

In this document I am going to describe the structure of my project. I will explain what belongs to the backend and the frontend and what each of the file is supposed to do.

First is to mention that the project on GitHub also contains the documentation in the folder "Documentation". When opening the project, we can see the following folders and files:

"Server.js" file: This is where I connect the API, my MongoDB database, with my server and add the possibility to start a server for my backend.

"package-lock.json" file: Contains links and versions to all the different libraries that are added.

"package.json" file: Contains information and the scripts to run the server and to build the project for deployment.

"routes/api/entries.js" file: This file is the connection to the API and gets, sends, or deletes the data from the database.

"models/Entry.js" file: This file indicates which information are saved in the database.

These files together are forming the backend of my website. The "client" folder contains the frontend:

"tailwind.config.js" file: A config file for Tailwind to changes some parameters.

"package-lock.json" file: Contains links and versions to all the different libraries that are added.

"package.json" file: Contains information and the scripts to run the frontend server and the proxy to the backend to connect them.

The "src" folder is where all the magic is happening starting at the bottom:

"store.js" file: Implementing the store from Redux and it hold the state tree of my website.

"setupTests.js", "serviceWorker.js", "index.js" and "App.test.js" are automatically generated components of React and did not got changed by me. However, the "index.js" file contains the whole application and sends it via root to the "index.html" file which is located in the "public" folder.

"App.js" file: This file is the core part of frontend and combines every component from the "components" folder and includes the store from Redux that was implemented.

"App.css" & "tailwind/style.css" file: These are the components for the Tailwind framework which I used for the design.

Before I will go into the "components" folder I will explain the "reducers" and "actions" folder which are part of Redux.

The reducers serve as the tool that can change the state of the website. The "index.js" file in the "reducers" folder combines every used reducer. Although I only have one it is still needed.

"entryReducer.js" file: This contains the state of the data that are saved in the database. It changes this state depending on the action that it gets and can either delete, add, or just display the entries that are made. The actions and that the reducers can do are defined in the folder actions:

"types.js": Just defines every action that are available.

"entryActions.js" file: Contains every actions of the reducers and is the connection not the API.

This is the main part of the data management of the data that are saved in the database. The “components” folder includes all the visible components of the website that will be created in the “App.js” file.

“NavBar.js” file: Written in JSX it contains HTML, CSS, and JavaScript at once. The navigation bar only contains two buttons “Previous Week” and “Next Week”. I was not able to include any functions for these, so they are merely decoration for now.

“Header.js” file: Only contains the title of the website and is mere decoration.

The folder “Calendar” contains the main part of my visible website and is the only thing that currently has any function.

“Calender.js” file: Creates the days based on an array that contains the names Monday to Friday. Therefore, five days are created with a date which is manually added so it will not change.

“Days.js” file: Here is defined how the days are designed which means the blocks that can be seen on my website. It also includes the file “AssignBtn.js”.

“AssignBtn.js” file: This file is creating the form which can be used to enter the name and the reason. Most of the action is happening here which means the store that was created is used here to save the incoming data that is coming from the form. Several methods defined in this file that dictate how the form must work and how the assign and delete button are working and how the data will be shown, deleted, or added. However, still is not finished because of the problem that every entry that was made in any of the forms will be shown in every block.

### **How to open the Website locally:**

In the “package.json” file of the backend is a script that allows me to start both the backend and frontend at the same time. To activate this script, open the terminal and type “npm run dev” and wait until the frontend website will be opened in the browser. To stop this server, open the terminal and use the hotkey “ctrl+c”.