BGSzC Pestszentlőrinci Közgazdasági és Informatikai Szakgimnáziuma

1184 Budapest Hengersor 34.

**Partyez**

***The party finder***

Konzulens tanár: Készítette:

név: -Szekrényes Gábor Saját név: -Szepesi Szilárd

-Selever Viktor

* **Bevezetés**

Kezdetben nagyon sokat gondolkodtunk, hogy mit lehet még kitalálni, ami hasznos is és egyben nincs nagy piaci ellenfele. Sok ötletelés után arra jutottunk, hogy egy

bulikat összegyűjtő weboldalt fogunk megalkotni.

* **Feladat leírás**

Egy olyan weboldal létrehozását tűztük ki feladatként magunknak a vizsga keretein belül, ami felgyorsítja a partik megtalálását a fiatalabb korosztálynak, kényelmes, egyszerű és mindenek elött gyors (angol nyelven van, hogy nagyobb réteget fedjen le).

* **A felhasznált ismeretek**

**Az alábbi ismereteinket használtuk fel a munkánk során:**

Azokat az ismereteinket alkalmaztunk, amelyek magukba foglalják a szoftverfejlesztés

alapelveit, tesztelési módszereit, hibakeresést és automatizált tesztelési

eszközök használatát.

• Felhasználtuk a HTML, JavaScript beépített segéd ait valamint előző

projektjeinkből az ismereteinket.

• Valamint MySQL workbench-ben az adatbázis ismereteinket a munkáink során.

* **A felhasznált szoftverek**

**Az alábbi szoftvereket használtuk fel a munkánk során:**

• Visual Studio Code

• XAMPP, PhpMyAdmin

* Böngésző (pl. Chrome, Brave,Edge)
* Grok (ai)
* **Felhasználói dokumentáció**
* **A program általános specifikációja**

Az feladatunk egy bulik, keresésére alkalmas könnyen használható oldal itt pár kattintás és egy regisztráció után (életkor) jelentkezhetünk a közel jövőbeli partikra.

Az oldalunkon regisztrálni, és bejelentkezni kell, ehhez megalkottunk egy letisztult felületet

Regisztrációkor a jelszó el mentődik az adatbázisunkba jelszó típussal.

Lehetőséged van egy helyen megnézni a legújabb bulikat és jelentkezni rájuk könnyedén.

* Rendszerkövetelmények

**Hardver követelmények:**

* Projektünket első sorban Windows rendszerekre terveztük, ezért a minimális követelmény az alábbi:
* Processzor: 1 gigahertzes (GHz) vagy gyorsabb processzor vagy SoC
* Memória: 1 gigabájt (GB) 32 bites rendszerhez vagy 2 GB 64 bites operációs rendszerhez
* Merevlemez-terület: 16 GB 32 bites operációs rendszerhez vagy 20 GB 64
* bites operációs rendszerhez
* Videokártya: A DirectX 9-es vagy újabb verziójával kompatibilis, WDDM 1.0
* illesztőprogrammal
* Kijelző: minden kijelző méretben működik
* Csak kliens oldali hardver követelményeink vannak, mert az oldalunk a felhőben
* tárolva elérhető

**Szoftver követelmények:**

Projektünkhöz szűkség van az alábbi szoftverekre:

• Windows 10 vagy újabb rendszer

• Linux

• MacOS

Webböngésző: A PartyEz webes alkalmazásként érhető el, ezért szükséges egy modern webböngésző a program használatához. Ajánlott böngészők: Google Chrome, Opera, Firefox, Microsoft Edge, Safari stb.

Internetkapcsolat: A Partyez webes alkalmazásként működik, így szükség van

internetkapcsolatra a program használatához és az adatok szinkronizálásához.

* A program használatának a részletes leírása

1. Bevezetés a PartyEz használatába

A PartyEz egy bulik keresésére alkalmas oldal, ahol egyszerűen lehet közelgő bulikat keresni, ezekre jelentkezni. A felhasználónak nem kell többé orákik kérdezősködni baráti körében, hogy ki tart bulit mostanság, csak egyszerűen fellép oldalunkra és az ő zenei ízlése szerint, Időbeosztása szerint a lehető legközelebbi bulira elmenni

Könnyen kezelhető:

A PartyEz rendkívül könnyen használható. Azoknak, akik nem rendelkeznek mély technikai ismeretekkel vagy akik nem szeretnének hosszú tanulási időt befektetni egy új rendszer megtanulásába, a Partyez ideális választást jelent. Egyszerűen hozzáférhető és könnyen kezelhető, ami segít a hatékony keresésben és az információk gyors és hatékony megtudásában.

Stabil rendszer:

A Partyez megbízható és stabil platformokon fut. Ez azt jelenti, hogy kevésbé valószínű, hogy elveszítenénk az adatainkat vagy hogy a rendszer összeomlik. Ez a stabilitás biztonságot nyújt azoknak, akik fontos információkat tárolnak a Partyez felületén.

2. Regisztráció és bejelentkezés:

* Regisztráció folyamata:

A regisztráció gomb a navbárban található. Teljes név, email cím és jelszó és születési dátum megadása: A regisztrációs űrlapon a felhasználónak meg kell adnia teljes nevét, email címét születési dátumát (Ezzel szűrjük a kiskorú személyeket akik nem használhatják oldalunkat, ezt ellenőrizzük.) és kiválasztania egy jelszót. A jelszót kétszer kell megadni a hibák elkerülése érdekében.

* Regisztráció gomb megnyomása: Miután a felhasználó kitöltötte az összes szükséges mezőt és kiválasztotta a szerepkörét, rákattint a "Regisztráció" gombra. Ezzel a regisztrációs folyamat megtörténik és a felhasználó át lesz irányítva az Partyez bejelentkező oldalára. Ahoz, hogy jelentkezhessünk bulikra kötelező regisztrálni.
* Bejelentkezés folyamata:

Email cím és jelszó megadása: A felhasználók először megadják az email címüket és a hozzátartozó jelszót a Partyez-be való belépéshez. Az email cím és a jelszó azonosítja és hitelesíti a felhasználót a rendszerben.

Bejelentkezés gomb megnyomása: Miután az email címet és a jelszót megadták, a felhasználó rákattint a "Bejelentkezés" gombra. Ez arra szolgál, hogy az oldal ellenőrizze az adott felhasználó azonosságát és hitelesítse a belépést. Ezután az oldal bedobja önt a létrehozott profiljába.

Regisztrációra való átirányítás: Ha a felhasználó még nem rendelkezik fiókkal a Partyez alkalmazásban, akkor a bejelentkezési oldalon lehetősége van arra, hogy regisztráljon. Ehhez egyszerűen kattintson a Sign In, vagy a Sign Up gombra, amik egyből átdobja a kívánt oldalra.

* Profil: A profil gombra kattintva érhető el és minimális formálhatósággal van ellátva,

mert nem ez a fontos számunkra. Továbbá itt található az összes buli, amire jelentkezett a felhasználó, innen törölni is tudja a nemkívánt programokat.

* Services: A navbárban helyezkedik el ez a gomb is és rákattintva megjelennek az aktuális programok, amikre jelentkezni lehet, továbbá egy kedvcsináló szöveg, egy kép a helyszínről és egy rövid leírás található minden egyes programnál.
* **Fejlesztői dokumentáció**
* **Az alkalmazott fejlesztői eszközök**

A projekt során az alábbi fejlesztői eszközöket és technológiákat használtuk

**Programozási nyelvek:**

• JavaScript (JS): A kliensoldali interaktivitás biztosításához.

• Cascading Style Sheets (CSS): A weboldal vizuális stílusának kialakításához.

• HyperText Markup Language (HTML): A weboldal szerkezetének

felépítéséhez.

• Structured Query Language (SQL): Az adatbázis-műveletek végrehajtásához.

**Fejlesztői környezet:**

• Microsoft Visual Studio Code: A kód szerkesztésére és kezelésére.

**Adatbázis-kezelő:**

• XAMPP: A helyi szerver és adatbázis-kezelő szolgáltatásokhoz

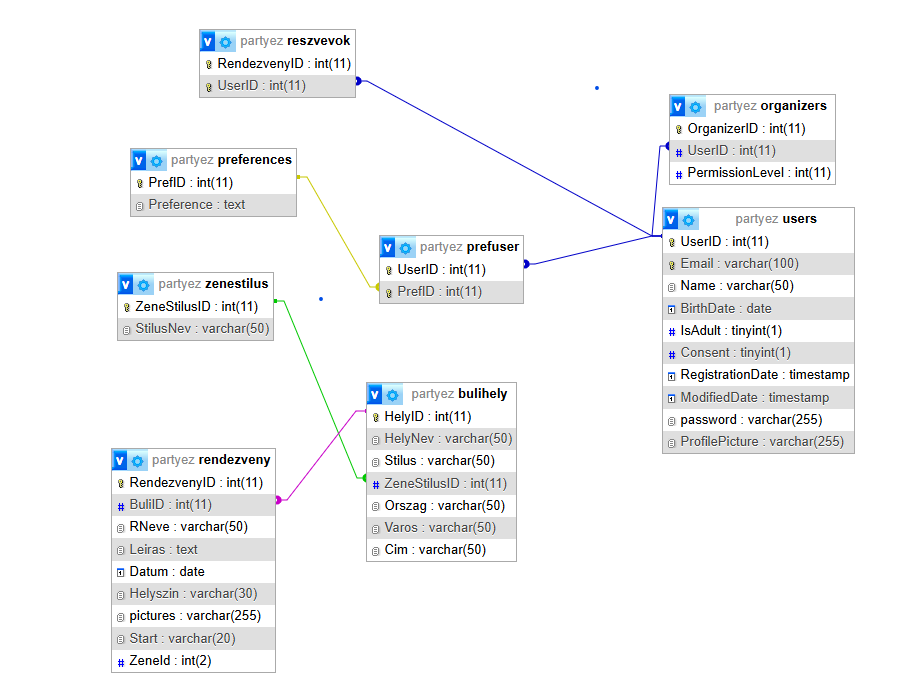
**Szövegszerkesztő:**

• Microsoft Word: A dokumentációk és egyéb szöveges anyagok készítéséhez.

Ezen eszközök használata biztosította a projekt zökkenőmentes fejlesztését és a különböző komponensek hatékony integrációját**.**

* **Adatmodell leírása**

Egy olyan adatbázist hoztunk létre, ami eltárolja a felhasználóink adatait és ezzel biztonságos használatot biztosít. A különböző partikat is itt tároljuk és a hozzákuk tartozó adatokat



## **1. Részletes feladatspecifikáció, algoritmusok**

### **1.1 A program célja**

A React alapú webalkalmazás célja egy eseményekkel kapcsolatos felület biztosítása, amely lehetővé teszi a felhasználók számára, hogy:

* Saját profiljukat megtekinthessék és szerkeszthessék
* Profilképüket módosíthassák egy előre megadott képlistából
* Megnézhessék, mely eseményekre jelentkeztek
* Jelentkezéseiket törölhessék

### **1.2 Komponensek specifikációja**

#### **1.2.1 Home.jsx**

* **Cél:** A főoldal felépítése három fő szekcióval: HeroSection, Cards, Footer.
* **Paraméterek:** Nincs
* **Visszatérési érték:** JSX elem (HTML struktúra)
* **Fontosabb függőségek:**
  + HeroSection: figyelemfelkeltő bemutatkozó rész
  + Cards: események vagy egyéb fontos információk vizuális megjelenítése
  + Footer: lábléc információk

#### **1.2.2 Profile.jsx**

* **Cél:** A bejelentkezett felhasználó profiljának megjelenítése, szerkesztési lehetőséggel.
* **Állapotkezelés:**
  + userData: felhasználói adatok objektuma
  + registrations: felhasználó esemény-jelentkezései
  + loading, error, successMessage: folyamatállapotok
* **Fontos metódusok:**

|  |  |
| --- | --- |
| **Függvény** | **Leírás** |
| fetchUserData() | Betölti a felhasználó adatait és jelentkezéseit |
| handleLogout() | Kijelentkezteti a felhasználót |
| handleEditImage() | Megnyitja a profilkép választó modal ablakot |
| handleImageSelect(img) | Kiválasztja az új profilképet |
| handleSaveImage() | Elmenti a kiválasztott profilképet az adatbázisba |
| handleDeleteRegistration(id) | Törli a kiválasztott jelentkezést |
| toggleRegistrations() | Mutatja/elrejti az eseményeket |
| handleImageError() | Hiba esetén fallback képet tölt be |
| formatDate(date) | Magyar formátumban jeleníti meg a dátumokat |

## **Tesztelési dokumentáció**

### **2.1 Tesztelési módszerek**

* **Fekete doboz tesztelés:** A felhasználói felület funkcióinak tesztelése a kimenet alapján.
* **Fehér doboz tesztelés:** A belső logika (API hívások, állapotváltozások) ellenőrzése manuálisan és logok alapján.

### **2.2 Tesztesetek**

#### **Teszteset 1: Normál bejelentkezett felhasználó**

* **Bemenet:** Érvényes userId van a localStorage-ban
* **Eredmény:**
  + Felhasználói profil megjelenik
  + Profilkép betöltődik
  + Jelentkezett események listája látható a "Show me my parties" gombra
* **Elvárt viselkedés:** Betöltés után minden adat megjelenik hibamentesen
* **Üzenetek:** „Loading…”, „Hi, Felhasználónév”
* **Tesztelési módszer:** Fekete doboz

#### **Teszteset 2: Extrém - nincs userId**

* **Bemenet:** Nincs bejelentkezett felhasználó (userId hiányzik)
* **Eredmény:**
  + „You have to log in first!” hibaüzenet
  + Átirányítás lehetősége bejelentkezésre
* **Elvárt viselkedés:** Az oldal ne töltsön be semmilyen profilt
* **Tesztelési módszer:** Fekete doboz, érvénytelen bemenet

#### **Teszteset 3: Sikeres profilkép mentés**

* **Bemenet:** Kiválasztott kép a modalból, majd „Save” gomb
* **Eredmény:**
  + API hívás sikeres
  + „Profile picture updated” (vagy hasonló) üzenet
  + A kép azonnal frissül
* **Elvárt viselkedés:** Állapotfrissítés és modal bezárás 1,5 másodperc múlva
* **Tesztelési módszer:** Fehér doboz – állapot és adatfrissítés ellenőrzése

### **2.3 Hibák, tapasztalatok**

|  |  |  |
| --- | --- | --- |
| **Hiba** | **Leírás** | **Javítás** |
| Nincs userId esetén nem történik navigáció | Felhasználó nem tud visszajutni a bejelentkezéshez | Hibaüzenet mellé navigációs link hozzáadva |
| Kép betöltési hiba | Invalid vagy nem létező kép URL | handleImageError() fallback képpel megoldva |

## **1. Függvényleírások**

### **SignUp komponens**

#### **handleSubmit(e)**

* **Cél**: A felhasználói regisztráció lebonyolítása.
* **Bemenet**: HTML submit esemény.
* **Kimenet**: Átirányítás bejelentkezésre siker esetén; hibaüzenet hiba esetén.
* **Működés**: Fetch POST kérés formData alapján a /register API-végpontra.

### **SignIn komponens**

#### **handleSubmit(e)**

* **Cél**: Bejelentkezés kezelése és userID tárolása.
* **Bemenet**: HTML submit esemény.
* **Kimenet**: Átirányítás /profile oldalra; hiba esetén üzenet.
* **Működés**: Fetch POST kérés /login végpontra; localStorage frissítése.

### **Services komponens**

#### **fetchEvents()**

* **Cél**: Események lekérése az adatbázisból.
* **Kimenet**: events állapot frissül.
* **Hiba esetén**: error állapot frissül.

#### **handleRegister()**

* **Cél**: Felhasználó eseményre való regisztrációja.
* **Bemenet**: nincs.
* **Kimenet**: Siker/hiba státuszüzenet.
* **Működés**: POST kérés a /reszvetel végpontra a userId és RendezvenyID mezőkkel.

### **Tesztesetek**

#### **Normál teszteset – Sikeres regisztráció**

* **Bemenet**: Valós email, jelszó, születési dátum, név.
* **Elvárt viselkedés**: "Sikeres regisztráció!" üzenet → átirányítás bejelentkezésre.
* **Valós eredmény**: Megfelelt.
* **Összefoglalás:**
* **Önértékelés**

Nagy út állt előttünk és hiányos volt a tudásunk, viszont a tutorial videók és a mesterséges intelligencia ellenőrzésével sikerült túl lendülnünk a legtöbb nehézségen.

**Viktor:** Én személy szerint megtanultam átfogóan a css formázást és a jsx-ek használatát, továbbá jobb lettem a dokumentációban és a powerpointnál szabadon engedhettem a kreativitásomat. Sokkal jobb vagyok a részletek és dizájnok kidolgozásában, fejlődött a problémamegoldó képességem és végre megtanultam jól navigálni magamat a kódok közt.

**Szilárd:** Megtanulta a legprecízebben megkérdezni az ait és felhasználni az onnan kapott információkat és átvinni gyakorlatba. Ezáltal megtanulta a backend problémák megoldását és a konzol kapcsolat létrehozását. Továbbá neki köszönhetőek az adatbázis kapcsolatok és az adatfeltöltések.

Összeségében mindketten fejlődtünk a szoftver tesztelés és fejlesztésben.

* Továbbfejlesztési lehetőségek

Rengeteg irányból fogunk tudni fejleszteni, többek között szeretnénk egy átfogóbb szűrőt, olyan felhasználókat, akik tudnak feltölteni bulikat egy adott sablonba és egy klikkszámlálót, ami segít, hogy a preferenciák is létrejöhessenek

* Felhasznált irodalom

**Hibajavításhoz/Hibakereséshez:** <https://grok.com/> (2025.04.19)

**Frontend formázáshoz:** <https://www.youtube.com/@briandesign> (2025.04.02)

**Adatfeltöltéshez:** <https://openai.com/chatgpt/overview/> (2025.04.13)