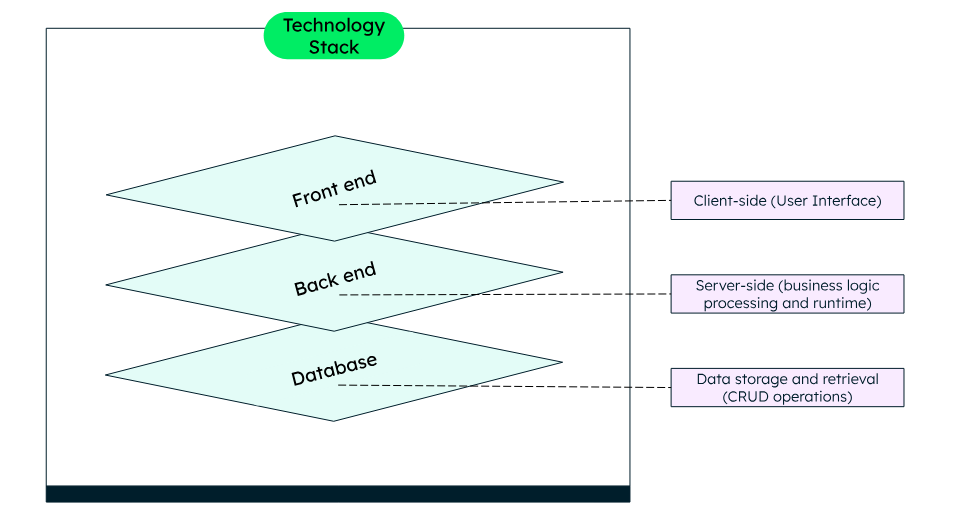
**MERN (MongoDB, Express, Angular, Node), where the traditional Angular.js front-end framework is replaced with React.js**

**Technology stack**

Before diving into the MERN stack, let's quickly understand what a technology stack is. A technology stack is a set of technologies you choose and use to build a web application, mobile application or similar applications. A good technology stack must give a seamless user experience as well as be scalable and cost-effective. A typical tech stack contains a frontend, backend and database and is known as a full technology stack.



Usually, the basic front-end or user interface technologies remain the same, i.e., HTML, CSS and JavaScript. Depending on the project requirements, you can use libraries and frameworks, like React or Angular, that are built upon these UI technologies.

Back-end consists of a server, where your application logic resides. You can write the application logic in one or more programming languages like JavaScript, Java, Python, or use frameworks like Django, Spring, Express.js. To execute the programs, your application needs a runtime like Node.js, JRE (Java Runtime Environment).

The database is the storage hub, where all the application related data is stored. You can choose to store your data in a tabular structure (using relational database systems), or using non-relational, also called NoSQL, such as document structure, graph structure and so on and select the database accordingly. Some examples of databases are MongoDB, Oracle, MySQL.

What is the MERN stack?

A technology stack can be custom (developers can choose the technologies depending on their project requirements) or pre-built (where the technologies have been pre-decided).

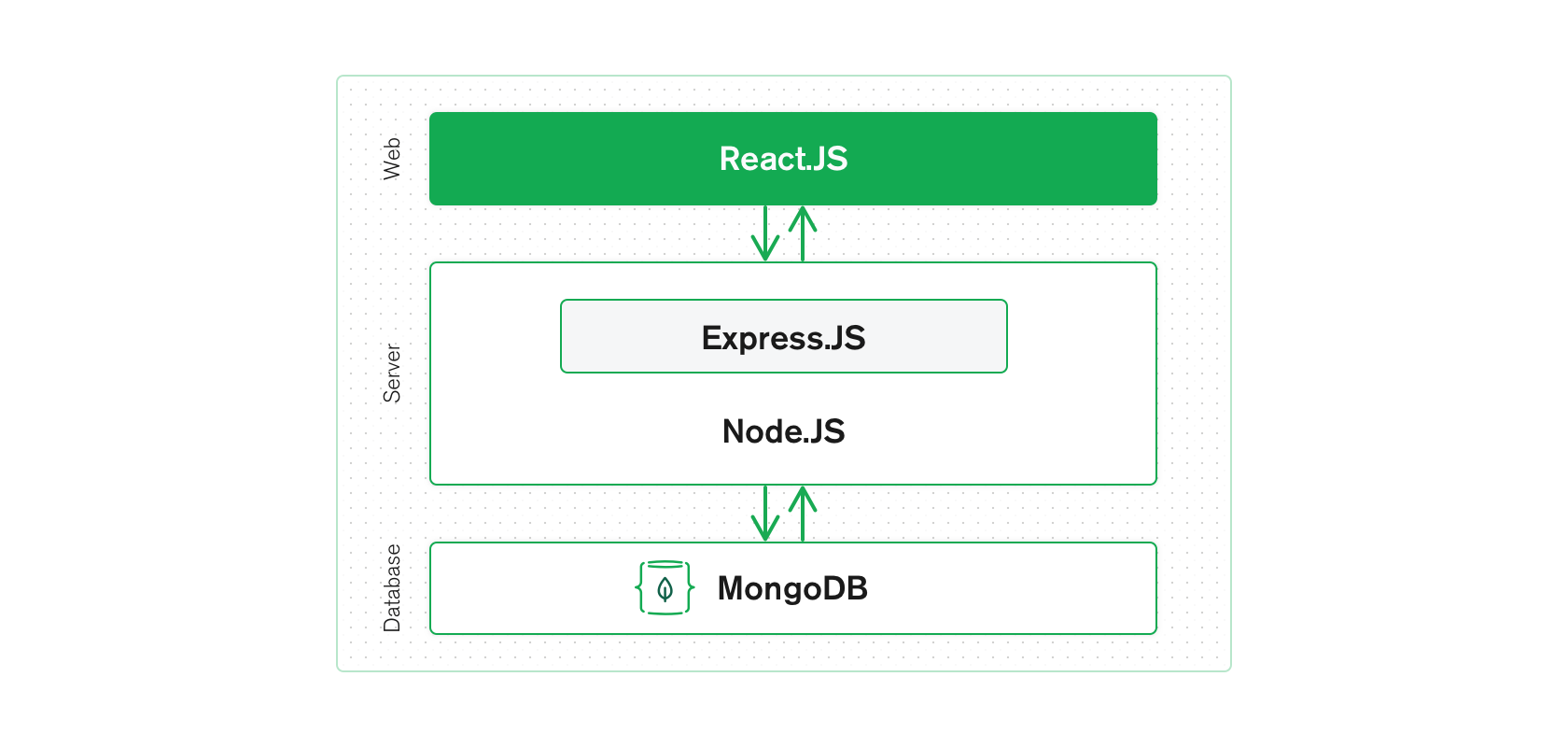
MERN is a pre-built technology stack based on JavaScript technologies. MERN stands for **M**ongoDB, **E**xpress, **R**eact, and **N**ode, after the four key technologies that make up the stack.

* MongoDB — document database
* Express(.js) — Node.js web framework
* React(.js) — a client-side JavaScript framework
* Node(.js) — the premier JavaScript web server (runtime)

Express and Node make up the middle (application) tier. Express.js is a server-side web framework, and Node.js is the popular and powerful JavaScript server platform. Regardless of which variant you choose, ME(RVA)N is the ideal approach to working with JavaScript and JSON, all the way through.

How does the MERN stack work?

The MERN architecture allows you to easily construct a three-tier architecture (front end, back end, database) entirely using JavaScript and JSON.



**React.js front end**

The top tier of the MERN stack is React.js, the declarative JavaScript framework for creating dynamic client-side applications in HTML. React lets you build complex interfaces through simple components, connect them to data on your back-end server, and render them as HTML.

React's strong suit is handling stateful, data-driven interfaces with minimal code and hassle, and it has all the features you'd expect from a modern web framework: great support for forms, error handling, events, lists, and more.

**Express.js and Node.js server tier**

The next level down is the Express.js server-side framework, running inside a Node.js server. Express.js bills itself as a “fast, unopinionated, minimalist web framework for Node.js,” and that is indeed exactly what it is. Express.js has powerful models for URL routing (matching an incoming URL with a server function), and handling HTTP requests and responses.

By making XML HTTP Requests (XHRs), GETs, or POSTs from your React.js front end, you can connect to Express.js functions that power your application. Those functions, in turn, use MongoDB's Node.js drivers, to access and update data in your MongoDB database.

**MongoDB database tier**

If your application stores any data (user profiles, content, comments, uploads, events, etc.), then you're going to want a database that's just as easy to work with as React, Express, and Node.js.

That's where MongoDB comes in: JSON documents created in your React.js front end can be sent to the Express.js server, where they can be processed and (assuming they're valid) stored directly in MongoDB for later retrieval.

**MERN Stack**

* Pre-defined stack of technologies that are known to work well together
* Focus is on JavaScript and JavaScript based on technologies and frameworks
* Streamlined and unified development approach
* Promotes code reusability due to single language components that can be used across

**Full Stack**

* Designers choose different technologies based on the requirements of a project
* Requires a broader skill set that to be learnt depending on the project
* More flexibility in choosing tools and frameworks for development
* Promotes flexibility and versatility as developers can handle different aspects of the application