Gabriel Ogungbade **Assignment 2**

Information Technologies

ITMI23SP 02.02.2025

Full stack programming 2025

Miika Reijonen

# Assignment report

GitHub*: https://github.com/w0daz/learning\_assignment\_2*

* 1. **(The first task or feature.)**

For the first task, I downloaded the project base from the GitHub Classroom environment and set it up locally. After navigating to the client directory, I installed the dependencies using the npm i command.

I verified that the setup was correct by starting the app with npm run dev, and it loaded successfully in the browser.

* 1. **(Second task or feature.)**

I added a navbar element to the top of the application. This was implemented using a nav tag in App.tsx. I styled the navbar by giving it a background color (burlywood) in the index.css file and adding a fixed position to ensure it stayed at the top.

The navbar contains navigation text, "WhatToDoApp," for the application.

A computer code with text

Description automatically generated with medium confidence

* 1. **(Third task or feature.)**

I added a form element inside the div with the container class in App.tsx. Inside the form, I included a text input field with a placeholder "New task" and a button labelled "Add new task." Below the form, I added a <ul> element to prepare for displaying tasks, though I left the list content empty at this stage.

**A computer screen shot of text

Description automatically generated**

* 1. **(Fourth task or feature.)**

I enabled React’s useState hook by importing it and creating two state variables: taskList (an array to hold tasks) and newTask (for the input value).

Inside the App component, I added a handleSubmit function that updates the taskList with the new task and clears the input field after submission. This function is connected to the form’s onSubmit event, and I used the setNewTask function to dynamically update the input field as the user types.

**A computer screen with text

Description automatically generated**

* 1. **(Fifth task or feature.)**

I updated the <ul> element to display tasks dynamically. Using the .map() method, I iterated over the taskList array and rendered each task as an <li> element. Each <li> has a unique key based on the index from .map().

**A computer screen shot of a computer code

Description automatically generated**

* 1. **(Sixth task or feature.)**

I implemented task deletion by creating a deleteTask function in the App component. This function filters out the selected task from taskList and updates the state with the filtered array.

To integrate this functionality, I added a "Delete" button to each <li> and used the onClick event to call deleteTask with the corresponding task.

A computer code with colorful text

Description automatically generated with medium confidence