OTODO – finalization phase

Concept

When the concept of OTODO was born – the web-based TODO app with possible expansions to serve as a calendar or task synchronizer – the work on concept phase begun. I had to choose what technology would be a good fit to serve the purpose of this app. I wanted to keep in mind that this may become larger than just a TODO application, so I had to choose technologies in a way that would be easier to expand in the future.

I first pondered with the idea of using a service such as Userbase for user authentication and data. I ultimately dropped the idea, because I didn't want to rely too much on third party services, which may be risky in the long run, also seeing building my own backend as a better learning experience.

I decided on Javascript as the backend language, since I wanted to learn building NodeJS applications. It was an obvious choice, seeing how it's the de facto standard for building backend in Javascript.

I knew I had to use an SQL database, so I did research on different SQL databases and decided on SQLite, since it's relatively easy to use, very popular among begginers and has bindings for all major programming languages.

I wanted to use a frontend framework, also as a learning experience, since I already know how to build frontends with pure HTML + CSS + Javascript combination. I read about the most popular ones (React, Vue, Angular), but they all seemed very heavy and an overkill for such a small, simple application. I also felt they come with a steep learning curve.

After some time I discovered a frontend framework called Svelte: designed to be used in small applications. I read the documentation, followed the tutorial and really liked the choices Svelte developers made.

That's how I came up with the whole technology stack for OTODO – SQLite, NodeJS and Svelte.

Building

I decided to begin with the frontend, to see how I felt about building applications using the Svelte framework. I read the tutorial once more, saw some examples and built a simple "Hello world" application. After I got this hands-on experience I started adding features to the frontend, beginning with a header and page layout.

The first component I added was the login form, which also serves as the signup form. I wrote the HTML code, then added CSS for styling. When I got the design I wanted to achieve, I started to work on the functionality (writing callback code for buttons, validating data before it was sent to the server and so on).

When I had the first design finished and waiting to be connected to the backend, I felt it was time to work on the server side of OTODO. I added the NodeJS backend, running Express. Then I could send POST requests from the frontend and receive responses from the backend.

Then I followed a SQLite tutorial and when I knew how to operate SQLite I added the node-sqlite3 package to my project. I then started working on the functionality I needed from this database at the moment – user authentication and storage. For password security (hashing and salting) I decided to use bcrypt. I created a single table in which I stored the username, hashed password and the task list.

When this part of the project was operational (I could sign up and login users) I switched to work on displaying the task list and lettings users modify it. I settled on the design an started to code. I used svelte *each* loop to achieve having as many instances of a task on the webpage as needed,

With my frontend done I begun work on the backend and it was relatively easy since I already had the database working, storing users.

When I finished that, the work was done and the application was up and running.

Last steps

With most of the hard work done, I cleaned-up some of the code. I added documentation in the form of comments for the sake of future reference and added test cases.

The very last step of my work was to create installation, run and test instructions. Luckily, using npm, this process is very easy and straight-forward.