

GIBZ FIEŠTA

Projekt Erläuterung

Inhaltlich geht es in der Projektarbeit des Moduls 326 um die Erstellung einer Software, welche von den Lehrpersonen des GIBZ zur langfristigen Evaluation ihrer digitalen, unterrichtsbezogenen Anwendungskompetenzen genutzt werden kann: Aus einem Set definierter Kompetenzen sollen Lehrpersonen per Selbstdeklaration die individuelle Erreichung unterschiedlicher Kompetenzen erfassen und planen. Zur Unterstützung der Lehrpersonen können für die Kompetenzen unterschiedliche Ressourcen bereitgestellt werden. Wir haben uns hier für ein Webtool entschieden. Die Anforderungen an dieses Tool sind unten aufgelistet.

Anforderungen

Funktional

- Rollensystem (Admin, Guest, User)
 - Berechtigungen "User":
 - * Kompetenzen einsehen und als "kann ich" / "kann ich nicht" markieren
 - Berechtigungen "Admin":
 - * Kompetenzen löschen
 - * Kompetenzen erstellen
 - * Ressourcen für Kompetenzen zur Verfügung stellen
 - * User registrieren, löschen, bearbeiten, Account deaktivieren -> CRUD
 - Berechtigungen "Guest":
 - * Login nicht möglich d.h. Applikation nicht benutzbar
 - * kann Lehrer - Profile einsehen (abgeschlossene Kompetenzen & unsensitive Details (Name, Email, Geburtsjahr / Alter))
- Filterung der Kompetenzen

Nicht Funktional

Vom PDF

- Wird mit Rust & Svelte (JS) gemacht
- Testing
 - Testkonzept & Testfälle dokumentieren
 - Unit Tests decken >60% ab (API Tests)

Eigene

Muss

- Dark-Mode Switch

- Responsive UI Design / Layout
- Auth für API - Requests
- Passwörter hashed

Kann

- Blockiert den Login für 3min für den User, wenn das Passwort 3x falsch eingegeben wurde.
- Erweiterte Logfunktionen:
 - Login / Logout wird Dokumentiert -> Sichtbar für Admin
 - Login - IP's werden geloggt

Technologien

Da unser Plan eine Web-Applikation zu entwickeln ist brauchen wir passende Programmiersprachen und Tools, um dieses Vorhaben umzusetzen.

Fürs Frontend haben wir uns für **Svelte** Entschieden. Es ist einfach zu verstehen und vielseitig einsetzbar.

Beim Backend fiel unsere Entscheidung auf **Rust**, da es eine moderne, effiziente bzgl. der Ressourcen und beliebte Sprache ist, die für Backend - Services viel verwendet wird.

Natürlich müssen die Daten auch irgendwo gespeichert werden. Unsere Wahl fällt auf **MongoDB**, da wir mit diesem Tool schon vertraut sind und es Erfahrungsgemäss einfach zu verwenden ist. Obwohl die Applikation sich eigentlich für eine relationale Datenbank eignen würde, haben wir uns trotzdem dagegen entschieden, weil:

- Schutz vor SQL - Injections (Mongo - Injections gibts auch, aber viel seltener)
- Responses von Queries sind direkt in JSON - Format (keine Umwandlung nötig)
- Integrierte Library für div. Sprachen zur einfachen Verwendung von Queries
- Skalierbarer als MySQL (Ressourcenfreundlicher)
- Bessere Übersicht für Admins
- Schnellere Responses bei simplen Queries

Erfahrung

Ramon hat schon kleine Svelte-Kenntnisse, Rust hingegen ist für alle neu. MongoDB allerdings haben wir alle schon genutzt und verstehen wir zum grossen Teil auch recht gut.

Tools

Um die Seite zu entwickeln werden wir verschiedene Tools verwenden. Zum einen wäre da **VSCode**, ein gratis Open-Source Texteditor. Zusätzlich werden wir **Datagrip** und **Webstorm** der JetBrains Toolbox verwenden, um allfällige Probleme mit **VSCode** zu kompensieren.