

BMSZC

Bláthy Ottó Titusz informatikai technikum

Vizsgaremek

**Kvíz Játék
Quiz Game**

2024.

Konzulens:
Duffka Erik

Készítők:
**Bálint Eszter,
Boromisza András,
Dobi Gábor**



Budapesti Műszaki Szakképzési Centrum
Bláthy Ottó Titusz Informatikai Technikum
1032 Budapest, Bécsi út 134. OM azonosító:
203058/002



Eredetiség nyilatkozat

Alulírott tanuló kijelentem, hogy a vizsgaremek saját és csapattársa(i)m munkájának eredménye, a felhasznált szakirodalmat és eszközöket azonosíthatóan közöltem. Az elkészült vizsgaremekrészét képező anyagokat az intézmény archiválhatja és felhasználhatja.

Budapest, 2024.05.07.

Bálint Eszter Margit
Tanuló aláírása

Boromisza András
Tanuló aláírása

Dobi Gábor Zoltán
Tanuló aláírása



Budapesti Műszaki Szakképzési Centrum
Bláthy Ottó Titusz Informatikai Technikum
1032 Budapest, Bécsi út 134.
OM azonosító: 203058/002



Vizsgaremek konzultációs lap

Alulírott Duffka Erik Martin konzulens aláírással igazolom Bálint Eszter Margit, Boromisza András, Dobi Gábor Zoltán nevű tanulók konzultációkon való részvételét¹.

Dátum	Téma	Tanuló aláírása	Konzulens aláírása
2024.03.12.	Adatbázis fixálása		
2024.04.19.	Backend és Frontend		
2024.05.05.	Működési tesztelés		

¹ (A vizsgaremek leadásig kötelező minden tanulónak legalább három konzultációs alkalom, ennek hiányában a záródolgozat nem értékelhető!)



Budapesti Műszaki Szakképzési Centrum
Bláthy Ottó Titusz Informatikai Technikum
1032 Budapest, Bécsi út 134.
OM azonosító: 203058/002



VIZSGAREMEK ADATLAP

A Vizsgaremek készítői:

Neve: Bálint Eszter Margit
E-mail címe: eszter93b@gmail.com

Neve: Dobi Gábor
E-mail címe: dobigabor1@gmail.com

Neve: Boromisza András
E-mail címe: andras.boromisza@gmail.com

A Vizsgaremek témája: Egy oktatási, műveltségi kvíz játék, amely grafikus felületen keresztül egyzserre oktatja és szórakoztatja a felhasználót.

A Vizsgaremek címe: A világtörténelem szilánkjai

Konzulens neve:: Duffka Erik Martin

Kelt: Budapest, 2023. November 08

Bálint Eszter Margit

Dobi Gábor

Boromisza András

Tartalomjegyzék

• Eredetiség Nyilatkozat	2
• Vizsgaremek Konzultációs Lap	3
• Vizsgaremek Adatlap	4
• A vizsgaremek téma választása	6
• Elképzelések a program felépítéséről	8
• Kezdetek	10
• A program telepítési útmutatója	11
• Kvízzjáték Használata	14
• Adatbázis	20
• Backend	22
• Frontend	24
• Csapatkommunikáció	25
• Tesztek és ellenőrzések	27
• Feladatok	31
• Összegzés/Konklúzió	34
• Elmaradt célok	36
• Fejlesztési lehetőségek	37
• Felhasznált Irodalom	38

A vizsgaremek téma választása

A tanév elején nem volt konkrét elképzelésünk a vizsgaremekünk témájáról, így külön-külön kezdtük meg a feladat ezen részét. Az első megbeszélésünk alkalmával öt darab ötlettel ültünk össze. Ezeket átrágtuk, majd egyveleget készítettünk. Született kreatív, vicces és politikai ötlet is. Végül a műveltségi vetélkedőt szántuk „megalkotásunk” tárgyául.

A világ problémáit nem oldja meg inkább csak elgondolkodtat talán. Egyrészt a művészethez – építészet, szobrászat, festészet, zene – kell egy értékrend, amit hordozunk, és/vagy bővítünk. Kell az őseink tudása, amit eszközként használhatunk és tovább adhatunk. Az utóbbi megismeréséhez a legkönnyebb út, ha odamegyünk és megnézzük alkotásaikat. A vizsgaremekünk egyik célja (ahogyan az a kezdőlapon is olvasható), hogy az oldalra látogatónak ötlettel szolgáljunk – amennyiben érdeklődő különböző kultúrák, korszakok iránt – utazásait illetően, és ismereteinek elmélyítésére ösztönözzük. Másrészt ebben a felgyorsult világban nem marad idő olyan tudás ápolására, ami boldoggá, kreatívvá, szabadabbá tesz. Ezért a kvíz formájában szórakoztatóbbá tesszük. Utazás közben, esős szabad délutánon amikor már filmet sincs kedve nézni az embernek, megismerheti képekről, leírásokból a múltat. Talán kitűzi célul, hogy elmegy Rómába azért, hogy láthassa a Pantheon-t. Utána olvas hányszor dőlt romjaira, milyen – akkor még – hihetetlennek tűnő ötlettel alkottak maradandót.

A téma megválasztásának egyik szempontja volt az is, hogy mi az, amit adatbázisban lehetne kezelni. Hogyan és miként szervezzük a táblákat, illetve hogyan kapcsolódjanak egymáshoz. Ez a későbbiekben kerül kifejtésre, ami ebben a fejezetben lényeges, hogy szinte adott volt a képek, felhasználók (és az ezekhez tartozó id-k), játék kapcsolatrendszere. Nem törekedtünk a lényeg túlbonyolítására. A játékhoz tartozó képek könnyen fellelhetőek, használhatók és cserélhetőek voltak.

Az ötletfa megrázásával tovább tudtuk szélesíteni a látóterünket. Ez sem volt nehéz az oldal és a kvíz tartalmasságát tekintve, mivel nem csak egy művészeti ágra vagy korszakra volt lehetőségünk korlátozódni (például csak festészet és az is csak a kubizmus időszakából), hanem visszanyúlóan egészen az őskori barlangrajzokig mehettünk, onnan haladva napjainkig.

Célközönségnek szerettünk volna mindenkit megszólítani, a Képkitalálós Kvíz kvízzjáték mindenki számára ideális, aki szeretne egy kis szórakozást találni és egyúttal felfedezni a világ különféle helyszíneit. Azoknak különösen ajánlott, akik szeretnek utazni, vagy érdeklődnek más kultúrák és földrajzi helyek iránt.

Elképzések a program felépítéséről

A kezdetekben még ennél a pontnál is külön-külön fogtunk bele, ami az esetünkben jó döntésnek bizonyult. Változatos és sajátos elképzeléseket láthattunk egymástól. Csináltunk "vázakat", átbeszéltük kiemelve az úgymond tetszetős részeket.

Kívülről haladtunk befelé. "Nagyolástól" jutottunk a részletekig. Azaz, az egyértelmű dolgokat elhelyeztük, úgymint a navbar, footer, bejelentkezés és regisztráció. A játékról voltak mélyebb elgondolások. Például, hogy csak regisztrációval lehessen játszani. Az oldalunk értelme az, hogy ez "mindenkié". Bárki számára elérhetőek legyen az ismeretek a világunkról (erről fentebb a témaválasztásnál volt szó a miértjéről). Ez kissé ütközik azzal az alappal, hogy azt kell bemutatnunk, hogy mit tanultunk és ebből mit kell alkalmaznunk, hisz a Kkk-ban mind a kettő szempont. Vagy az egyik, vagy a másik, vagy mindkettőt prezentáljuk. E témában is viszonylag hamar zöld ágra vergődtünk. A könnyű részek nos hát könnyen mentek. Az érdekességek egy töltelék rész, ami marasztalásra készítheti az idelátogatót. Ez a komponens érdekes, néha vicces tartalmakkal tölti ki a felhasználó idejét, elméjét, illetve a mi oldalunkat. A világ rohamosan fejlődik, így az érdekes dolgok tárháza is bővül. Éppen ezért a "jövőben" dinamikus frissülő oldal kategóriájában tartozik a munkánk. Az idézőjelek közti jövőben szó fő oka, az időhiánya.

A fehér és a sötét háttérrel nem lehet mellé löni. Kérdés, hogy melyik legyen. A legegyszerűbb, ha maga választ az idelátogató, így a navbar-ban a lehetőség adott mindkét opcióra. A jobb oldalon elhelyezkedő gombok élénk színnel lettek kiemelve,

hiszen az oldalunknak ez egy lényeges része. A Home nézetben található carousel 3 képet váltogat. Volt olyan elképzelés, miszerint e felett egy idézet van beszúrva a műveltségről

- A műveltség a nagy kapcsolatrendszerek ismeretét jelenti. Kapcsolatban állunk anyagi és szellemi környezetünkkel: de a környezetünkhöz tartozik a millió fényévnyi távolságban

fellobbant csillag, feleségünk, a közérteni eladó és a Halotti beszéd. Helyünk ismerete az Univerzumban, az élők világában, a társadalomban, a családban, felismert helyzetünkből fakadó cselekvésünk: ez mind a műveltséghez tartozik. Röviden a műveltség: általános

viszonylatrendszer (mint történelmi folyamatok) ismerete - annyi konkrét tényanyaggal, amennyivel az általános elveket a valósághoz lehet rögzíteni. - Simonyi Károly - .

Gyönyörű gondolat és passzol a témához, de látva az összehatást valahogy mégsem illeszkedett. Néha a kevesebb több. Mellette a bal oldalán egy figyelemfelkeltő szöveg, mely az oldalunk lényegét is bemutatja. A menüsorba szántunk egy rólunk és egy kapcsolat "fület", viszont egy weboldalnak lényeges eleme a footer ami minden oldal alján megtalálható. Ebbe helyeztük el a kreatívan kitalált telefonszámot és e-mail címet. Az utolsó fejezetben a bakiknál erről bővebben lesz szó.

A navbar-ban elhelyezett kártyák menüpontra kattintva, az eredeti terv szerint nyolc darab kártya jelent meg. Mindegyik kártya felülete tartalmazott egy képet és alatta egy leírást/ismertetőt. Ez módosult hat darab kártyára. Továbbá ezen kártyák bármelyikére rákattint valaki, akkor elnavigálja a kártyán feltüntetett korszakhoz tartozó weboldalra. Az első gondolat a kártyák kinézetéről megmaradt: kép, alatt pedig a leírás. A korszakok: őskor, ókor, középkor, újkor, jelenkor, modernkor.

Kezdetek

(fejlesztői környezet és a keretrendszer kiválasztása)

Amikor az adatbázis készült, a Backend tantárgyból még nem tartottunk annál a résznél, amit mi elkezdtünk. Ez az adatbázis volt. Egyszerűbb a backendet összekötni a frontenddel - legalábbis mi így vélekedtünk erről a kérdésről. A JavaScript egy elterjedt programozási nyelv a weben. Mostanában a JavaScript lett az egyik fő programozási nyelv a böngészőkben történő fejlesztéshez. A modern böngészők széles körű támogatása lehetővé teszi az egyszerűtől a bonyolultabb webes alkalmazások létrehozását. Számunkra ez lehetőség volt arra, hogy egy nyelvben dolgozzunk, ez akkor az egyszerűbb megoldásnak tűnt. Ezen felül keresztplatformos (a Node.js), ezért az alkalmazás szinte bármilyen operációs rendszeren fut. Ez például növeli az alkalmazás elérhetőségét.

A Vue.js biztosította a gyors frontend fejlesztést. Erről a frontend fejezetben bővebben kitérünk, itt, ami a lényeg, hogy könnyen karbantartható, korrigálható, újrafelhasználható. Gyors a betöltési ideje, ami igazából ettől sokkal nagyobb volumenű projekteknél fontos, de nálunk sem elhanyagolható szempont.

A fejlesztői környezet a Visual Studio Code mind a hármanknál. PhpStorm-ban is lehetett volna ez igazából csak egybeesés, hogy ebben dolgoztunk.

A Backendet pedig a JavaScript alapú Node.js-ben fejlesztettük.

A Node.js lehetővé teszi a JavaScript futtatását a szerveroldalon is, így ugyanazt a nyelvet használhatjuk mind a frontend, mind a backend fejlesztéséhez. Ez jelentős előnyökkel jár a fejlesztők és a csapatok számára, mivel lehetővé teszi a teljes stack-en történő fejlesztést, és megkönnyíti a kódbázis megosztását a kliens- és szerveroldal között és az Express.js keretrendszer egy könnyűsúlyú és rugalmas keretrendszert biztosít a Node.js-hez. Ez a kombináció lehetővé teszi a gyors és hatékony szerveroldali fejlesztést, valamint a JavaScript ismeretének felhasználását a backend területén.

A program telepítési útmutatója

A Kvízzjáték során különböző programok, és technológiákat, modulokat használtunk fel, amik mind szükséges külön-külön a játék működéséhez.

Előfeltételek

A kvízzjáték telepítése előtt bizonyos szoftverek telepítésére van szükség:

- Node.js: A Node.js a JavaScript környezet, amelyen a backend fut. Töltse le és telepítse a legfrissebb Node.js verziót a Node.js hivatalos weboldaláról.
- Vue.js
- MySQL adatbázis: A MySQL adatbázis szükséges az alkalmazás adattárolásához.

Telepítse a MySQL-t az operációs rendszerednek megfelelően.

Vue.js frontend telepítése

1. Nyissa meg a parancssort (Command Prompt vagy Terminal) a számítógépén.
2. Navigáljon a kvízzjáték frontendjének gyökérkönyvtárába.
3. Telepítse a szükséges függőségeket a következő paranccsal:

```
npm install
```

4. Indítsa el a frontend fejlesztői szerverét a következő paranccsal:

```
npm run dev
```

5. Nyisd meg a webböngészőjét, és nyissa meg a terminál ablakban a szerver által megjelölt címet pl: `http://localhost:5173` - hogy megjelenítse a kvízzjátékot.

Node.js backend telepítése

1. Nyisson meg egy újabb parancssort.
2. Navigáljon a kvízzjáték backendjének gyökérkönyvtárába.
3. Telepítse a szükséges függőségeket a következő paranccsal:

```
npm install
```

4. Hozza létre a `'.env'` fájlt a projekt gyökérkönyvtárában, és állítsa be a következő környezeti változókat:

`MYSQL_HOST='127.0.0.1'`

`MYSQL_USER='root'`

`MYSQL_PASSWORD=""`

`MYSQL_DATABASE='quiz_db'`

Az Ön gépén a beállítások és az értékek eltérhetnek, ezeket csak példaként adtuk meg, az Ön gépén beállított MySQL felhasználónevére és jelszavára van szükség.

5. Indítsa el a backend szerverét a következő paranccsal:

```
npm run dev
```

6. A backend szerver most fut a `http://localhost:3000` címen.

Adatbázis telepítése

1. Nyissa meg a MySQL parancssorát vagy egy grafikus felhasználói felületet, mint például a phpMyAdmin.

2. Hozz létre egy új adatbázist a következő paranccsal:

```
sql
```

```
CREATE DATABASE quiz_db;
```

3. Hozzon létre egy új felhasználót és adj neki hozzáférést az adatbázishoz:

```
sql
```

```
CREATE USER 'username'@'localhost' IDENTIFIED BY 'password';
```

```
GRANT ALL PRIVILEGES ON quiz_db * TO 'username'@'localhost';
```

```
FLUSH PRIVILEGES;
```

Ne felejtse lecserélni a username és password értékeket a saját felhasználónevedre és jelszavadra.

Importálja a környezetbe a `quiz_db.sql` fájlt.

A Node.js Backend és a Frontend akkor fog helyesen futni, ha a MySQL adatbázis fut.

A program használata

A kvízzjáték használatához A vue keretrendszerben futatott oldálnak, a Node.js szervernek, és a MySQL adatbázisnak is egyszerre kell elérhetőnek lennie.

Ekkor a webböngésződben a Vue.js által futtatva a felkínált helyi címen elérhető a játék, ami alapbeállításban a <http://localhost:5173> -as cím és itt lesz elérhető a játék.

A program pontos és részletes használatáról a következő fejezetben részletesen kitérünk.

Két teszt felhasználó lett regisztrálva a rendszerbe,

- az egyik sima felhasználóként, adminisztrációs jogok nélkül „email: **test@test.hu**, jelszó: **1234test**”.

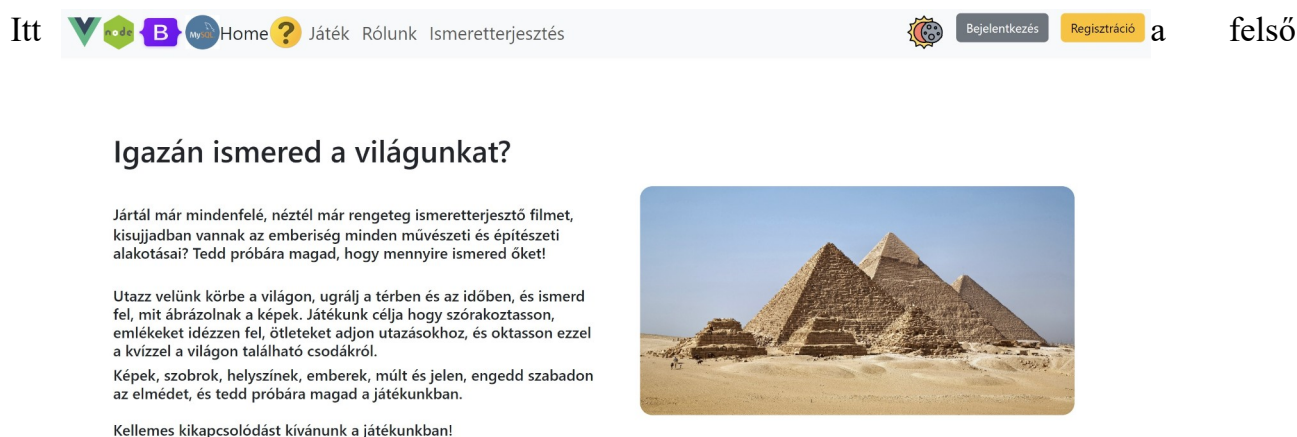
A másik felhasználó adminisztrációs jogokkal, az admin menü használatára is alkalmasan „email: **test@blathy.info**, jelszó: **1234test**”.

Természetesen lehetőség van további felhasználók és adminisztrátorok regisztrálására is.

A részletes használatra a következő pontban térünk ki.

Kvízjáték Használata

A játék betöltődése utáni induló képernyő:



menüsorból kiválasztva lehet navigálni az elemek között.

Home: Ez a főmenü, a kezdő oldal, ide tölt be a játék.

Játék: Itt lehet magát a játék részt elérni, elindítani.

Rólunk: Itt egy rövid ismertető van rólunk, a készítőkről, a projekt céljáról, az iskolánkról.

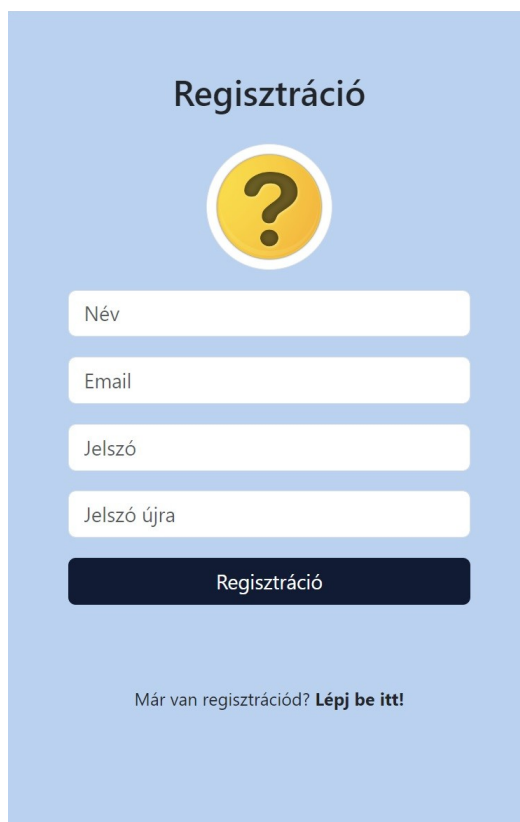
Ismeretterjesztés: Itt különböző történelmi korokról lehet megtudni érdekességeket, és tematikusan az adott „külső” oldalakra vezető linkek vannak, amik az adott korról bővebben és tudományos aprólékossággal írnak.

A **Nap/Hold ikon** világos és sötét mód közötti váltást tesz lehetővé.

A **Bejelentkezés** gomb a már regisztrált felhasználót belépteti, és a Játékban a továbbiakban névvel tud részt venni.

A **Regisztráció** gomb a felhasználó regisztrációját teszi lehetővé.

Regisztráció:

The image shows a registration form on a light blue background. At the top, the word "Regisztráció" is written in bold. Below it is a yellow circular icon with a black question mark. There are four white input fields stacked vertically, labeled "Név", "Email", "Jelszó", and "Jelszó újra". Below these fields is a dark blue button with the text "Regisztráció" in white. At the bottom of the form, there is a link that says "Már van regisztrációd? Lép be itt!".

Regisztrációkor 4 adatot kell megadnunk:

Egy felhasználó nevet ezt a '**Név**' mezőbe írunk, egy email cím, amit az '**Email**' mezőbe írhatunk (ugyan úgy, mint a felhasználó nevet), illetve egy jelszót, amit a '**Jelszó**' mezőbe írunk (ugyan úgy mint a felhasználó névnél és emailnél), és ezt azonosan a megadottal, újra meg kell adnunk a '**Jelszó újra**' mezőben.

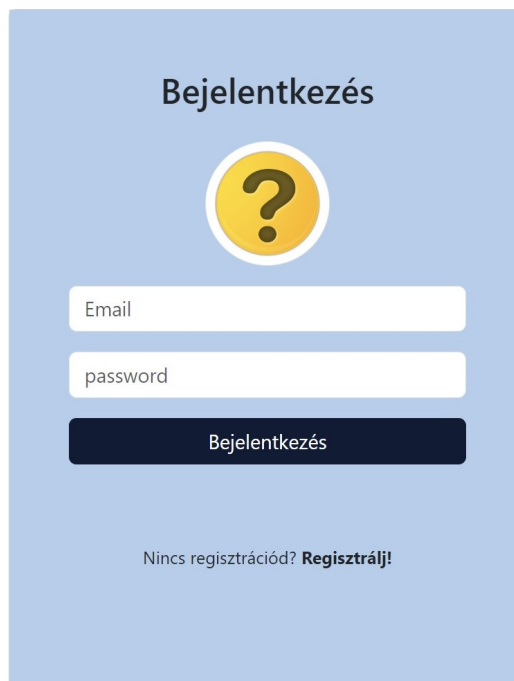
Minden adat szükséges a regisztrációhoz! Amennyiben valamelyiket nem adjuk meg, abban

az esetben a program jelzi nekünk.

Adatok kitöltése után a 'Regisztráció' gombra kattintva regisztrálhatunk is. Amennyiben még is van regisztrált fiókunk, abban az esetben a 'Belépés' felíratra kattintva visszatérhetünk a belépés menühöz.

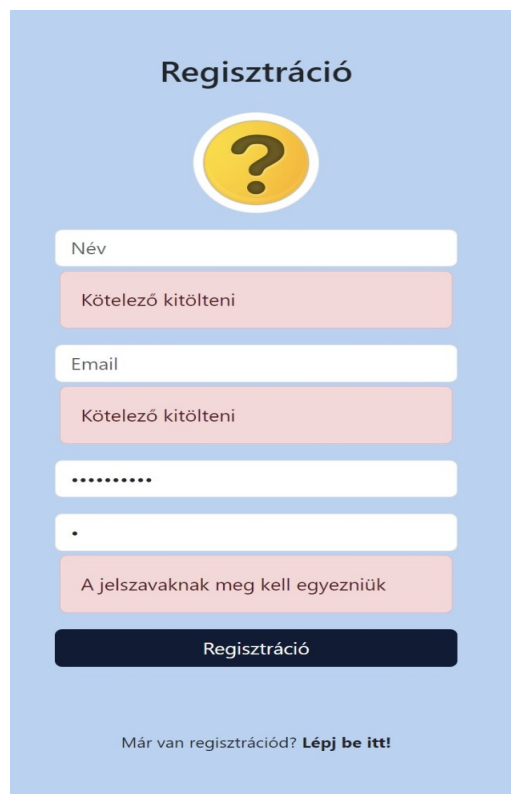
Admin funkcióval regisztráció. Amennyiben a regisztráció során a user az email címnél '@blathy.info' végződést ad meg, akkor a felhasználót admin jogokkal regisztrálja a rendszer, és a belépést követően egyéb adminisztrációs menükhöz, funkciókhoz is hozzáfér.

Bejelentkezés:



The login form is titled "Bejelentkezés" and features a yellow question mark icon. It contains two input fields: "Email" and "password". Below these fields is a dark blue button labeled "Bejelentkezés". At the bottom, there is a link that says "Nincs regisztrációd? **Regisztrálj!**".

akkor a program erre figyelmezteti.



The registration form is titled "Regisztráció" and features a yellow question mark icon. It contains several input fields: "Név" (Name), "Email", and a password field with a strength indicator. Below the password field is a red box with the text "Kötelező kitölteni" (Mandatory to fill). At the bottom, there is a dark blue button labeled "Regisztráció". At the very bottom, there is a link that says "Már van regisztrációd? **Lépj be itt!**".

A bejelentkezésnél a korábban regisztrált felhasználó tud belépni a játékba, és a korábban megadott felhasználó nevével vehet részt a játékban így.

Belépésnél a regisztrációnál megadott email címet kell az 'Email' mezőbe beírni, és a megadott jelszót a 'Jelszó' mezőbe.

Ezután a 'Bejelentkezés' gombra kattintva a program - ellenőrzés után - belépteti a felhasználót.

Amennyiben a korábban megadott regisztrációs adatait a bejelentkezésnél hibásan adta meg,

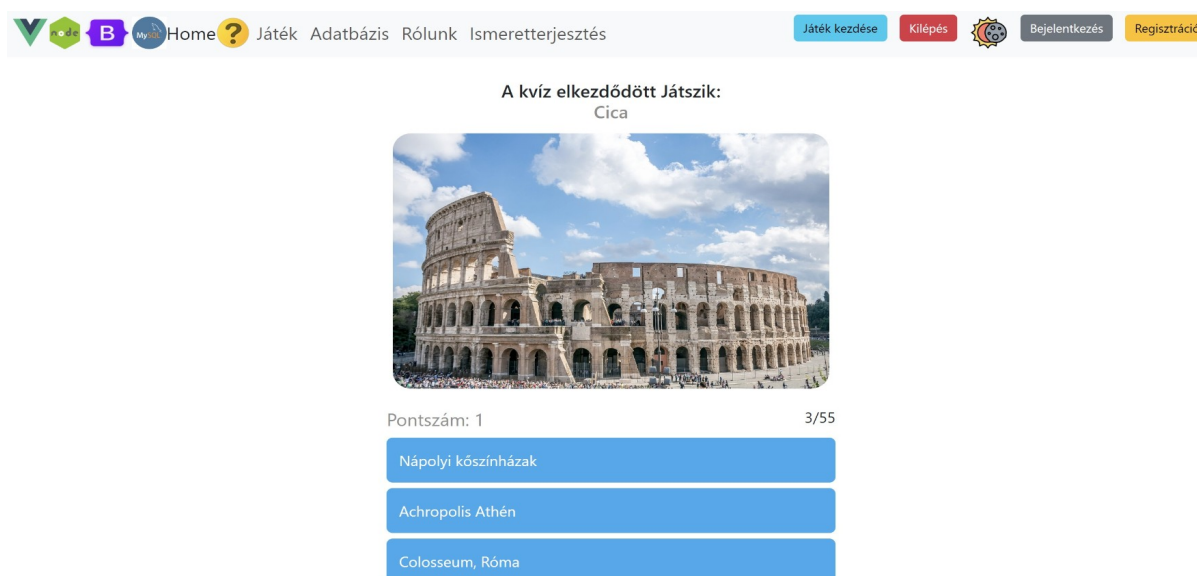
Mind a regisztrációkor, mind a bejelentkezéskor a program az adatokkal kapcsolatos elvárásokra -azok hibásan teljesülése esetén, figyelmezteti a felhasználót:

Csak a helyen megadott adatokkal lehet regisztrálni és belépni.

A sikeres bejelentkezés után további két gomb jelenik meg jobb oldalt felül, a „**Játék kezdése**”, valamint a „**Kilépés**”.

Amennyiben a **Játék kezdése** gombra kattint a felhasználó, a Játék oldal nyílik meg, és elkezdődik a játék, amennyiben a **Kilépés** gombra kattint, a program kilépteti.

Játék oldal:



A játék során a képernyő középső -felső részén egy kép jelenik meg, ami alatt három válasza adási opció van.

A kép felett a bejelentkezett játékos neve olvasható, amennyiben nem jelentkezett be valaki, akkor az 'Anonymus' név olvasható.

A kép alatt bal oldalon a játékos által az aktuális játékban elért pontszám, a jobb oldalon a kérdés sorszáma, és az összes kérdés (kép) száma olvasható.

Helyes válasz esetén a jó választ tartalmazó mező zöldre változik, és a játékos



pontszáma egy ponttal megnövekszik, amiről egy a kép alatt 0.8 másodpercre megjelenő felirat tájékoztatja.

Helytelen válasz esetén minden válaszokat tartalmazó mező vörösre változik, és a kép alatt a 'Rossz válasz!' felirat jelenik meg.

A válaszok kiválasztása után a válaszmezők 0,8 másodpercre inaktívvá válnak, és ezt követően betöltődik a következő kép.

A játék addig tart, amíg a felhasználó az összes képre megválaszolja a kérdést, vagy pedig amíg meg nem szakítja a játékot.

Az összes kérdés megválaszolását követően, a játék

végén elmentődik a játékos pontszáma, a neve, és a alábbi tartalom jelenik meg, amely tájékoztatja a felhasználót az elért pontjairól, a kérdések számáról, valamint a legjobb 10 játékos által elért eredményt megmutatja.

A bejelentkezés nélkül játszó játékosok a listában 'Név nélkül' megnevezéssel szerepelnek.

A játékban a jobb eredményeket elérő játékosok kiszorítják a kevésbé magas pontszámot elérőket, de a 10-es limittel elejét vesszük a kezelhetetlenül hosszú listának.

A kvíz elkezdődött Játsszik:
Csacska



Pontszám: 2

Rossz válasz!

3/55

Nápolyi kőszínházak

Achropolis Athén

Colosseum, Róma

A végére értél a kvíznek! Játsszik:
Csacska

Gratulálunk!! Végére értél a kvíznek!

Az eredményed: 14 ennyi kérdésből: 55

Új játék Kilépés


A legjobb eredmények:

Név nélkül	48
Név nélkül	25
Csacska	20
Név nélkül	16
Név nélkül	15
Csacska	14
Név nélkül	12
Cica	12
Név nélkül	12
Csacska	10

Amennyiben adminisztrációs joggal rendelkező felhasználó lép be a rendszerbe pl: „email: **'test@blathy.info'** jelszó: **'1234test'** akkor egy további menüpont is megjelenik a navigációs sávon, az „**Adatbázis**”

Ekkor az admin joggal rendelkező felhasználónak lehetősége van az adattáblákat szerkeszteni, lekérdezni a tartalmukat, módosítani, hozzáadni, törölni.

(A users és scores tábláknál csak a törlést gondoltuk fontosnak, a questions táblánál minden művelet lehetőségét megvalósítottuk)


Home
Játék
Adatbázis
Rólunk
Ismeretterjesztés

Játék kezdése
Kilépés
Bejelentkezés
Regisztráció

Az adatbázis egy mySQL szerveren fut, amit egy Node.js szerver szolgál ki a Vue.js felé. Ez a menü csak admin jogosultsággal érhető el. Lehetőség van Kérdéseket módosítani, létrehozni, lekérdezni és törölni, felhasználókat és eredményeket törölni.

Kérdések tábla

id	Név	elérési út	válasz 1	válasz 2	válasz 3	Művelet
#	név	elérési út	Válasz	Válasz	Válasz	Felvétel
1	piramis	/src/img/1.jpg	A gízai piramisok	Alexandriai világítótorony	Hamurapi sziklatemploma	Módosítás Törles
2	A Magyar Parlament	/src/img/3.jpg	Alexandriai világítótorony	A Magyar Parlament	Nápolyi közházak	Módosítás Törles
3	moai	/src/img/moai.jpg	Húsvét-Szigetek, Moai szobrok	Sehell szigetek, törzsi szobrok	Arámiai összobrok	Módosítás Törles
4	colosseum	/src/img/colosseum.jpg	Nápolyi közházak	Achropolis Athén	Colosseum, Róma	Módosítás Törles
5	bison	/src/img/bison.jpg	Neandervölgyi barlangrajz	Bison barlang, Altamira	Bivaly fresko, Nápoly	Módosítás Törles
6	Stonhenge	/src/img/stonehenge.jpg	Közép-Afrika, templom romok	Törzsi közházak, Seychelles szigetek	Stonehenge, Anglia	Módosítás Törles
7	Adam	/src/img/adam.jpg	Ádám Teremtése, Sixtus-kápolna	Apa és Fia, Leonardo de Vinci	Érintés, ismeretlen művész	Módosítás Törles

A sorok jobb szélén található gombokkal lehet az adott műveletre utasítást adni.

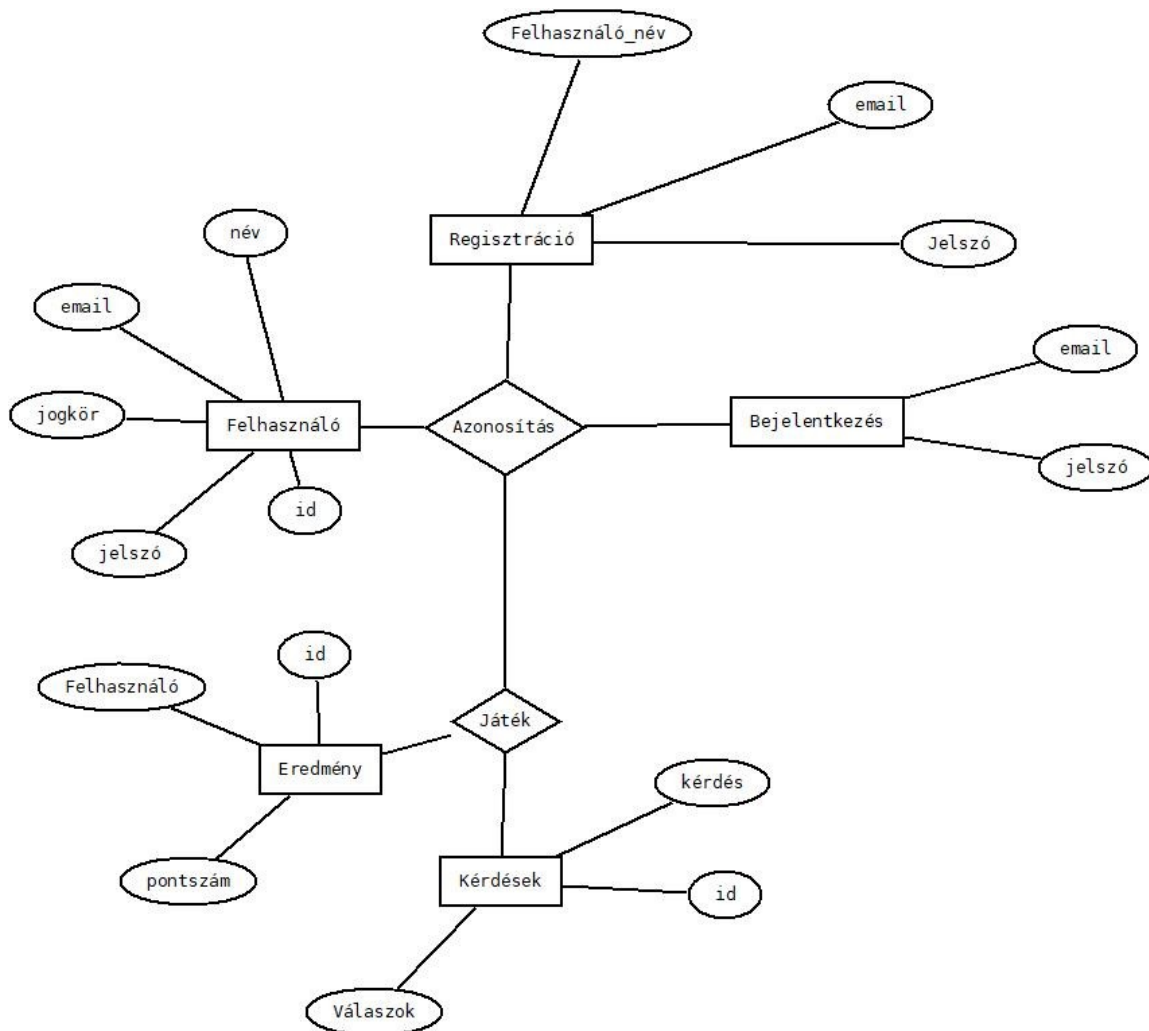
Az admin jogosultsággal rendelkező felhasználó is természetesen részt vehet a játékban is.

Adatbázis

A játék adatbázisát egy **MySQL** adatbázisban létrehozott táblákban tároljuk, és a backend onnan éri el őket, a teljes **CRUD** műveleteket megvalósítva.

A **XAMPP** Control Panel-en keresztül a **phpMyAdmin**-t használtuk. Az ott létrehozott **quiz_db** adatbázisban három tábla található – **questions**, **scores**, **users** táblák-

Alapvetően törekedtünk a legegyszerűbb, legátláthatóbb megoldásra, amit a folyamatosan felmerülő igények szerint bővítettünk.



Három adattáblát készítettünk, és ezekben hoztuk létre a megfelelő formátumban és az igényeknek a mezőket.

- **questions:** itt található a felhasználó neve és a hozzátartozó id-ja, a képek, plusz a 3 válasz, ami mindhárom jellemzően kapcsolódik a képhez, és nagyjából bármelyik lehetne helyes. amik közül az egyik a helyes

a helyes válasz számát minden kérdésnél a 'correctAnswerDB' mező adja meg.

Így minden kérdésnél specifikusabban tudjuk a képre vonatkozó válaszlehetőségeket szabályozni,

mintha egy külön táblából véletlenszerűen választanánk a „kitöltő” kérdéseket.

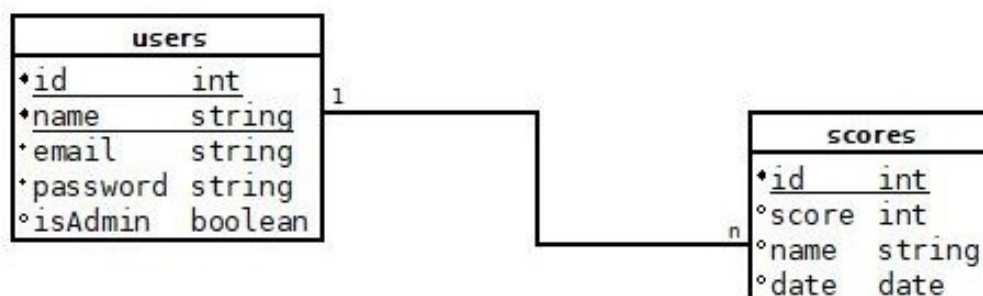
Ebből a táblából történik az adott kérdéshez a megfelelő kép címeinek az átadás is.

questions	
*id	int
*name	string
*imagePath	string
*answer1	string
*answer2	string
*answer3	string
*correctAnswerDB	int

A **users** táblában a regisztráció során megadott adatok kerülnek, amik a felhasználó későbbi beléptetését, a játékeredményei nyomon követését, valamint adott esetben az adminisztrációs jogokkal történő rendelkezését is ez a tábla tárolja.

- **scores:** a név, id és az elért eredmény

- **users:** a felhasználó adatai, az email, jelszó stb.



Az eredmények a scores táblában mentődnek el az adott user által használt névvel.

Backend

A backend a JavaScript nyelven, Node.js és Express.js keretrendszerben íródott. Ennek a témakörnek az eleje az adatbázis fejezethez kapcsolódik. Inkább leíró, mint magyarázó rész, majd ezt követően lesz szó arról, hogy a különböző útvonalak hogyan vannak definiálva a backend számára.

A database.js egy modult tartalmaz, ami kapcsolatot teremt az adatbázissal, és különböző funkciókat meghatároz az adatbázis műveletek elvégzéséhez. A műveletek végrehajtása során a függvények aszinkron módon működnek, ezért hatékony az adatbázissal történő kommunikáció. Funkciók, amik segítenek a backendnek az adatok lekérdezésében, frissítésében, törlésében, és létrehozásában:

1. `getQuestions()`: visszaadja az összes kérdést az adatbázisból
2. `getQuestions(id)`: visszaadja az adott azonosítójú kérdést az adatbázisból
3. `createQuestion (name, imagePath, answer1, answer2, answer3, correctAnswerDB)`: létrehoz egy új kérdést az adatbázisban a megadott adatok alapján
4. `deleteQuestion(id)`: törli az adott azonosítójú kérdést az adatbázisból
5. `updateQuestion(id, name, imagePath, answer1, answer2, answer3, correctAnswerDB)`: frissíti az adott azonosítójú kérdést az adatbázisban a megadott adatok alapján
6. `createScore(score, name)`: létrehoz egy új pontszámot az adatbázisban a megadott pontszám és név alapján
7. `getScores()`: visszaadja az összes pontszámot az adatbázisból
8. `getTopScores()`: visszaadja a legjobb 10 pontszámot az adatbázisból, rendezve a pontszám szerinti csökkenő sorrendben
9. `getUsers()`: visszaadja az összes felhasználót az adatbázisból
10. `getUsers(id)`: visszaadja az adott azonosítójú felhasználót az adatbázisból

11. `registerUser(name, email, password)`: új felhasználót regisztrál az adatbázisban név, mail és jelszó alapján
12. `loginUser(email, password)`: belépteti a felhasználót az adatbázisban az adott mail cím és jelszó alapján

Az `app.js`-ben található egy Express alkalmazás, aminek célja, hogy az útvonalakat definiálja a backend számára. Minden útvonal publikus.

Útvonalak és a hozzájuk tartozó funkció, API végpontok:

1. **get /questions**: visszaadja az összes kérdést a `getQuestions()` függvény segítségével
2. **get /questions/:id**: visszaadja az adott azonosítójú kérdést a `getQuestion(id)` függvény segítségével
3. **post /questions**: létrehoz egy új kérdést a kérdés adatainak elküldésével a tartalomban a `createQuestion()` függvény segítségével
4. **put /questions/:id**: frissíti az adott azonosítójú kérdést a kérdés adatainak elküldésével az `updateQuestion()` függvény segítségével
5. **delete /question/:id**: törli az adott azonosítójú kérdést a `deleteQuestion(id)` függvény segítségével
6. **get /users**: visszaadja az összes felhasználót a `getUsers()` függvény segítségével
7. **delete users/:id**: törli az adott azonosítójú felhasználót a `users` táblából
8. **delete scores/id**: törli az adott azonosítójú eredményt a `scores` táblából

Ezek a NodeMysql mappában találhatóak.

Frontend

A Vue.js-t a tanultak alapján könnyen tudtuk alkalmazni a webes alapú vizsgaremekünkhöz. Képességeinkhez mérten egy letisztult munkát tudtunk kiadni a kezeink közül. A különálló oldalak képesek együtt működni az elképzeléseinkkel úgy, hogy az App.vue-ban összesen tizenhét sor található a tag-ekkel együtt. Az ilyen és ehhez hasonló módszerek, amiket biztosít ez a library felgyorsította az előrehaladásunkat.

A reszponzivitás a bootstrap alkalmazásával történt. Ezt a terminálban töltöttük be úgy, mint a vue-router-t és az axios-t is például. Telefonon és tableten is átlátható "formát ölt" az oldal kinézete.

A komponenseknek köszönhetően az alkalmazás különálló részekre bontható, ami megkönnyítette a fejlesztést. Az első komponens, ami elkészült az a navbar. Úgymond – a kezdeti lépésekben - vázlatosan elhelyeztük benne a menüpontokat és a gombokat. Mivel könnyedén szerkeszthető, bármikor visszatérhettünk erre a kérdésre, így haladtunk tovább a második lépésre – ami a kinézetet illeti. A nézetek létrehozásával bővült a fejlesztés menete. Az adatok validációjához a vee és yup csomagokat használtuk.

Rendkívül jó volt, hogy a Vue engedelmes eszközként belesimult a kezeinkbe, és a felmerülő új igényeket csomagok hozzáadásával tudtuk rögtön kielégíteni, mint pl Pinia state-store management eszköz.

Igyekeztünk kihasználni a keretrendszer adta rendkívül nagy rugalmasságot, a beépített és importált technikák adta megoldási lehetőségeket.

A Bootstrap által a kezünkbe adott eszközökkel külön öröm volt a felületek megjelenítésének a kialakítása.

Igyekeztünk minél több eszközt felhasználni ezekből, még ha ez nem is lett volna mindig indokolt.

Csapatkommunikáció

A folyamatos kommunikációnkat a Messenger, Teams (főleg a konzulens tanárral) és a Discord segítette. Jelenleg, miközben ez a mondat íródik, most is Messengeren írogatunk egymásnak.

A GitHub-ot használtuk a kód megosztására és verziókezelésre.

Esetenként fájlokat direktben is cseréltünk egymással ímélen, Discordon és felhő-tárhely szolgáltatón keresztül.

A Trello-t használtuk a feladatok nyomon követésére és a munkamegosztásra. Itt osztottuk ki a különböző feladatokat és követtük azok előrehaladását.

A Trello táblák segítettek nekünk strukturálni a feladatokat és megosztani azokat egymás között.

GitHub pull request-eket használtunk a kódváltoztatások és új funkciók beillesztésének követésére és áttekintésére.

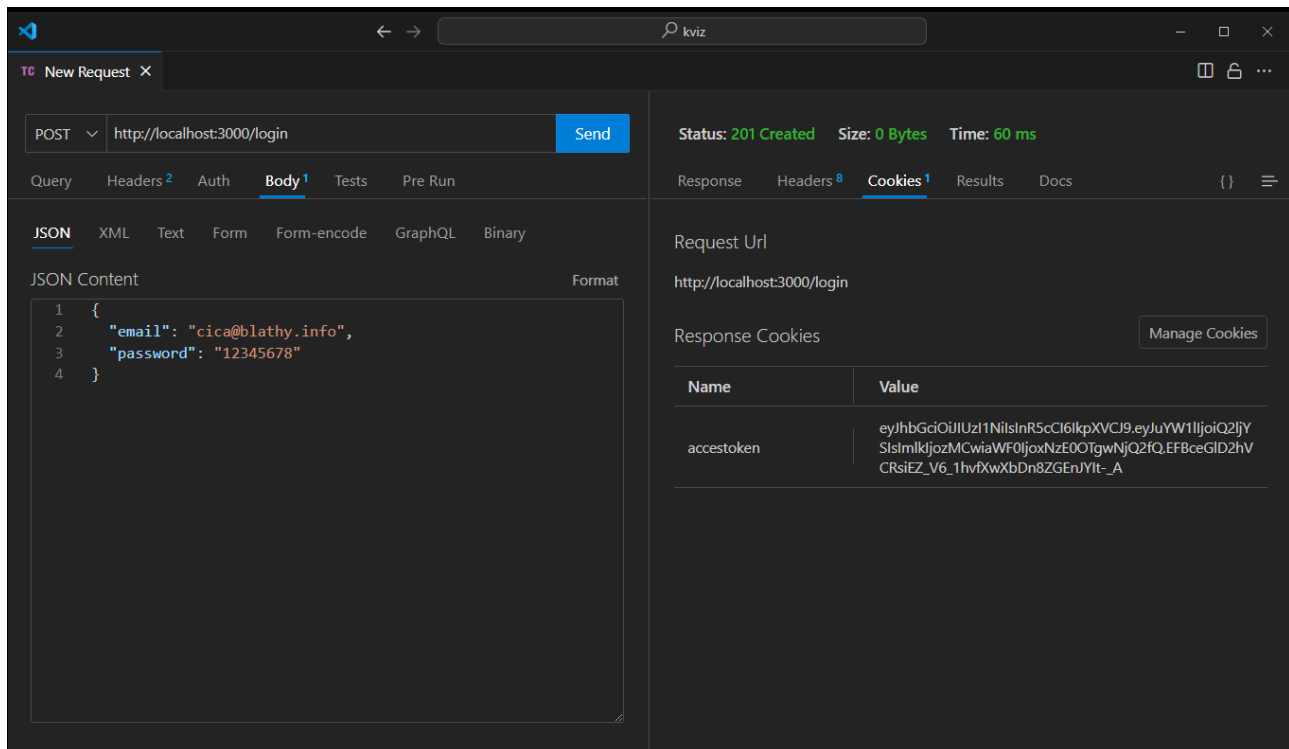
Ahogy közeleg a vizsgaleadás határideje, úgy szaporodik a beszélgetéseink száma. Minden apró részletet és változtatást – amennyiben van – folyamatosan megosztunk egymással a kezdetek óta. Ezt azért tartottuk kiemelkedően fontosnak, mert bár a projekt kicsi (az életben adódó munkákhoz képest), mi nem ismerjük egymást. Nem tudtuk mekkora a különbözőségünk mértéke, miben és mennyire igazodjunk a másik két félhez úgy, hogy közben mi is a kellő mértékben érvényesülni tudjunk. Az első gondolatok talán ezek, amikor csapatba verődnek olyan emberek, akik ezelőtt még egy szót sem beszéltek egymással. Nálunk ez nem okozott nehézséget. Szerencsére mind a hármunknak van humorérzéke és ami fontosabb, hajlandóság a közös célra.

Átbeszéltük, hogy ehhez milyen eszközök és irány szükséges.

A projekt előrehaladtával kiütköztek a különbségek például az oldal színvilágát tekintve. Mindenkinek meg volt a maga meggyőződése ami mellett érvélt, így jutottunk a jó megvalósításhoz. Az, hogy a kezdetben külön-külön fogtunk hozzá a munkához, közelebb hozott minket egymás ízlés világához, gondolatmenetéhez, ismeretéhez. Így tudtunk egy csapatként tovább haladni.

Tesztek és Ellenőrzések

A kvízzjáték fejlesztése során nem csak a felhasználói felületet, hanem a backend rendszert is folyamatosan teszteltük és ellenőriztük, hogy a megírt rész működőképességéről meggyőződjünk, valamint a már korábban megírt részek működőképessége is megmaradt.



Számtalan alkalommal hoztak felszínre a tesztek problémáikat. Mivel a projekt különböző elemeit külön, sokszor önállóan fejlesztettük, a teszteken keresztül tudtunk a helyes integrációról is meggyőződni.

A folyamatos tesztelés elengedhetetlen része a projektnek.

A backend oldali tesztek és ellenőrzések biztosítják, hogy az alkalmazás megfelelően működjön, és az adatok helyesen kezelődjenek.

Egységtesztek

Egységteszteket írtunk a backend kód különböző részeire, beleértve az adatbázis műveleteket, az üzleti logikát és az API visszatérési értékeit.

```
const resultUser = await createUser('Kalapácska', 'kalap@abaaadddaba.com', 'blalawwl')
console.log(resultUser)

const resultScores = await getScores()
console.log(resultScores)
```

A tesztek célja a különálló komponensek helyes működésének biztosítása és a hibák azonnali felfedezése.

Pl itt a funkciók megírások konzolosan azonnal teszteltük is a működésüket, a szerver a tesztet lefutattja, és az eredményt a konzolon megjeleníti, visszajelzést adva az eredményességről.

Felhasználói Felület Tesztelése

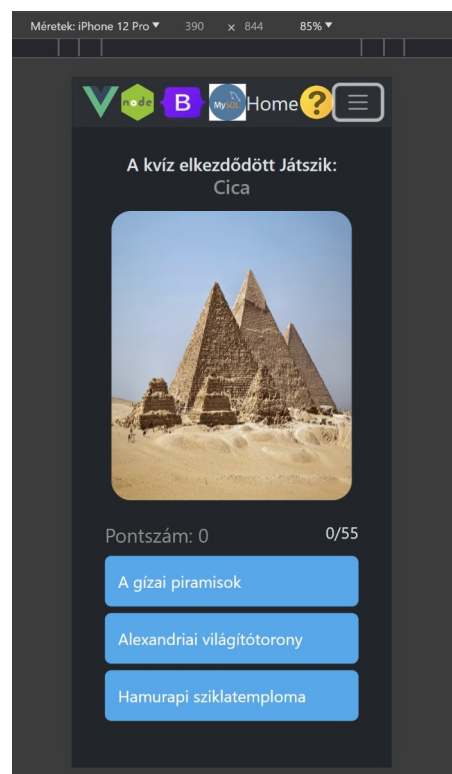
Az alkalmazás felhasználói felületét különböző böngészőkben és különböző eszközökön teszteltük, hogy biztosítsuk a megfelelő megjelenést és működést minden platformon.

A projekt egyik alapigénye volt, hogy mind PC-n, mind mobil platformokon is működjön, ezért a megjelenítést is mobil kijelző mértben is ellenőriztük rendszeresen.

Itt egy iPhone 12 Pro képernyőjére igazítva teszteljük a programunk helyes megjelenítését.

A Selenium automatizált tesztelést is használtunk felhasználói felület ellenőrzésére és az interaktív funkciók működésének biztosítására.

Funkcionális Tesztek:



a

Teszteltük az összes funkciót, beleértve a kérdések megjelenítését, a válaszokat, a pontszámítást és az eredmények kezelését.

A Selenium segítségével automatizált tesztek készítettünk a különböző funkciók működésének ellenőrzésére különböző bemenetekkel és körülményekkel.

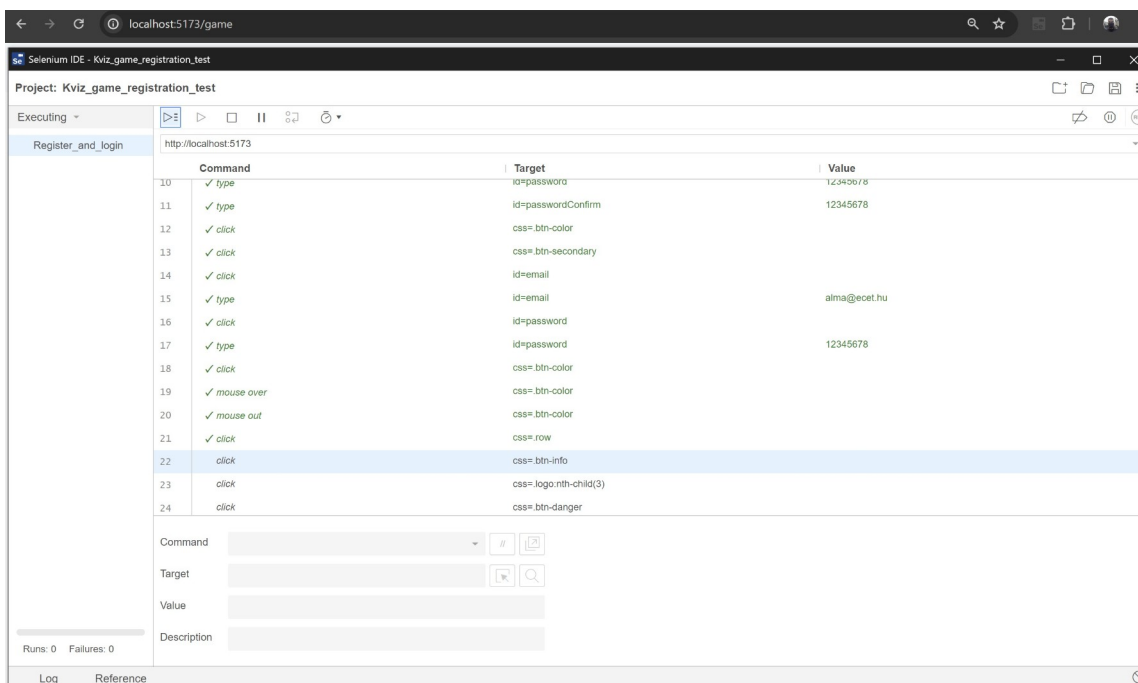
A fejlesztés során folyamatosan végeztünk mind manuális és automatizált tesztek minden új funkció vagy változtatás bevezetésekor.

A Selenium tesztek automatizálták a gyakran ismételt tesztek, és lehetővé tették számunkra, hogy azonnal észleljük és javítsuk az esetleges hibákat.

Következtetés

A kvíztérkép fejlesztése során végzett tesztek és folyamatos ellenőrzések segítettek biztosítani a szoftver minőségét és megbízhatóságát. A Selenium használata lehetővé tette számunkra, hogy hatékonyan teszteljük az alkalmazás interaktív részeit és automatizáljuk a tesztelési folyamatot, ezáltal időt és erőforrásokat takarítva meg.

A programot a Selenium tesztelési webes keretrendszer segítségével automatizáltan is teszteltük, teszteseteket írtunk a bejelentkezés, a regisztráció, és a játékmenet ellenőrzésére is, hogy a folyamatos módosítások mellett is mindig ellenőrizni tudjuk a program működőképességét.



MySQL adatbázis tesztelését is külön végeztük, hiszen a MySQL adatbázis kulcsfontosságú eleme a kvízzjáték alkalmazásnak, és annak biztosítani kell a megfelelő működését és adatintegritását. A tesztek segítségével biztosíthatjuk az adatbázis megfelelő működését és az adatok megbízhatóságát, ami kulcsfontosságú az alkalmazás zavartalan működéséhez.

A MySQL adatbázis tesztelése során ellenőrizzük az adatbázis műveletek helyes végrehajtását, az adatok megfelelő tárolását és a lekérdezések pontos eredményeit. A MySQL adatbázis tesztelése elengedhetetlen a kvízzjáték alkalmazás sikere és stabilitása szempontjából.

Az adatbázis adatait, mezőit mindhárman külön-külön is manipuláltuk, ezért a tesztelés itt is kiemelt fontosságú volt.

Feladatok

Gábor: A projektfeladat elején az adatbázis elkészítését vállaltam a kvízzjátékhoz. Ez a feladat több szempontból is érdekes volt számomra, és kihívásokkal teli tapasztalatokkal gazdagított.

Az első lépés az adatbázis tervezése volt, ami magában foglalta a mezők meghatározását és az adatok strukturálását. Mivel a kvízzjáték különböző kategóriákra és nehézségi szintekre épült, gondoskodnom kellett arról, hogy az adatbázis képes legyen kezelni ezt a sokféleséget.

Egy másik érdekes része a projektnek az volt, amikor a mezőket készítettem és beállítottam az adatbázisban. Itt fontos volt figyelembe vennem a játék logikáját és az adatok összehangolását a játékkal, hogy a játékosok számára összetett és kihívást jelentő kérdéseket készíthessek.

A fejlesztés során sok érdekes feladattal találkoztam, például a felhasználói felület és a szerveroldali logika összekapcsolásával. Ezek a feladatok megmutatták számomra, hogy az adatbázis-tervezés és -kezelés milyen fontos szerepet játszik egy játékfejlesztési projektben, és milyen lehetőségeket kínál a kreativitás és a problémamegoldás számára.

Összességében nagyon élveztem ezt a projektet, és nagy örömmel láttam, hogy az általam tervezett és elkészített adatbázis hozzájárul a kvízzjáték élményéhez. Ez a tapasztalat további motivációt adott számomra a játékfejlesztés terén való további munkához és tanuláshoz.

András: A játék logikájának, működésének részeként számos fontos funkciót kellett megvalósítanom, például a kérdések és válaszok kezelését, a játékosok válaszainak

ellenőrzését és a pontszámítást. Az adatbázisból vagy fájlból betöltött kérdéseket időzítve és időben megfelelően jelenítettem meg a játékosoknak, figyelembe véve a játék dinamikáját és nehézségi szintjét.

A játékosok válaszait megfelelően értékeltem, és a helyes és helytelen válaszok alapján frissítettem a pontszámokat. Fontos volt a pontszámítás részletes tervezése és megvalósítása annak érdekében, hogy a játékosoknak pontos visszajelzést adjak a teljesítményükről és lehetővé tegyem a versenyképességet.

A játék végén összegyűjtöttem és megjelenítettem az eredményeket, lehetőséget adva a játékosoknak arra, hogy áttekinthessék a teljesítményüket és újra próbálkozzanak, ha szeretnék. A hibakezelés és ellenőrzés során kiemelt figyelmet fordítottam arra, hogy minden forgatókönyvet lefedjek és biztosítsak egy zökkenőmentes játékelményt minden játékos számára.

Az első lépés az volt, hogy közösen kitaláltuk a keretrendszereket, a platformokat, és ebben kellett kialakítani az alkalmazás szerkezetét.

A backend funkciók megtervezése során meghatároztam az API végpontokat és a velük kapcsolatos vezérlőket. Ezek a végpontok különböző funkciókat tettek lehetővé, például kérdések lekérdezése, válaszok kezelése, felhasználói adatok frissítése stb.

A teljes fejlesztés során folyamatosan teszteltük a frontendet, és a backendet, beleértve az API-k működését és az adatbázis kapcsolódását.

Eszter:

A kvízzjáték designjának tervezése és kialakítása kihívást jelentett, de egyben rendkívül inspiráló és örömteli feladat is volt számomra. A játék designjának megalkotása során nagy hangsúlyt fektettem arra, hogy egy felhasználóbarát és vonzó felhasználói élményt nyújtsak, amely élvezetessé teszi a játékot minden egyes játékos számára.

Miután meghatároztam az alapvető struktúrát, nekiláttam a tényleges design kialakításának. Nagy hangsúlyt fektettem arra, hogy a játékot élénk és vonzó vizuális stílussal ruházzam fel,

amely megfelel a játék hangulatának és témájának. Például a kezdőképernyőn színes és csábító elemeket használtam, hogy azonnal megragadjam a játékosok figyelmét és érdeklődését.

A menük és ablakok tervezése során fontos volt számomra, hogy egyszerű és könnyen áttekinthető legyen a felhasználók számára. Széles körben használtam az ikonokat és gombokat, hogy navigálásuk intuitív legyen, és könnyen elérjék a kívánt funkciókat és lehetőségeket.

Az egész folyamat során arra törekedtem, hogy a játék designja és menüi ne csak esztétikailag vonzóak legyenek, hanem funkcionálisan is hatékonyak és felhasználóbarátak legyenek. Ennek eredményeként sikerült egy olyan játékot létrehoznom, amelyben a játékosok könnyen és élvezettel navigálhatnak, miközben teljes mértékben átélhetik a kvízzjáték izgalmát és kihívásait. Kiemelve még egy picit, többek között a footer-t készítettem, a stílusnál beállítottam fixre a pozíciót abból a meggondolásból, hogy maradjon fixen az oldal alján. Magam sem tudom miért tettem ilyet talán csak a fáradtság viccelődött az agyammal. Tehát az oldalon ahogy görgettem fel vagy le, a monitor alján fixen maradt a footer, így a görgetéssel együtt mozgott. Tíz perc után már kezdett egy kicsit frusztrálni, hogy nem tudom ennek okát, ezért korrigálni sem tudom. Húsz perc után olyanokat is próbáltam, ami tudtam, hogy nem használ – például ráolvasást a bibliából –, így pihentettem a dolgot öt percig. Visszatértem a problémához és egy bizonyos idő elteltével odasétáltam az ablakhoz, kinyitottam, kezembe vettem a laptom – de csak mert eszembe jutott egy előadás a gravitációról, amit anno a fizika tanárom tartott –, és arra gondoltam kipróbálok még egy valamit. Kitöröltem a 'position: fixed;' sort. Ez után ott funkcionált a footer ahova szántam. Számomra ebből az a tanulság, hogy nem elég csak pár percre pihentetni valamit. Ha valami nagyon nem megy másra kell gondolni, kiszakadni az adott feladatból egy kis időre.

Összegezés/Konklúzió

Első lépésként bemutattuk egymásnak az ötleteinket, amelyekkel készültünk a projektbe. Mindannyian lelkesen vetettük bele magunkat az ötletek megosztásába, és örömmel láttuk, hogy mindegyik ötletnek megvolt a maga egyedisége és potenciálja. Nehéz volt választani, mivel mindegyik ötlet megfelelt volna a vizsgakövetelményeknek, de végül sikerült összeraknunk egy vázlatot, amelyben minden fontos pontot összefoglaltunk.

Az oldal szerkezete eleinte nem volt egységes, hiszen mindenki más elképzelésekkel rendelkezett arról, hogy hogyan kellene kinéznie és működnie az oldalknak. Azonban ahogyan haladtunk előre a folyamatokban, egyre tisztábban láttuk, hogyan kellene megvalósítani az egyes részleteket, és hogyan kellene összekapcsolni azokat egymással.

Az adatbázis létrehozása során figyelmet fordítottunk arra, hogy minden szükséges adatot és információt tartalmazzon a játékosokról és a játékhoz kapcsolódó adatokról. A felhasználók regisztrációjától kezdve a pontszámok tárolásáig minden fontos részletet beépítettünk az adatbázisba.

A backend fejlesztése során kiemelten fontos volt az adatbázis összekapcsolása az oldallal, hogy a játék folyamatosan és hatékonyan működjön. A Node.js használata lehetővé tette számunkra, hogy egyszerűen és hatékonyan kezeljük az adatokat és kiszolgáljuk az API-kat a frontend számára.

Ez a projekt rendkívül érdekes tapasztalat volt számunkra, mivel lehetőséget kaptunk arra, hogy megértsük és megtanuljunk, hogyan lehet egy "egyszerű" projektet részekre és részfeladatokra bontani. Megtapasztaltuk, hogy milyen fontos a kommunikáció és az együttműködés a csapatban, és mennyire nagy segítséget jelentenek a modern eszközök,

mint például a Trello, Git, GitHub, Teams és Discord, amikor egyeztetni és munkafolyamatokat megosztani kell egymással, és összehangolni a folyamatokat.

Ezek az élmények és tapasztalatok segítettek minket abban, hogy hatékonyabban és összehangoltabban dolgozzunk együtt, és sikeresen hozzuk létre a projektet.

Mindenképpen nagyon tanulságos és hasznos volt számunkra a közös munka, és remek hogy ezt a fajta részekre bontást, és együttműködést, összehangolást, csapatkommunikációt „iskolai környezetben” tudtuk megtapasztalni.

Elmaradt Célok

A játék elkészítésén kívül az egyik célunk volt ezt egy webtárhelyen elérhetővé tenni szélesebb körben, több száz képpel, tematikusan kérdéskörökre bontva.

Ez azonban részben „fejlesztési idő”, részben megfelelő tárhely hiányában nem valósult meg.

Felmerült, és az egyik fejlesztési variációnkban időre ment a kvíz, egy számláló mérte az időt, valamint visszaszámolt a gondolkodási időből, ezt azonban nem tudtuk megfelelően a kérdések számával, a pontozásos rendszerrel integrálni, ezért elvetettük.

Felmerült, hogy a játék működhetne egyfajta dekoratív fényképalbumként is, ami véletlenszerűen teljes képernyőn megjeleníti a kérdésekben szereplő képeket, amennyiben a játékos túl sokáig inaktív, azt feltételezve, hogy közben mással foglalkozik, és így kelleme háttérképet biztosítana, valamint meggátolná a képernyő azonos képpontjainak a túlműködését.

Felmerült a konzulens tanárunk részéről, hogy az 'Anonym' -nem bejelentkezett usernek csak egy demo verziót, pár kérdést lehetne elérni, ezt nem valósítottuk meg időszűke miatt, valamint részleges egyet nemértés miatt.

Fejlesztési lehetőségek

Számtalan egyéb ötletünk van, valamint a fent említett célok teljesítése és finomítása.

Jutalmazási rész, nyereményjáték a legjobban teljesítő játékosoknak,
fórum és hozzászólási lehetőség, ahol emlékeket, tippeket lehetne az adott képek -és
helyszínekről megosztani egymással, véleményeket írni stb.

A játékot több száz vegyes témájú képpel bővíteni, és a választható kategóriákat is
kidolgozni.

Lehetne egy webes tárhelyen elérhetővé tenni, és további adatvédelmi és autentikációs
fejlesztések után a „nagyközönség” elé engedni.

Felhasznált Irodalom

Alapvetően a tanáraink által tanított és tanult módszereket és technológiákat használtuk a project elkészítése során.

A tanultakat a szakirodalommal, az interneten elérhető anyagokkal egészítettük ki.

W3 Sools <https://www.w3schools.com/>

Vue.js <https://vuejs.org/>

Node.js <https://nodejs.org/en>

Node Express framework <https://expressjs.com/>

Node Bcrypt <https://www.npmjs.com/package/bcrypt>

Node jwt JasonWebToken <https://www.npmjs.com/package/jsonwebtoken>

Node Mysql2 <https://sidorares.github.io/node-mysql2/docs>

XAMP <https://www.apachefriends.org/hu/index.html>

MySQL <https://www.mysql.com/>

Bootstrap <https://getbootstrap.com/>

Axios <https://axios-http.com/docs/intro>

VeeValidate & Yup <https://vee-validate.logaretm.com/v4/>

Pinia <https://pinia.vuejs.org/>

Adobe Stock Free Images <https://stock.adobe.com/hu/>

Pexels Free Images <https://www.pexels.com/hu-hu/>

Pixabay <https://pixabay.com/photos/>

ChatGPT 3.5 <https://chatgpt.com/>

Dia <http://dia-installer.de/>

Youtube programozási videók <https://www.youtube.com/>