INF-2300: Computer Communication

Assignment 2

Iver Mortensen

UiT id: imo059@uit.no

GitHub user: Solcelle

October 5, 2023

1 Introduction

This report describes the design and implementation of a web client application using the React framework in JavaScript. The goal is to create a functional web client capable of making HTTP requests and communicating with a server.

2 Technical background

The implementation of this assignment consisted of using the JavaScript framework React to handle http requests. React would serve as the user interface of the web-application and the functionality to perform the different type of requests to the server. Reacts component-based structure allows simple and reusable code for a simple web application. The react framework simplifies data handling which makes it an ideal framework for seamless server communication.

3 Design

The design of the website is created in such a way that it is easy for the user to understand what each button does. These buttons include a checkbox that marks the task as done, a delete button that deletes a specific task when it is no longer required. And an add task button with a text field to add new tasks.

4 Implementation

4.1 GET request

The web client was built using the React, trying both JavaScript fetch and the axios library to do request to the server. When the website is opened or refreshed a GET request to the server to get the todo list stored currently on the server. This ensures that the web client is up to date with the server.

4.2 POST request

The web client contains a text field which is used to send POST requests to the server. These request are sendt in json format containing the text of the todo. The server then gives this text an ID and a "done" status used to indicate if the todo has been completed. This is sendt back to the client in the response. This is stored in the todo on the client side as well.

4.3 PUT and DELETE request

Each todo on the user interface on the web client has a checkbox and a delete button. Each time the checkbox of a todo is changed, a PUT request of the todo's ID is sent to the server with a updated "done" status, indicating the change of the todo.
If the delete button of a todo is pressed, the client removes the todo to update the client side. A DELETE request of the todos ID is sent to the server to update the server side.

5 Discussion

When implementing the web client I did not see the need to use the GET request on individual todos. In addition I didn't see the need for a delete all button. Each todo has its own delete button which gives more control. If all todos need to be revmoved they can be removed with its remove button. It reduces the clutter of the web client and keeps its simplicity.

6 Conclusion

The main takeaway from this project was learning and experiencing the client-side of web development and learning how to use the react framework in JavaScript. The implementation was successful as it contains all the functionality required in the assignment paper, with the exception of a "delete all" and "get item id", as I did not see a need for these features.