Evaluation Frontend

Die Vorentscheidung welche Frameworks/Bibliotheken in die Frage für das Frontend kommen, wurde durch Überlegung an eine schnelle und leichte Integration, von Back- und Frontend, entschieden, hierdurch sind wurden Folgende Frameworks/Bibliotheken in die engere Auswahl gebracht:

* Vue.js
* React.js
* Angular

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Vue.js | React.js | Angular | Kommentar |
| Architektur | Komponenten | Komponenten | Komponenten | jedes der Frameworks arbeitet Komponentenbasiert |
| Größe und Leistung | Leichtgewichtig, gute Leistung | Leichtgewichtig, gute Leistung | Durchschnittliche Größe, leistungsfähig | Vue.js und React.js sind was den Umfang des Frameworks angeht vergleichbar, lediglich Angular ist komplexer |
| Entwicklungswerkzeuge | Vielfältige Tools und Erweiterungen, z. B. Vue CLI | Vielfältige Tools und Erweiterungen, z. B. Create React App | Umfangreiche Werkzeugsätze und Entwicklungsumgebungen,  z. B. Angular CLI |  |
| Integration | Gut geeignet | Gut geeignet | Gut geeignet | Alle Frameworks/Bibliotheken können gut mit dem geplanten Umgehen |
| Testsupport | Gut integriert | Gut integriert | Gut integriert | Jeder Kandidat hat eine Integrierte Testunterstützung |

Letztendlich hat sich die Auswahl durch die bisherigen Erfahrungen des Entwicklers für ie Kombination aus Angular und Flask entschieden. Dies bietet eine leistungsstarke Lösung für die Entwicklung moderner Webanwendungen. Durch die Integration dieser beiden Frameworks können Entwickler eine klar strukturierte, skalierbare und performante Anwendung entwickeln. Fortführend wird auch eine klare Trennung zwischen Front- und Backend, durch eine Anbindung Angular durch eine Restful API von Flask, sichergestellt. Dahingehend werden Skalierbarkeit und Wartbarkeit der Anwendung weiter unterstützt.

Die grundlegende Architektur von Angular bietet hier dem Frontend Entwickler eine bessere wieder Verwendbarkeit seiner bereits erstellten Komponenten.