



ZÁRÓDOLGOZAT

Készítették:

Kerényi Tamás – Czibik Lóránt Patrik – Nagy Gergő

Konzulens:

Németh Bence

Miskolc

2023.

Miskolci SZC Kandó Kálmán Informatikai Technikum

Miskolci Szakképzési Centrum

SZOFTVERFEJLESZTŐ- ÉS TESZTELŐ SZAK

ZÁRÓDOLGOZAT

Quizter

Kvízes weboldal

Kerényi Tamás – Czibik Lóránt Patrik – Nagy Gergő

2022-2023

Tartalomjegyzék

TÉMAVÁLASZTÁS	3
HASZNÁLT TECHNOLÓGIÁK, KÖRNYEZETEK, NYELVEK	3
Vue.js – Frontend	3
Single-File Component	3
API stílusok	4
Pinia	5
A frontend felépítése	6
Navigációs Sáv	6
Lábléc	7
Szint	8
Téma gomb	8
Töltés	9
Hiba	9
App	9
Főoldal	10
Kvíz Beállító	11
Játékmenet	12
Regisztráció	13
Bejelentkezés	14
Jelszó visszaállítása	14
Profil	15
Ranglista	16
Kérdések	17
Kérdés	18
Új kérdés	19
Rólunk	20
Adatvédelem	20
Node.js – Backend	21
Express.js	21
MongoDB – Adatbázis	22
Docker	23
Mi az a Docker?	23
Mik azok a konténerek?	23
Miért használunk konténereket?	23
FORRÁSKÓD	24
FORRÁSJEGYZÉK	24

TÉMAVÁLASZTÁS

A szakdolgozatunk témájára sok ötletünk volt, de végül egy kvízes weboldal mellett döntöttünk. Szerettünk volna valami olyat alkotni, ami interaktív, szórakoztató és később is szívesen tekintünk rá vissza.

A Quizter a játzmán felül megvalósít egy felhasználói platformot is, ahol megtekinthető mások profilja vagy eredménye a ranglistán. Ezen kívül képes az adminisztrátorok számára a kérdések listázására, módosítására vagy törlésére.

HASZNÁLT TECHNOLÓGIÁK, KÖRNYEZETEK, NYELVEK

Vue.js – Frontend

A Vue egy javascript keretrendszer felhasználói felületek fejlesztésére. Az alap HTML-re, CSS-re, Javascriptre épül, és biztosít egy deklaratív és komponens alapú programozási modellt, amely elősegíti a felhasználói felületek hatékony fejlesztését, legyen az egyszerű vagy komplex.



1. ábra: A Vue logója [\[1\]](#)

A Vue egy olyan keretrendszer és ökoszisztéma, amely a frontend fejlesztéshez szükséges általános funkciók többségét lefedi. A web azonban rendkívül változatos - a dolgok, amelyeket a webre építünk, drasztikusan eltérhetnek formájukban és méretükben. Ezt szem előtt tartva a Vue-t úgy tervezték, hogy rugalmas és fokozatosan átvehető legyen. [\[2\]](#)

A felhasználási esettől függően a Vue különböző módon használható:

- Statikus HTML továbbfejlesztése építési lépés (build step) nélkül.
- Webkomponensként történő beágyazás bármely oldalra.
- Egyoldalas alkalmazás (SPA).
- Fullstack / Server-Side Rendering (SSR).
- Jamstack / Statikus oldal generálás (SSG).
- Asztali, mobil, WebGL és akár a terminál célzása.

Single-File Component

A legtöbb Vue projektben a komponenseket egy HTML-szerű fájl formátumban, az úgynevezett Single-File Component (más néven *.vue fájlok, rövidítve SFC) segítségével írják. A Vue SFC, ahogy a neve is jelzi, egyetlen fájlba foglalja a komponens logikáját (JavaScript), sablonját (HTML) és stílusait (CSS). Az SFC a Vue egyik legmeghatározóbb jellemzője, és a Vue komponensek írásának ajánlott módja. [\[2\]](#)



2. ábra: A HTML5 logója [\[3\]](#)



3. ábra: A CSS3 logója [\[4\]](#)



4. ábra: A Javascript logója [\[5\]](#)

API stílusok

A Vue komponensek két különböző API stílusban írhatóak: Options API és Composition API.

Az Options API segítségével egy komponens logikáját beállítás részekkel, mint például *data*, *methods* és *mounted* segítségével határozzuk meg. Az beállítások által definiált tulajdonságok ezen belül függvények, amelyek a komponens példányára mutatnak.

A Composition API segítségével egy komponens logikáját importált API-funkciók segítségével határozzuk meg. Az SFC-kben a Composition API-t jellemzően a `<script setup>` segítségével használjuk. A `<script setup>` részben deklarált import-ok és felső szintű változók / függvények közvetlenül felhasználhatóak a *template*-ben. [\[2\]](#)

A frontend kódja az **Options API** stílusban íródott.

Pinia

A projekthez a Pinia *állapotkezelő keretrendszert* használtuk. A Pinia egy tároló könyvtár a Vue-hoz, ami lehetővé teszi az állapot megosztást komponensek vagy oldalak között.



A Pinia az alábbiakat nyújtja:

- Devtools támogatás:
 - Idővonal az akciók, mutációk követésére.
 - A store-ok a használt komponensekben jelennek meg. *5. ábra: A Pinia logója* [\[6\]](#)
 - Időutazás (time travel) és könnyebb hibakeresés.
- Hot Module Replacement (HMR):
 - A store-ok módosítása az oldal újratöltése nélkül.
 - Minden meglévő állapot megtartása fejlesztés közben.
- Pluginok: bővítsd ki a Pinia funkcióit pluginekkal.
- Megfelelő TypeScript támogatás vagy automatikus kitöltés Javascript használók számára.
- Szerver oldali renderelés támogatása.

A Vuex-hez képest a Pinia egyszerűbb API-t kínál kevesebb komplikációval, és Composition stílusú API-val. [\[7\]](#)

A frontend felépítése

A frontend kinézete egy minimalista dizájnt követ, de azért figyeltünk arra, hogy ne legyen minden túl minimalista. Törekedtünk arra, hogy az oldal reszponzív legyen a mobilos használat érdekében. A bemutatás a komponensekkel kezdődik, majd az oldalakkal folytatódik, ami azért fontos, mivel a komponensek többször előfordulhatnak, és nem önállóan, hanem az alábbi oldalakon használva. Az oldalak leírása az **App** résznél kezdődik.

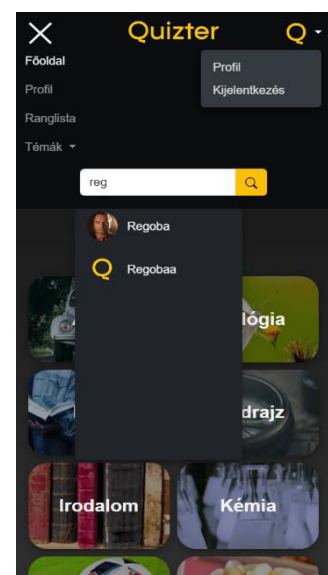
Navigációs Sáv

A navigációs sávval (navigation bar) a fő-, profil, ranglista, kvíz beállító, bejelentkezés és regisztráció oldalakra lehet navigálni. A kereső segítségével más felhasználókat lehet keresni.

Felépítése:

- A logó ami a főoldalra navigál.
- Navigációs linkek a fő-, profil és ranglista oldalakra.
- A “Témák” lenyíló menü, ami átnavigál a kvíz beállító oldalra a beállított téma alapján.
- Keresőmező és keresógomb.
- Bejelentkezett felhasználó esetén a felhasználónév és alatta a szint komponens. Mellette a felhasználó képe, amire kattintva egy lenyíló menü segítségével megtekinthető a profil vagy ki lehet jelentkezni.
- Nem bejelentkezett felhasználó esetén egy felhasználó ikon, amire kattintva egy lenyíló menü segítségével elérhető a bejelentkezés vagy regisztráció oldal.

Az oldalon mobilos nézetben egy hamburger menü jelenik meg a bal oldalon, a logó közepén, és a felhasználói kép vagy felhasználó ikon jobb oldalt.



Lábléc

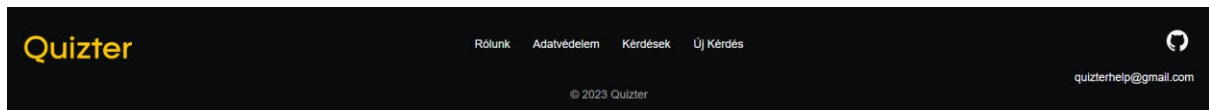
A lábléccel (footerrel) a fő-, rólunk és adatvédelem oldalakra lehet navigálni. Adminként a kérdések és új kérdés oldalakat is el lehet érni innen.

6. ábra: A navigációs sáv asztali nézete, témák lenyíló menü megnyitva

7. ábra: A navigációs sáv mobilos nézete, felhasználó lenyíló menü megnyitva

Felépítése:

- A logó, ami a főoldalra navigál.
- Ha nem admin a felhasználó akkor csak a rólunk és adatvédelem oldalakra navigációs linkek.
- Ha admin a felhasználó akkor az előbb említett oldalakon kívül a kérdések és új kérdés oldalakra navigációs linkek.
- Egy github ikon, ami a frontend forráskódjának a gyűjteményére (repository) linkel.
- Az email cím, kattintásra egy gmail ablakot nyit meg ahol írhat a felhasználó.
- A szerzői jog szövege.



8. ábra: A lábléc asztali nézete



9. ábra: A lábléc mobilos nézete

Szint

A szint komponens a felhasználó tapasztalat pontja alapján adja vissza a szintet és azt, hogy milyen távol van még a következő szint.

Felépítése:

- A szint középre igazítva.
- A szint haladás, ami a tároló szélességének az adott százaléka.

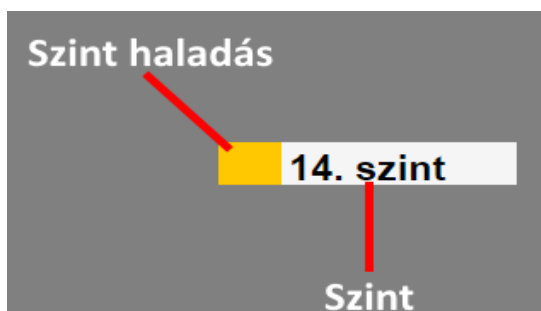
exp: A jelenlegi tapasztalatpont

szXpMin: A jelenlegi szinthez szükséges tapasztalatpont

szXpMax: A következő szinthez szükséges tapasztalatpont

$$\underbrace{\frac{100}{\text{szXpMax} - \text{szXpMin}}}_{\text{Slope}} * \text{exp} + \underbrace{\left(-1 * \frac{100}{\text{szXpMax} - \text{szXpMin}} * \text{szXpMin} \right)}_{\text{yintercept}}$$

10. ábra: A szint számítása



12. ábra: A szint komponens

Szint	Szükséges xp
1	0
2	100
3	200
4	300
5	500
6	750
7	1,000
8	1,500
9	2,000
10	3,500
11	5,000
12	7,500
13	10,000
14	15,000
15	20,000
16	30,000
17	50,000
18	75,000
19	100,000
20	200,000
21	300,000
22	400,000
23	500,000
24	600,000
25	700,000
26	800,000
27	900,000
28	1,000,000
29	1,100,000
30	1,200,000

11. ábra: A szint haladás számítása

Téma gomb

A téma gomb komponens jeleníti meg a főoldalon a témákat. Paraméterként kapja a téma nevét, amit a tároló közepén jelenít meg, és a téma azonosítóját, ami szerint beállítja a téma képét és átnavigál a kvíz beállító oldalra.



13. ábra: A téma gomb komponens

Töltés

A töltés komponens egy forgó körszelet, ami a Quizter logó 'Q' betűjének egy része. Ez jelenik meg amíg a felhasználó vár a szerver válaszára.



14. ábra: A töltés komponens

Hiba

A hiba komponens csakis akkor jelenik meg, ha probléma akad a töltés során. A "Vissza" gombbal a főoldalra lehet navigálni.



15. ábra: A hiba komponens

App

Itt jelennek meg az oldalak a navigációs sáv (navigation bar) és a lábléc (footer) közé ágyazva. Az oldalak egy tárolóban vannak azért, hogy a lábléc (footer) az oldal alján maradjon minden méretben. Itt történik a bejelentkezett felhasználó automatikus beléptetése, ami a bejelentkezés során generált token alapján működik. Ezt a token-t az oldal eltárolja egy sütiben (cookie), ami 1 hét után lejár. Erre azért van szükség, hogy a böngésző bezárása után ne kelljen újra bejelentkezni.

Főoldal

A főoldalon keresztül lehet témát választani, vagy az előbb említett módon, a navigációs sávban található "Témák" lenyíló menü segítségével.

A rendelkezésre álló témák:

- Autók
- Biológia
- Fizika
- Földrajz
- Irodalom
- Kémia
- Sport
- Szórakoztatás
- Technológia
- Történelem
- Zene
- Vegyes



16. ábra: A főoldal asztali nézete

Felépítése:

- A logó, alatta a "Teszteld a tudásod" szöveg.

- A téma gombok, amik a kvíz beállító oldalra navigálnak.



17. ábra: A főoldal mobilos nézete

Kvíz Beállító

A kvíz beállító oldalon lehet a játszmát személyre szabni. A zöld szín jelöli a könnyű, a sárga a közepes és a piros a nehéz nehezítéseket. A beállítások és paramétereik nehézség szerint növekvő sorrendben:

1. A kérdések nehézsége:

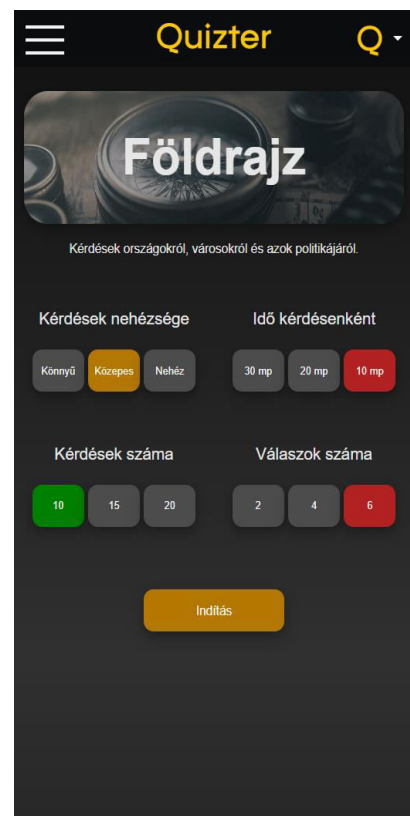
- Könnyű
- Közepes
- Nehéz

2. Az idő kérdésenként:

- 30 másodperc
- 20 másodperc
- 10 másodperc

3. A kérdések száma:

- 10 kérdés
- 15 kérdés
- 20 kérdés

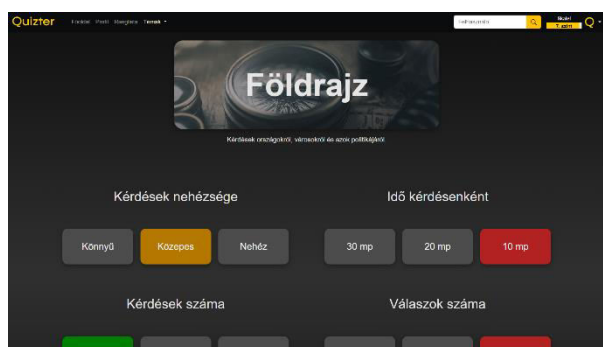


4. A válaszok száma:

- 2 válasz
- 4 válasz
- 6 válasz

Felépítése:

- A téma képe és szövege.
- A téma magyarázata.
- A beállítások, amik tartalmazzák a beállítás nevét, és alatta a 3 hozzá tartozó gombot.
- Az "Indítás" gomb, ami átnavigál a játékmenet oldalra, ha mind a 4 beállítás megvan



18. ábra: A kvíz beállító oldal mobilos nézete

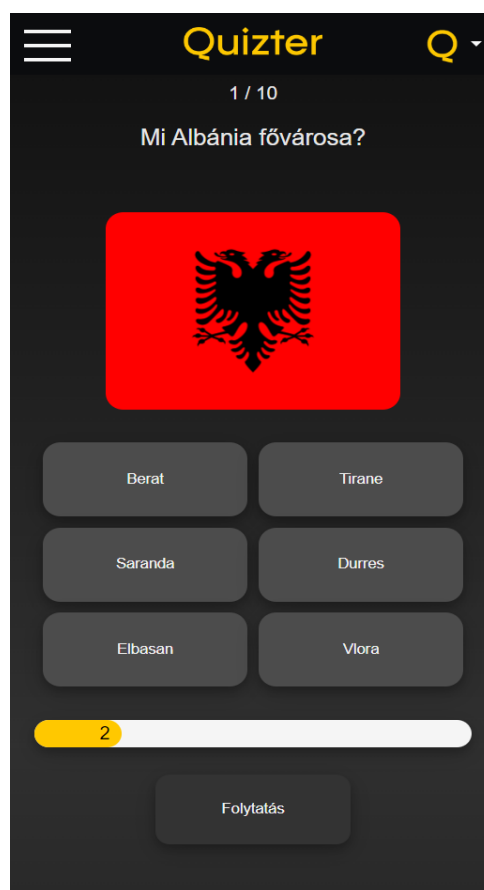
19. ábra: A kvíz beállító oldal asztali nézete

adva.

Játékmenet

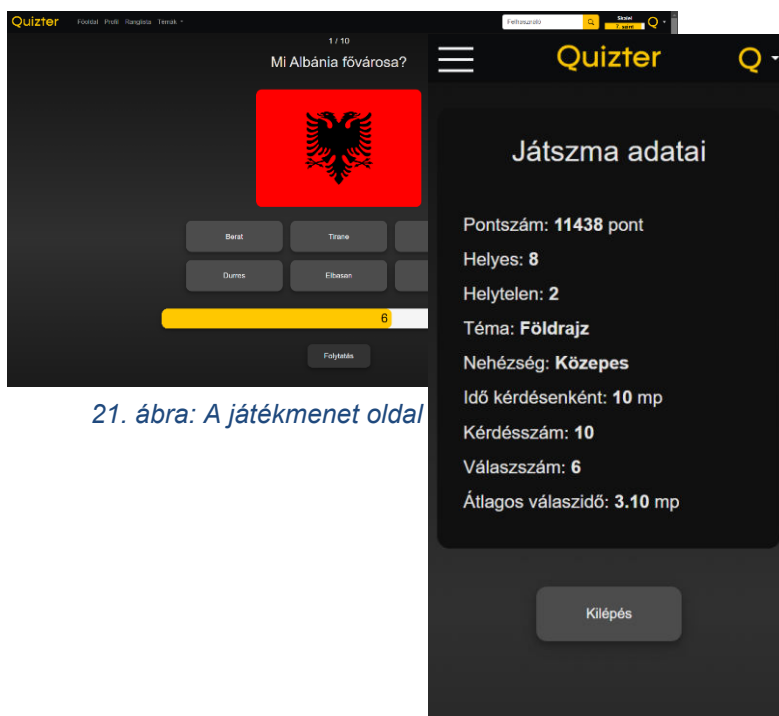
A játékmenet oldalon körönként jelennek meg új kérdések. Ezekre az adott időn belül kell választ adni. Ha ez nem sikerül, akkor az helytelen válasznak számít. Az adott időn belül megadott helytelen válasz nem ér pontot. Viszont ha helyes, akkor a maradt idő szerint növekszik a pontszám. A játszma végén a nehézség és válasz szám nehezítése alapján szorzódik a pontszám (közepes: 1,25, nehéz: 1,5), frissül a bejelentkezett felhasználó tapasztalat pontja, játszma száma, és személyes rekordja, ha nagyobb a pontszám az előzőnél, majd egy listában jelennek meg a játszma adatai.

Felépítése:



- Egy számláló, ami jelzi a jelenlegi és maradandó körök számát.
- A kérdés szövege.
- A kérdés képe.
- A választott mennyiségű (2, 4 vagy 6) válasz gombja, ami kattintás után annak helyessége szerint lesz kiszínezve (zöld, ha helyes, piros, ha helytelen). Válaszadás után kikapcsolnak.
- Egy visszaszámláló, ami jelzi a maradandó időt, és jobbról balra csökken a szélessége másodpercenként. Az animáció fél másodperc alatt történik.
- A “Folytatás” gomb, ami elindítja a következő kört. Az utolsó körben lecseréli a “Befejezés” gomb. Válaszadás után használható.

A játszma végeztével egy lista jelenik meg, ami tartalmazza az adott játszma beállításait és a felhasználó teljesítményét. Ha a játékos új személyes rekordot döntött, akkor azt is kijelzi.



21. ábra: A játékmenet oldal

22. ábra: A játékmenet oldal mobilos nézete

22. ábra: A játszma adatai lista

Alatta egy gomb, ami visszavigál a főoldalra.

Regisztráció

A regisztráció oldalon hozható létre egy Quizter profil. A 4 beviteli mező reguláris kifejezés (regex) alapján vizsgálja a beírt tartalmat. Ha a felhasználónév vagy email foglalt, vagy nem egyezik az ismételt jelszó, akkor a beviteli mező alatt egy üzenet jelenik meg.

Felépítése:

23. ábra: A regisztráció oldal asztali és mobilos nézete

- A felhasználónév beviteli mező, a beírt szöveg betű vagy szám lehet.
- Az email beviteli mező, a beírt szövegnek tartalmaznia kell a “@” és “.” karaktereket.
- A jelszó beviteli mező, a beírt szövegnek tartalmaznia kell betűt és számot.
- Az ismételt jelszó beviteli mező, a beírt szövegnek egyeznie kell a jelszóéval.
- Egy jelölőnégyzetet (checkbox), ami azt jelzi, hogy a felhasználó elfogadja az adatvédelmi nyilatkozatot.
- A “Regisztráció” gomb.

Bejelentkezés

A bejelentkezés oldalon is vizsgálva van a felhasználónév és jelszó formátuma. Ezen felül, ha nem helyesek a bejelentkezési adatok, akkor a bemeneti mezők alatt megjelennek tájékoztató üzenetek.

Felépítése:

- A felhasználónév beviteli mező, a beírt szöveg betű vagy szám lehet.
- Az “Elfelejtett jelszó?” link, ami előhoz egy felugró ablakot. Itt a felhasználó megadja az email címét, ahova a jelszó visszaállítással kapcsolatosan küldünk egy emailt.
- A jelszó beviteli mező, a beírt szövegnek tartalmaznia kell betűt, számot és speciális karaktert.
- A “Bejelentkezés” gomb.

24. ábra: A bejelentkezés oldal asztali és mobilos nézete

Jelszó visszaállítása

A jelszó visszaállítása oldalon módosítható a felhasználó jelszava. Az előbb említett email ide irányítja át a felhasználót. Ha nem megfelelő az url-ben található jelszó változtatási kérelem tokenje, akkor az oldal nem tölt be, és a kérelem automatikusan elutasított.

Felépítése:

- A jelszó beviteli mező, a beírt szövegnek tartalmaznia kell betűt és számot.

25. ábra: A jelszó visszaállítása oldal asztali és mobilos nézete

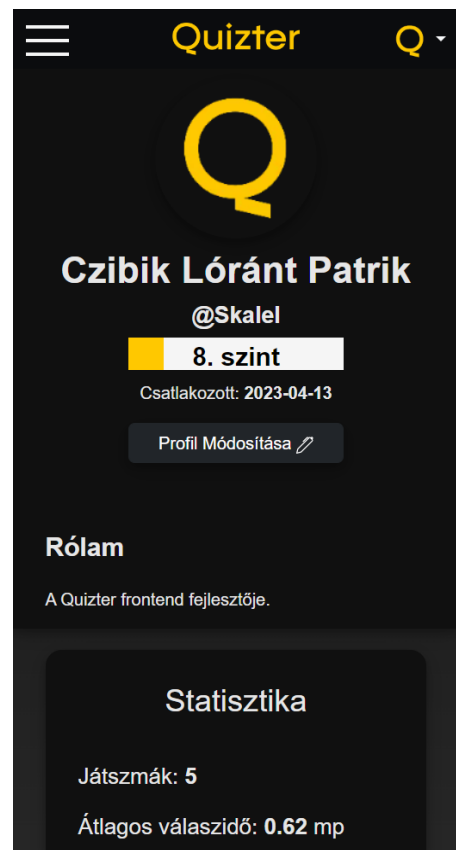
- Az ismételt jelszó beviteli mező, a beírt szövegnek egyeznie kell a jelszóéval.
- A “Jelszó módosítása” gomb.

Profil

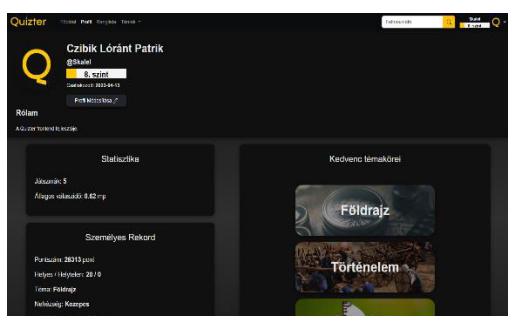
A profil oldalon tekinthetők meg a felhasználók adatai. Ha a bejelentkezett felhasználó a saját profilját nyitja meg, akkor szerkeszthető a profilkép, név, leírás és a 3 beállítható téma.

Felépítése:

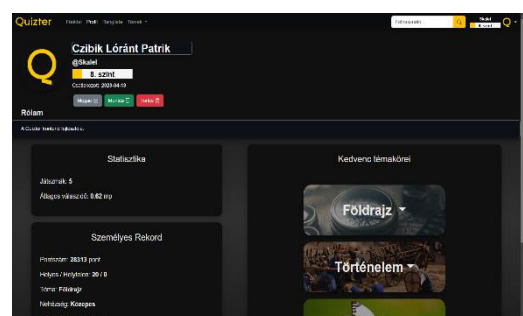
- A profilkép.
- A név.
- A felhasználónév.
- A szint komponens.
- A csatlakozási dátum.
- A felhasználó leírása.
- Szerkesztés esetén három gomb: “Mégse”, “Mentés” és “Törlés”.
- A felhasználó statisztikája, ami tartalmazza azt, hogy mennyi játszmája van és az átlagos válaszáidejét.
- A felhasználó személyes rekordja, ami tartalmazza a pontszámot, a helyes / helytelen válaszok számát, a témát, nehézséget, kérdésszámot és válasszámot.
- A felhasználó által beállított 3 téma.



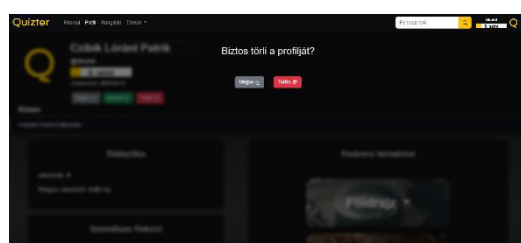
26. ábra: A profil oldal mobilos nézete



27. ábra: A profil oldal asztali nézete



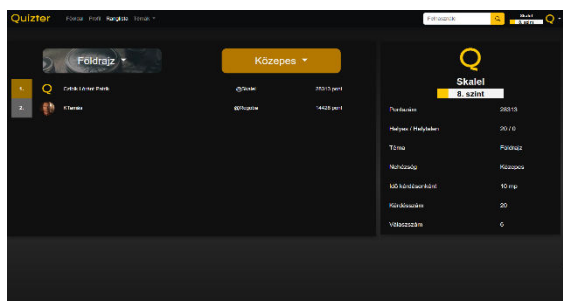
28. ábra: A módosítás opció kiválasztva



29. ábra: A törlés opció kiválasztva

Ranglista

A ranglista oldalon téma és nehézség szerint szűrhetőek a személyes rekordok. 2 részre van bontva, az elsőben a felhasználók eredményei, és a második részben pedig a választott rekord részletesebb leírása. Mobilos nézetben érintésre automatikusan legördül ehhez a részhez, és nem jelenik meg a név a ranglistán.



30. ábra: A ranglista oldal asztali nézete

Felépítése:

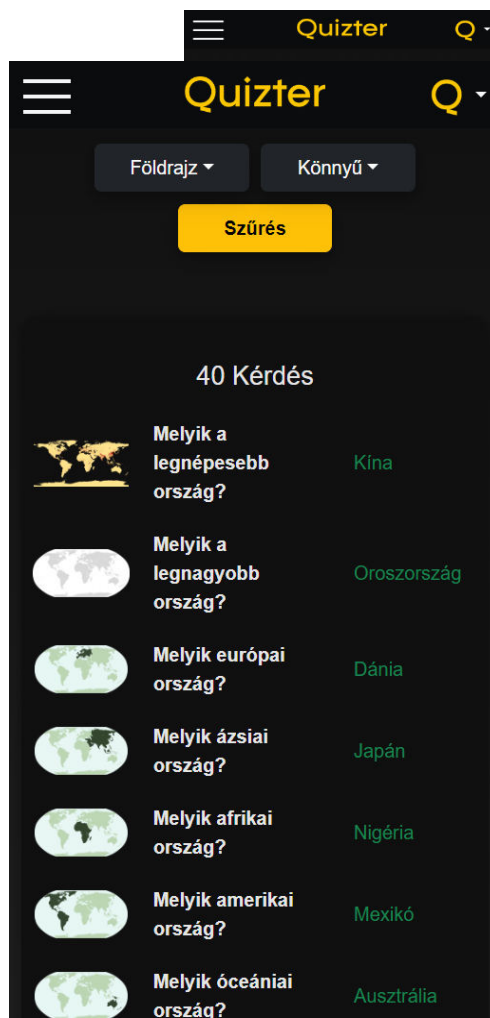
1. Ranglista:

- A téma és nehézség szűrője, amit egy lenyíló menü segítségével lehet beállítani.
- A helyezés.
- A profilkép.
- A felhasználónév.
- A név.
- A pontszám.

2. Rekord:

- A profilkép, ami kattintásra átirányít az adott felhasználó oldalára.
- A felhasználónév.
- A szint komponens.
- A pontszám.
- A helyes / helytelen válaszok száma.
- A téma.
- A nehézség.

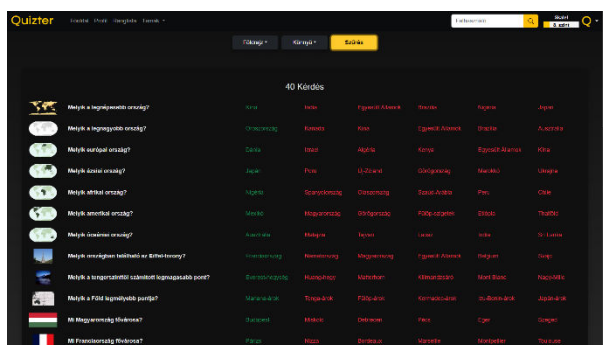
Kérdések



A kérdések oldalon jelennek meg a kérdések téma és nehézség szerint. Egy számláló visszaadja a kérdések számát. Ez mind egy táblázatban van listázva, aminek az egyik sorára kattintva megjelenik a kérdés oldalon a választott elem. Mobilos nézetben nem jelennek meg a kérdések helytelen válaszai.

Felépítése:

- A téma és nehézség lenyíló menü.
- A “Szűrés” gomb, ami ezek szerint listáz.
- A kérdés számláló.
- A kérdések képei, szövegei, 1 helyes és 5 helytelen válaszai.



32. ábra: A kérdések oldal asztali nézete

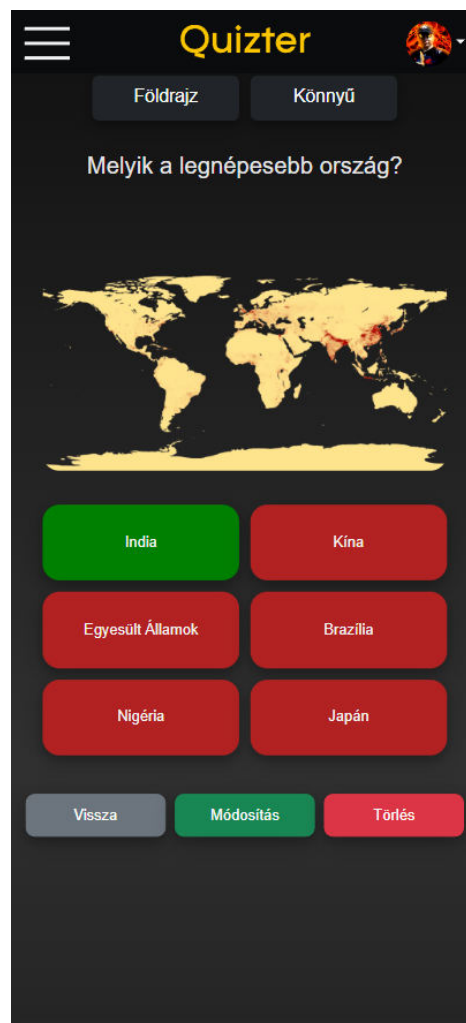
33. ábra: A kérdések oldal mobilos nézete

Kérdés

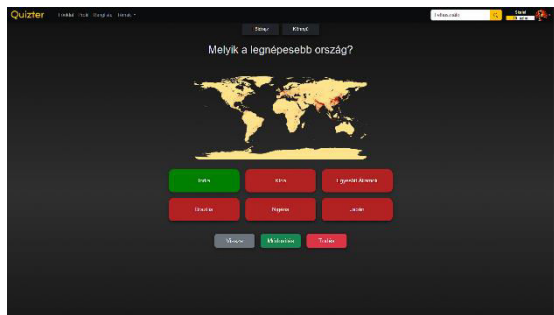
A kérdés oldalon jelenik meg az adott kérdés témája, nehézsége, szövege, képe és 6 válasza. A kérdés megtekinthető, módosítható, vagy törölhető.

Felépítése:

- A téma és nehézség.
- A kérdés szövege.
- A kérdés képe.
- A kérdés 6 válasza.
- A “Vissza” gomb, ami a kérdések oldalra navigál.
- A “Módosítás” gomb, ami a témát és nehézséget lenyíló menüvé, a kérdés szövegét, képét és válaszait pedig beviteli mezővé alakítja. Egy “Mégse”, “Mentés” és kép megjelenítő gombot is létrehoz.

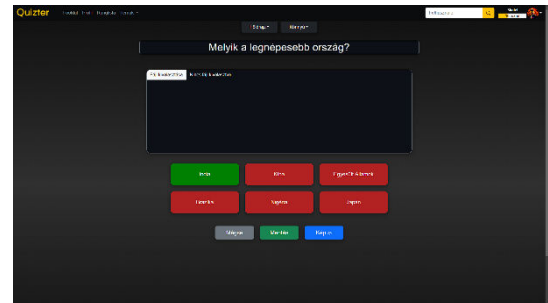


- A “Törlés” gomb, ami a felhasználó megerősítése után eltávolítja a kérdést.

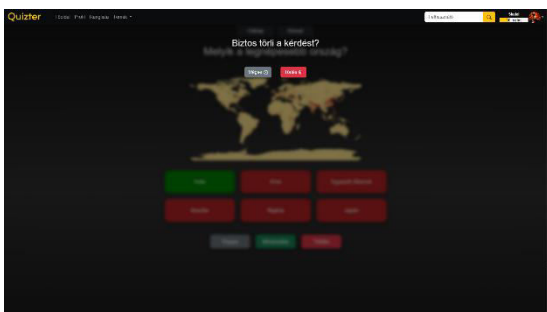


35. ábra: A kérdés oldal asztali nézete

34. ábra: A kérdés oldal mobilos nézete



36. ábra: A módosítás opció kiválasztva



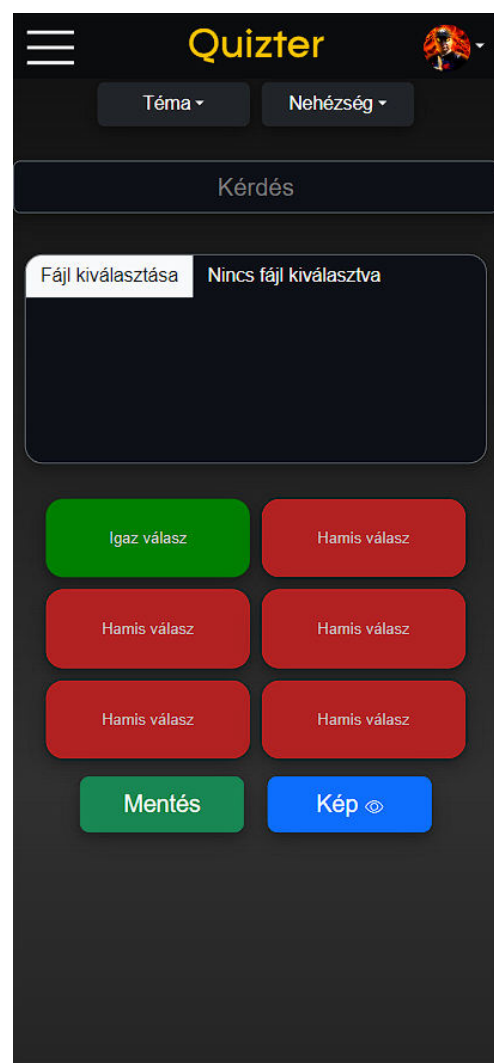
37. ábra: A törlés opció kiválasztva

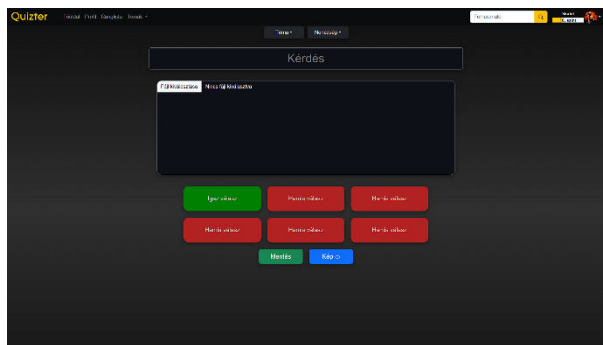
Új kérdés

Az új kérdés oldalon lehet létrehozni kérdéseket. Ehhez be kell állítani a kérdés témáját, nehézségét, szövegét, képét és 6 válaszát. A “Mentés” gomb lenyomásával egy vizsgálat történik, hogy be lett-e állítva a téma, nehézség és kép. Ha igen, akkor az adatok elküldésre kerülnek.

Felépítése:

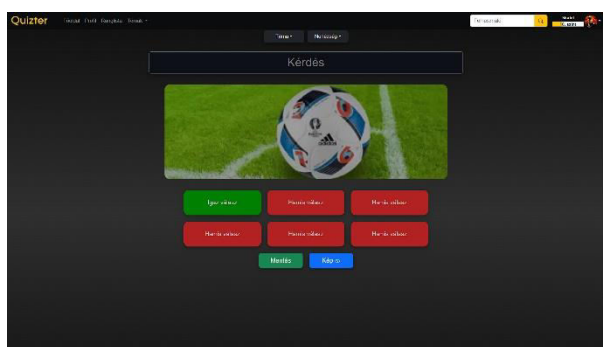
- A téma és nehézség lenyíló menü.
- A kérdés szöveg beviteli mező.
- A kérdés kép beviteli mező.
- A 6 válasz beviteli mezője.
- A “Mentés” gomb.
- A kép megjelenítő gomb, ha a felhasználó megadott képet.





38. ábra: Az új kérdés oldal asztali nézete

39. ábra: Az új kérdés oldal mobilos nézete



Rólunk

A rólunk oldalon megjelennek a képeink, a nevünk és a szerep, amit a projektben betöltöttünk. Ezen kívül még egy carousel tartalmazza a fejlesztés néhány kezdetleges képeit. Az oldal alján pedig az email, amin keresztül el lehet minket érni.

Felépítése:

- A logó, alatta a "Rólunk" szöveg.
- A bemutatkozás szövege.
- A 3 kép, alatta a név és a betöltött szerep(ek).
- A "Galéria a fejlesztésről" szöveg, alatta a fejlesztés carousel.

40. ábra: A kép opció kiválasztva

- Az “Elérhetőség” szöveg, alatta az email.



41. ábra: A rólunk oldal asztali nézete

Adatvédelem

Az adatvédelem oldal a felhasználót tájékoztatja arról, hogy hogyan kerülnek kezelésre és tárolásra az adatai. Ezen kívül tartalmazza az emailt, amin keresztül el lehet érni minket.



42. ábra: Az adatvédelem oldal

Node.js – Backend

A Node.js egy olyan futtató környezet, ami lehetőséget nyújt JavaScriptben írt programok futtatására szervereken. JavaScriptet leginkább a kliens oldalon szokták használni, a böngészőn keresztül. Könnyebbséget nyújt, ha már dolgoztunk frontend oldalon, hogy nem kell egy teljesen új nyelvet megtanulni ahhoz, hogy a backend-et is implementáljuk (Pl: Express.js, Koa, Meteor.js). A másik előnye az, hogy a non-blocking természetéből adódóan alkalmas egyszerre több ezer bejövő 'egyszerű' kérés kezelésére. Ezzel szemben egyetlen számításigényes utasítássorozat végrehajtására nem tűnik éppen ideális



43. ábra: A Node.js logója [8]

választásnak. Rengeteg fontos modullal rendelkezik, ezek többek között: HTTP, HTTPS, UDP, Net, DNS, TLS/SSL, URL, Query String, File System, Path, Events, Process, Child Processes, Buffers, Streams, Crypto. A listából látszik, hogy a Node.js igen alacsony szintű hozzáférést tesz lehetővé a szerver különböző rendszereihez, viszont magasabb szintű rendszereket önmagában nem kínál. Ehhez külső modulokat használhatunk, amik JavaScript és C++ komponensekből állhatnak (ahogy a beépített modulok is). [\[9\]](#)

Express.js

Ezt a keretrendszert úgy tervezték, hogy olyan funkciókkal rendelkezzen, amelyeket más keretek, például a **Rails** vagy a **Sinatra** ihlette. Ez azt jelenti, hogy amikor egy tapasztalt fejlesztő kezét rá helyezi, akkor kényelmesebbnek tartja a már ismert fogalmak kezelését, de a **Node.js** és Csak a JavaScript használatával megnövekszik a fejlesztési sebesség.

Az **Express.js** legfontosabb jellemzői:

1. Minimalista

A keretrendszer mögött az a gondolat nem szerepel, hogy más kérdésekbe kerüljön, amelyek nem közvetíthető az alkalmazásunk és a szerver között, mert ez túlbonyolítja, csakis a legszükségesebb dolgokat biztosítja számunkra.

2. Rugalmas

Minimális tulajdonsága miatt nagy rugalmasságot élvez, mivel a fejlesztő választhatja meg a munkavégzési módját, és megengedheti magának, hogy saját megoldásait megvalósítsa, vagy kész megoldásokat, például **ORM**-et építsen be különféle típusú adatbázisokkal.

3. Alkalmazásokhoz készült

Az **Express.js** célja, hogy a webes alkalmazások, oldalak, akár **REST** szolgáltatások, akár hibrid alkalmazások készítésének megkönnyítése. [\[10\]](#)

MongoDB – Adatbázis

A legnépszerűbb és legelterjedtebb dokumentum-orientált adatbázis rendszer, ami:

- No-SQL adatmodelre épül
- JSON vagy BSON formában tárolja az adatokat
- nagy performancia jellemzi
- nagy skálázhatóság
- nincs SQL injection



44. ábra: A mongoDB logója [\[11\]](#)

A NoSQL (Not only SQL) adatbázisok népszerűsége az utóbbi években kezdett meredeken emelkedni. A relációs adatbázisoktól eltérő megközelítés fő céljai között említhetjük az óriási mennyiségű adatok feldolgozását (bigdata), a skálázhatóságot és a valós idejű rendszereket. Egyre több fejlesztési projekt alapjául szolgál NoSQL adatbázis és az újszerű felépítés új megközelítést igényel a fejlesztőktől és a felhasználóktól. [\[12\]](#)

The screenshot shows the MongoDB Compass interface. On the left, the 'Databases' sidebar lists 'admin', 'config', 'local', and 'test'. The 'test' database is selected, and the 'collections' tab is active. The main area displays a table of collections with the following data:

Collection Name	Storage size	Documents	Avg. document size	Indexes	Total index size
kardesoks	36,160 KB	80	549,301 B	1	36,160 KB
rakordoks	20,48 B	8	192,00 B	1	36,864 B
users	32,77 B	1	3,76 B	2	73,73 B

45. ábra: Az adatbázis táblái

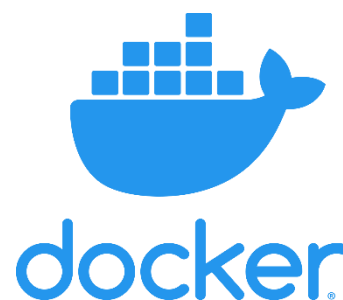
[illegible]

46. ábra: Szűrt kérdések az adatbázisban

Docker

Mi az a Docker?

A Docker egy platform és eszközkészlete, amellyel a fejlesztők, rendszer adminisztrátorok könnyedén tudják telepíteni, kihelyezni (deploy) az alkalmazásaikat egy homokozóba (sandbox, mint konténer), hogy ott futtassák őket a gazda gép operációs rendszerén, például egy Linux-on. A kulcsfontosságú nyereség a Docker használatával az az, hogy ez engedi a felhasználóknak, hogy becsomagoljanak egy alkalmazást annak együtt egy szabványosított egységbe. Nem úgy, mint a virtuális



45. ábra: A Docker logója [13]

igényelnek túlzottan sok erőforrást, így még hatékonyabban tudja felhasználni az „alatta lévő” rendszer erőforrásait

Mik azok a konténerek?

Összehasonlítva őket a virtuális gépekkel, amelyeknél a gazda gép operációs rendszere és a gazda gép erőforrásait használja a „vendég” virtualizált operációs rendszer, amely nagyon sok erőforrást igényel az ő saját működéséhez. Ezzel szemben a konténerek más megközelítést alkalmaznak: a konténerek a gazda operációs rendszer alacsony szintű mechanikáját kihasználva a virtuális gépek elszigeteltségének nagy részét biztosítják, a számítási igény és teljesítmény töredéke mellett.

Miért használunk konténereket?

A konténerek olyan logikai csomagolási mechanizmust kínálnak, amelyben az alkalmazások elvonakoztathatók attól a környezettől, amelyben ténylegesen futnak. Ez a szétválasztás lehetővé teszi, hogy a konténer alapú alkalmazások könnyen és következetesen telepíthetők legyenek, függetlenül attól, hogy a célkörnyezet egy privát adatközpont, a nyilvános felhő vagy akár egy fejlesztő személyes laptopja. A fejlesztők így kiszámítható (értsd: mindig mindenhol nagyjából ugyanannyi erőforrást használó) környezeteket hozhatnak létre, amelyek elszigeteltek a többi alkalmazástól, és bárhol futtathatók.

Üzemeltetési szempontból a hordozhatóság mellett a konténerek az erőforrások pontosabb ellenőrzését is lehetővé teszik, így az infrastruktúra hatékonyabbá válik, ami a számítási erőforrások jobb kihasználását eredményezheti.

Ezen előnyök miatt a konténerek (és a Docker) széles körben elterjedtek. Az olyan vállalatok, mint a Google, a Facebook, a Netflix és a Salesforce a konténereket a nagy mérnöki csapatok produktívabbá tételére és a számítási erőforrások jobb kihasználására használják. [14]

FORRÁSKÓD

A szakdolgozat alapját képező alkalmazás forráskódja, valamint jelen szakdolgozat a csapat közös GitHub repositoryjában található, a következő elérhetőség alatt:

<https://github.com/CLorant/quizter>

FORRÁSJEGYZÉK

[1] Vue logó

Letöltés dátuma: 2023. 04. 19.

https://upload.wikimedia.org/wikipedia/commons/thumb/9/95/Vue.js_Logo_2.svg/1024px-Vue.js_Logo_2.svg.png

[2] Introduction | Vue.js

Letöltés dátuma: 2023. 04. 20.

<https://vuejs.org/guide/introduction.html>

[3] HTML5 logó

Letöltés dátuma: 2023. 04. 19.

https://en.wikipedia.org/wiki/File:HTML5_logo_and_wordmark.svg

[4] CSS logó

Letöltés dátuma: 2023. 04. 19.

https://en.wikipedia.org/wiki/File:CSS3_logo_and_wordmark.svg

[5] Javascript logó

Letöltés dátuma: 2023. 04. 19.

https://upload.wikimedia.org/wikipedia/commons/thumb/3/3b/Javascript_Logo.png/640px-Javascript_Logo.png

[6] Pinia logó

Letöltés dátuma: 2023. 04. 19.

<https://en.wikipedia.org/wiki/Pinia#/media/File:Pinialogo.svg>

[7] Introduction | Pinia

Letöltés dátuma: 2023. 04. 20.

<https://pinia.vuejs.org/introduction.html>

[8] Node.js logó

Letöltés dátuma: 2023. 04. 21.

https://upload.wikimedia.org/wikipedia/commons/thumb/d/d9/Node.js_logo.svg/640px-Node.js_logo.svg.png

[9] Node.js

Letöltés dátuma: 2023. 04. 20.

<http://www.inf.u-szeged.hu/~tarib/javascript/nodejs.html#irodalomjegyzek>

[10] Megkezdjük az Express.js használatát

Letöltés dátuma: 2023. 04. 20.

<https://hu.admininfo.info/empezando-trabajar-con-express>

[11] MongoDB logó

Letöltés dátuma: 2023. 04. 21.

https://upload.wikimedia.org/wikipedia/commons/thumb/9/93/MongoDB_Logo.svg/640px-MongoDB_Logo.svg.png

[12] No-SQL – MongoDB adatbáziskezelő

Letöltés dátuma: 2023. 04. 20.

<https://masterfield.hu/hu/tanfolyamok/nosql-mongodb-adatbaziskezelo-tanfolyam>

[13] Docker logó

Letöltés dátuma: 2023. 04. 29.

<https://www.docker.com/wp-content/uploads/2022/03/vertical-logo-monochromatic.png>

[14] Docker - 1. rész: Alapok, telepítés, első használatba vétel

Letöltés dátuma: 2023. 04. 20.

<https://attila.gludovatz.hu/posts/docker-1-resz-alapok-telepites-első-hasznalatba-vétel>