# *címer szakgim kicsi (1)*

4032 Debrecen, Jerikó u. 17.

OM azonosító: 203033

*🕿:52/ 503-150 Fax: 52/314-204*

[*http://www.dszcberegszaszi.hu*](http://www.dszcberegszaszi.hu)

*E-mail:*

[*titkarsag@dszcberegszaszi.hu*](mailto:titkarsag@dszcberegszaszi.hu)

**Szakképesítés megnevezése: Szoftverfejlesztő  
OKJ száma: 54 213 05**

**ZÁRÓDOLGOZAT**

**QuizTime**

**Készítette**: **Konzulens:**  
Éles Zoltán Beke Béla

**Debrecen, 2023**

# Tartalomjegyzék

[Tartalomjegyzék 2](#_Toc133614483)

[1 Bevezetés 4](#_Toc133614484)

[1.1 Köszönetnyilvánítás 4](#_Toc133614485)

[1.2 Témaválasztás indoklása 4](#_Toc133614486)

[1.3 Elméleti és gyakorlati jelentőség 4](#_Toc133614487)

[2 Felhasználói dokumentáció 5](#_Toc133614488)

[2.1 Telepítés és indítás 5](#_Toc133614489)

[2.2 Hardware és Software igény 5](#_Toc133614490)

[2.3 A webes felület célja 5](#_Toc133614491)

[2.4 Lehetőségek a weboldalon 5](#_Toc133614492)

[2.4.1 Regisztráció és bejelentkezés 6](#_Toc133614493)

[2.4.2 A nav báron található dolgok. 7](#_Toc133614494)

[2.4.3 Kvíz témaválasztása 11](#_Toc133614495)

[2.4.4 Kvíz nehézség kiválasztása 11](#_Toc133614496)

[2.4.5 Kvíz elindítása 12](#_Toc133614497)

[2.4.6 Az eredmény megnézése 13](#_Toc133614498)

[2.5 A weboldal dizájnja 14](#_Toc133614499)

[3 Fejlesztői dokumentáció 15](#_Toc133614500)

[3.1 Fejlesztői környezet 15](#_Toc133614501)

[3.2 Adatszerkezet, típusok, változók 15](#_Toc133614502)

[3.2.1 A weblapon használt adatszerkezetek 15](#_Toc133614503)

[3.2.2 Az adatbázis különböző táblái és azok mezői 15](#_Toc133614504)

[3.2.3 Felhasználó tábla 16](#_Toc133614505)

[3.2.4 Kérdések tábla 16](#_Toc133614506)

[3.2.5 Választás tábla 17](#_Toc133614507)

[3.2.6 Téma tábla 17](#_Toc133614508)

[3.2.7 Nehézség tábla 17](#_Toc133614509)

[3.2.8 Eredmény tábla 18](#_Toc133614510)

[3.3 A weblapon használt változók 18](#_Toc133614511)

[3.4 Algoritmusok az oldalon 19](#_Toc133614512)

[3.4.1 A regisztráció algoritmusa 19](#_Toc133614513)

[3.4.2 A beléptetés algoritmusa 19](#_Toc133614514)

[3.4.3 Képfeltöltés 20](#_Toc133614515)

[3.4.4 Új téma hozzáadásának algoritmusa 20](#_Toc133614516)

[3.4.5 Főoldalon megjelenő kvízek algoritmusa 22](#_Toc133614517)

[3.4.6 Kvíz algoritmusa 23](#_Toc133614518)

[3.5 Tesztdokumentáció 24](#_Toc133614519)

[3.6 Fejlesztési lehetőségek 24](#_Toc133614520)

[3.7 Fejlesztéshez ajánlott 24](#_Toc133614521)

[3.8 Hibaüzenetek az oldalon 25](#_Toc133614522)

[4 Összefoglalás 26](#_Toc133614523)

[4.1 Szakdolgozat célja 26](#_Toc133614524)

[5 Irodalomjegyzék 27](#_Toc133614525)

[5.1 Források 27](#_Toc133614526)

# Bevezetés

## Köszönetnyilvánítás

Szeretném megköszönni Beke Béla tanárúrnak, aki segített a záródolgozat elkészítésében, nélküle ez a záródolgozat nem jöhetett volna létre.

## Témaválasztás indoklása

A webfejlesztés közelebb áll hozzám, mint a komplex programozás így adta magát, hogy milyen témájú legyen a weboldal. A felület lehetőséget biztosít a felhasználók számára különböző kvízek kipróbálására, illetve újakat is hozzáadni a meglévő témákhoz.

## Elméleti és gyakorlati jelentőség

Lényegében az oldal annyi jelentőséggel bír, hogy megtudhatjuk hogyan lehet regisztrációs, illetve bejelentkezős felületet készíteni és hogyan is jön létre a session bejelentkezéskor. Hogyan kapja meg az adatbázis az eredményeket, amit a felhasználó elér a kvízből, illetve a weblap hogyan kapja meg az adatbázistól a megfelelő adatokat a kvíz eléréséhez. A webes oldalak admin felületének felépítésébe is beleláthatunk, persze alap szinten, valamint az oldal hátterében álló adatbázis felépítésébe is.

# Felhasználói dokumentáció

## Telepítés és indítás

A szoftver webes alapú ezért szükség van a feltelepített böngészők egyikére, illetve internethozzáférésre. A fájlt le kell menteni a „htdocs” nevű mappába, amit a xampp telepítése után hoz létre a program. Alapértelmezett telepítési helye: (C:\xampp\htdocs). Ha a fájlt átmásoltuk a megfelelő mappába, utána indítsuk el a xampp-ot majd azon belül az Apache, illetve MySQL-t a „Start” gomb megnyomásával. Az adatbázist fel kell másolni a localhostra. Ezt úgy tehetjük meg, hogy a MySQL-nél található Admin gombra rákattintunk, majd felül kiválasztjuk az "Importálás" lehetőséget, majd tallózzuk a quiztime.sql fájlt és indítás. Ez után nyissuk meg a böngészőt majd írjuk be az oldal címét.

http://localhost/quiztime/index.php

admin bejelentkezési adatok: [**admin@admin.hu**](mailto:admin@admin.hu) **| adminadmin**

## Hardware és Software igény

Mivel ez egy weboldal szükség van egy olyan eszközre, ami rendelkezik internettel. Illetve mivel az oldal szerverét localhoston futtatjuk szükséges egy szerver (Xampp). Valamint szükség van egy böngészőre a megnyitáshoz például: Google Chrome, Internet Explorer, Opera.

## A webes felület célja

A weboldal célja, hogy a felhasználó felmérje a tudását különböző témákban, ahol

* Nehézséget tud választani
* Meg tudja tekinteni az eddig elért legjobb eredményeket és a sajátját

## Lehetőségek a weboldalon

Az oldalon bejelentkezés nélkül használható dolgok (Vendég)

* Toplista megtekintése
* Kvíz elindítása minden témában, illetve nehézségen

Az oldalon bejelentkezéssel elérhető dolgok

* Előzmények megtekintése

Adminisztrátor joggal elérhető dolgok

* Új téma létrehozása
* Felhasználó adminisztrátori jogának megváltoztatása
* Felhasználó jelszavaknak megváltoztatása

### Regisztráció és bejelentkezés

A regisztráció az oldal tetején található „Regisztráció” gomb megnyomásával elérhető. Szükség van egy felhasználónévre, később a felhasználónévvel láthat minket a többi felhasználó. Szüksége van egy Email cím-re, illetve jelszóra, amivel be tudunk jelentkezni a funkciók eléréséhez.

A jelszónak minimum 6 karakterből kell állnia. Amennyiben nincs meg a 6 karakter az oldal hibát fog adni. Ha megadtunk minden adatot és még nem foglalt egyik adat sem, akkor a Regisztráció gomb megnyomásakor át dob a bejelentkező felületre.

A képen Webhely látható

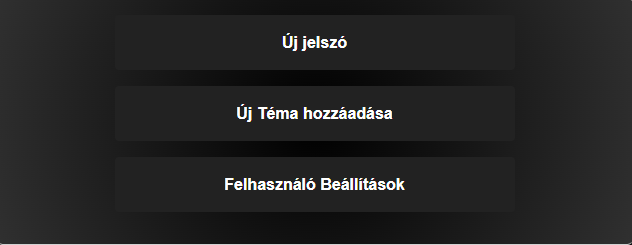
Automatikusan generált leírásBejelentkezés. Itt a korábban regisztrált adatokat kell megadni (Email cím és jelszó) a bejelentkezéshez majd, ha az adatok megfelelőek akkor a bejelentkezés gombra kattintva sikeresen bejelentkeztünk. Hibás adat megadása esetén, az oldal értesítést küld róla.

### A nav báron található dolgok.

A képen szöveg látható

Automatikusan generált leírás Az admin panel gomb csak bejelentkezés után és admin jogosultsággal rendelkező felhasználóknak látható, illetve megjeleníthető.

A gombra kattintva, további lehetőségünk van az oldalon. Amennyiben nem rendelkezünk adminisztrátor jogosultsággal, esetleg nem vagyunk bejelentkezve, az oldal meglátogatása után automatikus vissza kerülünk a főoldalra.



Az admin panelban jelenleg 3 funkció érhető el, Felhasználó fiókok jelszavaknak megváltoztatása, új téma hozzáadása, illetve felhasználói fiókok jogosultságainak megváltoztatására.

* Új jelszó

A képen szöveg látható

Automatikusan generált leírás

Itt nincs más dolgunk, mint beírni a szövegdobozba az új jelszót majd lementeni. Ugyanúgy, mint regisztrációnál minimum 6 karakterből kell álljon az új jelszó. Első oszlopban a felhasználó ID-ját láthatjuk, második oszlopban a felhasználónevét, utána az email címét, és az aktuális jogosultságot.

* Új téma hozzáadása

A képen diagram látható

Automatikusan generált leírás

Itt megtudjuk adni az új témának a nevét, a boritóját a kérdéseket, illetve a hozzá tartozó válaszokat, emellett természetesen a jó választ is ki tudjuk jelölni.

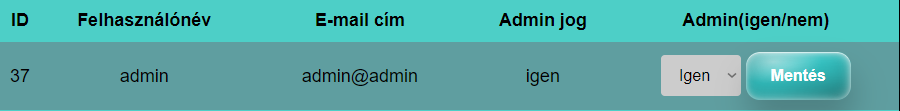
Ezután nincs más dolgunk, mint az oldal alján található hozzáadás gombra rá kattintani. Minden mező kitöltése kötelező, ha esetlegesen valami kimarad, az oldal visszajelzést küld erről. Minden kérdésnél csak egy válasz jelölhető meg. Az újonnan kreált téma a többihez hasonlóan rendelkezni fog 3 nehézségi fokozattal.

Kvíz lementése után az oldal vissza jelzést küld, ha sikeresen le mentettük a kvízt

A képen embléma látható

Automatikusan generált leírásHa sikeresen létrejött az kvízt, nincs más dolgunk, mint visszamenni a főoldalra és elindítani.

* Felhasználó beállítások:



Itt tudjuk a felhasználók jogosultságát beállítani. Szimplán ki kell választani, hogy adminisztráció joggal rendelkezzen-e az adott felhasználó, majd a mentés gombra kattintva sikeresen cseréltük a felhasználó jogosultságát.

A képen szöveg látható

Automatikusan generált leírás Az előzmények gombra kattintva megtekinthetjük az eddig megoldott kvízeink eredményét, dátumát, mennyi idő alatt oldottuk meg illetve, hogy milyen nehézségen, illetve témában csináltuk meg.

A képen asztal látható

Automatikusan generált leírás

Mindig a legutolsó eredmény lesz a lap tetején. Természetesen itt csak az adott felhasználó a saját játék előzményeit tudja megtekinteni.

A képen szöveg látható

Automatikusan generált leírásA toplista gombra kattintva megtekinthetjük a legjobb játékosok eredményeit, az adott témákban, illetve nehézségeken. Ezenkívül láthatjuk, hogy a felhasználók mikor és milyen teljesítménnyel vitték véghez a különböző kvízeket.

Itt elsősorban a játékosok pontszámait veszi figyelembe az oldal, egyező pontszám esetén a megoldási idő dönt.

A képen asztal látható

Automatikusan generált leírás

Illetve található egy szűrő funkció a toplista oldalon, amivel leszűrhetjük, hogy milyen témában, illetve milyen nehézségen szeretnénk látni a legjobb 10 játékost.

Az újonnan hozzáadott témák ugyan úgy megjelennek a szűrőben.



Végül a kijelentkezés gombbal kijelentkezhetünk a felhasználóból. Kijelentkezés után az oldal visszavisz minket a főoldalra.

### Kvíz témaválasztása

A képen Webhely látható

Automatikusan generált leírás

A témaválasztáshoz nincs más dolgunk, mint a kezdőlapon található tárgyak közül az egyiket kiválasztani.

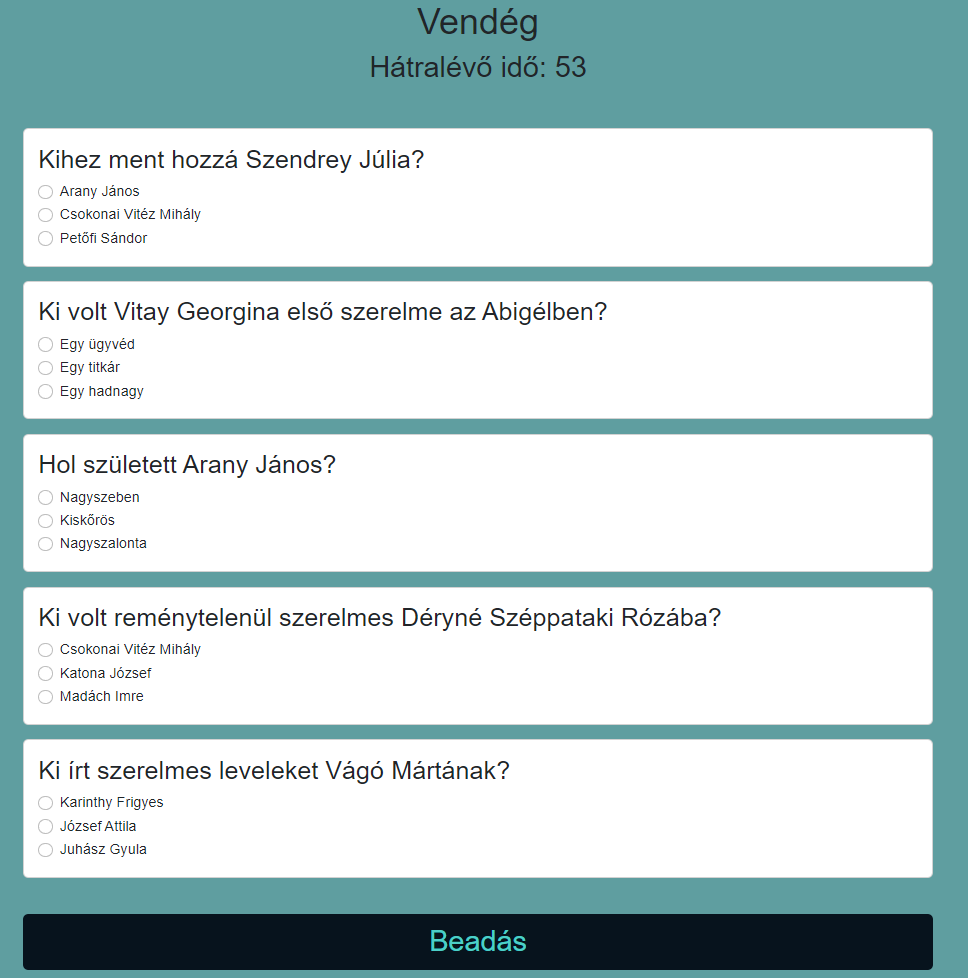
### Kvíz nehézség kiválasztása

A nehézség kiválasztásához szimplán rá kell vinni a kurzort a témára, amiben a kvízt szeretnénk megoldani majd a 3 megjelenített gomb közül ki kell választani a nekünk megfelelőt és rá kattintani. 3 nehézség található

* Könnyű (5 kérdésből álló kvíz)
* Közepes (10 kérdésből álló kvíz)
* Nehéz (15 kérdésből álló kvíz)

Mind a 3 nehézségre 60 másodperc áll rendelkezésre a kvíz megoldásához, ha ez letelik akkor az addig megoldott adatokat kapjuk eredményként.

### Kvíz elindítása



A kvíz elindul amint kiválasztottuk a megfelelő nehézséget.

Az idő, a kvíz megjelenése után automatikus elindul. A kérdésre a választ kiválasztani úgy tudjuk, hogy a válaszok mellett megtalálható fehér gombra rá kattintunk, vagy magára a válasz szövegére. Minden kérdésnél 1 jó válasz van. Ha minden kérdést megválaszoltunk akkor nincs más teendőnk, mint a beadás gomra rá kattintani, és átdob minket az eredményt megjelenítő oldalra, ahol megtekinthetjük az eredményünket.

Az oldal tetején az aktuális felhasználó neve jelenik meg, amivel le menti a kvízt az eredményekbe, ha nincs aktuális felhasználó akkor „Vendég” a képen látható módon.

### Az eredmény megnézése

Amint beadtuk a megoldott kvízünket, esetleg lejárt az idő automatikusan át dob minket egy másik oldalra, ahol megtekinthetjük a kvízen elért eredményünket.

A képen szöveg látható

Automatikusan generált leírás

Ki írja, hogy melyik témában, milyen nehézségen, mennyi idő alatt, hány pontosra, illetve, hogy mikor csináltuk meg a kvízt.

Az itt található adatokat az oldal automatikusan feljegyzi az előzmények oldalon, illetve, ha az adott eredményekkel a legjobb eredmények közé kerülünk akkor a toplistában is megtalálhatjuk magunkat.

Mindemellett található egy „Vissza” gomb, amivel az oldal át irányít minket a kezdőlapra.

## A weboldal dizájnja

Korábban néhány felületet már említettem, itt most egészében megmutatom, hogy több témával hogyan is néz ki az oldal. Új téma hozzáadása esetén, a meglévőkhöz hasonlóan jelenik meg.



Az oldalon gyakran alkalmazom a hover hatást, amely lehetővé teszi, hogy az egérmutató fölé helyezésekor különböző vizuális elemek jelenjenek meg vagy változzanak meg a felhasználói élmény fokozása érdekében.

Az oldalhoz tartozó CSS az saját, azonban használok Bootstrap keretrendszert az egyszerűbb fejlesztés érdekében.

# Fejlesztői dokumentáció

## Fejlesztői környezet

Mivel weboldal fejlesztést választottam így az alap eszközöket használtam. Az eszközök a következőek: Visual Studio Code a kódoláshoz, Xampp a lokális környezet kialakításához.

## Adatszerkezet, típusok, változók

### A weblapon használt adatszerkezetek

Mint minden oldalon, ahol található regisztráció úgy itt is használ az oldal integer-t, illetve string típusú adatokat. A következő pontban látható az adatbázis ábrája, ami a weboldal adattárolásáért felel benne a következő adattípusokkal: integer, varchar, date, time.

### Az adatbázis különböző táblái és azok mezői

A képen szöveg látható

Automatikusan generált leírás

Az oldalon beérkező adatok tárolására a quiztime adatbázis nyújt lehetőséget. Az adatok külön feldolgozó programokon jutnak el ide.

### Felhasználó tábla

A képen szöveg látható

Automatikusan generált leírásElsősorban a felhasználók alap adatainak a tárolására szolgál. Első értéke az „ID” szám integer típusú és önmagától generálódik amikor, egy új felhasználó regisztrál az oldalon. Ez azért fontos mert a későbbiekben nagy szerepe lesz a beléptetésnél, ez a felhasználó azonosítója, illetve megkülönbözteti a rekordokat egymástól.

A „felhasznalo” a felhasználó nevek tárolására szolgál rövid szöveg típusban (varchar), például a toplistán szereplő név ez lesz. A jelszó a bejelentkezéshez szükséges, itt vannak tárolva a felhasználók jelszavai, ez is szintén rövid szöveg típusban tárol (varchar). Az „email” mező az email címeket tartalmazza, az oldalra való belépéshez nélkülözhetetlen.

Az „admin” a felhasználók státuszát tárolja, ez egy (integer típusú szám) ez az adminisztrátorok és felhasználók megkülönböztetésére szolgál az oldalon, valamint az adminisztrátorok ennek segítségével képesek belépni az adminisztrátor felületre

### Kérdések tábla

A képen szöveg látható

Automatikusan generált leírásElsősorban a kérdések tárolására szolgál. Első értéke az ID, integer típusú szám, ami külön veszi a kérdéseket egymástól, így minden kérdéshez egy külön ID kapcsolódik. Második eleme a „tema\_id”, ami azért kell, hogy az adott kérdés melyik témához kapcsolódik, ez szintén egy integer típusú szám. Illetve található egy „szoveg”, ami magát a kérdés szövegét tárolja rövid szöveg típusban (varchar).

### A képen szöveg látható Automatikusan generált leírásVálasztás tábla

Elsősorban a válaszok tárolására szolgál. Első értéke az ID, integer típusú szám, ami külön választja a válaszokat egymástól, minden válasz egy külön ID-val van ellátva. Második eleme a „kerdes\_id”, ami szintén egy integer típusú szám, ami a válaszokat a megadott kérdéshez kapcsolja, így a válaszok hozzákapcsolódnak a kérdéshez.

Található egy „szoveg”, ami a válaszok szövegéért felel, ami a kiíratásához kell, ez egy rövid típusú szöveg (varchar). A „jovalasz” integer típusú szám, ami ellenőrzi, hogy a megadott válasz helyes-e.

### A képen szöveg látható Automatikusan generált leírásTéma tábla

A különböző témák tárolására szolgál. Az első értéke egy integer típusú szám az ID. Minden téma egy külön ID-val rendelkezik, amivel akármikor el lehet érni az adott témákat. Illetve található benne egy „nev” rövid szöveg típusú (varchar) ami a különböző témák szövegét tárolja, például az előzményeknél megtalálható, hogy melyik témát csináltuk utoljára. Mindemellett található egy „kep\_url” szöveg típusú (text) ami a képek lementéséhez szükséges.

### A képen szöveg látható Automatikusan generált leírásNehézség tábla

A nehézségi szintek tárolására szolgál. Az első értéke egy integer típusú szám az „ID”. Minden nehézségi szint egy különböző ID-t kapott, amivel a szinteket el lehet különíteni egymástól. Található benne egy „nev” rövid szöveg típusú (varchar) ami a különböző nehézségi fokozatok szövegét tárolja, például a kezdőoldalon a kvíz elindításnál, ahol ki lehet választani, hogy milyen nehézségen szeretnénk megcsinálni a kvízt.

### Eredmény tábla

A képen szöveg látható

Automatikusan generált leírásEz a tábla az eredmények tárolására szolgál, amit a felhasználók érnek el az adott kvízeken. Elsősorban található egy „ID”, ami egy integer típusú szám, ez arra van, hogy minden sornak legyen egy egyedi ID-ja így meglehessen különböztetni a sorokat egymástól. A következő az a „felhasznalo\_id” ezzel tudjuk a felhasználók eredményeit elkülöníteni egymástól, így például az előzményekben csak az adott, éppen bejelentkezett felhasználó eredményeit lehet látni, ez szintén egy integer típusú szám. A következő az a „tema\_id”, ami szintén egy integer típusú szám, ezzel a kvíz befejezése után az eredménybe a témát mentjük le, hogy melyiket csinálta a felhasználó. A következő a „pontszam”, ami szintén egy integer típusú szám, ezzel a felhasználó által elért pontszámot tároljuk. A következő a „mikor”, ami egy date adattípus, ami a dátumot tárolja, hogy mikor csinálta a kvízt a felhasználó. A következő az „ido”, ami egy time adattípus, ami az időt tárolja. Ebbe van tárolva, hogy a felhasználó mennyi idő alatt oldotta meg a kvízt. Illetve található benne egy „nehezseg\_id”, ami egy integer típusú szám, ebbe van tárolva, hogy a felhasználó milyen nehézségen oldotta meg a kvízt.

## A weblapon használt változók

A weblapon használt $\_SESSION[’felhasznalo’]: Az aktuális felhasználó nevét tartalmazza, amennyiben be van jelentkezve. $\_SESSION[’felhasznalo\_id’]: Az éppen bejelentkezett felhasználó azonosítóját tartalmazza. Ez az érték kerül átadásra az adatbáziskérés paramétereként. Ez a változó az oldalra való belépéstől az oldalról való kijelentkezésig él. Minden egyes regisztrációnál generálódik az adatbázisban a felhasznalo táblán belül egy id mezőben letárolt érték, ami a felhasználó azonosítója lesz.

Az oldalon továbbá $\_GET[] illetve $\_POST[] változókat is használunk. A $\_GET[] változókat az oldal paraméterezésére, míg a $\_POST[] változókat a feldolgozó programokhoz való adatok eljuttatására használjuk fel.

## Algoritmusok az oldalon

A következő algoritmusok a program elkészítésének sorrendjében lettek leírva itt a szakdolgozatban.

### A regisztráció algoritmusa

A célja, hogy a felhasználónak legyen lehetősége be regisztrálni az oldalra, amivel több funkciót tud elérni, majd ezáltal be tudjon jelentkezni.

Lépései sorrendben:

* A felhasználó ki töltött-e minden mezőt (felhasználónév, e-mail cím, jelszó) ellenőrzése, hogy nem maradt-e üres mező, ha igen akkor hibaüzenet küldése.
* A beírt e-mail cím leellenőrzése, hogy valódi-e.
* A jelszó hosszúságának ellenőrzése strlen függvénnyel, hogy megvan-e 6 karakter hosszú, ha nincs akkor hibaüzenet küldése.
* A használni kívánt email cím leellenőrzése egy „if” függvénnyel, hogy nincs-e még használatban az e-mail cím. Ha használatban van, hibaüzenet küldése.
* Abban az esetben, ha kitöltötte a regisztrációs mezőket, valamint az e-mail cím még nincs használatban, akkor jelszó titkosítása bcryptel, majd adatbázisba való feltöltés, vissza igazolás az oldalon és tovább dob a bejelentkezés oldalra.

### A beléptetés algoritmusa

Az algoritmus célja, hogy a felhasználó biztonságosan be tudjon lépni az oldalra a regisztrációt követően és ezt követően a saját nevében tudjon játszani a kvízzel.

Lépései:

* A fent említett $\_POST[]-os változókból a $\_POST[email] összehasonlítása az adatbázisban, ehhez a fiókhoz tartozó email mezővel. (Felhasználó e-mailjének ellenőrzése)
* Majd a $\_POST[jelszo] bcrypt függvénnyel való átalakítása után összehasonlítás jelszó mező tartalmával. (A felhasználó jelszavaknak ellenőrzése)
* Eltérés vagy üresen hagyott mező esetén hibaüzenet küldése.
* Amennyiben az ellenőrzés sikeres felhasználó beléptetése.
* Mindezt követően főoldalra ugrás beléptetett módban.

### Képfeltöltés

Kép feltöltésre, a téma hozzáadásánál van lehetőség, hogy a főoldalon látható témának legyen háttere. Kép kiválasztása után, első lépésben ellenőrizzük, hogy a képfájl felöltése megtörtént-e. Ha a feltöltés sikeres volt, akkor elmentjük a fájl nevét és átmeneti nevét a "$filename és $tempname változókba. Ezután megadunk egy útvonalat a kép elmentésére. Következő lépésben a fájl másolását a "move\_uploaded\_file() fügvénnyel átmásoljuk a kiválasztott helyre. Ha a fájl másolása sikeres volt, akkor a $kep\_url változóban eltároljuk az útvonalat, kivágjuk az útvonal elejéről a "../" részt, majd visszatérünk a kép mentésének folyamatához. Ha a kép mentése nem sikerült, akkor a $kep\_url változóban üres string lesz tárolva, és így nem kerül elmentésre a kép. Ezen kívül a fájl mappájában lévő „images” fájlba is lementi a képet.

    public function save\_quiz($quiz\_data)

    {

        $tema\_nev = $quiz\_data["nev"];

        if (isset($\_FILES["kep\_url"])) {

            $filename = $\_FILES["kep\_url"]["name"];

            $tempname = $\_FILES["kep\_url"]["tmp\_name"];

            $kep\_url = "../kozos/images/" . $filename;

            if (!move\_uploaded\_file($tempname, $kep\_url)) {

                $kep\_url = "";

            } else {

                $kep\_url = ltrim($kep\_url, "../");}

        } else {

            $kep\_url = "";

        }

    }

### Új téma hozzáadásának algoritmusa

Az oldal lehetővé teszi egy új kvíz témájának hozzáadását, a kérdésekkel és válaszokkal együtt. Az űrlap a kérdések maximális számát és a válaszok maximális számát állítja be, majd a kérdéseket és válaszokat ciklusok segítségével generálja. Az adatokat az űrlapról POST módszerrel kapjuk meg, majd a kvíz objektumok segítségével elmentjük azokat az adatbázisban. Az adatbázis kapcsolatot az adatbazis.php fájl tartalmazza. A quiz osztályban található metódusok a kvíz adatok elmentésére szolgálnak az adatbázisban. A get\_tema metódus segítségével lekérdezzük az összes kvíz témát az adatbázisból. A save\_tema metódus az új kvíz témáját menti az adatbázisba, majd visszaadja annak az azonosítóját. A save\_kerdesek metódus elmenti a kvíz kérdéseit és visszaadja azok azonosítóját. A save\_valasztas metódus az adatbázisba menti a kérdésekre adott válaszokat és visszaadja azok azonosítóját.

 public function save\_tema($nev, $kep\_url)

    {

        $sql = "INSERT INTO tema VALUES ('null', ?, ?);";

        $result = $this->db->RunSQL($sql, $nev, $kep\_url);

        $tema\_id = $this->db->get\_last\_insert();

        return $tema\_id;

    }

    public function save\_kerdesek($tema\_id, $kerdesek)

    {

        $sql = "INSERT INTO kerdesek VALUES ('null', ?,?);";

        $result = $this->db->RunSQL($sql, $tema\_id, $kerdesek);

        $kerdesek\_id = $this->db->get\_last\_insert();

        return $kerdesek\_id;

    }

    public function save\_valasztas($kerdes\_id, $szoveg, $jovalasz)

    {

        $sql = "INSERT INTO valasztas VALUES ('null', ?,?,?);";

        $result = $this->db->RunSQL($sql, $kerdes\_id, $szoveg, $jovalasz);

        $valasztas\_id = $this->db->get\_last\_insert();

        return $valasztas\_id;

    }

### Főoldalon megjelenő kvízek algoritmusa

Az adatok lekérdezéséhez az oldal használ egy "adatbazis.php" fájlban definiált "Database" osztályt. Az adatok lekérdezése SQL lekérdezések segítségével történik, amelyek a tantárgyakat és nehézségi szinteket tartalmazó "tema" és "nehezseg" táblákból olvassák ki az adatokat. A tantárgyak és nehézségi szintek iterálása a foreach ciklussal történik, majd minden tantárgyhoz és nehézségi szinthez gombot rendelnek hozzá, amelyekre kattintva a felhasználó eljuthat a kvíz oldalra a kiválasztott tantárgy és nehézségi szint alapján.

$tantargy = $db->FetchData("SELECT \* FROM tema");

        $nehez = $db->FetchData("SELECT \* FROM nehezseg");

        foreach ($tantargy as $targy): ?>

            <?php

            if (!is\_null($targy['kep\_url']) && $targy['kep\_url'] != "" && file\_exists($targy['kep\_url'])) {

                $bg\_kep = $targy['kep\_url'];

            } else {

                $bg\_kep = 'kozos/' . $targy['nev'] . '.jpg';

            }

            ?>

            <div class="card" style="background-image: url('<?= $bg\_kep ?>');">

                <div class="card\_\_inner">

                    <?php foreach ($nehez as $i => $asd): ?>

                        <a href="quiz.php?targy=<?= $targy['nev'] ?>&nehez=<?= $asd['id'] ?>"><button class="button-31">

                                <?= $asd['nev'] ?>

                            </button></a>

                    <?php endforeach ?>

                </div>

            </div>

        <?php endforeach ?>

### Kvíz algoritmusa

Az első lépésben inicializálunk egy adatbázis objektumot és lekérjük a GET paraméterekből a kvíz témáját és nehézségét. Ezután elkészítjük az SQL lekérdezést a kérdések és válaszok lekérdezéséhez, ahol a témának megfelelően szűrünk és véletlenszerűen kiválasztunk $nehez darab kérdést, a nehézség szorzója alapján. Az SQL lekérdezéssel lekérjük a felhasználó által kiválasztott válaszokat és összehasonlítjuk azokat a helyes válasszal. Ha a válasz helyes, akkor növeljük a pontszámot. Az eredményeket pedig az eredmény táblában tároljuk el. Található egy számláló, amely megjeleníti a hátralévő időt a kvíz kitöltéséhez. A számláló JavaScript segítségével működik, amely a setInterval () függvénnyel a megadott időközönként csökkenti a számlálót, és ha az idő lejár, akkor az űrlapot beküldi.

foreach ($\_POST as $key => $value) {

        if (strpos($key, 'radio') === 0) {

            $kerdes\_id = str\_replace('radio ', '', $key);

            $sql = "SELECT jovalasz FROM valasztas WHERE id = $value";

            $result = $db->RunSQL($sql);

            if ($result->num\_rows > 0) {

                $row = $result->fetch\_assoc();

                if ($row['jovalasz'] == 1) {

                    $pontszam++;

                }

            }

        }

    }

var count = 60, timer = setInterval(function () {

            document.getElementById("counter").innerHTML = (count--) - 1;

            document.getElementById("countera").value = count;

            if (count == 0) {

                clearInterval(timer);

                document.getElementById("forma").submit();

            }

        }, 1000);

## Tesztdokumentáció

Az oldal le lett tesztelve több fajta böngészővel

* Google Chrome
* Internet Explorer
* Opera

Nagy különbség nem volt tapasztalható. Az oldal teszteléséhez közeli barátaimat kértem meg, segítség nélkül tudták használni az oldalt

## Fejlesztési lehetőségek

* Admin panel bővítése: Témák eltávolítása, kérdés/válaszok cserélése
* Profil beállítás: Profilkép beállítása, más emberek előzményei megtekintése.
* Kvíz megoldása után, az elrontott válaszok megnézése.

## Fejlesztéshez ajánlott

* **Git** egy elosztott verziókezelő rendszer, amelyet kódbázisok kezelésére, nyomon követésére és verziókezelésére használunk
* **Visual Studio Code** egy nyílt forráskódú fejlesztői környezet, amelyet kifejezetten webes és felhő alapú alkalmazások fejlesztéséhez terveztek. Támogatja több nyelv és keretrendszer használatát. Az alkalmazás sok hasznos funkciót kínál, például kódkiegészítést, hibakeresést, verziókezelést, kódbemutatót és sok más lehetőséget, amelyek javítják a fejlesztői hatékonyságot.
* **phpMyAdmin** egy ingyenes webes alkalmazás, amelyet a MySQL adatbázis kezelésére használnak. Áttekinthető, könnyen kezelhető adatbáziskezelő és szerkesztő alkalmazás.

## Hibaüzenetek az oldalon

Bejelentkezéskor, ha rossz e-mail címet vagy jelszót adunk meg: A képen szöveg látható

Automatikusan generált leírás

Regisztrációnál, ha üresen hagyunk egy mezőt:

A képen szöveg látható

Automatikusan generált leírás

Email címnél @ jel elhagyása esetén:



Amennyiben már regisztráltak a megadott email címmel:

A képen szöveg látható

Automatikusan generált leírás

# Összefoglalás

Nagyon élveztem az oldal elkészítésének minden percét, és úgy érzem, remek gyakorlási lehetőség volt. A projekt fejlesztése során kihívásokkal és nehézségekkel szembesültem, de ezeknek köszönhetően sikerült fejlesztenem kódolási készségeimet és új dolgokat tanulnom. Végül is elégedett vagyok a munkámmal, és büszke vagyok arra, hogy létrehoztam ezt az oldalt.

## Szakdolgozat célja

A szakdolgozatom célja egy webes felület létrehozása volt, amely lehetővé teszi a felhasználók számára, hogy felmérjék tudásukat különböző témákban és nehézségi szinteken.

Illetve, hogy egy olyan platformot biztosítson, ahol a felhasználók saját tudásszintjüket, illetve a többiekhez képest való teljesítményüket is felmérhetik. A felület használatával a felhasználók lehetőséget kaptak a saját tudásuk javítására is, hiszen az eredményeket megtekinthették és összehasonlíthatták másokéval. Az oldal több témakört és nehézségi szintet tartalmazott, így mindenki megtalálhatta a saját tudásszintjének megfelelő teszteket.

# Irodalomjegyzék

A szakdolgozathoz használt források:

* <https://getbootstrap.com>
* <https://www.w3schools.com>
* <https://www.php.net>

Könyvek:

* Robin Nixon: Learning PHP, MySQL & JavaScript
* Jon Duckett: PHP & MySQL - Server-side Web Development
* Jon Duckett: Web Design with HTML, CSS, JavaScript and jQuery Set

## Források

<https://www.freepik.com> Az oldal háttérképe

<https://www.tankonyvkatalogus.hu> A honlapon használt képek