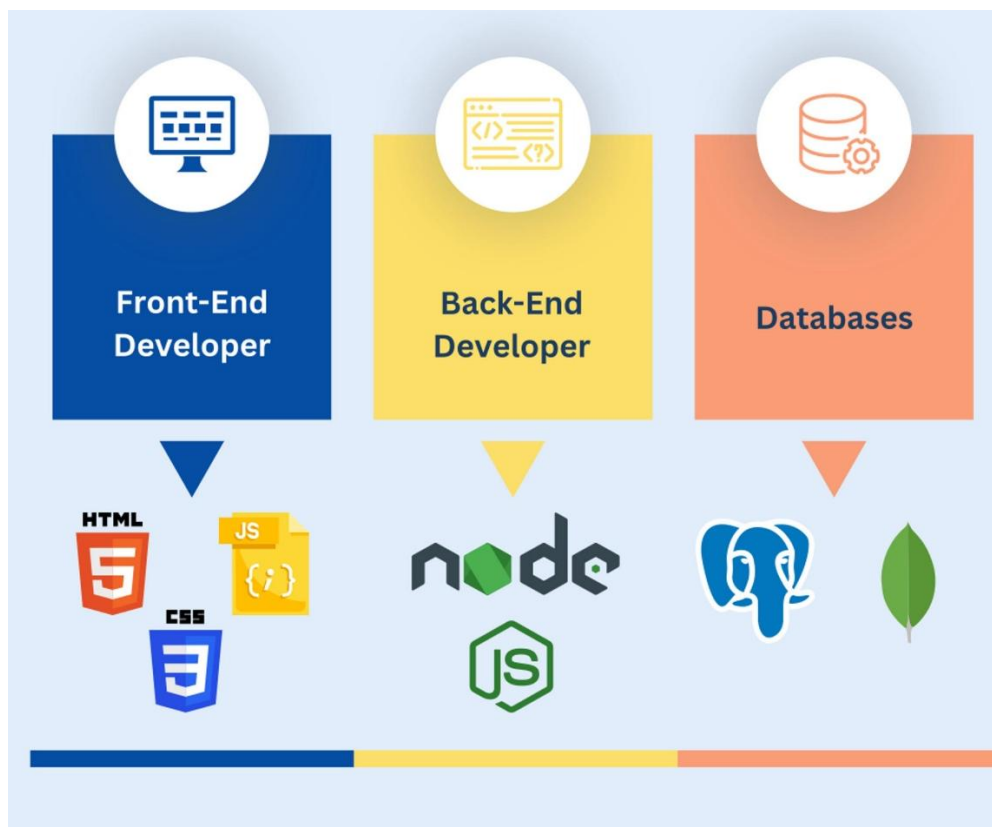


What is Full stack development:

Full stack development refers to the process of developing the client-facing front end and server-based back end of a web application. A full stack developer can work on any part of the development process, from helping to do work with databases and server logic, to the visual design of a web page. Because it allows an individual programmer, or a small group of programmers, to propel the whole product forward, it is a valuable role in modern web development.

The impact of user-experience is one of many considerations in the world of web applications. Difficulty level and assumptions influence a user's cognitive load and user-app interaction expectations. Every interactive interface is driven by a complicated architecture of code that drives business logic, database interfacing, data analytics, and calculations. The role of the full stack developer lies in the space that bridges the operations of the application underneath what the user can see.

The following technologies are frequently used in full stack development: Node.js, Python (Django), PHP, HTML, CSS, JavaScript (for the front end), and MongoDB and MySQL (for the database). The popularity of JavaScript-based stacks such as MERN (MongoDB, Express, React, Node) has made the entire stack more efficient and accessible.



1. Technologies used in Full stack development:

a. Front End Development:

Front-end development is whatever a user sees or interacts with on an application or website via an internet browser. Front-end development consists of three foundational pillars - HTML, CSS, and JavaScript. As front-end technologies such as Vue.js, React or Bootstrap increase and enhance the entire process.

The primary goal of front-end development is to provide a seamless, responsive, and beautiful experience for users. Developers focus on creating layouts, handling user input and clicks, and styling content.

✓ HTML:

The standard language for generating and organizing online material is HTML. It uses tags like `<h>`, `<p>` and `` to arrange text, images, links, and other elements. HTML is the fundamental structure of every web site. It just manages what can be seen on the site; it has no control over how things seem or act. A browser could not determine what to show without HTML.

The elements of HTML can be styled and graphically presented with CSS. It regulates a webpage's design, colors, fonts, spacing, and general look. CSS helps programmers maintain the organization and cleanliness of their code by separating style from contents. Styles can be used from a distant file, within the page, or directly. CSS improves a website's aesthetic appeal and cross-device usability.

✓ CSS:

With CSS, you can style and graphically present an HTML element. CSS controls everything about the design of your web pages: the design itself, colors, fonts, spacing, and overall appearance. CSS allows developers to keep their code organized and cleaner by separating design from content. You can use styles from an external file, from within the page, or inline. CSS relates to your website's visual charm and capability to work across a range of devices.

✓ Javascript:

CSS can be used to complement styling or graphic formatting to the HTML elements. It regulates the presentation of a webpage. The styling features include the design of elements like boxes, colors, fonts, spacing, and the overall look of the design. By utilizing CSS, developers can essentially separate the design aspect away from the content aspect this aids in keeping the codewell formatted. External or separate files or the page or inline styles can be used. CSS not only makes a website look visually more appealing but improves the usability across devices.

b. Backend Development:

Back-end development is when web program component users don't see. The back end is responsible for server actions, database connections, authorization, and logic that is

performed for every request. With the back end, user input is performed and reacts when a user clicks an option or submits a form. The back end is a connector between the front-end, server, and database. Without the back end, dynamic websites could not do anything beyond static pages.

✓ NodeJS:

Node.js is a JavaScript runtime that allows developers to build server-side application code with JavaScript. Node.js has an event driven, non-blocking structure, making it ideal for building fast and scalable applications. Node.js is commonly used for microservices, real-time applications, and APIs. Node's V8 engine (like Google Chrome) ensures great performance. A primary benefit of Node is that it provides developers the ability to use a single language (JavaScript) for the entire stack.

✓ Python

It is an established back-end programming language because of its simplicity and easy readability. Django is an efficient technology that allows developers to create safe, scalable applications quickly. On the other hand, Flask offers more control over the building process and is more lightweight. Both allow for design, user's authorization, and access to databases. Python, because of its active community and ease of use, is great for both beginner and experienced back-end developers.

✓ PHP

PHP is a favored scripting language, purpose-built for web development. PHP is available on almost all web servers and provides the backbone for many popular content management systems (CMS) including WordPress. PHP enables the creation of dynamic pages and makes connecting to databases, such as MySQL, effortless. Frameworks like Laravel bring about modern features like routing, templating, and security. PHP is still dependable and relied on in many web implementations, even considering more recent options.

✓ Java

It is a powerful object-oriented programming language that is utilized in enterprise and large-scale applications. With its integrated tools and auto-configuration, Spring Boot is a Java-based platform that makes back-end development easier. It supports microservices, security, RESTful APIs, and more. Java apps are renowned for their scalability, stability, and security. Business-critical systems, healthcare, and finance all make extensive use of Spring Boot.

c. Database:

A well-structured method of managing, storing, and retrieving data for applications is known as a database. Databases are a core component of full stack development as they persist user, content, and transactional data among other potential data. Databases work in tandem with back-end code to respond to queries and provide

data and results to the front-end. Databases establish the organization, persistence, and availability of data to users. There are two principal types of databases: relational (SQL) and non-relational (NoSQL).

✓ MySQL:

One of the most commonly produced Relational database management system (RDBMS) is MySQL. It uses SQL (Structured Query Language) for interacting with the data that it organizes in structured tables with rows and columns. MySQL is open-source, reliable, and used on often on sites like Facebook and WordPress. It works very well for applications with complex queries and must maintain data consistency. It easily integrates with back-end languages such as, Python, Node.js, PHP, etc.

✓ Oracle:

Oracle is an enterprise level relational database system with exceptional efficiency that is used by large organizations. It has very good tools for managing transactions, securing data, and improving efficiency. Oracle supports Distributed Database systems, PL/SQL (its procedural language) and many advanced features of SQL. It is most suitable for mission-critical systems in government, communications, and banking. Scale and reliability is what Oracle is best known for, despite being complex and commercial.