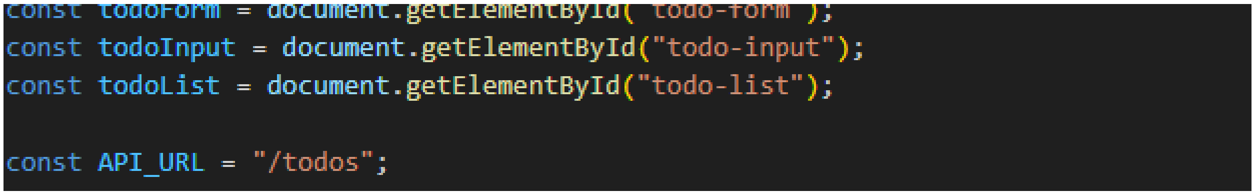
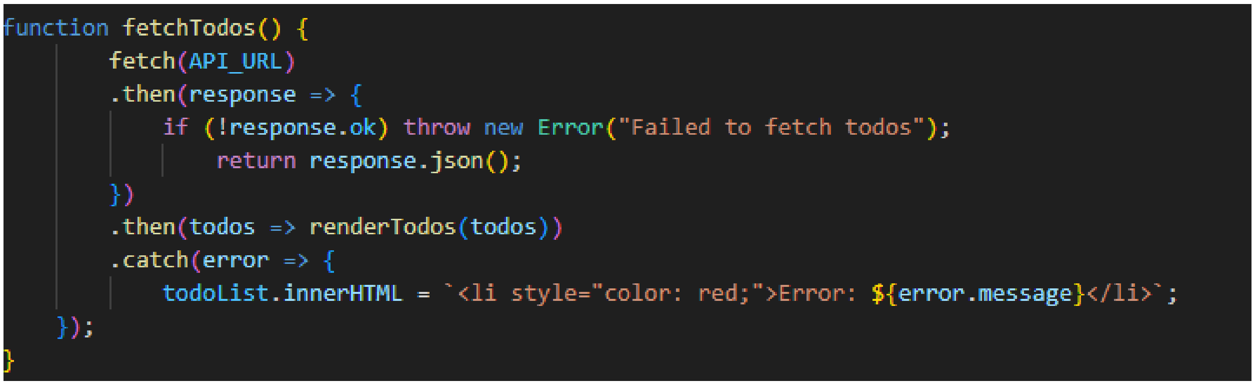
Name: Jherick Isaiah F. Montemar

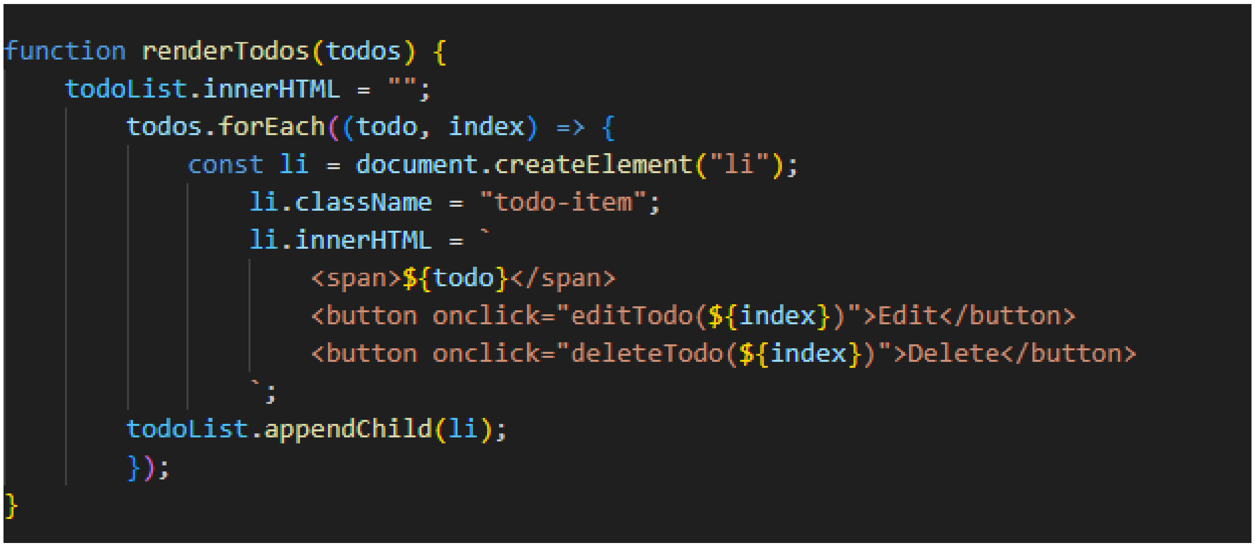


```
const todoForm = document.getElementById("todo-form");
const todoInput = document.getElementById("todo-input");
const todoList = document.getElementById("todo-list");

const API_URL = "/todos";
```

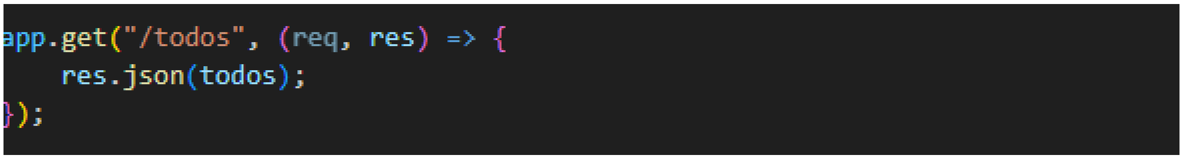
On this part this is the definition for todo form, I also already declare the url so that i just call the name whenever I try to request to API URL.



```
function fetchTodos() {
        fetch(API_URL)
        .then(response => {
            if (!response.ok) throw new Error("Failed to fetch todos");
                return response.json();
        })
        .then(todos => renderTodos(todos))
        .catch(error => {
            todoList.innerHTML = `<li style="color: red;">Error: ${error.message}</li>`;
    });
}
```

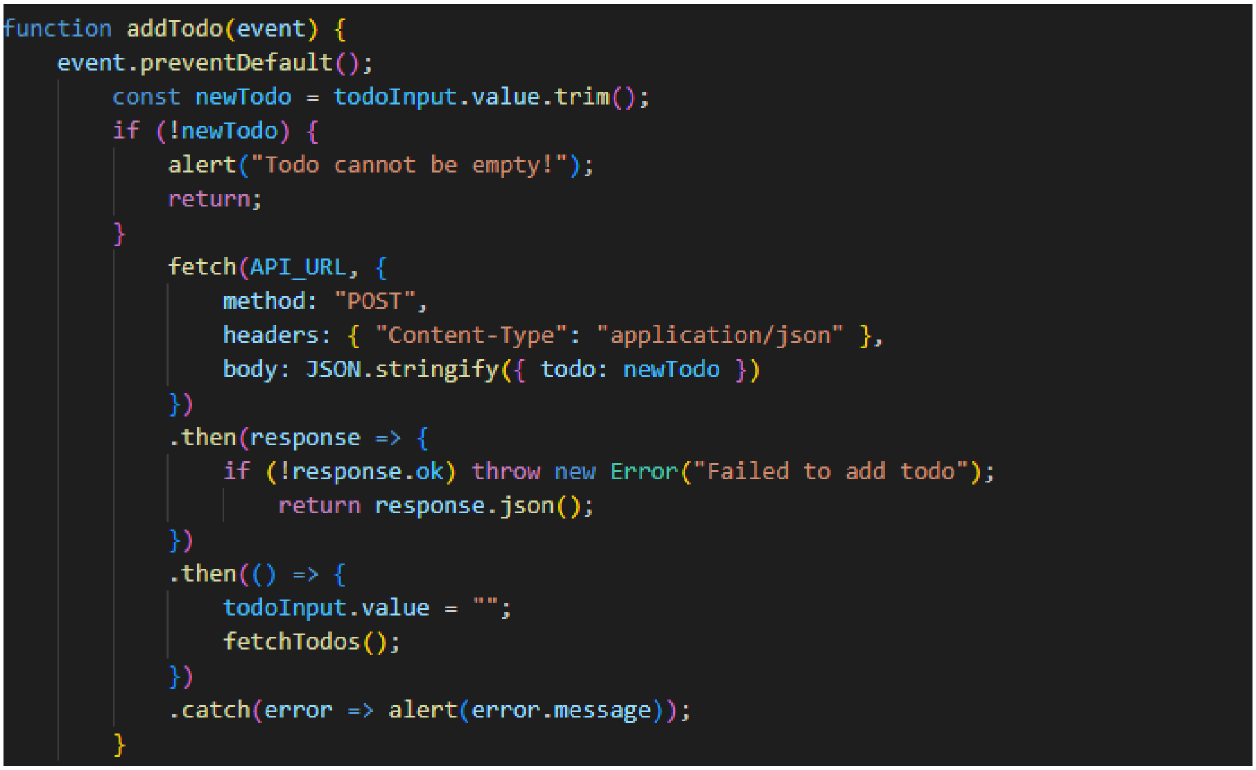


```
function renderTodos(todos) {
    todoList.innerHTML = "";
        todos.forEach((todo, index) => {
            const li = document.createElement("li");
                li.className = "todo-item";
                li.innerHTML = `
                    <span>${todo}</span>
                    <button onclick="editTodo(${index})">Edit</button>
                    <button onclick="deleteTodo(${index})">Delete</button>
                `;
        todoList.appendChild(li);
        });
}
```

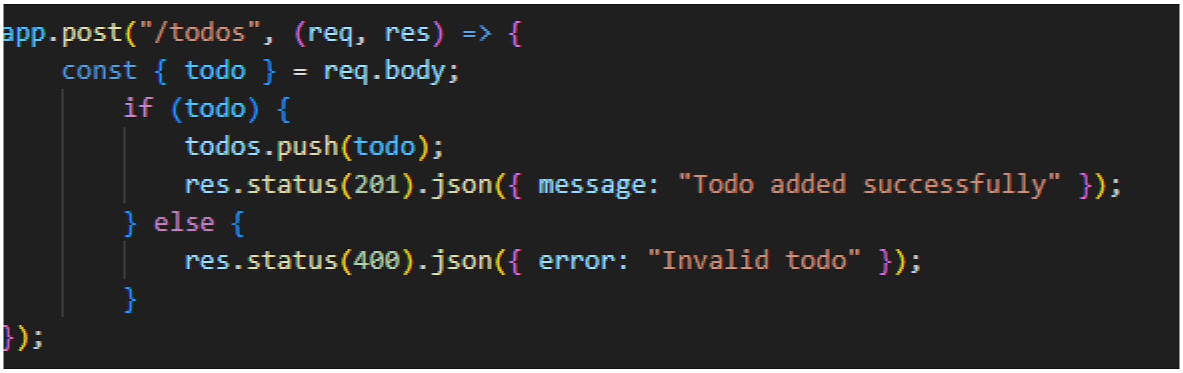


```
app.get("/todos", (req, res) => {
    res.json(todos);
});
```

These are the part for showing the list of to do task.

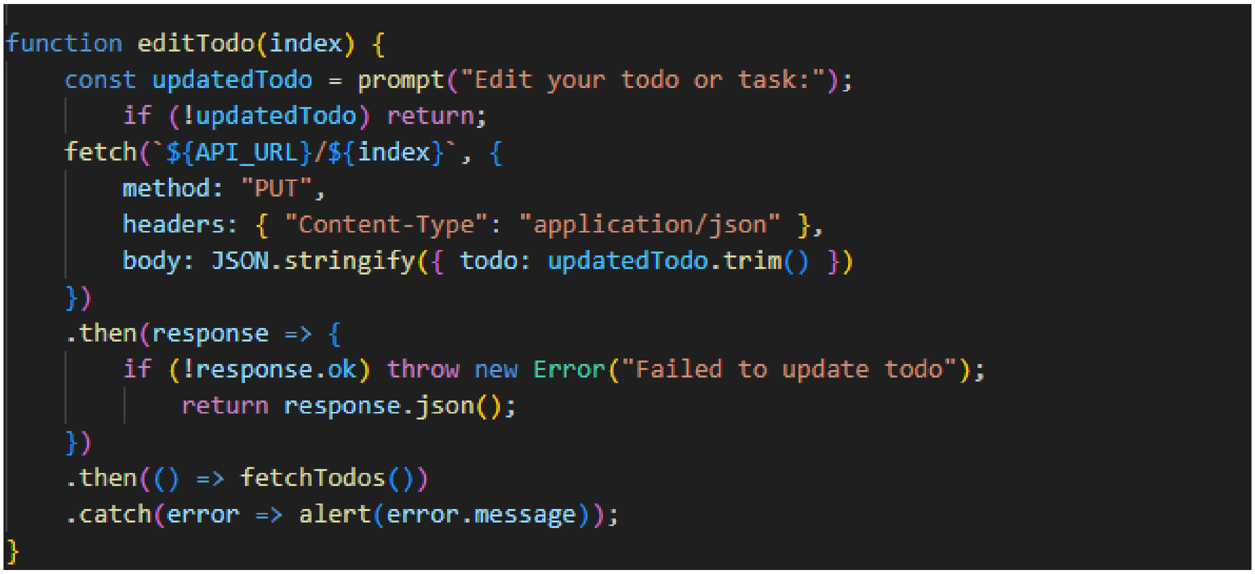


```
function addTodo(event) {
    event.preventDefault();
        const newTodo = todoInput.value.trim();
        if (!newTodo) {
            alert("Todo cannot be empty!");
            return;
        }
            fetch(API_URL, {
                method: "POST",
                headers: { "Content-Type": "application/json" },
                body: JSON.stringify({ todo: newTodo })
            })
            .then(response => {
                if (!response.ok) throw new Error("Failed to add todo");
                    return response.json();
            })
            .then(() => {
                todoInput.value = "";
                fetchTodos();
            })
            .catch(error => alert(error.message));
        }
```

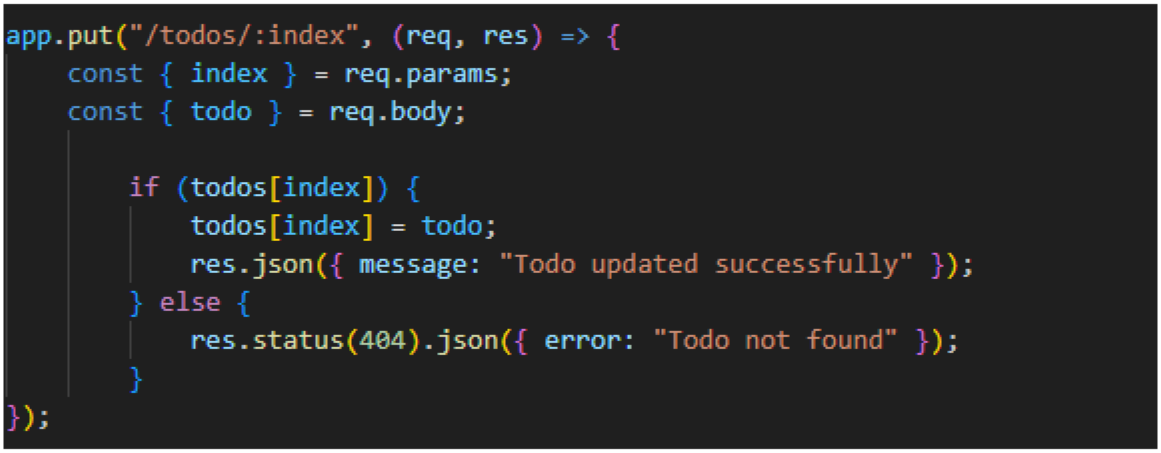


```
app.post("/todos", (req, res) => {
    const { todo } = req.body;
        if (todo) {
            todos.push(todo);
            res.status(201).json({ message: "Todo added successfully" });
        } else {
            res.status(400).json({ error: "Invalid todo" });
        }
});
```

For adding the new list into the array data.

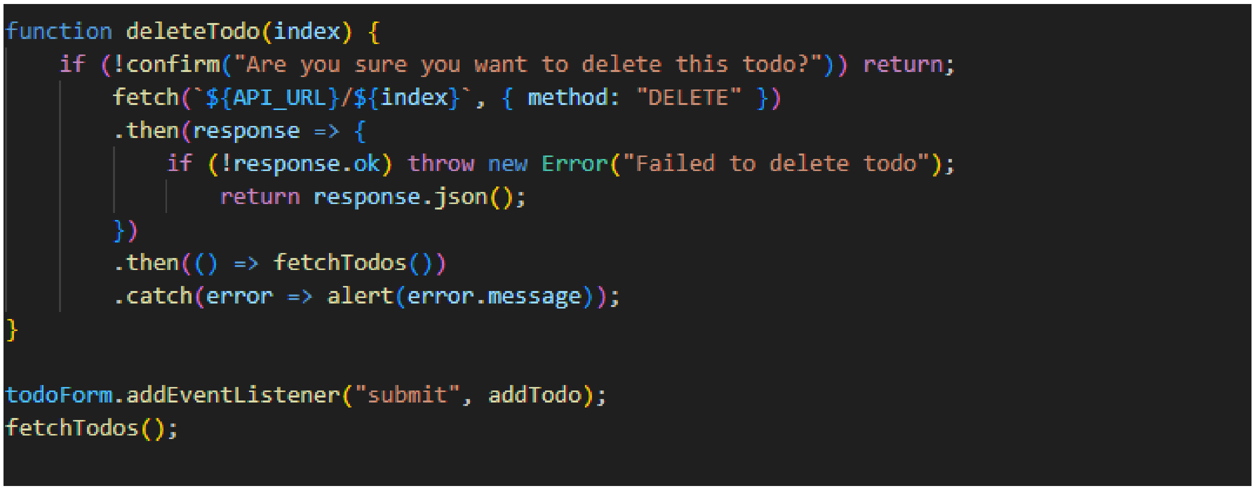


```
function editTodo(index) {
    const updatedTodo = prompt("Edit your todo or task:");
        if (!updatedTodo) return;
    fetch(`${API_URL}/${index}`, {
        method: "PUT",
        headers: { "Content-Type": "application/json" },
        body: JSON.stringify({ todo: updatedTodo.trim() })
    })
    .then(response => {
        if (!response.ok) throw new Error("Failed to update todo");
            return response.json();
    })
    .then(() => fetchTodos())
    .catch(error => alert(error.message));
}
```

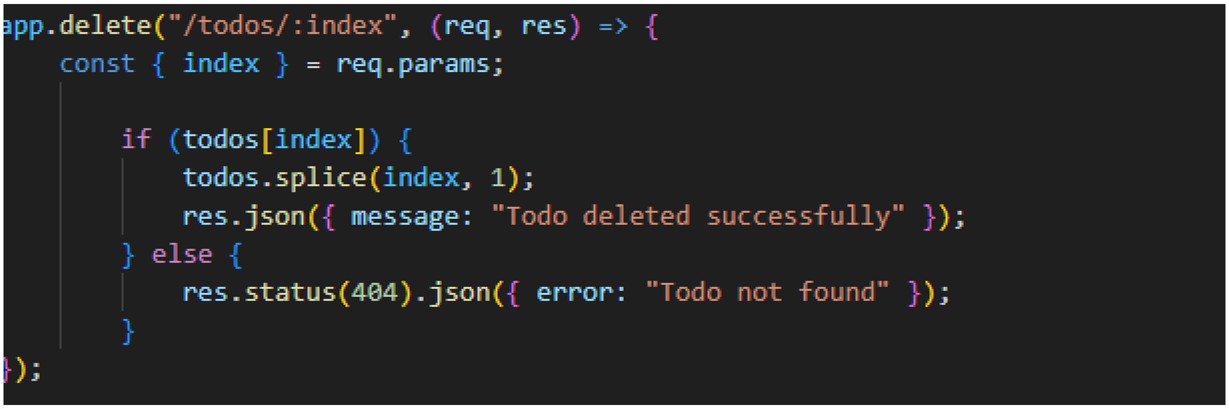


```
app.put("/todos/:index", (req, res) => {
    const { index } = req.params;
    const { todo } = req.body;

        if (todos[index]) {
            todos[index] = todo;
            res.json({ message: "Todo updated successfully" });
        } else {
            res.status(404).json({ error: "Todo not found" });
        }
});
```

For editing what was on the added to do task.



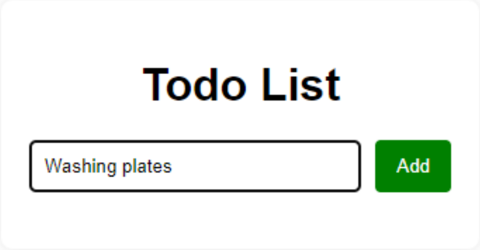
```
function deleteTodo(index) {
    if (!confirm("Are you sure you want to delete this todo?")) return;
        fetch(`${API_URL}/${index}`, { method: "DELETE" })
        .then(response => {
            if (!response.ok) throw new Error("Failed to delete todo");
                return response.json();
        })
        .then(() => fetchTodos())
        .catch(error => alert(error.message));
}

todoForm.addEventListener("submit", addTodo);
fetchTodos();
```



```
app.delete("/todos/:index", (req, res) => {
    const { index } = req.params;

        if (todos[index]) {
            todos.splice(index, 1);
            res.json({ message: "Todo deleted successfully" });
        } else {
            res.status(404).json({ error: "Todo not found" });
        }
});
```

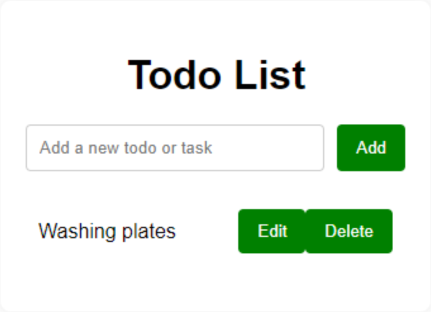
For deleting the todo or task.

And now I try to run if everything works



Todo List

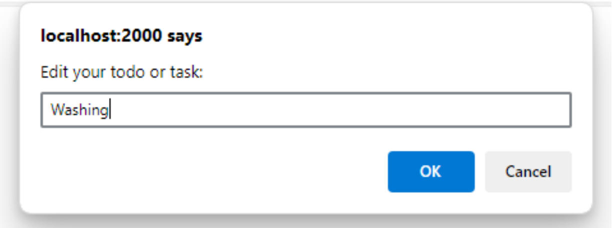
Washing plates



Todo List

Add a new todo or task

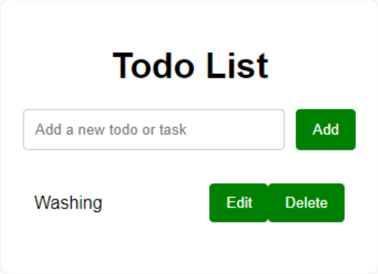
Washing plates



localhost:2000 says

Edit your todo or task:

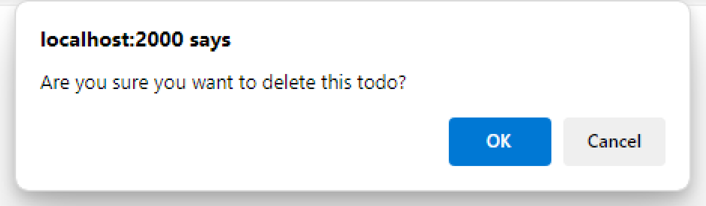
Washing



Todo List

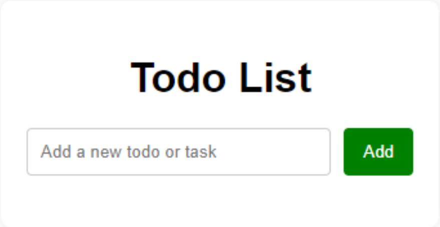
Add a new todo or task

Washing



localhost:2000 says

Are you sure you want to delete this todo?



Everything is working, we can now add, delete, edit, and show the todo list with node js as our backend.