**Technologie Backend**

**Verwendung von Python als Programmiersprache ausfolgenden Gründen**

**Vielseitigkeit und Flexibilität**: Python ist eine äußerst vielseitige Programmiersprache, die für eine breite Palette von Anwendungen geeignet ist. Es bietet Flexibilität bei der Entwicklung von Backend-Funktionalitäten für Webanwendungen, unabhängig von ihrer Komplexität oder ihrem Umfang.

**Große Auswahl an Frameworks**: Python verfügt über eine Vielzahl von Web-Frameworks wie Django, Flask und Pyramid, die die Entwicklung des Backends erleichtern. Diese Frameworks bieten Funktionen wie Datenbankintegration, URL-Routing, Sicherheit und vieles mehr, was die Entwicklung beschleunigt und die Wartbarkeit verbessert.

**Riesige Entwickler-Community**: Python hat eine große und aktive Entwickler-Community, die kontinuierlich an der Verbesserung von Frameworks, Bibliotheken und Tools arbeitet. Dies bedeutet, dass Entwickler auf ein umfangreiches Ökosystem von Ressourcen und Unterstützung zugreifen können, was die Entwicklung effizienter macht und Problemlösungen erleichtert.

**Einfache Integration von Bibliotheken und APIs**: Python bietet eine einfache Integration von Bibliotheken und APIs, was es Entwicklern ermöglicht, auf eine Vielzahl von Funktionen und Diensten zuzugreifen. Dies ist besonders wichtig für Webanwendungen, die externe Dienste oder Datenbanken integrieren müssen.

**Skalierbarkeit und Leistung**: Obwohl Python im Vergleich zu einigen anderen Sprachen möglicherweise nicht die höchste Leistung bietet, sind viele der beliebten Frameworks und Bibliotheken gut optimiert und skalierbar. Dies ermöglicht es Entwicklern, leistungsfähige und skalierbare Webanwendungen zu erstellen, die mit wachsenden Nutzerzahlen umgehen können.

**Kurze Entwicklungszeiten**: Python ist bekannt für seine klare und lesbare Syntax, die die Entwicklung beschleunigen kann. Dies bedeutet, dass Entwickler in kürzerer Zeit hochwertigen Code schreiben können, was insgesamt zu kürzeren Entwicklungszeiten führt.

**Bestehende Erfahrung im Team:** Es besteht bereits Erfahrung innerhalb des Teams mit Python-Entwicklung.

**Auswahl Framework (Django vs. Flask)**

|  | Flask | Django |  |
| --- | --- | --- | --- |
| Kostenlos | X | X |  |
| Entwicklungszeit |  | X |  |
| Benutzerfreundlichkeit Anfänger | X |  | Flask ist leichter zu lernen und zu handhaben und ist somit auch für Anfänger geeignet. |
| Funktionen |  | X | Flask ist für zusätzliche Funktionen meist auf Erweiterungen von Drittanbietern angewiesen |
| Leistung | X | X | Flask ist leichtgewichtiger und ermöglicht es Projekte schneller aufzusetzen. Django ist bei größeren Projekten besser geeignet. |
| Skalierbarkeit |  | X | Django ist für große und komplexe Projekte besser geeignet |
| Geschwindigkeit | X |  | Flask ist flexibler und schneller als Django |

Beide Frameworks bringen ihre Vor- und Nachteile. Es ist jedoch nicht klar, ob die Vorteile von Django mit diesem Projekt vollkommen ausgenutzt werden können. Die Anfängerfreundliche Entwicklung mit Flask ist allerdings ein sehr großer Vorteil, da nicht alle Mitglieder des Teams Vorerfahrungen mit Python haben.   
Somit ist unser Framework für das Backend Flask.