**Why I chose MERN (**[**MongoDB**](https://www.mongodb.com/)**,**[**Express.js**](https://expressjs.com/)**,**[**React**](https://reactjs.org/)**, and**[**Node.js**](https://nodejs.org/en/)**) stack for**[**Twos**](https://www.twosapp.com/) **-** Parker Klein Aug 7, 2022

This story was inspired by a friend who just started learning web and mobile app development so I thought it would be useful to share.

Graphical user interface, text, application, email, Teams

Description automatically generated

An email from my friend asking about what tech stack I used to build [Twos](https://www.twosapp.com/) and why

**Background**

I first started building [Twos](https://www.twosapp.com/): an app and website that helps you remember \*things\* in 2015.

I was studying computer science at [Vanderbilt University](https://www.vanderbilt.edu/), but I had zero experience in web and app development.

We were learning the fundamentals of programming, but I had no clue how to put it all together into an app.

I started watching a lot of online tutorials and researching the best technologies to build a web and mobile app.

Here is the list of technologies I chose to use and why.

Icon

Description automatically generated

MERN stack ([MongoDB](https://www.mongodb.com/), [Express.js](https://expressjs.com/), [React](https://reactjs.org/), [Node.js](https://nodejs.org/en/))

[**MongoDB**](https://www.mongodb.com/)**(database)**

I chose [MongoDB](https://www.mongodb.com/) as our database because I read good things about the flexibility and performance of non-relational databases.

I was learning about relational databases (SQL and MySQL) in college and they were more complex.

I used mLab for hosting, which has now become [MongoDB Atlas](https://www.mongodb.com/atlas/database).

[**Node.js**](https://nodejs.org/en/)**(server)**

I chose [Node.js](https://nodejs.org/en/) because asynchronous servers were a hot topic at the time. They performed well (still do) and I was learning [JavaScript](https://www.javascript.com/) to practice with basic web development.

A [JavaScript](https://www.javascript.com/) library was easier to pick up rather than learning Ruby, Java, [Python](https://www.python.org/), [Closure](https://clojure.org/), or Elixir in parallel.

[**Express.js**](https://expressjs.com/)**(routing)**

I chose [Express.js](https://expressjs.com/) because it was also a [JavaScript](https://www.javascript.com/) library and pairs well with [Node.js](https://nodejs.org/en/).

[**React**](https://reactjs.org/)**(front end) (originally AngularJS)**

I originally chose [AngularJS](https://angular.io/) because it is a [JavaScript](https://www.javascript.com/) framework, was hot at the time ([Angular](https://angular.io/) 1, I think they are on [Angular](https://angular.io/) 14 now, 😂), and it is backed by [Google](http://www.google.com/).

I switched to [React](https://reactjs.org/) mainly for [React Native](https://reactnative.dev/).

The first version of the [Twos](https://www.twosapp.com/) mobile app was a wrapped web application using [Apache Cordova](https://cordova.apache.org/). It was not good.

I wanted more of a native feel and a lot of hype was starting around [React](https://reactjs.org/) and [React Native](https://reactnative.dev/) so I thought it would be a good transition (it was wonderful).

**Other (domain and server hosting)**

For domain hosting, I use [Google Domains](https://domains.google/). They make it super simple to buy and configure domains and I like [Google](http://www.google.com/).

For server hosting I use [Heroku](https://dashboard.heroku.com/). [Heroku](https://dashboard.heroku.com/) is super easy to set up and is free for a long time. I still have websites that use their free hosting. For their basic paid tier, it is $7/mo so once you’re ready to launch, it can still be cost-effective and performant!

**High-level**

Since I was just starting out, I wanted to keep things as simple as possible. I was learning [JavaScript](https://www.javascript.com/) so using all [JavaScript](https://www.javascript.com/) libraries and frameworks optimized for practice and speed.

I wouldn’t change a thing.