FOOD SHOP MANAGEMENT

Câmpan Tudor-Cosmin

Contents

General Presentation

My goal was to learn some Typescript and working with React library starting from scratch (and maybe some basic html/css). I started by doing some beginner tutorials for react and html/css and reading some syntax guides for javascript. Then I watched video guides an how to start developing in react. I have chosen an app that I have done in Java in order to already know what to do.

The web app consists in an food order system. It is managed by the admin who can add products, remove products and edit them. The normal user can order food from a table that is shown and gets a bill. Of course, a new user can register. It has a login system that recognizes if you have logged in as an user or as an admin. It is useful for restaurants that have their delivery.

Features

The **client** can:

-Register;

-Visualise the set of products;

-Select the products that he/she wants(if they are in stock);

-Navigate between the pages;

-Log In/Log Out;

-See the bill with the ordered products, price and order id.

The **admin** can:

-Add products in the list ;

-Edit a product by its name;

-Remove a product by its name.

The **app** can:

-Recogize which user is logged in or whether someone is logged in or not;

-Store the data in a backend made in a sepparate project;

-Handle the wrong data introduced by the user and give proper error messages.

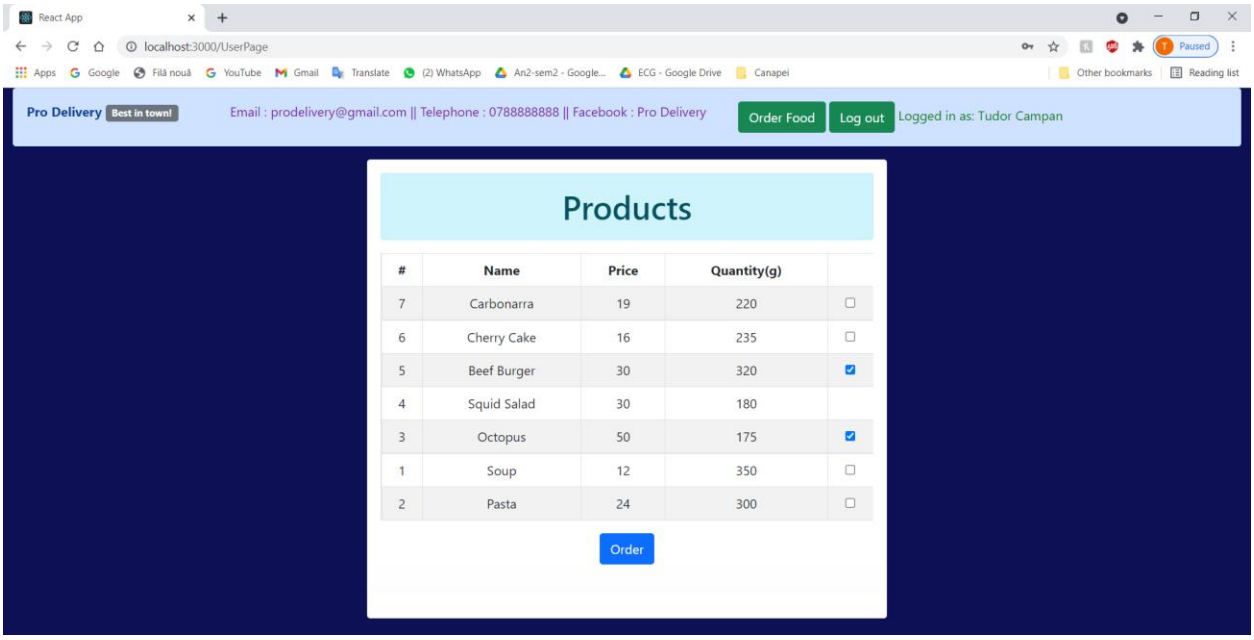
-Modify the product(s) quantity when an order is placed.

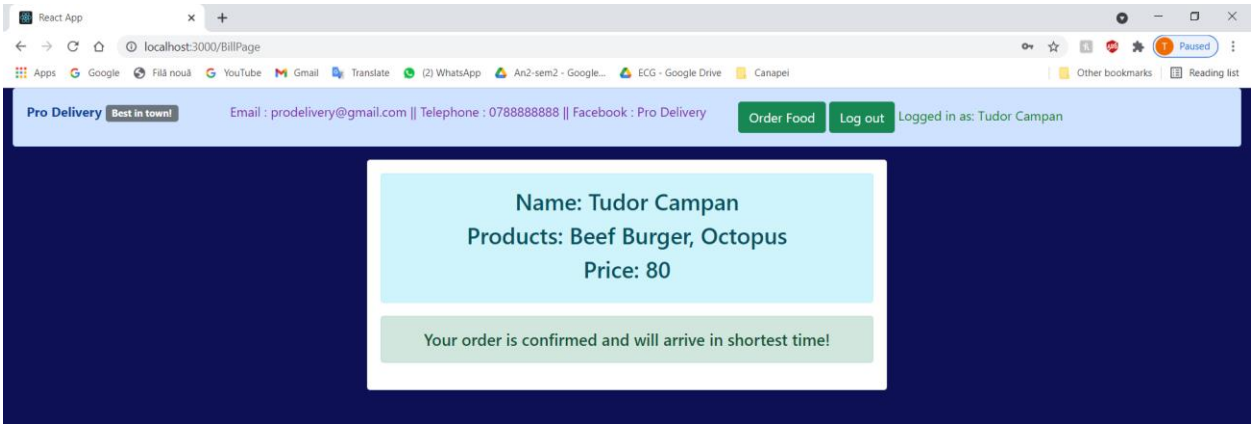
-Also, the app features a header from where you can navigate between pages, shows contact info and shows which user is logged in.

Use cases

Use case 1:

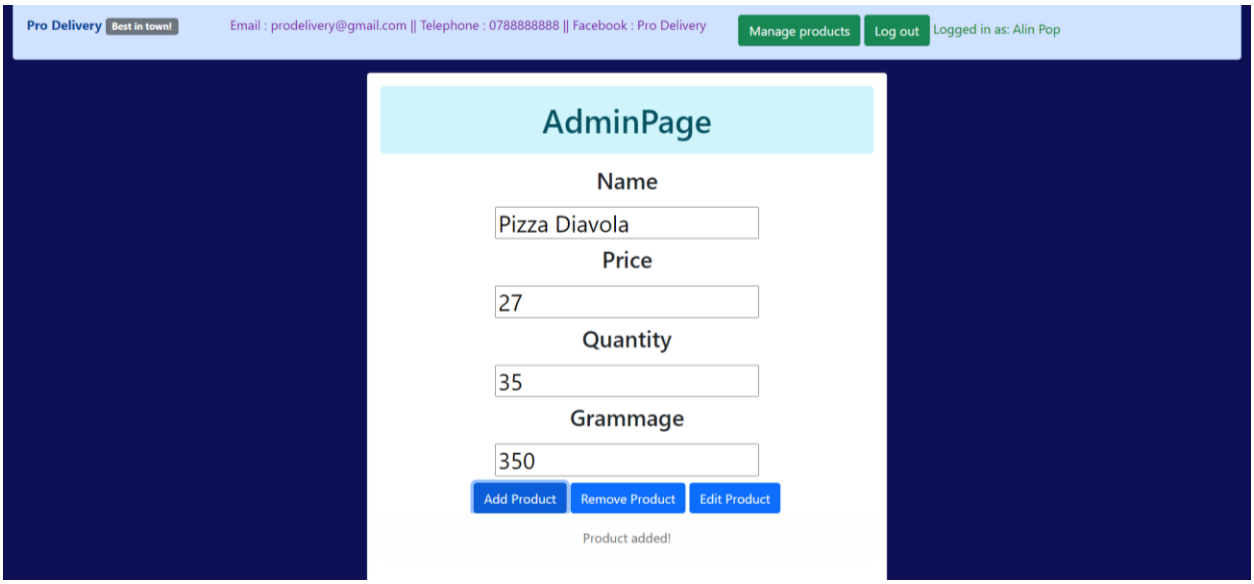
An user wants to order Beef Burger and Octopus. He accesses the app, logs in and the page with the products is shown. He selects the products, click order and the bill page with information on the price and ordered products is shown.





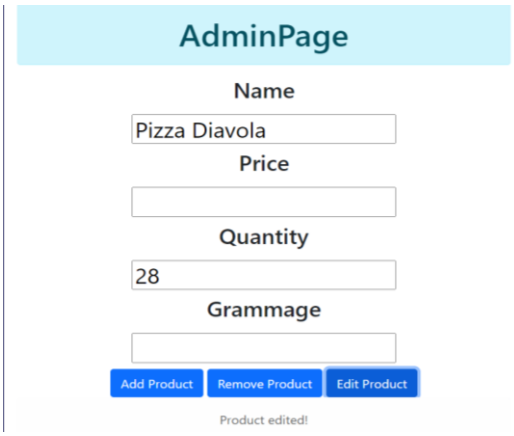
Use case 2:

The admin wants to add a product called Pizza Diavola. He logs in, clicks manage products and will be redirected to the admin page. He fill al the text boxes with information about the product and clicks add product. The product will be added, present in the JSON file where data is stored and visible in the list of products which is shown to the user.



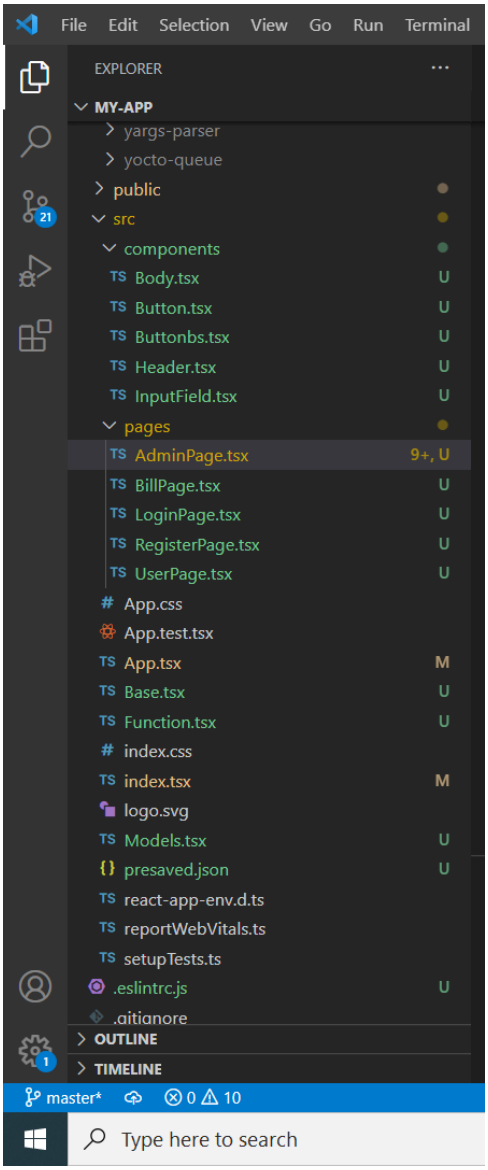
Use case 3:

The admin wants to edit the product called Pizza Diavola and put a different quantity. He types the product name and the wanted quantity, clicks Edit Product and the product will be edited.

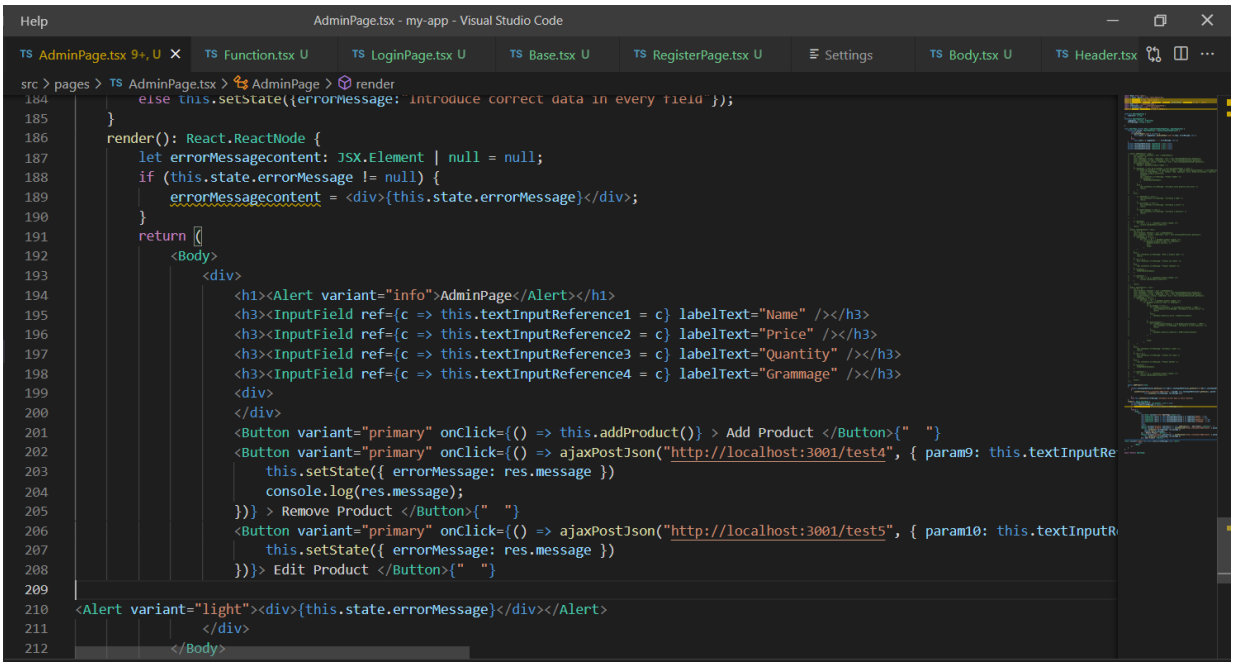


Construction

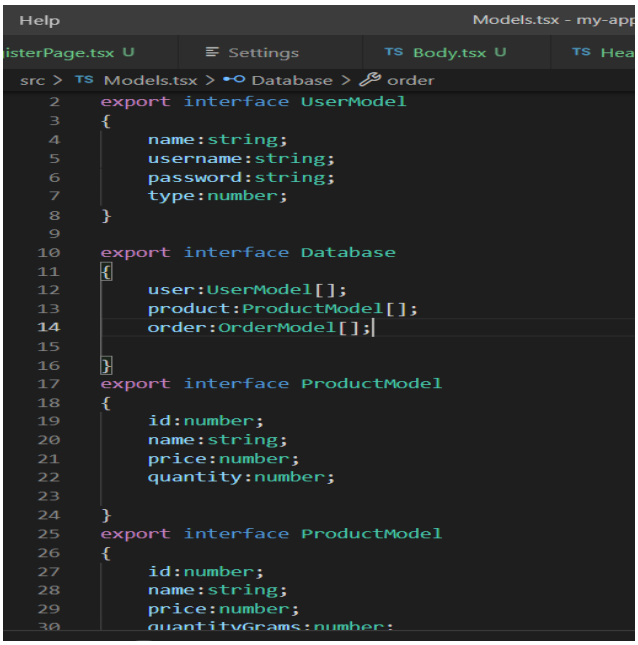
The program is constructed using React principles(diviging the program in as many components as you can, in order to be more modular) and using some OOP style(for the entities like User, Products, Orders etc.)



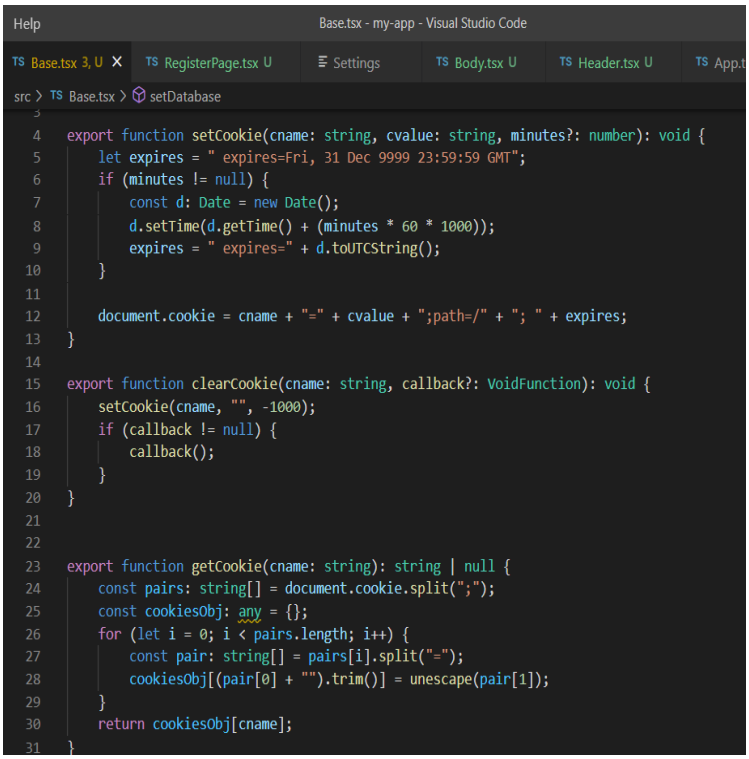
I used components for Body, Button, Header and InputField and I made a different file for each page. The components are reused in every place they are present and they each have props which are changed acording to the requirements. The pages are Register,Log In, User Page, Admin Page and Bill Page. I used **React Router** to navigate between the pages.



The Models involved OOP Principles as I created interfaces with properties for every model that I needed to use.



I stored the “in-app” data such as which user is logged in using cookies, more specific 3 functions : setCookie, getCookie and clearCookie.

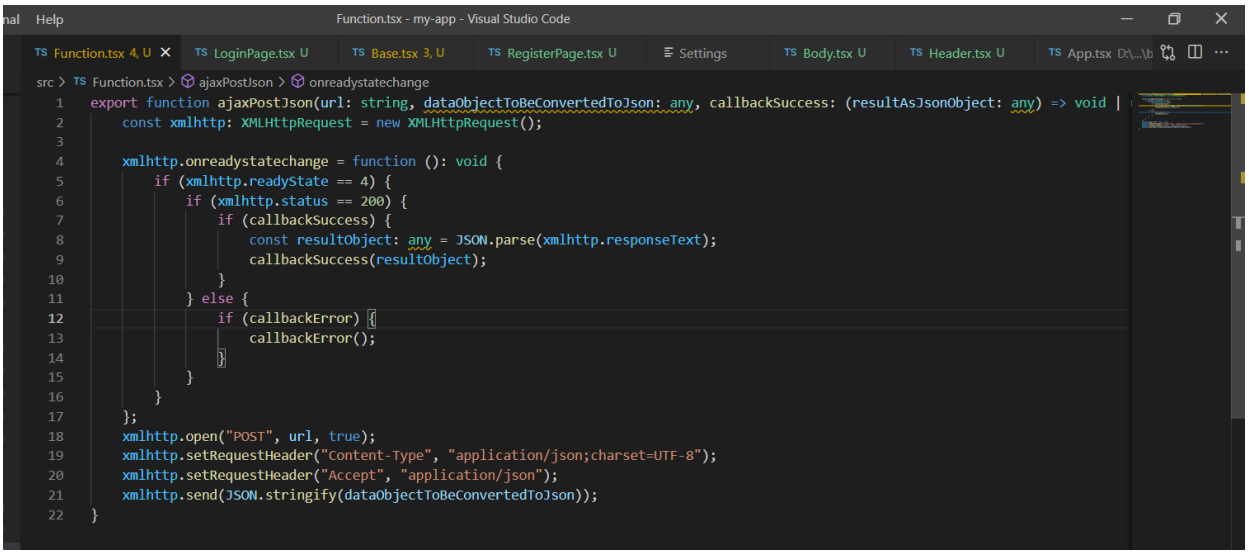


Data storage:

Initially, I had no volatile storage besides a JSON file which I pre-wrote and saved it in my app folder . I used a database which I read from the file and stored in system local storage.

But, since I was almost done with the rest, I developed a back-end app on port 3001, where I moved 75% of the logic of the program, manipulated the data and stored the data. It is stored in a JSON file which remains untouched until I change the data, so now I have a volatile way of storage.

I linked the front end to the backend with a function that was provided to me: ajaxPostJson.



Interface Design:

For the interface design part, react.Component library was used and as a helper I imported the **React Bootstrap** library to help me provide a nicer design.

Conclusion and future improvements

As a conclusion, I can say that I was introduced in the web development world and I learned a little bit of Typescript using the React framework. It wasn't that easy, but not really hard and I am happy that I used my time to improve my software development skills. I'd like to thank my collegue Andrei Vass for all the help that he provided (which was a lot in my opinion).

In the future, I would definetly improve the design of the app. Also, I'd like to add more functionalities as the app is really simplistic at the moment(such as increasing the number of a specific item when ordering). Also, more admin features can be added , such as accepting orders and comunicating with the clients.