MERN: MongoDB, ExpressJS, NODEJS, ReactJS.

- Frontend : ReactJS.
- Backend : NodeJS, ExpressJS.
- DataBase : MongoDB.
  SQL (optional) or as a secondary DB.

Introduction:

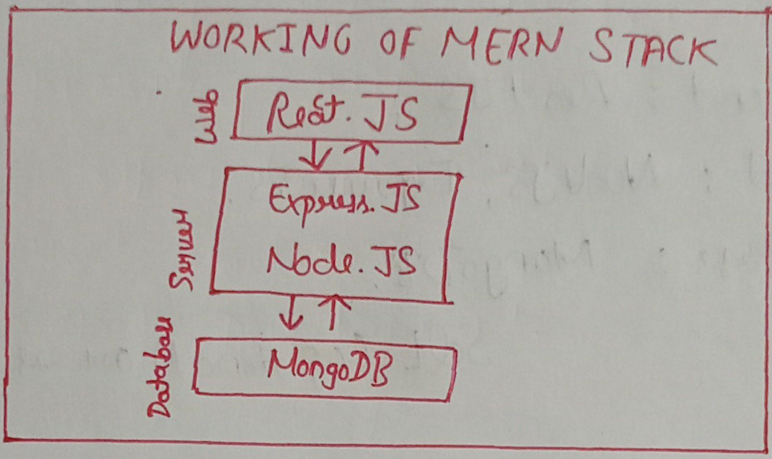
The MERN stack is a widely adopted full-stack development framework that simplifies the creation of modern web applications. Using JavaScript for both frontend and backend enables developers to efficiently build robust, scalable, and dynamic applications.

↳ What is MERN stack?

MERN stack is a JavaScript stack that is used for easier and faster deployment of full-stack applications. MERN stack comparies of 4 technology namely: MongoDB, Express, React and NodeJS. i.e.;

- MongoDB: Non Relational Database.
- ExpressJS: NodeJS web server.
- ReactJS: JavaScript Frontend Framework
- NodeJS: JavaScript Web Server or used for Backend related tasks.

↳ How MERN stack Works?



↳ Roadmap to become a MERN Stack Developer:

- Learn basics of HTML, CSS, and JavaScript.
- Learn React which is a framework or frontend library for building UI.
- Learn NodeJS which is JavaScript Runtime environment.
- Learn ExpressJS, a framework built upon NodeJS to simplify ~~a~~ the process of creating web application and API building.
- Learn MongoDB, a NoSQL database to store or retrieve data from database.

In this course, for learning MERN stack, we will first Revise the basics of JavaScript...

* Introduction to JavaScript...

JavaScript is a Versatile, dynamically typed programming language used for interactive web applications, supporting both client-side and server-side development, and integrating seamlessly with HTML, CSS and a rich standard library.

- JS is a single-threaded language that executes one task at a time.
- It is an Interpreted Language which means it executes the code line by line.
- The data type of the variable is decided at run-time in JS that's why it called dynamically typed.

↳ Key features of JS :

(i) Client-side Scripting : JS runs on the user's browser, so has a faster response time without needing to communicate with the server.

(ii) Event-driven : JS can respond to user actions {clicks, keystocks } in real-time.

(iii) Asynchronous : JS can handle tasks like fetching data from servers without freezing the UI.

↳ Client-side nature of JS :

- Involves controlling the browser and its DOM (Document Object Model).
- Handles user events like clicks and form inputs.
- Common libraries includes AngularJS, ReactJS and VueJS.