

**Technischer Hintergrund**

Unsere Komponenten bestehen aus: Socket.io, Node.js, MongoDB, Express die Anwedung wurde mit HTML (Hypertext Markup Language, CSS (Cascading Style Sheets) und JavaScript verwendet.

Node.js ist eine serverseitige Plattform in der Softwareentwicklung zum Betrieb von Netzwerkanwendungen. Herzstück ist Googles V8-JavaScript-Engine, die ursprünglich für Chrome entwickelt wurde. Die Engine erhöht die Performance, indem sie JavaScript direkt in nativen Maschinencode kompiliert (Just-In-Time-Kompilierung). Eine entscheidende Besonderheit des Frameworks ist außerdem seine Fähigkeit zur Kommunikation mit Events sowie zur gleichzeitigen Verarbeitung mehrerer Client-Verbindungen, was Node.js vor allem für die Webserver-Entwicklung interessant macht.

Socket.IO ist eine JavaScript-Bibliothek für Echtzeit-Webanwendungen. Mit Socket.IO ermöglichen wir eine bidirektionale Echtzeit-Kommunikation zwischen Webclients und Server, ohne dabei merkliche Verarbeitungszeiten zu generieren oder ein ständiges Refreshen der Webseite durchzuführen. Socket.IO bietet Echtzeit-Analyse, binäre Datenströme, Sofortnachrichten und Zusammenarbeit an Dokumenten. Es bietet somit, dass Broadcasting an mehrere Sockets, wodurch es auch möglich ist unsere MP3´s oder MP4´s an alle Clients zu verschicken.

Wir haben uns für MongoDB entschieden. MongoDB ist eine dokumentorientierte NoSQL-Datenbank. Die Datenbank kann JSON-ähnliche Dokumente verwalten. Die Entscheidung für Mongo DB bestand, weil wir für unsere Anwendung keine komplexen Suchanfragen und Joinalgorithmen benötigen. Unsere Datenhaltung besteht aus einer Collection mit den Attributen user\_ID, username und password.

Express ist ein serverseitiges Webframework für Node.js. Express stellt die Datei app.js bereit, in welcher die Middleware und der Controller sind, außerdem startet dieser den Webserver.































