

Frontend (HTML + CSS + JavaScript

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>To-Do List</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="container">
    <h1>To-Do List</h1>
    <input type="text" id="taskInput" placeholder="Add new task">
    <button onclick="addTask()">Add Task</button>
    <ul id="taskList"></ul>
  </div>
  <script src="script.js"></script>
</body>
</html>
```

CSS (styles.css):

```
body {
  font-family: Arial, sans-serif;
  background-color: #f4f4f4;
  margin: 0;
  padding: 0;
}

.container {
  width: 80%;
  margin: 0 auto;
  max-width: 600px;
  padding: 20px;
  background: #fff;
  border-radius: 8px;
  box-shadow: 0 0 10px rgba(0,0,0,0.1);
}

h1 {
  margin-bottom: 20px;
}
```

```
input[type="text"] {  
  width: 70%;  
  padding: 10px;  
  font-size: 16px;  
}
```

```
button {  
  padding: 10px;  
  font-size: 16px;  
}
```

```
ul {  
  list-style-type: none;  
  padding: 0;  
}
```

```
li {  
  padding: 10px;  
  border-bottom: 1px solid #ddd;  
  display: flex;  
  justify-content: space-between;  
  align-items: center;  
}
```

```
li.completed {  
  text-decoration: line-through;  
  color: #888;  
}
```

```
button.edit, button.delete {  
  background: #007bff;  
  color: #fff;  
  border: none;  
  padding: 5px 10px;  
  cursor: pointer;  
}
```

```
button.delete {
  background: #dc3545;
}
JavaScript (script.js):
document.addEventListener('DOMContentLoaded', () => {
  loadTasks();
});
```

```
function addTask() {
  const taskInput = document.getElementById('taskInput');
  const taskText = taskInput.value.trim();
  if (taskText === '') return;

  fetch('/tasks', {
    method: 'POST',
    headers: { 'Content-Type': 'application/json' },
    body: JSON.stringify({ text: taskText })
  })
  .then(response => response.json())
  .then(task => {
    appendTask(task);
    taskInput.value = '';
  });
}
```

```
function appendTask(task) {
  const taskList = document.getElementById('taskList');
  const li = document.createElement('li');
  li.dataset.id = task.id;

  li.innerHTML = `
    <span class="${task.completed ? 'completed' : ''}">${task.text}</span>
    <button class="edit" onclick="editTask(${task.id})">Edit</button>
    <button class="delete" onclick="deleteTask(${task.id})">Delete</button>
  `;
  taskList.appendChild(li);
}
```

```

function editTask(id) {
  const newText = prompt('Enter new task text:');
  if (newText === null) return;

  fetch(`/tasks/${id}`, {
    method: 'PUT',
    headers: { 'Content-Type': 'application/json' },
    body: JSON.stringify({ text: newText })
  })
  .then(response => response.json())
  .then(task => {
    const taskElement = document.querySelector(`li[data-id="${id}"]`);
    taskElement.querySelector('span').textContent = task.text;
  });
}

```

```

function deleteTask(id) {
  fetch(`/tasks/${id}`, { method: 'DELETE' })
  .then(() => {
    const taskElement = document.querySelector(`li[data-id="${id}"]`);
    taskElement.remove();
  });
}

```

```

function loadTasks() {
  fetch('/tasks')
  .then(response => response.json())
  .then(tasks => {
    tasks.forEach(task => appendTask(task));
  });
}

```

Backend (Node.js with Express)

```
mkdir todo-app
```

```
cd todo-app
```

```
npm init -y
```

```
npm install express body-parser
```

```
const express = require('express');
```

```
const bodyParser = require('body-parser');
```

```
const app = express();
```

```
const port = 3000;

app.use(bodyParser.json());
app.use(express.static('public'));

let tasks = [];
let nextId = 1;

app.post('/tasks', (req, res) => {
  const newTask = { id: nextId++, text: req.body.text, completed: false };
  tasks.push(newTask);
  res.json(newTask);
});

app.get('/tasks', (req, res) => {
  res.json(tasks);
});

app.put('/tasks/:id', (req, res) => {
  const id = parseInt(req.params.id, 10);
  const task = tasks.find(t => t.id === id);
  if (task) {
    task.text = req.body.text;
    res.json(task);
  } else {
    res.status(404).send('Task not found');
  }
});

app.delete('/tasks/:id', (req, res) => {
  const id = parseInt(req.params.id, 10);
  tasks = tasks.filter(t => t.id !== id);
  res.sendStatus(204);
});

app.listen(port, () => {
  console.log(`Server running at http://localhost:${port}`);
});
```

node server.js