



***Finish This!***

Individuelle Abschlussarbeit BLJ

Levi Agostinho Horta

Adnovum AG

**25.06.2025**  **Unterschrift:**

Inhaltsverzeichnis

[1. Änderungstabelle 2](#_Toc201674938)

[2. Einleitung 2](#_Toc201674939)

[2.1 Projekziele 3](#_Toc201674940)

[2.2 Benutzte Technologie 3](#_Toc201674941)

[2.3 Bekannte Risiken 3](#_Toc201674942)

[3. Planung 4](#_Toc201674943)

[3.1 Terminplan 4](#_Toc201674944)

[3.2 Risikoanalyse 5](#_Toc201674945)

[3.3 Entscheidungsmatrix 5](#_Toc201674946)

[4. Hauptteil 6](#_Toc201674947)

[4.1 Umsetzung und Zwischenschritte 6](#_Toc201674948)

[4.1.1 Phase 1: Grundlagen (Tag 1-3) 6](#_Toc201674949)

[4.1.2 Phase 2: Multiplayer-Features (Tag 4-8) 6](#_Toc201674950)

[4.1.3 Phase 3: Finalisierung (Tag 9-12) 7](#_Toc201674951)

[4.2 Ergebnis der Arbeit 7](#_Toc201674952)

[4.2.1 Kernfuntionalitäten 7](#_Toc201674953)

[4.2.2 Technische Leistungen 7](#_Toc201674954)

[4.3 Funktionsbeschreibung 8](#_Toc201674955)

[4.3.1 Komponenten 8](#_Toc201674956)

[4.3.2 Spielablauf 8](#_Toc201674957)

[4.4 Arbeitsjournal 8](#_Toc201674958)

[4.4.1 Tag 1: 02.06.2025 8](#_Toc201674959)

[4.4.2 Tag 2: 03.06.2025 9](#_Toc201674960)

[4.4.3 Tag 3: 04.06.2025 9](#_Toc201674961)

[4.4.4 Tag 4: 10.06.2025 9](#_Toc201674962)

[4.4.5 Tag 5: 11.06.2025 10](#_Toc201674963)

[4.4.6 Tag 6: 16.06.2025 10](#_Toc201674964)

[4.4.7 Tag 7: 17.06.2025 10](#_Toc201674965)

[4.4.8 Tag 8: 18.06.2025 11](#_Toc201674966)

[4.4.9 Tag 9: 20.06.2025 11](#_Toc201674967)

[4.4.10 Tag 10: 23.06.2025 11](#_Toc201674968)

[4.4.11 24.06.2025 12](#_Toc201674969)

[4.5 Testplan 12](#_Toc201674970)

[5. Anhang 13](#_Toc201674971)

[5.1 Glossar 13](#_Toc201674972)

[5.2 Codeausschnitte 14](#_Toc201674973)

[5.2.1 Emoji-Voting System 14](#_Toc201674974)

[5.2.2 Polling-Mechanismus für Echtzeit-Updates 15](#_Toc201674975)

[5.2.3 Datenbankabfrage für Sätze mit Votes 15](#_Toc201674976)

[5.3 Screenshot-App 16](#_Toc201674977)

[5.4 KI-Chat-Auszüge 17](#_Toc201674978)

[5.5 Quellen und Literatur 17](#_Toc201674979)

[6. Fazit und Reflexion 17](#_Toc201674980)

# Änderungstabelle

|  |  |  |  |
| --- | --- | --- | --- |
| **Wer** | **Was** | **Wann** | **Version** |
| Levi Agostinho Horta | Dokumenation starten | 02.06.2025 | 1.0 |
| Levi Agostinho Horta | Einleitung und Planung geschrieben | 03.06.2025 | 1.1 |
| Levi Agostinho Horta | Entscheidungsmatrix und Doku erweitert mit Bildern. | 04.06.2025 | 1.2 |
| Levi Agostinho Horta | Arbeitsjournal erweitert | 10.06.2025 | 1.3 |
| Levi Agostinho Horta | Arbeitsjournal erweitert | 11.06.2025 | 1.4 |
| Levi Agostinho Horta | Arbeitsjournal erweitert | 16.06.2025 | 1.5 |
| Levi Agostinho Horta | Arbeitsjournal erweitert | 17.06.2025 | 1.6 |
| Levi Agostinho Horta | Arbeitsjournal erweitert | 18.06.2025 | 1.7 |
| Levi Agostinho Horta | Arbeitsjournal erweitert | 20.06.2025 | 1.8 |
| Levi Agostinho Horta | Arbeitsjournal und Titelblatt erweitert | 23.0.6.2025 | 1.9 |
| Levi Agostinho Horta | Arbeitsjournal erweitert | 24.06.2025 | 2.0 |
| Levi Agostinho Horta | Vollständige Dokumentation mit allen Anforderungen | 25.06.2025 | 2.1 |

# Einleitung

**Einleitung**

Im Rahmen der individuellen Abschlussarbeit BLJ habe ich eine kreative Web-App namens **"Finish This!"** entwickelt. Ziel war es, eine Multiplayer-Anwendung zu schaffen, bei der Spieler einen Satzanfang erhalten und diesen kreativ beenden. Anschliessend können alle Spieler die besten Endungen per Emoji bewerten. Dies fördert sowohl Kreativität als auch Interaktion.

## Projekziele

**Projektziele**

* Entwicklung einer funktionsfähigen Multiplayer-Web-Anwendung
* implementierung eines benutzerfreundlichen Voting-Systems
* Erstellung einer responsiven Benutzeroberfläche
* Integration einer Datenbank für persistente Datenspeicherung
* Deployment der Anwendung auf einer Cloud-Plattform

## Benutzte Technologie

**Benutze Technologie**

Die Umsetzung erfolgte mit modernen Webtechnologien:

* **Frontend**: React mit TypeScript für typsichere Entwicklung
* **Backend**: Node.js mit Express.js für die API-Entwicklung
* **Datenbank**: PostgreSQL für zuverlässige Datenspeicherung
* **Deployment**: Render für Cloud-Hosting
* **Versionskontrolle**: Git/GitHub für Projektverwaltung

## Bekannte Risiken

**Bekannte Risiken**

* WebSocket-Kommunikation im Multiplayer
* Zeitmanagement für Feature-Umsetzung
* UI/UX-Design für einfache Benutzerführung
* Datenbankintegration und Performance-Optimierung
* Deployment-Konfiguration und Produktivumgebung

# Planung

## Terminplan

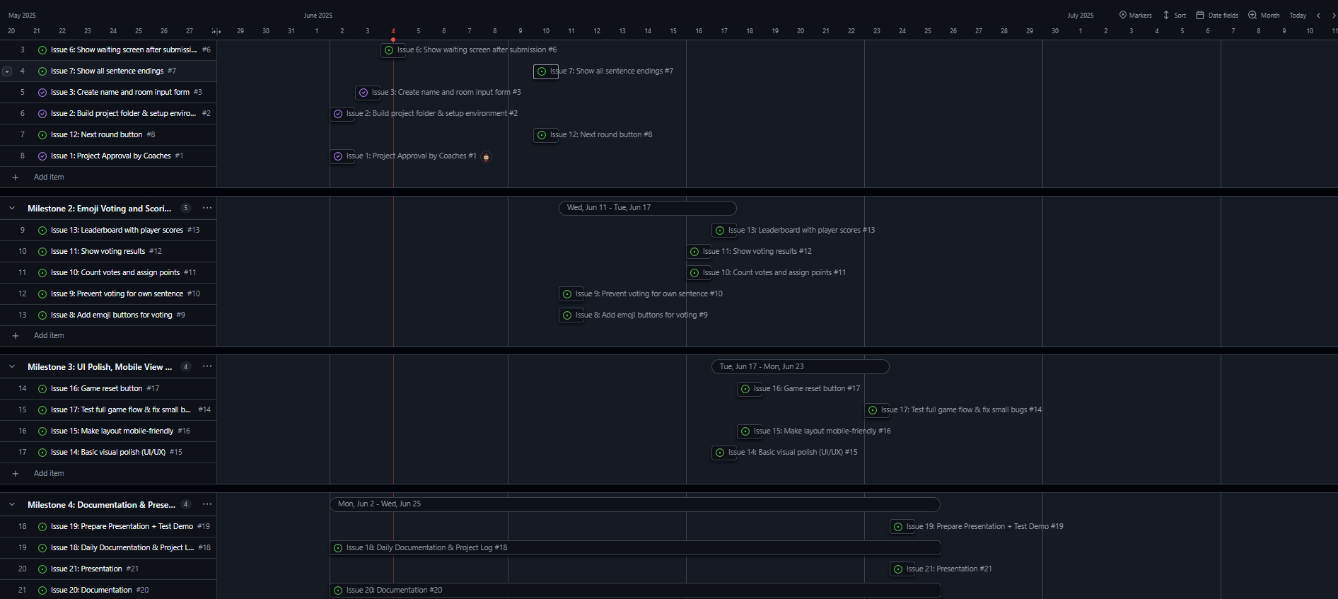
**Terminplan**

|  |  |  |
| --- | --- | --- |
| **Tag** | **Umsetzung** | **Status** |
| Tag 1 | Projektsetup, Grundstruktur, GitHub eingerichtet | ✅ |
| Tag 2 | JoinForm mit State | ✅ |
| Tag 3 | Satzanfang und Eingabe Feld | ✅ |
| Tag 4 | Next Round Button und Emojis voting | ✅ |
| Tag 5 | Verbindung von Frontend und backend | ✅ |
| Tag 6 | Ranking von Sätzen und Podium von Spielern | ✅ |
| Tag 7 | UI und Design angepasst sowie Benutzerfreundlicher gemacht | ✅ |
| Tag 8 | PostgreSQL implementiert und Voting und Ranking System funktionsfähig gemacht | ✅ |
| Tag 9 | Warte Zimmer implementiert und Benutzerfreundlichkeit verbssert | ✅ |
| Tag 10 | Deployement auf Render und Name von beigetretene Spieler sichtbar machen. | ✅ |
| Tag 11 | Präsentation anfangen und fertig machen sowie Doku | ✅ |
| Tag 12 | Bug fixes und Präsentationsvorbereitung | ✅ |

## 3.2 Risikoanalyse

**Risikoanalyse**

|  |  |  |  |
| --- | --- | --- | --- |
| Risiko | Wahrscheinlichkeit | Auswirkung | Massnahmen |
| Zeitüberschreitung | Mittel | Hoch | Priorisierung der Kernfeatures |
| Backend- Komplexität | Hoch | Mittel | Schrittweise Implementierung |
| Deployment-Probleme | Mittel | Hoch | Frühe Tests der Deployment-Pipeline |
| UI/UX-Herausforderungen | Niedrig | Mittel | Styling-Verbesserungen |



**Abbildung von Roadmap**

## 3.3 Entscheidungsmatrix

**Entscheidungsmatrix**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Option** | **Aufwand** | **Nutzen** | **Komplexität** | **Technische Marchbarkeit** | **Entscheid** |
| Satzspiel mit Voting | Mittel | Hoch | Mittel | Hoch | Gewählt |
| Websocket-Spiel | Hoch | Hoch | Hoch | Mittel | Später |
| Lokale Spieleingabe | Niedrig | Mittel | Niedrig | Sehr hoch | Umgesetzt |
| React + TypeScript | Mittel | Hoch | Mittel | Hoch | Gewählt |
| PostgreSQL Datenbank | Mittel | Hoch | Mittel | Hoch | Gewählt |
| Mobile-First Design | Hoch | Hoch | Hoch | Mittel | Implementiert |

# Hauptteil

## Umsetzung und Zwischenschritte

**Umsetzung und Zwischenschritte**

Die App wurde mit React (TypeScript) umgesetzt. Zunächst wurde das Projekt strukturiert und das Frontend gestartet. Anschliessend folgten JoinForm, die Speicherung von Eingaben und die Anzeige eines zufälligen Satzanfangs.

### Phase 1: Grundlagen (Tag 1-3)

**Phase 1: Grundlagen (Tag 1-3)**

* Projekt-Setup mit React und TypeScript
* Grundlegende UI-Komponenten (JoinForm, GameScreen)
* Lokale State-Verwaltung
* Basis-Funktionalität für Satzanfänge und Eingaben

### 4.1.2 Phase 2: Multiplayer-Features (Tag 4-8)

**Phase 2: Multiplayer-Features (Tag 4-8)**

* Backend-Integration mit Node.js/Express
* Emoji-Voting-System
* PostgreSQL-Datenbankintegration
* Ranking- und Scoring-System
* API-Endpunkte für alle Spielfunktionen

### 4.1.3 Phase 3: Finalisierung (Tag 9-12)

**Phase 3: Finalisierung (Tag 9-12)**

* Wartezimmer-Funktionalität
* Reset-Mechanismen
* Deployment-Optimierung
* UI/UX-Verbesserungen
* Mobile Responsiveness

## Ergebnis der Arbeit

**Ergebnis der Arbeit**

Der Spieler kann einen Namen und einen Raumcode eingeben, woraufhin ein Satzanfang angezeigt wird. Danach hat er die Möglichkeit, diesen mit maximal 200 Zeichen zu beenden. Die Eingabe wird validiert.

### 4.2.1 Kernfuntionalitäten

**Kernfunktionalitäten**

* Multiplayer-Raumverwaltung mit Raumcodes
* Zufällige Satzanfang-Generierung aus einer kuratierten Liste
* Eingabevalidierung (max. 200 Zeichen, Pflichtfelder)
* Emoji-basiertes Voting-System (😍, 😂, 🤔, 💩)
* Echtzeit-Scoring und Ranking
* Wartezimmer mit Host-Kontrolle
* Reset-Funktionalität für neue Spiele
* Responsive Design für alle Gerätetypen

### 4.2.2 Technische Leistungen

**Technische Leistungen**

* Vollständige REST-API mit 12 Endpunkten
* PostgreSQL-Integration mit optimierten Queries
* TypeScript-Integration für Typsicherheit
* Deployment auf Render mit Umgebungsvariablen
* Git-Versionskontrolle mit strukturierten Commits

## Funktionsbeschreibung

**Funktionsbeschreibung**

### 4.3.1 Komponenten

**Komponenten**

* **JoinForm**: Spielername + Raumcode Eingabe, Wechsellogik
* **GameScreen**: Anzeige eines zufälligen Satzanfangs + Eingabefeld für Satzende
* **WaitingRoom**: Warte-Raum für Spieler mit einem Host, der entscheiden kann, wann das Spiel startet – egal, ob der Timer fertig ist oder nicht
* **Validation**: Eingabeprüfung auf leeres Feld und maximale Länge
* **Zustandsverwaltung**: via React useState

### 4.3.2 Spielablauf

**Spielablauf**

1. **Spielstart und Raum-Beitritt**: Eingabe von Spielername und Raumcode
2. **Wartezimmer-Mechanik**: Anzeige aller beigetretenen Spieler mit Timer
3. **Spielrunde**: Zufällige Auswahl eines Satzanfangs, Eingabe der Beendigung
4. **Bewertungsphase**: Emoji-Voting für alle Sätze ausser dem eigenen
5. **Ranking**: Automatische Berechnung und Anzeige der Punkte
6. **Reset-Option**: Vollständiger Neustart des Spiels möglich

## Arbeitsjournal

### Tag 1: 02.06.2025

**Tag 1: 02.06.2025**

Heute haben wir mit der Abschlussarbeit begonnen.

Ich habe gelernt, wie Typescript genau funktioniert, und mir wurde klar, wie ich mit React arbeite und die nötigen Installationen durchführe.

Ausserdem hatte ich Probleme mit den Styles und den Farben der Startseite. Nachdem ich andere um ihre Meinung gebeten hatte, konnte ich schliesslich eine gute Lösung finden.

Heute ging es mir gut und ich war gut gelaunt.

### Tag 2: 03.06.2025

**Tag 2: 03.06.2025**

Heute habe ich an meinem Abschlussprojekt weitergearbeitet.

Ich habe gelernt, wie man einen zufälligen Satz aus einer JSON-Datei ausliest und wie man in Typescript einen beliebigen Satz auswählt und auch, wie man ein Eingabefeld erstellt, um die Endung der Sätze zu bestimmen.

Ausserdem hatte ich Schwierigkeiten mit dem Input-Feld und dem Styling, aber ich habe es so hinbekommen, dass es nicht schlecht aussieht.

Heute war ich gut gelaunt und es ging mir gut.

### Tag 3: 04.06.2025

**Tag 3: 04.06.2025**

Heute habe ich an dem ersten Meilenstein weitergearbeitet und bin damit fast fertig geworden.

Ich habe gelernt, wie ich alle Antworten von Fake-Spielern anzeigen lassen kann und auch, wie man mit einem Effekt solche Ladepunkte erstellen kann.

Ausserdem hatte ich noch Probleme mit diesem Effekt, weil ich nicht wusste, wie ich ihn implementieren sollte. Mithilfe des Internets konnte ich dann aber eine gute Lösung finden.

Heute ging es mir gut und ich war gut gelaunt.

### Tag 4: 10.06.2025

**Tag 4: 10.06.2025**

Heute habe ich Meilenstein 1 fertiggestellt und mit Meilenstein 2 begonnen.

Ich habe gelernt, wie man ein Emoji-System hinzufügt und wie man es stylen kann, um alles kompakt und passend miteinander zu gestalten.

Ausserdem hatte ich Schwierigkeiten mit dem Styling, weil man den Next-Round-Button nicht sah und auch mit den Emojis bei den Auswahlen. Nachdem ich verschiedene Vorschläge aus dem Internet ausprobiert hatte, gelang es mir.

Heute war ich gut gelaunt und es ging mir gut.

### Tag 5: 11.06.2025

**Tag 5: 11.06.2025**

Heute habe ich mit Meilenstein 2 weitergemacht.

Ich habe gelernt, wie ich das Frontend mit dem Backend verbinde und welche Anpassungen ich vornehmen muss, damit alles reibungslos funktioniert.

Ausserdem hatte ich noch Probleme mit ein paar Erweiterungen, die ich währenddessen umsetzen musste, damit alles flüssig laufen würde. Am Ende des Tages funktionierte jedoch alles.

Mir ging es heute gut und ich war gut gelaunt.

### Tag 6: 16.06.2025

**Tag 6: 16.06.2025**

Heute bin ich fast mit Meilenstein 2 fertig geworden.

Ich habe gelernt, wie ich ein Ranking der besten Sätze erstelle und wie ich ein Podium erstelle, das die Spieler nach ihren besten Sätzen ranken.

Ausserdem hatte ich noch Schwierigkeiten, weil meine GET-sentence-Anfrage nicht mehr funktioniert hat. Nachdem ich der URL dann wieder richtig angepasst hatte, hat es funktioniert.

Heute war ich gut gelaunt und es ging mir gut.

### Tag 7: 17.06.2025

**Tag 7: 17.06.2025**

Heute habe ich Meilenstein 2 abgeschlossen und mit Meilenstein 3 weitergemacht.

Ich habe gelernt, wie ich meine Oberfläche benutzerfreundlicher gestalten und Animationen hinzufügen kann, damit alles realistischer wirkt.

Ausserdem hatte ich Probleme mit Positionen und Grössen, aber am Schluss konnte ich alles in Ordnung bringen.

Heute ging es mir gut und ich war gut gelaunt.

### Tag 8: 18.06.2025

**Tag 8: 18.06.2025**

Heute habe ich den Meilenstein 3 fast fertig gemacht.

Ich habe gelernt, wie man PostgreSQL benutzt und wie man auch die Installation durchführt und auch Voting, sowie Ranking System funktionsfähig gemacht.

Ausserdem hatte ich Schwierigkeiten, mit den Voting und Ranking System aber als ich dann einen separaten Endpunkt machte, also einen Für die Sentences und einen für die Ranking von den Spielern, und dann ging es.

Heute war ich gut gelaunt und es ging mir gut.

### Tag 9: 20.06.2025

**Tag 9: 20.06.2025**

Heute gab es einen Ausfall der BMS, weshalb ich noch an meinem Abschlussprojekt arbeiten konnte.

Ich habe gelernt, wie ich ein Wartezimmer erstellen und eine Funktion einbinden kann, die es einem Host ermöglicht, den Start des Spiels zu bestimmen.

Ausserdem hatte ich Probleme, diese Force-Start-Funktion einzubinden, aber nachdem ich mein Backend und Frontend korrigiert hatte, funktionierte es.

Heute ging es mir gut und ich war gut gelaunt.

### Tag 10: 23.06.2025

**Tag 10: 23.06.2025**

Heute habe ich an der Veröffentlichung meines Spiels gearbeitet und zusätzlich die Option hinzugefügt, den Namen der beigetretenen Spieler zu sehen.

Ich habe gelernt, wie man das Deployment auf Render durchführt und wie man dort seine Datenbank aus PostgreSQL integriert.

Ausserdem hatte ich Schwierigkeiten, meinen Code anzupassen, da ich die URLs verbessern und auch ein paar Fetch-Funktionen anpassen musste.

Heute war ich gut gelaunt und es ging mir gut.

### 24.06.2025

**Tag 11: 24.06.2025**

Heute habe ich mit der Präsentation begonnen, sie fertiggestellt und einige Fehler behoben.

Ich habe gelernt, wie ich meinen Render-Server immer deployen kann, sobald mein GitHub-Repository aktualisiert ist, und auch, wie ich dies verwalten kann.

Ausserdem hatte ich Probleme mit meiner Dokumentation, weil ich nicht genau wusste, wie ich sie am besten gestalten sollte. Nachdem ich mal begonnen hatte, wurde mir klarer, wie es geht.

Heute ging es mir gut und ich war gut gelaunt.

## Testplan

**Testplan**

|  |  |  |  |
| --- | --- | --- | --- |
| **Testfall** | **Beschreibung** | **Erwartetes Verhalten** | **Status** |
| TF-001 | Leeres Nickname-Feld | Fehlermeldung | ✅ |
| TF-002 | Satzende länger als 200 Zeichen | Fehlermeldung | ✅ |
| TF-003 | Satzende > 200 Zeichen | Eingabe wird abgeschnitten/Fehlermeldung | ✅ |
| TF-004 | Satz korrekt gespeichert | Ausgabe in Konsole + Bestätigung | ✅ |
| TF-005 | Emoji-Voting funktional | Vote wird gezählt und angezeigt | ✅ |
| TF-006 | Eigener Satz nicht votbar | Voting-Buttons deaktiviert | ✅ |
| TF-007 | Next Round Reset | Neuer Satzanfang + Input-Reset | ✅ |
| TF-008 | Ranking korrekt | Spieler nach Score sortiert | ✅ |
| TF-009 | Game Reset | Vollständiger State-Reset | ✅ |
| TF-010 | Mobile Responsiveness | Layout passt sich an | ✅ |
| TF-011 | Wartezimmer Timer | 60-Sekunden-Countdown funktioniert | ✅ |
| TF-012 | Host Force-Start | Host kann Spiel manuell starten | ✅ |
| TF-013 | Datenbankabfrage | < 100ms | ✅ |

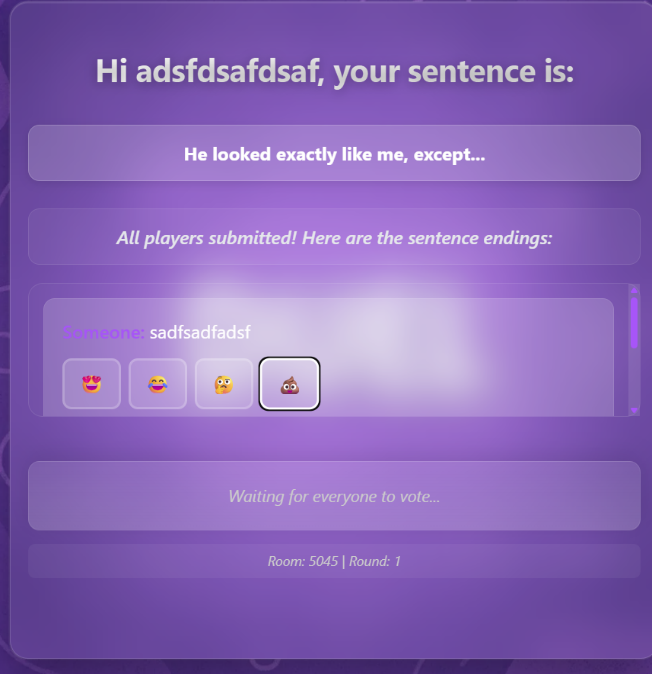


Abbildung von TF-05

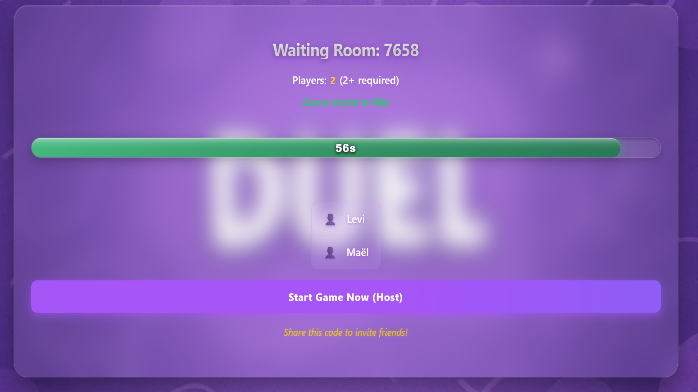


Abbildung von TF-011 und TF-12

# Anhang

## Glossar

Glossar

**API** - Application Programming Interface, Schnittstelle zwischen Frontend und Backend.

**CORS** - Cross-Origin Resource Sharing, Sicherheitsmechanismus für Web-Anfragen.

**DOM** - Document Object Model, Struktur einer HTML-Seite.

**Express.js** - Web-Framework für Node.js.

**PostgreSQL** - Relationale Open-Source-Datenbank..

**React** - JavaScript-Library für Benutzeroberflächen.

**REST** - Representational State Transfer, Architekturstil für Web-Services.

**State** - Zustand einer React-Komponente.

**TypeScript** - Typsichere Erweiterung von JavaScript.

**useState** - React-Hook für State-Management.

## Codeausschnitte

**Codeausschnitte**

### 5.2.1 Emoji-Voting System

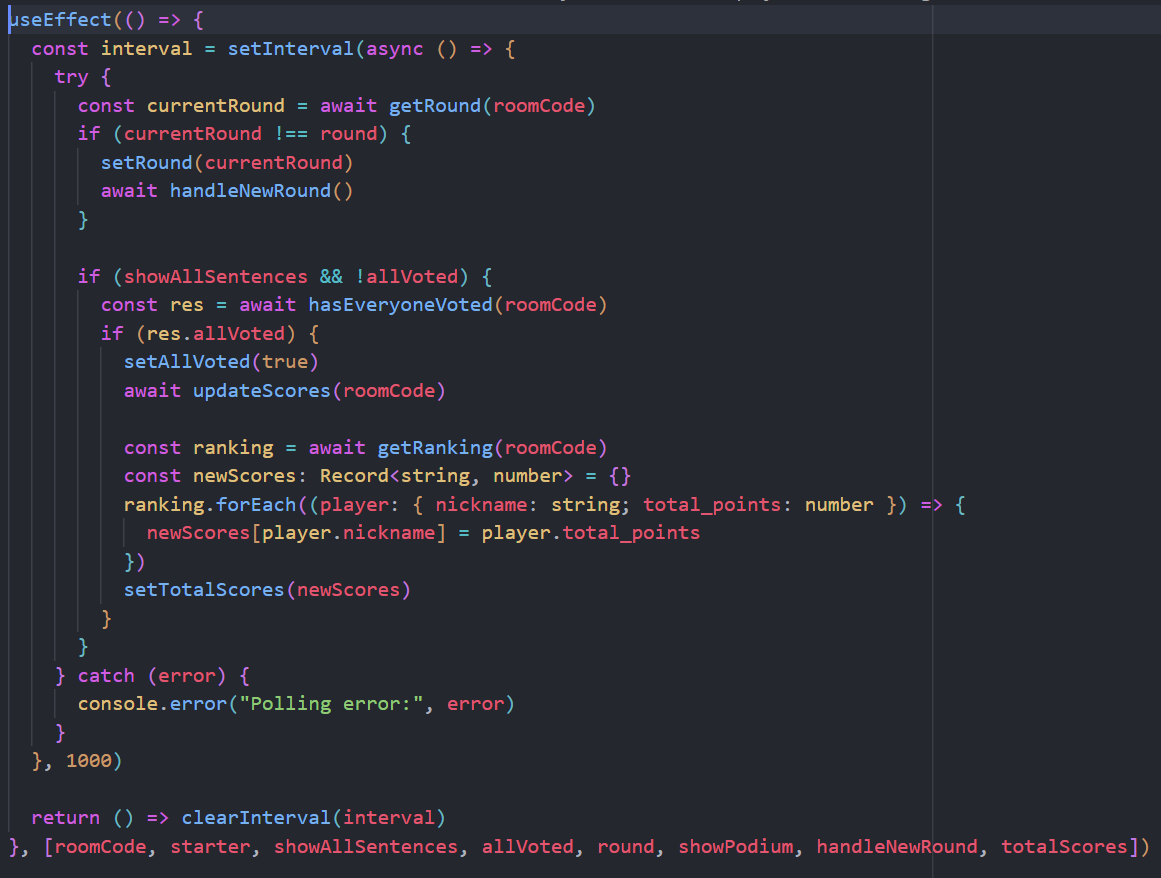
**Emoji-Voting System**



Zeigt das Emoji-Voting-System mit der Punktevergabe und der lokalen State-Verwaltung für Benutzerstimmen.

### 5.2.2 Polling-Mechanismus für Echtzeit-Updates

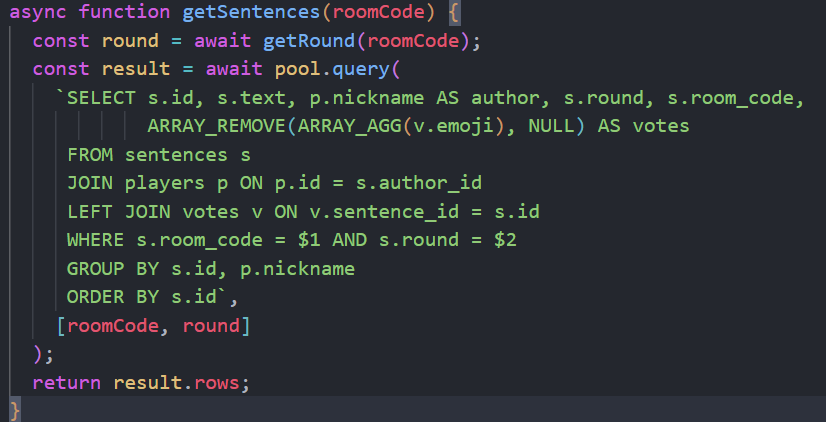
**Polling-Mechanismus für Echtzeit-Updates**



Demonstriert den Polling-Mechanismus, der für Echtzeit-Updates zwischen den Spielern sorgt und verschiedene Spielzustände überwacht.

### 5.2.3 Datenbankabfrage für Sätze mit Votes

**Datenbankabfrage für Sätze mit Votes**



Präsentiert eine komplexe SQL-Abfrage, die Sätze mit ihren zugehörigen Votes und Autoren verknüpft.

## Screenshot-App

Screenshot-App

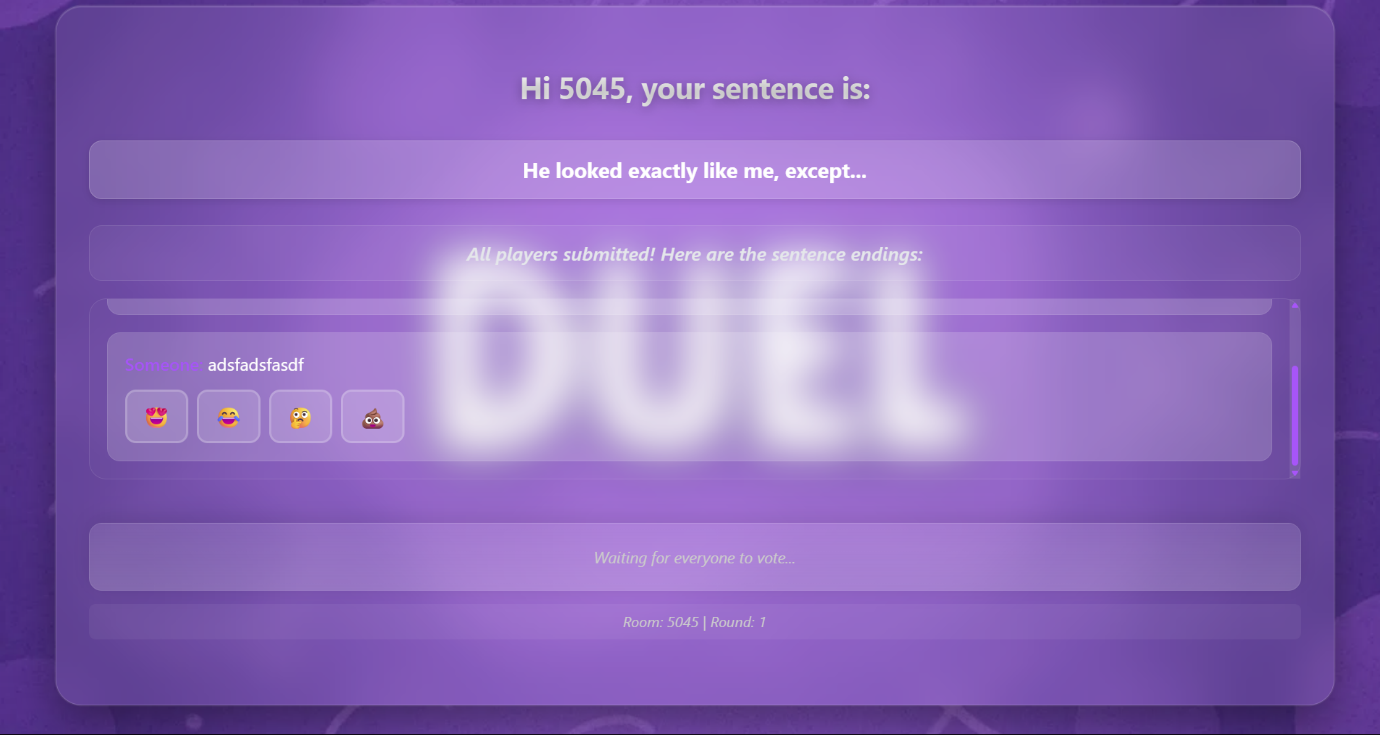


Abbildung von der Game-Screen

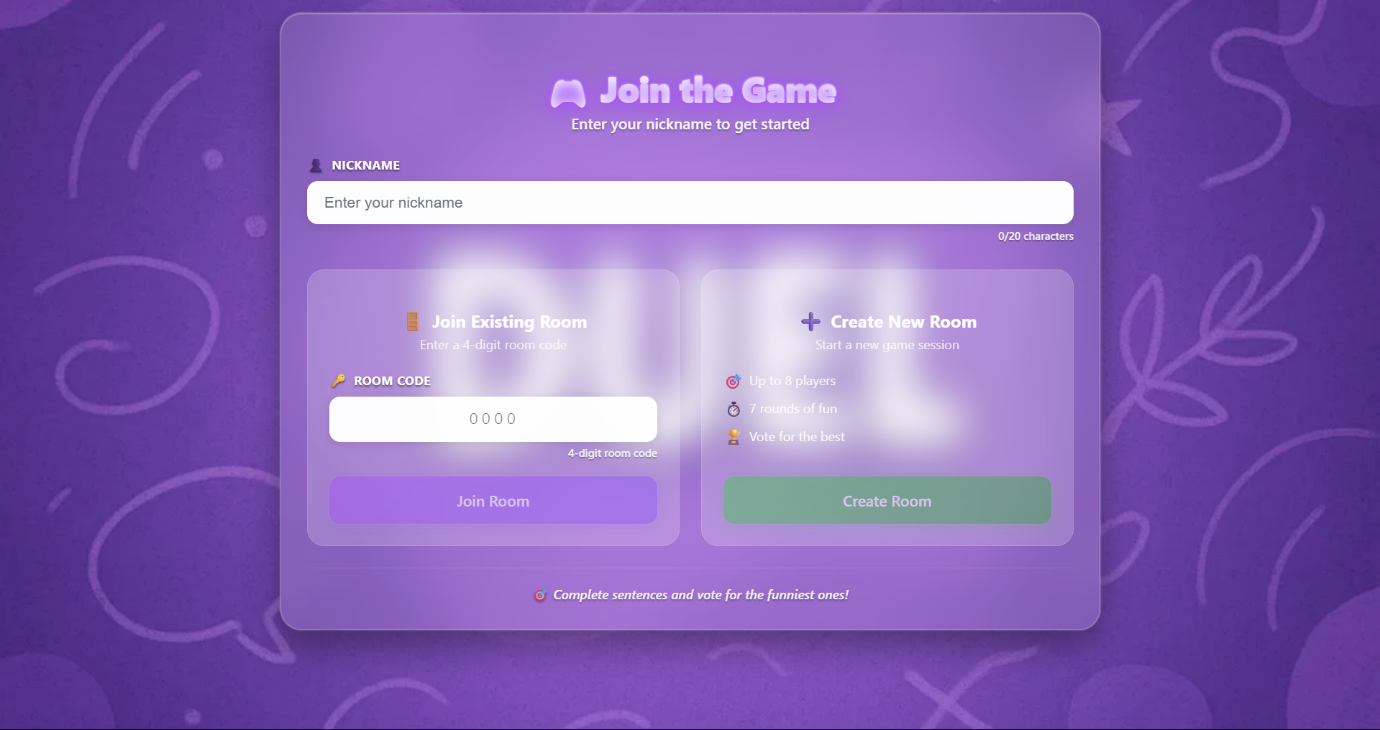


Abbildung von Startseite der App

## KI-Chat-Auszüge

**KI-Chat-Auszüge**

* Hilfe bei TypeScript-Typdefinitionen
* PostgreSQL-Query-Optimierung
* CSS-Flexbox-Layout-Probleme
* React-useEffect-Dependencies
* Deployment-Konfiguration für Render

## Quellen und Literatur

**Quellen und Literatur**

* React Documentation - <https://react.dev/>
* TypeScript Handbook - <https://www.typescriptlang.org/docs/>
* PostgreSQL Documentation - <https://www.postgresql.org/docs/>
* Render Deployment Guide - <https://render.com/docs>
* Chatgpt - <https://chatgpt.com/>
* Claude AI - <https://claude.ai/>
* Deepl-Write - [https://www.deepl.com/de](https://www.deepl.com/de/write?utm_term=&utm_campaign=CH%7CPMAX%7CC%7CGerman&utm_source=google&utm_medium=paid&hsa_acc=1083354268&hsa_cam=22206903038&hsa_grp=&hsa_ad=&hsa_src=x&hsa_tgt=&hsa_kw=&hsa_mt=&hsa_net=adwords&hsa_ver=3&gad_source=1&gad_campaignid=22200549015&gbraid=0AAAAABbqoWAP9h2mnSFFwCB7qf4WGKfxz&gclid=EAIaIQobChMIn8uqquGJjgMVlpCDBx0-jT0ZEAAYASAAEgLhvfD_BwE)
* Rechtschreibpruefung24.de - [https://rechtschreibpruefung24.de](https://rechtschreibpruefung24.de/?gad_source=1&gclid=Cj0KCQiA4fi7BhC5ARIsAEV1YiYz0_Oq_huOYYNcPF7zxZ6UDi9Cb_LM1MU6JUg6yJKSaDRt4I4pfEkaAg9IEALw_wcB)

# Fazit und Reflexion

**Fazit und Reflexion**

Das Projekt „Finish This!“ war für mich ein grosser Erfolg. Ich konnte alle geplanten Funktionen umsetzen und sogar noch Extras wie ein Reset-System und Designverbesserungen einbauen. Die App ist online, funktioniert stabil und macht im Multiplayer-Modus richtig Spass.

Technisch habe ich viel gelernt: TypeScript hat zu weniger Fehlern geführt und mit React konnte ich den Code besser strukturieren. Die Datenbank mit PostgreSQL war zuverlässig und schnell und das Deployment über Render war einfach und praktisch.

Besonders stolz bin ich darauf, dass ich schwierige Dinge wie den Spielstatus (State) gut im Griff hatte und die Kommunikation mit dem Backend reibungslos funktionierte. Auch das Design funktioniert jetzt auf allen Geräten gut.

Ich habe in diesem Projekt sehr viel gelernt – über moderne Web-Technologien, Datenbanken, Backend-Entwicklung und darüber, wie man ein solches Projekt gut plant. Auch die Dokumentation und das Testen waren wichtig, um den Überblick zu behalten.

Für das nächste Mal würde ich früher mit automatisierten Tests beginnen, eventuell auch WebSockets einbauen und mehr auf den Benutzer-Login und die Performance achten.

Insgesamt hat mir das Projekt grossen Spass gemacht. Es war grossartig, eine eigene Idee umzusetzen und zu sehen, wie alles zusammenpasst.