

Web-applikation für organisatorische Planung im Alltag auf Basis von C++



Alma Kosuta
Bea Janott
Jake Pettke
Leo Hepting
Mikel Thiele
Philip Adam

Agenda

- Appvorstellung (Problem=>Lösung)
- Ausgangssituation
- Nutzeranforderungen
- Initiierungsphase
- Technische Umsetzung
- Architektur
- Kommunikation
- Request Flow
- Datenspeicherung
- Planungsphase
- Konzeptionsphase (Architektur)
- Implementierungsphase
- Dokumentationsphase
- Abschlussphase

Appvorstellung

+ Neue Liste erstellen

Meine Liste

Zweite Liste

Bald fällige Aufgaben

Meine Liste

+ Neues Todo erstellen

Filter

<input type="checkbox"/> Erste Aufgabe	Hoch	15.02.2026
<input type="checkbox"/> Zweite Aufgabe	Mittel	17.02.2026
<input type="checkbox"/> Dritte Aufgabe	Niedrig	19.02.2026
<input type="checkbox"/> Vierte Aufgabe	Hoch	21.02.2026
<input type="checkbox"/> Fünfte Aufgabe	Mittel	23.02.2026
<input type="checkbox"/> Sechste Aufgabe	Niedrig	25.02.2026

Liste abschließen

Ausgangssituation (Problem)

- Fehlender zentraler Überblick über offene Aufgaben
- Potenzielle Doppelarbeit
- Aufgaben und Fristen werden übersehen



Nutzeranforderungen (Lösung)

- Zentrale Verwaltung von Aufgaben
- Eingabe von: Titel, Beschreibung, Status, Priorität, Deadline
- Einfache Sortier- und Filterfunktionen
- Lokale Speicherung ohne Cloud
- Schnelle Bedienbarkeit und intuitive Oberfläche

The screenshot displays a web application for task management. On the left, a sidebar contains three buttons: '+ Neue Liste erstellen', 'Meine Liste' (selected), and 'Zweite Liste'. The main area, titled 'Meine Liste', shows two task entries. Each entry consists of a checkbox, a title, a priority label, and a due date. The first task is 'Erste Aufgabe' with a red 'Hoch' (High) priority label and a due date of '15.02.2026'. The second task is 'Zweite Aufgabe' with a yellow 'Mittel' (Medium) priority label and a due date of '17.02.2026'. On the right, a filter panel is open, showing options to '+ Neues Todo erstellen' and a 'Filter' button. Below these, there are sections for 'Filtern nach' (Filter by), 'Priorität' (Priority) with a dropdown menu set to 'Alle', 'Name' with a search input 'Nach Name filtern', 'Fälligkeitsdatum' (Due date) with a date input 'TT.MM.JJJJ', and a 'Filter zurücksetzen' (Reset filter) button.

Task Title	Priority	Due Date
<input type="checkbox"/> Erste Aufgabe	Hoch	15.02.2026
<input type="checkbox"/> Zweite Aufgabe	Mittel	17.02.2026

Technische Umsetzung

- Browserbasierte Bedienung über lokalen HTTP-Server
- Speicherung lokal (Socket nimmt Json und speichert in eine lokale Json)
- Programmiersprachen; C++, React-Typescript

Architektur

Frontend (React / Vite)

↓ HTTP

Backend (C++ HTTP-Server)

↓

JsonService

↓

mainDATA.json

API-Kommunikation Frontend ↔ Backend

<u>HTTP</u>	<u>Endpoint</u>	<u>Zweck</u>
GET	/todos	Todos abrufen
POST	/todos	Neues Todo anlegen
PUT	/todos/{id}	Todo ändern
PATCH	/todos/{id}/checked	Status ändern
DELETE	/todos/{id}	Todo löschen



Request Flow

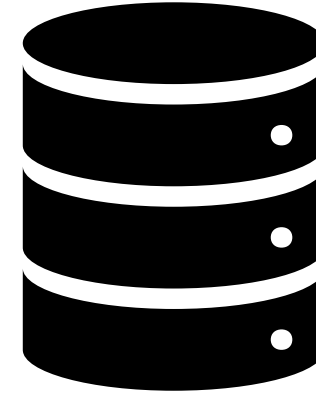
- User klickt „Add“
- Frontend → POST /todos
- Backend → JsonService::AddTodo()
- JSON-Datei wird gespeichert
- Response 201 + neues Todo
- React State Update



Datenspeicherung

mainDATA.json

```
{  
  id: 1,  
  name: "Meine Liste",  
  todos: [  
    {  
      checked: false,  
      priority: "HIGH",  
      dueDate: "15.02.2026",  
      title: "Erste Aufgabe",  
      id: 1  
    },  
    {  
      checked: false,  
      priority: "MEDIUM",  
      dueDate: "17.02.2026",  
      title: "Zweite Aufgabe",  
      id: 10  
    }  
  ]  
}
```



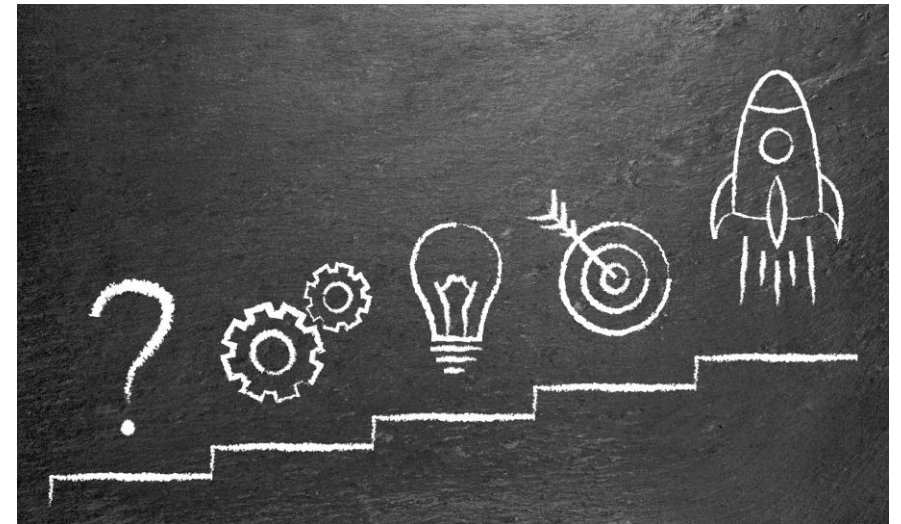
Initiierungsphase

- Gemeinsame Besprechung der Projektidee:
Was soll entwickelt werden?
- Erste Zieldefinition: Funktionsumfang,
grober Zweck der Anwendung.



Planungsphase

- Rollenklärung: Wer übernimmt welche Aufgabe?
- Kompetenzanalyse: Wer kann welche Programmiersprachen und Tools? Oder nutzt es als Gelegenheit es zu lernen?
- Entscheidung über eingesetzte Technologien (z. B. C++, JSON, Sockets, Frontend-Tools)
- Aufteilung in Arbeitsbereiche (Backend, Frontend, Dokumentation)



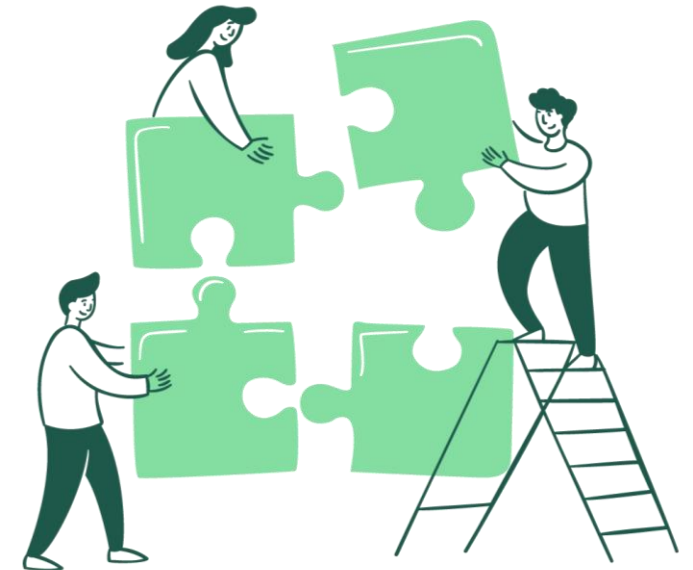
Konzeptionsphase

- Erstellung eines Mockups für das Frontend
(2 Personen)
- Erstellung einer Request Page als Grundlage
für die Kommunikation Backend → Frontend
- Definition der Services im Backend
(Socket-Kommunikation, JSON-Struktur)



Implementierungsphase

- Backend-Team (2 Personen):
 - Person A: Implementierung der Socket-Kommunikation
 - Person B: Erstellung des JSON-Service zur Datenverarbeitung
- Frontend-Team (2 Personen):
 - Umsetzung des Mockups in funktionale UI
 - Implementierung der Request Page



Dokumentationsphase

- Dokumentations-Team (2 Personen):
 - Konsolidierung aller Informationen
 - Sammlung von Screenshots aus Frontend/Backend
 - Erstellung der vollständigen Projektdokumentation: Aufbau, Funktionen, technische Erläuterungen
 - Runbook



Abschlussphase

- Zusammenführen von Backend, Frontend und Dokumentation
- Finaler Projektstand präsentiert
- Ergebnisse gesichert und festgehalten

