

Tech Stack

Gegenüberstellung

Frontend

Kategorie	HTML	XML	JSX
Bekanntheit	Standard und weit verbreitet.	Eher speziell, nicht so verbreitet.	Spezifisch für React, könnte eine Lernkurve bedeuten.
Leichtgewichtigkeit	Einfache Syntax und Struktur.	Schwere Syntax im Vergleich zu HTML.	Spezifisch für React, könnte eine Lernkurve bedeuten.
Aufwand	-	Lernkurve für Entwickler, die mit HTML vertraut sind.	Lernkurve für Entwickler, die nicht mit JSX vertraut sind.

Kategorie	CSS	SASS/LESS
Leichtgewichtigkeit	Einfache Syntax und Struktur.	Bietet erweiterte Funktionen, aber Lernkurve.
Aufwand	-	Lernkurve für Entwickler, die nicht mit Präprozessoren vertraut sind.

Kategorie	JS	TypeScript	ES6
Bekanntheit	Grundlegende Sprache für Webentwicklung.	Typisierte Übermenge von JavaScript.	Moderne JavaScript-Funktionen.
Aufwand	-	Etwas steilere Lernkurve.	Einige Funktionen möglicherweise nicht in älteren Browsern unterstützt.

Kategorie	Vue.js	React	Angular
Leichtgewichtigkeit	Einfache Integration und Verwendung.	Große Community, mehr Ressourcen.	Große Community, mehr Ressourcen.
Dokumentation	Gut dokumentiert.	Gut dokumentiert.	Gut dokumentiert.

Kategorie	Vue.js	React	Angular
Aufwand	Für umfangreichere Projekte könnte React/Angular von größeren Communities profitieren.	-	-

Backend

Kategorie	Node.js/Express.js	Koa.js/Hapi.js
Leichtgewichtigkeit	Sehr leichtgewichtig und schnell.	Auch leichtgewichtig, aber möglicherweise spezifischere Use Cases.
Skalierbarkeit	Bekannt für hohe Geschwindigkeit und Skalierbarkeit.	Gute Skalierbarkeit, aber möglicherweise weniger schnell als Node.js.
Community-Support	Große und aktive Community.	Aktive Community, aber möglicherweise kleiner als die von Node.js/Express.js.
Dokumentation	Gut dokumentiert.	Gut dokumentiert, aber möglicherweise nicht so umfassend wie die von Node.js/Express.js.

Kategorie	PostgreSQL mit Neon	MySQL, MongoDB, SQLite
Datenbanktyp	Relationale Datenbank.	Unterschiedliche Typen: MySQL (relational), MongoDB (NoSQL), SQLite (eingebettet).
ACID-Transaktionen	Unterstützt ACID-Transaktionen für Datenintegrität.	Je nach Typ können ACID-Transaktionen unterschiedlich implementiert sein.
Performance	Sehr schnell, besonders in Verbindung mit Neon.	Performance variiert je nach Typ und Anwendungsfall.
Persönliche Präferenz	Persönliche Entscheidung.	Abhängig von den spezifischen Anforderungen und Präferenzen des Teams.

Kategorie	SocketIO	WebSocket-Implementierungen in anderen Sprachen
Echtzeitkommunikation	Sehr leichtgewichtig, unterstützt Real-Time Data.	Direktere Kontrolle über WebSocket-Implementierungen, könnten komplexer zu handhaben sein.
Integration mit Node.js	Sehr gut unterstützt in Node.js durch eigenes Package.	Implementierung in anderen Sprachen möglicherweise spezifischer.
Funktionsumfang	Hoher Funktionsumfang trotz leichtgewichtiger Natur.	Abhängig von der Implementierung, könnte weniger Funktionalität bieten als SocketIO.
Kompatibilität	Weit verbreitet und gut kompatibel mit verschiedenen Plattformen.	Kompatibilität kann je nach Implementierung variieren.

Authentifizierung

Kategorie	OAuth 2.0	JWT (JSON Web Tokens)	OpenID Connect
Sicherheit	Hohe Sicherheit durch etablierten Industriestandard.	Sicher, aber erfordert sichere Handhabung von Secret Keys.	Basierend auf OAuth 2.0, bietet zusätzliche Identitätsmanagementfunktionen.
Verbreitung	Breit unterstützter Standard in der Industrie.	Weit verbreitet, besonders für tokenbasierte Authentifizierung.	Verbreitet, bietet einen Standard für Identitätsmanagement.
Komplexität	Komplexer, da mehrere Schritte und Rollen beteiligt sind.	Einfach zu implementieren und zu verstehen.	Zusätzliche Funktionen können die Implementierung komplexer machen.
Flexibilität	Relativ flexibel, ermöglicht verschiedene Authentifizierungstypen.	Begrenzte Flexibilität, spezialisiert auf tokenbasierte Authentifizierung.	Flexibel, bietet zusätzliche Funktionen für Identitätsmanagement.

Kategorie	OAuth 2.0	JWT (JSON Web Tokens)	OpenID Connect
Token-Ausstellung	Token durch Authorisierungsserver ausgestellt.	Token werden vom Server signiert und können dezentral überprüft werden.	Token-Ausstellung durch Identitätsprovider.
Nutzungsszenarien	Geeignet für Anwendungen mit externen Diensten, die OAuth 2.0 unterstützen.	Gut für webbasierte Anwendungen, APIs und Single Sign-On.	Spezialisiert auf Identitätsmanagement, gut für Single Sign-On.

Tests

Kategorie	Cypress	Selenium	Jest	Protractor
Einfachheit	Intuitive Syntax.	Kann komplexer sein.	Einfache Syntax für JS.	Spezifisch für Angular.
Geschwindigkeit	Schnelle Testausführung.	Kann langsamer sein.	Schnell für JS.	Optimiert für Angular.
Plattformen	Cross-Browser-Unterstützung.	Breite Browser-Unterstützung.	Browser-unabhängig.	Hauptsächlich für Angular.
Community	Aktive und umfangreiche Community.	Große Community, nicht spezifisch für Frontend.	Starke JS-Community.	Starke Angular-Community.
Testarten	E2E, Integration, Unit.	Hauptsächlich E2E, anpassbar.	Hauptsächlich Unit, Integration.	Spezialisiert auf E2E für Angular.
Berichterstattung	Klare und detaillierte Berichte.	Möglicherweise zusätzliche Integration.	Eingebaute Berichterstattung.	Konfigurierbare Berichterstattung.
Automatisierung	Effektive Automatisierung.	Effektiv, erfordert mehr Konfiguration.	Effektiv für verschiedene Tests.	Spezialisiert auf Angular-Automatisierung.
Integration	Einfache CI/CD-Integration.	Möglicherweise mehr Konfiguration.	Nahtlose CI/CD-Integration.	Integration in CI/CD für Angular.

Kategorie	Cypress	Selenium	Jest	Protractor
Lernkurve	Geringe Lernkurve, JS vorausgesetzt.	Kann steil sein, besonders für Anfänger.	Geringe Lernkurve für JS.	Erfordert Angular-Wissen.
Flexibilität	Anpassbare und erweiterbare Testfälle.	Anpassbar, erfordert mehr Konfiguration.	Flexibel und erweiterbar.	Spezialisiert auf Angular, anpassbar.

Ausgewählte Technologien

Frontend

Die statischen visuellen Komponenten werden mit HTML, CSS und JavaScript umgesetzt, da diese Technologien standardisiert sind und eine hohe Geschwindigkeit des Frontends gewährleisten.

Dynamisch generierte Inhalte sollen durch die Integration von Vue.js realisiert werden, was durch einen einfachen Import ermöglicht wird.

Backend

Der Server basiert auf Node.js und Express.js. Node.js wurde aufgrund seiner hervorragenden Geschwindigkeit und Skalierbarkeit ausgewählt. Express.js zeichnet sich durch seine extrem schlanke Struktur aus, die zu einer besonders schnellen Performance beiträgt.

Die Datenbank wird relational sein, um eine strukturierte Verwaltung der Daten zu ermöglichen. Der bevorzugte Dialekt für die Datenbank ist PostgreSQL, basierend auf persönlichen Entscheidungen. Die Datenbank wird auf Neon gehostet, da Neon für seine Schnelligkeit bekannt ist und im Vergleich zur Konkurrenz immer verfügbar ist.

Echtzeitdaten werden über Socket.io ausgetauscht. Socket.io wird in Node.js durch ein eigenes Package optimal unterstützt und bietet trotz seiner umfangreichen Funktionalität eine einfache Implementierung.

Authentifizierung

Für die Authentifizierung wird OAuth 2.0 verwendet. Diese Technologie wird sowohl von den externen Diensten, die in diesem Projekt verwendet werden, gefordert, als auch aufgrund ihrer hohen Sicherheitsstandards ausgewählt.

Tests

Für das Testen wird Cypress verwendet, um eine effiziente und umfassende Überprüfung der Anwendung zu gewährleisten.