1. **Instalar express fs**

npm install express fs

1. **Server:**
2. const express = require("express");
3. const fs = require("fs");
4. const app = express();
5. const port = 3000;
6. app.use(express.json());
7. app.use(express.static("public")); // Servir archivos estáticos
8. // Leer todos los elementos
9. app.get("/todos", (req, res) => {
10. fs.readFile("todos.json", (err, data) => {
11. if (err) throw err;
12. res.json(JSON.parse(data));
13. });
14. });
15. // Crear un nuevo elemento
16. app.post("/todos", (req, res) => {
17. fs.readFile("todos.json", (err, data) => {
18. if (err) throw err;
19. let todos = JSON.parse(data);
20. let newTodo = { id: Date.now(), title: req.body.title, checked: false };
21. todos.push(newTodo);
22. fs.writeFile("todos.json", JSON.stringify(todos), (err) => {
23. if (err) throw err;
24. res.json(newTodo);
25. });
26. });
27. });
28. // Actualizar un elemento
29. app.put("/todos/:id", (req, res) => {
30. fs.readFile("todos.json", (err, data) => {
31. if (err) throw err;
32. let todos = JSON.parse(data);
33. let todo = todos.find((t) => t.id == req.params.id);
34. if (todo) {
35. todo.title = req.body.title;
36. todo.checked = req.body.checked;
37. fs.writeFile("todos.json", JSON.stringify(todos), (err) => {
38. if (err) throw err;
39. res.json(todo);
40. });
41. } else {
42. res.status(404).send("Todo not found");
43. }
44. });
45. });
46. // Eliminar un elemento
47. app.delete("/todos/:id", (req, res) => {
48. fs.readFile("todos.json", (err, data) => {
49. if (err) throw err;
50. let todos = JSON.parse(data);
51. todos = todos.filter((t) => t.id != req.params.id);
52. fs.writeFile("todos.json", JSON.stringify(todos), (err) => {
53. if (err) throw err;
54. res.send("Todo deleted");
55. });
56. });
57. });
58. app.listen(port, () => {
59. console.log(`Server running at http://localhost:${port}`);
60. });

**3. Style:**

/\* Include the padding and border in an element's total width and height \*/

\* {

  box-sizing: border-box;

}

/\* Remove margins and padding from the list \*/

ul {

  margin: 0;

  padding: 0;

}

/\* Style the list items \*/

ul li {

  cursor: pointer;

  position: relative;

  padding: 12px 8px 12px 40px;

  background: #eee;

  font-size: 18px;

  transition: 0.2s;

  /\* make the list items unselectable \*/

  -webkit-user-select: none;

  -moz-user-select: none;

  -ms-user-select: none;

  user-select: none;

}

/\* Set all odd list items to a different color (zebra-stripes) \*/

ul li:nth-child(odd) {

  background: #f9f9f9;

}

/\* Darker background-color on hover \*/

ul li:hover {

  background: #ddd;

}

/\* When clicked on, add a background color and strike out text \*/

ul li.checked {

  background: #888;

  color: #fff;

  text-decoration: line-through;

}

/\* Add a "checked" mark when clicked on \*/

ul li.checked::before {

  content: "";

  position: absolute;

  border-color: #fff;

  border-style: solid;

  border-width: 0 2px 2px 0;

  top: 10px;

  left: 16px;

  transform: rotate(45deg);

  height: 15px;

  width: 7px;

}

/\* Style the close button \*/

.close {

  position: absolute;

  right: 0;

  top: 0;

  padding: 12px 16px 12px 16px;

}

.close:hover {

  background-color: #f44336;

  color: white;

}

/\* Style the header \*/

.header {

  background-color: #f44336;

  padding: 30px 40px;

  color: white;

  text-align: center;

}

/\* Clear floats after the header \*/

.header:after {

  content: "";

  display: table;

  clear: both;

}

/\* Style the input \*/

input {

  margin: 0;

  border: none;

  border-radius: 0;

  width: 75%;

  padding: 10px;

  float: left;

  font-size: 16px;

}

/\* Style the "Add" button \*/

.addBtn {

  padding: 10px;

  width: 25%;

  background: #d9d9d9;

  color: #555;

  float: left;

  text-align: center;

  font-size: 16px;

  cursor: pointer;

  transition: 0.3s;

  border-radius: 0;

}

.addBtn:hover {

  background-color: #bbb;

}

**4. index.js**

document.addEventListener("DOMContentLoaded", getTodos);

function getTodos() {

  fetch("/todos")

    .then((response) => response.json())

    .then((data) => {

      const ul = document.getElementById("myUL");

      ul.innerHTML = "";

      data.forEach((todo) => {

        const li = document.createElement("li");

        li.textContent = todo.title;

        li.id = todo.id;

        if (todo.checked) {

          li.classList.add("checked");

        }

        ul.appendChild(li);

        addCloseButton(li);

      });

    })

    .catch((error) => console.error("Error fetching todos:", error));

}

function newElement() {

  const inputValue = document.getElementById("myInput").value;

  if (inputValue === "") {

    alert("You must write something!");

    return;

  }

  fetch("/todos", {

    method: "POST",

    headers: { "Content-Type": "application/json" },

    body: JSON.stringify({ title: inputValue }),

  })

    .then((response) => response.json())

    .then((todo) => {

      const ul = document.getElementById("myUL");

      const li = document.createElement("li");

      li.textContent = todo.title;

      li.id = todo.id;

      ul.appendChild(li);

      addCloseButton(li);

    })

    .catch((error) => console.error("Error adding todo:", error));

  document.getElementById("myInput").value = "";

}

function addCloseButton(li) {

  const span = document.createElement("SPAN");

  const txt = document.createTextNode("\u00D7");

  span.className = "close";

  span.appendChild(txt);

  li.appendChild(span);

  span.onclick = function () {

    fetch(`/todos/${li.id}`, { method: "DELETE" })

      .then(() => {

        li.style.display = "none";

      })

      .catch((error) => console.error("Error deleting todo:", error));

  };

  li.onclick = function () {

    fetch(`/todos/${li.id}`, {

      method: "PUT",

      headers: { "Content-Type": "application/json" },

      body: JSON.stringify({

        title: li.textContent,

        checked: !li.classList.contains("checked"),

      }),

    })

      .then((response) => response.json())

      .then((todo) => {

        li.classList.toggle("checked");

      })

      .catch((error) => console.error("Error updating todo:", error));

  };

}

**5. index.html**

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <link rel="stylesheet" href="style.css" />

    <title>TodoList</title>

  </head>

  <body>

    <div id="myDIV" class="header">

      <h2>My To Do List</h2>

      <input type="text" id="myInput" placeholder="Title..." />

      <span onclick="newElement()" class="addBtn">Add</span>

    </div>

    <ul id="myUL"></ul>

    <script src="index.js"></script>

  </body>

</html>

**6. todos.json**

[

  { "id": 1, "title": "Hit the gym", "checked": false },

  { "id": 2, "title": "Pay bills", "checked": true },

  { "id": 3, "title": "Meet George", "checked": false },

  { "id": 4, "title": "Buy eggs", "checked": false },

  { "id": 5, "title": "Read a book", "checked": false },

  { "id": 6, "title": "Organize office", "checked": false }

]

**7. Estructura de Carpeta**

Imagen que contiene Diagrama

El contenido generado por IA puede ser incorrecto.

**8. run server**

node server.js