



## Project Description

This project is about creating a web application from front to back that incorporates a complete web server, a responsive user interface, RESTful APIs, and create, read, update, delete (CRUD) operations against a cloud-hosted database.

## Technologies Utilized

- Visual Studio Code (integrated development environment)
- Sketch (visual design and prototyping)
- GitHub (source code repository)
- Heroku (application hosting platform)
- mLab (MongoDB database as a service)
- Terminal (Mac OS command line)
- Homebrew (Mac OS package manager)
- Node.js (JavaScript runtime)
- Node Package Manager (npm) (external libraries, frameworks)
- Express (web application framework)
- React (JavaScript library)
- MongoDB (JSON-like document database)
- Mongoose (Node.js object data modeling)
- HTML5 (markup)
- Cascading Style Sheets (CSS) (markup styling)
- JavaScript (as a 'full stack' language)
- Postman (API testing)

### What problem does your app focus on?

The problem this app focuses on is a question every single person must answer. **Have you paid your bills?**

Each person has their own unique set of bills to pay. Every day, each of must determine where we are with respect to paying these bills. In my case, I have numerous recurring bills where the amount generally does not vary over time and where the due date is on a fixed, recurring schedule. For me, I just need a way to consolidate them so that I can more easily keep track of each bill's payment status at any given time.

### **How does your app solve this problem?**

This app strives to solve this problem by providing users a way to easily create and manage as many bills entries as desired in a single, consolidated, easy to access location. Each bill stores the essential information required for paying the bill (online) and for checking the status, at a glance.

The web application can be accessed from any internet-connected personal computer or mobile device, making it accessible from virtually anywhere there cellular signal or Wi-Fi hotspot present.

## What steps did you take to build the app?

- Paper sketches
- Wireframe documents using Sketch application
- Static web page mock-ups (HTML, CSS)
- Numerous online tutorials (web blogs and YouTube videos)
- Tutorial example starter code
- Ongoing code publishing to GitHub
- Final deployment to Heroku

The wireframes show four screens of a mobile application:

- No Bills:** A screen titled "No Bills" for "May 8, 2019". It has two buttons: "+ New Bill" and "+ New PayDay". Below is a table with columns "Due", "Bill Name", and "Status".
- Add Bill:** A screen titled "Add Bill" for "May 8, 2019". It has input fields for "Bill Name", "Bill Payment URL", and "Bill Due Date" (with a date picker). There is a "Status" section with a "PAID" checkbox and "Save" and "Cancel" buttons.
- All Bills:** A screen titled "All Bills" for "May 8, 2019". It displays a list of bills with columns "Bill Name", "Due", and "Status". Each item has a "View" link. The list includes items like "Home Mortgage", "Reck HSA Dues", "DirecTV Now", "City of Austin Utilities", "PayDay Deposit", "American Express Card", "Verizon iPhone", "Texas Gas Service", "USAA Car Insurance", "24 Hour Fitness", "BBFCU Auto Loan", "PayDay Deposit", and "Spectrum Internet".
- View Bill:** A screen titled "View Bill" for "May 8, 2019". It shows details for a selected bill, including "Bill Name", "Bill Payment URL", "Bill Due Date", and "Status". It has "Edit Bill" and "Delete Bill" buttons.

The mockups show five desktop browser windows for the application:

- No Bills (index.html):** A simple page with a "Create Bill" button.
- All Bills (index-read-all.html):** A page with a "Create Bill" button and a table showing a list of bills.
- Add Bill (index-create-one.html):** A form for adding a new bill with fields for "Bill Name", "Payment URL", "Due Date", and "Status".
- Edit Bill (index-edit-one.html):** A form for editing an existing bill with fields for "Bill Name", "Payment URL", "Due Date", and "Status".
- View Bill (index-update-delete-one.html):** A page showing bill details and "Edit" and "Delete" buttons.

## Live Demo

### Deployment Address

<https://bills-mern-app.herokuapp.com/>

### GitHub Address

<https://github.com/michaeldcastillo/bills-mern-app>

### Obstacles I encountered

- I experienced many moments of severe anxiety. Seriously. It was not fun.
- I could not write code without lots of starter example(s).
- The organization of files for both front and back end code was challenging and took a lot of time.
- I could not determine the correct database schema or determine whether a SQL database would have been a better choice.
- React was and continues to be difficult for me (JSX, component architecture, classes vs functions, state & props, routing, basically everything!).
- I need more practice with JavaScript ES6+ programming constructs (spread operator, destructuring, object and array methods, arrow functions).
- Responsive design is not easy without Bootstrap training wheels (I need to spend time with CSS Grid and CSS Flexbox, etc.).
- There is currently no form validation (not good for any production application).
- I completed a full tutorial on user authentication with Passport.js and JSON Web Tokens, but unfortunately, I was unable to successfully login with my newly-created users.
- I reviewed several blogs and tutorials covering authentication with Auth0, but ultimately could not figure out how to implement that either.
- Deployment went fairly smooth, but required a great deal of time and research.
- I'm sure I'm leaving out about 1,000 other things that caused me grief and need improvement...

### Questions?