

2025

VIZSGAREMEK PROJEKTMUNKA

CompetitionManager

CompetitionManager

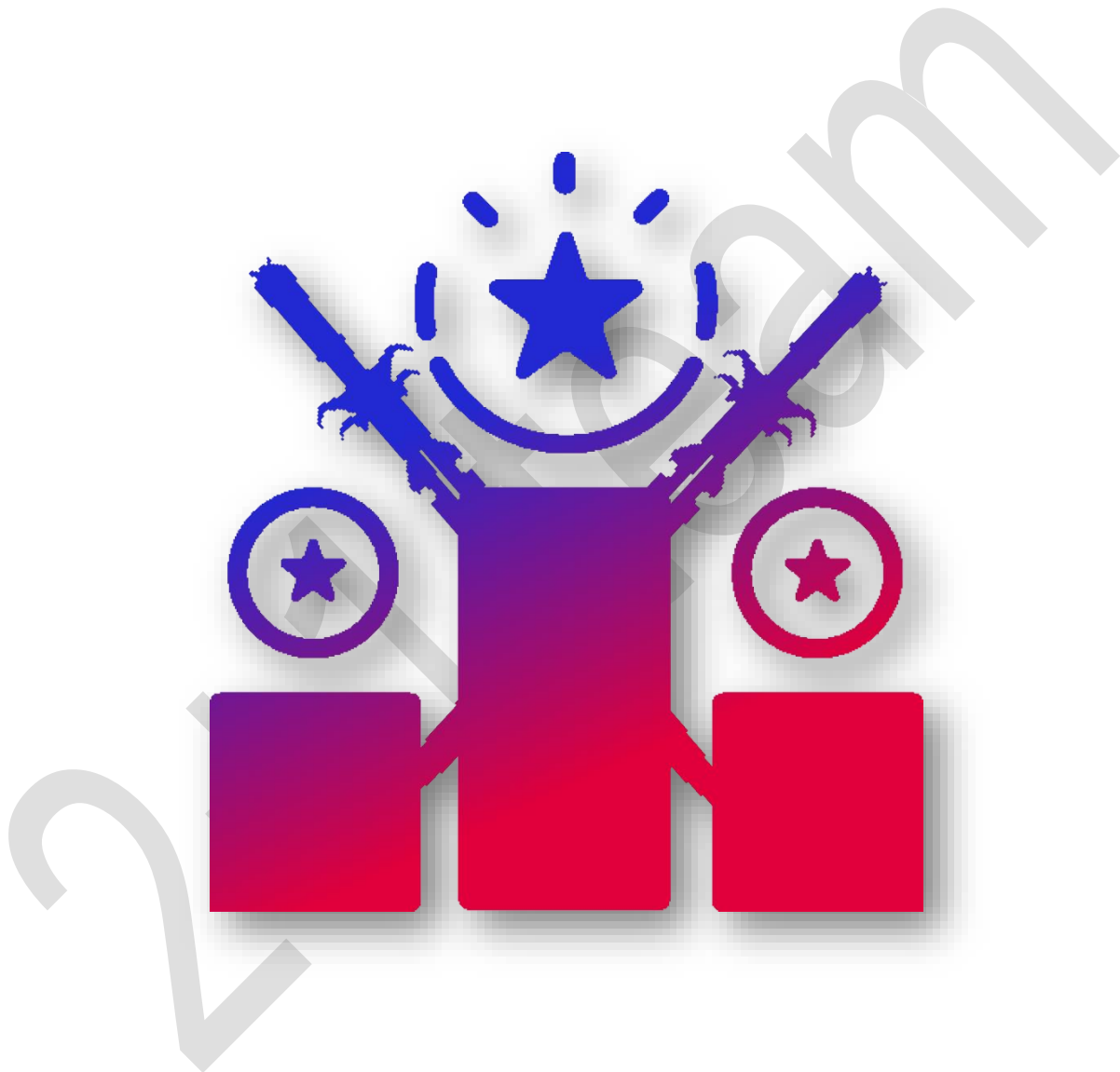
Projektmunka

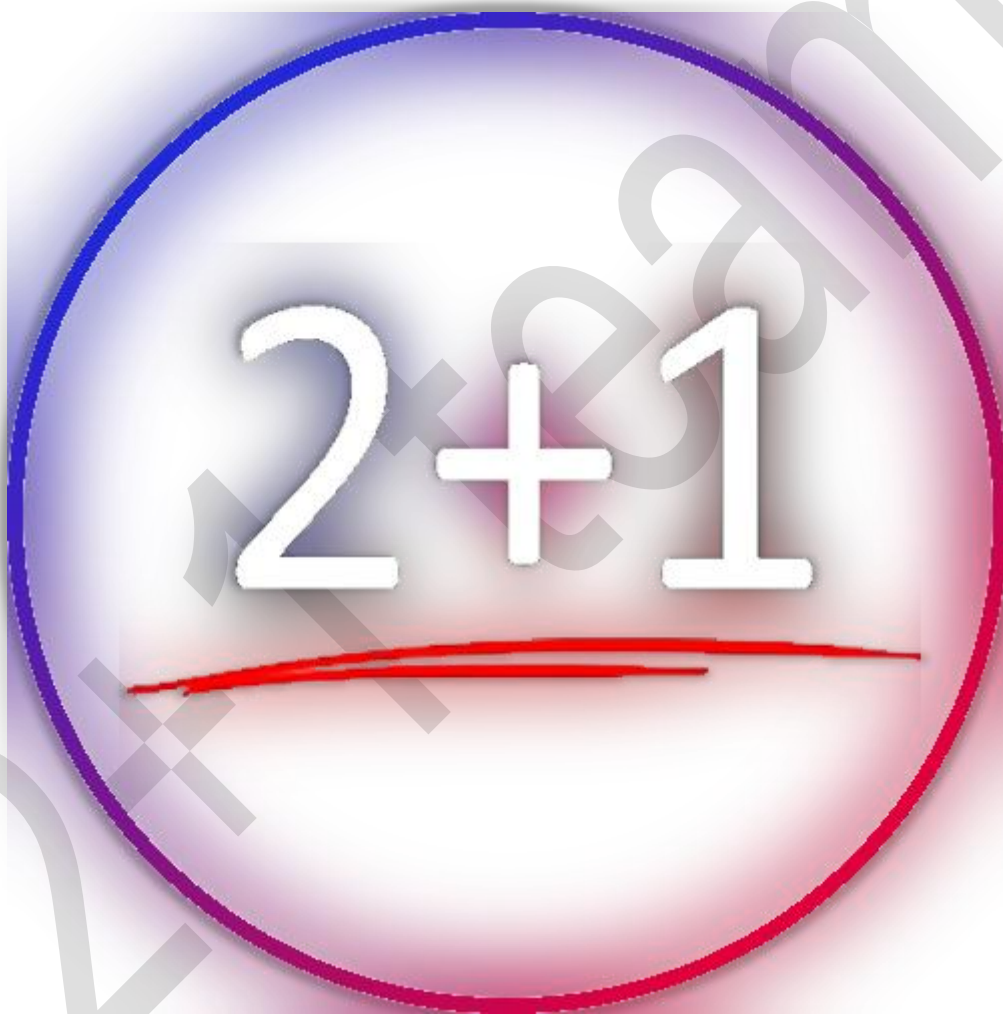
Témavezető

Juhászné Kovács Ildikó

Készítették

Szabó Dávid, Szürke Levente, Szántó Dávid





2+1 team

NYÍREGYHÁZA

2025

oldal 2 / 30

Tartalomjegyzék

1. Feladat megadása	4
2. Szereplők és igényeik	4
3. Use-Case funkció lista	
- Adminisztrátor (versenykezelő)	5
- Felhasználó (versenyző, szülők stb.)	5
4. Magas szintű rendszerterv	6
5. Screenek	7
6. Modellek	13
7. Alkalmazások	
- Frontend	14
- Backend	14
- Asztali alkalmazás	14
- Adatbázis	15
- Tesztelés	15
8. Routing	16
9. Implementáció	24
10. Tesztelés	25
11. Éles üzemmód	28
12. Összegzés	29

Feladat megadása

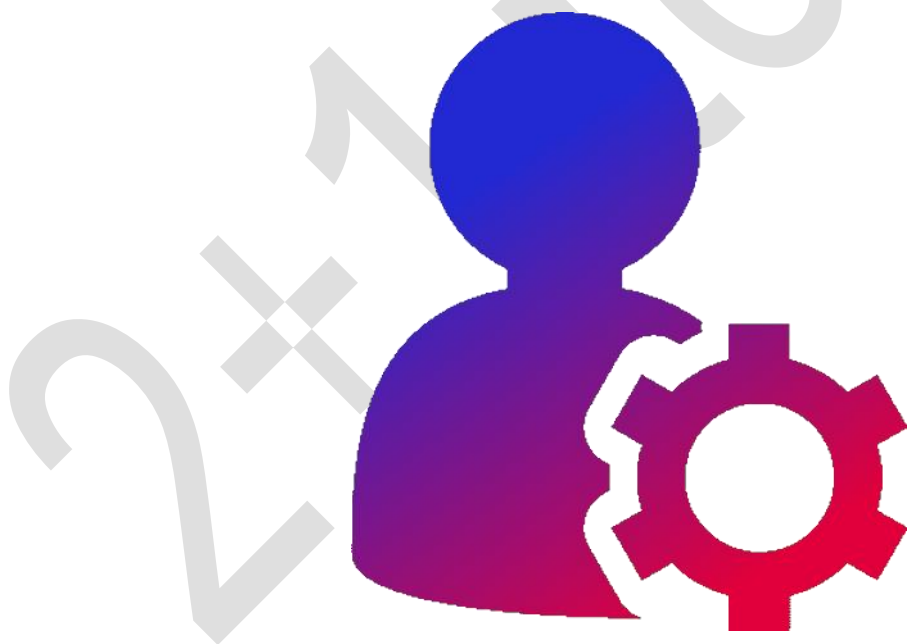
Feladatunk egy olyan versenykezelő program fejlesztése, mely megkönnyíti a versenyek lebonyolítását és egy helyen összesíti a nemzetközi és hazai eredményeket, hogy mindenki elérhesse azokat. A program minden teljes verseny eredményét adatbázisban fogja rögzíteni, hogy azok később egy weboldalon lekérdezhetőek legyenek a versenyzők/szülők számára.

Szereplők és igényeik

Ahogy azt fent is említettük, projektünk több részből fog állni, melyet különböző felhasználók fognak igénybe venni.

Adminisztrátor (Versenykezelő)

Az adminisztrátor vagy versenykezelő az a személy, aki az asztali alkalmazást kezeli. Az asztali alkalmazás felelős a versenyek lebonyolításáért, a versenyzők kezeléséért, és a versenyek feltöltéséért. Mivel versenyt nem kezelhet akárki, az asztali alkalmazást csak felhasználói fiókkal rendelkező személyek használhatják. Felhasználói fiókot csak annak megigénylésével lehet szerezni, mivel a program különböző szövetségek vagy kisebb szervezetek számára lett tervezve.



Felhasználó (versenyzők, szülők stb.)

A felhasználó az, aki az a weboldaltól lekérdezi egy adott versenyző eredményeit. A weboldalhoz éppen ezért bárki hozzáférhet. A weboldal letisztult és lényegre törő lesz, a felhasználónak minél egyszerűbben tudnia kell lekérdeznie a versenyzők eredményeit.

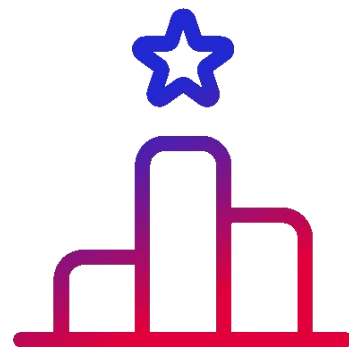


Use-Case funkció lista

A projektben a felhasználók két csoportra vannak osztva. Az Admin asztali alkalmazást és a weboldalt használók. A két csoport közötti különbség az, hogy az Admin használók azok, akik a versenyzőket és versenyeket kezelik. Nekik joguk van a versenyzők adatainak módosításához, míg az a felhasználó csoport amelyik csak a weboldalhoz fér hozzá, csupán csak megtekinteni tudják a versenyeket és a versenyzők eredményeit.

Adminisztrátor (Versenykezelő)

- Versenyzők megtekintése
- Versenyző felvétele
- Versenyző adatainak módosítása
- Versenyző Törlése
- Körverseny indítása
- Tábla verseny indítása
- Teljes verseny indítása



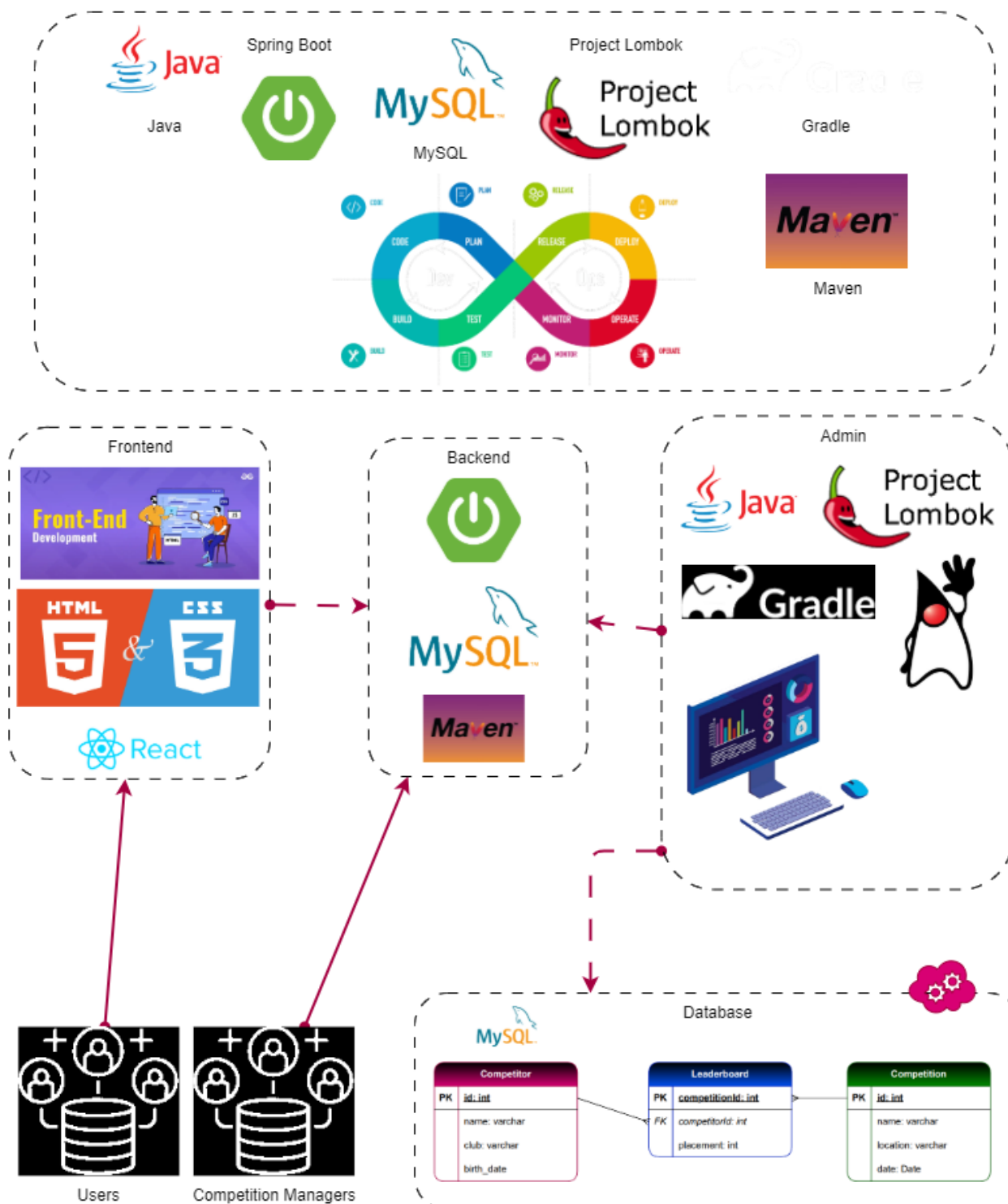
Felhasználó (versenyzők, szülők stb.)

- Versenyek megtekintése
- Versenyzők versenyeredményeinek megtekintése

Magas szintű rendszerterv



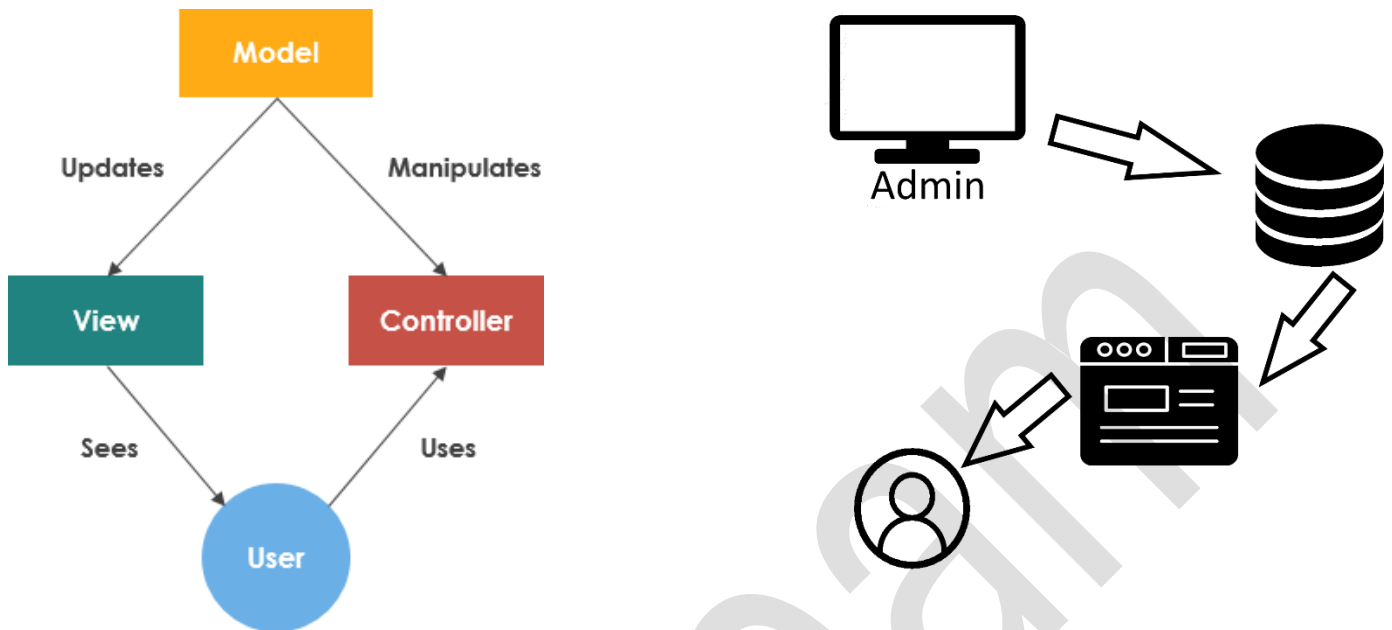
CompetitionManager Tech Stack



A projekthez szükséges technológiák strukturált diagramja

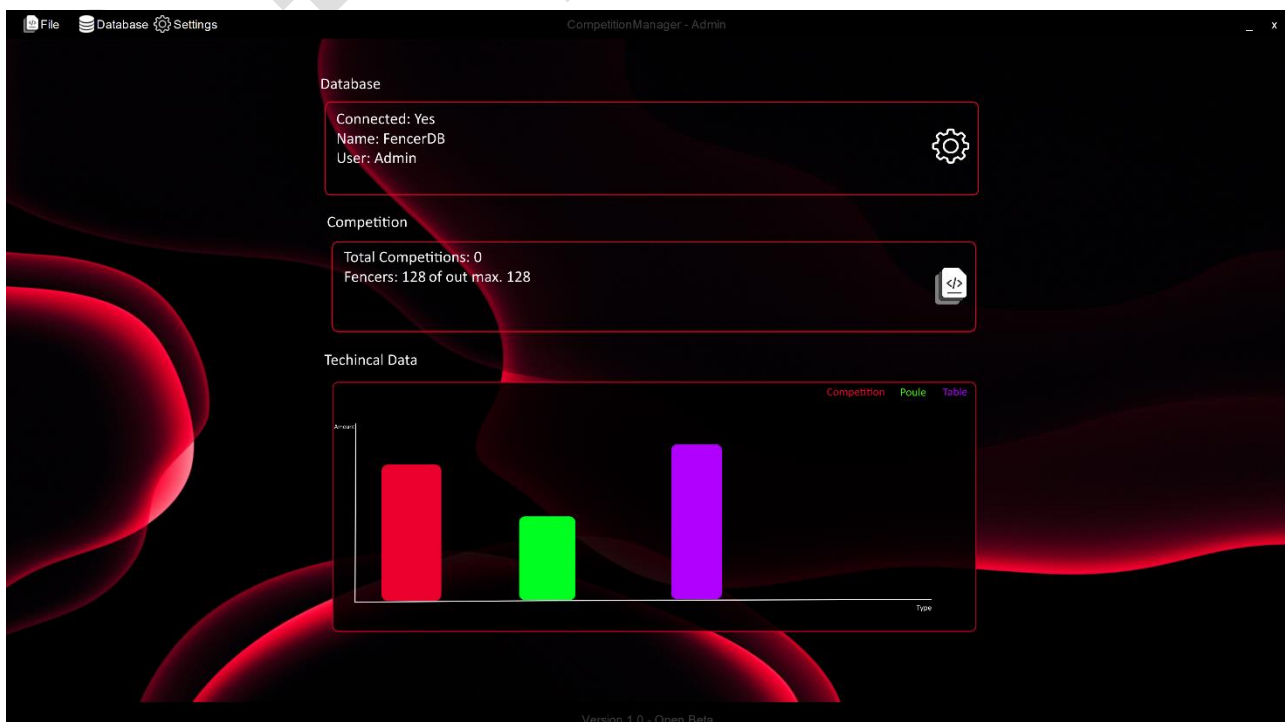
2+1 team fejlesztése

A fejlesztés során az MVC modellt követjük, a letisztultság, egyszerűség mégis magas hatékonyság elérésének érdekében. Az Admin felületről fognak bekerülni az adatok az adatbázisba, amit a felhasználók a frontendend lekérdezhetnek.



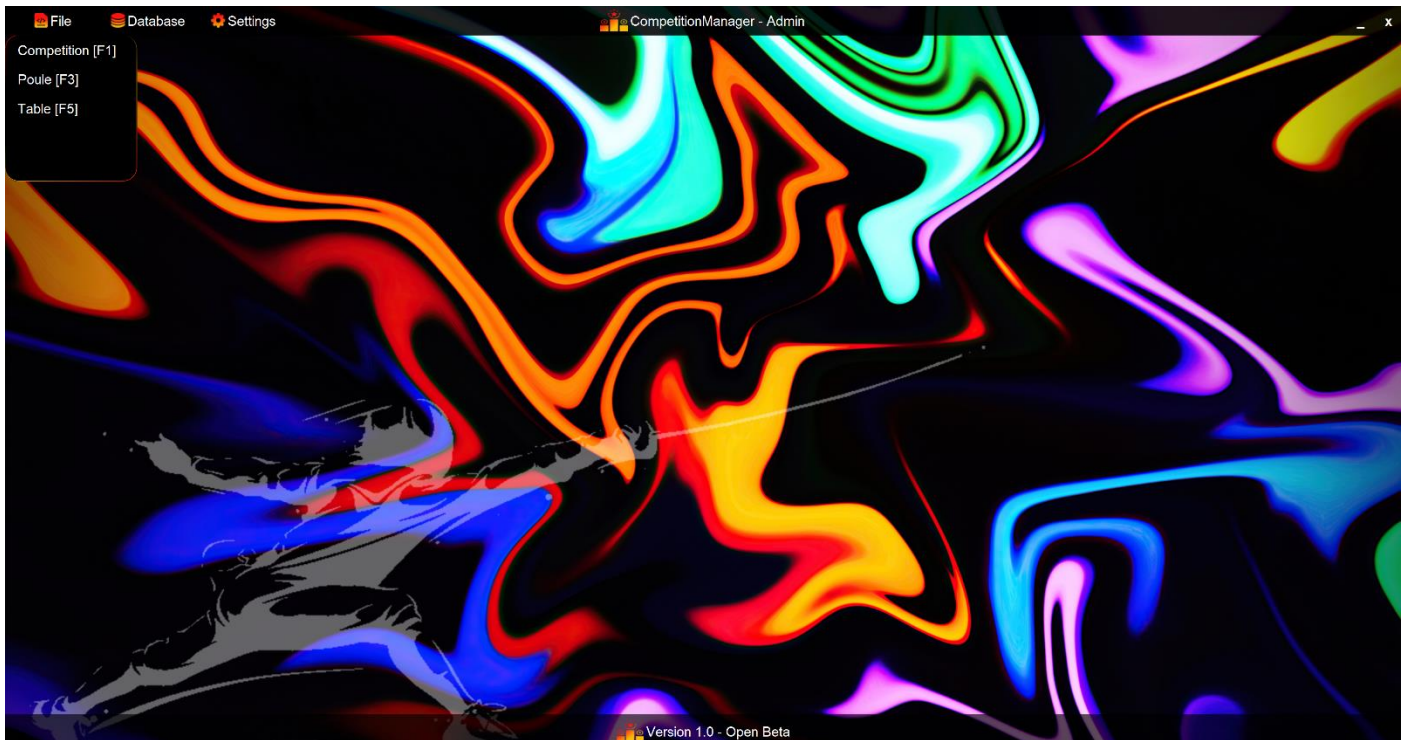
Screenek

Projektünk több részből áll, de mindegyik részre igaz, hogy az igényes kinézet, a letisztultság, és a könnyed kezelhetőség elsőbbséget élvez listánkon. Az asztali alkalmazásnak nagy mennyiségű adatot kell tudnia feldolgozni és legtöbbször a felhasználó az, aki megadja a szempontokat, melyhez egy hosszú listában görgetve keresni már nem elég. Éppen ezért minden adat módosítással kapcsolatos elemet kereső funkcióval fogunk ellátni és ezek az elemek könnyen elérhetőek lesznek.

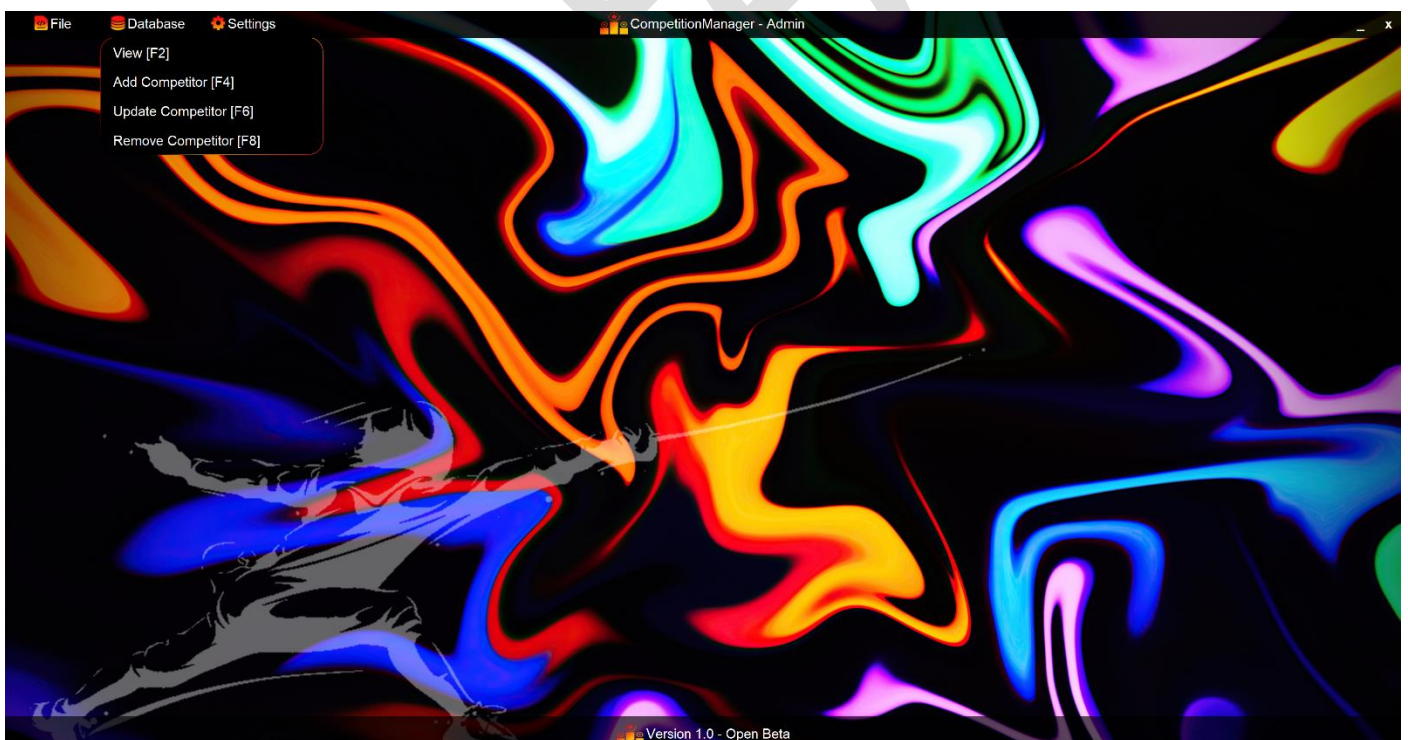


2+1 team fejlesztése

A fenti képen az asztali alkalmazás kezdő oldalának sablonja látható.

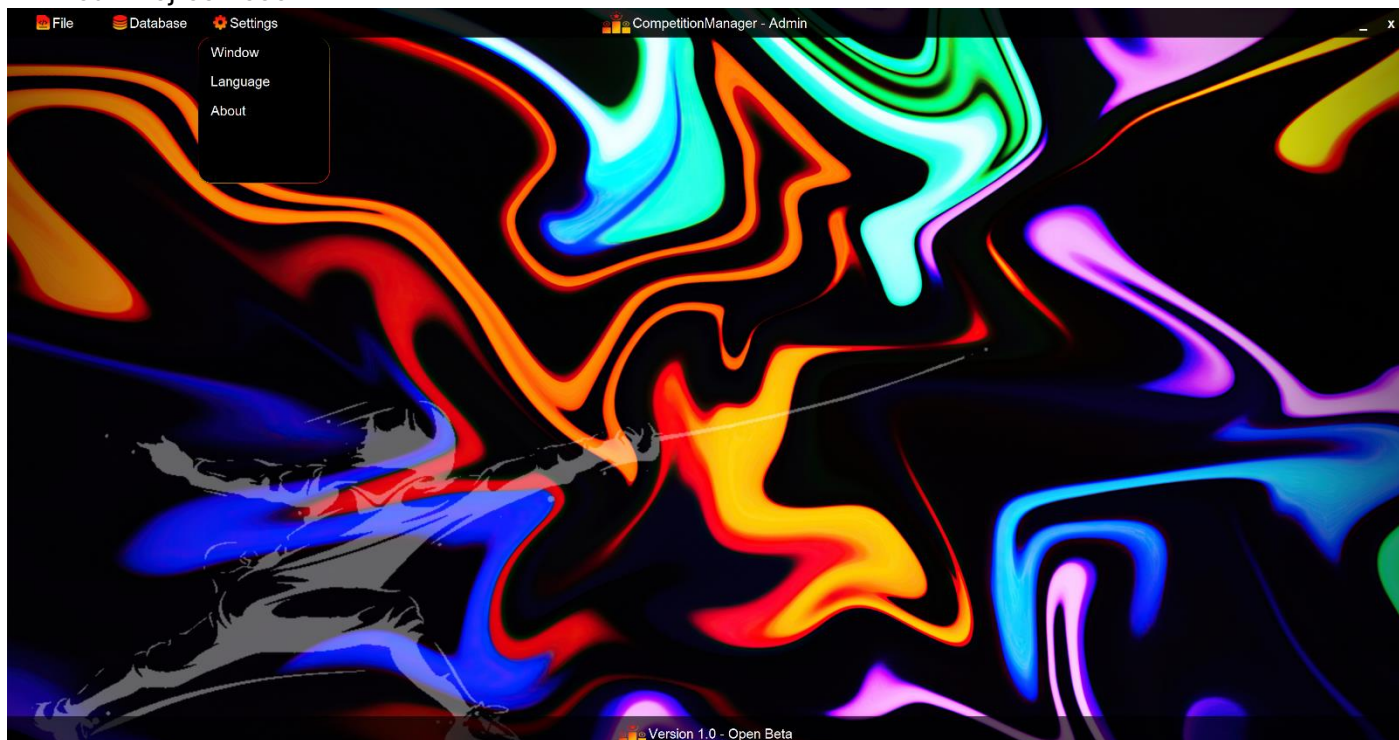


A „File” menü felületi terve



A „Database” menü felületi terve

2+1 team fejlesztése



A „Settings” menü felületi terve



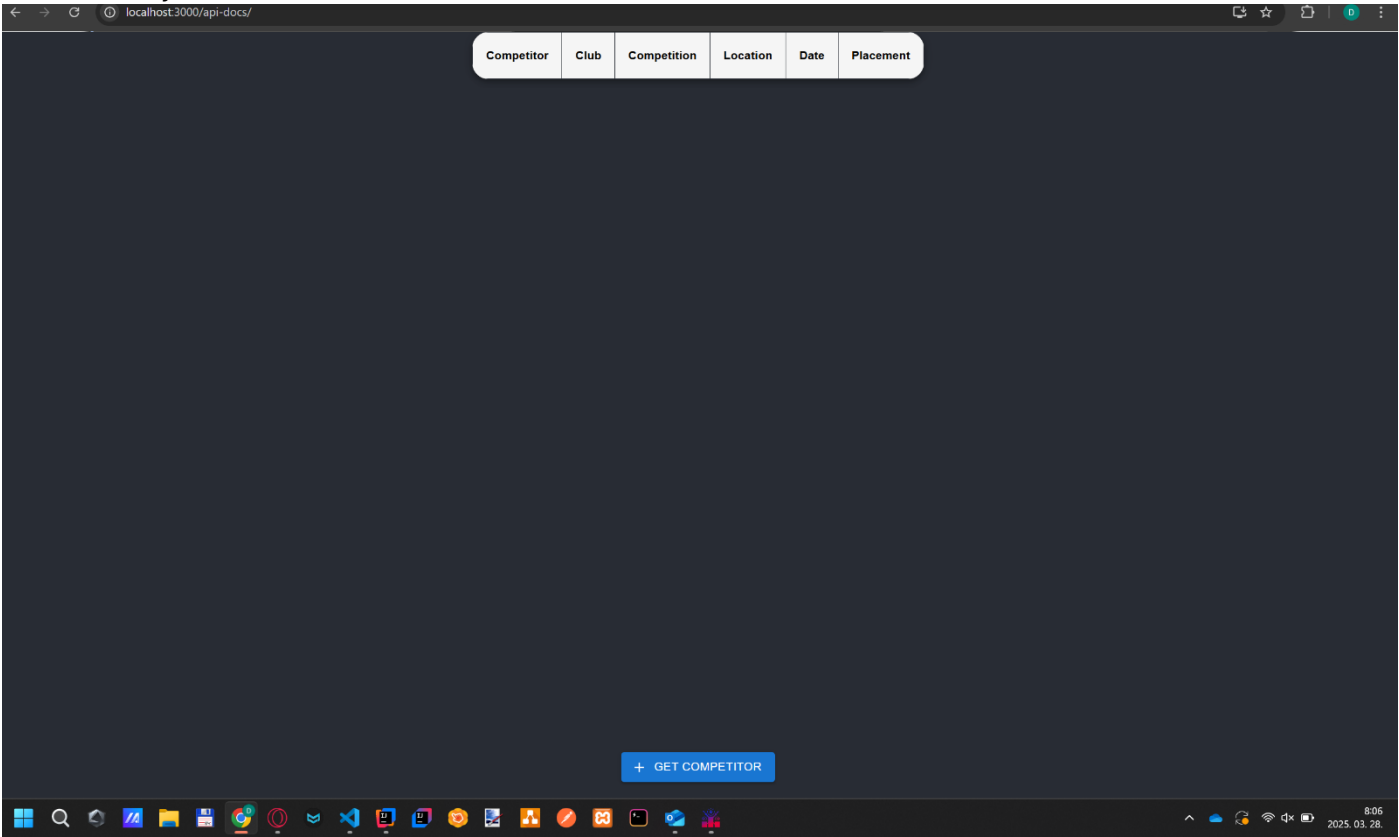
A körök felületi terve – A színek és azok variációi a végső verzióban eltérhetnek, a fentiek csupán tervek.

2+1 team fejlesztése

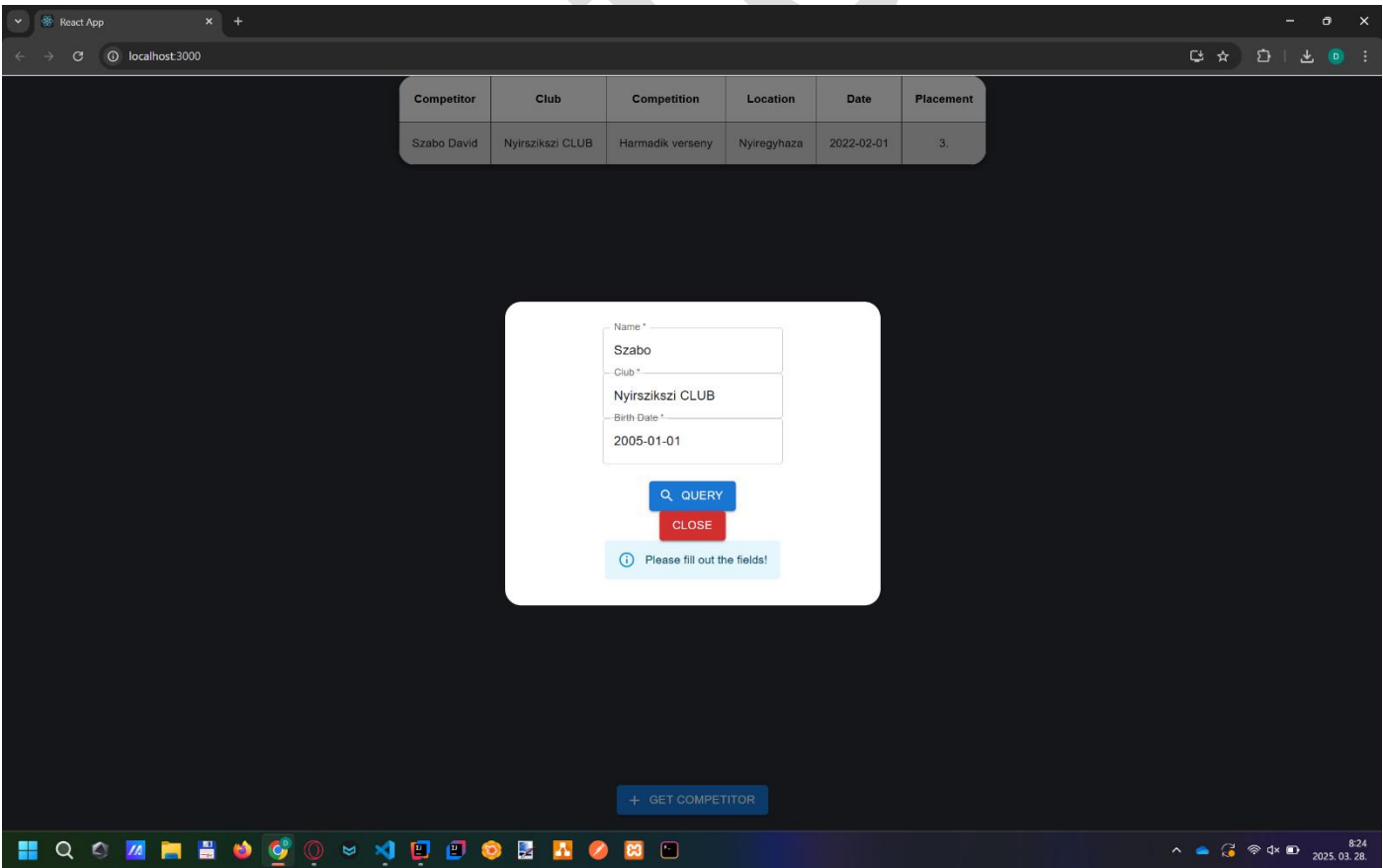
1	Szabo David				
		Szabo David			
8	---				
			Szabo David		
5	---				
		Lakatos Turret			
4	Lakatos Turret				
				Szabo David	
3	Szanto David				
		Szanto David			
6	---				
			Szurke Levente		
7	---				
		Szurke Levente			
2	zurke Levente				

A tábla felületi terve

2+1 team fejlesztése

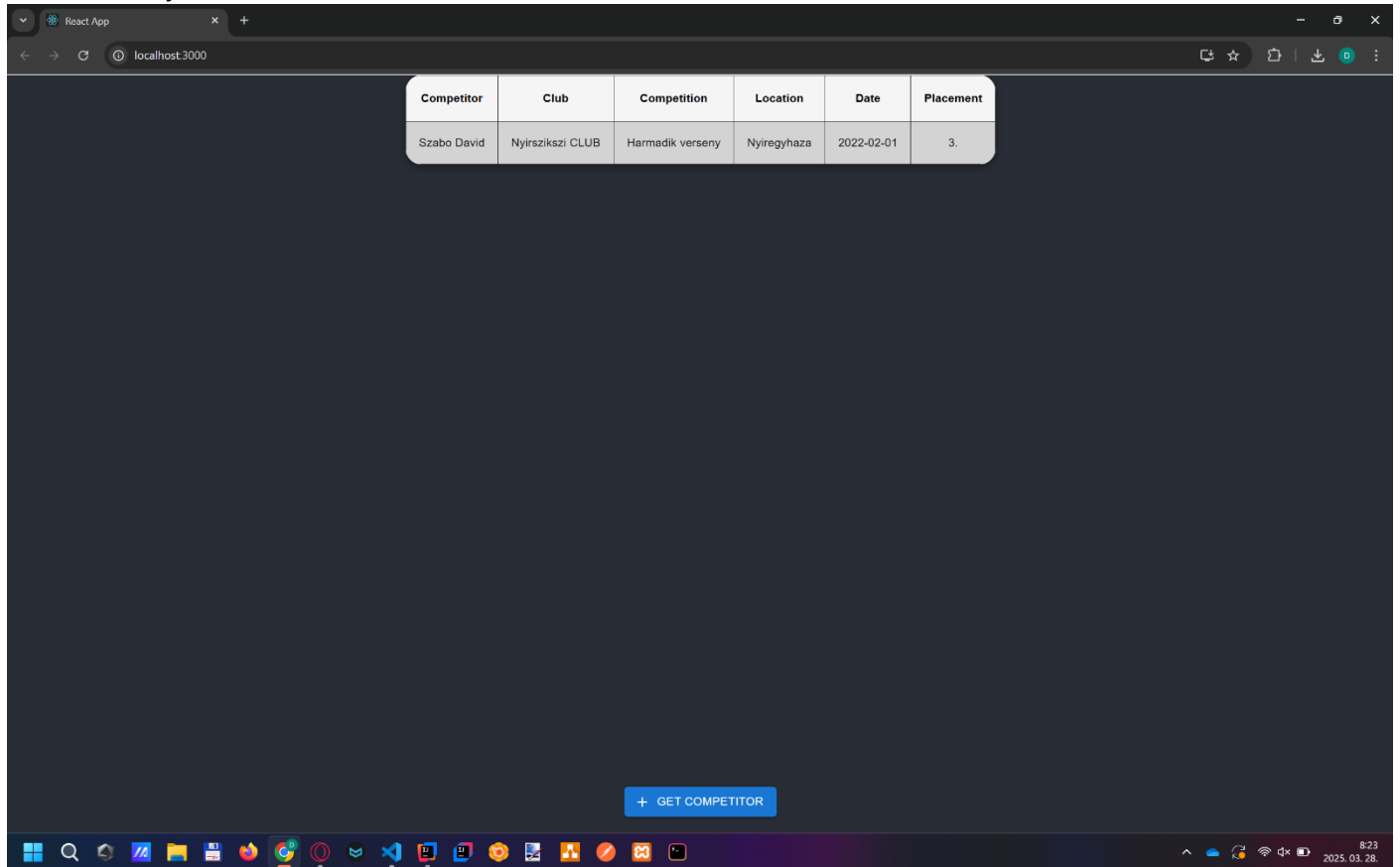


A weboldal kezdőoldalának sablonja



A weboldal adatbeviteli paneljének terve

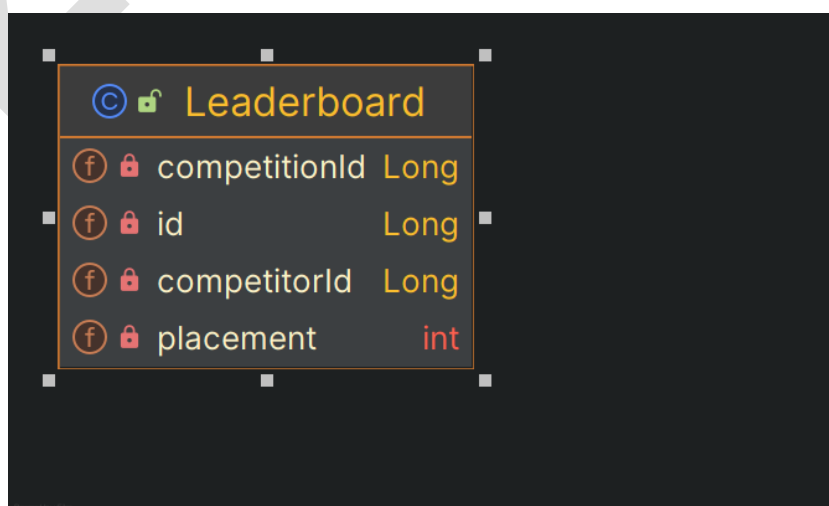
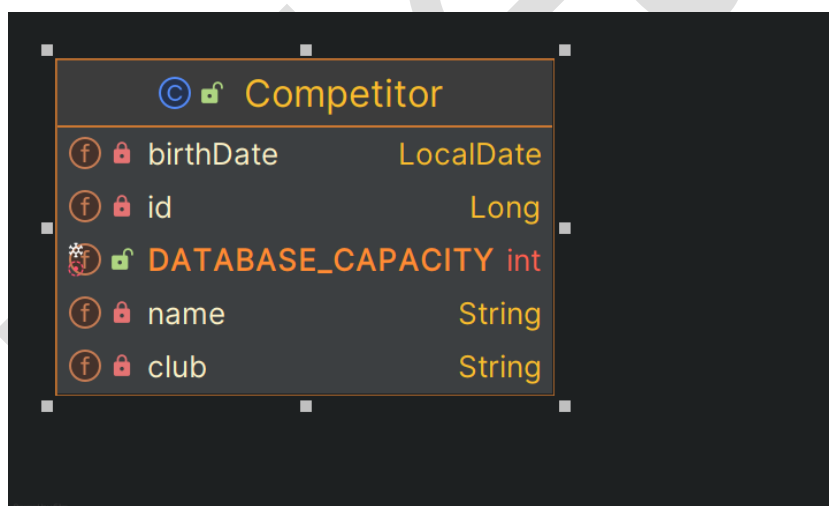
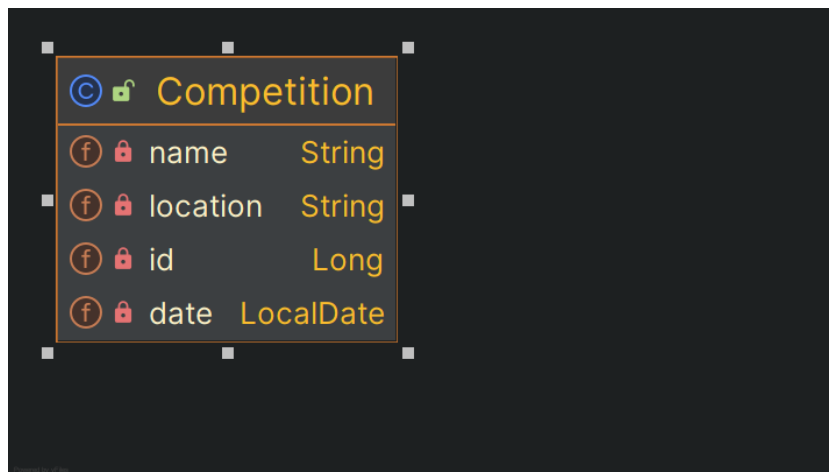
2+1 team fejlesztése



Az adatmegjelenítő táblázat terve

Modellek

Mivel versenyzőket és versenyeket, illetve a versenyzők egyes versenyein elért eredményeket szeretnénk tárolni, több modellre is szükségünk lesz.



Alkalmazások

Frontend: React, HTML

A React az egyik nyílt forráskódú JavaScript könyvtár. Interaktív felhasználói felületek felépítésére szolgál. Ez egy hatékony, deklaratív és rugalmas könyvtár. A Model-View- Controller (MVC) V, azaz View elemével foglalkozik. Ez nem egy teljes keret, hanem csak egy előlapi könyvtár. Lehetővé teszi összetett felhasználói felületek létrehozását vagy elkészítését, összetevőkként ismert, elkülönített és apró kóddarabok felhasználásával. Az alkotóelemek fő előnye, hogy az egyik alkotóelemre történő változtatás nem érinti a teljes alkalmazást.

A HTML (Hypertext Markup Language) egy leíró nyelv, melyet weboldalak elkészítésére használhatunk.

Forrás: <https://hu.education-wiki.com/9050114-what-is-react>

Backend: Springboot

A Java Spring Boot egy olyan, nyílt forráskódú eszköz, amely megkönnyíti a mikroszolgáltatások és webalkalmazások Java-alapú keretrendszerek használatával való létrehozását. A Boot bármely definíciója szerint a beszélgetésnek a Java nyelvvel kell kezdődnie, amely az alkalmazásfejlesztés egyik legnépszerűbb és legszélesebb körben használt fejlesztési nyelve és számítástechnikai platformja. A fejlesztők a világ minden táján a Java tanulásával kezdik meg a kódolás elsajátítását. A rugalmas és felhasználóbarát Java számos alkalmazás esetében a fejlesztők kedvence, a közösségi médiától, a webtől és a játékalalmazásoktól kezdve a hálózatkezelésig és a nagyvállalati alkalmazásokig.

Forrás: <https://azure.microsoft.com/hu-hu/resources/cloud-computing-dictionary/what-is-java-spring-boot>

Asztali alkalmazás: Java, Swing

A Java általános célú, objektumorientált programozási nyelv, amelyet a Sun Microsystems fejlesztett a '90-es évek elejétől kezdve egészen 2009-ig, amikor a céget felvásárolta az Oracle.

A Java Swing egy platformfüggetlen grafikus könyvtár. A benne található komponensek az MVC modellt követik.

Források:

[https://hu.wikipedia.org/wiki/Java_\(programoz%C3%A1si_nyelv\)](https://hu.wikipedia.org/wiki/Java_(programoz%C3%A1si_nyelv))

<https://www.inf.szte.hu/~zizo/html/alkfejl/swing.html>

Adatbázis: MySQL

Többfelhasználós, többszálú relációs adatbázis-kezelő szerver. A MySQL 1995-ben készítette a svéd MySQL AB. A MySQL adatbázis szerver a legnépszerűbb lett a webalkalmazások használatában. A MySQL AB kettős licenccel hozta forgalomba. Egy GNU GPL és egy zárt forráskódú változatban. Kezdetben csak Linuxra valósították meg.

Forrás: https://szit.hu/doku.php?id=oktatas:adatbazis-kezeles:mysql:mysql_jegyzet

Tesztelés: JUnit 5, Postman

A JUnit egy egységteszt-keretrendszer a Java programozási nyelvhez. A tesztvezérelt fejlesztés (TDD) szabályai szerint ez annyit tesz, hogy a kód írásával párhuzamosan fejlesztjük a kódot tesztelő osztályokat is (ezek az egységtesztek).

A Postman egy komplett eszköztár API fejlesztők részére. A programmal gyorsan és hatékonyan lehet dolgozni az API-val, mivel támogatja a fejlesztők minden munkafolyamatát, továbbá elérhető Mac OS X, Windows, Linux és Chrome felhasználók számára is.

Források:

<https://hu.wikipedia.org/wiki/JUnit>

<https://support.shoprenter.hu/hc/hu/articles/115014740187-Postman-haszn%C3%A1lata-%C3%A9s-p%C3%A9ldak%C3%B3k>

Routing

Összegzés

- Competitor
 - Get all
 - Get
 - New
 - Update
 - Delete
- Competition
 - Get all
 - Get
 - New
 - Update
- Leaderboard
 - Get all
 - Get
 - New
 - Update
- Test

Megjegyzés

A végpontok URL-jének kezdete a szoftver telepítése után változni fog, a fejlesztési idő alatt végig a következő lesz:

localhost:8080

Az API a fejlesztés és tesztelés ideje alatt dummy adatokkal szolgálja ki a klienseket.

Competitor

Get all

Az összes adatbázisban lévő **versenyző** lekérdezése.

2+1 team fejlesztése
URL: /api/competitor/

Body: Nincs

Használat: Admin Frontend

Válasz:

- ECONNREFUSED - Nem jött létre kapcsolat
 - Az összes rögzített versenyző egy tömbben
-

Get

Egy versenyző lekérdezése az adatbázisból **id** alapján.

URL: /api/competitor/{id}

Body: Nincs

Használat: Admin Frontend

Válasz:

- ECONNREFUSED - Nem jött létre kapcsolat
- 404 Not found

```
{  
  "message": "Competitor with '{id}' id not found",  
  "status": "NOT_FOUND"  
}
```

- A megadott id-vel rendelkező versenyző objektum
-

New

Új versenyző felvétele az adatbázisba.

URL: /api/competitor/new

Body:

```
{  
  "name": "",  
  "club": "",  
  "birthDate": ""
```

2+1 team fejlesztése
{

Használat: Admin

Válasz:

- ECONNREFUSED - Nem jött létre kapcsolat
 - A létrehozott versenyző objektum.
-

Update

Egy létező versenyző valamely adat(ainak) módosítása.

URL: /api/competitor/update/{id}

Body:

```
{  
  "name": "",  
  "club": "",  
  "birthDate": ""  
}
```

Használat: Admin

Válasz:

- ECONNREFUSED - Nem jött létre kapcsolat
- 404 Not found

```
{  
  "message": "Competitor with '{id}' id not found",  
  "status": "NOT_FOUND"  
}
```

- A megadott id-vel rendelkező versenyző objektum a frissített értékekkel
-

Delete

Egy létező versenyző törlése az adatbázisból.

URL: /api/competitor/delete/{id}

Body: Nincs

2+1 team fejlesztése

Használat: Admin

Válasz:

- ECONNREFUSED - Nem jött létre kapcsolat
- 404 Not found

{

"message": "Competitor with '{id}' id not found",

"status": "NOT_FOUND"

}

- 204 No content

Competition

Get all

Az összes adatbázisban lévő **verseny** lekérdezése.

URL: /api/competition/

Body: Nincs

Használat: Admin Frontend

Válasz:

- ECONNREFUSED - Nem jött létre kapcsolat
- Az összes rögzített verseny egy tömbben

Get

Egy verseny lekérdezése az adatbázisból **id** alapján.

URL: /api/competition/{id}

Body: Nincs

Használat: Admin Frontend

Válasz:

- ECONNREFUSED - Nem jött létre kapcsolat
- 404 Not found

{

2+1 team fejlesztése

"message": "Competition with '{id}' id not found",

"status": "NOT_FOUND"

}

- A megadott id-vel rendelkező verseny objektum

New

Új verseny felvétele az adatbázisba.

URL: /api/competition/new

Body:

```
{  
  "name": "",  
  "location": "",  
  "date": ""  
}
```

Használat: Admin

Válasz:

- ECONNREFUSED - Nem jött létre kapcsolat
- A létrehozott verseny objektum.

Update

Egy létező verseny valamely adat(ainak) módosítása.

URL: /api/competition/update/{id}

Body:

```
{  
  "name": "",  
  "location": "",  
  "date": ""  
}
```

Használat: Admin

Válasz:

- ECONNREFUSED - Nem jött létre kapcsolat
- 404 Not found

```
{  
"message": "Competition with '{id}' id not found",  
"status": "NOT_FOUND"  
}
```

- A megadott id-vel rendelkező verseny objektum a frissített értékekkel

Leaderboard

Get all

Az összes adatbázisban lévő **ranglista elem** lekérdezése.

URL: /api/leaderboard/

Body: Nincs

Használat: Admin Frontend

Válasz:

- ECONNREFUSED - Nem jött létre kapcsolat
- Az összes rögzített ranglista elem egy tömbben

Get

Egy ranglista elem lekérdezése az adatbázisból **id** alapján.

URL: /api/leaderboard/{id}

Body: Nincs

Használat: Admin Frontend

Válasz:**

- ECONNREFUSED - Nem jött létre kapcsolat
- 404 Not found

```
{  
"message": "Leaderboard with '{id}' id not found",
```

2+1 team fejlesztése
"status": "NOT_FOUND"

}

- A megadott id-vel rendelkező ranglista elem objektum

New

Új ranglista elem felvétele az adatbázisba.

URL: /api/leaderboard/new

Body:

```
{  
  "competitionId": "",  
  "competitorId": "",  
  "placement": ""  
}
```

Használat: Admin

Válasz:

- ECONNREFUSED - Nem jött létre kapcsolat
- A létrehozott ranglista elem objektum

Update

Egy létező ranglista elem valamely adat(ainak) módosítása.

URL: /api/leaderboard/update/{id}

Body:

```
{  
  "competitionId": "",  
  "competitorId": "",  
  "placement": ""  
}
```

Használat: Admin

Válasz:

2+1 team fejlesztése

- ECONNREFUSED - Nem jött létre kapcsolat
- 404 Not found

```
{  
"message": "Leaderboard with '{id}' id not found",  
"status": "NOT_FOUND"  
}
```

- A megadott id-vel rendelkező ranglista elem objektum a frissített értékekkel

Test

Az API-val való kapcsolat tesztelése.

Metódus: GET

URL: /api/test/

Body: Nincs

Használat: Admin Frontend

Válasz:

- ECONNREFUSED - Nem jött létre kapcsolat
- 202 Accepted

```
{  
"message": "Connection established"  
}
```

CompetitionPlacement

Komplex lekérdezése, mely versenyző id alapján vissza adja a versenyző nevét, az összes olyan versenyt, amelyen a versenyző résztvett (Verseny név, Verseny helyszín, Verseny dátum), és a helyezést amelyet a versenyző az adott versenyen elért.

Metódus: GET

URL: /api/placements/{id}

Body: Nincs

Használat: Admin `` Frontend

Válasz:

2+1 team fejlesztése

- ECONNREFUSED - Nem jött létre kapcsolat
- 404 Not found

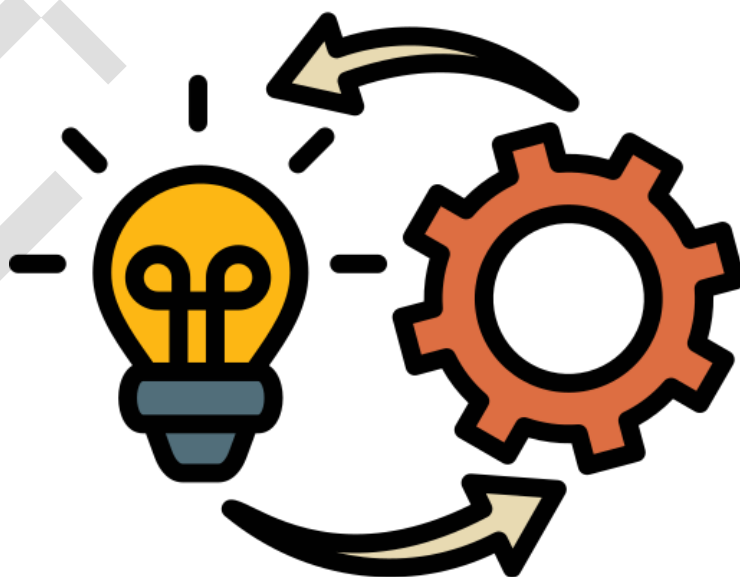
```
{  
"message": "Competitor with '{id}' id not found",  
"status": "NOT_FOUND"  
}
```

- 200

```
{  
"competitorName": "",  
"competitionName": "",  
"competitionLocation": "",  
"competitionDate": "",  
"placement": ""  
}
```

Implementáció

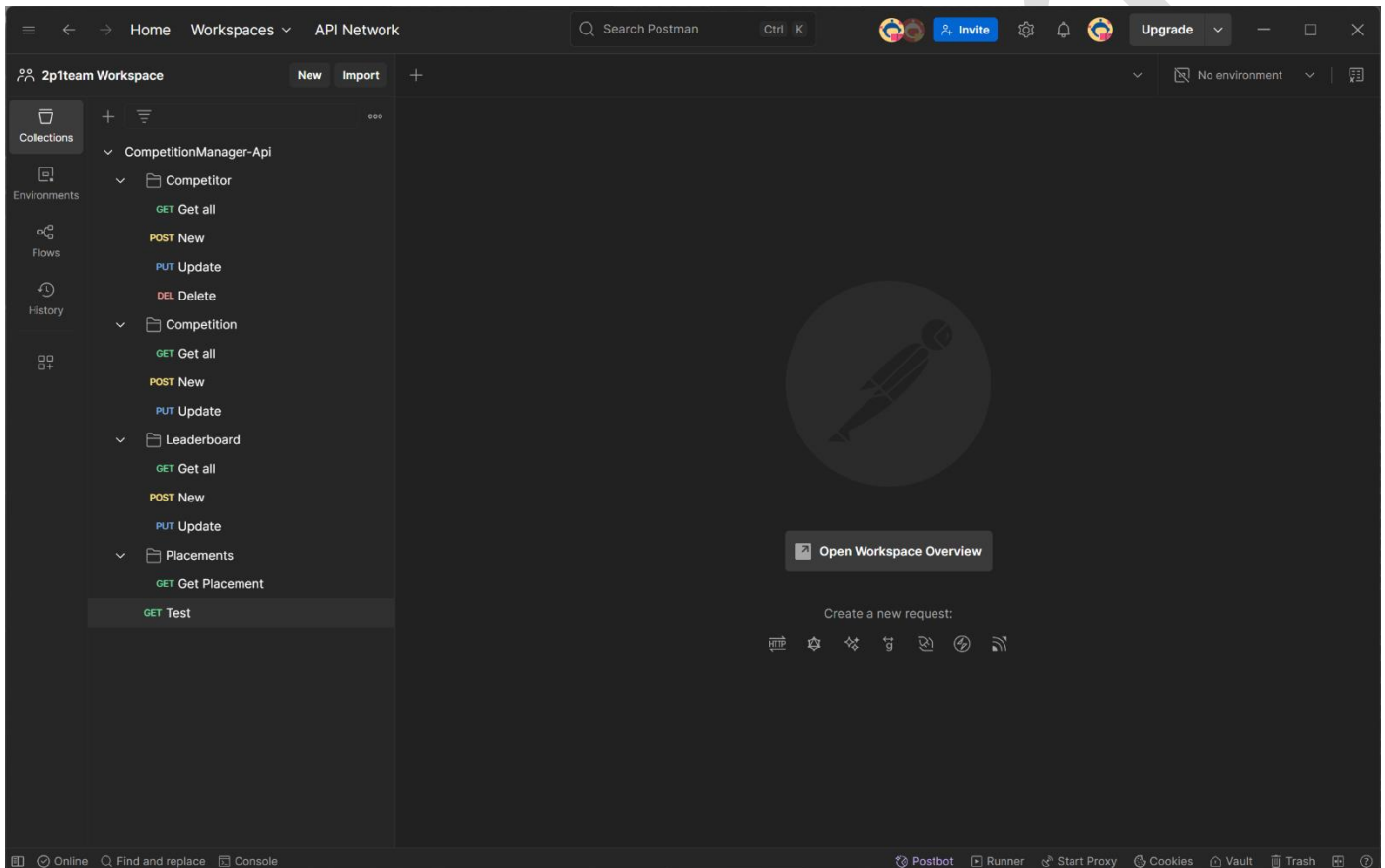
Az implementáció során a tervek valósággá válnak. Ez az a szakasz amikor a tervezett modelleket, UI-t és egyéb dolgot megvalósítunk és korrigálunk, ha szükséges.



Tesztelés

Az Admin felületet (asztali alkalmazást) használnak szüksége lesz a számítógépén egy Java futtató környezetre. A weboldal bármilyen böngészőben elfut, ezt teszteltük Google Chrome, Mozilla Firefox, Brave és Opera GX-szel is.

Szerver oldalról egy MySQL szerverre és szintén egy Java-ra lesz szükségünk. Ezek futtatása a fejlesztés alatt a legtöbbször egy gépen történt párhuzamosan a többi alkalmazással, de két alkalommal (csak helyi hálózaton) másik gépről volt futtatva az API és az adatbázis. Az API fejlesztése közben minden végpont elkészültekor teszteltük azt, melyet Postman-ben oldottunk meg.



Az asztali alkalmazás tesztelése JUnit 5-tel történt, illetve egy tesztelő grafikus ablak is fejlesztésre került melyben saját komponenseink megjelenését tesztelhattuk. Fontos volt még a versenykezelést tesztelnünk, mivel ez a program lényege, így ebben nem lehet hiba.

2+1 team fejlesztése

The image displays two screenshots of the Postman API client interface, showing the results of API tests performed on a 'CompetitionManager-API'.

Top Screenshot: GET all Competitor

- Collection:** CompetitionManager-API / Competitor / Get all
- Method:** GET
- URL:** localhost:8080/api/competitor/
- Status:** 200 OK
- Response Time:** 27 ms
- Response Size:** 640 B
- Body (JSON):**

```
[
  {
    "id": 9,
    "name": "V33",
    "club": "K3",
    "birthDate": "2002-01-01"
  },
  {
    "id": 10,
    "name": "V4",
    "club": "K4",
    "birthDate": "2003-01-01"
  },
  {
    "id": 11
  }
]
```

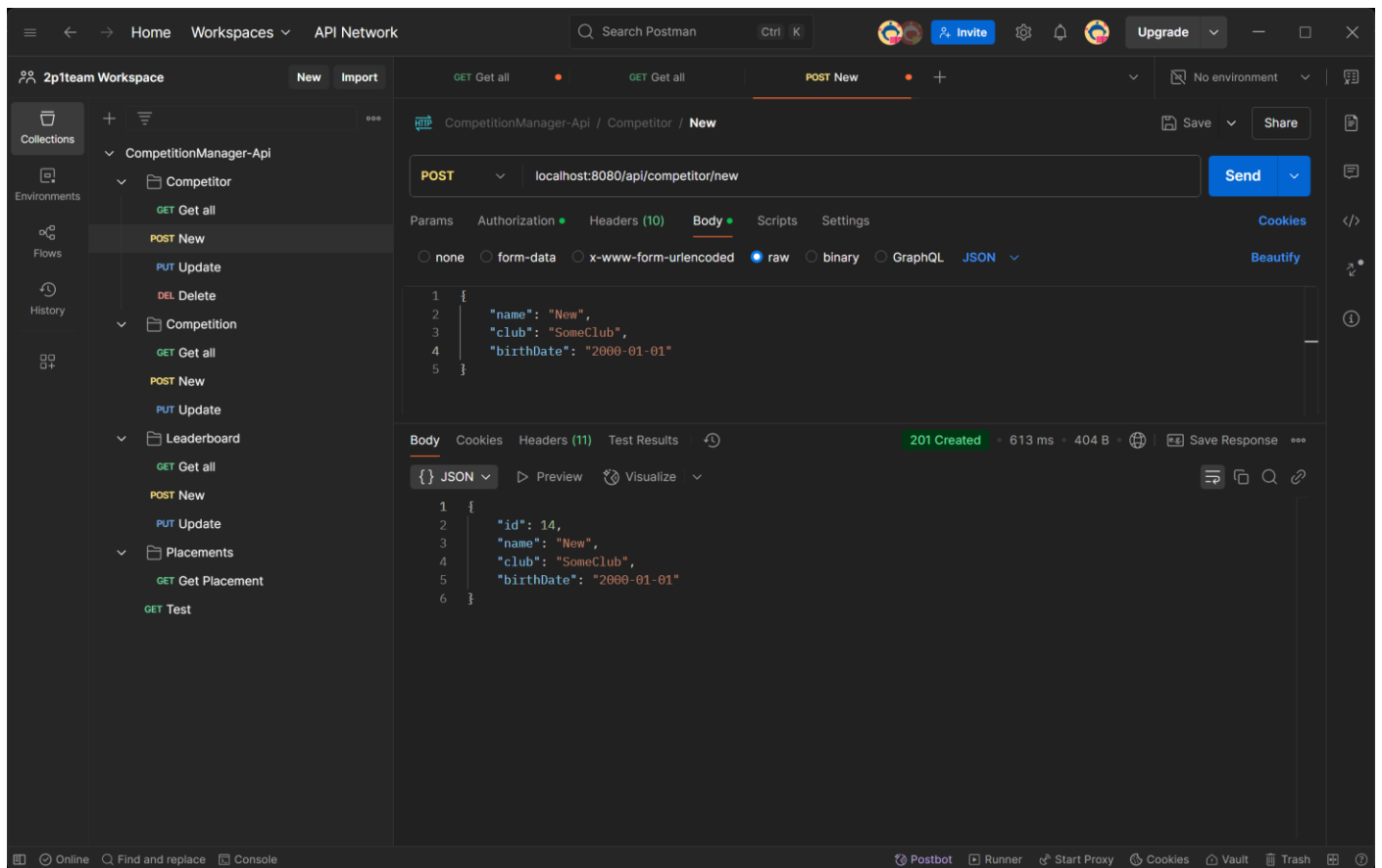
Bottom Screenshot: GET all Leaderboard

- Collection:** CompetitionManager-API / Leaderboard / Get all
- Method:** GET
- URL:** localhost:8080/api/leaderboard/
- Status:** 200 OK
- Response Time:** 32 ms
- Response Size:** 1.1 KB
- Body (JSON):**

```
[
  {
    "id": 14,
    "competitionId": 14,
    "competitorId": 9,
    "placement": 4
  },
  {
    "id": 31,
    "competitionId": 15,
    "competitorId": 9,
    "placement": 2
  },
  {
    "id": 32,
    "competitionId": 15,
    "competitorId": 10,
    "placement": 3
  },
  {
    "id": 36,
    "competitionId": 16,
    "competitorId": 10,
    "placement": 3
  }
]
```

Az API tesztelése Postman-ben

2+1 team fejlesztése



Új versenyző felvételének tesztelése Postman-ben

Éles üzemmód

A projektben elkészült szoftverek futtatásához szükséges a Java, mely a következő oldalról letölthető: <https://www.java.com/en/download/>. Töltsük le, majd indítsuk el a telepítőt és kövessük az utasításokat.

Az adatbázishoz szükségünk lesz a XAMPP-ra. Ezt a következő linken tudjuk letölteni: <https://www.apachefriends.org/hu/index.html>. Töltsük le, majd indítsuk el a telepítőt és kövessük az utasításokat.

A Node.js oldaláról töltsük le az általunk használt operációs rendszerhez készült telepítőt, majd futtassuk azt.

Visual Studio Code-ban, vagy terminálban (cmd) navigáljunk a Frontend mappájába, majd írjuk be az npm -i parancsot, hogy települjenek a kellő könyvtárak. Ha VS Code-ban nyitottuk meg, nyissunk azon belül egy terminált.

Indítsuk el a XAMPP-ot majd indítsuk el a MySQL-t a mellette lévő „Start” gombbal. Ugyan ezt tegyük meg az Apache-al is, majd nyomjunk az „Admin” gombra és importáljuk a kapott adatbázist.

Ezután a Frontend-nél írjuk be az npm start parancsot, ezzel elindul a weboldal.

Az Admin futtatásához csupán dupla kattintás az alkalmazás ikonjára.

Összegzés

Projektünk célja volt a vívók egy helyre gyűjtése, egy nagyobb, összetartóbb közösség érdekében. A versenyek, eredmények egy helyen való prezentálása megkönnyíti a klubok közötti barátságos versengést is. Ehhez azonban stabil és biztonságos rendszert kellett fejlesztenünk mely kihívásokkal járt. Fontos volt, hogy az összes felhasználói felület a lehető legletisztultabb és könnyen kezelhetőbb legyen. Az asztali alkalmazásnál különösen odafigyeltünk a versenykezelés helyességére és letisztult megjelenítésére. A weboldalt igyekeztünk minél reszponzívabbá tenni és nem elrejteni a lényegét rajta.

A rövid fejlesztési idő azonban valamennyire megkötötte a kezünket, így maradtak még fejlesztési lehetőségek. Ilyenek például a csapatversenyek kezelése, a klubok és verseny helyszínek külön entitássá alakítása (csak kiválasztani kelljen, ne beírni), vagy a versenyek valós időben történő webes megjelenítése. A versenyzők teljesítményéről statisztikát is lehetne készíteni, weben megjeleníteni. Az asztali alkalmazásból kimaradt az ablak funkció, aminek bár a gombja megtalálható, nem csinál semmit. Funkciója az ablak méretének állítása lenne, illetve, hogy melyik képernyőn jelenjen meg alapértelmezetten (amennyiben több van mint 1).

Az Admin asztali alkalmazás Git repository-ja:

<https://github.com/szdav1/CompetitionManager-Admin>

<https://github.com/szdav1/CompetitionManager-Admin-Application> - Futtatható állomány

Az API Git repository-ja:

<https://github.com/szdav1/CompetitionManager-API>

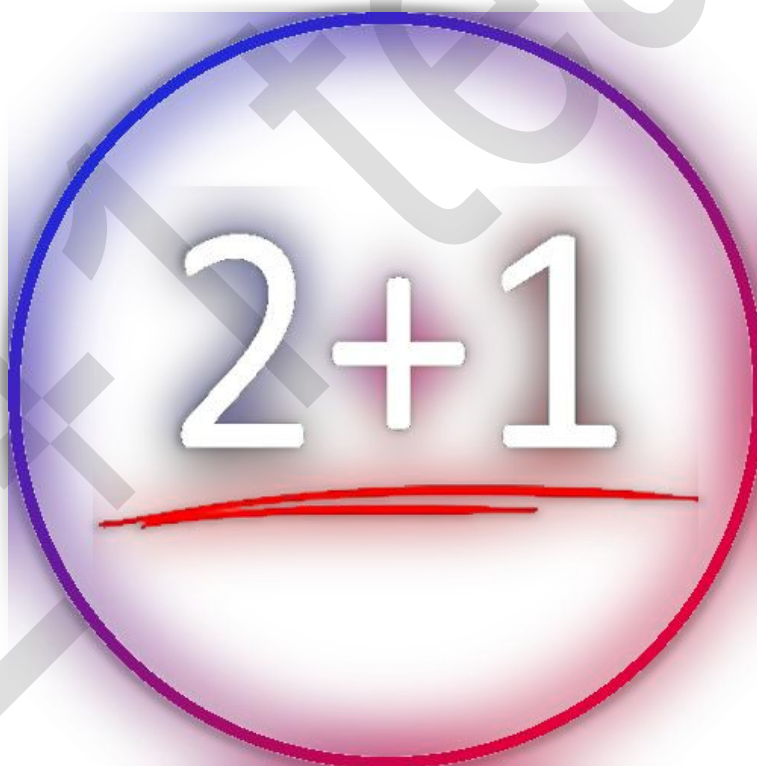
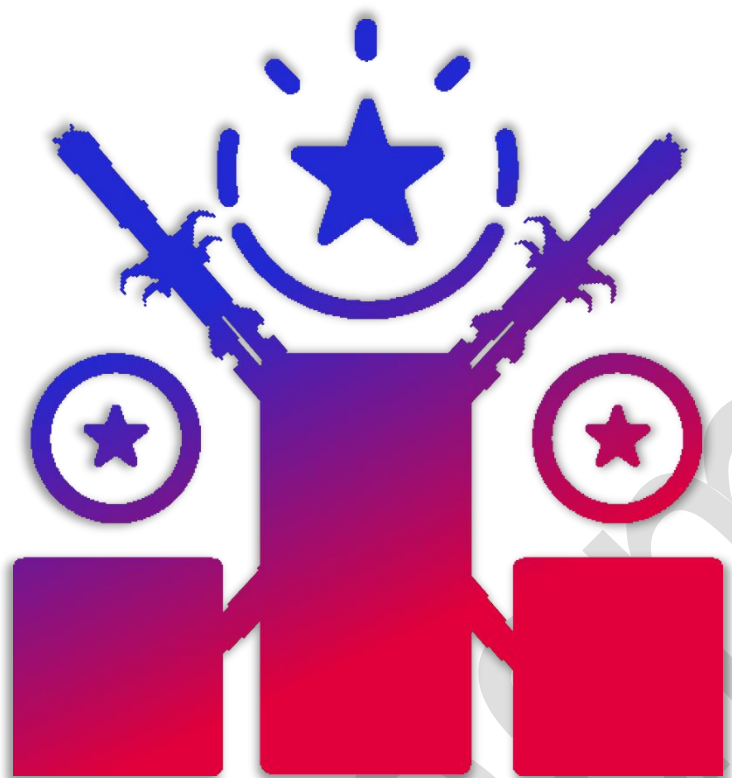
<https://github.com/szdav1/CompetitionManager-API-Application> - Futtatható állomány

A Frontend Git repository-ja:

<https://github.com/szdav1/CompetitionManager-Frontend>

A teljes dokumentáció, tervek, rajzok:

<https://github.com/szdav1/CompetitionManager-Docs>



NYÍREGYHÁZA

2025