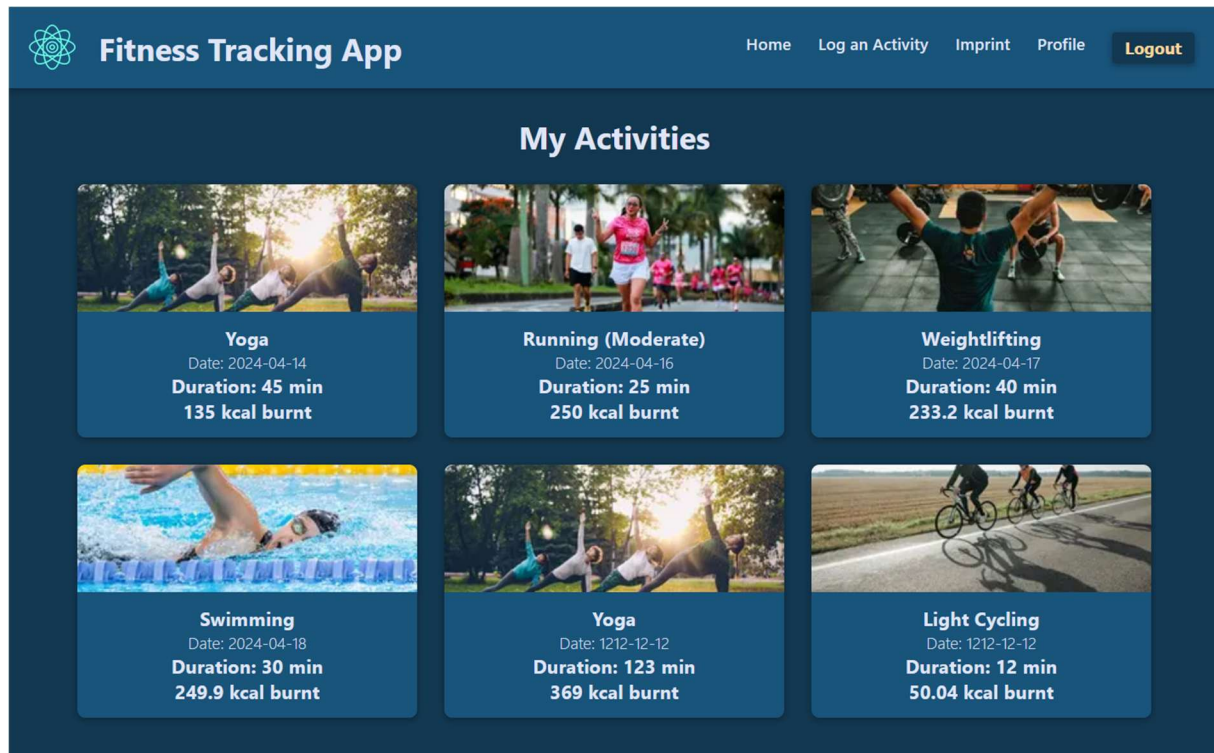




Fitness Tracking App

Projektarbeit React mit Next.js – ÜK 294 Frontend



Projektname: Fitness Tracking App
Projektmitglied 1: Lara Saurer
Projektmitglied 2: Jonas Vetsch

Firma: Die Schweizerische Post AG

Datum: 10.04.26



Inhaltsverzeichnis

Fitness Tracking App.....	1
Projektarbeit React mit Next.js – üK 294 Frontend.....	1
TEIL 1	4
Abstract.....	4
Anforderungsanalyse.....	5
Zielgruppe.....	5
Pflichtenheft.....	6
Lieferumfang.....	7
Planung	8
TEIL 2	9
Sitemap	9
Login.....	9
Home.....	9
ActivityDetailsPage.....	9
CreateActivityPage.....	9
EditActivityPage.....	9
ProfilePage	9
Impressum.....	9
Layout	10
Mockup Startseite.....	10
Farbkonzept	11
Ozean Türkis (#133851).....	11
Fitness Türkis (#19547B)	11
Fitness Gelb (#FFD89B)	11
Off White (#E0E6F8).....	11
TEIL 3	12
Technische Dokumentation	12
Testkonzept	13
Testumgebung	13
Testfälle	13
Testprotokoll.....	22
Bildnachweise Webseite	22



Fazit Lara	23
Fazit Jonas	23
Was lief gut/schlecht?	23
Wie seid ihr mit dem Endergebnis zufrieden?	23
Was habt ihr gelernt?	23
Ist alles vorhanden oder was fehlt noch?	24



TEIL 1

Abstract

Diese Arbeit beschreibt die Entwicklung unserer Webapplikation «Fitness Tracking App»: Ein intuitiver Sport-Aktivitäten-Tracker für Menschen bis 30 Jahre. Die App wird mit React, Next.js und einem JSON-Server umgesetzt und implementiert Authentifizierung mit JWT. Das Hauptziel des Projekts ist es, sportliche Aktivitäten unkompliziert zu erfassen und den damit verbundenen Kalorienverbrauch basierend auf Durchschnittswerten sofort sichtbar zu machen. Die Applikation ermöglicht es den Usern, ihre Aktivitäten in einer Übersicht zu sehen sowie neue Aktivitäten über eine vollständige CRUD-Funktionalität zu erfassen, zu bearbeiten oder zu löschen. Die Datenhaltung basiert auf einer effizienten JSON-Server-Architektur. Neben der reinen Aktivitätsverfolgung bietet ein simpler Profilbereich sowie ein optionales (falls genügend Zeit) Dashboard für Ernährung und Flüssigkeitszufuhr einen ganzheitlichen Überblick über den sportlichen Lifestyle.



Anforderungsanalyse

Zielgruppe

Die Fitness Tracking App ist für junge Menschen gedacht, die sportlich aktiv sind oder es werden möchten. Dabei beschränken wir uns bewusst nicht auf eine Nische (z.B. Schwimmen), sondern haben eine bewusst breite Auswahl von sportlichen Aktivitäten, die ein*e Nutzer*in tracken kann. Damit möchten wir eine breite Zielgruppe ansprechen.

Wir möchten Menschen im Alter von 14 bis 35 Jahren ansprechen. Diese Entscheidung wird sich in unserem Design widerspiegeln. Wir möchten das Design der App modern und minimalistisch halten.

Unsere Zielgruppe ist englischsprachig. Die Fitness Tracking App wird in ihrer ersten Version nur auf Englisch verfügbar sein.

Die Fitnesstracking App wird als Web-App (SPA) auf Desktopgeräten genutzt. Es ist für die erste Version der App noch keine Mobile-Version vorgesehen.

Menschen, die die Fitnesstracking App nutzen, möchten einen Einblick in ihr sportliches Aktivitätsniveau bekommen. Sie möchten sehen, wann sie welche Aktivitäten gemacht haben. Dies erlaubt ihnen, einen vertieften Einblick in ihr sportliches Verhalten zu bekommen und dieses besser zu reflektieren. Zudem gibt dieser Einblick auch eine bessere Grundlage für die Planung der Erholungszeiten.

Menschen, die die Fitnesstracking App nutzen, möchten auch verstehen, wie viel Energie sie für die einzelnen Aktivitäten jeweils verbraucht haben. Dies wird über die Berechnung und Anzeige der ungefähren Kalorienanzahl ermöglicht.



Pflichtenheft

Funktionale Anforderungen

ID	Beschreibung	Art
FA-01	Authentifizierung: Die App ermöglicht es Nutzern, sich mit E-Mailadresse und Passwort einzuloggen.	muss
	Authentifizierung: Die App ermöglicht es Nutzern, sich über einen Logout-Button im Header auszuloggen.	muss
FA-02	Aktivitäten erfassen: Die App ermöglicht es eingeloggten Benutzern, sportliche Aktivitäten zu erfassen mit den Attributen Aktivitätstyp, Anzahl Minuten und Datum (Tag, Monat, Jahr). Dies geschieht über eine Add/Edit-Seite, die ein Formular zur Eingabe der Attribute einer Aktivität beinhaltet.	muss
FA-03	Aktivitäten Übersicht: Die App zeigt dem eingeloggten Benutzer auf der Startseite alle seine erfassten Aktivitäten an. Dabei werden der Aktivitätstyp, die Dauer in Minuten, das Datum und die Anzahl verbrannter Kalorien angezeigt.	muss
FA-04	Aktivitäten Detailseite: Durch einen Klick auf eine Aktivität auf der Startseite gelangt der Nutzer zur Detailseite der angeklickten Aktivität. Dort werden erneut der Aktivitätstyp, die Dauer in Minuten, das Datum und die Anzahl verbrannter Kalorien angezeigt. Zudem hat der Nutzer auf der Detailseite die Möglichkeit, die Aktivität zu bearbeiten und zu löschen.	soll
FA-05	Aktivität bearbeiten: Bestehende Aktivitäten können über eine Add/Edit Seite bearbeitet werden. Dabei können alle Attribute (Typ, Dauer, Datum) angepasst werden.	soll
FA-06	Kalorienberechnung: Die App kann eine vereinfachte Kalorienberechnung machen, bei der für jeden Aktivitätstyp eine vordefinierte Anzahl Kalorien pro Minute Aktivität hinterlegt ist. Die Kalorienberechnung rechnet diese Anzahl Kalorien x Anzahl Minuten.	soll
FA-07	Profilanzeige: Eine Profilage zeigt die E-Mail-Adresse des aktuell angemeldeten Benutzers an.	soll
FA-08	Dashboard (Zusatz): Anzeige einer Übersicht über Wasser- und Essenseinnahme (nur falls Zeitbudget reicht).	kann
FA-09	Registrierung: Ermöglicht neuen Nutzern, ein Konto zu erstellen (gemäss Absprache mit Coach Mike aktuell nur als "kann" eingestuft).	kann



Nichtfunktionale Anforderungen

ID	Beschreibung	Art
NFA-01	Jede einzelne Seite des Projekts soll die HTML W3C Validation fehlerfrei durchlaufen.	muss
NFA-02	Responsive Design: Die App ist für Desktopgeräte gelayoutet. Das Layout ist responsive, damit unterschiedliche Fenstergrößen unterstützt werden.	muss
NFA-03	Benutzerfreundlichkeit: Die Navigation ist intuitiv. Um zur Erfassungsseite zu gelangen, um eine neue Aktivität zu erfassen, sollten mit maximal drei Klicks von der Startseite aus nötig sein.	soll
NFA-04	Datensicherheit: Passwörter werden im JSON-Server (simuliert) und im Frontend niemals im Klartext in der UI angezeigt.	muss
NFA-05	Fehlertoleranz: Bei fehlerhaftem Login oder wenn die Validierung eines Formularfelds fehlschlägt, erhält der User eine verständliche Fehlermeldung.	muss
NFA-06	Mobile Layout: Das Layout der Webapp ist auch für mobile Geräte optimiert.	kann

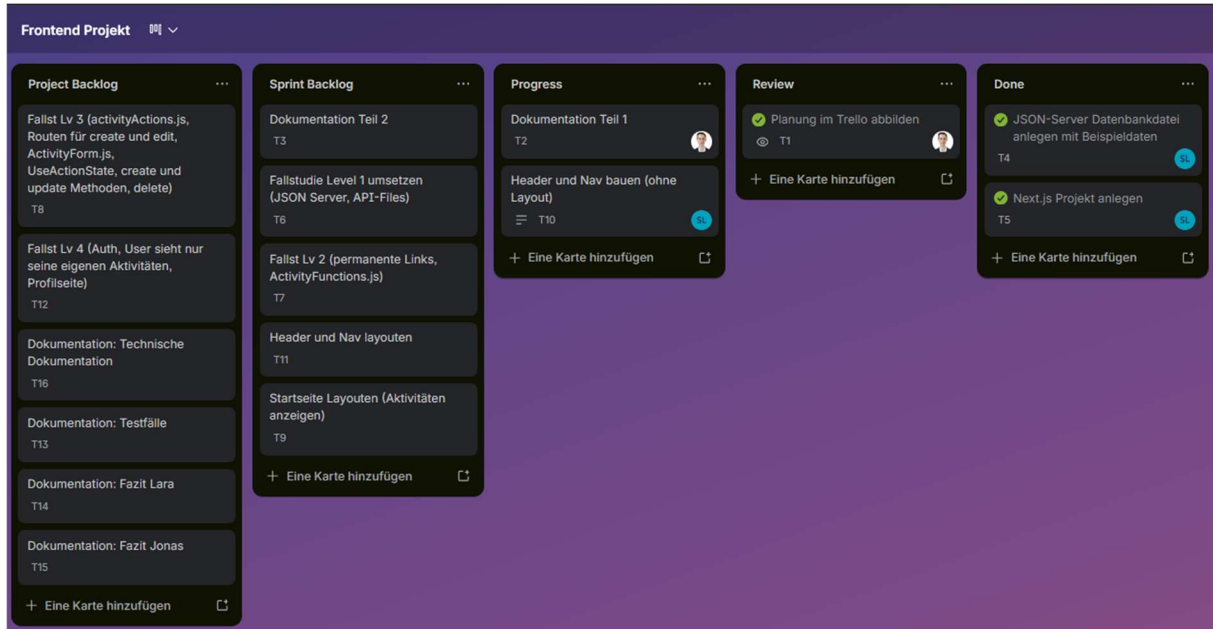
Lieferumfang

Der Lieferumfang und die Vorgaben über Speicherort werden dem mitgegebenen Lastenheft entnommen.



Planung

Die Planung haben wir in Trello umgesetzt und uns dabei an den von Scrum inspirierten Aufbau gehalten, den wir bereits vom Project Training Modul kannten: Project Backlog, Sprint Backlog, Progress, Review, Done.



Wir führen im Rahmen dieses Projekts kein echtes Scrum durch (es gibt schliesslich auch kein PO). Wir teilen das Projekt aber dennoch in 4 Sprints auf, die jeweils einen Tag dauern.



TEIL 2

Sitemap

Login

Diese Seite dient als Einstiegspunkt für die Applikation, da nicht eingeloggte User keinen Zugriff auf die Funktionen haben. Sie enthält ein Formular für E-Mail und Passwort, um sich mit einem in der Datenbank vordefinierten User-Profil zu authentifizieren.

Home

Die Home-Seite bietet dem User eine Übersicht über seine erfassten sportlichen Aktivitäten. Von der Home Seite aus gelangt man über einen Klick auf eine erfasste Aktivität auf deren Detailseite. Falls Zeit, zeigt die Home-Seite ein Dashboard mit Informationen zur Ernährung und Flüssigkeitsaufnahme an.

ActivityDetailsPage

Auf dieser Detailseite werden die spezifischen Informationen einer einzelnen Aktivität angezeigt, insbesondere der Typ, die Dauer und das Datum. Zudem berechnet die App hier basierend auf der Dauer und dem Aktivitätstyp den ungefähren Kalorienverbrauch für die gewählte Sportart. Auf dieser Seite findet sich auch der Button zum Löschen des Datensatzes.

CreateActivityPage

Diese Seite enthält ein Formular zum Erfassen neuer Aktivitäten. Der User wählt hierbei aus einer vorgegebenen Liste den Aktivitätstyp (z. B. Schwimmen oder Joggen) aus und gibt die verbrachte Zeit in Minuten sowie das Datum (Jahr, Monat, Tag) an. Die Uhrzeit wird nicht erfasst.

EditActivityPage

Hier können bestehende erfasste Aktivitäten bearbeitet werden, falls sich der User beispielsweise bei der Datumsangabe geirrt hat. Die Seite stellt die Funktionen bereit, um die Daten im JSON-Server zu aktualisieren.

ProfilePage

Die Profilstelle zeigt dem angemeldeten User seine persönlichen Kontoinformationen an, in der ersten Version der App ist dies die hinterlegte E-Mail-Adresse.

Impressum

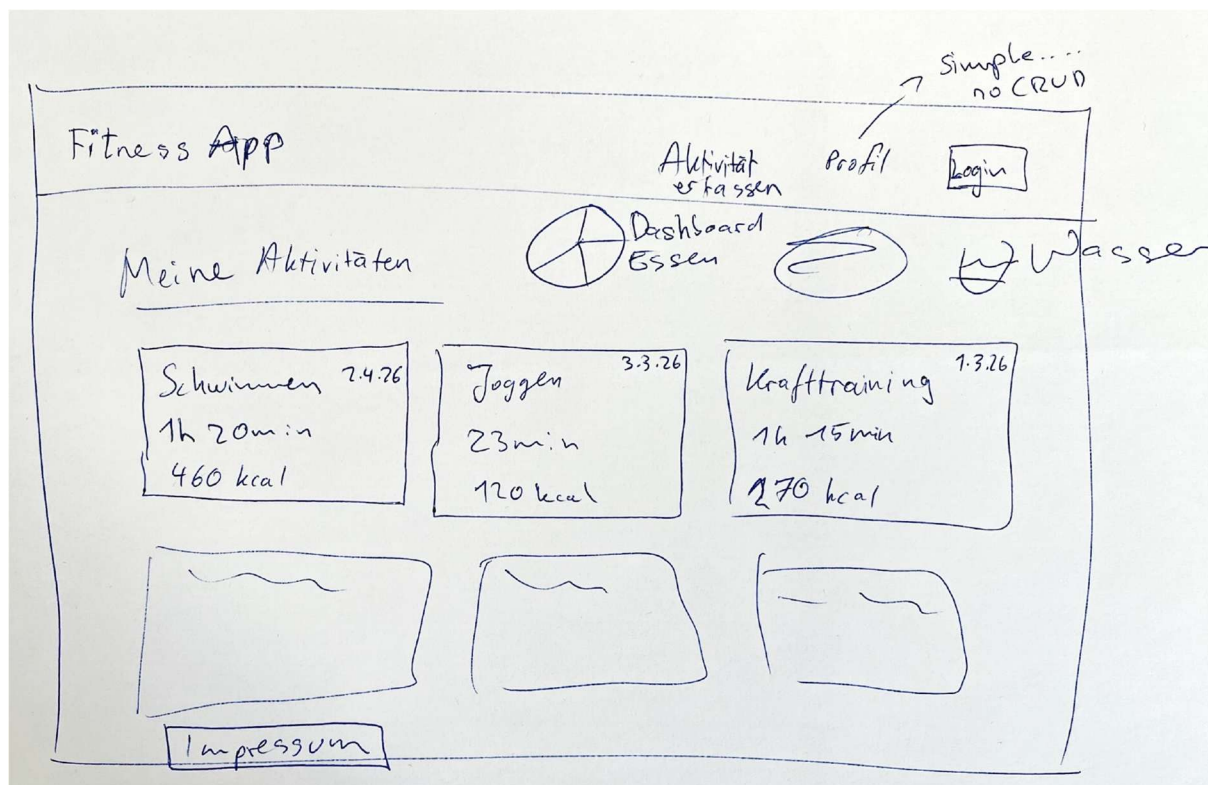
Die Impressumsseite ist klassischerweise über den Footer zu erreichen. Sie enthält die rechtlich vorgeschriebenen Informationen über die Webseite und deren Betreiber. Wir haben uns entschieden, den Link zum Impressum in den Header aufzunehmen, da das Impressum der einzige Link im Footer gewesen wäre. Durch die Integration des Impressum-Links in den Header fällt der Footer weg und es bleibt mehr Platz für die zentralen Seiteninhalte.

Diverses:

Der Button fürs Login und Logout finden sich im Header.

Layout

Mockup Startseite



Das User Interface ist in drei horizontale Bereiche unterteilt: Header, Hauptinhalt und Footer.

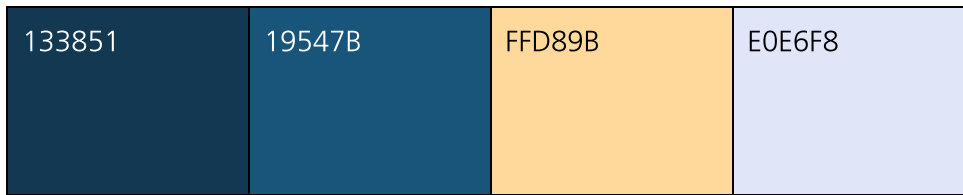
Im **Header** befindet sich auf der linken Seite das Logo bzw. der Name der App. Auf der rechten Seite sind die Navigationselemente angeordnet: «Aktivität erfassen», daneben findet sich der Zugang zum «Profil». Zudem findet sich im Header ein Login-Button (bzw. Logout, je nach Status).

Der **Hauptbereich** ist zweigeteilt. Im oberen Segment ist ein optionales Dashboard skizziert, das ein Kreisdiagramm und ein dynamisches Wasserglas (zur Anzeige der Flüssigkeitszunahme am heutigen Tag) enthält. Diese dienen der schnellen Übersicht über die Ernährung (Kreisdiagramm) sowie die Flüssigkeitsaufnahme (Wasserglas). Darunter folgt der Kernbereich der App unter dem Titel „Meine Aktivitäten“. Hier werden die erfassten Aktivitäten als Grid-Layout mit einzelnen Karten (Cards) dargestellt. Jede Karte repräsentiert eine Trainingseinheit und enthält Details: den Namen des Aktivitätstyps (z. B. Schwimmen, Joggen, Krafttraining), das Datum der Durchführung, die Dauer in Minuten und den daraus resultierenden Kalorienverbrauch.

Den unteren Abschluss der Seite bildet der **Footer**, in dem ein Link zum Impressum platziert ist.



Farbkonzept



Ozean Türkis (#133851)

Dieser tiefe Ton bildet das solide Fundament der App und vermittelt Professionalität sowie Seriosität. Er sorgt für einen modernen, aufgeräumten Look und dient als ruhiger Hintergrund für alle Seiten.

Fitness Türkis (#19547B)

Das kräftigere Türkis bringt Dynamik und Frische in das Interface und wird primär für Kartenhintergründe (z.B. Cards, Inputfelder, ...) aber auch für Buttons genutzt. Es assoziiert Vitalität und Bewegung, was den sportlichen Fokus der App ideal unterstreicht.

Fitness Gelb (#FFD89B)

Als energetische Akzentfarbe wird Fitness Gelb für Hervorhebungen wie den Kalorienverbrauch aber auch für Links eingesetzt. Es symbolisiert Optimismus und Erfolg, wodurch wir die Motivation unserer User beim Erreichen ihrer Ziele unterstützen.

Off White (#E0E6F8)

Dieses sanfte Weiss nutzen wir als Standardfarbe für Schriften, es sorgt für gute Lesbarkeit und ein sauberes Design. Durch die Wahl eines Off Whites im Gegensatz zu hartem Weiss (#FFFFFF) federn wir harte Kontraste ab und verleihen der Applikation ein modernes und leichtes Erscheinungsbild.



TEIL 3

Technische Dokumentation

Im Rahmen des Projekts kommen die folgenden Technologien zum Einsatz:

TECHNOLOGIE	VERSION
Next.js (Frontend Framework)	v16.2.2
React (JavaScript Library, UI Library)	v19.2.4
react-dom	v19.2.4
Node.js (JavaScript Laufzeitumgebung)	v24.13.0
nodemon (Vereinfacht das Arbeiten in Node.js-Umgebungen)	v3.1.14
JavaScript (ES6+) (Programmiersprache)	Aktuellster Standard (April 2026), genaue Version ist Browserabhängig
CSS Modules (Cascading Style Sheets)	Standard (CSS wird nativ unterstützt)
JSON-Server (Backend/API Mocking)	v0.17.4
CORS (Express Middleware, erlaubt Datenverkehr zwischen Backend und Frontend (CORS))	v2.8.6 (Standard für JSON-Server Setups)
ESLint (Linter/Code-Qualität)	Version 9
ESLint-Config-Next	v16.2.2
NPM (Package Manager)	v11.12.0
Git (Versionsverwaltung)	v2.53.0.windows.2
GitHub (Remote Repository zur Zusammenarbeit an Code)	Aktuellster Standard im April 2026
VS Code (IDE, Code Editor)	v1.115.0



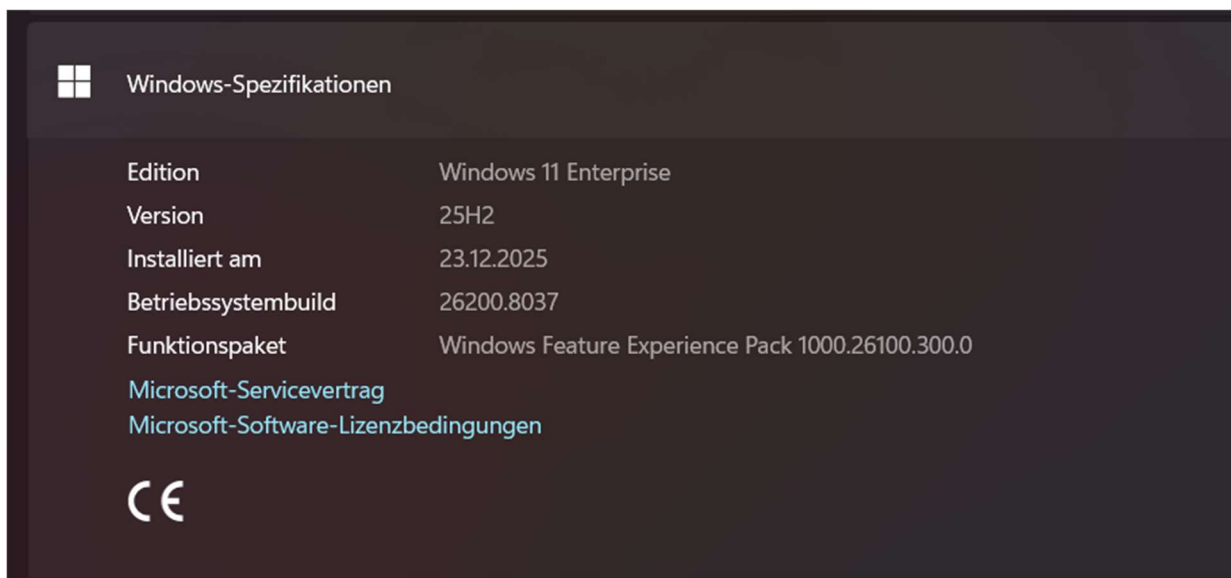
Testkonzept

Testumgebung

Für die Durchführung der Tests haben wir eine standardisierte und moderne Infrastruktur eingesetzt. Die Testumgebung basiert auf dem Betriebssystem Windows 11 Enterprise in der Version 25H2.

Um gute Kompatibilität und ein gutes Nutzererlebnis sicherzustellen, finden die Tests in den jeweils aktuellen Versionen der Browser Mozilla Firefox und Microsoft Edge statt. Damit wird gewährleistet, dass die Applikation unter realistischen Bedingungen auf einem aktuellen geschäftlichen Standard-Client verifiziert wird.

Software	Version / Build	Release-Datum
Mozilla Firefox	149.0.2	07. April 2026
Microsoft Edge	146.0.3856.109	06. April 2026



Testfälle

Abschnitt	Inhalt
ID	Testfall 1
Anforderungen	FA-03 (Aktivitäten Übersicht)
Vorbedingungen	<ul style="list-style-type: none">- Die lokale Entwicklungsumgebung (Frontend und JSON-Server) läuft fehlerfrei.- In der Datenbank (db.json) existiert mindestens ein Benutzer, für den bereits Aktivitäten erfasst wurden.



- Dieser Benutzer ist aktuell in der Applikation eingeloggt.

Ablauf

1. Die Startseite (/) der Applikation wird im Browser aufgerufen.

2. Der Inhaltsbereich unter der Überschrift "Meine Aktivitäten" wird betrachtet.

Erwartetes Resultat

- Die in der Datenbank vorhandenen Aktivitäten des eingeloggten Benutzers werden erfolgreich geladen und als Karten (Cards) im Grid-Layout dargestellt.

- Auf jeder Karte werden die Attribute "Aktivitätstyp", "Datum", "Dauer in Minuten" sowie die "verbrannten Kalorien" korrekt und vollständig angezeigt.



Abschnitt	Inhalt
ID	Testfall 2
Anforderungen	FA-03 (Die App zeigt dem eingeloggten Benutzer auf der Startseite alle <i>seine</i> erfassten Aktivitäten an)
Vorbedingungen	<ul style="list-style-type: none">- Die lokale Entwicklungsumgebung (Frontend und JSON-Server) läuft.- In der Datenbank (db.json) existieren mindestens zwei Benutzer (z.B. User A und User B).- Für <i>beide</i> Benutzer sind jeweils Aktivitäten in der Datenbank erfasst.- User A ist aktuell in der Applikation eingeloggt.
Ablauf	<ol style="list-style-type: none">1. Die Startseite (/) der Applikation wird im Browser aufgerufen.2. Die angezeigten Aktivitäten werden notiert.3. Die angezeigten Aktivitäten werden manuell mit den Einträgen in der db.json Datei verglichen.
Erwartetes Resultat	<ul style="list-style-type: none">- Es werden auf der Startseite ausschliesslich Aktivitäten angezeigt, die mit der userId von User A verknüpft sind.- Aktivitäten, die User B gehören, werden weder angezeigt noch im Hintergrund (im DOM oder in der Network-Response für die UI) an den Client von User A geleakt.



Abschnitt	Inhalt
ID	Testfall 3
Anforderungen	FA-01 (Authentifizierung), FA-07 (Profilanzeige)
Vorbedingungen	<ul style="list-style-type: none">- Die lokale Entwicklungsumgebung (Frontend und JSON-Server) läuft.- In der Datenbank (db.json) existiert ein gültiger Benutzer (z. B. admin@admin.ch) mit einem Passwort.- Der Benutzer ist aktuell nicht eingeloggt.
Ablauf	<ol style="list-style-type: none">1. Die Startseite wird aufgerufen und im Header auf den Button "Login" geklickt.2. Im Formular werden die korrekte E-Mail-Adresse und das Passwort eingegeben.3. Der "Log In"-Button wird geklickt.4. Nach erfolgreichem Login wird im Header auf den Link "Profile" geklickt.5. Anschliessend wird im Header auf den Button "Logout" geklickt.
Erwartetes Resultat	<ul style="list-style-type: none">- Nach Schritt 3 wird der Benutzer eingeloggt und weitergeleitet.- Nach Schritt 4 wird auf der Profilseite die E-Mail-Adresse des aktuell angemeldeten Benutzers korrekt angezeigt.- Nach Schritt 5 ist der Benutzer vollständig ausgeloggt (die Session ist beendet) und er wird auf die Startseite (/) weitergeleitet. Der "Profile"-Link ist nun nicht mehr sichtbar.



Abschnitt	Inhalt
ID	Testfall 4
Anforderungen	FA-02 (Aktivitäten erfassen), FA-06 (Kalorienberechnung)
Vorbedingungen	<ul style="list-style-type: none">- Die App läuft und der JSON-Server ist gestartet.- Ein Benutzer ist erfolgreich eingeloggt.- Die Faktoren für die Kalorienberechnung pro Aktivitätstyp sind im Code oder Backend definiert.
Ablauf	<ol style="list-style-type: none">1. Im Navigationsmenü wird auf "Aktivität eintragen" geklickt.2. Im Formular wird ein spezifischer Aktivitätstyp ausgewählt (z.B. "Joggen").3. Bei "Dauer" wird ein spezifischer Wert eingegeben (z.B. "60" Minuten).4. Ein gültiges Datum wird ausgewählt und auf "Save" geklickt.5. Auf der Startseite (oder der Detailseite) wird die neu erstellte Aktivität überprüft.
Erwartetes Resultat	<ul style="list-style-type: none">- Das Formular wird fehlerfrei übermittelt und der Benutzer wird auf die Startseite weitergeleitet.- Die neue Aktivität erscheint mit dem korrekten Typ, Datum und der eingegebenen Dauer in der Übersicht.- Wichtig: Der angezeigte Kalorienverbrauch ist nicht statisch, sondern entspricht exakt dem berechneten Wert (Dauer in Minuten × vordefinierter Kalorienwert für den Typ "Joggen").



Abschnitt	Inhalt
ID	Testfall 5
Anforderungen	FA-05 (Aktivität bearbeiten), FA-04 (Aktivitäten Detailseite)
Vorbedingungen	<ul style="list-style-type: none">- Die lokale Entwicklungsumgebung (Frontend und JSON-Server) läuft.- Ein Benutzer ist erfolgreich eingeloggt.- In der Datenbank (db.json) existiert mindestens eine von diesem Benutzer erfasste Aktivität.
Ablauf	<ol style="list-style-type: none">1. Auf der Startseite (/) wird auf eine bestehende Aktivitäts-Karte geklickt, um die Detailseite zu öffnen.2. Auf der Detailseite wird auf den Link "Edit" geklickt.3. Im nun geöffneten Formular (/activities/edit/[id]) wird mindestens ein Wert geändert (z. B. die Dauer in Minuten).4. Der Button "Save" wird geklickt.
Erwartetes Resultat	<ul style="list-style-type: none">- Nach dem Klick auf "Save" wird das Formular erfolgreich übermittelt und der Benutzer wird auf die Startseite zurückgeleitet.- Die geänderten Werte (z.B. die neue Dauer) werden auf der entsprechenden Aktivitäts-Karte auf der Startseite sofort korrekt angezeigt.



Abschnitt	Inhalt
ID	Testfall 6
Anforderungen	FA-04 (Aktivitäten Detailseite - Löschen)
Vorbedingungen	<ul style="list-style-type: none">- Die lokale Entwicklungsumgebung (Frontend und JSON-Server) läuft.- Ein Benutzer ist erfolgreich eingeloggt.- Der Benutzer hat mindestens eine eigene Aktivität, die er löschen kann.- (Optional: Man notiert sich im Vorfeld die ID der Aktivität in der db.json, um den Löschvorgang auch im Backend zu verifizieren).
Ablauf	<ol style="list-style-type: none">1. Auf der Startseite (/) wird auf eine zu löschende Aktivität geklickt, um deren Detailseite zu öffnen.2. Auf der Detailseite wird auf den Button "Delete" geklickt.3. Es erscheint ein Bestätigungsdialog ("Möchtest du diese Aktivität wirklich löschen?").4. Im Dialog wird auf "OK" (bzw. Bestätigen) geklickt.
Erwartetes Resultat	<ul style="list-style-type: none">- Die Aktivität wird erfolgreich aus dem System gelöscht und der Benutzer wird auf die Startseite weitergeleitet.- Auf der Startseite ist die gelöschte Aktivität nicht mehr sichtbar.



Abschnitt	Inhalt
ID	Testfall 7
Anforderungen	FA-01 (Authentifizierung)
Vorbedingungen	<ul style="list-style-type: none">- Die lokale Entwicklungsumgebung läuft.- In der Datenbank existieren User A und User B.- User B hat eine Aktivität erfasst (ID ist für den Test bekannt, z.B. /activities/edit/5).- Zu Beginn des Tests ist kein Benutzer eingeloggt.
Ablauf	<p>Teil 1 (Nicht eingeloggt):</p> <ol style="list-style-type: none">1. Der Pfad /activities/create wird manuell in die Adresszeile des Browsers eingegeben und aufgerufen. <p>Teil 2 (Eingeloggt, fremde Daten):</p> <ol style="list-style-type: none">2. User A loggt sich regulär ein.3. User A gibt manuell die URL zur Bearbeitung der fremden Aktivität von User B ein (z.B. /activities/edit/5) und ruft diese auf.
Erwartetes Resultat	<ul style="list-style-type: none">- Zu Teil 1: Der unauthentifizierte Zugriff wird blockiert. Der User wird stattdessen auf die Login-Seite (/login) umgeleitet.- Zu Teil 2: Der Zugriff auf die fremde Ressource wird verweigert. Die Applikation fängt den Versuch ab (z.B. durch einen Error oder eine Umleitung auf die Startseite), und die fremden Daten werden auf keinen Fall im Formular angezeigt.



Abschnitt	Inhalt
ID	Testfall 8
Anforderungen	NFA-05 (Fehlertoleranz)
Vorbedingungen	- Die lokale Entwicklungsumgebung läuft. - Ein Benutzer ist erfolgreich eingeloggt.
Ablauf	1. Im Navigationsmenü wird auf "Aktivität eintragen" geklickt. 2. Im Formular wird ein Aktivitätstyp ausgewählt und eine Dauer eingegeben. 3. Das Feld für das Datum wird absichtlich leer gelassen. 4. Der Button "Save" wird geklickt.
Erwartetes Resultat	- Das Formular wird nicht an den JSON-Server übermittelt (der Request wird idealerweise schon im Frontend gestoppt). - Der Benutzer erhält eine verständliche Fehlermeldung (z.B. "Bitte wählen Sie ein Datum aus"), die ihn auf die fehlende Eingabe hinweist. - Die Applikation stürzt nicht ab.

Abschnitt	Inhalt
ID	Testfall 9
Anforderungen	NFA-02 (Aktivitäten erfassen - Validierung)
Vorbedingungen	- Die App läuft und ein Benutzer ist eingeloggt.
Ablauf	1. Navigiere zu "Track a new Activity". 2. Wähle einen Aktivitätstyp und eine gültige Dauer. 3. Wähle im Datum-Feld ein Datum aus, das in der Zukunft liegt. 4. Klicke auf "Save Activity".
Erwartetes Resultat	Das Formular wird abgelehnt. Eine Fehlermeldung ("Das Datum darf nicht in der Zukunft liegen.") wird angezeigt.



Testprotokoll

Testfall Nr	Erfolgreich?	Bemerkung / Erklärung wenn nicht erfolgreich
Testfall 1	Ja	Es werden, wie erwartet, alle Attribute angezeigt.
Testfall 2	Ja	Es werden, wie erwartet, nur die Aktivitäten des aktuell eingeloggten Users angezeigt.
Testfall 3	Ja	Der Benutzer wurde eingeloggt, die E-Mail ist auf der Profilseite sichtbar. Der Benutzer wurde wieder ausgeloggt.
Testfall 4	Ja	Die Aktivität wurde erfasst, der Kalorienverbrauch wurde korrekt berechnet und in die Datenbank gespeichert.
Testfall 5	Ja	Die Startseite zeigt die geänderte Aktivität an, die Attribute wurden aktualisiert.
Testfall 6	Ja	Die Aktivität wurde gelöscht und das Rerouting hat geklappt.
Testfall 7	Ja	Teil 1: Das Rerouting auf /login hat geklappt. Teil 2: Es fand eine Umleitung auf die Error.js statt.
Testfall 8	Ja	Das Formular wurde nicht ans Backend geschickt. Stattdessen erschien eine klare Fehlermeldung.
Testfall 9	Ja	Auch hier: Klare Fehlermeldung, dass das Datum nicht in der Zukunft liegen darf.

Bildnachweise Webseite

Für unsere App nutzen wir als Symbolbilder für die Aktivitätstypen folgende Ressourcen:

Cycling: Photo by Tofros.com: <https://www.pexels.com/de-de/foto/strasse-menschen-fahrrader-draussen-5807581/>

Running: Photo by Adailton Batista: <https://www.pexels.com/de-de/foto/35261915/>

Swimming: Photo by Jim De Ramos: <https://www.pexels.com/de-de/foto/wettkampfschwimmer-in-aktion-30550684/>

Yoga: Photo by Jari Lobo: <https://www.pexels.com/de-de/foto/yoga-kurs-im-freien-in-sonniger-parklage-36715220/>

Weightlifting: Photo by Victor Freitas: <https://www.pexels.com/de-de/foto/person-die-langhantel-drinnen-hebt-2261485/>

Walking: Photo by Jari Lobo: <https://www.pexels.com/de-de/foto/stadt-mann-frau-strasse-15659357/>

Diese sind auch im Impressum so aufgeführt.



Fazit Lara

Mit dem Projekt bin ich im Gesamten sehr zufrieden, in diesem Projekt habe ich sehr viel gelernt und konnte nicht nur sehr viel Fachliches dazu lernen, sondern auch im Allgemeinen zu Lernstrategien und allgemeines Lernen und Verständnis konnte ich sehr viel lernen. Was sehr gut lief war die Zusammenarbeit mit Jonas und auch die Unterstützung auf Hinsicht des Fachgesprächs. Jonas hat mir dabei immer wieder viel erklärt und hat mir auch sehr gut geholfen damit ich ideal vorbereitet war fürs Fachgespräch. In diesem Sinn ein riesiges DANKE an Jonas, dass du mir geholfen und mich so unterstützt hast. Was zudem auch gut lief war das Projekt im Praktischen. Ich finde das Projekt ist gut gekommen. Mit dem Endergebnis bin ich persönlich sehr zufrieden und es ist in diesem Stand alles Vorhanden. Ich nehme von diesem Projekt sicher viel Praktisches mit und möchte es auch in einem weiteren Privaten Projekt umsetzen und es auch noch vertiefen. Dieses Thema ist sehr spannend und es hat mir wirklich sehr spass gemacht und im Allgemeinen habe ich wirklich viel gelernt. Nicht nur fachlich, sondern auch methodisch :)

Fazit Jonas

Was lief gut/schlecht?

Die Zusammenarbeit war angenehm und erfolgreich: Die Arbeitsphasen waren immer sehr konzentriert, die Methode (Scrum, Kanban mit Pull-Request-Workflow) hat gut funktioniert und hat dem Endprodukt gedient.

Wir hatten gegen Ende des Projekts uns nochmal auf die Fachgesprächvorbereitung konzentriert, entsprechend mussten wir nach dem Fachgespräch noch einige letzte «Offene Enden» schliessen im Code und in der Dokumentation. Doch diesen Kompromiss gingen wir bewusst ein, um so gut vorbereitet wie möglich in das Fachgespräch zu gehen.

Wie seid ihr mit dem Endergebnis zufrieden?

Ich bin sehr zufrieden mit dem Gesamtergebnis: Wir konnten alle muss-Kriterien erreichen und hatten sogar noch Zeit, uns grob um das Styling zu kümmern. Das Styling ist nicht perfekt, es könnte noch viel verbessert werden. Doch ein Grund-Styling ist da: Die App sieht ansprechend aus und die Oberfläche ist benutzerfreundlich

Was habt ihr gelernt?

Ich persönlich habe nochmal meine React-Kenntnisse vertiefen können: Ich habe gelernt, genauer zu unterscheiden, was clientseitig und was serverseitig ausgeführt wird und wie das mit der Hydrierung genau funktioniert.

Zudem haben wir uns bewusst entschieden, ein Zweierteam zu bilden wo auch Wissenstransfer während dem Projekt stattfinden kann: Ich habe mich also nicht nur in der Rolle eines Projektmitarbeiters, sondern auch eines React-Coaches wiedergefunden. Durch diese zusätzliche Aufgabe, die ich gerne wahrgenommen habe, wurde ich zusätzlich herausgefordert. Neben fachlichen Inhalten musste ich mir auch viel Gedanken über Lernmethoden, Scrum und Recherchetechniken machen, um auch hier drin eine Unterstützung zu bieten.



Ist alles vorhanden oder was fehlt noch?

Wir konnten alle muss-Kriterien des Anforderungskatalog erfüllen. In einem weiteren Projekt könnten wir die Registrierung umsetzen und die Dashboardelemente (Flüssigkeitszunahme und Essenseinnahme) umsetzen. Zudem könnte auch das Styling nochmal verbessert werden.