**Мини-проект: «Менеджер задач» (To-Do List)**

**🔧 Что создаём?**

Интерактивное приложение "Список задач" — ты можешь:

* Добавлять задачи
* Отмечать как выполненные
* Удалять задачи
* Хранить задачи в массиве
* Отображать их динамически в DOM

**🧱 Структура HTML**

html

КопироватьРедактировать

<h1>Мои задачи</h1>

<input type="text" id="taskInput" placeholder="Новая задача">

<button id="addBtn">Добавить</button>

<ul id="taskList"></ul>

**🧠 Логика на JavaScript (базовая версия)**

js

КопироватьРедактировать

const input = document.getElementById("taskInput");

const btn = document.getElementById("addBtn");

const list = document.getElementById("taskList");

let tasks = [];

btn.addEventListener("click", () => {

const text = input.value.trim();

if (text === "") return;

const task = { text: text, done: false };

tasks.push(task);

input.value = "";

renderTasks();

});

function renderTasks() {

list.innerHTML = "";

tasks.forEach((task, index) => {

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

li.textContent = task.text;

if (task.done) li.style.textDecoration = "line-through";

// Отметить как выполненную

li.addEventListener("click", () => {

task.done = !task.done;

renderTasks();

});

// Кнопка удаления

const del = document.createElement("button");

del.textContent = "✖";

del.addEventListener("click", (e) => {

e.stopPropagation();

tasks.splice(index, 1);

renderTasks();

});

li.append(del);

list.append(li);

});

}

**✅ Используемые темы**

| **Концепт** | **Как используется** |
| --- | --- |
| DOM | getElementById, createElement, append |
| События | addEventListener, click |
| Массивы | Хранение и перебор задач (forEach) |
| Объекты | Каждая задача — объект {text, done} |
| Функции | renderTasks(), обработчики событий |
| Условные операторы | if (text === "") return; |

**🧠 Идеи для расширения**

1. ✅ Хранение задач в localStorage (сохраняются после перезагрузки)
2. ✅ Фильтрация: Все / Выполненные / Активные
3. ✅ Сортировка задач
4. ✅ Темная тема
5. ✅ Drag-and-drop
6. ✅ Асинхронная загрузка "примеров задач" с fetch()