004-Login-Formular

Frontend Aufgabe

Anforderungsniveau: Mittel

1 Ziel der Aufgabe

Ziel dieser Aufgabenstellung ist es, Ihre Kenntnisse in moderner Webentwicklung mit React und Next.js zu prüfen. Sie implementieren eine Authentifizierungslösung unter Verwendung von React Hook Form und NextAuth.js mit einem Custom Credentials Provider. Optional sind das Dockerisieren des Projekts sowie die Trennung in dedizierte Server- und Client-Komponenten.

2 Aufgabenbeschreibung

Erstellen Sie eine Next.js-Applikation, die folgende Funktionalitäten bietet:

1. **Login-Formular:**
   - Implementieren Sie ein Login-Formular mit React Hook Form.
   - Validieren Sie Benutzername und Passwort (z. B. Pflichtfelder, Mindestlänge).
2. **Authentifizierung:**
   - Konfigurieren Sie NextAuth.js in `/pages/api/auth/[...nextauth].ts` bzw. `/app/api/auth/[...nextauth]/route.ts`.
   - Verwenden Sie einen Custom Credentials Provider in der Provider-Liste.
   - Implementieren Sie den `authorize()`-Callback so, dass die Anmeldedaten serverseitig über eine interne API überprüft werden.
3. **Geschützte Seite:**
   - Erstellen Sie eine Beispielseite (z. B. `/dashboard`), die nur angemeldeten Nutzern zugänglich ist.
   - Verwenden Sie `getServerSideProps` (oder Next.js App Router Server Components) zur Session-Validierung.

3 Anforderungen

- **Code-Organisation:** Klare Ordnerstruktur, modulare Komponenten.
- **Typensicherheit:** TypeScript für alle Dateien.
- **Styling:** Minimalistisch (z. B. Tailwind CSS oder CSS Modules).
- **Dokumentation:** Kurze README mit Installations- und Startanleitung.
- **Tests (optional):** Unit- oder Integrationstests mit Jest oder React Testing Library.

4 Optionale Erweiterungen

1. **Dockerisierung:**
   - Erstellen Sie ein `Dockerfile` sowie ggf. `docker-compose.yml`, damit Frontend und serverseitige API in Containern laufen.
2. **Trennung in Server- und Client-Komponenten:**
   - Nutzen Sie Next.js App Router und deklarieren Sie klare use client und use server Dateien.
   - Servieren Sie API-Logik und Datenzugriff in Server Components bzw. API-Routen.

5 Zusätzliche Anforderungen für Experten

- Die Backend-API darf niemals direkt vom Client (Browser) aus abgefragt werden. Verwenden Sie ausschließlich serverseitige Funktionen bzw. NextAuth-Credentials-Callbacks, um alle API-Aufrufe zu tätigen.
- Denken Sie an Sicherheitsaspekte: Schutz vor Brute-Force, sichere Passwort-Hashing-Verfahren (z. B. bcrypt), Zero-Trust-Ansatz.

6 Abgabe und Bewertung

- Stellen Sie Ihr Projekt auf GitHub bereit und senden Sie uns den Link.
- Beurteilt werden:

- Funktionalität: Erfüllt die Anwendung alle Kernanforderungen?
- Codequalität: Lesbarkeit, Struktur, Best Practices.
- Sicherheit: Einhaltung der Vorgabe, keine direkten Client-API-Aufrufe.
- Dokumentation: Verständliche README, ggf. Kommentare im Code.
- Optionale Features: Docker, Trennung der Komponenten.

Viel Erfolg bei der Umsetzung!
Stand: 31. Mai 2025