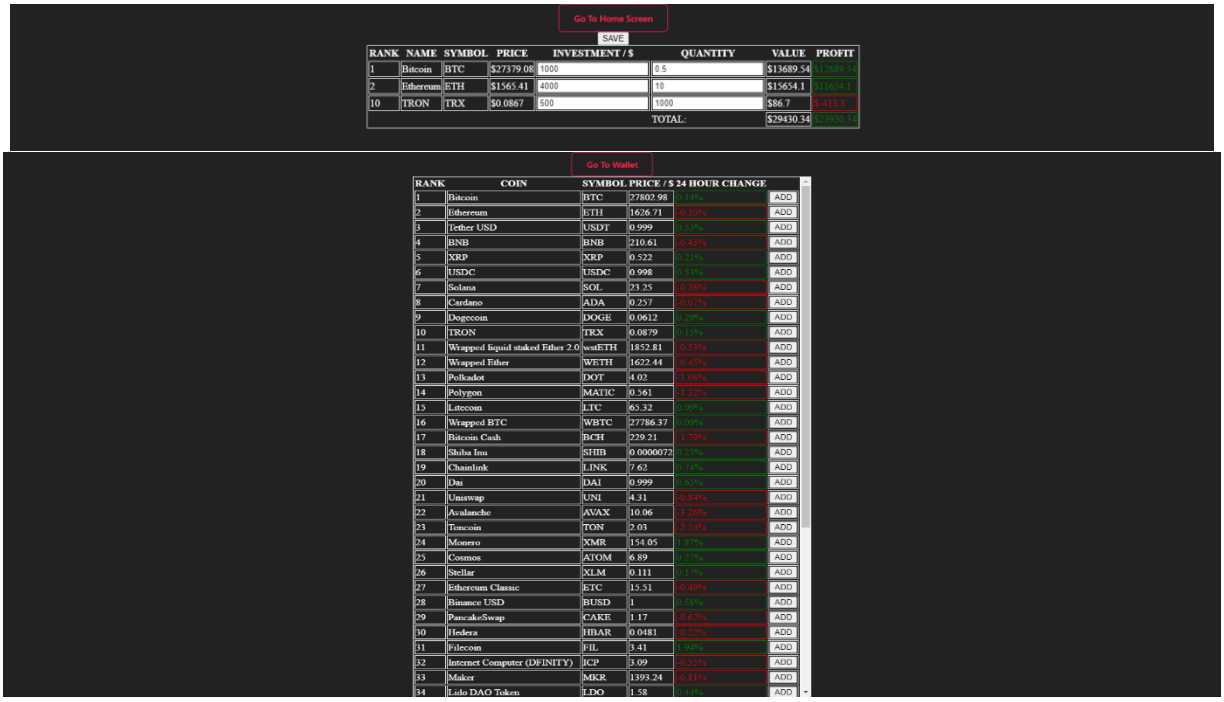
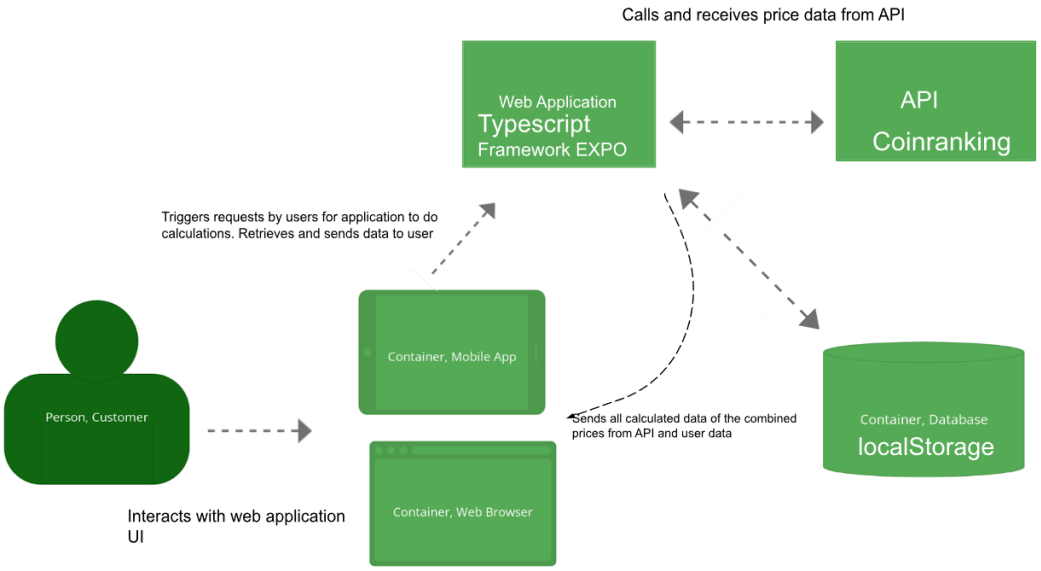
For my web development project I decided to make a cryptocurrency portfolio tracker app. The app consists of a home page on which you have a table of the top 50 crypto currencies by market capitalisation value. Table for each of the coins show its rank, name, symbol, current price and its % change in price in the last 24 hours. On every table row next to each coin is a button reading “ADD”, which when pressed present the user with two prompts, one asking for the investment value put into the coin, and one for the quantity of the coin bought. When the add button is pressed it has the functionality of then adding the respective coin from which the row was pressed into the user’s wallet, along with the values entered in the prompts. By pressing a button at the top of the page reading “Go To Wallet”, the page is redirected to a new page which also has a table of coins, but now only the coins which were added to the wallet earlier by the add button. The columns are also a but different on this table, are as follows: rank, name, symbol, current price, value of money invested into that coin, quantity of the coin owned, and the value of holding in that coin and profit in that coin. At the bottom of the table is also a cell showing the total value of the entire portfolio as well as the total profit in that coin, and all the quantity and investment cells for each coin contain an input element of type number. Additionally, there is another button at the top of the table which is to be pressed on changes in values of the quantity/investment or to fetch a newer value of the coin`s price. The wallet page then also has a button to go back to the home screen in case the user wants to see all the coins again or add another to their wallet.



The making of the project started out with the planning of the idea making use of architecture diagrams and figuring out what technologies were to be used. The final chosen technologies were typescript being used for the logic in the app, making use the framework expo, to fetch the needed data of the coins the API coinranking was used, and as for the database the integrated browser database localStorage was utilised. The initial developing of the app started out as setting up a project and installing expo onto it, which gave a basic home page layout which had an empty home and second page from which I could start developing. I then began with finding myself around the API, which I began by just testing out whether fetching data worked by printing everything out to the console. Next, I began with designing of the UI, which was primarily just the tables. To test out the appearance of the tables I first hard coded an array of coins with random values for each property to see how they would look in the table, as well as figuring out how to display an array of objects in a table, while some of the properties in each cell may change. Once that was finalised, I began figuring out how to map the data retrieved from the API into the array of coins which would be displayed in the table. After that came the first debugging and figuring the correct order to put all the functions in, to

get the data to be displayed. After a few minor tweaks and visual styling fixes, I moved on to creating the second page which displays the wallet and includes even more logic which uses the data from the API combined with the user inputted data to calculate the holdings values and profits for each coin.



As for changes made between the initial planning and the final product, I decided to utilise typescript for the ‘backend’ logic as well since the data base used was the browsers built in localStorage, and since backend is generally used to handle requests for retrieving data from a database, and the database is directly connected to the frontend I didn’t add a backend. And I chose to use the localStorage database since I believe it is most appropriate for my app since the only data I would store in a database is the users input for quantity and investment, and additionally the localStorage is available only on the physical device the browser opens the app in and not on some sort of server run backend, and since people that would potentially use this app care a lot for safety would rather use an app where their user data is not available on any server to be seen but only by them on their device ensuring the safety of their data.