

Kosárlabdás Applikáció Dokumentáció

Az általam készített kosárlabdás applikáció egy Express.js alapú webalkalmazás, amely különböző funkciókat biztosít a felhasználóknak, mint például fórum használata, jegyvásárlás, profilkezelés stb. Az alábbiakban összefoglalom, hogyan működik az applikáció, hogyan indul el és milyen főbb részei vannak.

1. Indítás és Alapkonfiguráció

1. **Szerver indítása:** Az alkalmazás az **app.js** fájlban definiált szerverrel indul el. Az **app.listen(3000, ...)** sor elindítja a szerveret a 3000-es porton.

```
app.listen(3000, () => { console.log('Server is running on http://localhost:3000'); });
```

2. **Middleware-ek konfigurálása:**

- **Session kezelés:** A felhasználói munkamenetek kezelése érdekében az **express-session** middleware van konfigurálva.
- **Body Parser:** A bejövő kérések testének elemzéséhez a **body-parser** middleware van használva.
- **Statikus fájlok kezelése:** Az **express.static** segítségével a **public** mappa tartalma statikus fájlként szolgáltatva.

2. Adatbázis Kapcsolat

Az adatbázis kapcsolat a **config/db.js** fájlban van definiálva, amely létrehozza a MySQL kapcsolódási objektumot és csatlakozik a MySQL adatbázishoz.

3. Routerek és Kontrollerek

A különböző funkciók különálló router fájlokban vannak definiálva, amelyek a megfelelő controller függvényeket hívják meg.

3.1. Autentikáció

- **Regisztráció (/register):**
 - A felhasználó kitölti a regisztrációs űrlapot.
 - A **register** controller függvény (**authController.js**) hash-eli a jelszót és elmenti az új felhasználót az adatbázisba.
 - Sikeres regisztráció után a felhasználó a bejelentkezési oldalra kerül átirányításra.
- **Bejelentkezés (/login):**
 - A felhasználó megadja az email címét és jelszavát.
 - A **login** controller függvény ellenőrzi a felhasználó adatait, és ha helyesek, beállítja a munkamenetet.
 - Sikeres bejelentkezés után a felhasználó a **dashboard** oldalra kerül átirányításra.
- **Kijelentkezés (/logout):**

- A **logout** kontroller függvény megszünteti a munkamenetet és átirányítja a felhasználót a bejelentkezési oldalra.

3.2. Fórum

- **Fórumbejegyzések listázása (/forum):**
 - A felhasználó megtekintheti a fórum főoldalát, ahol az összes fórumbejegyzés megjelenik.
 - A **listPosts** kontroller függvény lekéri az összes bejegyzést és azok hozzászólásait, majd rendereli a **forum.ejs** nézetet.
- **Új bejegyzés létrehozása (/forum):**
 - A bejelentkezett felhasználó új fórumbejegyzést hozhat létre.
 - A **createPost** kontroller függvény elmenti az új bejegyzést az adatbázisba, majd visszairányítja a felhasználót a fórum főoldalára.
- **Bejegyzés részletei (/forum/:id):**
 - A felhasználó egy konkrét bejegyzésre kattintva megtekintheti annak részleteit és hozzászólásait.
 - A **listPosts** kontroller függvény lekéri a bejegyzés részleteit és hozzászólásait, majd rendereli a **post_detail.ejs** nézetet.
- **Hozzászólás hozzáadása (/forum/addComment/:postId):**
 - A felhasználó hozzászólást írhat egy bejegyzéshez.
 - Az **addComment** kontroller függvény elmenti a hozzászólást az adatbázisba, majd visszairányítja a felhasználót a bejegyzés részletei oldalra.
- **Hozzászólás törlése (/forum/deleteComment/:commentId):**
 - Az adminisztrátorok törölhetik a hozzászólásokat.
 - A **deleteComment** kontroller függvény törli a hozzászólást az adatbázisból, majd visszairányítja a felhasználót a fórum főoldalára.

3.3. Meccsek

- **Meccsek listázása (/games):**
 - A **listGames** kontroller függvény lekéri az összes meccset az adatbázisból és megjeleníti a **game.ejs** nézetben.

3.4. Jegyek

- **Jegyek listázása (/tickets):**
 - A felhasználó megtekintheti a megvásárolt jegyeit.

- A **listTickets** kontroller függvény lekéri a felhasználó jegyeit az adatbázisból és megjeleníti a **ticket.ejs** nézetben.
- **Jegyvásárlás oldal megjelenítése (/tickets/buy):**
 - A felhasználó kiválaszthat egy meccset és egy ülőhelyet, majd jegyet vásárolhat.
 - A **showBuyTicketPage** kontroller függvény lekéri az összes meccset és ülőhelyet, majd megjeleníti a **buy_ticket.ejs** nézetet.
- **Jegyvásárlás (/tickets/buy):**
 - A felhasználó kiválasztja a meccset és az ülőhelyet, majd megvásárolja a jegyet.
 - A **buyTicket** kontroller függvény elmenti a jegyvásárlást az adatbázisba, majd visszairányítja a felhasználót a jegyek oldalára.

Az alkalmazás az alábbi funkciókat valósítja meg az adott feladat követelményei alapján:

1. Felhasználó és szesszió kezelés:

- **Bejelentkezés és regisztráció:**
 - A felhasználók regisztrálhatnak az alkalmazásba, ahol a jelszavak titkosítva vannak tárolva a biztonság érdekében.
 - Bejelentkezéskor a felhasználók szessziója létrejön, és a munkamenet során végig bejelentkezve maradnak.
 - Kijelentkezés funkció is megvalósult, amely megszünteti a felhasználó szesszióját.
- **Szerepkörök kezelése:**
 - Az alkalmazás megkülönbözteti az adminisztrátorokat és a normál felhasználókat, különböző jogosultságokkal.
 - Adminisztrátorok törölhetik a fórum bejegyzéseket és hozzászólásokat.

2. Megjelenítés:

- **Fórum:**
 - A felhasználók megtekinthetik az összes fórum bejegyzést egy listában.
 - Kattintásra részletesen megjelennek a bejegyzések és a hozzászólások.
- **Meccsek listázása:**
 - A felhasználók megtekinthetik a közelgő meccsek listáját.
- **Jegyek:**
 - A felhasználók megtekinthetik a megvásárolt jegyeiket egy listában.

3. Keresés:

- **Fórum kereső:**
 - A felhasználók kereshetnek a fórum bejegyzések között cím és tartalom alapján.

4. Űrlapon keresztüli adat beküldés:

- **Új fórum bejegyzés létrehozása:**
 - A felhasználók új fórum bejegyzéseket hozhatnak létre egy űrlapon keresztül.
- **Hozzászólás hozzáadása:**
 - A felhasználók hozzászólásokat adhatnak hozzá a fórum bejegyzésekhez.
- **Jegyvásárlás:**
 - A felhasználók jegyeket vásárolhatnak a meccsre egy űrlapon keresztül.
- **Adatok törlése:**
 - Az adminisztrátorok törölhetik a hozzászólásokat és a bejegyzéseket.

5. Állomány feltöltés:

- **Profilkép feltöltés:**
 - A felhasználók profilképet tölthetnek fel, amely a felhasználói profil oldalon jelenik meg.

6. Kliens oldali hálózati kérés alkalmazása:

- HTTP POST kérések használata: Az űrlapok standard HTTP POST kérésekkel történő beküldése, MIME típusok kezelése (application/x-www-form-urlencoded és multipart/form-data).

Megvalósítás Részletezése

1. Felhasználó és szesszió kezelés:

- A felhasználók regisztrálhatnak a nevükkel, e-mail címükkel és jelszavukkal. A jelszavak titkosítva vannak tárolva a **bcryptjs** használatával.
- A felhasználók bejelentkezhetnek, és a munkamenet során végig bejelentkezve maradnak. A bejelentkezett felhasználók szessziója egy cookie-ban van tárolva.
- Az adminisztrátorok és normál felhasználók különböző jogosultságokkal rendelkeznek. Az adminisztrátorok például törölhetnek fórum bejegyzéseket és hozzászólásokat.

2. Megjelenítés:

- A fórum oldal listázza az összes fórum bejegyzést, és lehetővé teszi a felhasználók számára, hogy megtekintsék a részletes tartalmakat és hozzászólásokat.

- A meccsek oldal listázza a közelgő meccseket, beleértve a hazai és vendég csapatokat, valamint a meccs időpontját és helyszínét.
- A jegyek oldal listázza a felhasználó által megvásárolt jegyeket, beleértve a meccs részleteit és a jegy árát.

3. Keresés:

- A fórum oldalon keresési funkció érhető el, amely lehetővé teszi a felhasználók számára, hogy cím és tartalom alapján keressenek a bejegyzések között.

4. Űrlapon keresztüli adat beküldés:

- A felhasználók új fórum bejegyzéseket hozhatnak létre egy űrlap segítségével. Az űrlap beküldése után a bejegyzés megjelenik a fórum oldalon.
- A felhasználók hozzászólásokat adhatnak hozzá a fórum bejegyzésekhez. Az űrlap beküldése után a hozzászólás megjelenik a bejegyzés alatt.
- A felhasználók jegyeket vásárolhatnak a meccsre egy űrlapon keresztül. Az űrlap beküldése után a jegy megjelenik a felhasználó jegyei között.
- Az adminisztrátorok törölhetik a fórum bejegyzéseket és hozzászólásokat.

5. Állomány feltöltés:

- A felhasználók profilképet tölthetnek fel, amely a felhasználói profil oldalon jelenik meg. A profilkép feltöltése egy űrlapon keresztül történik, és a kép a szerveren kerül tárolásra.

Jövőbeli Fejlesztési Lehetőségek

1. Kliens oldali hálózati kérés alkalmazása:

- **Fetch API használata:**
 - Az űrlapok beküldéséhez és adatok frissítéséhez a Fetch API használata lehetővé tenné a dinamikusabb és reszponzívabb felhasználói élményt.
 - Keresési eredmények dinamikus frissítése a Fetch API segítségével anélkül, hogy az egész oldalt újratöltenénk.

2. További adatbázis műveletek:

- **Bővített jogosultságkezelés:**
 - Finomabb szerepkörök és jogosultságok megvalósítása, például moderátorok, VIP felhasználók stb.

3. További megjelenítési és keresési funkciók:

- **Részletesebb meccs információk:**
 - Részletesebb adatok megjelenítése a meccsekhez kapcsolódóan, például csapat statisztikák, helyszíni információk stb.

- **Szűrés és rendezés:**
 - További szűrési és rendezési lehetőségek megvalósítása a fórum bejegyzések, meccsek és jegyek listázásánál.

4. Felhasználói élmény javítása:

- **Reszponzív design:**
 - A felhasználói felület továbbfejlesztése a különböző eszközökön való jobb megjelenés érdekében.
- **UI/UX javítások:**
 - Felhasználóbarátabb navigáció, jobb értesítési rendszerek, stb.

Összegzés

Az applikáció jelenlegi állapotában számos fontos funkciót kínál a felhasználók számára, beleértve a fórum használatát, jegyvásárlást és profilkezelést. A fent említett további fejlesztési lehetőségekkel az alkalmazás még teljesebb és felhasználóbarátabb lehet.