Aufgabe 2 – Interaktiver Lebenslauf mit Vue.js, TypeScript und APIs

Einführung:

Im ersten Labor haben Sie einen interaktiven Lebenslauf mit reinem HTML5, CSS und JavaScript erstellt. Dabei ging es darum, das Fundament des Web-Frontends zu verstehen: Struktur, Gestaltung und Interaktivität durch eigene DOM-Manipulationen.

Im zweiten Labor entwickeln Sie nun einen neuen, vollständigen und erweiterten Lebenslauf, diesmal unter Einsatz moderner Webtechnologien aus dem Frontend-Umfeld:

- **Vue.js** als Framework zur komponentenbasierten Entwicklung
- **TypeScript** für typsichere, strukturierte JavaScript-Programmierung
- **Mindestens eine externe API**, um dynamische Inhalte in den Lebenslauf zu integrieren.

Wichtige Hinweise für die Bearbeitung der Aufgabe:

- Verwenden Sie ausschließlich folgende Technologien: Vue.js (v3+) und Typescript, HTML5, CSS (oder SCSS).
- Es geht nicht darum, möglichst schnell einen fertigen Lebenslaufprototypen zu haben, sondern den Zusammenhang der Bausteine und das Zusammenspiel der Frameworkkonzepte zu verstehen
- Vermeiden Sie es, Lösungen direkt durch KI-Tools generieren zu lassen. Nutzen Sie KI höchstens um Fachbegriffe oder Syntaxdetails nachzuschlagen.
- Bitte reichen Sie das Projekt „Lebenslauf 2“ **als ZIP-Datei** (ohne Abhängigkeiten) über Moodle ein. Stellen Sie dabei sicher, dass externe Abhängigkeiten und Dependency Ordner NICHT Teil Ihrer Einreichung sind. Nicht fristgerechte Einreichungen oder solche die ein anderes Format beinhalten (z.B.: einzelne Dateien), können nicht berücksichtigt werden.
- **Achten Sie bitte darauf**: Nach entpacken Ihrer Abgabe müssen die Befehle `npm install`, `npm run build` und/oder `npm start` fehlerfrei durchlaufen.

Teilaufgabe 1: Komponentenbasierte Strukturierung mit Vue.js und TypeScript

Ziel: Entwickeln Sie die Grundstruktur eines modernen webbasierten Lebenslaufs, indem Sie Inhalte sinnvoll in Vue-Komponenten aufteilen und ein konsistentes Layout erstellen. Sie können hierfür Ihren Lebenslauf aus Laboraufgabe 1 nutzen, oder einen komplett neuen Lebenslauf erstellen.

Hinweise: Ziehen Sie zur Umsetzung dieser Teilaufgabe folgende Leitfragen zu Hilfe:

- Welche Inhalte sollen in einem neuen, modernen CV enthalten sein? (z. B. persönliche Daten, Skills, Projekte, Ausbildung, Social-Media-Profile …)
- Wie können Sie diese Inhalte sinnvoll in Komponenten zerlegen? (z. B. „ProfileHeader.vue“, „ExperienceList.vue“, „SkillBadge.vue“)

- Wie organisieren Sie Ihr Projekt in einer übersichtlichen Ordnerstruktur?
- Welche Vorteile bietet Ihnen TypeScript für den Aufbau Ihrer Komponenten?
- Wie gestalten Sie das Layout, sodass die Komponenten trotz unterschiedlicher Inhalte ein konsistentes Erscheinungsbild haben?
- Wie nutzen Sie **Props** und **Emits** oder ähnliches, um Daten zwischen Komponenten zu strukturieren?

Teilaufgabe 2: Interaktivität & API-Anbindung

Ziel: Machen Sie den Lebenslauf interaktiv, sodass Nutzer:innen mit den Inhalten des Lebenslaufs interagieren können. Binden Sie hierfür eine geeignete externe Programmierschnittstelle (API) an, mit der Sie Ihre Webseite um externe Inhalte anreichern können.

Hinweise & Leitfragen: Ziehen Sie zur Umsetzung dieser Teilaufgabe folgende Leitfragen zu Hilfe:

- **Auswahl der Programmierschnittstelle**:
  - Welche externe API könnte Ihren Lebenslauf sinnvoll erweitern? Schauen Sie sich doch einmal folgende Webseiten als Inspiration an:
    - https://bund.dev/apis
    - https://github.com/public-apis/public-apis
  - Welche Art von Daten passen zu Ihrem Lebenslauf, z.B.: Bücher-API für eine Liste Ihrer Lieblingsbücher, Regierungsschnittstellen für regionale Daten, etc.?
- **Integration der Programmierschnittstelle in Ihren Lebenslauf**:
  - Wie können Sie Daten in Vue.js abrufen (z.B.: `fetch()`, `axios`, `async`/`await`)?
  - Wo sollten API-Aufrufe stattfinden (z.B.: in einer Lifecycle-Methode)?
- **Zustand & Darstellung**:
  - Wie speichern Sie API-Daten im reaktiven Vue-Zustand?
  - Wie gehen Sie mit Ladezuständen oder Fehlerfällen um?
  - Wie passen Sie das UI an, wenn Daten sich ändern?
- **Interaktivität & Nutzerführung**:
  - Welche Bereiche des Lebenslauf sollen interaktiv sein (z.B.: aufklappbare Sektionen, Filter für Skills, Dark-Mode, Tab-Navigation, etc.)?
  - Wie nutzen Sie Vue-Konzepte wie …
    - Computed Properties
    - Watchers
    - Dynamische Klassen
    - Übergänge / Animationen (`<transition>`)

Erwartete Lernziele:

Nach Abschluss dieser Laborübung sollten Sie:

- den grundlegenden Aufbau eines Vue-Projekts verstehen
- selbstständig Vue-Komponenten erstellen und strukturieren können
- TypeScript in einer Webanwendung sicher verwenden können
- externe APIs einbinden und deren Daten im UI darstellen
- moderne Frontend-Interaktivität umsetzen
- das Zusammenspiel aus Framework, Typsystem und Webarchitektur verstehen

Reflexionsfragen:

Folgende Reflexionsfragen dienen Ihnen als Anhaltspunkt der Lerninhalte:

- Wie haben Sie Ihren Lebenslauf in Komponenten zerlegt und warum?
- Welche externe API haben Sie verwendet und wie haben Sie deren Daten in Ihr Design integriert?
- Welche Architektur- oder Strukturierungsentscheidungen haben sich im Nachhinein als besonders sinnvoll erwiesen?
- Gab es Stellen, an denen Vue.js oder TypeScript Ihnen geholfen haben, Probleme eleganter zu lösen als in Labor 1?
- Welche Interaktivität steigert Ihrer Meinung nach die Benutzerfreundlichkeit Ihres Lebenslaufs am meisten – und warum?
- Welche Herausforderungen sind beim Arbeiten mit APIs aufgetreten und wie würden Sie diese in Zukunft adressieren?