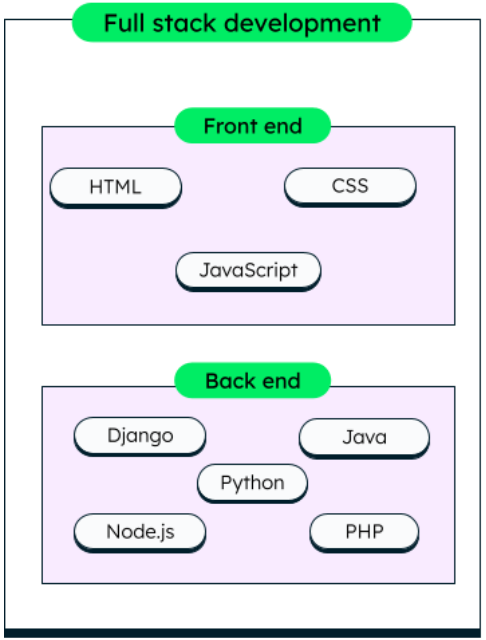
APL-II(Mern Stack)

Full Stack Web Development

- Full stack development refers to the end-to-end application software development, including the front end and back the front end consists of the user interface, and the back end takes care of the business logic and application workflows.

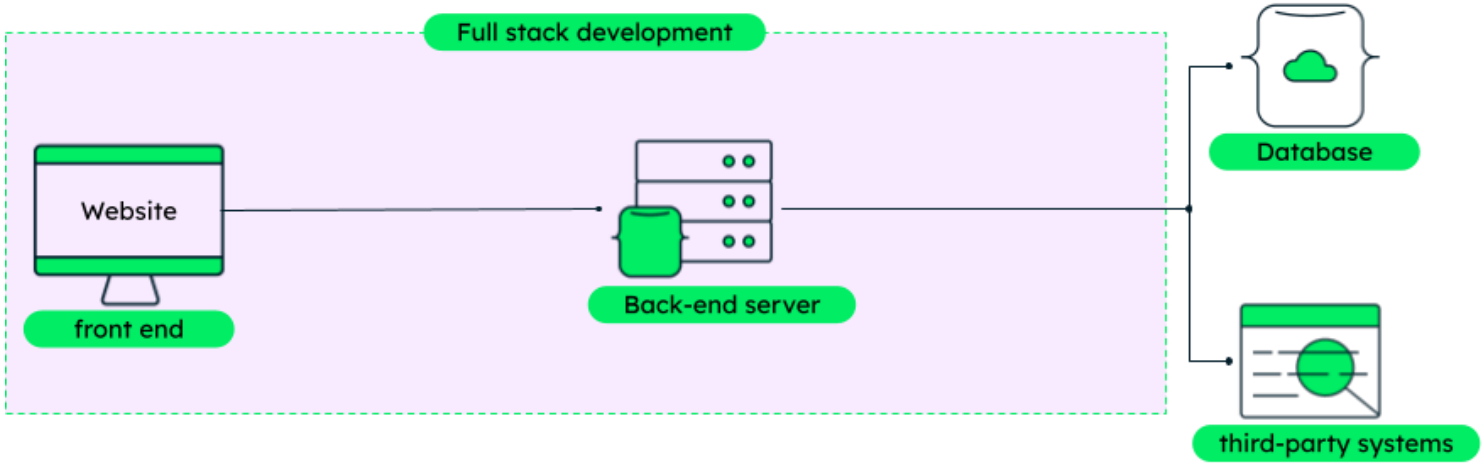
Full Stack Development

- The main components of a full stack development are the front-end, back-end and database
- The website UI can be built using various, front-end technologies like HTML, CSS, Javascript
- The back end is written in programming languages like Java or Python. Further, a good web application would need scalability, event handling, and routing, which are usually handled by libraries and frameworks like SpringBoot or Django.
- The back end also consists of logic that can connect the application to other services and databases. For example, all the user and transaction data is stored in a database through specific drivers handled on the back end.



What is a full stack developer?

- A full stack developer is one who can single-handedly implement both the front-end and back-end workflows, like placing the order or changing the user profile.



What do a full stack developer do?

- Full stack developers must have knowledge of an entire technology stack, i.e., the set of technologies that are used to build an end-to-end application quickly and efficiently.
- Full stack developers should be able to judge whether the selected technologies are the right choice for their project during the early phases. Some responsibilities of a full stack developer are to:
    - Help in choosing the right technologies for the project development and testing both on the front end and the back end.
    - Write clean code across the stack by following the best practices of the tools used.
    - Be up to date with the latest technologies and tools to make the best technology usage decisions.

What languages do full stack developers use?

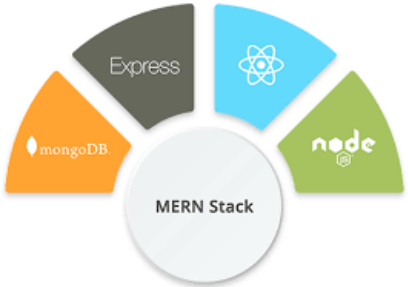
- Full stack developers are free to use any set of languages that are compatible with each other and the overall application framework.
- JavaScript is a popular language often used by full-stack developers as it’s one of the very few languages that can be used both on the front end and back end.
- Front end languages:- HTML, CSS, JavaScript.
- **Front End Libraries and Frameworks:- AngularJS, React.js, Bootstrap, jQuery, SASS,** Semantic-UI, Foundation, Materialize, Backbone.js, Express.js, Ember.js etc
- Back end languages: Python, Java, R, Ruby, Node.js, PHP, C#, REST, GO .
- **Back End Frameworks:** Express, Django, Rails, Laravel, Spring etc.
- **Database:** Database is the collection of inter-related data which helps in efficient retrieval, insertion and deletion of data from database and organizes the data in the form of tables, views, schemas, reports etc. => **Oracle, MongoDB** and **Sql**

Popular Stacks

- **MEAN Stack:** MongoDB, Express, AngularJS and Node.js.
- **MERN Stack:** MongoDB, Express, ReactJS and Node.js
- **MEVN Stack:** MongoDB, Express, **VueJS** and Node.js
- **Django Stack:** Django, python and MySQL as Database.
- **Rails or Ruby on Rails:** Uses Ruby, PHP and MySQL.
- **LAMP Stack:** Linux, Apache, MySQL and PHP.
- There are many advantages of hiring full stack developers for web application development:
  - ❑ Complete ownership and understanding of the project
  - ❑ Saves both project time and cost, and enhances productivity
  - ❑ Faster bug fixing due to knowledge of complete system
  - ❑ Easy knowledge transfer to other team members
  - ❑ Better division of work amongst team members

What is MERN Stack?

- collection of powerful technologies and robust, used to develop scalable master web applications comprising **backend, front-end,** and **database components**.
  - **M** stands for **MongoDB (Database),** mainly used for preparing document database and is a NoSQL (Non-Structured Query Language) Database System
  - **E** stands for **Express,** mainly used for developing Node.js web framework
  - **R** stands for **React,** mainly used for developing a client-side JavaScript framework
  - **N** stands for **js,** mainly used for developing the premier JavaScript web server



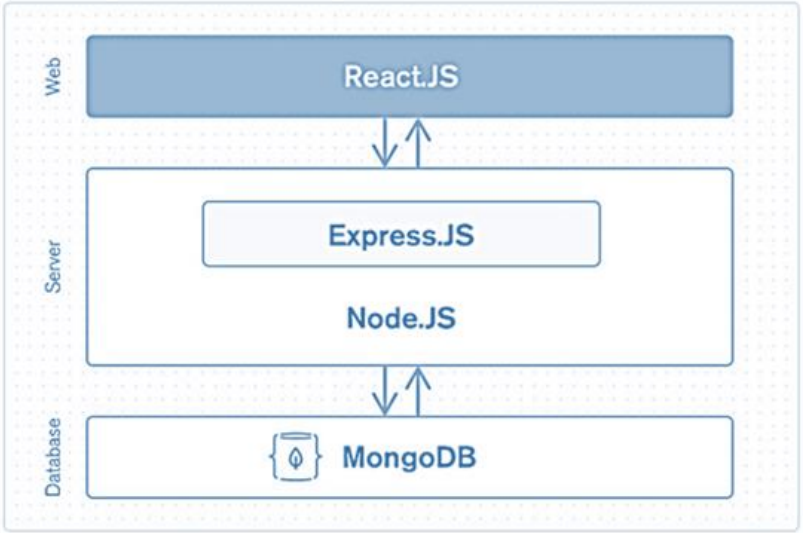
Why should we choose MERN Stack for building Mobile and Web applications?

- it was earlier named as MEAN stack, MERN Stack is one of the variations of MEAN
- in MEAN, ' A ', i.e., Angular.js is replaced by ' R ', i.e., React.js in MERN
- MERN Stack is mainly used for faster development of smaller applications
- MEAN stack is a mainly better option for large-scale applications

1. Cost-effective	2. SEO friendly	3. Better performance
4. Improves Security	5. Provide the fastest delivery	6. Provides faster Modifications
7. Open Source	8. Easy to switch between client and server	

Architectural Structure of MERN Stack and its working?

- MERN has a 3-tier Architecture syste
- These layers are as follows:
  1. Web as front-end tier
  2. Server as the middle tier
  3. Database as backend tier

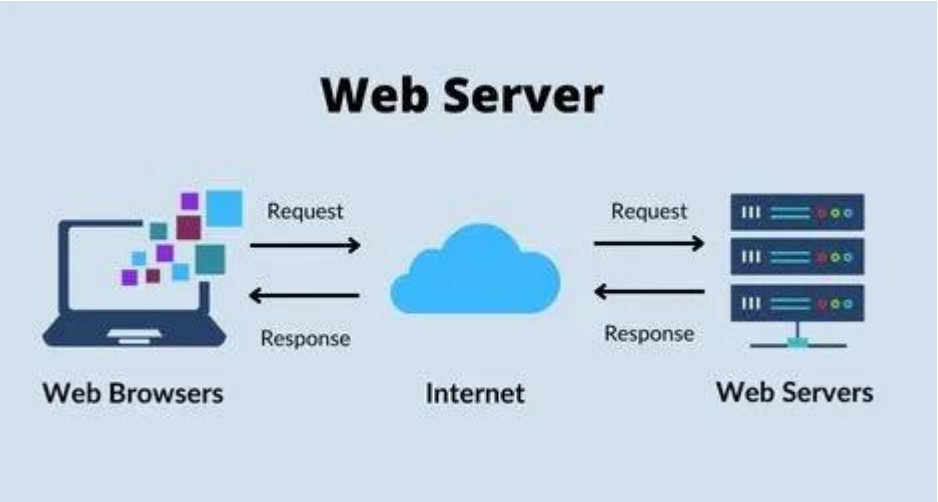


Web or front-end tier

- The front end of a website is everything the user either sees or interacts with when they visit the website.
- It is responsible for the total look and feel of an online experience
- Some examples of front-end applications that enable web languages are HTML, JavaScript, and CSS
- handled by React.js.
- open-source front-end JavaScript libraries used for building Web applications
- famous for creating **dynamic client-side applications**
- construct complex interfaces by using single components
- connects those complex interfaces to data available on the backend server
- React is used to create mobile applications (React Native)
- React allows the reusability of code
- It permits users to create large web applications that can easily change the data of the page even without reloading the page.

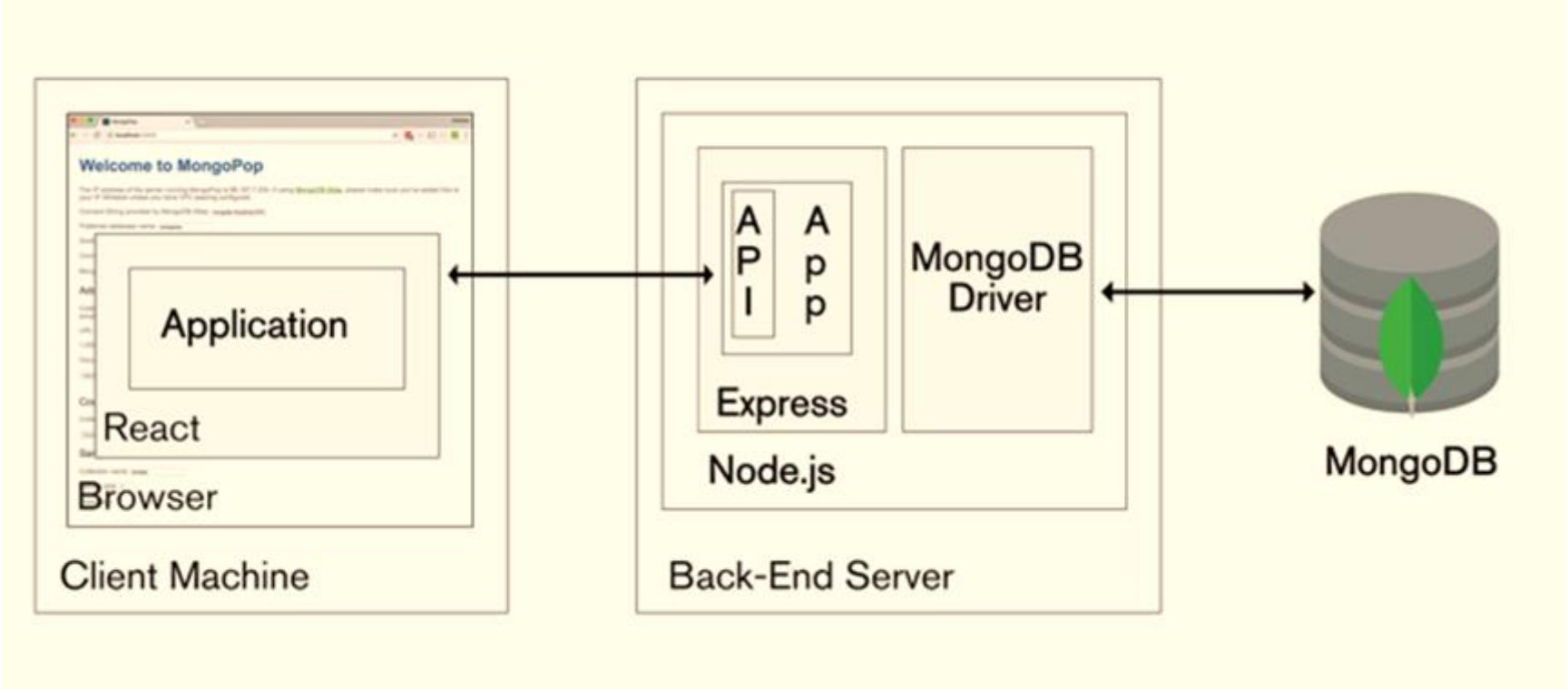
Server or middle-tier

- A server is a computer program or device that provides a service to another computer program and its user, also known as the client.
- On the World Wide Web, for example, a Web server is a computer that uses the HTTP protocol to send Web pages to a client's computer when the client requests them

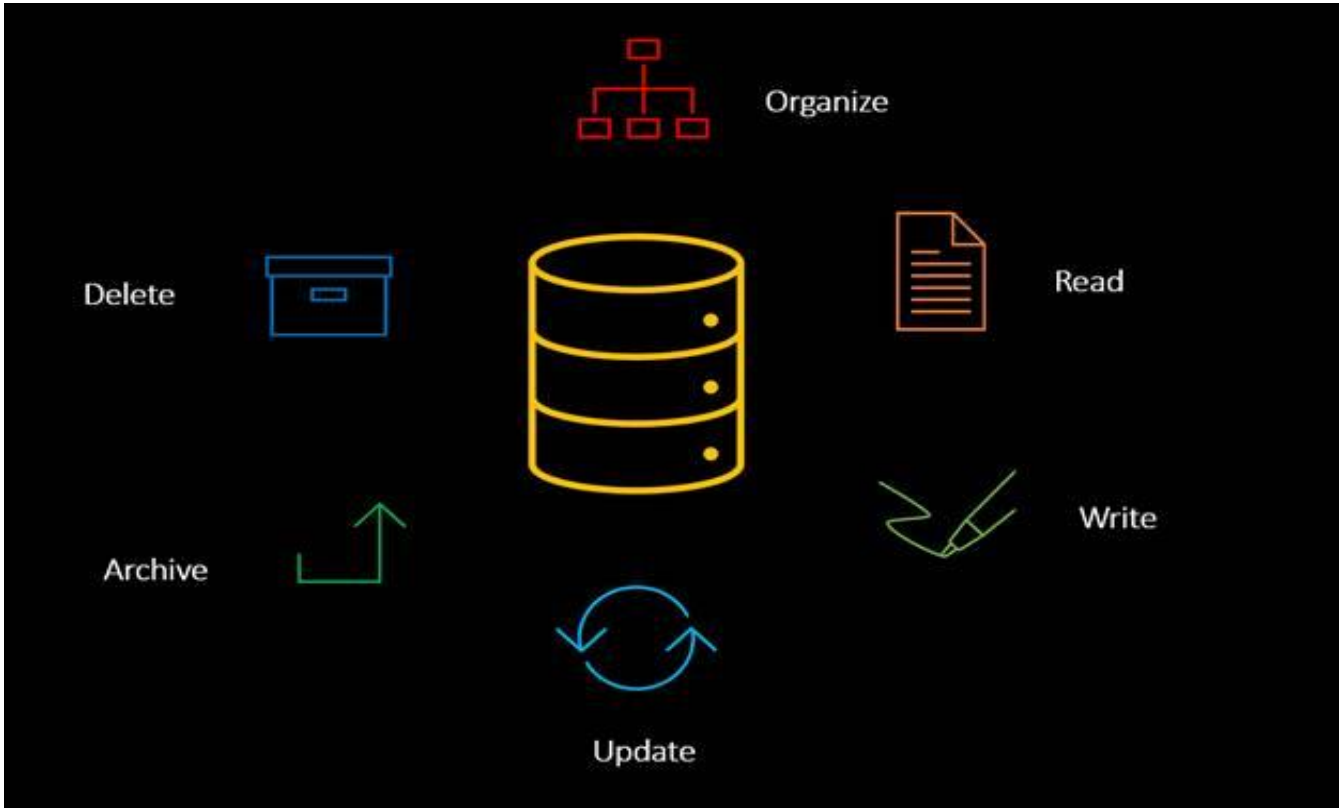


Server or middle-tier

- mainly handled by two components of the MERN stack, i.e., **Express.js** and **Node.js.**
- These two's components handle it simultaneously because Express.js maintained the Server-side framework, running inside the Node.js server
- Express.js is one of the widely used backend development JavaScript Frameworks. It allows developers to spin up robust APIs (Application Programming Interface) and web servers much easier and simpler
- Node.js plays a very important role in itself. It is an open-source server environment, and it is a cross-platform runtime environment for executing JavaScript code outside a browser
- Node and Express make up the middle application or tier. Node.js very powerful and popular JavaScript server platform, and Express.js is a server-side web framework

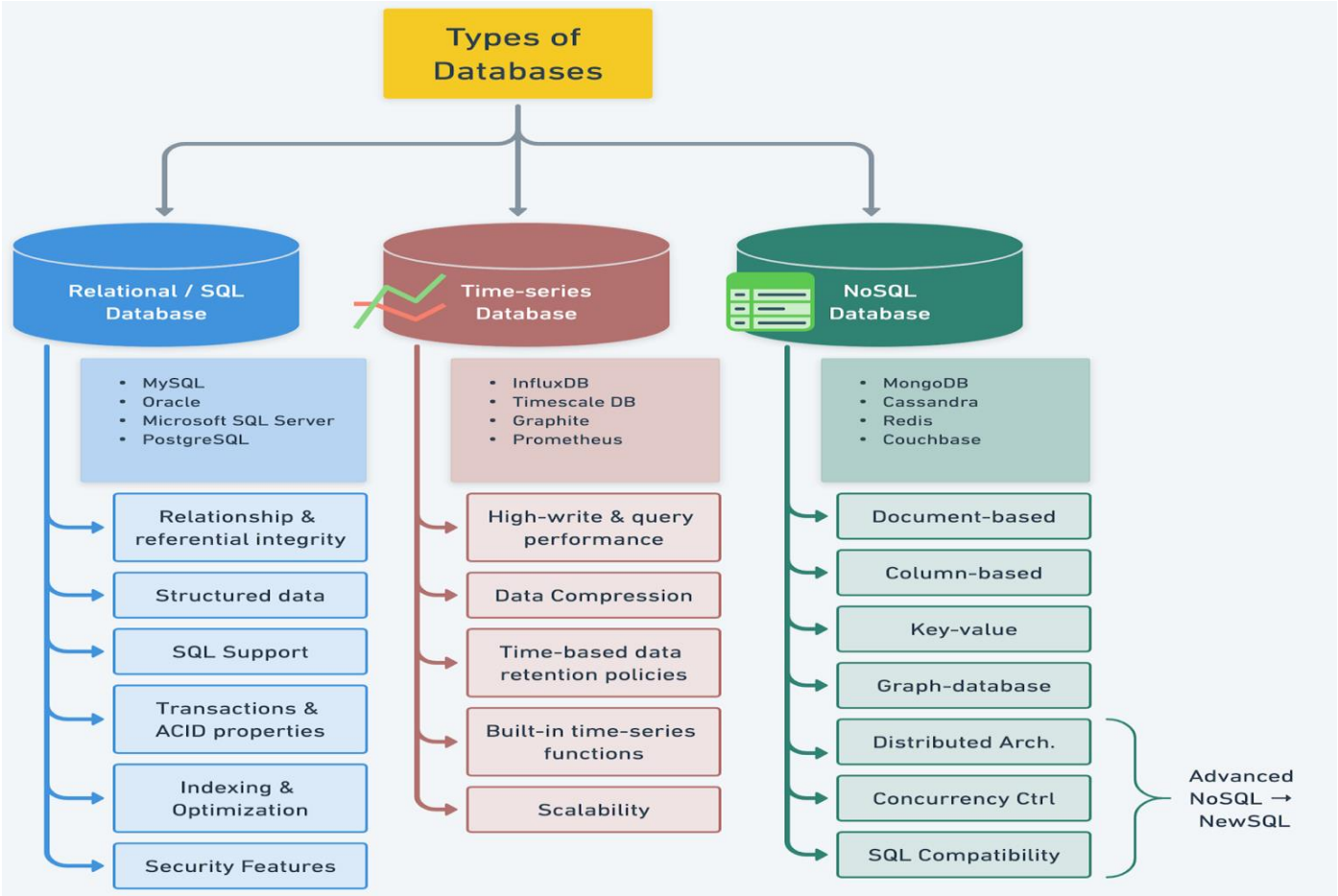

Welcome to MongoPop
Application
React
Browser
Client Machine
API
App
Express
MongoDB Driver
Node.js
Back-End Server
MongoDB

Database



Database as backend tier

- mainly handled by MongoDB
- It mainly stores all the data for **safety purposes**. It maintains a proper record, which usually returns the data to the user whenever required. It mainly stores the data in the database.
- It generates two or more replica files of the data so that whenever the system fails, it can retrieve the exact information or data that the user wanted earlier
- MongoDB is not based on the table-like relational database structure
- it provides an altogether different mechanism for the retrieval and storage of data
- the most popular NoSQL (NoSQL or Non Structured Query Language) database, an open-source document-oriented database.
- The term 'NoSQL' typically means a non-relational database that does not require a fixed schema or proper relational tables to store the necessary data in it
- . MongoDB stores the data in a different format other than the relational tables, consisting of rows and columns.
- MongoDB is a source-available cross-platform document-oriented database program
- MongoDB uses JSON-like documents with optional schemas

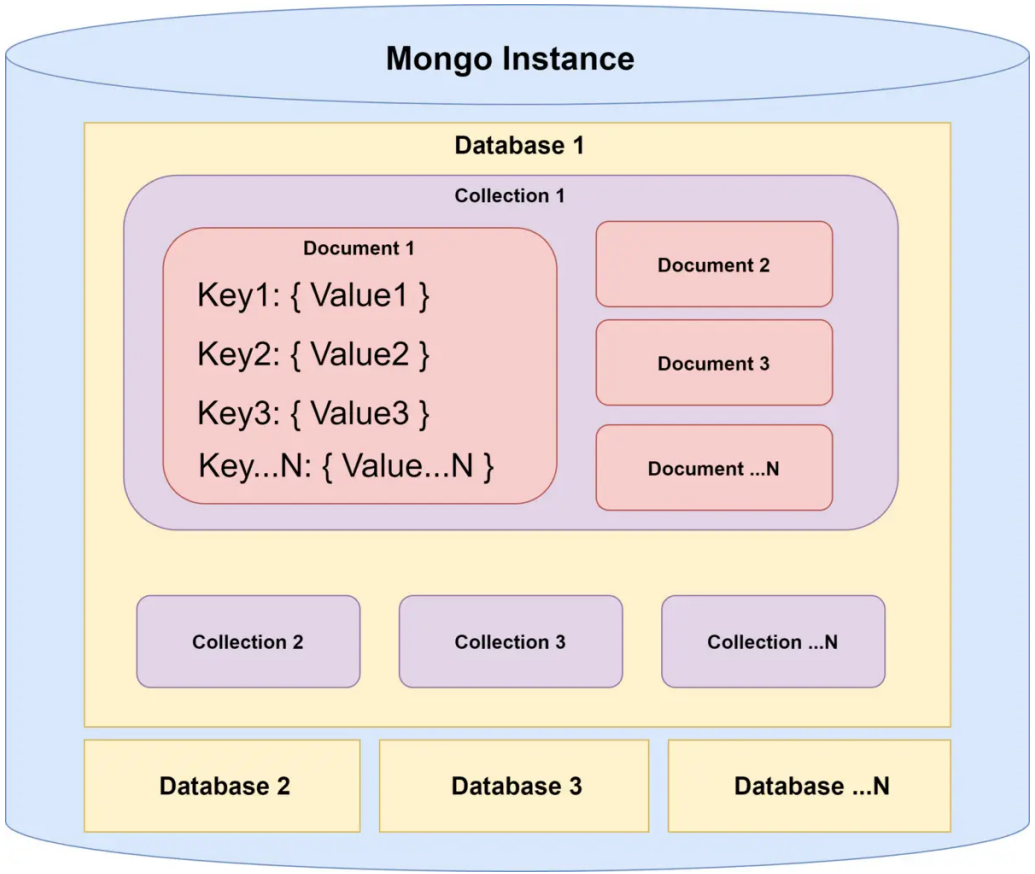

Types of Databases
Relational / SQL Database
Time-series Database
NoSQL Database
MySQL
Oracle
Microsoft SQL Server
PostgreSQL
InfluxDB
Timescale DB
Graphite
Prometheus
MongoDB
Cassandra
Redis
Couchbase
Relationship & referential integrity
Structured data
SQL Support
Transactions & ACID properties
Indexing & Optimization
Security Features
High-write & query performance
Data Compression
Time-based data retention policies
Built-in time-series functions
Scalability
Document-based
Column-based
Key-value
Graph-database
Distributed Arch.
Concurrency Ctrl
SQL Compatibility
Advanced NoSQL → NewSQL

- **Mongo DB** is the most popular NoSQL (NoSQL or Non Structured Query Language) database, an open-source document-oriented database.
- is not based on the table-like relational database structure
- it provides an altogether different mechanism for the retrieval and storage of data.
- storage format in which the data is stored is known as BSON, which stands for Binary JavaScript Object Notation; its binary structure encodes length and type of information, which allows it to be parsed much more quickly.
- It allows a highly scalable and flexible document structure
- It is very faster as compared to RDBMS due to its efficient storage and indexing techniques
- In MongoDB, complex join operations are not available; hence, it cannot support complex transactions
- MongoDB uses JavaScript for coding as a language which is one of the great advantages.
- It is Schemaless as any data stored which is stored in a separate document.

Important features of MongoDB -

- **Schema-less Database:** MongoDB has this one of the great features, a single collection comprises multiple documents, and these documents may further comprise the different numbers of values, fields, and so on.
- **Indexing:** In MongoDB, every data item has provided a particular index, categorized as primary and secondary indices. With this indexing, data retrieval is easier for the user; it saves a lot of time.
- **Document Oriented:** all the data has been stored in documents instead of tables like SQL. Also, these documents have their unique object id. In these documents, the informative data is stored in fields, i.e., **key-value** pairs instead of columns and rows, making the data much more flexible and easier to fetch out rather than applying queries for every data compared to RDBMS.
- **Faster -** Each data item has its index value, making it easier for us to retrieve any data without wasting time writing queries and making logic accordingly.

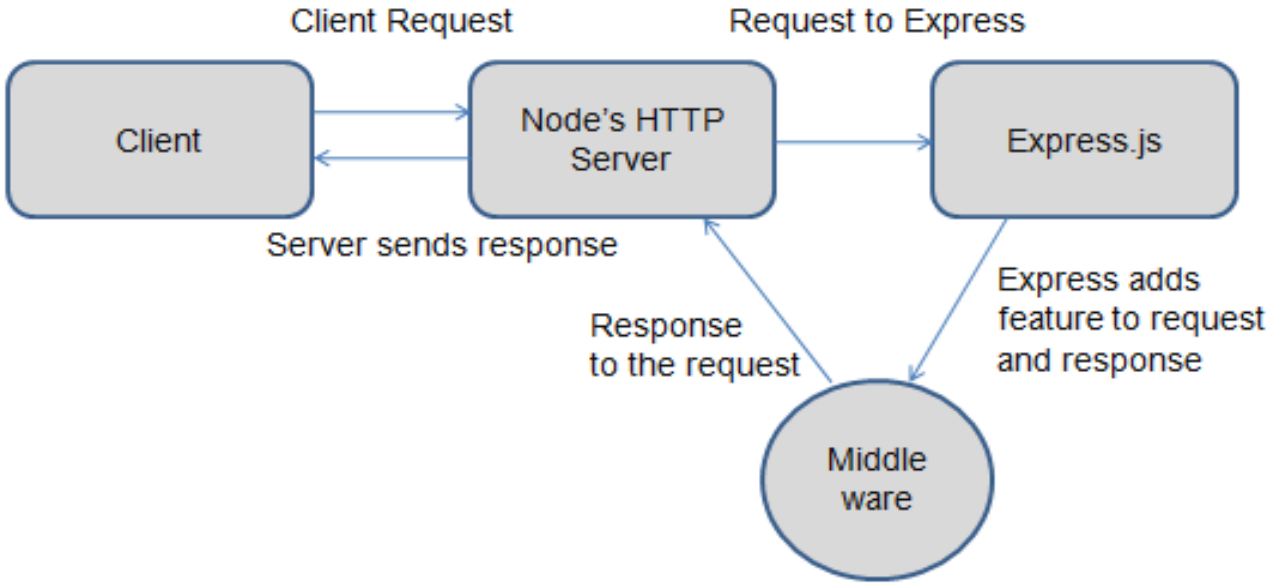
- **Scalability:** MongoDB is more scalable with the help of sharding. It provides horizontal scalability. Here the term sharding means distributing data on multiple servers; in this, a large amount of data has been divided into multiple small data chunks with the help of shard key. These types of **data chunks** are evenly distributed across shards that reside across many physical servers.
- **High Performance:** very high performance and has data persistency due to the presence of its great features like **indexing, scalability, replication**, etc.
- **Replication and Highly Available -** MongoDB increases the availability of data due to creating multiple copies of data on different servers. Providing redundancy or data replication ultimately protects the database from any hardware failure and protects the data from being lost in the future.
- **Aggregation:** GROUPBY clause performs various operations on the grouped data to get the unique or computed
- **Simple Environment Setup -**. One can easily set up MongoDB in their system without applying much effort.




```
db.users.insertOne(          ← collection
  {
    name: "sue",             ← field: value
    age: 26,                 ← field: value
    status: "pending"        ← field: value
  }                                          } document
)
```

Express JS

- Express is a JavaScript server-side(backend) framework that runs within **js**
- A framework is a structure that you can build software on. It serves as a foundation, so you're not starting entirely from scratch. Frameworks are typically associated with a specific programming language and are suited to different types of tasks
- provides the developer with a platform to create and maintain robust servers.
- used for building and designing web and mobile applications easily and quickly by providing server-side logic
- allows developers to spin up robust APIs (**Application Programming Interface**) and web servers much easier and simpler.
- makes robust web servers easier to organize your application's functionality with routing and middleware
- also adds helpful functionalities to Node.js HTTP (**HyperText Transfer Protocol**) objects


Client Request
Request to Express
Client
Node's HTTP Server
Express.js
Server sends response
Response to the request
Express adds feature to request and response
Middle ware

Some important features of Express

- makes Node.js web and mobile application development much easier and faster.
- has a very simple environment setup.
- Express is very easy to connect with Databases like MongoDB.
- Based on HTTP methods and URLs, Express allows you to define the routes of your application.
- Routing mainly aims to describe code that needs to be run in response to any request received by a server. Routing is generally done based on the sequence of URL patterns and the HTTP method, which is associated with the request.
- to perform additional tasks and functions on any request and response, you can easily use various middleware modules present in Express.
- If any error occurs and you want to handle it, you can easily handle it by using error handling middleware.

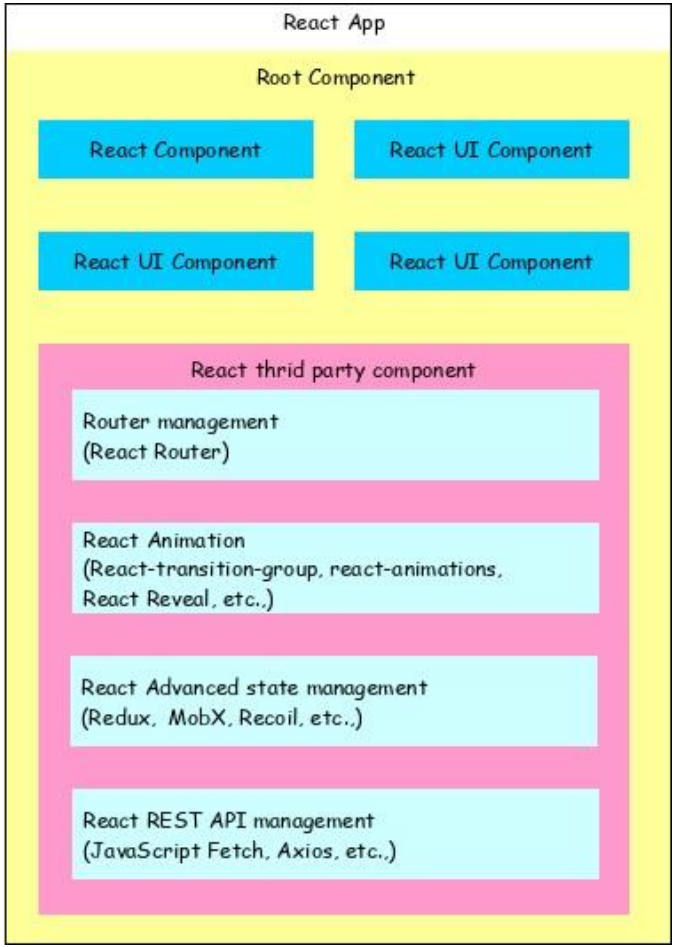
Middleware

- Middleware is software that lies between an operating system and the applications running on it. Essentially functioning as hidden translation layer, middleware enables communication and data management for distributed applications
- Middleware is software and cloud services that provide common services and capabilities to applications and help developers and operators build and deploy applications more efficiently. Middleware acts like the connective tissue between applications, data, and users.

- the most popular open-source front-end JavaScript libraries used for building Web applications
- it has some prerequisites that one should follow, that you must download Node packages in your system with their latest versions. Also, you must have an understanding of HTML, CSS and JavaScript.
- used to build user interfaces, especially for a single page web application.
- It is not a JavaScript framework. It is just a JavaScript library developed by Facebook
- React is also used for making a grip over the view layer for mobile and web applications.
- It allows us to create reusable UI (User Interface) components.
- It was first created by software engineer Jordan Walke, who works for Facebook.
- React was first deployed in the Facebook news feed.
- It allows developers to create large web applications that can easily change the data of the page even without reloading the page.
- The main objective of reacting is that it only works on user interfaces in the application, whether mobile or web.
- It is very fast, simple and scalable.
- React is also used with a combination of other JavaScript libraries or frameworks.

The salient features of *React library* are as follows −

- Solid base architecture
- Extensible architecture
- Component based library
- JSX based design architecture
- Declarative UI library



Some important features of React -

- **Easy to learn -** it is very easier for a beginner to learn it and make web and mobile applications using this front-end framework. Angular is referred to as a ' Domain Specific Language ', so it is implied that it is quite difficult to understand it. For Learning React, you need the basic knowledge of CSS and HTML.
- **Simple -** React is one of the simplest open-source JavaScript front-end frameworks for building web and mobile applications. It uses the component-based approach, uses plain and simple JavaScript, and a well-defined lifecycle, which makes react much simpler and easier. It uses a simple syntax named JSX, which allows learners or developers to mix HTML with JavaScript to make it easier for them to apply and use it for making efficient web and mobile applications.
- **Data Binding -** React uses an application architecture known as Flux to control data flow to components via one control point called the dispatcher. It uses **one-way data binding**, which is easier to debug self-contained components of large React applications.

Some important features of React -

- **Native Approach -** React is used to create mobile applications (React Native) and web applications. React allows the reusability of code and can easily support it, which has many benefits and is much time saver. So simultaneously, at the same time, we can make **IOS, Web applications** and Android.
- **Performance -** React has very fast performance due to the immutability of data. As the name suggests, we can predict that the immutable data structures never change and allows you to compare direct object references instead of doing deep-tree comparisons. The above reason ultimately affects the performance of reacting and makes it faster.
- **Testability -** There are some state functions in the react, where various react views are treated as these functions of the states, and we can easily manipulate with the state we pass to the react view. Also, we can take a look at the output and triggered actions, functions, events, etc.

- an open-source server environment, and it is a cross-platform runtime environment for executing JavaScript code outside a browser
- not a programming language, and even it is not a framework.
- used for building and developing numerous backend services like net applications, mobile applications
- It may be a free ASCII text file platform and may be utilized by anybody.
- It will run on numerous operative systems like **Windows, Mac, Linux, Unix**, etc.
- even be used for agile development and prototyping.
- provides extremely ascendable and really quick services to the users.
- It contains a Non-blocking or, can say, Asynchronous nature.

Some important features of Node.js-

- **Easy Scalability:** js is highly scalable because it uses a single-threaded model with event looping. Node.js uses a single-threaded program, and this program will be able to provide service to many requests.
- **Fast:** The event loop in Node.js handles all asynchronous operations, so Node.js acts like a fast suite. It runs on the V8 engine developed by Google.
- **Easy to learn and debug code:** js is quite easy to learn and debug because it uses JavaScript for running code of web-based projects and various web and mobile applications.
- **Real-time web apps:** js plays a key role in making real-time web applications, if someone wants to build gaming apps and chat applications, then Node.js is a much better option because of its faster synchronization.
- **Caching Advantage:** js provides the caching property in which a single module is cached. Sometimes you do not need to re-execute the same lines of code because it has already been cached using Node.js.

Some important features of Node.js-

- **Data Streaming:**In Node.js, hypertext transfer protocol (HTTP) requests and responses area unit thought-about as 2 separate events. They're knowledge streams, thus once you method a file at the time of loading, it'll scale back the time and create it quicker once the info is given within the style of transmissions. It additionally permits you to stream audio and video files at lightning speed.
- **Object-Oriented Approach:** A huge complaint against Node.js was its JavaScript heritage, which frequently involved many procedural spaghetti codes. With the release and general adoption of ES6, Classes are built into the framework, and the code looks syntactically similar to C#, Java and SWIFT.
- **Event-Driven** and Asyncronization- All Apis of the Node.js library area unit asynchronous, that is, non-blocking. **Corporate Support:** Node.js uses JavaScript, so most companies are combining front-end and backend Teams into a single unit.