web development stacks

In web application we have three main parts and every part has its own services.here the application parts and services:

* Data tier services as database.
* Logic tier services as backend.
* Presentation tier services as frontend.

A web development stack is a collection of tools and technologies used to create websites or web applications. Here are the most popular web stack used:

* .Net, used C# language, SQLserever for database and .Net framework. We used it for big web applications that need performance.
* The MERN stack, composed of MongoDB, Express.js, React, and Node.js.
* The MEAN stack substitutes React with Angular, an all-in-one framework created by Google programmers.
* The Serverless stack, relying on cloud infrastructure instead of a local server, simplifies infrastructure management and offers high scalability and cost-effectiveness.
* The LAMP stack, composed of Linux, Apache, MySQL, and PHP.
* The PERN stack is similar to MERN but uses PostgreSQL, a reliable, secure SQL-based database system, instead of MongoDB.
* The Ruby on Rails stack allows developers to create an entire web application by combining Ruby on Rails with HTML, CSS, and JavaScript, known for its security and simplicity.

.Net is a full stack development for web application and there are many tools and technologies used for it. Here the most popular tools used for .Net:

* frontend : for the client side rendering there are many frameworks like Blazor, Angular and next.Js . For the server side rendering we use MVC/Blazor. For the IDE we use VScode.
* Backend: the language used for backend is C# and the framework is .Net core.the IDE used are Visual studio + SSMS. We connect between the frontend and backend by using API.
* Database: we use SQLserver. The framework that connects between backend and database is Entity framework (ORM).
* Testing: we have two testing, one for API testing using postman and the other is unit testing using Xunit.
* deployment : we use Docker or Azure.
* Security: we use OWASP
* Logging: serilog
* Gateway: stripe