Prototyping Projektdokumentation

Name: Sarangan Nanthakumar

E-Mail: nanthsar@students.zhaw.ch

URL der deployten Anwendung: https://rosterhubnba.netlify.app/

1. Einleitung

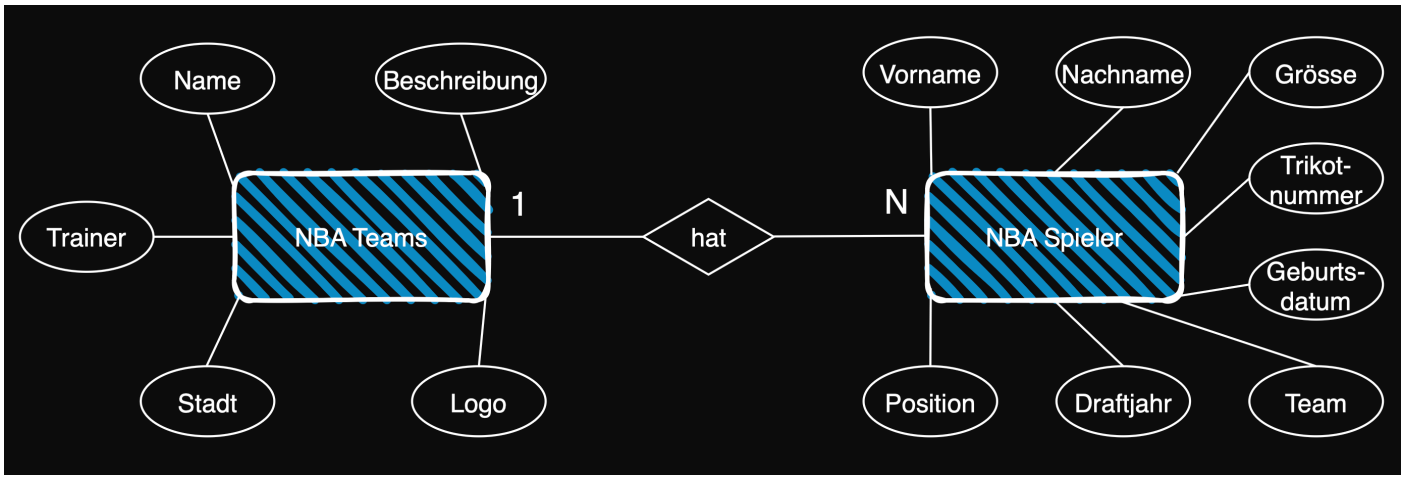
NBA RosterHub ist eine webbasierte Anwendung zur Verwaltung von NBA-Teams und Spielern. Die Plattform bietet eine zentrale Übersicht über alle Teams sowie deren zugehörige Spieler. Nutzerinnen und Nutzer können neue Spieler erstellen, bestehende aktualisieren oder löschen und erhalten detaillierte Informationen zu Position, Grösse, Trikotnummer, Geburtsdatum und Draftjahr.

Ein wesentliches Merkmal der Anwendung sind die intelligenten Verknüpfungen innerhalb der Daten. Beispielsweise wird beim Erstellen oder Bearbeiten eines Spielers das Team nicht manuell eingetippt, sondern aus einer Liste der verfügbaren Teams ausgewählt. Auch die Position eines Spielers ist über ein Dropdown auswählbar, wobei automatisch das passende Bild zur Position angezeigt wird. Auf diese Weise wird die Dateneingabe effizienter und konsistenter.

Die Teams sind mit ihren offiziellen Logos, Städten und Konferenzen dargestellt. Eine benutzerfreundliche Oberfläche ermöglicht es, auf einfache Weise durch die Daten zu navigieren und Zusammenhänge zwischen Teams und Spielern zu erkennen.

Als Erweiterung bietet die Anwendung eine simulierte Draft-Lotterie, welche den NBA-Draft basierend auf realistischen Wahrscheinlichkeiten nachbildet. Sie ergänzt die Hauptfunktionen der Team- und Spielerverwaltung auf interaktive Weise.

2. Datenmodell



Das folgende ER-Diagramm stellt das zugrunde liegende Datenmodell der Anwendung dar. Es besteht aus zwei Hauptentitäten: NBA Teams und NBA Spieler, die in einer 1:N-Beziehung zueinander stehen. Ein Team kann mehrere Spieler haben, während ein Spieler genau einem Team zugeordnet ist.

Die Umsetzung in der Datenbank erfolgt über zwei Collections (nba_teams und nba_player), wobei die Spieler-Einträge jeweils eine Referenz (team_id) auf das zugehörige Team enthalten. Somit kann die 1:N Beziehung verknüpft werden.

3. Beschreibung der Anwendung

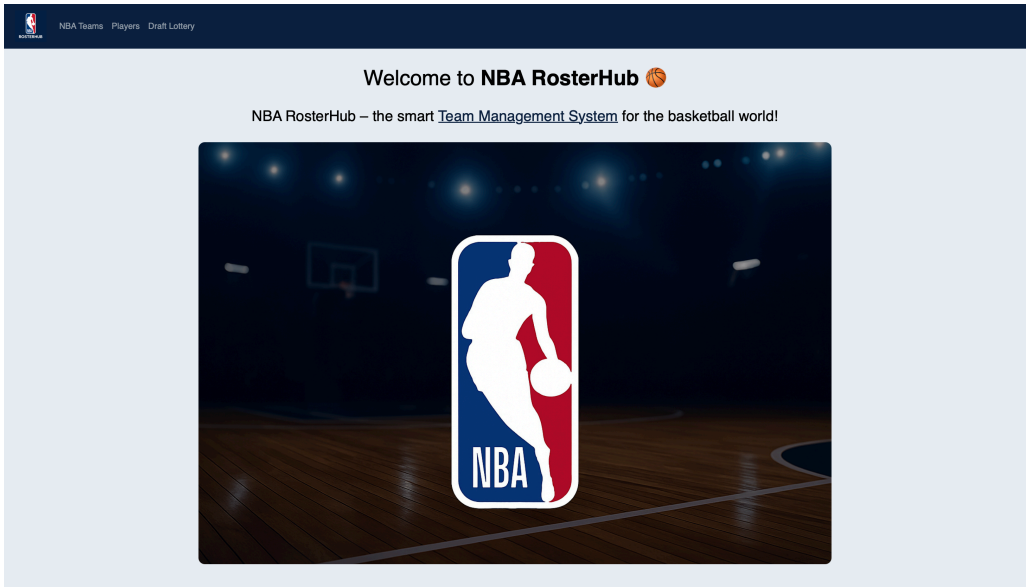
Die Anwendung besteht aus mehreren Seiten, auf denen NBA-Teams, Spieler sowie eine Draft-Lotterie verwaltet und dargestellt werden. Die Seiten bieten neben der Anzeige auch Funktionen zur Erstellung, Bearbeitung und Löschung von Datensätzen. Durch erweiterte Funktionen können die NBA-Spieler direkt mit den Teams verknüpft werden.

Als Hauptfarbe wurde über die gesamte Anwendung ein zur NBA passendes Dunkelblau (#0C2340) verwendet, ergänzt durch ein helles Blau (#e6ecf0) als Hintergrundfarbe. Die Navigation erfolgt über eine Leiste am oberen Rand, die in der Hauptfarbe gehalten ist.

Diese dynamische Darstellung verbessert die Übersichtlichkeit und macht Zusammenhänge unmittelbar sichtbar. Die Gestaltung wurde mit Hilfe von Bootstrap und zusätzlichem CSS verfeinert.

3.1. Startseite

Route: /



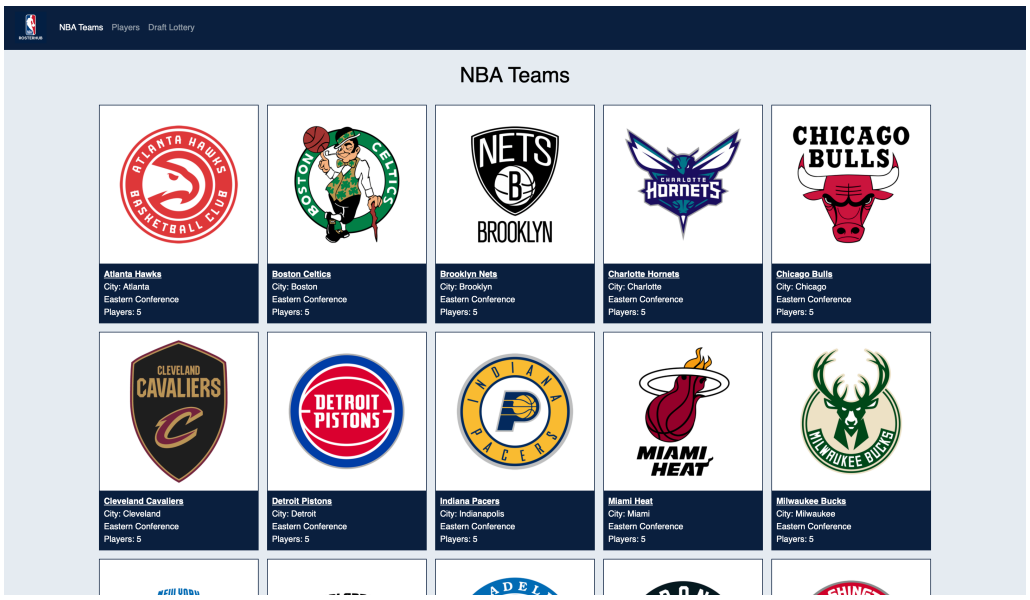
Die Startseite begrüsst die Nutzenden mit dem Titel der Anwendung NBA RosterHub. Die Seite enthält einen kurzen erklärenden Untertitel, der den Zweck der Anwendung beschreibt, sowie ein zentriertes NBA-Bild als visuelles Intro.

Dateien:

- routes /+page.svelte

3.2. NBA Teams

Route: /teams



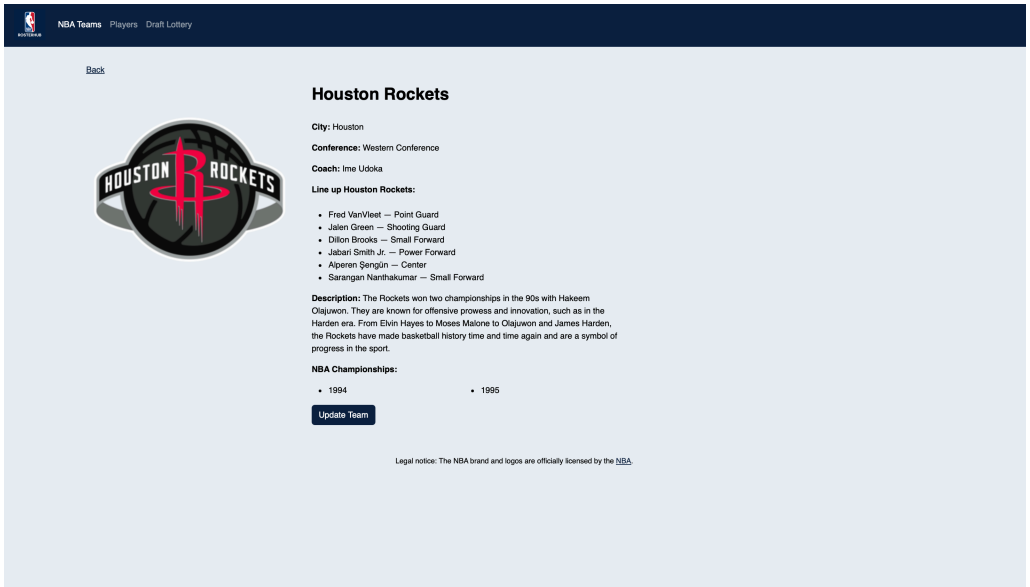
Auf dieser Seite werden alle NBA-Teams übersichtlich dargestellt. Die Logos werden durch Hinterlegen des Pfades aus dem Ordner /static/images/ geholt. Mit Hilfe der Serverfunktion "getTeams" im +page.server.js werden zu jedem Team die wichtigsten Informationen wie Teamname, Stadt und Conference mit angezeigt. Dank erweiterter Funktionen wie "getPlayers" ist zusätzlich auch die Anzahl der zugehörigen Spieler ersichtlich, dargestellt durch den Eintrag Players. Bootstrap mit den Grids helfen bei der Darstellung. Durch Klicken auf den Teamnamen oder das Teamlogo wird die Team-Detailseite /teams/[teams_id] geöffnet.

Dateien:

- lib/db.js
- routes /teams/+page.svelte
- routes /teams/+page.server.js

3.2.1. Team Detailseite

Route: /teams/[teams_id]



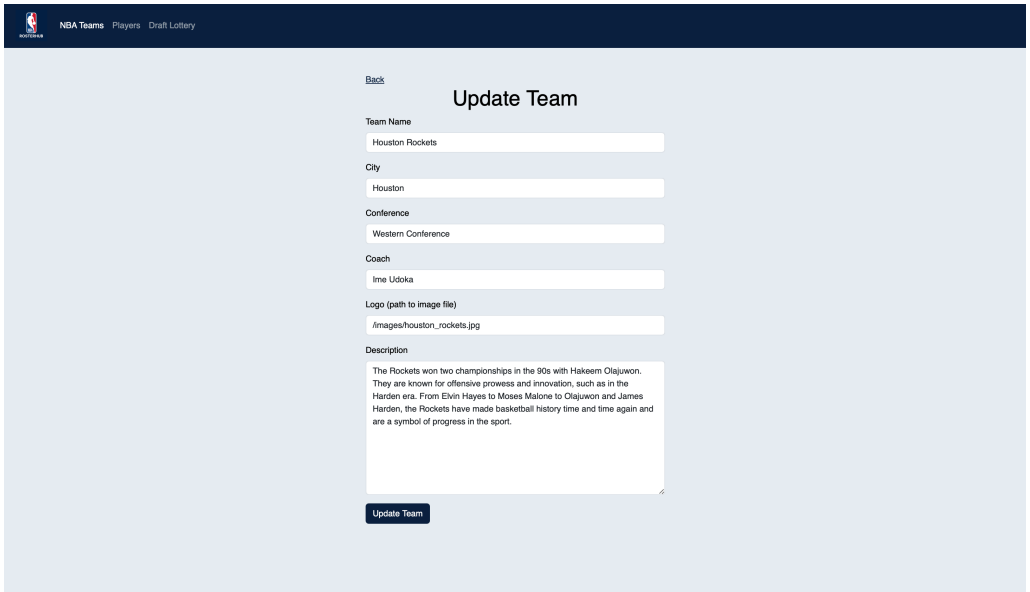
Mithilfe der ID wird die Seite des ausgewählten Teams korrekt aufgerufen. Hier werden die Daten eines bestimmten Teams angezeigt, darunter Logo, Name, Stadt, Conference, Coach, Beschreibung, Titel sowie die zugehörigen Spieler mit Namen und Position. Dank der dynamischen Funktion werden die Spielerinformationen automatisch basierend auf der Teamzuordnung eingeblendet. Mit einem Klick auf "Back" gelangt man zurück zur Teamübersicht. Der Button "Update Team" führt zur Seite /teams/[teams_id]/update, auf der die Teaminformationen bearbeitet werden können.

Dateien:

- lib/db.js
- routes /teams/[teams_id]/+page.svelte
- routes /teams/[teams_id]/+page.server.js

3.2.2. Team Update

Route: /teams/[teams_id]/update



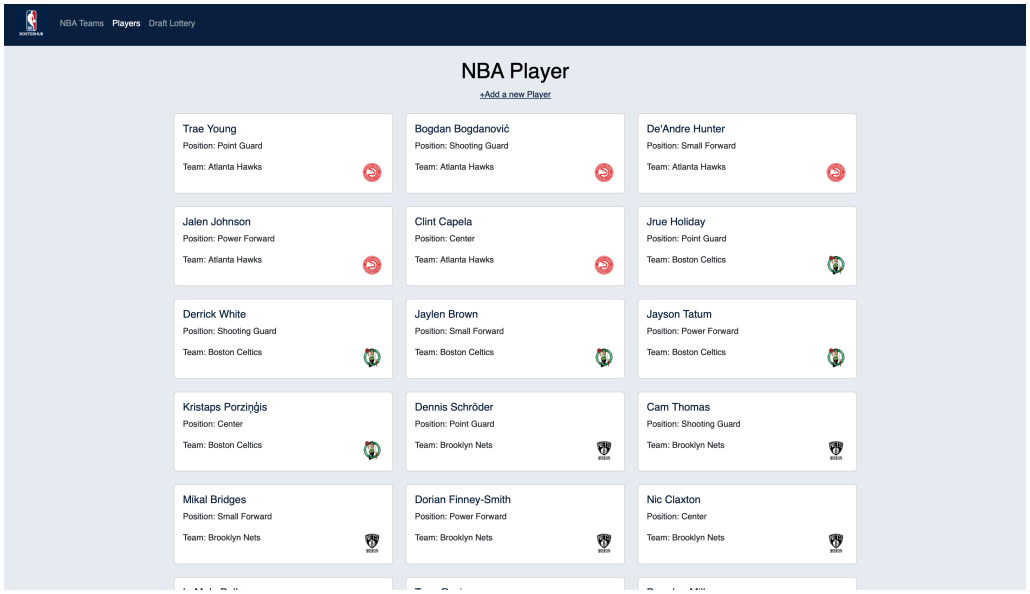
Auf dieser Seite werden die bereits hinterlegten Informationen zum Team dank Funktionen " updateTeam" aus dem Server-Teil automatisch geladen. Die Felder "Team Name", "City" und "Conference" sind schreibgeschützt, da sich diese Angaben in der Regel nicht oder nur selten ändern. Die restlichen Felder "Coach", "Logo" und "Description" können direkt bearbeitet werden. Durch das Klicken auf das Eingabefeld wird eine Aktualisierung möglich. Mit einem Klick auf „Update Team“ werden die neuen Daten gespeichert.

Dateien:

- lib/db.js
- routes /teams/[teams_id]/update/+page.svelte
- routes /teams/[teams_id]/update/+page.server.js

3.3. Players

Route: /players



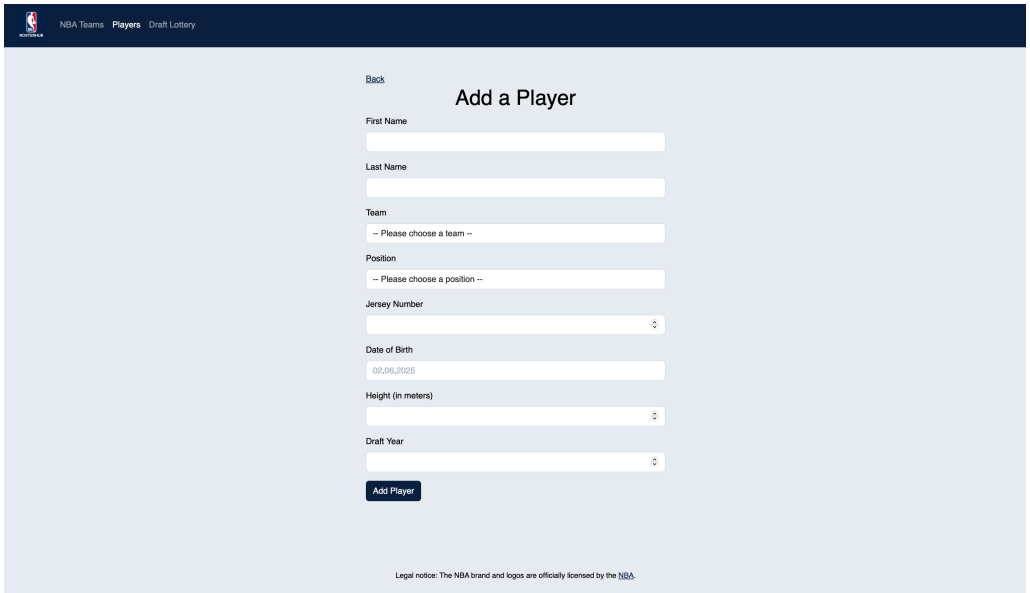
Auf dem Tab "Players" ist die gesamte NBA-Spielerdatenbank sichtbar. Zu jedem Spieler werden die wichtigsten Informationen wie Name, Position und das zugehörige Team mit "getPlayers" und "getTeams" angezeigt. Durch die Verknüpfung mit der Team-Kollektion werden auch Teamname und Logo des jeweiligen Teams dargestellt. Die Darstellung erfolgt in einem gridartigen Layout mit Hilfe von Bootstrap. Durch Klicken auf einen Spielernamen gelangt man zur Detailseite /players/[players_id], auf der weitere Informationen zum jeweiligen Spieler sichtbar sind. Ein Klick auf "+Add a new Player" führt zur Seite /players/create, wo ein neuer Spieler erfasst werden kann.

Dateien:

- lib/db.js
- routes /players/+page.svelte
- routes /players/+page.server.js

3.3.1. Spieler erstellen

Route: /players/create



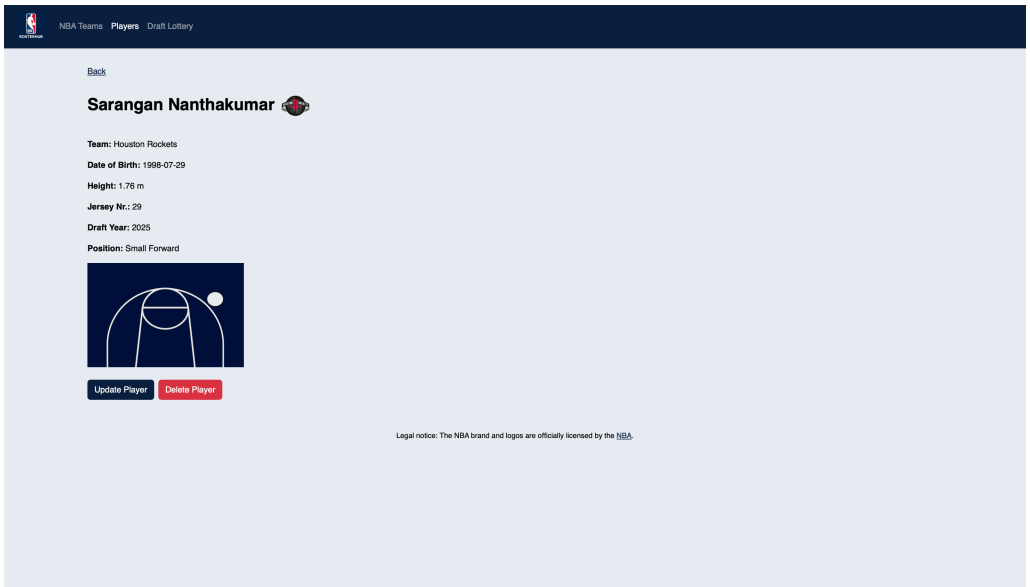
Auf dieser Seite können neue Spieler erfasst werden. Alle Felder sind Pflichtfelder, somit müssen alle Felder ausgefüllt werden. Nach dem Klick auf "Add Player" wird der neue Spieler in die Datenbank mit der Funktion "createPlayer" eingefügt und man wird automatisch zur Übersicht auf /players weitergeleitet.

Dateien:

- lib/db.js
- routes /players/create/+page.svelte
- routes /players/create/+page.server.js

3.3.2. Spieler Detailseite

Route: /players/[players_id]



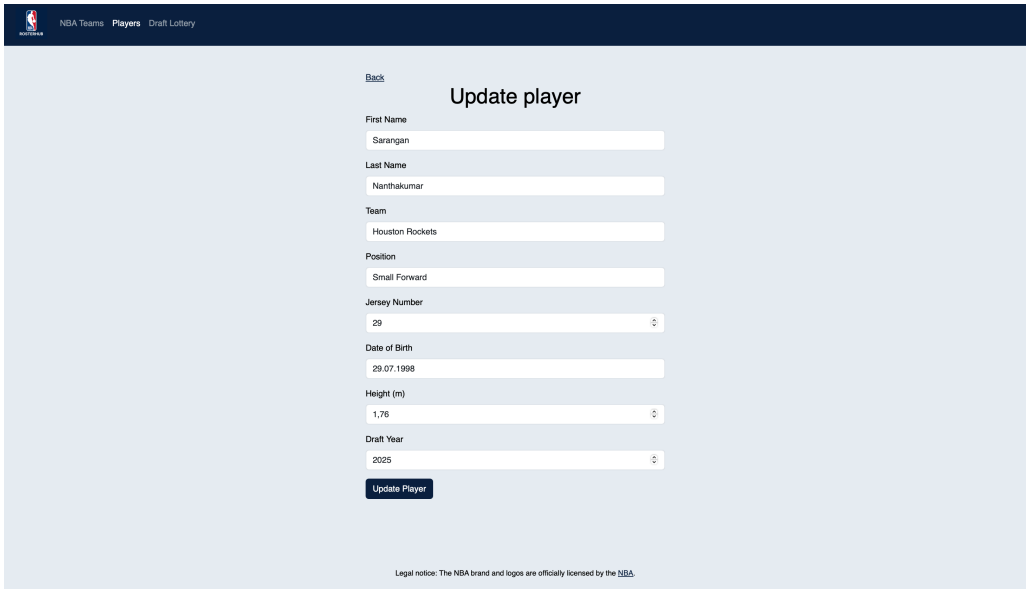
Auf der Detailseite eines Spielers werden durch die übergebene ID alle zugehörigen Informationen aus der Datenbank geladen. Mithilfe von Funktionen "getTeams" und "getPositionImage" werden unter anderem der Teamname, das Teamlogo sowie ein passendes Bild zur Position angezeigt. Mit dem Bild wird die Position des Spielers visuell verdeutlicht. Über die Buttons "Update Player" und "Delete Player" kann der Datensatz entweder bearbeitet oder vollständig gelöscht werden. Das Löschen wird mit der Funktion " deletePlayer" ausgeführt.

Dateien:

- lib/db.js
- routes /players/[players_id]/+page.svelte
- routes /players/[players_id]/+page.server.js

3.3.3. Spieler Update

Route: /players/[players_id]/updateplayer



Hier werden die bereits erfassten Spielerinformationen geladen, und man kann die Daten direkt bearbeiten. Mit einem Klick auf den Button "Update Player" werden die Informationen aktualisiert.

Dateien:

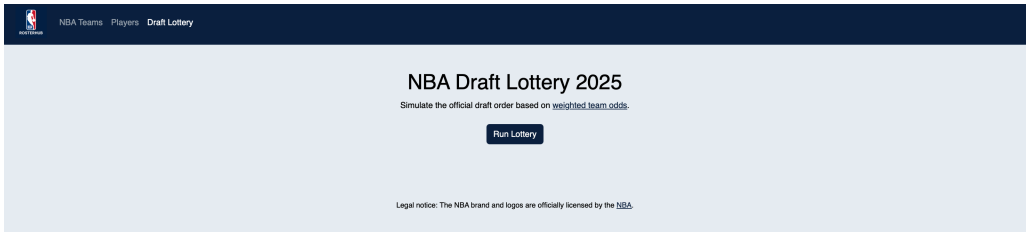
- lib/db.js
- routes /players/[players_id]/updateplayer/+page.svelte
- routes /players/[players_id]/updateplayer/+page.server.js

4. Erweiterungen

Draft Lottery: Simuliert eine gewichtete Ziehung für die Draft-Reihenfolge.
1:N Beziehung: Spieler sind über die team_id mit ihrem Team verknüpft.
Dynamische Updates: Positionsbilder und Dropdown-Auswahlen passen sich automatisch an.
Erweiterung Update- und Deletebuttons: Bestätigungsabfrage vor dem Löschen, Erfolgsmeldung bei Updates.
Erweiterung Navigationsleiste: Aktive Seite wird hervorgehoben, rechtlicher Hinweis eingeblendet.

4.1. Draft Lottery

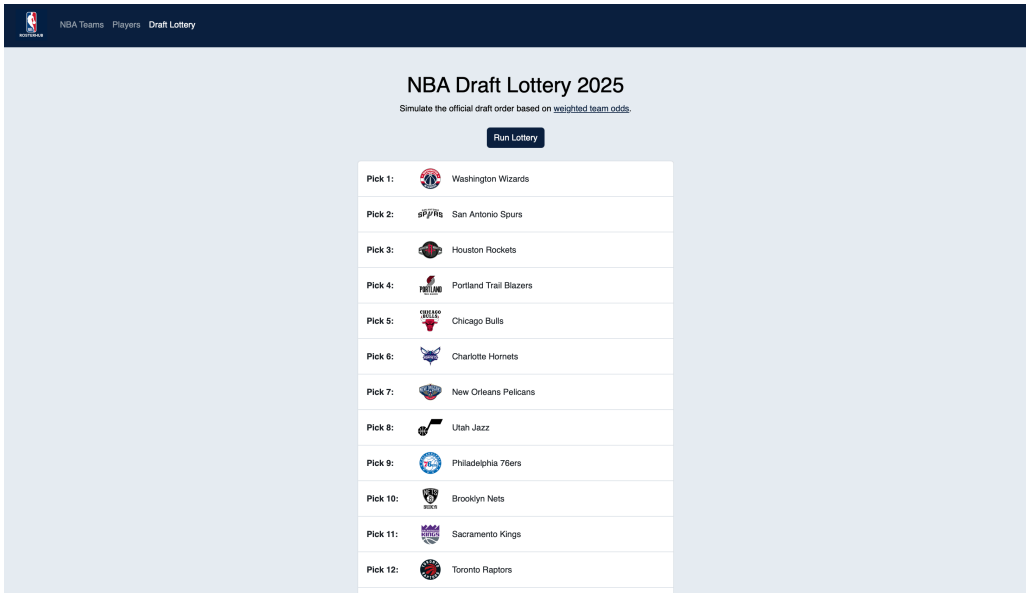
Route: /draft



Auf der Seite /draft wurde als Erweiterung neben Teams und Spielern eine Draft Lottery integriert. Alle NBA-Teams, die die Playoffs verpasst haben, erhalten die Chance auf einen Erstrundenpick. Je schlechter das Abschneiden in der regulären Saison, desto höher ist die Wahrscheinlichkeit auf einen besseren Pick.

Die Wahrscheinlichkeiten wurden mithilfe der Variable "odds" hinterlegt. Über den Link "weighted team odds" gelangt man zur offiziellen NBA-Seite mit weiteren Informationen zu den Wahrscheinlichkeiten.

Durch Klick auf "Run Lottery" wird die Draft-Lotterie gestartet. Die Funktion "runDraft" simuliert eine gewichtete Ziehung. Jedes Team erhält entsprechend seiner odds eine Anzahl an Losen. So hat das Team mit den besten Chancen mehr Lose im Pool als andere Teams.



Mit der Funktion "getTeams" werden Teamname und Logo aus der Datenbank geladen und angezeigt. Die visuelle Darstellung wurde mit Bootstrap realisiert.

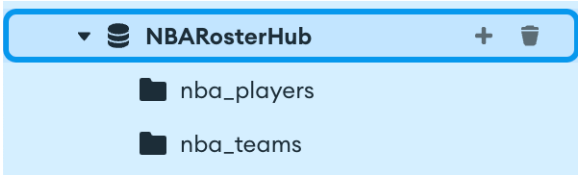
Dateien:

- lib/db.js
- routes /draft/+page.svelte
- routes /draft/+page.server.js

4.2. 1:N Beziehung

Die /players-Seiten zeigt dank erweiterter Funktionen zusätzlich zum Spielerprofil auch das Team-Logo und den Teamnamen an. Auf der /teams-Seite werden neben den allgemeinen Teaminformationen auch automatisch alle zugehörigen Spieler aufgelistet, sowohl beim Hinzufügen eines neuen Spielers als auch beim Teamwechsel. Somit wird die Anzahl Spieler pro Team ebenfalls automatisch aktualisiert.

Aus MongoDB die Kollektionen:



Kollektion nba_teams

Beispieldaten:

```
_id: ObjectId('683a1d902e3fe28bfdf80d85')
team_id : 1
team_name : "Atlanta Hawks"
coach : "Quin Snyder"
city : "Atlanta"
conference : "Eastern Conference"
description : "The Atlanta Hawks were originally founded in 1946 as the Buffalo Bison…"
logo : "/images/atlanta_hawks.jpg"
titel : Array (1)
  0: 1958
```

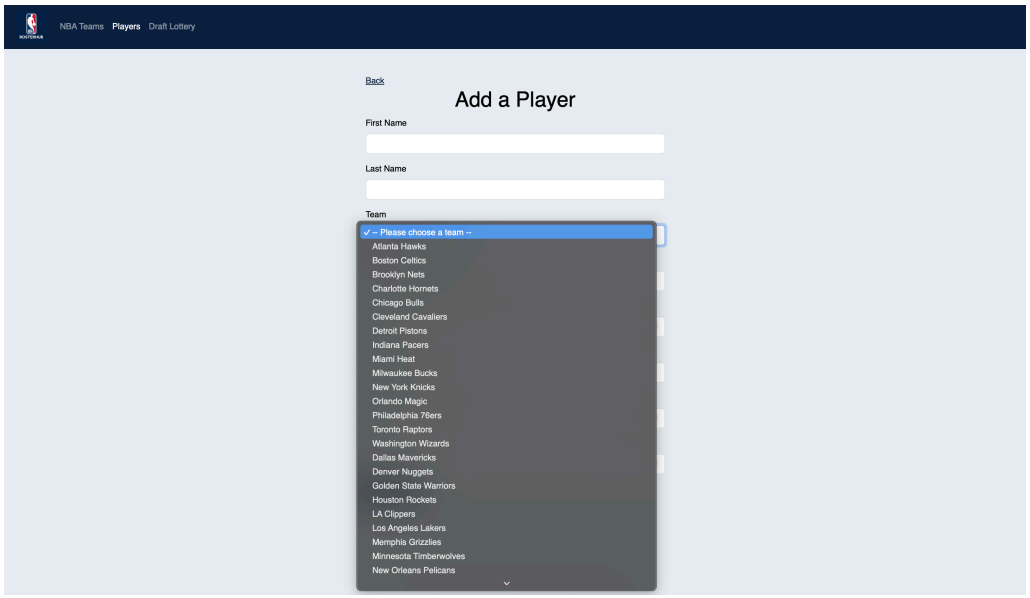
Kollektion nba_players
Beispieldaten:

```
_id: ObjectId('6839dd6b2e3fe28bfdf80cbb')
player_id : 1
first_name : "Trae"
last_name : "Young"
team_id : 1
position : "Point Guard"
trikotnummer : 11
geburtsdatum : "1998-09-19"
größe : 1.85
draft_jahr : 2018
```

Die Verknüpfung basiert auf der in beiden Kollektionen (nba_teams und nba_players) hinterlegten team_id. Mithilfe der Funktionen "getPlayersTeam" und "getPlayersByTeam" werden die Team-IDs verglichen und so die Daten korrekt miteinander verknüpft.

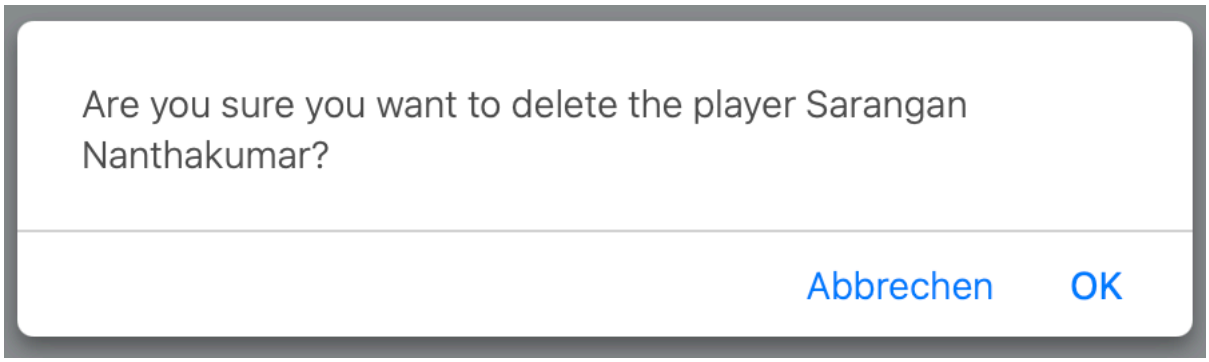
4.3. Dynamische Updates

Als Erweiterung zum Positionsbild wird das Bild dynamisch anhand der Spielerposition gewählt (z. B. Center → center.png). Wenn die Position eines Spielers geändert wird, aktualisiert sich das Bild automatisch. Dies wird durch die Funktion "getPositionImage" ermöglicht.



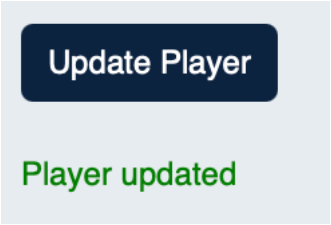
Zusätzlich wurden beim Erstellen und Aktualisieren von Spielern erweiterte Eingabefelder integriert. Die Auswahl von Team und Position erfolgt benutzerfreundlich über Dropdowns. Mithilfe von Funktionen und der each-Schleife werden nur die Teams angezeigt, die in der nba_teams-Kollektion vorhanden sind. Eine doppelte Eingabe ist nicht nötig.

4.4. Erweiterung Update- und Deletebuttons



Vor dem Löschen eines Spielers wird durch die Funktion "confirmDelete" eine Bestätigungsabfrage eingeblendet. Diese schützt vor unbeabsichtigtem Löschen und verbessert die Benutzerfreundlichkeit der Anwendung.

Zum Update wird eine Meldung eines erfolgreichen Updates angezeigt.



4.5. Erweiterung Navigationsleiste

Mit Hilfe der Datei +layout.svelte wird die aktuell aktive Seite visuell hervorgehoben. Dadurch erkennt der Benutzer auf einen Blick, auf welchem Tab er sich befindet. Zudem wurde in +layout.svelte definiert, dass auf allen Seiten die urheberrechtlichen Hinweise angezeigt werden.