

## Struktur des Projekts

### Überblick

Das Projekt ist in drei Hauptkomponenten unterteilt: Frontend, Backend und Datenbank.

### Frontend

Das Frontend wurde ohne Frameworks entwickelt und besteht aus HTML, CSS und JavaScript. Die Struktur des Frontends folgendermaßen:

- Ordner *“assets”*: Enthält alle Bilder, die im Memory Spiel angezeigt werden.
- Ordner *“data”*: Enthält JSON-Dateien, worin die Bilder nach Themen sortiert sind.
- **HTML-Dateien:**
  1. *startpage.html*: Startseite des Spiels.
  2. *game.html*: Hauptspielseite.
  3. *scores.html*: Seite für die Punktestände.
- **JavaScript-Dateien:**
  1. *startpage.js*: Logik für die Startseite.
  2. *game.js*: Logik für das Hauptspiel, importiert *player.js*, *card.js* und *timer.js*.
  3. *player.js*: Definiert die Spielerklasse und ihre Variablen.
  4. *card.js*: Definiert die Kartenklasse und ihre Funktionen.
  5. *timer.js*: Definiert die Timerklasse und ihre Funktionen.
  6. *score.js*: Logik für die Punktestandsseite.
- **CSS-Dateien:**
  1. *style.css*: Stile für alle Seiten.

### Backend

Das Backend wurde mit Java 17 und Spring Boot 3.3.0 entwickelt. Es umfasst die folgenden Komponenten:

- *BackendApplication*: Startpunkt der Spring Boot Anwendung.
- *CorsConfig*: Konfiguration für Cross-Origin Resource Sharing (CORS).
- *PlayerEntity*: Entity-Klasse für Spieler.
- *PlayerRankDTO*: Data Transfer Object für Spieler-Ranglisten.
- *PlayerRepository*: Schnittstelle für die Datenbankzugriffe.
- *PlayerService*: Geschäftsdienstlogik für Spieleroperationen.
- *PlayerController*: REST-Controller für Spieler-Endpoints.
- *Exception (Folder)*: Enthält Ausnahmebehandlungen.

- **Externe Bibliotheken und Frameworks:**

1. *Lombok*: Reduziert Boilerplate-Code durch automatische Generierung von Getter, Setter und anderen Methoden.
2. *Maven*: Build- und Dependency-Management-Tool.
3. Spring Boot Dependencies:
  - *spring-boot-devtools*: Entwicklertools für Spring Boot.
  - *spring-boot-starter-web*: Starter für Webanwendungen.
  - *spring-boot-starter-data-jpa*: Starter für JPA und Datenbanken.
  - *spring-boot-starter-actuator*: Überwachungs- und Management-Tools für die Produktion.

### **Datenbank**

Die Anwendung verwendet eine *PostgreSQL*-Datenbank, um Spielerinformationen zu speichern und abzurufen.

### **Gitignore**

Die *.gitignore*-Datei ist so konfiguriert, dass sie IDE-spezifische Dateien während Git-Operationen ignoriert.

Diese Struktur ermöglicht eine klare Trennung von Verantwortlichkeiten und erleichtert die Wartung und Weiterentwicklung der Anwendung.