

Testprotokoll

Blackbox Test

Chat App

Projekt	Chat App
Datum	19.03.2026
Tester	
Version	1.0

1 Testumgebung

Parameter	Wert
Browser	Chrome / Firefox
URL	http://localhost
Methode	Manueller Blackbox Test

2 Beschreibung der Systemteile

Komponente	Technologie	Aufgabe
Frontend	React, Vite, Socket.IO Client, Nginx	Benutzeroberfläche für den Chat. Stellt Verbindung zum Backend über WebSocket her und zeigt Nachrichten, Teilnehmer und Verbindungsstatus an.
Backend	Node.js, Express, Socket.IO	REST-API und WebSocket-Server. Verwaltet Benutzer-Sessions, Räume und leitet Nachrichten in Echtzeit weiter.
Datenbank	MongoDB	Persistente Speicherung von Nachrichten und Benutzerdaten. Alternativ steht ein JSON-Fallback-Store zur Verfügung.
Infrastruktur	Docker, Docker Compose, Nginx	Containerisierung aller Dienste. Nginx fungiert als Reverse Proxy für das Frontend.

3 Vorschläge zur Ausfallsicherheit

■ Frontend

- Automatischer Reconnect bei WebSocket-Verbindungsabbruch ist bereits implementiert (5 Versuche)
- Client-seitige Nachrichten-Queue: Nachrichten lokal zwischenspeichern und bei Reconnect nachsenden

■ Backend

- Mehrere Instanzen hinter einem Load Balancer betreiben (z. B. Nginx upstream oder HAProxy)
- Bei mehreren Instanzen Socket.IO mit Redis Adapter koppeln, damit alle Instanzen dieselben Räume/Events teilen

■ Datenbank

- MongoDB Replica Set einrichten: primärer Node schreibt, sekundäre Nodes lesen
- Regelmässige automatische Backups des Volumes einrichten
- Redis als Caching-Layer einsetzen, um häufige Lesezugriffe zu beschleunigen und die Datenbank zu entlasten

■ Infrastruktur

- Für Produktion: Kubernetes für automatisches Failover, Skalierung und Rolling Updates einsetzen
- Monitoring mit Prometheus + Grafana einrichten, um Metriken wie CPU, RAM und Fehlerraten zu überwachen
- Nginx als Reverse Proxy mit TLS-Terminierung konfigurieren (Let's Encrypt)
- Secrets (z. B. MONGO_ROOT_PASSWORD) über Kubernetes Secrets verwalten statt .env-Dateien

4 Testfälle

Nr.	Beschreibung	Eingabe	Erwartetes Ergebnis	Status
T01	Beitreten mit gültigem Benutzernamen und Raum	Username: Max, Room: general	Benutzer tritt dem Raum bei, Chatverlauf wird angezeigt	[OK]
T02	Beitreten ohne Benutzernamen	Username: (leer), Room: general	Fehlermeldung wird angezeigt, kein Beitritt	[OK]
T03	Beitreten ohne Raumnamen	Username: Max, Room: (leer)	Fehlermeldung wird angezeigt, kein Beitritt	[OK]
T04	Nachricht senden	Nachricht: Hallo! → Senden	Nachricht erscheint im Chat	[OK]
T05	Leere Nachricht senden	Nachricht: (leer) → Senden	Nachricht wird nicht gesendet	[OK]
T06	Zwei Benutzer im selben Raum	Tab 1: Max / general, Tab 2: Anna / general	Beide sehen Nachrichten des jeweils anderen in Echtzeit	[OK]
T07	Benutzer in verschiedenen Räumen	Tab 1: Max / general, Tab 2: Anna / privat	Nachrichten sind voneinander getrennt	[OK]
T08	Teilnehmerliste aktualisiert sich	Tab 2 schliessen	Tab 1 zeigt Benutzer als offline/entfernt	[OK]
T09	Chatverlauf nach Neuverbindung	Seite neu laden, wieder einloggen	Vorherige Nachrichten werden geladen	[OK]
T10	Sonderzeichen in Nachricht	Nachricht: <code><script>alert(1)</script></code>	Text wird als plain text angezeigt, kein Script ausgeführt	[OK]
T11	Langer Text	Nachricht mit 500 Zeichen	Nachricht vollständig angezeigt, kein Layout-Fehler	[OK]
T12	Backend nicht erreichbar	Backend stoppen, App öffnen	Fehlermeldung oder Reconnect-Versuch wird angezeigt	[OK]