

Projekt Cosmo-Chat



Team

Aileen Schulz
Gerhard Schmitt
Sandro Giovanni Zito

Startup 08.02.2023

betreut durch Techstarter Team Ivan Babayev, Colin Schmidt, Benjamin Rüling und Stefan Gaugel

verwendete Frameworks & Tools

Figma
Miro
socket.io
Angular
Express
node.js
VSC
GITHUB
WiX

Repository

<https://github.com/xmoonflower/Projekt-Cosmo-Chat.git>

1. Einleitung
2. Technische Grundlagen
 - 2.1 Anforderungsanalyse
 - 2.2 Programmablauf
 - 2.3 Verwendung von technischen Werkzeugen
3. Beschreibung der Entwicklung
 - 3.1 Software-Architektur der Lösung
 - 3.2 Übersicht und Zusammenspiel der Komponenten
 - 3.3 Beschreibung der Komponenten
 - 3.3.1 Komponente 1 (“Backend”) Express
 - 3.3.1.1 Aufgaben und Funktionsweise der Komponente
 - 3.3.1.2 Schnittstellen der Komponente
 - 3.3.2 Komponente 2 (“Frontend”) Angular
 - 3.3.2.1.1 Aufgaben und Funktionsweise der Komponente
 - 3.3.2.1.2 Schnittstellen der Komponente
 - 3.4 Übersicht Verzeichnisse und Dateien (CLI tree)
4. Einrichtung und Betrieb der Software-Lösung
5. Retrospektive
6. Datenschutz
7. Lizenz und Disclaimer
8. Zusammenfassung und Ausblick

1. Einleitung

Zielsetzung und Motivation dieser Arbeit

Wir möchten eine sichere und legale Plattform bieten, welche es ermöglicht, User in Echtzeit zu verbinden. Es wird möglich sein, über das Interface in verschiedenen Chats zu kommunizieren. Die Zusammenstellung unserer Techniken soll eine sichere Kommunikation gewährleisten. Die Oberfläche soll eine Übersicht zeigen, wo Gruppen- und Einzelchats, sowie eine Liste aller User und welche, die aktiv sind, darstellen. Die Kanäle sollen als Backup über eine Datenbank verfügbar sein. Der Chat-Empfänger wird über eine neue Nachricht informiert. Der User kann Direktnachrichten verschicken an Einzelpersonen oder Gruppen.

2. Technische Grundlagen

2.1 Anforderungsanalyse

Anforderungsanalyse für den “Cosmo-Chat” auf Angular Code

Eine Chat-Software soll entwickelt werden, die es Benutzern ermöglicht, miteinander in Echtzeit zu kommunizieren. Die Anwendung soll mit Angular implementiert werden. Die Zielgruppe für die Chat-Software sind alle Benutzer, die sich über eine einfache und schnelle Kommunikation in Echtzeit unterhalten möchten, wie z.B. Teams, Freunde oder Familienmitglieder.

Anforderungen

Benutzerregistrierung: Benutzer sollen sich ohne E-Mail-Adresse & ohne Passwort registrieren können. Ein Username kann bzw. wird vom System generiert.

Login: Benutzer sollen sich mit ihren Anmeldeinformationen anmelden und eine Sitzung starten können.

Chaträume: Benutzer können in einem Main-Chatraum mit allen gleichzeitig chatten, die restlichen Unterhaltungen können nur in einem privaten Chat abgehalten werden.

Nachrichten: Benutzer sollen in Echtzeit Nachrichten an andere Benutzer senden und empfangen können.

Benachrichtigungen: Nutzer sollen über neue Nachrichten benachrichtigt werden.

Anhänge: Benutzer sollen Dateien wie Bilder, Videos und Dokumente an andere Benutzer senden können.

Sicherheit: Die Chat-Software soll sicher sein und die Daten der Benutzer geschützt werden.

Zugänglichkeit: Die Chat-Software soll auf verschiedenen Geräten und Browsern zugänglich sein.

Technische Anforderungen

Angular: Die Chat-Software soll mit Angular als Frontend implementiert werden.

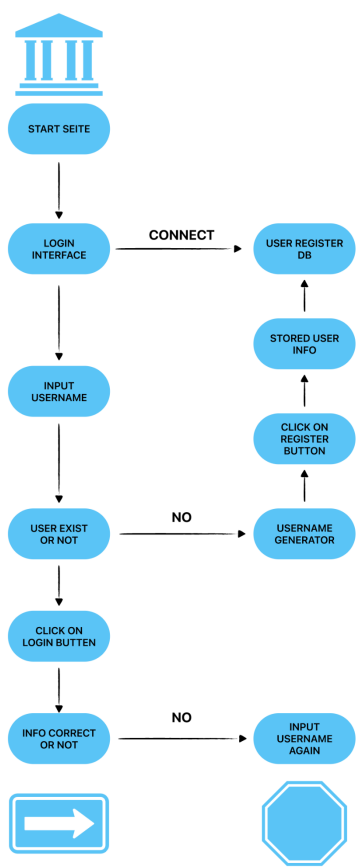
Express: Die Chat-Software soll Express als Backend-Service verwenden.

Real-time Datenbank: Die Chat-Software soll eine Echtzeit-Datenbank verwenden, um Nachrichten in Echtzeit zu senden und zu empfangen.

Zusammenfassung

Diese Anforderungsanalyse beschreibt die wichtigsten Funktionen und Anforderungen für eine Chat-Software auf Angular. Die Anwendung soll Benutzern eine einfache und sichere Möglichkeit bieten, in Echtzeit zu kommunizieren, indem sie Nachrichten an andere Benutzer senden und empfangen. Die Anwendung soll mit Angular als Frontend-Framework implementiert werden und Express als Backend-Service verwenden.

2.2 Programmablauf



2.3 Verwendung von technischen Werkzeugen

Für unser Projekt verwenden wir Miro als Dashboard zur Planung des Projektes. Das Design für die UI benutzen wir Figma und unser Logo haben wir bei WiX erstellt. Die Codierung findet auf Visual Studio Code statt, wo wir Packages und Dependencies von Angular implementiert haben, um eine App zu erstellen, die online geht. Die API-Schnittstellen sind über socket.io und Express und stellen das Backend dar. Angular und Express verwendet node.js. Der ganze Projektordner als Repository ist aufrufbar auf GitHub.

3. Beschreibung der Entwicklung

3.3.1

Wir benutzen für unser Projekt "Cosmo-Chat", das Backend Express. Express ermöglicht, dass im Hintergrund Operationen vorgenommen werden, die mit verschiedenen Schnittstellen kooperieren können.

Da wir mit unzähliger **HTTP-Dienstprogrammmethoden** und **Middlewarefunktionen** unser Projekt gestalten, und sich einer Erstellen einer leistungsfähigen API schnell und einfach. (gestaltet sich das Erstellen einer leistungsfähigen API schnell und einfach.)

3.3.2

Als Frontend haben wir uns für das Framework Angular entschieden. Angular hat den Vorteil einer Konsistenz **“Codekonsistenz”**, Produktivität **“Richtige Weise tun”**, Wartbarkeit **“In Gruppen am Projekt arbeiten”**, Modularität **“Code in Buckets zu Organisieren”**, Frühzeitige Fehlererkennung **“Angular wurde mit TypeScript erstellt, was viele Vorteile mit sich bringt”**, und eine gute Zusammenfassung bei Entwicklungsteam.

Das Frontend ist der Teil einer Webseite, mit der der Benutzer direkt interagiert. Es kann auch als die „Client-Seite“ bezeichnet werden. Das Frontend enthält Elemente, die Benutzer direkt wahrnehmen: Textschriftarten, Farben und Stile, Bilder, Grafiken und Tabellen, Schaltflächen, Navigationsmenü usw.

Die Sprachen, die zur Entwicklung des Frontends einer Website verwendet werden, sind HTML, CSS und JavaScript. Die Struktur, das Design, das Verhalten und der Inhalt von allem, was in Browsern beim Öffnen von Websites, Webanwendungen oder mobilen Apps angezeigt wird, werden von Frontend-Entwicklern implementiert.

Ein gutes Frontend muss eine hohe Leistung aufweisen (schnelles Laden) und „responsive“ sein. Eine responsive Seite wird auf Geräten aller Größen, vom Computermonitor bis zum Smartphone, korrekt angezeigt. Alle wichtigen Teile des Frontends der Website sollten so gerendert werden, dass sie der Bildschirmgröße entsprechen.

Angular

Ein Open Source Webanwendungsframework. Angular ist ein komponentenbasiertes Framework, das deklarative HTML-Vorlagen verwendet. Der Compiler des Frameworks übersetzt die Vorlagen in optimierte JavaScript-Anweisungen. das hauptsächlich zum Entwickeln von Single-Page-Webanwendungen und dynamischen Webanwendungen verwendet wird.

4. Einrichtung und Betrieb der Software-Lösung

Inbetriebnahme der Lösung (Installationsanweisung)

Hier steht eine Installationsanweisung

5. Retrospektive

Bewertung Arbeitsweise, Zusammenarbeit und Erkenntnisse im Team

Milestone I: Zusammen im Team wurde die Zeit der ersten Woche an strukturellem Planning gearbeitet sowie die Planung des Design und Konzeptes als auch die ersten schriftlichen Dokumentationen festgehalten. Das Team war zufrieden.

Milestone II: Die Dokumentation wurde hinreichend gedeckt und erste Coding Prozesse erarbeitet.

Milestone III:

6. Datenschutz

In diesem Abschnitt beschreiben wir explizit, wie mit den personenbezogenen Daten der Nutzer umgegangen wird. Daten werden nicht an Dritte weitergegeben.

Personenbezogene Daten müssen dem Zweck angemessen und erheblich sowie auf das für die Zwecke der Verarbeitung notwendige Maß beschränkt sein („Datenminimierung“). Das Prinzip der Datenminimierung sorgt auch dafür, dass von bereits vorhandenen personenbezogenen Daten keine neuen abgeleitet werden dürfen.

Sie können unsere Webseiten besuchen, ohne Angaben zu Ihrer Person zu machen. Bei jedem Aufruf unserer Webseite speichert der Webserver lediglich automatisch ein sogenanntes Server-Logfile, das z.B. Ihre IP-Adresse, Datum und Uhrzeit des Abrufs, übertragene Datenmenge und den anfragenden Provider (Zugriffsdaten) enthält und den Abruf dokumentiert.

7. Lizenz und Disclaimer

Diese Software ist urheberrechtlich geschützt (C) 2023, Aileen Schulz, Sandro Giovanni Zito. Alle Rechte vorbehalten, außer wie unten angegeben.

Rechtsfragen

1. Wir versprechen nicht, dass diese Software funktioniert. (Aber wenn Sie Fehler finden, teilen Sie uns dies bitte mit!)
2. Sie können diese Software für alles verwenden, was Sie wollen. Sie ist OpenSource und Sie müssen uns nicht bezahlen.
3. Sie dürfen nicht vorgeben, dass Sie diese Software geschrieben haben. Wenn Sie es in einem Programm verwenden, müssen Sie irgendwo in Ihrer Dokumentation bestätigen, dass Sie den Code von Cosmo-Chat verwendet haben.

Hiermit wird die Erlaubnis erteilt, diese Software (oder Teile davon) für jeden Zweck ohne Gebühr und vorbehaltlich dieser Bedingungen zu verwenden, zu kopieren, zu modifizieren und zu verteilen.

1. Wenn ein Teil des Quellcodes dieser Software weitergegeben wird, müssen Sie den Urheberrechtshinweis, diese Liste der Bedingungen und den folgenden Haftungsausschluss enthalten.
2. Weder der Name des Urheberrechtsinhabers noch die Namen seiner Mitwirkenden dürfen ohne ausdrückliche vorherige schriftliche Genehmigung verwendet werden, um von dieser Software abgeleitete Produkte zu unterstützen oder zu bewerben.

Disclaimer / Urheberrecht

DIESE SOFTWARE WIRD VON DEN URHEBERRECHTSINHABERN UND MITWIRKENDEN VON "Cosmo-Chat" ZUR VERFÜGUNG GESTELLT UND ALLE GEWÄHRLEISTUNGEN, EINSCHLIESSLICH DER MARKTFÄHIGKEIT UND EIGNUNG FÜR EINEN BESTIMMTEN ZWECK, WERDEN AUSGESCHLOSSEN. IN KEINEM FALL HAFTET DER INHABER DES URHEBERRECHTS ODER DIE MITARBEITER FÜR FOLGESCHÄDEN (NUTZUNGS-, DATEN- ODER GEWINNVERLUST; ODER GESCHÄFTSUNTERBRECHUNG), WELCHE VERURSACHT WERDEN KÖNNEN UND AUF JEDER THEORIE DER HAFTUNG, DIE IN IRGEND EINER WEISE AUS DER VERWENDUNG DIESER SOFTWARE ENTSTEHEN.