Technikai dokumentáció

Betropolis Casino

Csoporttagok:

**Szabó Tamás:** projektvezető, fejlesztő

**Rácz Ákos Csaba:** fejlesztő

Közös munka, átláthatóság:

A hatékony, akadálymentes munka érdekében a folyamatos élő és digitális (pl. Discord, Messenger alkalmazások) kommunikáció mellett a Trello projektmenedzsment szoftvert használtuk feladataink átlátásához.

*(Képek a projekt ,,docs” mappáján belül a ,,Trello” mappában találhatóak.)*

1. Bevezetés

1.1 Cél

Ez a technikai dokumentáció a weboldal fejlesztői és karbantartói számára készült, hogy megértsék a rendszer működését és karbantarthatóságát.

1.2 Közönség

Ez a dokumentáció a fejlesztők, rendszermérnökök és karbantartók számára készült.

2. Architektúra és Technológiák

2.1.1. Felhasználói Interfész (UI)

A felhasználói interfész réteg felelős a weboldal vizuális megjelenítéséért és az interakcióért a felhasználóval. A weboldalunk reszponzív, és a következő technológiákat alkalmazza:

* **JS Framework:** A projekthez egy modernebb JavaScript keretrendszert használtunk a Full-Stack keretrendszerén keresztül, a ***Svelte JS***-t a ***SvelteKit***-ben.
* **Stílus:** Oldalunk dizájnjához és reszponzívitásához a ***TailwindCSS*** keretrendszert használtuk.
* **Interaktivitás, reaktivitás:** Az oldal interaktivitását és reaktivitását a ***Svelte*** technológiáinak segítségével valósítottuk meg ***TypeScript***-ben.

2.1.2. Alkalmazás Logika

Az alkalmazás logika rétegének része a frontend és a backend. Az projekt elkészítése során törekedtünk a tiszta kód elvének betartására.

* Frontend Réteg: A frontend alkalmazás a felhasználói interfészért felelős, valamint a felhasználói oldalon történő ellenőrzésekért, adatok feldolgozásáét, illetve kaszinónk játékainak futtatásáért felelős.
* Backend Réteg: A backend réteg felelős a felhasználói kérések feldolgozásáért, az adatbázishoz való hozzáférésért és adatok fel- és leküldéséért.

2.1.3. Adatbázis

Az adatbázis réteg tárolja és kezeli az alkalmazás által használt adatokat.

* Típus és Struktúra: Az alkalmazás adatbázisához MySQL relációs adatbázisrendszert használunk.
* ORM: Az adatbázis és az alkalmazás közötti kommunikációhoz a mysql2 (npm) nevű könyvtárat alkalmaztuk.

2.2 Technikai specifikáció

Frontend:

* HTML v5
* CSS v3
* ***TailwindCSS*** v3.4.9
* ***JavaScript*** (ES6+), ***TypeScript*** v5
* ***Svelte JS*** v5 (Full-Stack)
* ***Svelte i18n*** (npm)

Backend:

* ***Sveltekit*** v3 (Full-Stack)
* ***JavaScript, TypeScript***
* mysql2 (npm)

Adatbázis: MySQL

Futtatási környezet:

* ***Node JS*** v22.10+ LTS (Kódnév: Jod)
* ***npm v10***
* Vite v5
* XAMMP (MySQL adatbázishoz)

3. Adatbázis

Az MySQL adatbázisunk 7 táblát tartalmaz:

* users: Felhasználók adatai.
* user\_session: Felhasználói munkamenetek.
* statistics: Felhasználói statisztikák a játékokból.
* game: Játékainkról tárolt információ.
* bonus: Főoldal bónuszjátékának táblája.
* friends: Baráti kapcsolatok.
* friend\_requests: Barátkérések.

Az adatbázis szerkezetéről UML diagram is készült, ez a projekt *,,docs”* mappájában a *,,db”* könyvtárban található a forrásfájl mellett.

4. Frontend

A ***SvelteKit*** a ***Svelte JS*** Full-Stack keretrendszere, ebből adódóan a szerver és kliens oldali kód is ***JavaScript/Typescript/Svelte*** szintaxissal.

Az alábbi formátumú fájlok futnak le a kliensen:

* +layout.svelte: Az oldal szerkezetét adja meg. Több oldal is használhatja ugyanazt a szerkezetfájlt.
* +page.svelte: Az oldal. A <script> tag-ben található az oldal logikája (***JS/TS/Svelte***), a <style> tag-ben az oldalhoz tartozó lokális stílusok (CSS), illetve az oldal tartalma (HTML).
* +layout.svelte: A szerkezethez tartozó kliens oldali fájl. Az oldal nyelvének beállításához használtuk.

A ***SvelteKit***ben a navigálás úgynevezett *,,Nested Routing”* formában történik, így a weboldal route-jai *a projekt/src/routes* mappájának a szerkezete alapján működnek.

5. Backend

5.1 LoaderS

Betöltőket a +page.server.ts (PageServerLoad) és a +layout.server.ts (LayoutServerLoad) fájlokban használunk. Adatbázisból való adat lekérésére az oldal betöltésekor, illetve route-ok levédésére alkalmaztuk.

5.2 API

A projekt API-ja a *projekt/src/routes/api* mappában található. A mappákon belül a +server.ts fájlok tartalmazzák az API kódját.

* /friend-requests: Barátkérések küldése, lekérése és törölté, inaktívvá változtatása.
* /friends: Barát adatok lekérése, barátság elfogadása és törölté. inaktívvá változtatása.
* /profile: Profil adatainak módosítása a profil oldalon.
* /funds: Egyenleg feltöltése és visszaigénylése a központban.
* /balance-update: Az egyenleg lekérése és frissítése a játékok játszásakor.
* /bonus: A főoldalon lévő bónuszjáték eredményének felvezetése az adatbázisba.
* /sign-up és /sign-in: Felhasználó regisztrálása, beléptetése.
* /sign-out és /sign-out-all: Kijelentkezteti a felhasználót a jelenlegi vagy az összes létező munkamenetből.

6. Felhasznált Tartalmak

A könyvtár, ami ezeket tartalmazza *a projekt/src/lib*.

A projekthez felhasznált képek egy része generatív mesterséges intelligencia segítségével készült.

Az ikonok, egyéb képek és videórészletek a Flaticons-ról (flaticon.com) és a Pixabay-ről (pixabay.com) származnak.

A felhasznált betűtípusok az 1001 fonts (1001fonts.com) oldalról vannak: Borgens Burlesque, Dracutaz, Orange Knight

7. Tesztelés

A weboldal tesztelése manuálisan valósult meg. Néhány tesztesetről részletesebben lehet olvasni a Tesztesetek dokumentumunkban.