** **

**ZÁRÓDOLGOZAT**

Készítették:

Nagy Levente Ferenc – Veller Árpád – Csanálosi Bálint

Konzulens:

Kasza László Róbert

Miskolc

2025.

Miskolci SZC Kandó Kálmán Informatikai Technikum

Miskolci Szakképzési Centrum

SZOFTVERFEJLESZTŐ- ÉS TESZTELŐ SZAK

Ephemeral Courage

videójáték

Nagy Levente Ferenc – Veller Árpád – Csanálosi Bálint

2024-2025

# Bevezetés

Amikor ki kellett választani, hogy mi legyen a vizsga projektünk, mind a hárman valamilyen játékot szerettünk volna megvalósítani. Közös megegyezés alapján egy stratégiai lövöldözős top-down(2.5D) videójáték megvalósítására jutottunk. Ezalatt az év alatt meg kellett ismernünk a Unity motort, mivel ez nyújtotta a megfelelő programozási nyelv mellett (C#) és a szükséges eszköztárat.

Ebben a dokumentációban arról lesz szó, hogy milyen eszközökkel lett elkészítve a videójátékunk pre-alpha verziója, és a játékot kiszolgáló weboldal, API és adatbázis.

# Programozási nyelvek és technológiák



A HTML (HyperText Markup Language) egy leíró nyelv, melyet weboldalak készítéséhez fejlesztettek ki, és mára már internetes szabvánnyá vált a W3C támogatásával.



A CSS (Cascading Style Sheets) a számítástechnikában egy stílusleíró nyelv, mely a HTML vagy XHTML típusú strukturált dokumentumok megjelenését írja le. A CSS-t a weblapok szerkesztői és olvasói egyaránt használhatják, hogy átállítsák vele a lapok színét, betűtípusait, elrendezését, és más megjelenéshez kapcsolódó elemeit.



A JavaScript programozási nyelv egy objektumorientált, prototípus-alapú szkriptnyelv, amelyet weboldalakon elterjedten használnak. A JavaScript esetében a futási környezet jellemzően egy webböngésző, illetve annak JavaScript-motorja.

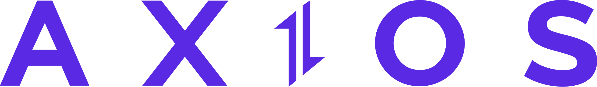
JavaScript alkalmazások futtathatók továbbá a böngészőn kívül is, Node.js vagy Deno segítségével, melyek Windows, macOS és Linux alapú operációs rendszerekkel is kompatibilisek.



A Bootstrap egy ingyenes és nyílt forráskódú CSS keretrendszer, amely a reszponzív, mobilra optimalizált front-end webfejlesztésre összpontosít. HTML, CSS és (opcionálisan) JavaScript alapú tervezési sablonokat tartalmaz tipográfiához, űrlapokhoz, gombokhoz, navigációhoz és egyéb felhasználói felület elemekhez.



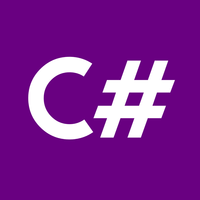
A React (React.js vagy ReactJS) egy ingyenes és nyílt forráskódú front-end JavaScript könyvtár, amelynek célja, hogy a komponenseken alapuló felhasználói felületek létrehozása „zökkenőmentesebbé” váljon.



Az Axios egy egyszerű, ígéreteken(promise) alapuló HTTP kliens a böngészőhöz és a Node.js-hez. Az Axios egy könnyen használható könyvtárat biztosít egy kis csomagban, nagyon kiterjeszthető interfésszel.



A Unity egy videójáték-motor, amelyet a Unity Technologies fejleszt. A Unity segítségével háromdimenziós illetve kétdimenziós videójátékokat, ezen kívül egyéb interaktív jellegű tartalmakat lehet létrehozni, például építészeti látványterveket vagy valós idejű háromdimenziós animációkat.



A C# a Microsoft által a .NET keretrendszer részeként kifejlesztett objektumorientált programozási nyelv. A C# az a programozási nyelv, ami a legközvetlenebb módon tükrözi az alatta működő, minden .NET programot futtató .NET keretrendszert, valamint erősen függ is attól: nincsen nem menedzselt, natív módban futó C# program.



A phpMyAdmin egy nyílt forrású eszköz, amit PHP-ban írtak a MySQL menedzselésére az interneten keresztül. Jelenleg képes készíteni és eldobni adatbázisokat, készíteni/eldobni/módosítani táblákat, törölni/módosítani/hozzáadni mezőket, SQL parancsokat futtatni és a mezőkön kulcsokat kezelni.



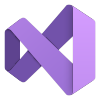
A XAMPP egy szabad és nyílt forrású platformfüggetlen webszerver-szoftvercsomag. Ez a szoftvercsomag egy integrált rendszert alkot, amely webes alkalmazások készítését, tesztelését és futtatását célozza, és ehhez egy csomagban minden szükséges összetevőt tartalmaz. A rendszer egyik nagy előnye az összehangolt elemek könnyű telepíthetősége.



A MySQL az egyik legelterjedtebb adatbázis-kezelő, aminek egyik oka lehet, hogy a teljesen nyílt forráskódú ,egyszerűen beállítható megoldást ad dinamikus webhelyek szolgáltatására.



A Visual Studio Code egy ingyenes, nyílt forráskódú kódszerkesztő, melyet a Microsoft fejleszt Windows, Linux és macOS operációs rendszerekhez.



A Visual Studio a Microsoft több programozási nyelvet tartalmazó fejlesztőkörnyezete, amely az évek során egyre több új programnyelvvel bővült. Jelenleg a F#, C++, C# és Visual Basic programozási nyelveket, valamint az XML-t támogatja. A csomag része még a MASM (Microsoft Macro Assembler) is, ami részleges assembly támogatást biztosít.

## Segéd Nuget Packagek



**Pomelo Microsoft.EntityFramework**



**Bcrypt** **Otp.NET**



**MailKit**



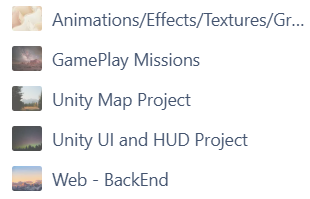
Az ASP.NET egy nyílt forráskódú, szerveroldali webalkalmazás-keretrendszer, amelyet webfejlesztésre terveztek dinamikus weboldalak előállítására. A Microsoft fejlesztette ki, hogy lehetővé tegye a programozóknak dinamikus webhelyek, alkalmazások és szolgáltatások létrehozását.

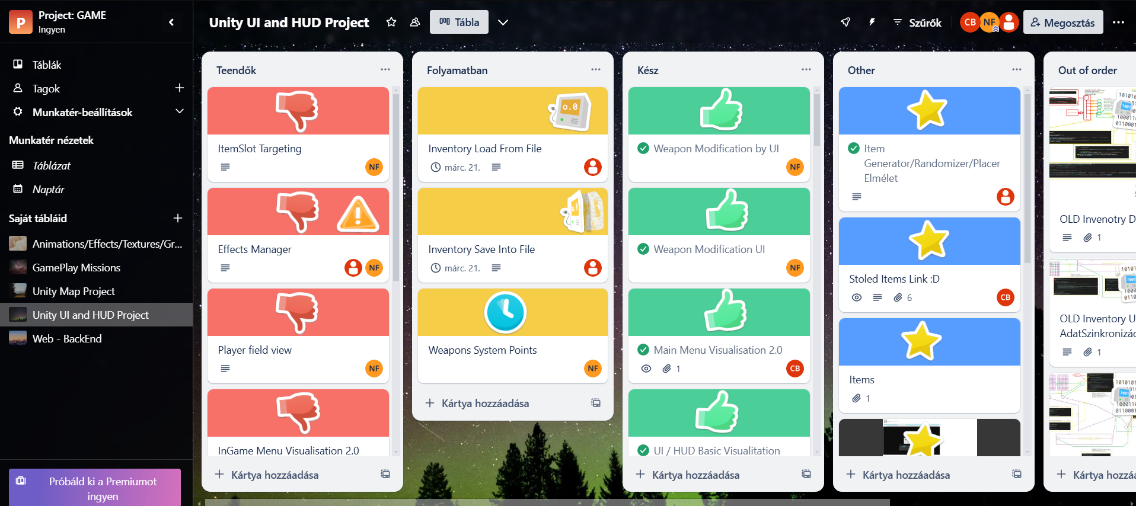
# Csapatmunka

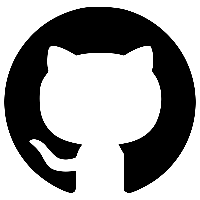
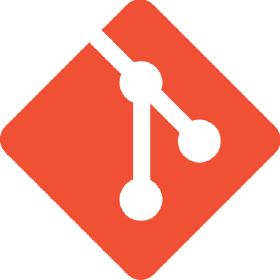


A Trellot használtuk a feladatok felosztására, kiadására, bizonyos dolgok tárolására. A Trello egy webalapú, Kanban stílusú listakészítő alkalmazás, amely az Atlassian leányvállalata.

Feladatainkat 5 külön táblára szedtük szét. Vannak olyan táblák, amin közösen dolgozunk, és olyanok amelyeken csak 1 ember dolgozik.



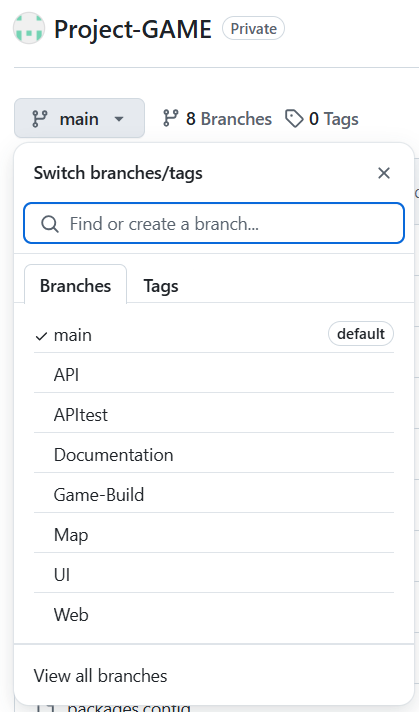




A GitHubot verziókövetésre használtuk a Git segítségével, a legfrissebb kódjainkat ide töltöttük fel.

A GitHub, Inc. egy egyesült államokbeli nemzetközi vállalat, amely a Git segítségével szoftverfejlesztési verziókövetés-szolgáltatást nyújt.

A Git egy nyílt forráskódú, elosztott verziókezelő szoftver, vagy másképpen egy szoftverforráskód-kezelő rendszer, amely a sebességre helyezi a hangsúlyt.

A Repositorynk 8 branchet tartalmaz. Minden tagnak megvan a saját branche és közös branchek is vannak, amelyek különállóak

# MySQL adatbázis (Database)

## Az adatbázis fogalma:

A számítástechnikában az **adatbázis** elektronikusan tárolt és elérhető adatok szervezett gyűjteménye. A kis adatbázisok fájlrendszerben tárolhatók, míg a nagy adatbázisok clusterekben vagy felhőalapú tárolókban vannak elhelyezve. Az adatbázisok tervezése kiterjed a formális technikákra és a gyakorlati megfontolásokra, beleértve az adatmodellezést, a hatékony adatreprezentációt és tárolást, a lekérdezési nyelveket, az érzékeny adatok biztonságát és védelmét, valamint az elosztott számítástechnika kérdéseit, beleértve az egyidejű hozzáférés és a hibatűrés támogatását.

## Mi az SQL?

Az **SQL**-adatbázisok, más néven relációs adatbázisok olyan rendszerek, amelyek táblagyűjteményeket tárolnak, és táblázatos, oszlopok és sorok formájában rendszerezik a strukturált adathalmazokat, mint a számolótáblák. Az adatbázisok strukturált lekérdezési nyelvvel (SQL-lel) készülnek, amely nem csupán az összes relációs adatbázist és relációsadatbázis-kezelő rendszert (RDBMS-t) alkotja, hanem lehetővé teszi, hogy „kommunikáljanak egymással”.

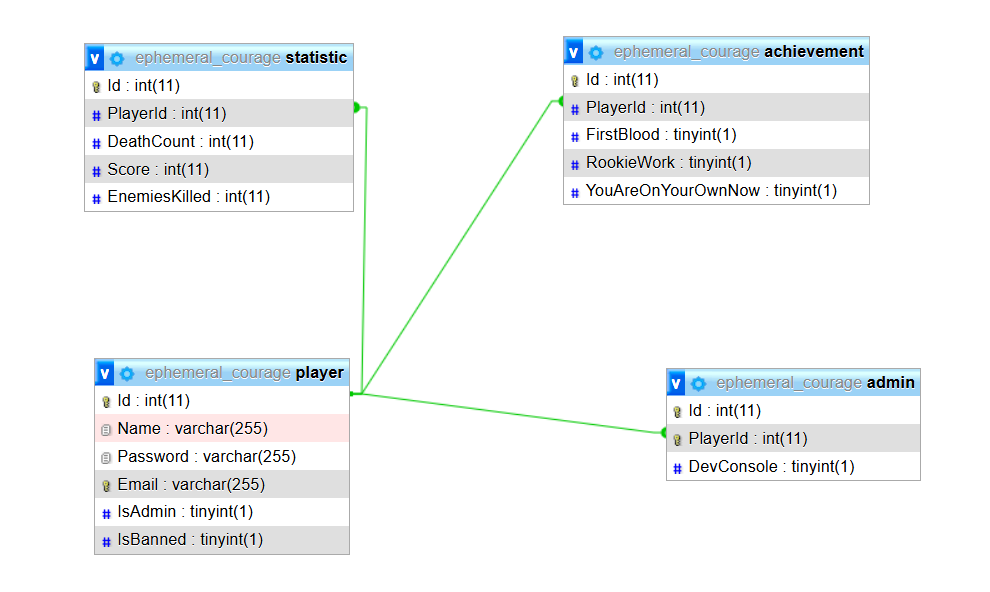
**Relációs** adatbázisnak nevezzük a relációs adatmodell elvén létrehozott adatok összességét, a relációs adatmodell fogalomrendszerében meghatározott ún. relációk egy véges halmazát. Relációs adatbázisokat relációsadatbázis-kezelőkkel hozhatunk létre, szerkeszthetünk és törölhetünk.

## A mi adatbázisunk

A mi projektünkhöz a **PHPMYADMIN** nevű adatbázis kezelő szoftvert használjuk,

aminek a működéséhez szükséges az **XAMPP** control panell.

### Az adatbázis terv:



# Backend

# (Web API, ASP.NET CORE)

## Az API jelentése

Az API-k közvetítőként működnek, lehetővé téve két különböző alkalmazás számára, hogy kommunikáljanak egymással. Ez a kommunikáció magában foglalhatja az adatok küldését és lehívását, vagy lehetővé teheti, hogy a különböző szoftverkomponensek interakcióba lépjenek egymással és feladatokat hajtsanak végre.

Egyértelmű módszerek és eszközök meghatározásával az API-k biztosítják, hogy a különböző szoftverek a mögöttes architektúrájuktól vagy technológiájuktól függetlenül megbízhatóan működjenek együtt. Emiatt az API-k döntő szerepet játszottak az internet növekedésében, lehetővé téve a felhőszolgáltatásokat, a mobilalkalmazások fejlesztését és a webalapú szolgáltatások integrációját. Napjainkban az API-k sokkal többek, mint az egyszerű kódkönyvtárak. Kifinomult eszközök, amelyek meghatározzák, hogyan működünk együtt a technológiával. 

Az API-k korlátozott számú olyan művelet és adatpont feltárásával működnek, amelyekkel a külső szoftverek kölcsönhatásba léphetnek. Ha egy szoftverrendszer hozzá szeretne férni egy másik rendszer által biztosított erőforráshoz (például adatokhoz vagy funkciókhoz), akkor kérelmet küld, amelyben részletezi, hogy milyen műveletet kell végrehajtania. Ezt a kérést egy API-n keresztül lehet intézni. Ha az API engedélyezett, a rendszer feldolgozza a kérelmet, és visszaküldi a választ.

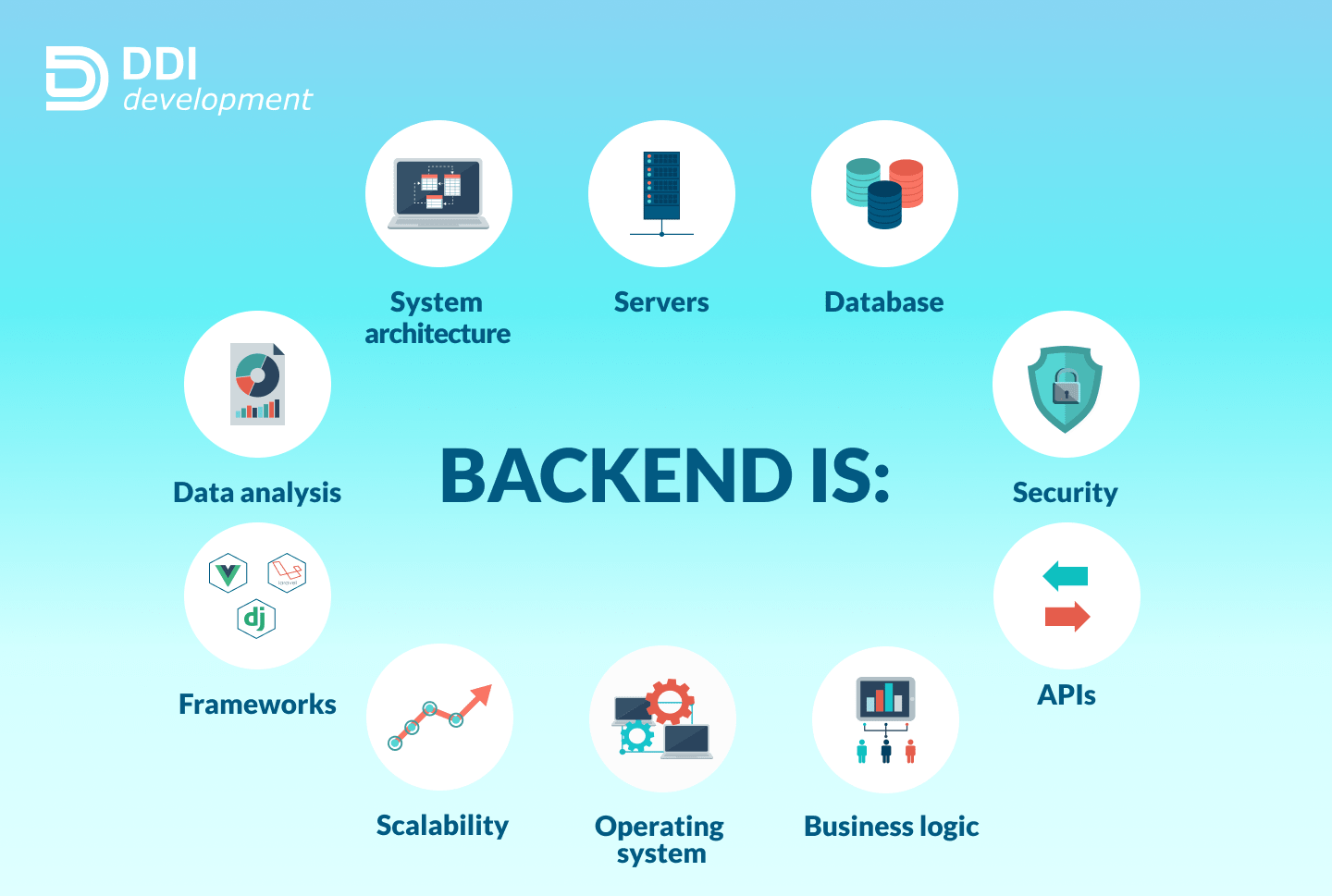
## Alapvető Komponensek és szerkezet

Az API alapvető összetevői és struktúrája végpontokat, metódusokat, kéréseket és válaszokat tartalmaz. A végpontok konkrét címek (web-API-khoz tartozó URL-címek), ahol API-k érhetők el. A metódusok azok a megengedett műveletek (például GET, POST, PUT, DELETE for HTTP API-k), amelyek ezeken a végpontokon hajthatók végre. A kérelmek tartalmazzák a művelet végrehajtásához szükséges adatokat és paramétereket, míg a válaszok az API által visszaadott adatokat.

Napjainkban az API-k tervezése elsősorban az API-interakció szabályait, valamint az API formázásának, érvényesítésének és dokumentálásának módját meghatározó séma alapján történik. A rendszerek közötti biztonságos, megbízható és hatékony kommunikáció biztosítására strukturált API-k magukban foglalják a rutinok, adatstruktúrák, objektumosztályok és változók specifikációit.

## A backend jelentése

A back-end fejlesztés eredménye az a kód, amely összeköti a webes**(jelen esetben a videójátékot is)** felületet egy adatbázissal, kezeli a felhasználói kapcsolatokat, és magát a webalkalmazást is működteti. A Backend fejlesztési nyelvek a webes alkalmazások „kulisszák mögötti” funkcióit kezelik.

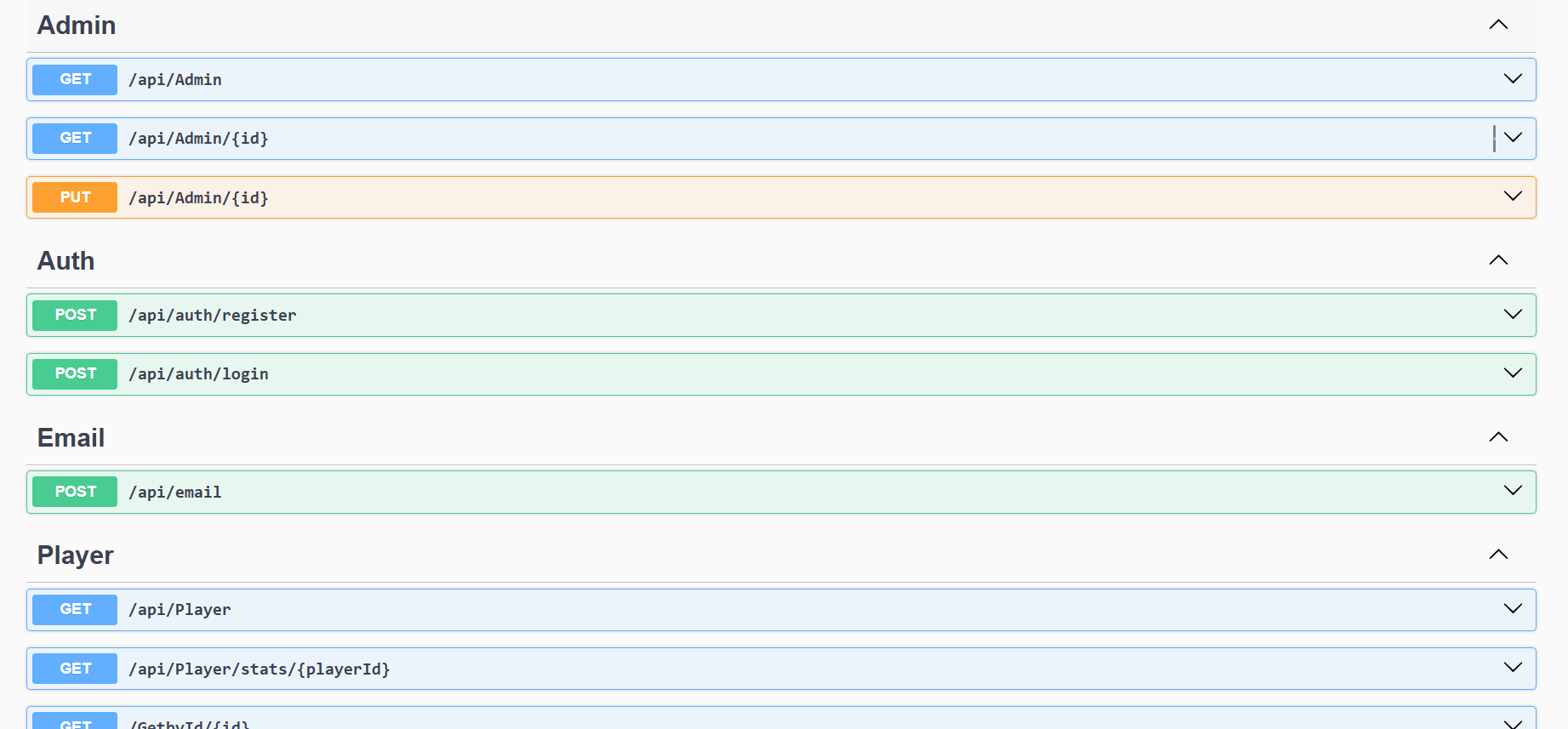


A Backend fejlesztés lényeges tulajdonsága, hogy a Frontenddel párhuzamosan működik úgy, hogy a felhasználó lehetőleg semmit se vegyen észre a háttérben folyó munkákból. A Backend fejlesztők elsősorban arra összpontosítanak, hogyan működik egy weboldal.

Ők olyan kódot írnak, amely az általuk fejlesztett alkalmazás működését és logikáját szolgálja, és a technológia, amelyen dolgoznak, soha nem látható közvetlenül a felhasználók számára.

A Backend programozók feladatai közé tartozhat API-k írása (Application Programming Interface, az adott program utasításait használó eszköz, ami jelentősen gyorsítja a fejlesztést), az adatbázissal való interakcióhoz szükséges kód megírása, könyvtárak létrehozása, üzleti folyamatokon és adatarchitektúrán való munka, és még sok más. Ez gyakran az adott szereptől és a vállalattól függ.

## A mi Swaggerünk:



A Swagger az OpenAPI specifikáción alapuló RESTful API-k létrehozására és dokumentálására szolgáló eszköz. Az OpenAPI specifikációnkat megtalálja itt. A paraméter formátuma lehet json/yaml, a válasz ekvivalens módon application/json vagy application/x-yaml formátumú.

# Frontend

# (React)

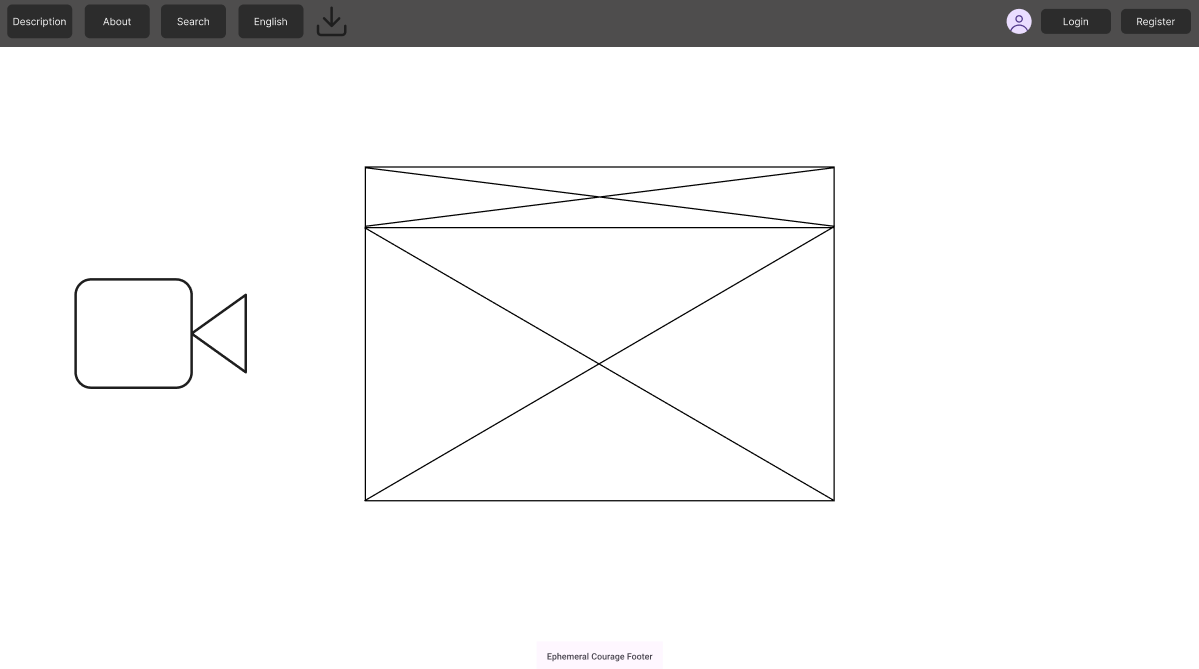
## Mit jelent a frontend?

A frontend a webfejlesztésnek az a rétege, ami kívülről látszik, amivel a felhasználó találkozik és interakcióba lép. Ennek a része a weboldal vizuális megjelenése, a navigáció, a gombok, a hivatkozások, a különböző felhasználó által kiváltott események, stb.

A mi frontendünkön a felhasználók megtudják nézni a statisztikáikat, le tudják tölteni a játékot bejelentkezés után, változtatni tudják az adataikat, törölni a fiókjaikat. Admin jogosultsággal rendelkező felhasználók külön opciókhoz jutnak hozzá, a sima felhasználók játékjogosultságát is befolyásolni tudják.

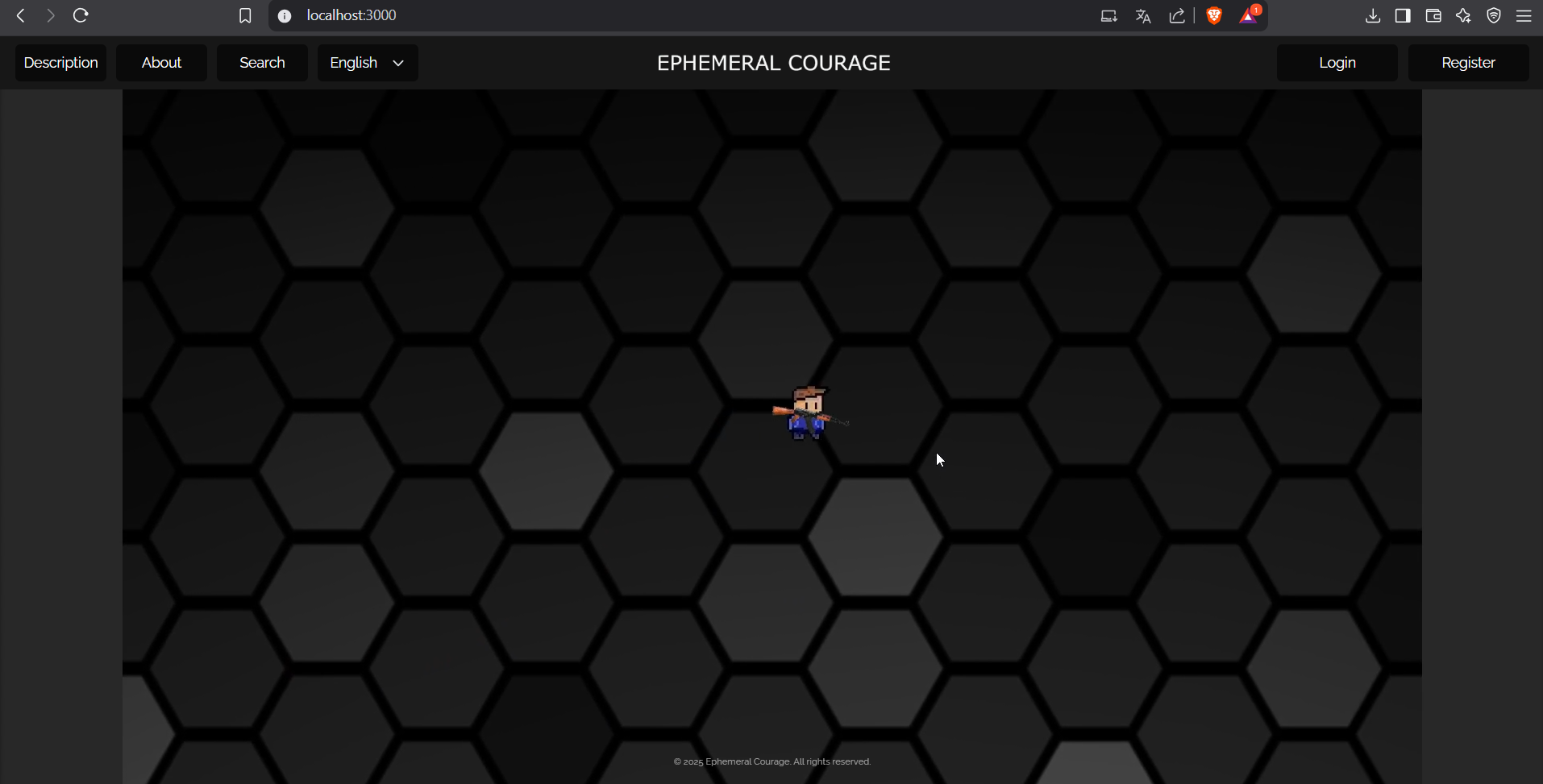
## Rácsterv

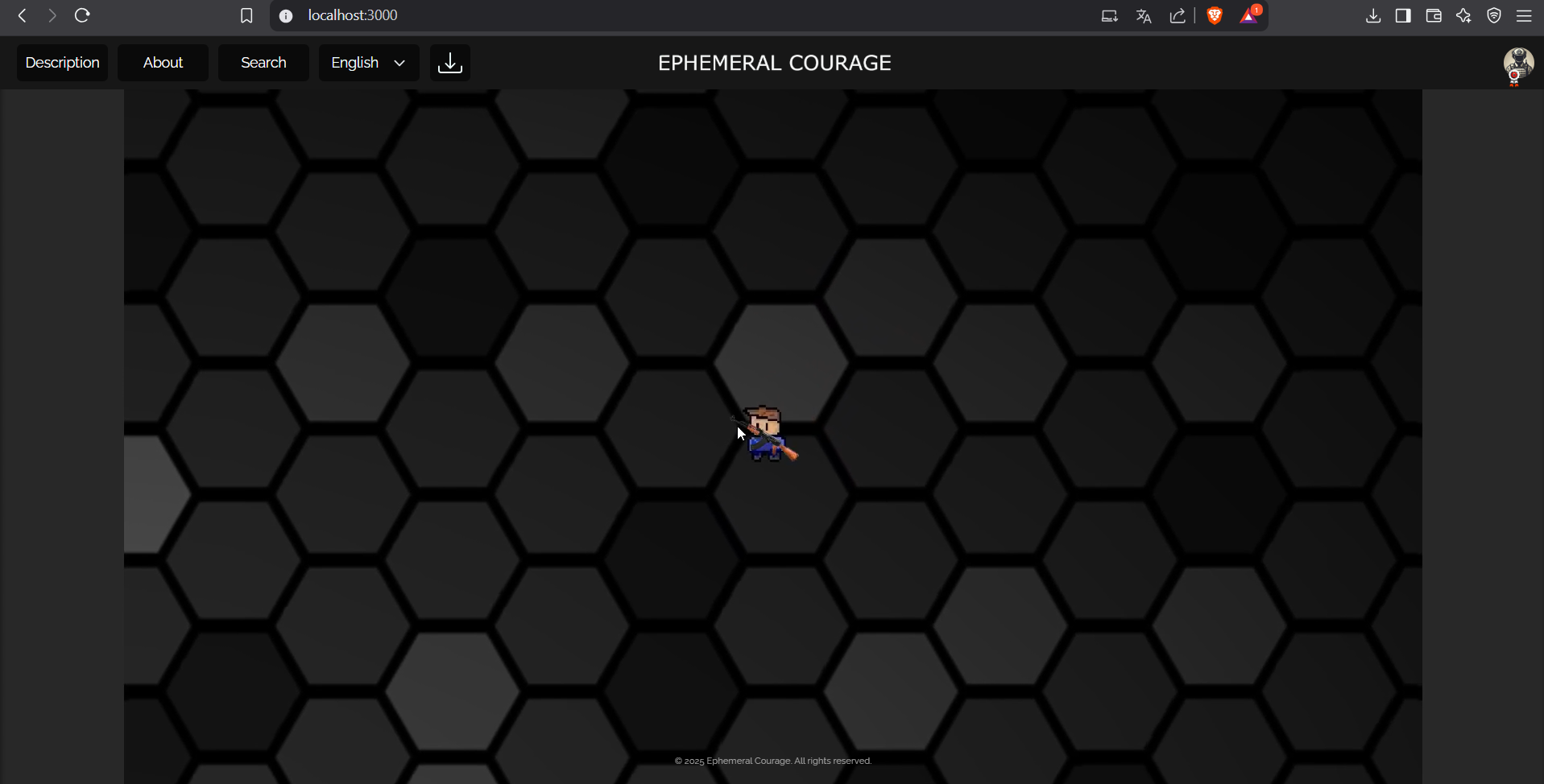
Az év elején készítettünk egy rácstervet, ahol megjelenítettük az általunk elképzelt oldal felépítését.

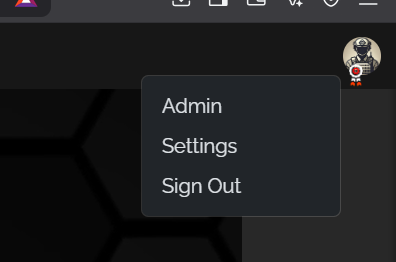


## Főoldal

A főoldal a rácsterv alapján készült el, a nem bejelentkezett felhasználókat 4 menüpont, és Bejelentkezés, Regisztráció gomb köszönti az oldal. Bejelentkezés után a Bejelentkezés, Regisztráció gombok eltűnnek, és helyette egy újabb menüpont jelenik meg, amely a felhasználó jogosultságai szerint jeleníti meg a bizonyos menüpontokat.

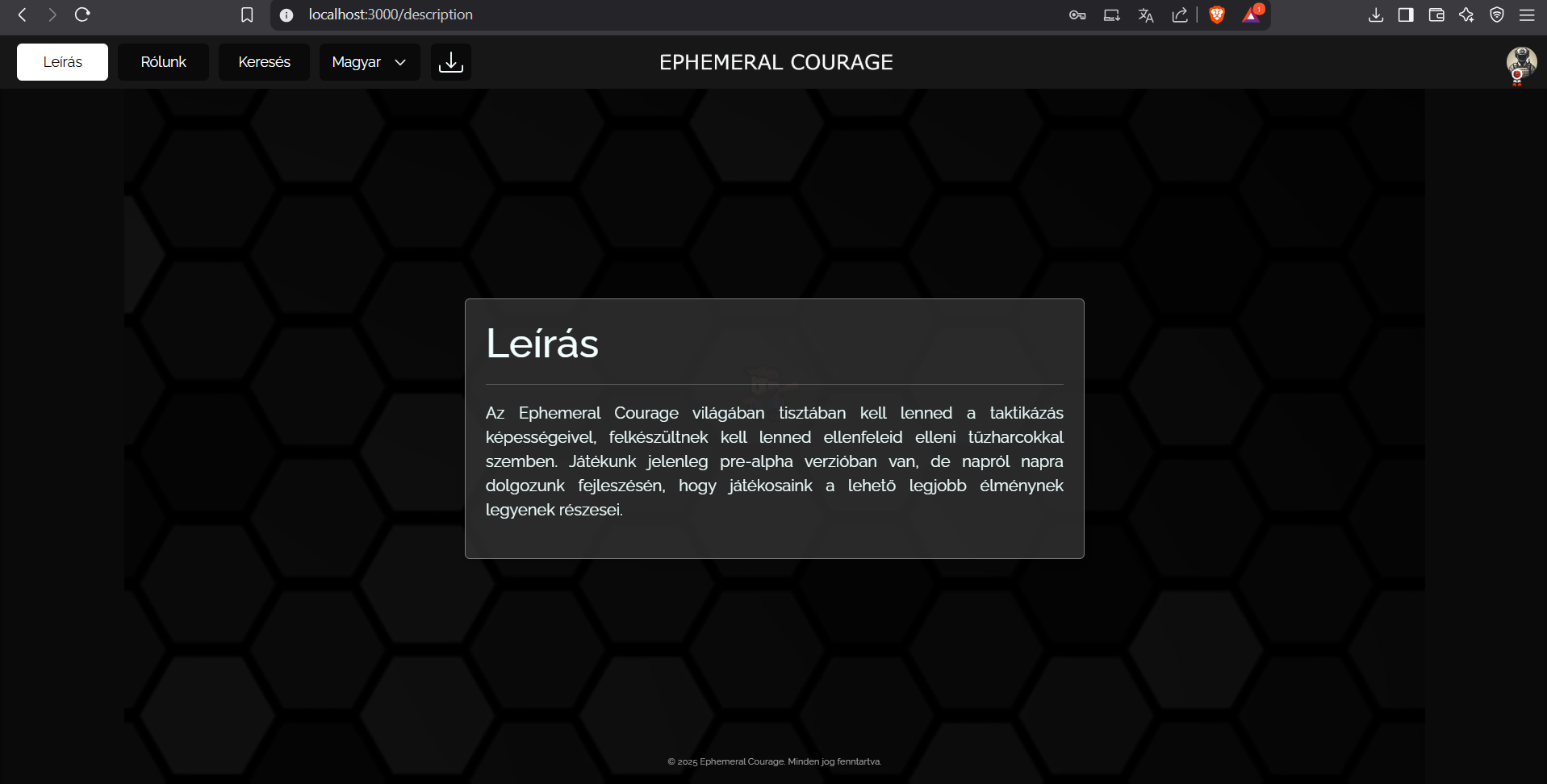




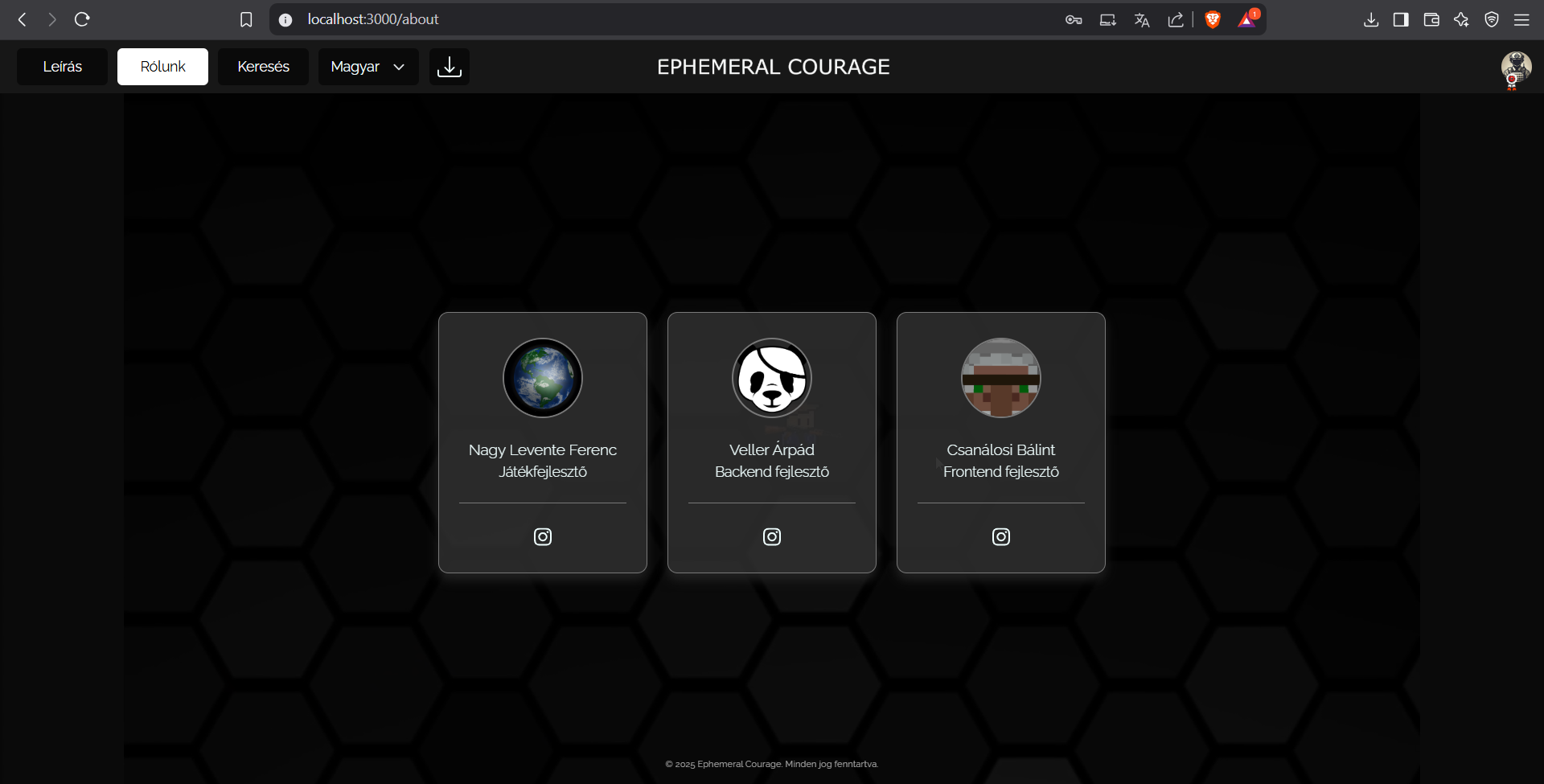


## Menüpontok

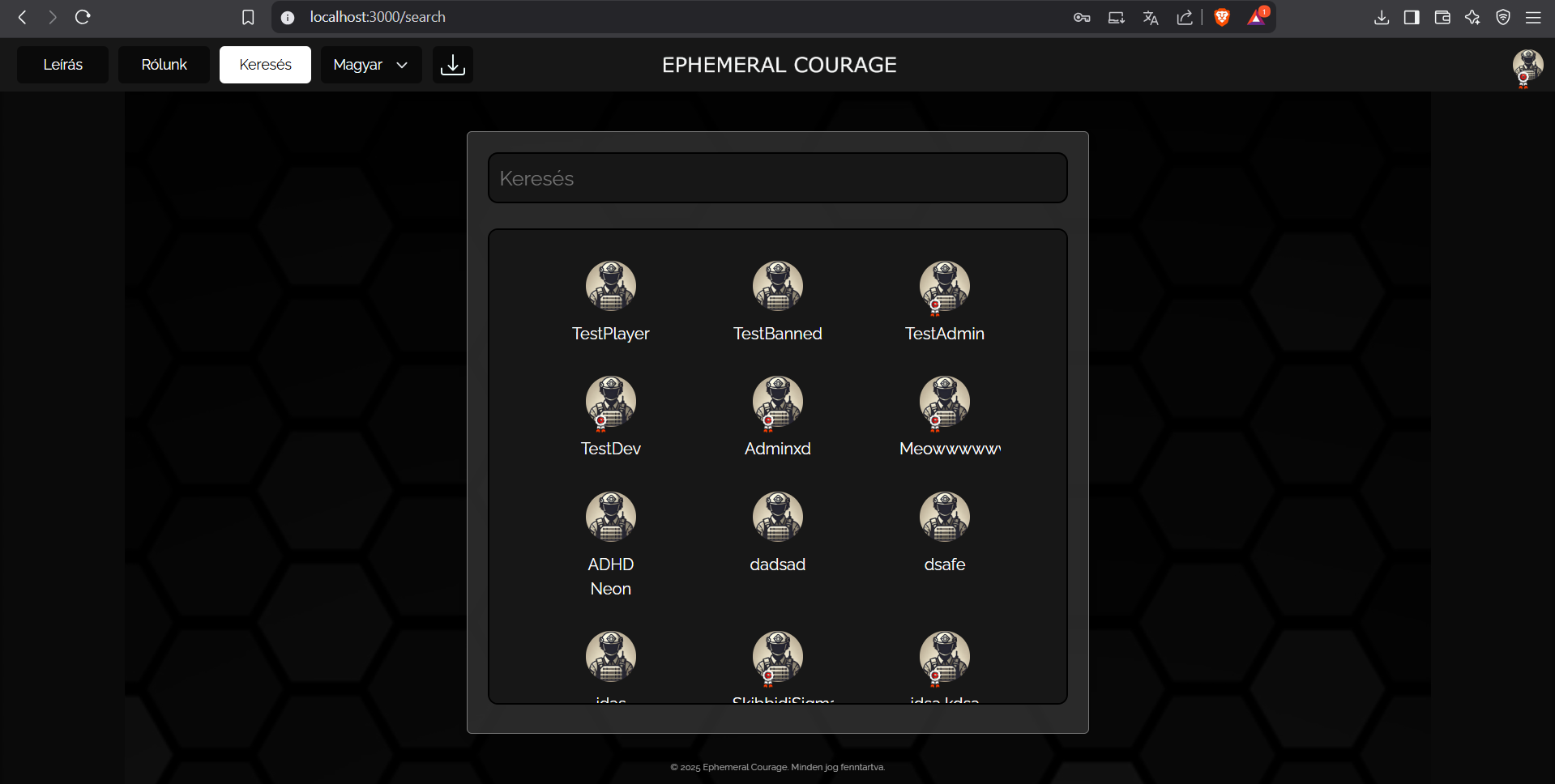
A Leírás (Description) menüpontban egy rövid ismeretőt olvashat a felhasználó a játékról.

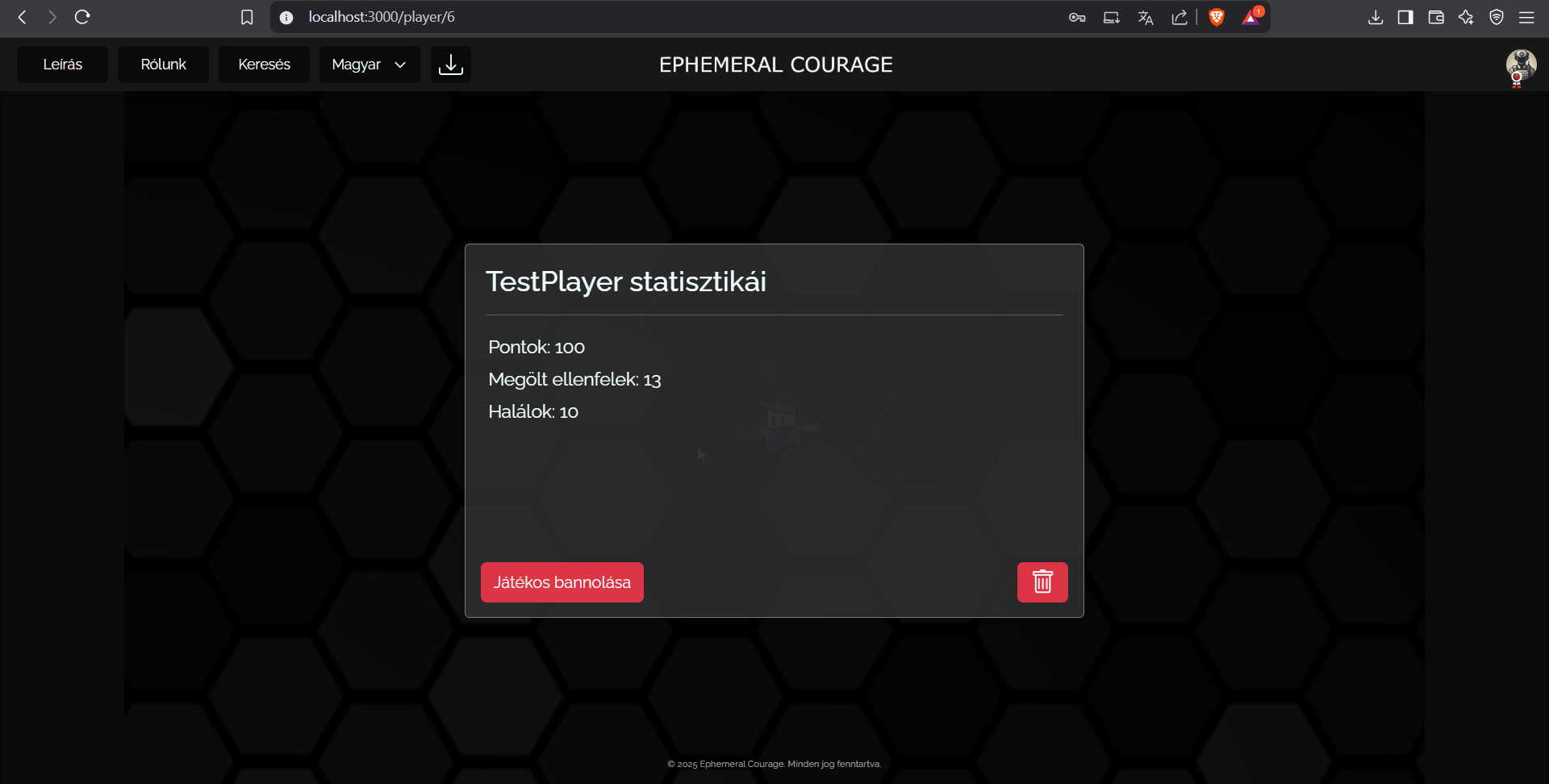


A Rólunk (About) menüpontban mi, a projekt készítői vagyunk rajta egy-egy kártyán, rajta a mi feladatrészünkkel.

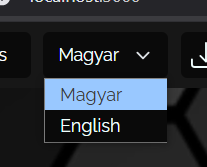


A Keresés (Search) menüpontban az eddig regisztrált összes felhasználó van kilistázva, rájuk kattintva meglehet tekinteni a statisztikájukat, esetleg azt, hogy ki vannak e bannolva. Admin jogosultsággal rendelkező felhasználók itt tudják kizárólagosan csakis a sima felhasználókat bannolni, unbannolni, törölni a fiókjukat.

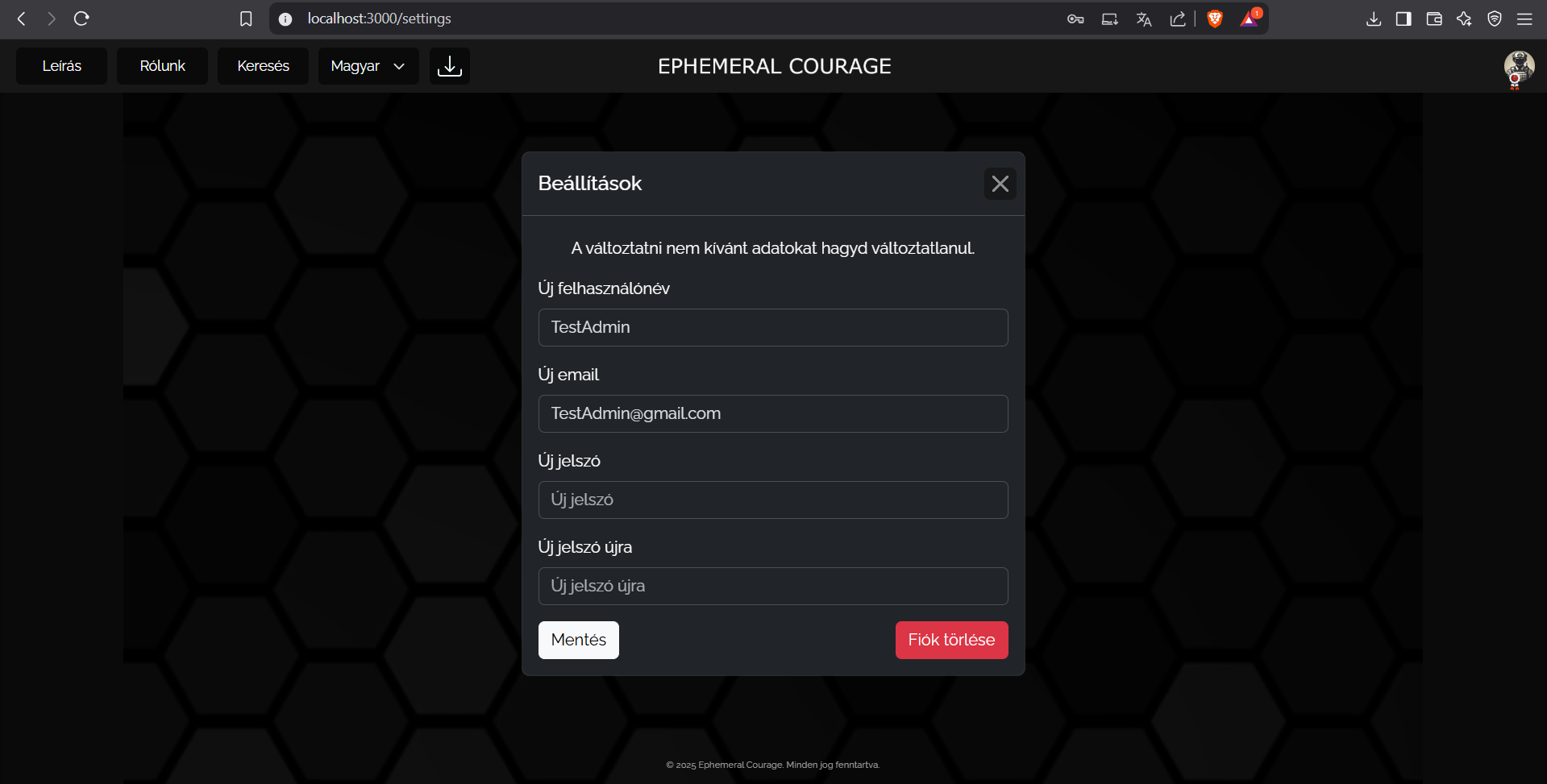




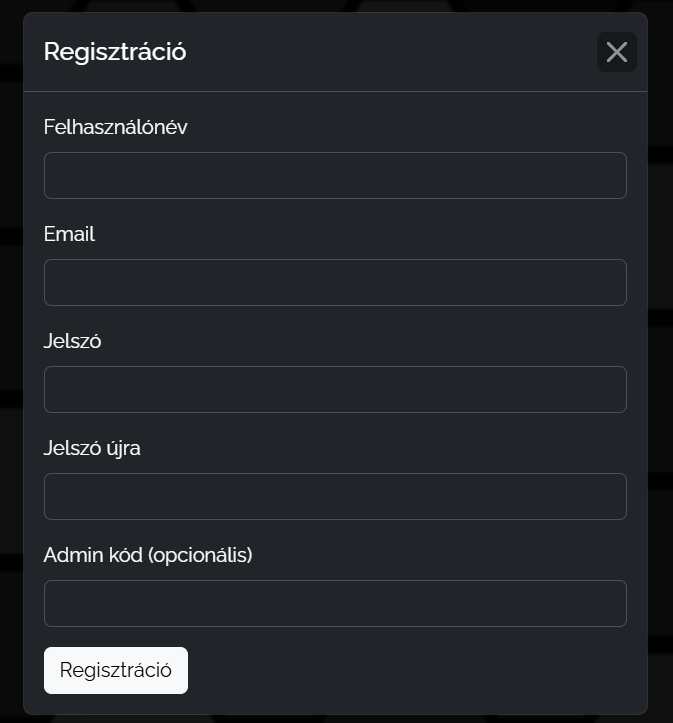
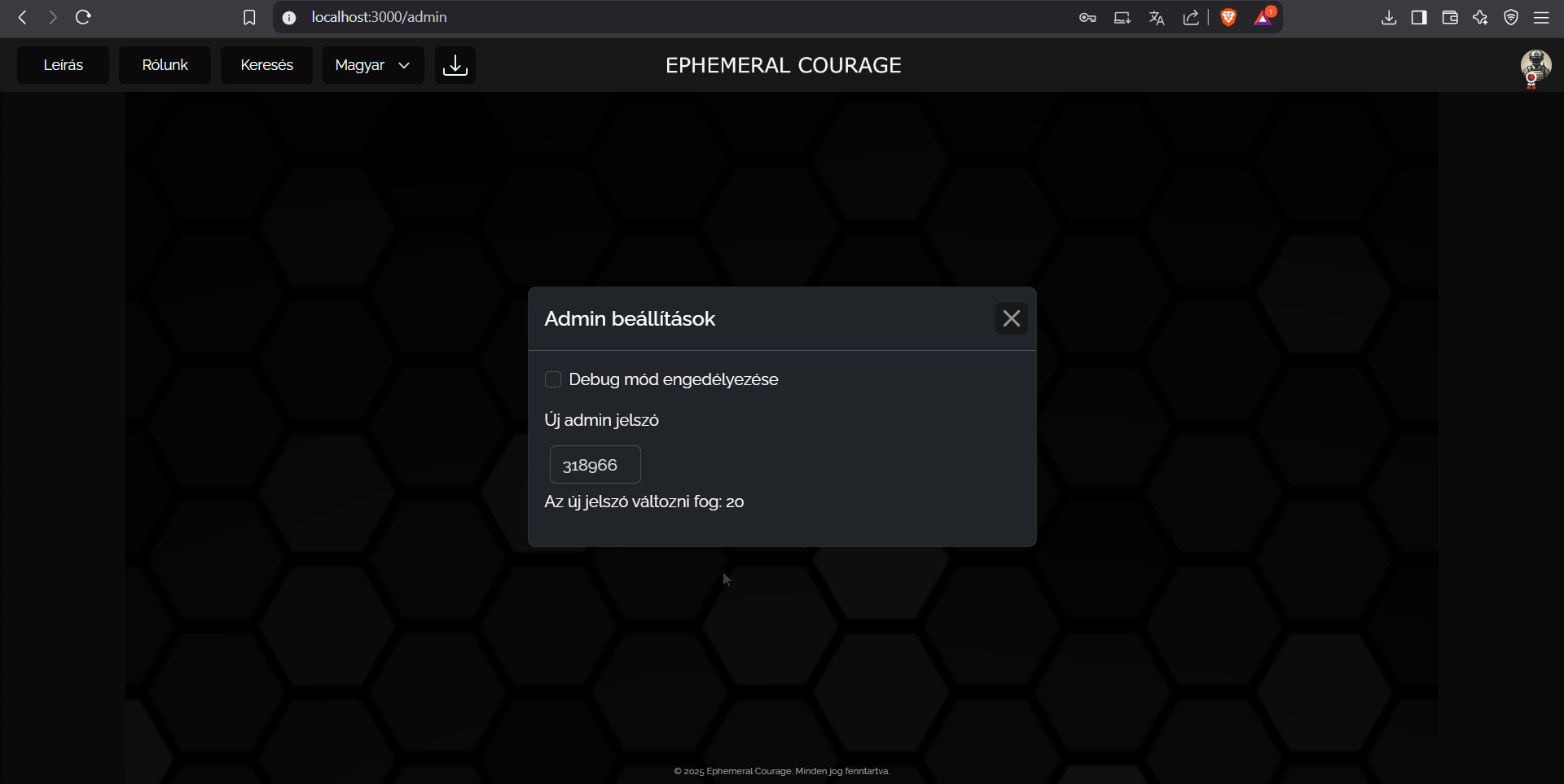
Az oldal magyar és angol fordítással rendelkezik.



A Beállítások (Settings) menüpontban, ami bejelentkezés után jelenik meg, a felhasználó megtudja változtatni az adatait, itt tudja törölni a fiókját is.



Az Admin menüpont csakis Admin jogosultsággal rendelkező felhasználóknak jelenik meg, itt tudják bekapcsolni a DevConsole-t ami a játékban ad engedélyt a DevConsole (itemek hozzáadása parancsokkal) használatához. Itt található meg az admin jelszó is, aminek a regisztrációnál van fontos szerepe, mivel ha helyesen beírják a jelszót, akkor automatikusan Adminként regisztrálja a felhasználót.



LEVI IDE KEZD EL

## Forrás linkek:

<https://stackoverflow.com>

<https://getbootstrap.com>

<https://www.w3schools.com>

<https://react.dev/learn>

<https://www.eft-ammo.com>

<https://escapefromtarkov.fandom.com/wiki/Escape_from_Tarkov_Wiki>

<https://tarkov.dev>

<https://www.youtube.com>

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

<https://tpointtech.com>

<https://www.microsoft.com>

<https://www.sap.com/hungary/products/technology-platform/integration-suite/what-is-api.html>

<https://www.google.com>