

VAS VÁRMEGYEI SZAKKÉPZÉSI CENTRUM  
HORVÁTH BOLDIZSÁR   
KÖZGAZDASÁGI ÉS INFORMATIKAI  
TECHNIKUM

A 5 0613 12 03 számú Szoftverfejlesztő és –tesztelő vizsgaremek

**A szoftveralkalmazás dokumentációja**

Készítették:

Rosta Hanna

Szalay Miklós

**SZOMBATHELY**

**2025**

# Bevezetés

A dokumentáció célja a táncverseny-eredménykezelő alkalmazás bemutatása, amely a Vas Vármegyei Szakképzési Centrum Horváth Boldizsár Közgazdasági és Informatikai Technikum diákjai, Rosta Hanna és Szalay Miklós által készített vizsgaremek. Az alkalmazás egy webalapú rendszer, amelynek célja a táncversenyek nevezéseinek, pontozásának és eredményeinek kezelése, valamint a felhasználók számára átlátható felület biztosítása a versenyek nyomon követésére. A rendszer alkalmas csapatok regisztrálására, bírók pontozásának rögzítésére, valamint az eredmények valós idejű megjelenítésére. A fejlesztés során a modern webtechnológiákra és adatbázis-kezelési megoldásokra támaszkodtunk, hogy egy stabil és felhasználóbarát alkalmazást hozzunk létre.

# Szoftver célja

Az alkalmazás célja egy olyan platform biztosítása, amely leegyszerűsíti a táncversenyek adminisztrációját és eredménykezelését. A rendszer lehetővé teszi:

* **Csapatok nevezését**: A tánccsapatok regisztrálhatják magukat egy adott versenyre, megadva a szükséges adatokat (csapatnév, létszám, kategória, zene linkje, csapattagok nevei).
* **Bírók regisztrációját és pontozását**: A bírók regisztrálhatnak a rendszerbe, és pontozhatják a nevezett csapatokat.
* **Eredmények megjelenítését**: A rendszer valós időben összesíti a bírók által adott pontszámokat, és megjeleníti az eredményeket a felhasználók számára.
* **Adminisztratív funkciók**: Az adminisztrátorok új versenyeket hozhatnak létre, kategóriákat rendelhetnek hozzájuk, és kezelhetik a nevezéseket.

A szoftver célja, hogy a táncversenyek szervezői, bírói és résztvevői számára egy átlátható, könnyen használható rendszert biztosítson, amely csökkenti az adminisztratív terheket és növeli a versenyek lebonyolításának hatékonyságát.

# Komponensek technikai leírása

Az alkalmazás két fő részből áll: a frontendből és a backendből, amelyek egy MySQL adatbázissal kommunikálnak.

## ****Frontend****

* **Technológia**: React (JavaScript könyvtár)
* **Verzió**: React 18.2.0
* **Függőségek**:
  + smooth-scroll: A sima görgetés megvalósításához.
  + bootstrap: Az alapvető stílusok és reszponzív dizájn biztosításához.
* **Leírás**: A frontend a felhasználói felületet biztosítja, amelyen keresztül a felhasználók interakcióba léphetnek a rendszerrel. A React komponensek modularitást és újrafelhasználhatóságot biztosítanak.
* **Főbb komponensek**:
  + **Navigation**: Navigációs sáv, amely három fő funkció elérését biztosítja: "Nevezési Űrlap", "Eredmények", "Bírói Regisztráció".
  + **Modal**: Egy általános modál komponens, amely különböző űrlapokat és táblázatokat jelenít meg (pl. nevezési űrlap, eredmények táblázat).
  + **ErrorBoundary**: Hibakezelő komponens, amely megakadályozza, hogy az alkalmazás összeomoljon renderelési hibák esetén.
* **Funkciók**:
  + Nevezési űrlap: Lehetővé teszi a csapatok regisztrálását egy adott versenyre és kategóriára.
  + Eredmények modál: Megjeleníti a versenyek eredményeit, szűrhető verseny és kategória szerint.
  + Bírói regisztráció: Lehetővé teszi új bírók regisztrálását a rendszerbe.
* **Felhasználói Felület Elemei**:
  + Űrlapok: Szövegmezők, legördülő menük, gombok (pl. "Nevezés", "Frissítés").
  + Táblázatok: Az "Eredmények" modálban egy táblázat jeleníti meg az eredményeket (oszlopok: Verseny, Kategória, Csapatnév, Összpontszám).
  + Stílusok: A felület reszponzív, mobiltelefonokon és asztali gépeken egyaránt jól használható.

## ****Backend****

* **Technológia**: ASP.NET Web API (C#)
* **Verzió**: .NET Core 3.1
* **Függőségek**:
  + Entity Framework: Adatbázis műveletek kezelésére.
  + Microsoft.AspNetCore.Cors: CORS engedélyezéséhez a frontend-backend kommunikációhoz.
* **Leírás**: A backend REST API-t biztosít, amely kezeli az adatbázis műveleteket és a frontend kéréseit. Az Entity Framework segítségével kommunikál az adatbázissal.
* **Főbb Kontrollerek**:
  + **VersenyController**:
    - GET /api/Verseny: Visszaadja az összes versenyt a hozzájuk tartozó kategóriákkal.
    - POST /api/Verseny: Új versenyt hoz létre a megadott kategóriákkal.
  + **CsapatController**:
    - POST /api/Csapat: Új csapatot regisztrál, és nevezést hoz létre egy versenyhez.
  + **NevezesController**:
    - GET /api/Nevezes: Visszaadja a nevezéseket és az összesített eredményeket, szűrhető verseny és kategória szerint.
  + **EredmenyController**:
    - POST /api/Eredmeny: Új pontozást rögzít egy adott nevezéshez.
  + **BirokController**:
    - GET /api/Birok: Visszaadja az összes bíró adatait.
    - POST /api/Birok: Új bírót regisztrál.
* **Funkciók**:
  + Adatbázis műveletek (CRUD: Create, Read, Update, Delete).
  + Pontozások összesítése: A bírók által adott pontszámok szummázása egy adott nevezéshez.
  + Validációk: Ellenőrzi, hogy a megadott kategóriák léteznek-e, és tartoznak-e a kiválasztott versenyhez.

## Adatbázis

* **Technológia**: MySQL (MariaDB 10.4.32)
* **Karakterkódolás**: UTF-8 (utf8mb4\_general\_ci)
* **Leírás**: Az adatbázis tárolja a rendszer adatait, például a versenyeket, kategóriákat, nevezéseket, csapatokat, bírókat és pontozásokat.
* **Táblák és Kapcsolatok**:
  + versenyek: Tárolja a versenyek adatait (Id, Nev, Idopont).
  + kategoriak: Tárolja a kategóriák adatait (Id, Nev).
  + versenykategoriak: Kapcsolótábla a versenyek és kategóriák között (VersenyId, KategoriaId).
  + csapatok: Tárolja a csapatok adatait (Id, Nev, Letszam, ZeneLink).
  + csapattagok: Tárolja a csapattagok adatait (Id, Nev, CsapatId).
  + nevezesek: Tárolja a nevezéseket (Id, VersenyId, CsapatId, KategoriaId, Datum).
  + eredmenyek: Tárolja a bírók által adott pontszámokat (Id, NevezesId, BiroId, Pontszam, Rogzitve).
  + birok: Tárolja a bírók adatait (Id, Nev, Email, Telefon, Jelszo, TitkosKulcs).
* **Kapcsolatok**:
  + A versenykategoriak tábla biztosítja a több-több kapcsolatot a versenyek és kategoriak táblák között.
  + A csapattagok tábla a csapatok táblára hivatkozik (CsapatId idegen kulcs).
  + A nevezesek tábla a versenyek, csapatok és kategoriak táblákra hivatkozik.
  + Az eredmenyek tábla a nevezesek és birok táblákra hivatkozik.

**Rendszerarchitektúra**

Az alkalmazás egy tipikus háromrétegű architektúrát követ:

* **Prezentációs Réteg (Frontend)**: A React-alapú felület, amely a felhasználói interakciókat kezeli.
* **Logikai Réteg (Backend)**: Az ASP.NET Web API, amely az üzleti logikát és az adatbázis műveleteket kezeli.
* **Adat Réteg (Adatbázis)**: A MySQL adatbázis, amely az adatokat tárolja.

A frontend és a backend közötti kommunikáció REST API-n keresztül történik, JSON formátumban. A kérések HTTPS protokollt használnak a biztonság érdekében

## ER modell

A képen szöveg, diagram, Betűtípus, Párhuzamos látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

# Működésének műszaki feltételei

Az alkalmazás működéséhez az alábbi műszaki feltételek szükségesek:

## Kliens Oldal (Frontend)

* **Böngésző**: Modern webböngésző (pl. Google Chrome, Mozilla Firefox, Microsoft Edge) legfrissebb verziója.
* **Internetkapcsolat**: Stabil internetkapcsolat a backend API-val való kommunikációhoz.
* **Operációs rendszer**: Bármely modern operációs rendszer (Windows, macOS, Linux), amely támogatja a fenti böngészőket.
* **Hardver**: Minimum 4 GB RAM, 1 GHz processzor a zökkenőmentes működéshez.

## Szerver Oldal (Backend)

* **Operációs rendszer**: Windows Server vagy Linux (pl. Ubuntu) az ASP.NET Core futtatásához.
* **Keretrendszer**: .NET Core 3.1 vagy újabb (az ASP.NET Web API futtatásához).
* **Adatbázis**: MySQL (MariaDB 10.4.32 vagy újabb) szerver.
* **Hardver**: Minimum 8 GB RAM, 2 GHz processzor, 20 GB szabad tárhely az adatbázis és a szerveralkalmazás számára.
* **Hálózat**: Stabil internetkapcsolat, HTTPS tanúsítvány a biztonságos kommunikációhoz.

## Fejlesztői Környezet

* **IDE**: Visual Studio 2022 (backend fejlesztéshez), Visual Studio Code (frontend fejlesztéshez).
* **Csomagkezelő**: npm (frontend függőségek telepítéséhez). Port: localhost:3000
* **Adatbázis kezelő**: phpMyAdmin az adatbázis kezelésére. Port: localhost:3307

## Bejelentkezés

1. ***Admin felület***:

<https://localhost:3000/admin/secret-page>

**Felhasználónév**: admin

**Jelszó**: admin123

1. ***Bírói pontozó felület***

Email: [teszt@teszt.hu](mailto:teszt@teszt.hu)

Jelszó: teszt

# Használatának bemutatása

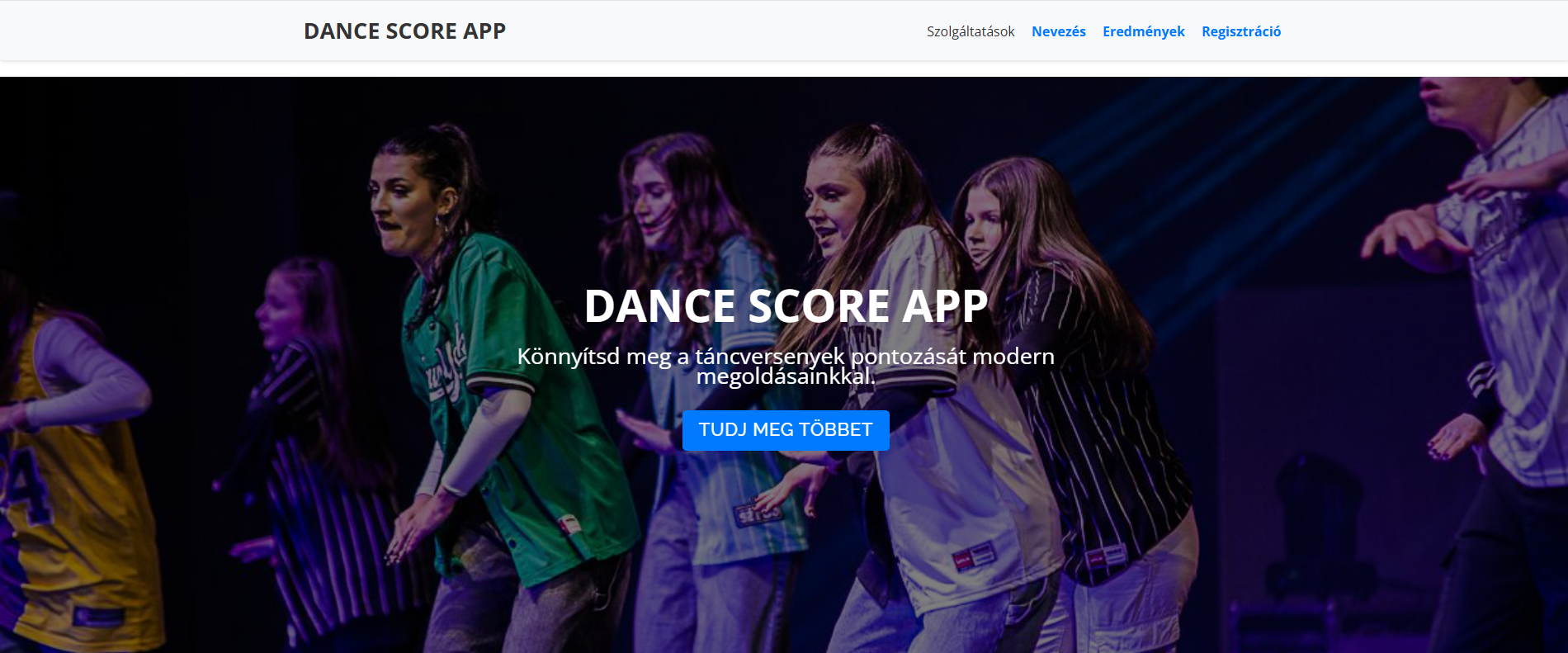
Az alkalmazás használata egyszerű és intuitív. Az alábbiakban röviden bemutatjuk a főbb funkciókat:

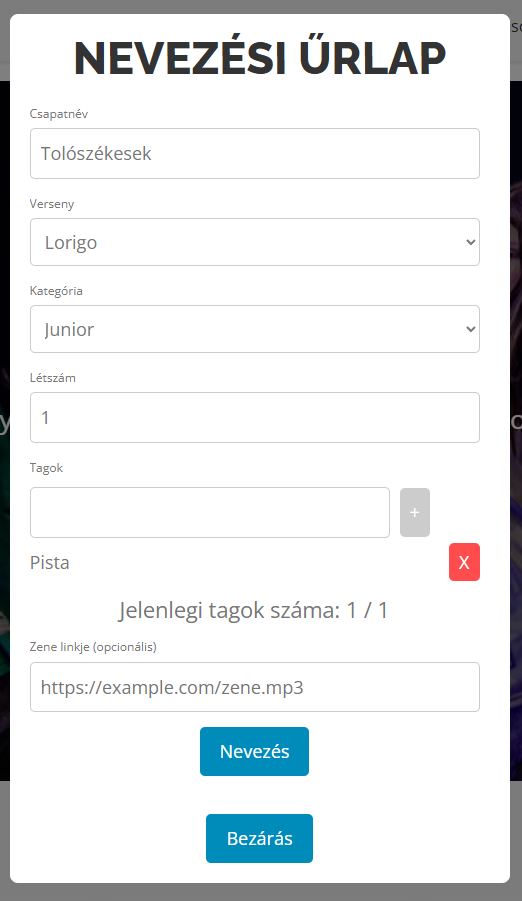
1. **Bejelentkezés és Navigáció**:
   * A felhasználók a navigációs sávon keresztül érhetik el a főbb funkciókat: "Nevezési Űrlap", "Eredmények", "Bírói Regisztráció".
   * Bejelentkezés nem szükséges a publikus funkciókhoz (pl. eredmények megtekintése), de a bírói pontozáshoz regisztráció és bírói azonosító szükséges.
2. **Csapat nevezése**:
   * A "Nevezési Űrlap" modálban a felhasználók kiválaszthatják a versenyt (pl. "Lorigo Bajnokság") és a kategóriát (pl. "Gyermek").
   * Meg kell adni a csapat nevét, létszámát, a csapattagok nevét, és opcionálisan egy zene linkjét.
   * A nevezés elküldése után a rendszer menti a csapatot és a nevezést az adatbázisba.
3. **Bírói regisztráció**:
   * A "Bírói Regisztráció" modálban a bírók regisztrálhatnak a rendszerbe, megadva nevüket, email címüket, telefonszámukat, jelszavukat és a bírói azonosítójukat.
   * A jelszó és a bírói azonosító titkosítva kerül mentésre az adatbázisba (SHA256 algoritmus használatával).
4. **Pontozás**:
   * A bírók az asztali alkalmazás felületén keresztül pontozhatják a nevezéseket. Minden bíró csak egyszer pontozhat egy adott nevezést.
   * A pontszámok az eredmenyek táblába kerülnek mentésre, és a rendszer automatikusan összesíti őket.
5. **Eredmények megtekintése**:
   * Az "Eredmények" modálban a felhasználók kiválaszthatják a versenyt és a kategóriát, majd megtekinthetik az eredményeket egy táblázatban.
   * A táblázatban látható a verseny neve, a kategória, a csapat neve és az összpontszám (a bírók által adott pontszámok összege).

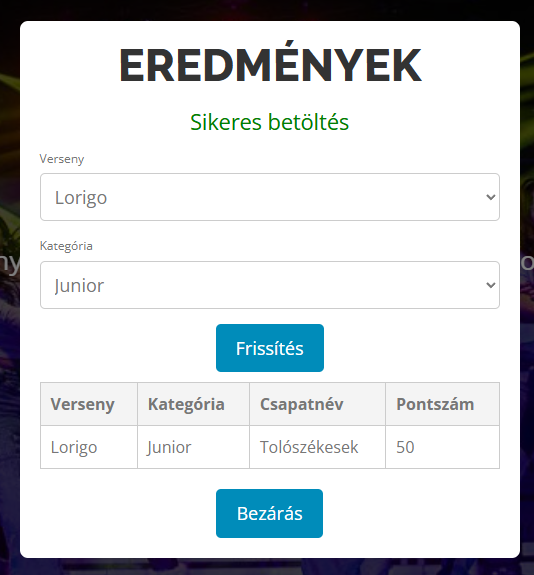
## Monitorképek

Az alábbiakban néhány monitorkép található, amelyek bemutatják az alkalmazás főbb felületeit:

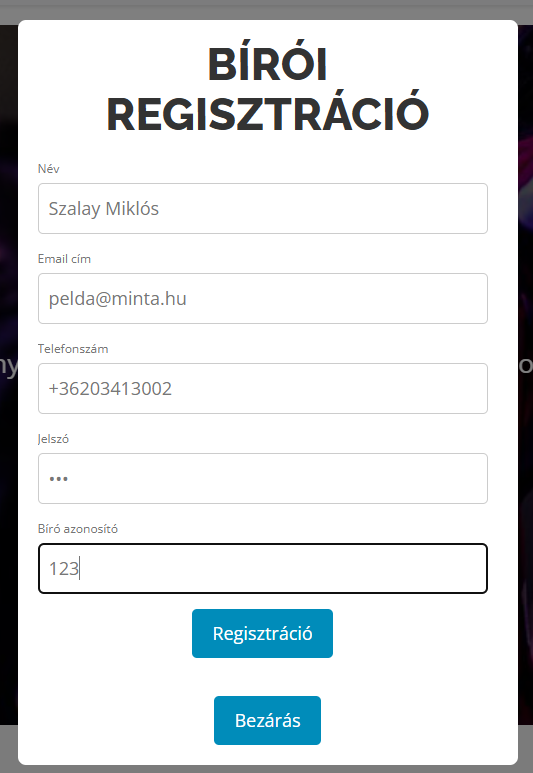
1. **Főoldal**:
   * A navigációs sáv látható, amelyen keresztül a modálok megnyithatók.



1. **Nevezési Űrlap Modál**:
   * A modálban látható a nevezési űrlap, ahol a felhasználók megadhatják a csapat adatait.
   * Példa: "Csapatnév: Tolószékesek", "Verseny: Lorigo ", "Kategória: Junior", "Létszám: 1", "Tagok: Pista".
2. **Eredmények Modál**:
   * A modálban egy táblázat látható, amely az eredményeket jeleníti meg.
   * Példa: "Verseny: Lorigo", "Kategória: Junior", "Csapatnév: Tolószékesek", "Összpontszám: 50".



1. **Bírói Regisztráció Modál**:
   * A modálban látható a bírói regisztrációs űrlap.
   * Példa: "Név: Szalay Miklós", "Email: pelda@minta.hu", "Telefonszám: +36203413002".



1. **Adminisztrátori felület**
   * Sikeres bejelentkezés után megjelenik a panel, ahol új versenyeket hozhatunk létre, pontosan megadva a verseny: nevét, időpontját és kategóriáját.
   * Majd, ha a „Verseny mentése” gombra kattintunk azonnal megjelenik a weboldalunk nevezési modáljában választási lehetőségként.

A képen szöveg, képernyőkép, képernyő, Betűtípus látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

# Továbbfejlesztési lehetőségek

Az alkalmazás jelenlegi verziója számos alapvető funkciót biztosít, de a következő továbbfejlesztési lehetőségek megvalósításával még hatékonyabbá tehető:

1. **Felhasználói Fiókok és Bejelentkezés**:
   * Jelenleg csak a bírók regisztrálhatnak, de a rendszer kiegészíthető egy általános bejelentkezési funkcióval, amely lehetővé teszi a felhasználók számára, hogy saját fiókot hozzanak létre, és nyomon kövessék nevezéseiket.
2. **Pontozási Rendszer Finomítása**:
   * A pontozás jelenleg egyszerű összeadáson alapul. A rendszer kiegészíthető súlyozott pontozással (pl. különböző bírók pontszámainak eltérő súlya lehet), vagy a legmagasabb és legalacsonyabb pontszámok elhagyásával az átlag kiszámításakor.
3. **Értesítési Rendszer**:
   * Bevezethető egy értesítési rendszer, amely emailben vagy a felületen keresztül értesíti a felhasználókat a nevezésük állapotáról, a pontozás eredményéről, vagy a versenyekkel kapcsolatos hírekről.
4. **Adminisztrátori Felület Bővítése**:
   * Az adminisztrátori felület jelenleg korlátozott funkcionalitással rendelkezik. Bővíthető például a nevezések szerkesztésével, törlésével, vagy a versenyek időpontjának módosításával.
5. **Mobilalkalmazás**:
   * Az alkalmazás jelenleg webes felületen érhető el, de egy mobilalkalmazás fejlesztésével (pl. React Native használatával) a felhasználók számára még kényelmesebbé tehető a hozzáférés.
6. **Statisztikák és Analitikák**:
   * A rendszer kiegészíthető statisztikai funkciókkal, például a csapatok teljesítményének időbeli elemzésével, vagy a bírók pontozási szokásainak vizualizációjával (pl. grafikonok, diagramok).
7. **Többnyelvű Támogatás**:
   * Az alkalmazás jelenleg magyar nyelven érhető el. A többnyelvű támogatás (pl. angol, német) bevezetésével nemzetközi versenyekre is alkalmassá tehető.

# Forrás:

A frontend alapját egy szabadon felhasználható sablon képezi mely itt érhető el:

<https://github.com/issaafalkattan/React-Landing-Page-Template>

