**Eötvös Loránd Tudományegyetem**

**Informatikai Kar**

**Adattudományi és Adattechnológiai Tanszék**

Figyelemhiányos hiperaktivitási-zavar öndiagnosztikát segítő webes alkalmazás

Szerző: Témavezető:

Koklár Tamás Bencsik Gergely

Programtervező informatikus BSc. Egyetemi docens - PhD

**Budapest 2023**



Tartalomjegyzék

[1. Bevezetés 3](#_Toc136268364)

[1.1. Motiváció 3](#_Toc136268365)

[1.2. A dolgozat célja 4](#_Toc136268366)

[2. Felhasználói dokumentáció 5](#_Toc136268367)

[2.1. Célközönség 5](#_Toc136268368)

[2.2. Minimum gépigény 5](#_Toc136268369)

[2.3. Kötelező szoftverek 5](#_Toc136268370)

[2.4. Első futtatás előtti beállítások 5](#_Toc136268371)

[2.5. A program indítása 6](#_Toc136268372)

[2.6. Fejléc és lábléc 7](#_Toc136268373)

[2.7. Oldalak és aloldalak részletes leírása 8](#_Toc136268374)

[2.7.1. Kezdőlap 8](#_Toc136268375)

[2.7.2. Bejelentkezés 8](#_Toc136268376)

[2.7.3. Regisztráció 8](#_Toc136268377)

[2.7.4. Kérdések 9](#_Toc136268378)

[2.7.5. Részletező oldal 11](#_Toc136268379)

[2.7.6. Statisztika oldal 12](#_Toc136268380)

[3. Fejlesztői dokumentáció 13](#_Toc136268381)

[3.1. Fejlesztői környezet 13](#_Toc136268382)

[3.1.1. Verziókövetés 13](#_Toc136268383)

[3.1.2. Kódszerkesztő program 14](#_Toc136268384)

[3.1.3. Adatbázis 14](#_Toc136268385)

[3.1.4. Keretrendszer 15](#_Toc136268386)

[3.1.5. Használt szoftverek verziói 15](#_Toc136268387)

[3.2. MVC architektúra 16](#_Toc136268388)

[3.2.1. Modell bemutatása 17](#_Toc136268389)

[3.2.2. Nézet bemutatása 20](#_Toc136268390)

[3.2.3. Vezérlő bemutatása 21](#_Toc136268391)

[3.2.4. Adatok továbbítása 22](#_Toc136268392)

[3.2.5. Adatbázis használata 22](#_Toc136268393)

[3.2.6. Jogosultság és hibakezelés 23](#_Toc136268394)

[3.3. Adatbázis bemutatása 23](#_Toc136268395)

[3.3.1. Users tábla 23](#_Toc136268396)

[3.3.2. Categories tábla 24](#_Toc136268397)

[3.3.3. Questions tábla 24](#_Toc136268398)

[3.4. Felhasználói diagram 26](#_Toc136268399)

[3.5. Felhasználói történet 27](#_Toc136268400)

[3.6. Tesztelés 31](#_Toc136268401)

[3.6.1. Jogosultságkezelés 31](#_Toc136268402)

[3.6.2. Regisztráció, bejelentkezés 33](#_Toc136268403)

[3.6.3. Kérdéssor kitöltése 34](#_Toc136268404)

[3.6.4. Statisztika 35](#_Toc136268405)

[4. Összefoglalás és további fejlesztési lehetőségek 35](#_Toc136268406)

[4.1. Összefoglalás 35](#_Toc136268407)

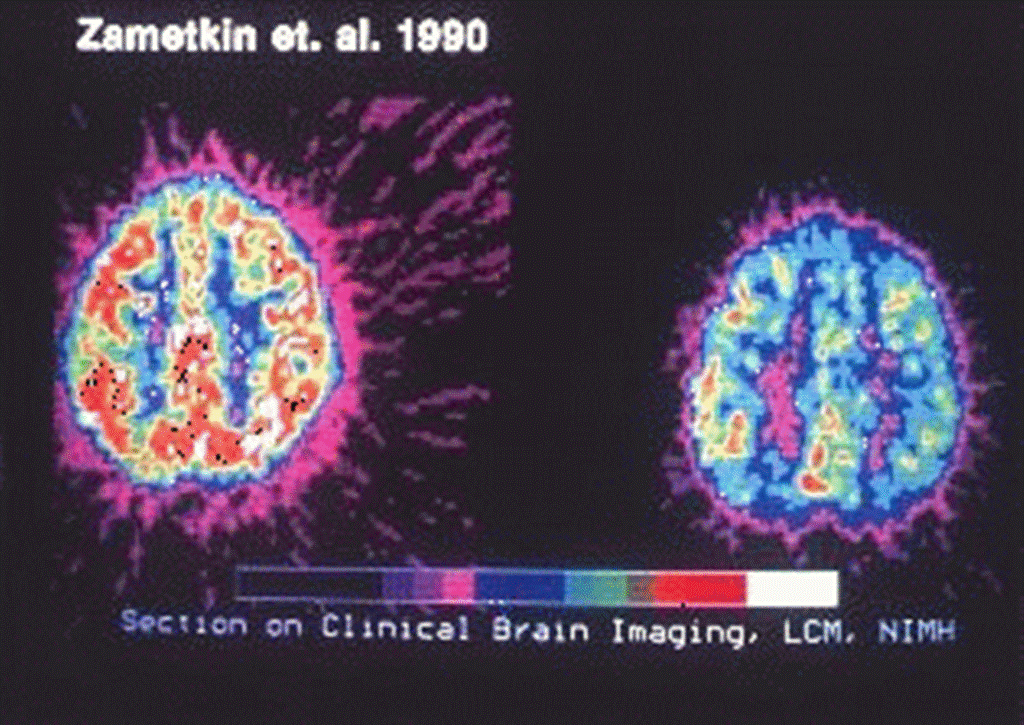
[4.2. További fejlesztési lehetőségek 36](#_Toc136268408)

[5. Irodalomjegyzék 37](#_Toc136268409)

# Bevezetés

## Motiváció

Az elmúlt időben egyre többet lehet hallani a mentálhigiénés témákról és különböző neurofejlődési rendellenességekről, mely tartalmazza többek közt az autizmus spektrum zavart és ADHD-t (figyelemhiányos hiperaktivitás zavar). Ezeknek a zavaroknak a felismerése és diagnosztikája általánosságban fiatalon megtörténik, viszont vannak, akiknek nem volt lehetőségük diagnózishoz jutniuk fiatal korukban és mai napig, már felnőve küzdhetnek meg ezeknek a zavaroknak a nehézségeivel. A felnőttek diagnosztizálása különösen nehéz tud lenni, tekintve, hogy azok a felnőttek, akiknek úgy kellett túlélniük már az óvodától kezdve egészen a munka világáig a mindennapjaikat, hogy nem voltak tisztába azzal, hogy nekik milyen nehézségeik vannak a mindennapi problémákon kívül, megtanultak alkalmazkodni a társadalom nagy részéhez. Napról napra, és percről percre aktívan tesznek azért, hogy ne tűnjön fel másoknak a mindennapi problémáikon kívüli saját nehézségeik. Ez az úgynevezett masking, maszkolás.



1.1. ábra - Bal oldalon: Egészséges személy agyi aktivitása.

Jobb oldalon: ADHD-ben élő személy agyi aktivitása. [3]

Erre a problémára először párom hívta fel a figyelmemet, aki valószínűleg ADHD-s viszont hivatalos diagnózishoz még nem jutott hozzá. Mivel napi rendszerességű problémákba ütközött, amikben én próbáltam segíteni neki, szépen lassan elkezdtünk utána járni, mi is lehet a nehézségeinek a kiváltó oka. Ekkor indult el egy öndiagnózis úton, ami alapján a legkézenfekvőbb választ az ADHD adta. Azonban itt nem állt le és próbált minél több hivatalos forrásból információt gyűjteni nem csak a nehézségei kiváltó okáról, hanem a felnőtt diagnózisról is. Ekkor szembesültünk azzal, hogy a várólista minimum másfél kétéves.

## A dolgozat célja

A szakdolgozatom célja, hogy segítse, azokat a fiatalokat vagy akár felnőtteket, akiknek nem volt lehetőségük még hivatalos diagnózist a kezükben tartani a saját nehézségeikkel kapcsolatban. Szeretném, hogy ez egy eszköz legyen arra, hogy az embereknek adjon egy kiindulási pontot annak érdekében, hogy legyen egy elképzelésük arról, milyen irányba induljanak el az öndiagnózis alatt és egyszerűbben lépjenek be egy ambulancia ajtaján, mikor a hivatalos diagnózisukra mennek.

# Felhasználói dokumentáció

## Célközönség

A fentiekben már említett fiatal felnőtt korosztálytól indulva egészen az idősebb korosztályig, azok a személyeket szeretném ezzel megcélozni, akik úgy gondolják, valamilyen plusz csomagot, nehézséget éreznek a mindennapi működésükben és nem tudják, hova forduljanak először. Sokaknak ijesztő és megterhelő lehet orvos elé állni, főleg, ha nem tudják milyen tünetekkel diagnosztizálnak valakit ADHD-val (mivel, a legtöbb tünet számukra mindennapi működés és teljesen normális). Ebben is szeretném segíteni ezeket a személyeket, hogy egy tiszta képpel tudjanak egy orvos elé állni, esetleg csak fejleszteni magukat jövőbeli diagnózis terve nélkül.

## Minimum gépigény

* CPU: 1 Ghz
* RAM memória: 1 GB
* Tárterület: 1 GB szabad hely a merevlemezen
* Operációs rendszer: Windows 7 vagy újabb

## Kötelező szoftverek

* PHP 8.0
* Composer 2.5.5 vagy újabb
* Node.js
* VSCode 18.16.0 vagy újabb
* DB Browser for SQLite 3.12.2 vagy újabb
* (opcionális) GitHub Desktop 3.2.3 vagy újabb

## Első futtatás előtti beállítások

A 2.3-as pontban említett szoftverek közül a PHP és Composer telepítéséhez létezik egy telepítő, ami le lehet tölteni [9] ennek futtatása során ki kell választani a 8.0-ás PHP-t. Amint befejeződött a telepítés az alábbi parancsokat egy terminálban kiadva vissza kell kapnunk a PHP és Composer verziószámait. Parancsok: „php -v” „composer -V”. Ez után újra kell indítani a számítógépünket, hogy a változtatások érvényesüljenek. Telepítenünk kell a legfrissebb verziójú Node.js-t [10] is. Node.js telepítését a „node -v” paranccsal ellenőrizhetjük le. A szakdolgozat quiz mappájában található egy „.env.example” nevű fájl. Ezt másoljuk le és nevezzük át a következőre: „.env”. Ezek után a quiz mappában nyissunk egy parancssort (CMD, Terminál) majd adjuk ki a „composer install” parancsot. Ez után a „php artisan key:generate” parancsot majd az „npm run build” parancsot. Amint ezek megvannak elkezdhetjük az adatbázis seedelését(kategóriák és test felhasználó létrehozása) ehhez a következő parancsot adjuk ki: „php artisan migrate --seed” itt a „Would you like to create it?” kérdésere a „yes” választ adjuk.

## A program indítása

Ha sikerült a 2.4-es pontban mindent végrehajtani akkor a „php artisan serve” paranccsal el is indíthatjuk a szerverünket.

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

2.5. ábra – php artisan serve parancs eredménye

A „Server running on” parancs utáni szögletes zárójelek között található linken fog futni a szerverünk. Ha ezt megnyitjuk egy erre alkalmas böngészőben akkor a következő oldal fogad minket:

A képen szöveg, képernyőkép látható

Automatikusan generált leírás

2.5. ábra - A weboldal kezdőlapja

## Fejléc és lábléc

Az alkalmazásban minden oldalon megtalálható egy fej- és lábléc. Rögzített pozícióban, tehát a fejléc csak az oldal tetején, a lábléc csak az alján található meg. A fejlécen található gombok változnak, ha a felhasználó be van jelentkezve, nincs bejelentkezve vagy be van jelentkezve és minden kérdésre válaszolt.

A fejléc tartalma esetenként:

* Ha a felhasználó nincsen bejelentkezve:
  + Kezdőlap gomb
  + Bejelentkezés gomb
  + Regisztráció gomb
* Ha a felhasználó be van jelentkezve:
  + Kezdőlap gomb
  + Kérdések gomb
  + Felhasználó név
    - Ide kattintva megjelenik a kijelentkezés gomb
* Ha a felhasználó be van jelentkezve és minden kérdésre válaszolt:
  + Kezdőlap gomb
  + Kérdések gomb
  + Statisztika gomb
  + Felhasználó név
    - Ide kattintva megjelenik a kijelentkezés gomb

A láblécen a szakdolgozat címe található, mellette a Laravel és a PHP verzió.

## Oldalak és aloldalak részletes leírása

### Kezdőlap

Ezt az oldalt látjuk először, amikor indul az alkalmazás vagy amikor a kezdőlap gombra kattintunk. Itt a témák felsorolását és az oldal rövid leírását találjuk.

### Bejelentkezés

Itt egy egyszerű bejelentkező oldalt találunk. A megfelelő adatok megadása után a bejelentkezés gombra kattintva megtörténik a bejelentkezés, az oldal átirányít a kérdések oldalra.

A képen szöveg, képernyőkép, szoftver, Weblap látható

Automatikusan generált leírás

. 2.7.2. ábra - Bejelentkezés

### Regisztráció

Itt egy egyszerű regisztrációs oldalt találunk. A megfelelő adatok megadása után a Regisztráció gombra kattintva megtörténik a regisztráció, az oldal átirányít a kérdések oldalra. A képen szöveg, képernyőkép, Téglalap, Párhuzamos látható

Automatikusan generált leírás

. 2.7.3. ábra – Regisztráció

### Kérdések

Ez az oldal a legtöbb információt megjelenítő oldal az egész alkalmazásban.

#### Témák kártyái

Minden témának van egy-egy kártyája a kérdések oldalon. Egy ilyen kártya tartalmaz egy képet (ez kategóriánként változó), a kategória nevét, a téma állapotát (megválaszolt vagy sem) és egy gombot ami a téma részletező oldalára visz ahol lehetőségünk van megválaszolni az adott témához tartozó kérdéseket.

A képen szöveg, clipart, Animációs film, képernyőkép látható

Automatikusan generált leírás

. 2.7.4.1. ábra - Téma kártya

#### Kategóriák kártya

Itt kerül megjelenítésre az összes adatbázisban eltárolt kategória.

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

. 2.7.4.2. ábra - Kategóriák kártya

#### Általános statisztika kártya

Itt található az általános statisztika, amiben megjelenik: regisztrált felhasználók száma, az összes kategória száma, az összes téma száma, az összes kitöltött téma száma és az éppen bejelentkezett felhasználó által kitöltött témák száma.

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírás

. 2.7.4.3. ábra - Általános statisztika kártya

#### Gyorselérés kártyák

Itt találhatóak kategóriánként a témák. Ezeket a kártyákat össze lehet csukni és a témákra kattintva megjelenik az adott téma részletező oldala, ahol meglehet válaszolni a témához kapcsolódó kérdéseket.

A képen szöveg, képernyőkép, szoftver, Betűtípus látható

Automatikusan generált leírás

. 2.7.4.4. ábra - Gyorselérés kártyák

### Részletező oldal

A részletező oldal tetején az adott téma címe és a „Vissza a kérdésekhez” gomb, ami visszavisz a kérdések oldalra. Alatta egy űrlap található, amin két rádiógomb és egy jelölőnégyzetes válaszadási lehetőség van. Ezek alatt pedig kettő hosszabb kifejtős kérdés található.

A képen szöveg, képernyőkép, szoftver, Téglalap látható

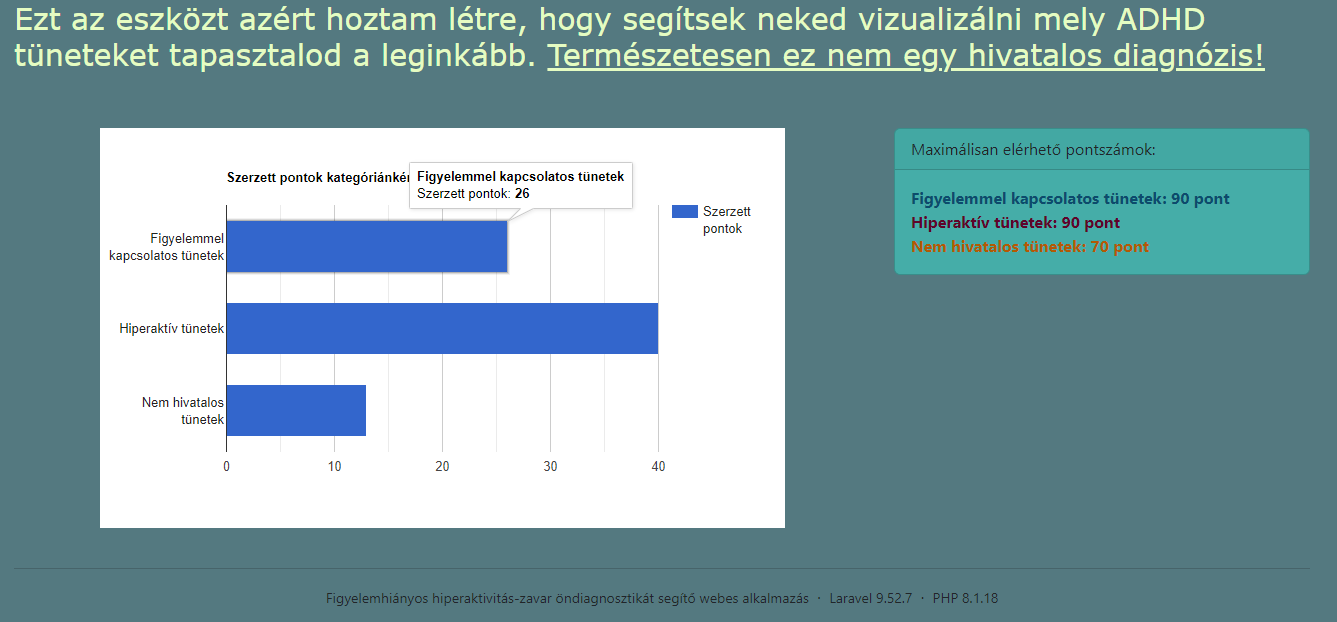
Automatikusan generált leírás

2.7.5. ábra - Részletező oldal űrlap

Az űrlap mellett pedig a kérdések oldalon is megtalálható gyorselérési kártyák találhatóak.

### Statisztika oldal

Ha egy felhasználó kitöltötte az összes témához tartozó kérdést akkor a fejlécen megjelenik egy „Statisztika” gomb amire kattintva átirányít minket az oldal a statisztika oldalra. Itt kategóriánként megjelennek a saját pontjaink és a maximálisan elérhető pontok is.



2.7.6. ábra - Statisztika oldal

# Fejlesztői dokumentáció

## Fejlesztői környezet

### Verziókövetés

A fejlesztés során fontos volt, hogy nyomon tudjam követni az alkalmazás bizonyos állapotait. Előfordult, hogy egy-egy változtatás nem úgy működött ahogyan szerettem volna ezért vissza kellett állnom egy korábbi verzióra.

A szoftverfejlesztés általában csapatban zajlik. Ilyenkor a kódot több fejlesztő szerkeszti a saját környezetében. Szinkronizálással biztosítani kell azt, hogy mindenki a legfrissebb kódbázissal tudjon dolgozni. Minden változást tudunk címkézni, leírást biztosítani hozzájuk. Ezzel segítve kollégáink munkáját. Jelen esetben fontos volt, hogy tudjam, hogy melyik változtatással mi került be a kódbázisba. Én ehhez a GIT verziókezelő rendszert használtam.

#### GitHub Desktop

A GitHub [12] által biztosított GitHub Desktop [13] alkalmazást használtam, ami rendkívül egyszerű felületet biztosított ahhoz, hogy a verziókövetés gördülékenyen haladjon. Minden változtatás Feltöltöttem, így könnyen visszanézhető, hogy hol mi történt.

A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

Automatikusan generált leírás

3.1.1.1. ábra - Github Desktop

### Kódszerkesztő program

Kódszerkesztésre a Visual Studio Code-ot [4] használtam, ami egy ingyenes nyílt forráskódú kódszerkesztő.

### Adatbázis

A témabejelentőm elkészítésekor még MySQL-t szerettem volna használni, de hamar rá kellett jönnöm, hogy teljesen felesleges hiszen kevés adatbázis táblával és egyszerű (integer, karakter, logikai és dátum) fogok dolgozni. Ezért döntöttem úgy, hogy inkább SQLite-ot használok.

#### Adatbázis kezelő

Az adatbázisom megtekintésére és kezelésére a „DB Browser for SQLite” [11] nevű programot használtam, ami letölthető az <sqlitebrowser.org> oldalról. Ebben a programban minden létező adatbázis táblát monitorozni és szerkeszteni tudtam.

### Keretrendszer

Az alkalmazásomhoz a Laravel nyílt forráskódú PHP alapú keretrendszert használtam. Laravellel egy háromrétegű, MVC (model-viev-controller magyarul: modell-nézet-vezérlő) felépítésű programot lehet készíteni, amelyben elkülönül a modell, nézet és a logika megvalósítása. Ennek segítségével sokkal átláthatóbb lett az alkalmazásom. További előnye a Laravelnek, hogy hozzáfér a relációs adatbázishoz, aminek az alapja az SQLite.

#### Miért pont a Laravel?

* A Laravel-t progresszív keretrendszernek is nevezik, mivel a fejlesztőkkel és a felhasználókkal együtt növekszik.
* Komplett funkcionális személyre szabhatóságot és könnyen bővíthetőséget is nyújt.
* Rendkívül jól dokumentált
* Jól optimalizált kódbázissal bír, emiatt nagyobb forgalmú weboldalak motorja is lehet. [2]
* Maximális biztonsággal rendelkezik, mert komoly üzleti rendszerek is létrejöttek a segítségével.

#### Kezdőcsomag

Az alkalmazás elkészítéséhez a Szerveroldali webprogramozáson is használt kezdőcsomagot használtam. Ez a csomag rendelkezik előre elkészített váz szerű oldalakkal és egy saját CSS osztállyal. [5]

### Használt szoftverek verziói

|  |  |
| --- | --- |
| Szoftver | Általam használt verzió |
| PHP | 8.1.18 |
| Laravel | 9.52.7 |
| Visual Studio Code | 1.78.2 |
| DB Browser for SQLite | 3.12.2 |
| Google Chrome | 113.0.5672.127 |
| GitHub Desktop | 3.2.3 |
| Composer | 2.5.5 |
| Node.js | 18.16.0 |
| npm | 9.6.6 |

## MVC architektúra

Ebben a háromrétegű szoftverben minden rétegnek megvan a saját feladata.

* A Modell az alkalmazás és az adatbázis közötti kommunikációért felel.
* A Nézet az oldalak megjelenítéséért felel a felhasználóknak.
* A Vezérlő kezeli a felhasználói interakciókat és a modellt, ezáltal módosíthatja az adatbázist is.

A képen szöveg, képernyőkép látható

Automatikusan generált leírás

3.3 ábra - MVC architektúra  
Forrás: [1]

### Modell bemutatása

#### Modell létrehozása

A „quiz” mappában (későbbiekben főkönyvtár) parancssort nyitva a következő paranccsal lehet létrehozni új modellt: „php artisan make:model <Modell neve> -m -s” A „-m” kapcsolóval létrehozzuk a modellünkhöz egy migrations-t is erről majd később. A „-s” kapcsolóval pedig egy seedert erről is később lesz szó. A modellek nevét érdemes angolul megadni, így az adatbázisban automatikusan többesszámként szerepel majd a táblánk neve. Én Question néven készítettem egy modellt, ami az adatbázisba questions néven került be.

#### Modell felépítése

A „-m” kapcsoló miatt a „főkönyvtár/database/migrations” mappában létrejött egy fájl, amiben létrejött egy új osztály, ami a Migrations osztályból származik le. Ebben a fájlban tudjuk megadni, hogy a táblánknak milyen típusú, nevű vagy akár hosszúságú mezői legyenek, hogy egyedi legyen-e vagy legyen-e alapértelmezett értéke. Az újonnan létrejött fájlunkban a Schema osztály create($a,$b) függvény segítségével hozzuk létre az adatbázis táblánkat. Az első paraméter a létrehozni kívánt táblánk neve, a második pedig egy függvény, ami egy Blueprint típusú változót vár paraméterként. A második paraméterben kapja meg a függvény a tábla mezőinek paramétereti.

A képen szöveg, képernyőkép, szoftver látható

Automatikusan generált leírás

3.2.1.2. ábra - create függvény

Szükségünk van az adatbázisunkban arra, hogy tudjuk, hogy mikor készült és mikor lett utoljára módosítva. Ezt a timestamps() adja meg nekünk created\_at és updated\_at névvel. Új sor beszúrásakor mindkét attribútum értéke a beszúrás idejével fog megegyezni. Egy már meglévő sor módosításakor a created\_at értéke nem változik viszont az updated\_at értéke a módosítás dátumára fog változni. Mivel egy questions tábla béli rekordhoz csak egy user és egy category tartozik ezért a questions táblában kell eltárolni a hozzá tartozó user-eket és category-kat. Ez a képen látható utolsó négy sorban történik meg. Ezen kívül definiálnunk kell ezeket a kapcsolatokat a táblákhoz tartozó modellekben is.

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

3.2.1.2. ábra – Question modell kapcsolat definíció

A képen szöveg, Betűtípus, képernyőkép, Grafika látható

Automatikusan generált leírás

3.2.1.2. ábra – User modell kapcsolat definíció

#### Táblák létrehozása

Megfelelően létrehozott Modellek és Migration-ök után a főkönyvtárban nyitott parancsorba a következő parancs kiadásával tudunk táblákat létrehozni: „php artisan migrate”. Ha ez a prarancs hiba nélkül lefutott akkor megjelenik a „főkönyvtár/database” mappában egy „database.sqlite” nevű fájl. Ebben a fájlban lesz eltárolva az összes adatbázistáblánk. Ezt a fájlt könnyen megtekinthetjük és szerkeszthetjük a DB Browser for SQLite nevű alkalmazásunkkal.

#### Táblák automatikus feltöltése

Korábban létrehoztunk egy seedert minden modellhez. A seederek abban segítenek nekünk, hogy alapvető adatokkal feltölthessük az adatbázisunkat. A seederek futtatásához a főkönyvtárban indított parancssorban következő paranccsot kell megadni: „php artisan migrate --seed”. Az én alkalmazásom esetében így létrejön a három alapkategória, egy test felhasználó és a hozzá tartozó kérdéscsoportok.

A képen szöveg, Betűtípus, képernyőkép látható

Automatikusan generált leírás

3.2.1.3. ábra - Kategória létrehozás

Minden felhasználó regisztrációja generál egy register eseményt, amit a „főkönyvtár/app/Listeners/RegisterListener.php” fájlban egy eseménykezelő kezel.

A képen szöveg, Betűtípus, képernyőkép, sor látható

Automatikusan generált leírás

3.2.1.3. ábra – RegisterListener handle függvény

### Nézet bemutatása

#### Útvonalak kezelése

Amikor egy keretrendszer nélküli, statikus HTML weboldalon a "/login" útvonalat hozzáadjuk az URL címhez, a program elkezdi az adott fájl keresését a főkönyvtárban. A "login" nevű fájl megtalálása esetén a weboldal az ahhoz tartozó tartalmat jeleníti meg. Ez például lehetővé teszi a felhasználók bejelentkezési folyamatának kezelését és a hozzá kapcsolódó funkcionalitások biztosítását. A Laravel keretrendszer más módon kezeli ezeket az útvonalakat. Ha egy Laravel projektben a "/login" útvonalra keresünk, akkor nem egy fájlt fog keresni, hanem a "főkönyvtár/routes/web.php" fájlban vizsgálja, hogy az adott útvonal definiálva van-e. A Laravel keretrendszer útvonalakat(routes) használ a különböző webes kérések kezelésére, és ezeket a web.php fájlban lehet definiálni. Tehát a "/login" útvonal esetén a Laravel a web.php fájlt ellenőrzi, hogy megtalálja-e az adott útvonal definiálását. A feladat megoldására a Route osztályban található get($a, $b) függvényt használhatjuk, ami két paramétert vár: az első paraméter az útvonal, a második pedig egy függvény. A függvény akkor fut le, ha az URL megegyezik az első paraméterben megadott útvonallal. Ha egy nézetet szeretnénk megjeleníteni a felületen, akkor a függvényben a view()-val kell visszatérnünk. Így definiálhatjuk az útvonalakat és a hozzájuk tartozó függvényeket a Laravel keretrendszerben:

A képen szöveg, Betűtípus, képernyőkép, sor látható

Automatikusan generált leírás

3.2.2.1. ábra - get függvény

A példában is látható hogy a home.index nézettel tér vissza a függvény. Ezért el is kezdi keresni a „főkönyvtár/resources/views/home” mappában az index.blade.php fájlt. Ha létezik a fájl akkor meg is jeleníti. Ha nem létezik akkor 404-es hibát kapunk. Fontos, hogy csak blade.php kiterjesztésű fájlokat keres az alkalmazás.

#### Blade

A Laravel egy Blade nevű sablont használ. A Blade legfőbb előnye az egyszerűsített szintaxisa. A .blade.php kiterjesztésű fájlok használhatják a Blade sablont, ezért ezt a kiterjesztést érdemes használni az összes nézet fájl létrehozásakor. A kettős kapcsos zárójelek között kódot írhatunk a Blade-ben. Például a "<?php echo "Teszt echo"; ?>" kód a Blade-ben egyszerűen a következőképpen írható: "{{ "Teszt echo" }}". Ezenkívül a Blade lehetővé teszi a PHP utasítások egyszerűbb használatát, mint például az if, else, foreach stb. A Blade által biztosított egyszerűbb szintaxisnak köszönhetően tisztábbá és olvashatóbbá tehetjük a kódunkat.

#### Layouts mappa

A Layouts mappában található app.blade.php fájlban található a fej- és lábléce minden oldalnak. Minden oldal igazából az app.blade.php oldalt egészíti ki úgy, hogy a „@yield('content')” részben kvázi belemásolja a fájlba az adott oldal tartalmát. Minden tartalomnak, amit a fej- és lábléc közé szeretnék beszúrni a következő címkével kell kezdődnie: „@section('content')” és a következővel kell végződnie: „@endsection”.

### Vezérlő bemutatása

Az utolsó réteg a Vezérlő (Controller). A Vezérlő segítségével kommunikál egymással a Modell és a Nézet. A Vezérlők a „főkönyvtár/app/Http/Controllers” mappában találhatóak. Azalkalmazásom egyszerre két Vezérlőt is használ, egyet a válaszok elmentésére és egyet a statisztika megjelenítésére és jogosultságkezelésére. Vezérlő létrehozása rendkívül egyszerű, ki kell adni az alkalmazás mappájában nyitott terminálban a következő parancsot: „php artisan make:controller <a vezérlőnk neve>” Igy létrejön egy fájl amiben egy osztályunk is lesz ami a Controller ősosztályból származik le.

### Adatok továbbítása

Az alkalmazásomban egy kivétellel (Kezdőlap) nem elég statikus felületet biztosítanom. Minden más oldal megfelelő megjelenítéséhez szükségem van az adatbázisban eltárolt adatokra. Az adattovábbítás többféleképpen is működhet, én az alkalmazásomban a view() függvény második paramétereként adom át például a kategóriák listáját, felhasználók számát és a kérdések listáját.

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

3.2.4 ábra - Adatok átadása a Nézetnek

### Adatbázis használata

Mint ahogy korábban említettem a Vezérlőkben végzünk minden adatbázissal kapcsolatos feladatot, mint például az adatok mentése vagy szerkesztése. Ezeket kétféleképpen is tehetjük. Az első módszer az SQL lekérdezések a második pedig a LAravel által biztosított Eloquent módszer. Az Eloquent módszer egyszerűbb szintaxisával teszi lehetővé számunkra az egyszerű használatot. Mindkét módszer használatánál megengedettek az SQL utasítások, mint például a where, select, jon, group by stb.

### Jogosultság és hibakezelés

Az alkalmazásom elkészítése során rendkívül fontos volt számomra, hogy biztonságos legyen az olyan felhasználókkal szemben is akik nem úgy használnák mint ahogyan én azt megálmodtam. Pontosan emiatt az alkalmazásom minden útvonala megfelelően le van kezelve. Ha egy felhasználó olyan útvonalat szeretne elérni, amihez nincsen jogosultsága akkor az oldal visszairányít vagy egy üzenettel fogadja a felhasználót.

## Adatbázis bemutatása

A program létrehozása során fontos volt eldönteni, hogy hogyan szeretném tárolni a felhasználók adatait. összesen 3 darab adatbázistábla létrehozására volt szükségem az alkalmazás megvalósításához. Csak egy a többhöz kapcsolat fordul elő.

### Users tábla

Ez a tábla tartalmazza a regisztrált felhasználók adatait. A tábla felépítése a következő:

|  |  |  |  |
| --- | --- | --- | --- |
| Név | Típus | Mit tartalmaz | Komment |
| id | id | A felhasználó egyedi azonosítóját | elsődleges kulcs, egyedi, növekszik |
| name | string | felhasználónév | nem egyedi érték |
| email | string | e-mail cím | egyedi érték |
| password | string | jelszó | hashelve tárolódik |
| category1\_points | integer | az első kategóriában szerzett pontokat | Kitöltésenként újra kiszámolódik |
| category2\_points | integer | a második kategóriában szerzett pontokat | Kitöltésenként újra kiszámolódik |
| category3\_points | integer | a harmadik kategóriában szerzett pontokat | Kitöltésenként újra kiszámolódik |
| done | boolean | azt, hogy a felhasználó kitöltötte-e az összes kérdést | ettől függ, hogy a felhasználó jogosult-e a statisztika megtekintésére |

### Categories tábla

Ez a tábla tartalmazza a kategóriák adatait.

|  |  |  |  |
| --- | --- | --- | --- |
| Név | Típus | Mit tartalmaz | Komment |
| id | id | A kategória egyedi azonosítóját | elsődleges kulcs, egyedi, növekszik |
| name | string | A kategória nevét | nem egyedi érték |
| color | string | A kategória megjelenítéséhez használt színt | #RRGGBBAA formátumú színkód |
| cover\_image\_path | string | A kategóriához tartozó téma borítóképének elérési útvonalát |  |

### Questions tábla

Ez a tábla tartalmazza a témák adatait. Egy témához egy felhasználó és egy kategória tartozik. Egy felhasználóhoz sok téma és egy kategóriához is sok téma tartozik.

|  |  |  |  |
| --- | --- | --- | --- |
| Név | Típus | Mit tartalmaz | Komment |
| id | id | A téma egyedi azonosítóját | elsődleges kulcs, egyedi, növekszik |
| name | string | A téma címét | nem egyedi érték |
| question1 | string | Az első kérdés szövegét |  |
| question1\_points | integer | Az első kérdés pontjait |  |
| question2 | string | A második kérdés szövegét |  |
| question2\_1 | boolean | A második kérdés első jelölőnégyzetének állapotát |  |
| question2\_2 | boolean | A második kérdés második jelölőnégyzetének állapotát |  |
| question2\_3 | boolean | A második kérdés harmadik jelölőnégyzetének állapotát |  |
| question2\_points | integer | A másodikkérdés pontjait |  |
| question3 | string | A harmadik kérdés szövegét |  |
| question3\_points | integer | A harmadik kérdés pontjait |  |
| question4 | string | A negyedik kérdés szövegét |  |
| question4\_text | text | A negyedik kérdés válaszának szövegét |  |
| question5 | string | Az ötödik kérdés szövegét |  |
| question5\_text | text | Az ötödik kérdés válaszának szövegét |  |
| done | boolean | Hogy minden kérdés meg lett-e válaszolva | Ettől függ egy téma állapota |
| sum | integer | A kérdésekre kapott pontok összegét |  |
| user\_id | unsignedBigInteger | A kérdéshez tartozó felhasználó azonosítóját | idegen kulcs |
| category\_id | unsignedBigInteger | A kérdéshez tartozó kategória azonosítóját | idegen kulcs |

## Felhasználói diagram

A következő felhasználói diagram segít a felhasználóknak megérteni a program működését úgy, hogy bemutatja, hogy melyik funkció hogyan érhető el és mitől függ.

A képen szöveg, diagram, vázlat, kör látható

Automatikusan generált leírás

3.4 ábra - Felhasználói diagram

## Felhasználói történet

A Felhasználói történeteket arra használjuk, hogy leírjuk, mit szeretne egy felhasználó elérni vagy milyen funkciókat vár el egy szoftvertől. Ez segít a fejlesztőket megérteni és megvalósítani ezeket az igényeket. A Felhasználói történetek rövid, egyszerű leírások, amiket könnyen érthet mindenki.

|  |  |  |
| --- | --- | --- |
| AS A | Felhasználó |  |
| I WANT TO | Regisztrálni |  |
| 1 | Given | Elindítottuk a programot |
|  | When | Rákattintok a Regisztráció gombra |
|  | Then | Megjelenik a regisztrációs felület |
| 2 | Given | Megjelent a regisztrációs felület |
|  | When | Megfelelően kitöltjük a regisztrációs űrlapot |
|  | Then | A jelszó helyén pontok jelennek meg |
| 3 | Given | Megfelelően kitöltöttük a regisztrációs űrlapot |
|  | When | Rákattintottunk a Regisztráció gombra |
|  | Then | Megjelenik a Kérdések oldal |
| I WANT TO | Bejelentkezni |  |
| 1 | Given | Elindítottuk a programot |
|  | When | Rákattintok a Bejelentkezés gombra |
|  | Then | Megjelentik a bejelentkező felület |
| 2 | Given | Megjelent a bejelentkező felület |
|  | When | Kitöltöm egy már regisztrált felhasználó adataival az űrlapot |
|  | Then | A jelszó helyén pontok jelennek meg |
| 3 | Given | Megfelelően van kitöltve az űrlap |
|  | When | Megnyomom a Bejelentkezés gombot |
|  | Then | Megjelenik a Kérdések oldal |
| I WANT TO | Megjeleníteni egy témához kapcsolódó kérdéssort |  |
| 1 | Given | Megjelenik a Kérdések oldal |
|  | When | Rákattintok a Téma kérdéseinek kitöltése gombra egy téma alatt |
|  | Then | Megjelenik az adott téma részletező oldala |
| 1 | Given | Megjelenik a Kérdések oldal |
|  | When | A Kérdesek oldalon jobb oldalt található gyorselérés kártyán rákattintok egy témára |
|  | Then | Megjelenik az adott téma részletező oldala |
| 1 | Given | Megjelenik egy kérdés részletező oldala |
|  | When | A kiválasztott kérdés részletező oldalán jobb oldalt található gyorselérés kártyán rákattintok egy témára |
|  | Then | Megjelenik a kiválasztott téma részletező oldala |
| I WANT TO | Elmenteni egy kérdéssorra adott válaszaimat |  |
| 1 | Given | Megjelenik a kiválasztott téma részletező oldala |
|  | When | Megválaszolom a kérdéseket |
|  | Then | Megfelelően megjelennek a kérdésekre adott válaszaim |
| 2 | Given | Megfelelően kitöltöttem az adott témához tartozó kérdéseket |
|  | When | Megnyomom a Kész gombot |
|  | Then | Elmentődnek a válaszaim és az adott téma Állapota zöld pipa lesz |
| I WANT TO | Megtekinteni a statisztika oldalt |  |
|  | Given | Kitöltöttem az összes témához tartozó kérdéssort |
|  | When | Rákattintok a Statisztika gombra |
|  | Then | Megjelenik a Statisztika oldal |

## Tesztelés

A tesztelés során manuális teszteket hajtottam végre, amit a következő pontokban szedtem össze.

### Jogosultságkezelés

#### Nem bejelentkezett felhasználó

Nem bejelentkezett felhasználóval szeretném elérni a következő oldalakat:

|  |  |  |  |
| --- | --- | --- | --- |
| Felkeresett oldal | Elvárt működés | Valós működés | Eredmény |
| /questions | Nem sikerül megnyitni. | Átirányít a bejelentkező oldalra. | OK |
| /statistics | Nem sikerül megnyitni. | Átirányít a bejelentkező oldalra. | OK |
| /questions/1/edit | Nem sikerül megnyitni. | Átirányít a bejelentkező oldalra. | 0K |

#### Bejelentkezett felhasználó

Bejelentkezett felhasználóval szeretném elérni az alábbi oldalakat. A felhasználó nem töltötte még ki az összes témához tartozó kérdést, a hozzá tartozó kérdések azonosítója 1-25 ig terjed.

|  |  |  |  |
| --- | --- | --- | --- |
| Felkeresett oldal | Elvárt működés | Valós működés | Eredmény |
| /questions/30/edit | Hibaüzenettel nem enged tovább az alkalmazás. | Hibaüzenettel nem enged tovább. | OK |
| /statistics | Hibaüzenettel nem enged tovább az alkalmazás. | Hibaüzenettel nem enged tovább. | OK |

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

3.7.1.2. ábra - /statistics oldal

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

3.7.1.2. ábra - /questions/30/edit oldal

### Regisztráció, bejelentkezés

#### Regisztrálás

|  |  |  |  |
| --- | --- | --- | --- |
| Esemény | Elvárt működés | Valós működés | Eredmény |
| A regisztrációs űrlapon már használatban lévő e-mail-t szeretnék megadni | Hibaüzenetet kapok és nem enged tovább az alkalmazás. | A következő hibaüzenetet kapom: „The email has already been taken.” és nem enged tovább az alkalmazás. | OK |
| A regisztrációs űrlapon 8 karakternél rövidebb jelszót adok meg. | Hibaüzenetet kapok és nem enged tovább az alkalmazás. | A következő hibaüzenetet kapom: „The password must be at least 8 characters.” és nem enged tovább az alkalmazás. | OK |
| A regisztrációs űrlapon egyedi e-mail címet adok meg, a jelszavam 8 karakternél hosszabb. | Sikeres regisztráció után átirányít az újonnan regisztrált felhasználó Kérdések oldalára. | Sikeres regisztráció után átirányít az újonnan regisztrált felhasználó Kérdések oldalára. | 0K |

#### Bejelentkezés

|  |  |  |  |
| --- | --- | --- | --- |
| Esemény | Elvárt működés | Valós működés | Eredmény |
| A bejelentkezés űrlapon rossz e-mailt vagy jelszót adok meg. | Hibaüzenetet kapok és nem enged tovább az alkalmazás. | A következő hibaüzenetet kapom: „These credentials do not match our records.” és nem enged tovább az alkalmazás. | 0K |
| A bejelentkezés űrlapon már regisztrált e-mail, jelszó párost adok meg. | Sikeres bejelentkezés után átirányít a bejelentkezett felhasználó Kérdések oldalára. | Sikeres bejelentkezés után átirányít a bejelentkezett felhasználó Kérdések oldalára. | 0K |

### Kérdéssor kitöltése

|  |  |  |  |
| --- | --- | --- | --- |
| Esemény | Elvárt működés | Valós működés | Eredmény |
| Az első vagy a harmadik kérdésnél található rádiógomb mezőkből semmi nincsen kiválasztva | Hibaüzenetet kapok és nem enged tovább az alkalmazás. | A következő hibaüzenetet kapom: „Ez egy radio button, legalább egy dolognak kiválasztva kell lennie!” és nem enged tovább az alkalmazás. | 0K |
| A negyedik vagy ötödik kérdésre adott válasz rövidebb, mint tíz karakter. | Hibaüzenetet kapok és nem enged tovább az alkalmazás. | A következő hibaüzenetet kapom: „Ennek a mezőnek legalább 10 karaktert tartalmaznia kell. Töltsd ki kérlek!” és nem enged tovább az alkalmazás. | 0K |
| Minden mező megfelelően van kitöltve és megnyomom a Kész gombot | Nem kapok hibaüzenetet és tovább enged az alkalmazás, a megválaszolt téma állapota zöld pipára vált. | Nem kapok hibaüzenetet és tovább enged az alkalmazás, a megválaszolt téma állapota zöld pipára vált. | 0K |

### Statisztika

|  |  |  |  |
| --- | --- | --- | --- |
| Esemény | Elvárt működés | Valós működés | Eredmény |
| Kitöltöm az utolsó témához tartozó kérdéseimet és minden téma állapota zöldre vált. | Megjelenik a Statisztika gomb a fejlécen. | Megjelenik a Statisztika gomb a fejlécen. | 0K |
| Elérhetővé válik a Statisztika gomb a fejlécen. | A gomra kattintva megjelenik a Statisztika oldal. | A gomra kattintva megjelenik a Statisztika oldal. | 0K |

# Összefoglalás és további fejlesztési lehetőségek

## Összefoglalás

A szakdolgozatomban egy Quiz alkalmazást hoztam létre, ami remélhetőleg tud segíteni azoknak a felnőtteknek akik valószínűleg ADHD-val rendelkeznek, úgy hogy elindulnak a diagnosztizáláshoz vezető úton. Természetesen szaktudásom nincsen a témában csak amennyit a rendelkezésemre álló írásokból, munkafüzetekből szereztem. Sikerült megvalósítanom majdnem mindent az eredeti tervekhez képest. A témánkénti részletező oldalakat sajnos nem valósítottam mert úgy érzem, hogy nem vagyok még megfelelően felkészült a témában. Szerintem minden hibát, esetleges problémát lekezeltem egy hibaüzenettel, egy egyszerű átirányítással vagy egy az oldalon található üzenettel. A környezet beállítása egyáltalán nem volt körülményes és minden adott volt ahhoz, hogy a terveim alapján elkészítsem az alkalmazást.

## További fejlesztési lehetőségek

Véleményem szerint egy alkalmazás sosem lehet tökéletes és soha sincsen teljesen kész. Ez igaz az én alkalmazásomra is hiszen ezt is mint minden más alkalmazást a világon tovább lehet fejleszteni, szépíteni, építeni. Az alkalmazásomhoz sok kisebb-nagyobb fejlesztésre van lehetőség. Ilyen például a fent említett részletező oldalak. Ilyen még az adminisztrátorként kérdések, kategóriák és témák hozzáadása, ezek törlése és szerkesztése. A különböző kérdések/témák értékelése felhasználóként hasznosság/realitás szempontból. Illetve ilyen még a témák alatti kommentelés is. A kérdéseket lehet pontisítani és módosítani, hogy szakmai szempontból a legmegfelelőbb képet alakíthassák ki a kitöltőről. A statisztika oldal lehetne sokkal részletesebb mélységekbe menőbb.

# Irodalomjegyzék

1. <https://hu.wikipedia.org/wiki/Modell-n%C3%A9zet-vez%C3%A9rl%C5%91>
2. <https://gremmedia.hu/laravel-keretrendszer-mit-erdemes-tudni-rola-tenyleg-legnepszerubb-keretrendszer>
3. <https://hu.wikipedia.org/wiki/Figyelemhi%C3%A1nyos_hiperaktivit%C3%A1s-zavar>
4. „Visual Studio Code,” <https://code.visualstudio.com/>
5. „Kezdőcsomag” <https://github.com/szerveroldali/blog_basic_assets>
6. <https://bhc.hu/betegsegek/hiperaktivitas>
7. <https://semmelweis.hu/pszichiatria/files/2016/10/adhd_el%C5%91ad%C3%A1s_magyar_20170403.pdf>
8. <https://www.theminiadhdcoach.com/adhd-symptoms>
9. „PHP,Composer,npm”<https://github.com/totadavid95/PhpComposerInstaller/releases/latest/download/PhpComposerInstaller.zip>
10. „Node.js” <https://nodejs.org/en/download>
11. „DB Browser for SQLite” <https://sqlitebrowser.org/>
12. „GitHub” <https://github.com/>
13. „GitHub Desktop” <https://desktop.github.com/>
14. „Laravel dokumentáció” <https://laravel.com/>