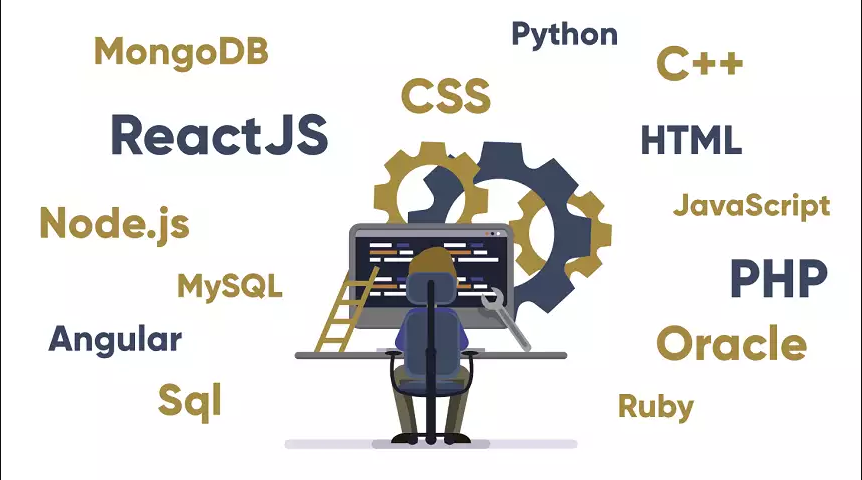
Web Development Stacks



1. state all the web application parts and services

|  |
| --- |
| 1. **Database**    * + **Description:** Stores user data, transactions, and application settings        - **Examples** 2. MySQL 3. MongoDB 4. SQL Server 5. PostgreSQL  * **ORM**  1. Entity Framework (C#) 2. Mongoose (Node.js) 3. Hibernate (Java) 4. **Frontend (Client-Side)**     * + **Description:** The user interface (UI) and experience (UX) of the web application that runs in a browser        - **Examples** 5. HTML 6. CSS 7. JavaScript  * **Frameworks**  1. React 2. Angular 3. Vue.js 4. **Backend (Server-Side)**    * + **Description:** Handles business logic, authentication, and database interactions        - **Server** 5. Node.js 6. Python (Django/Flask) 7. Ruby (Rails) 8. Java (Spring) 9. C# (.NET) 10. **API**     * + **Description:** Enables communication between frontend and backend         - **Examples** 11. REST 12. GraphQL 13. **Hosting**     * + **Description:** A platform that helps deploy and scale web applications efficiently.      * + - * **Examples**  1. Azure 2. AWS 3. **DevOps**  * **Description:** Automates deployment, testing, and scaling   + - * **Examples**  1. Docker 2. **Web Server**  * **Description:** Processes client requests and serves web pages   + - * **Examples**  1. Apache |

1. State all the web stacks that you know and compare between them : where to use, when to use, features

|  |
| --- |
| 1. **.Net**  * **Database, Programming Language, Framework** * **Database:** SQL server. * **Programming Language:** C#. * **Framework:** .Net / .Net Core. * **Where to use** * Enterprise apps * financial systems * E-commerce Platforms * **When to use** * For large-scale applications requiring reliability, scalability, and maintainability. * When security and enterprise-grade solutions are a priority * **Features** * High performance * Common Type System * strong IDE support (Visual Studio). * Security  1. **Java**  * **Database, Programming Language, Framework** * **Database:** Oracle SQL. * **Programming Language:** Java. * **Framework:** Spring. * **Where to use** * Enterprise apps. * E-commerce Platforms. * Social Media Apps. * **When to use** * When building complex systems that require multi-threading and robustness. * When building secure and scalable enterprise-level applications. * **Features** * Object-Oriented. * Platform Independence. * Scalability.  1. **Laravel**  * **Database, Programming Language, Framework** * **Database:** MySQL. * **Programming Language:** PHP. * **Framework:** Laravel. * **Where to use** * Web Applications. * E-commerce Platforms. * Content Management Systems. * **When to use** * When building web applications that need features like routing, middleware, and authentication. * When you want MVC architecture for clean code and separation of concerns. * **Features** * Eloquent ORM. * Routing & Middleware. * Security.  1. **Python**  * **Database, Programming Language, Framework** * **Database:** SQL Lite. * **Programming Language:** Python. * **Framework:** Django. * **Where to use** * Web Applications. * RESTful APIs. * Content Management Systems. * **When to use** * When rapid development is needed with a focus on clean code and simplicity. * When you need built-in admin interfaces for managing content and users. * **Features** * Django ORM. * Built-in Admin Interface. * Security.  1. **MERN**  * **Database, Programming Language, Framework** * **Database:** Mango DB. * **Programming Language:** JavaScript. * **Framework:** Express. * **Where to use** * Real-Time Applications. * E-commerce Platforms. * Social Media Apps. * **When to use** * When building apps that require real-time updates, like messaging or live feeds. * When you want fast development and scalability. * **Features** * Node.js for fast and scalable server-side applications. * MongoDB for a flexible, schema-less database solution. |

1. state all the tools of development that are used in .Net stack

|  |
| --- |
| * **Tools in the .NET Stack** * **Programming Language:** C# * **Frameworks:** .Net, .NET Core * **IDEs:** VS, SSMS * **Database:** SQL Server * **ORM:** Entity Framework Core, Dapper * **CI/CD:** GitHub Actions. * **Frontend integration:** Blazor, Angular, Nextjs, MVC |

**Reference**

<https://www.nobledesktop.com/classes-near-me/blog/best-web-development-stacks>

<https://softteco.com/blog/web-application-archtecture-explained>

<https://medium.com/@chandra.pcs/mastering-the-net-full-stack-a-comprehensive-guide-ade1776e16b7>

<https://fullscale.io/blog/top-5-tech-stacks/>

<https://dev.to/snevy1/10-best-web-development-stacks-409i>

<https://radixweb.com/blog/top-web-development-stacks>