

Sprint 1: Projekt-Setup und Grundgerüst – Detaillierte Schritt-für-Schritt-Anleitung

Ihr Name

4. Februar 2025

Inhaltsverzeichnis

1	Einleitung	2
2	Vorhandene Schritte	2
3	Noch zu erledigende Schritte in Sprint 1	2
4	Schritt-für-Schritt-Anleitung	2
4.1	Schritt 1: Erstellung der Ordnerstruktur	2
4.2	Schritt 2: Backend-Projekt mit Express aufsetzen	3
4.3	Schritt 3: Frontend-Projekt mit React als Progressive Web App (PWA) aufsetzen	4
4.4	Schritt 4: Kommunikation zwischen Frontend und Backend verifizieren . . .	4
5	Zusammenfassung von Sprint 1	5

1. Einleitung

Sprint 1 zielt darauf ab, die grundlegende Infrastruktur für das Gym Progress Tracking System aufzubauen und das Basisgerüst zu erstellen. In diesem Sprint integrieren wir alle bereits durchgeführten Schritte (wie die Einrichtung des Code-Editors, das Initialisieren des Git-Repositories und das Einrichten der lokalen Serverumgebung mit Node.js) in ein strukturiertes Gesamtprojekt. Außerdem richten wir das Backend (mit Node.js/Express) und das Frontend (als Progressive Web App mit React) ein und überprüfen die Kommunikation zwischen beiden Komponenten.

2. Vorhandene Schritte

Bereits abgeschlossen wurden:

- Installation und Konfiguration eines Code-Editors (z.B. Visual Studio Code).
- Einrichtung eines Git-Repositories und Erstellung einer sinnvollen Ordnerstruktur (zumindest teilweise).
- Installation von Git und Node.js, sowie Testen eines einfachen Express-Servers.

3. Noch zu erledigende Schritte in Sprint 1

In diesem Sprint werden folgende Aufgaben umgesetzt:

1. Vollständige Erstellung und Festlegung der Ordnerstruktur.
2. Aufsetzen des Backend-Projekts mit Express.
3. Aufsetzen des Frontend-Projekts mit React als Progressive Web App (PWA).
4. Verifizieren der Kommunikation zwischen Frontend und Backend in der lokalen Umgebung.

4. Schritt-für-Schritt-Anleitung

4.1. Schritt 1: Erstellung der Ordnerstruktur

Um eine saubere Projektstruktur zu erhalten, legen Sie im Hauptverzeichnis Ihres Projekts (z.B. `GymTrackingSystem`) folgende Unterordner an:

- **/frontend**: Enthält alle Dateien für die PWA.
- **/backend**: Enthält den Server-Code (Node.js, Express).
- **/docs**: Dokumentationen, Architekturdiagramme und Benutzerhandbücher.
- **/tests**: Unit-Tests und Integrationstests.

Praktische Anleitung über das Terminal:

1. Öffnen Sie das Terminal und navigieren Sie in das Hauptverzeichnis, in dem das Projekt erstellt werden soll (zum Beispiel):

```
cd ~/Projekte
```

2. Erstellen Sie den Hauptordner:

```
mkdir GymTrackingSystem
```

3. Wechseln Sie in diesen Ordner:

```
cd GymTrackingSystem
```

4. Erstellen Sie die Unterordner:

```
mkdir frontend  
mkdir backend  
mkdir docs  
mkdir tests
```

4.2. Schritt 2: Backend-Projekt mit Express aufsetzen

1. Navigieren Sie in den backend-Ordner:

```
cd backend
```

2. Initialisieren Sie ein neues Node.js-Projekt:

```
npm init -y
```

Dadurch wird eine `package.json` erstellt, in der alle Abhängigkeiten und Skripte verwaltet werden.

3. Installieren Sie Express:

```
npm install express
```

4. Erstellen Sie eine Datei namens `server.js` im `backend`-Ordner (siehe Anleitung im vorherigen Dokument). Fügen Sie folgenden Code ein:

```
const express = require('express');  
const app = express();  
const PORT = process.env.PORT || 3000;  
  
app.get('/', (req, res) => {
```

```
    res.send('Server läuft!');
  });

app.listen(PORT, () => {
  console.log('Server läuft auf Port ${PORT}');
});
```

5. Starten Sie den Server:

```
node server.js
```

Öffnen Sie Ihren Browser und rufen Sie `http://localhost:3000` auf, um zu bestätigen, dass der Server wie erwartet reagiert.

4.3. Schritt 3: Frontend-Projekt mit React als Progressive Web App (PWA) aufsetzen

1. Navigieren Sie in den `frontend`-Ordner im Hauptprojektverzeichnis:

```
cd ../frontend
```

2. Nutzen Sie **Create React App** zur schnellen Einrichtung einer React-Anwendung. Geben Sie dazu folgenden Befehl ein:

```
npx create-react-app .
```

(Der Punkt am Ende sorgt dafür, dass die App im aktuellen Ordner installiert wird.)

3. Konfigurieren Sie die Anwendung als PWA, indem Sie in der Datei `index.js` (im Ordner `src`) den Service Worker aktivieren. Ändern Sie (falls notwendig) den Standard-Code, um die PWA-Funktionalität zu unterstützen. Weitere Informationen finden Sie in der Dokumentation von Create React App.

4.4. Schritt 4: Kommunikation zwischen Frontend und Backend verifizieren

1. Stellen Sie sicher, dass der Backend-Server (Express) läuft (z.B. über `node server.js` im Ordner `backend`).
2. Im Frontend können Sie eine einfache API-Anfrage einrichten, um Daten vom Backend abzurufen. Erstellen Sie beispielsweise in der React-App einen Button, der beim Klicken eine Fetch-Anfrage an `http://localhost:3000` sendet.
3. Überprüfen Sie in der Konsole des Browsers und in Ihrem Backend-Terminal, ob die Anfrage empfangen und die erwartete Antwort (z.B. `SServer läuft!`) zurückgegeben wird.

5. Zusammenfassung von Sprint 1

In Sprint 1 haben Sie:

- Eine klare Ordnerstruktur (`/frontend`, `/backend`, `/docs`, `/tests`) erstellt.
- Das Backend mit Node.js und Express eingerichtet, inklusive eines einfachen Servers, der auf `http://localhost:3000` reagiert.
- Das Frontend-Projekt mit React als Progressive Web App initialisiert.
- Die Kommunikation zwischen Frontend und Backend in der lokalen Umgebung überprüft.

Dieser Sprint bildet das Grundgerüst des Projekts und legt die Basis für die weitere Entwicklung der Kernfunktionen des Gym Progress Tracking Systems.