**Studienarbeit zur Anwendung „GymJourney“**

Alexej Kunz (00193829)

Hochschule für Angewandte Wissenschaften Hof

Moderne App- und Webentwicklung

Sommersemester 2023

23.06.2023

Inhalt

[Abbildungsverzeichnis 3](#_Toc133446442)

[1 Einführung 4](#_Toc133446443)

[1.1 Anforderungsanalyse für GymJourney 4](#_Toc133446444)

[1.2 Architektur und Entwurf 5](#_Toc133446445)

[2 Implementierung 6](#_Toc133446446)

[3 Walkthrough durch die GymJourney Website 6](#_Toc133446447)

[4 Fazit 8](#_Toc133446448)

[Literaturverzeichnis 8](#_Toc133446449)

# Abbildungsverzeichnis

[1 React.js Logo 5](#_Toc133441818)

[2 Express.js Logo 5](#_Toc133441818)

[3 MySQL Logo 5](#_Toc133441818)

[4 Technologie-Stack 6](#_Toc133441818)

[5 Login-Seite 7](#_Toc133441818)

[6 Register-Seite 7](#_Toc133441818)

[7 Home-Seite 8](#_Toc133441818)

# Einführung

Die Website GymJourney richtet sich an sportbegeisterte Personen. Sie soll ihnen dabei helfen, möglichst einfach und unkompliziert die Übersicht über ihre Trainingspläne zu behalten. Im Folgenden wird eine Anforderungsanalyse für diese Website durchgeführt.

## Anforderungsanalyse für GymJourney

//Todo: Evtl kurze Einführung in Anforderungsanalyse und warum man diese macht, vlt mit Quelle, evlt weitere Anforderungen durch Anforderungen der Studienarbeit ableiten

* Bei der Anwendung soll es sich um eine Single Page Application (SPA) handeln
* Die Anwendung soll es dem Nutzer ermöglichen, einen Account zu erstellen und sich mit diesem Account anzumelden
* Die Anwendung soll eine sichere Aufbewahrung des Passworts des Nutzers gewährleisten, indem das Passwort zumindest gehasht und gesalted wird
* Die Sitzung des angemeldeten Nutzers soll über Cookies verwaltet werden
* Der Nutzer soll sich über einen Button von der Anwendung ausloggen können
* Es sollen nur angemeldete Nutzer Zugriff auf die Anwendung haben
* Der Nutzer soll in der Lage sein eigene Trainingspläne, Trainingseinheiten (auch Workouts genannt), und Übungen zu erstellen, angezeigt zu bekommen, zu bearbeiten und zu löschen
* Die Anwendung soll bei den Übungen YouTube-Videos einbinden können, sodass der Nutzer diese durch einen Link abspeichern kann und auf der Website anschauen kann
* Die Anwendung soll eine Fehlerseite anzeigen, sollte eine Route nicht gefunden werden
* Die Anwendung soll responsive sein und somit sowohl auf mobilen Endgeräten als auch auf Desktop-Geräten erfolgreich dargestellt werden können
* Die Anwendung soll nach der Hierarchiereihenfolge Trainingspläne, Trainingseinheiten und Übungen aufgebaut sein – wird ein Trainingsplan gelöscht so sollen auch alle Trainingseinheiten und Übungen dieses Trainingsplans gelöscht werden
* Der Nutzer soll nur auf seine eigenen Daten zugreifen können – Zugriffe auf Daten anderer Nutzer sollen vermieden werden
* Die Anwendung soll lediglich im Dark Mode gestaltet werden

## Architektur und Entwurf

In diesem Abschnitt soll die Architektur der Anwendung erläutert werden, wobei zunächst auf den eingesetzten Technologie-Stack eingegangen wird.

Das Frontend der Anwendung wurde mit React.js geschrieben. React.js ist ein JavaScript-Framework, das verwendet wird, um Benutzerschnittstellen zu erstellen. Es ist das beliebteste komponentenbasierte JavaScript-Framework [vgl. 1]. Alternativen hierzu wären unter anderem Angular.js, Vue.js oder Hof.js, jedoch wurde sich aufgrund persönlicher Vorerfahrung mit React.js für React.js entschieden. Für das Backend wurde Node.js mit dem Framework Express verwendet. Node.js an sich ermöglicht es unter anderem serverseitige Applikationen mit JavaScript zu schreiben. Node.js hat von sich aus jedoch nicht die Möglichkeit HTTP-Anfragen wie GET, PUT, POST, DELETE… zu unterstützen, weshalb Express genutzt wird, da es sonst nötig wäre, eine eigene Logik für das Verarbeiten von HTTP-Anfragen zu schreiben. Express ist das beliebteste Node-Webframework [vgl. 2]. Alternative Backend Frameworks wären unter anderem Laravel oder Ruby on Rails, jedoch wurde auch hier aufgrund von persönlichen Vorerfahrungen mit JavaScript für Express entschieden. Um Daten aus dieser Anwendung zu speichern, wird eine Datenbank benötigt. Hier wurde sich für MySQL entschieden, da man hier relativ einfach die geforderten voneinander abhängigen Entitäten abbilden kann und es eines der beliebtesten Datenbankmanagementsysteme ist, was wohl vor Allem daran liegt, dass es häufig in WordPress-Anwendungen zum Einsatz kommt [vgl. 3]. Mögliche Alternative wären hier unter anderem PostgreSQL, SQLite oder auch MariaDB. Eine Übersicht über den Technologie-Stack bietet Abbildung 4.

|  |  |  |
| --- | --- | --- |
| Frontend | Backend | Datenbank |
| React.js  Abbildung 1: React.js Logo [4] | Ein Bild, das Logo enthält.  Automatisch generierte BeschreibungExpress.js  Abbildung 2: Express.js Logo [5] | MySQL  Abbildung 3: MySQL Logo [6] |

Abbildung 4: Technologie-Stack

//TODO: Eventuell folgendes hinzufügen: Diagramm, das die verschiedenen Komponenten der Seite und die Beziehung zueinander zeigt, z.B. mit Flussdiagramm oder Mind Map. Auch Datenbankstruktur könnte man erklären (auch mit hashen und salten der Passwörter).

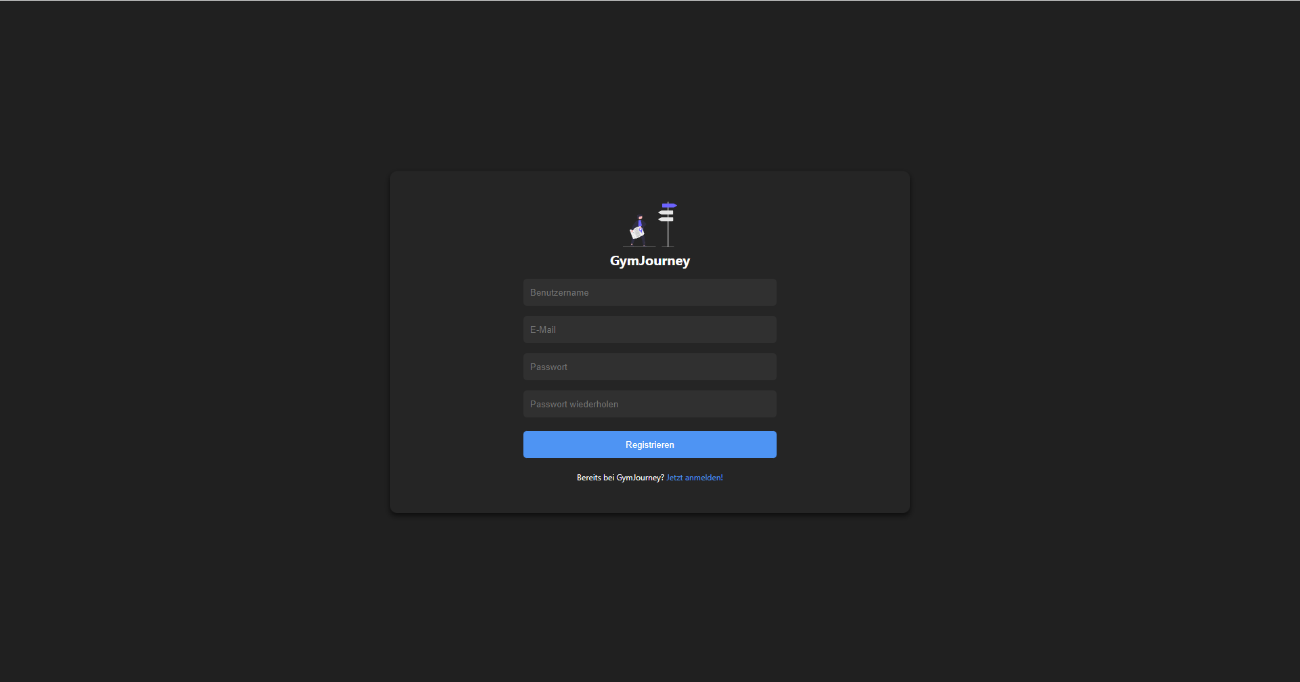
# Implementierung

# Walkthrough durch die GymJourney Website

Im Folgenden wird durch die GymJourney-Website geleitet, um die Bedienung sowie die Möglichkeiten der Website zu dokumentieren.

Zunächst landet der Nutzer auf der in Abbildung 5 dargestellten Login-Seite, zumindest wenn er nicht angemeldet ist. Grund dafür ist, dass der Nutzer für jede Funktionalität, die diese Website bietet, angemeldet sein muss. Sollte der Nutzer noch kein Account besitzen, so kann er auf den Link „Jetzt registrieren“ klicken und landet auf der Register-Seite, um sich einen neuen Account zu erstellen. Diese ist in Abbildung 6 abgebildet.

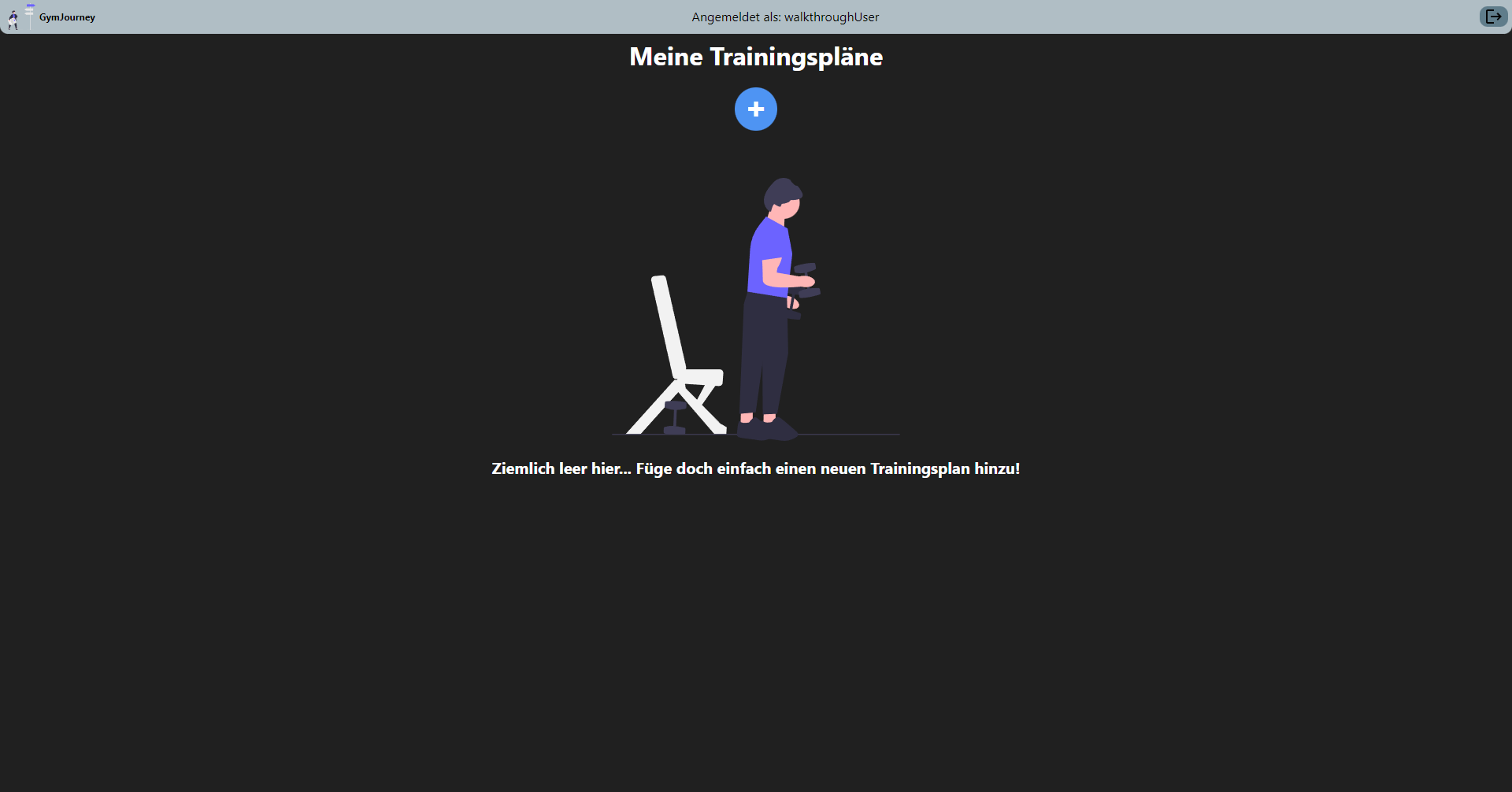
Ein Bild, das Text, Bildschirm, Dunkel enthält.

Automatisch generierte Beschreibung

**Abbildung 6:** Register-Seite

**Abbildung 5:** Login-Seite

//TODO: Sagen, dass man bei beiden auf dieser Hauptseite landet und Navigationsleiste ansprechen



**Abbildung 7:** Home-Seite

# Fazit

# Literaturverzeichnis

[1] Kinsta. *What Is React.js? A Look at the Popular JavaScript Library.* URL: <https://kinsta.com/knowledgebase/what-is-react-js/> (besucht am 26.04.2023).

[2] MDN Web Docs. *Express/Node introduction*. URL: <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Introduction> (besucht am 26.04.2023).

[3] Kinsta. *Was ist MySQL? Eine anfängerfreundliche Erklärung.* URL: <https://kinsta.com/de/wissensdatenbank/was-ist-mysql/> (besucht am 26.04.2023).

[4] Iconfinder. *Js, react js, logo icon.* URL: <https://www.iconfinder.com/icons/1174949/js_react_js_logo_react_react_native_icon> (besucht am 26.04.2023).

[5] Wikimedia Commons. *File:Expressjs.png.* URL: <https://commons.wikimedia.org/wiki/File:Expressjs.png> (besucht am 26.04.2023).

[6] Creazilla, Gil Barbara. *Mysql icon.* URL: <https://creazilla.com/nodes/3254121-mysql-icon> (besucht am 26.04.2023).

Interessante Dinge, die ich genutzt habe:

JWT-Token und Cookies zur Authorisierung.

<https://www.youtube.com/watch?v=1EuNnZEp2sQ&list=PLrNy47mmRpmPj5_W4OdjvrBbP1LCEVgSF&index=99&t=126s&ab_channel=LamaDev>

Minute 34:15

* Dadurch muss man die User-Id nicht überall herausfinden/mitgeben, sondern kann den Cookie nutzen

Ich nutze stand jetzt localStorage um den Nutzer dauerhaft zu speichern (I guess)

React Query (siehe verlinktes Video)

Bei Home nutze ich Home.css statt [Home.module.css](http://Home.module.css) da sonst das ändern der Styles des Popups nicht möglich ist

<https://react-popup.elazizi.com/css-styling>

forwardRef verwendet im Button weil es sonst in popup-js zu folgendem Fehler kam:

Check the render method of `ForwardRef`. at Button (http://localhost:3000/static/js/bundle.js:359:5) at http://localhost:3000/static/js/bundle.js:59182:27 at div at div at Home (http://localhost:3000/static/js/bundle.js:1086:68) at PrivateRoute (http://localhost:3000/static/js/bundle.js:279:5) at RenderedRoute (http://localhost:3000/static/js/bundle.js:53437:5) at Routes (http://localhost:3000/static/js/bundle.js:53927:5) at QueryClientProvider (http://localhost:3000/static/js/bundle.js:50471:21) at App at Router (http://localhost:3000/static/js/bundle.js:53865:15) at BrowserRouter (http://localhost:3000/static/js/bundle.js:52054:5) at AuthContextProvider (http://localhost:3000/static/js/bundle.js:868:5)