Beregszászi Pál

ZÁRÓDOLGOZAT

Madarász Anna  
2020

Tartalomjegyzék

[**1.** **Bevezetés** 2](#_Toc37772222)

[**2.** **Felhasznált technológiák bemutatása** 3](#_Toc37772223)

[2.1 C# 3](#_Toc37772224)

[2.2 MySQL /phpMyAdmin 3](#_Toc37772225)

[**3.** **Feladat részletezése** 3](#_Toc37772226)

[3.1 Adatok meghatározása 3](#_Toc37772227)

[3.2 Felhasználó 3](#_Toc37772228)

[**4.** **Rendszerterv** 3](#_Toc37772229)

[**5.** **Összefoglalás** 5](#_Toc37772230)

[**6.** **Irodalomjegyzék** 5](#_Toc37772231)

# **Bevezetés**

Egy esetlges online-oktatás keretében történő számonkérésnél hasznos lehetnek, olyan alkalmazások, melyek nem online weboldalakon elérhetőek, hanem asztali formában.

Záródolgozatomban bemutatom az alapkoncepciót, az elgondolást és végül a megvalósítását az általam készített programnak. Az elején igyekszek bemutatni, az elmúlt években tanult és elsajátított technológiákat elméleti szinten, hogy egy kis rálátást adjak az olvasónak. Aztán magára a munkafolyamatba engedek betekintést, lépésről lépésre, egészen a megvalósult programig. Mindeközben pedig részletesen bemutatva az alkalmazott fejlesztéseket és a technológiák megjelenési formáit a dolgozatomban. A kész programról, illetve a munkafolyamat egyes részleteiről készült fotók is segítik a megértést. Legvégül pedig egy rövid összegzésben, értékelem az elkészült programot, és megosztom az ezzel kapcsolatos gondolataimat, illetve véleményemet.

Záródolgozatomban egy, a tanulmányaim során, a mindennapokon is használható alkalmazást alkottam meg. A felelet-választós tesztek nem állnak messze tőlem. Volt alkamam már az egyetemi tanulmányai során is ilyen teszetet felhasználóként kitölteni. Jelenlegi programozási tanulmányaim során pedig el tudtam sajátanítani azt a tudást ami ahhoz kell, hogy egy általam készített alkalmazást létre tudjak hozni, olyan formában, ami felhasználóbarát, szórakoztató, játékos, mégis hasznos.

A fentiekből kiderülhet, hogy az alkalmazás,a mit készítettem egy több kategóriás kvízt, mely a jelenlegi egyetemi tanulmányaimhoz kapcsolódik. A felhasználó csak egy regisztráció, így egy profil létrehozása, után tud belépni, így kap „engedélyt” a kvíz kitöltésére. Az egyes kívzeken elért pontokat egy táblázatban lehet nyomonkövetni a felhasználó profilján, illetve, ott látható, hogy összesen hány pontot gyűjtött is eddig össze. Az egyes kategóriák, a biológiának több alterületére fókuszálnak. Egy kategóriában ??? darab random kérdést generál a program, amit előzetesen egy adatbázisban lett rögzítve.

Azért választottama C# programkörnyezetet, mert az elmúlt 2-3 évben, tanulmányaim során ezt használtuk a legtöbbször, illetve ez a mindennapokban használt legfőbb porgramozási nyelv, amiről úgy gondoltam, hogy szeretnék egy záródolgozat keretében is jobban belemerülni a témába.

Az elkészült program igényelhet egyéb továbbfejlesztést a továbbiakban, illetve némi „csinosítást” de jelenlegi állapotában is hasznos lehet a felhasználó számára. Az elkészítése során megszerzett és elsajátított tudást a jövőben is fogom tudni hasznosítani, akár leendő munkám során.

# **Felhasznált technológiák bemutatása**

## C#

A C++ egyik utódjának tekinthető a Microsoft Visual Studio.NET fejlesztési környezetében használt porgramozási nyelv a C#, mely 2002 óta érhető el a felhasználóknak. Egyben rendelkezik a C, C++ hatékonyságával, viszonylag egyszerűnek is mondható, illetve elég jól teljesít is gyorsaság terén, ami a Visual Basic egyik fontos jellemzője is. A C# egy objektumorientált programnyelv, mely felahsználóberát, alkalmazások készítésére alkalmas. A C# bázisnyelve a új .NET keretrenszernek. Amit még a C# programnyelv jellemzői közé tartozik, hogy Neumann-elvű, professzionális, akár rendzserporgram tervezésére is alkalmas nyelv. [[1](#_Irodalomjegyzék)]

## MySQL /phpMyAdmin

# **Feladat részletezése**

## Adatok meghatározása

## Felhasználó

# **Rendszerterv**

A záródolgozatomhoz, egy C# nyelven íródott, adatbázis alapú, asztali kvíz alkalmazást fejlesztettem. A munkafolyamatot a képzés alatt megtanított és a fentebb leírt technológiákkal felhasználásval készült.

A kész program 3 részből állt össze egy teljes, működő egységgé. Először is a kvíz kérdéseit hoztam létre, illete az adatbázis alapján egy Excel fájlban. melyek .*ods* kiterjesztéssel mentve, importálni lehetett a phpMyAdmin oldalon. 4 munkalap készült:

* User
* Kérdés
* Válasz
* Kategórák.

A *User* munkalap gyűjti a felhasználóneveket, jelszavakat, illetve a kvízek során összegyűjtött pontszámokat. Az egyes felhasználók rendelkeznek egy *id*-val is. A *Kérdés* munkalap tartalmazza az egyes kategóriák *id-*jához tartozó kérdéseket. A *Válasz* munkalap, az egyes kérdésekhez tartozó, 3 lehetséges választ tartalmazzák. Mindegyik válasz rendelkezik egy egyedi *id-*val, egy-egy hamis, vagy igaz jelzővel, attól függően, hogy a kérdéshez, melyik a helyes válasz. A *Kategória* munkalap tartalmazza, az egyedi *id-*val rendelkező kategóriákat.

A második lépés az volt, hogy a XAMPP telepeítését követően a *localhost/phpmyadmin* honlapon, egy az alkalmazáshoz illő adatbázist hoztam létre *bioquiz* néven, ahova az .ods kiterejsztéső Excel fájlt imporálni kellet*.* Ezen belül 4 tábla:, az Excel fájlnak megfelelően:

* *kategoriak*
* *kerdes*
* *user*
* *valasz*

A *kategoriak* tábla beállításánál típusaként *int(1)* lett, mivel a kategóriák száma nem haladja meg a10-t, ezért nem adtam neki nagyobb értéket, illetve *AUTO-INCREMENT-*t lett beálíltva,. A *megnevezes* oszlop esetében a típus *varchar(17)*, mivel a kategóriák egyes nevei nem hosszabbak 17 karakternél.

A *kerdes* tábla 3 elemet tartalamaz: az *id*, mely *int(2)* típsú, a *szoveg* *varchar(80)* , és a *kategoria\_id int(1)* típussal. A kapcsolati nézetben beállítottam az idegen kulcs megszorításoknál egy *kerdes\_kat* nevű megszorítási tuljadonságot, mely *ON DELETE NO ACTION, ON UPDATE CASCADE* funkcióval rendelkezik, a hozzá tartozó osztlop a *kategoria\_id,* mely a *kategoria* táblában lévő *id* oszlophoz kapcsolódik.

A *user* táblába gyűjti felhasználok adatait, melyeket egyedi *id-*val lát el. Hozzá kapcsolja a felhasználó adatait: név, felhasználónév, jelszó, pontszám, email. A Jelszó, egy MD5, kiterjesztéssel lett titkosítva. Az adattípusokat tekintve, a *nev*, a *felhasznalonev, a jelszo,* és az *email* mind *varchar*, a hossz pedig mindegyik a sajátjának megfelelő hosszússágú. A *pontszam* *int(10)* típusú.

A *valasz* tábla tartalamazza a feltett kérdésekre az adott lehetséges válaszokat. A szerkezetét tekintve 4 elemből áll: *id,* mely *int(3)*, *kerdes\_id* *int(2), valasz varchar(64), helyes(ENUM(’IGAZ’,’HAMIS’))* típusokkal. Ezek közül kiemelném, az utolsó, *helyes* oszlopot, melynek típusa *ENUM,* melynek értékkészletét én határoztam meg, ebben az esetben, igaz vagy hamis opció lehetésges. Az *id AUTO-INCREMENT* tulajdonsággal rendelkezik. A kapcsolati nézetet tekintve, itt is tettem egy idegen kulcs megszorítást, mely a *kerdes\_valasz* néven fut, *ON DELETE NO ACTION, ON UPDATE CASCADE* tulajdonsággal. A hozzá tartozó oszlop a *kerdes\_id*, ami a *kerdes* tábla, *id* oszlopához tartozik.

Ilyen beállításokkal elkészült a *bioqiuz* nevű adatbázisom.

A harmadik lépésként a Visual Studio telepítését követően, magának az alkalmazás vázának a létrehozása következett.

Első lépésként létrehoztam 4 különböző Form-ot, amelyek az alkalmazás alapja: Nyito\_Form, Login\_Form, Regisztracio\_Form, Osszegzo\_Form.

Mindegyik Form-on található egy Menüsor, mely tartalmazza a következő füleket: Kezdőlap, Kategóriák, Súgó.

A *Form\_Login*

A *Form\_Nyito*-on található, 3 gomb: Belépés, ahol a felhasználó, a már előzetesen regisztrációt követően beléphet. Egy Regisztráció gombra kattintva, egy regisztációs laphoz jut a felhasználó, amit ha kitölt, az adatai tárolódnak az adatbázisban, a regisztrált felhasználónévvel és a hozzá tartozó jelszóval be tud majd lépni. Amennyiben regisztráció közben akar a felhasznéló kilépni, a porgram figyelmeztető ablakkal adja a tudtára, hogy adati el fognak veszni, amennyiben félbehagyja a regisztrációt. Röviden mindegyik gombhoz tartozik, egy Form. A Kilépés gomb, pedig magából az alkalmazából lép ki. Egy feugró ablak, kérdezi meg a felhasználótól, hogy Biztosan ki akar-e lépni.

A *Form\_Profil*

A *Form\_Quiz*

A *Form\_Reg*

A *Form\_Osszegzo*

A *Program.cs*-ben deklaráltam az egyes *Form-*kat: (kép)

Illetve, meghívtam őket: (kép)

Az adatbázis és az alkalmazás összekapcsolása volt a következő lépés, amit a következő képpen építettem ki: (kép) (*class AB*)

A konnektor létrehozásához a *Program.cs-*ben egy osztályt hoztam létre, melynek feladata a konnektor megnyitása, illetve lezárása. Visszatérési értékként magát a kapcsolatot hozza létre

A *Form\_Reg-*et úgy kellet létrehozni, hogy a felhasználó által kitöltött adatok,bekerüljenek az adatbázisba, és az el is tárolja őket.

A *Form\_belepes* felépítésében először is a belépési adatoknak meg kellett felelni, az regisztrációnál megadott belépési adatoknak. Amennyiben ez nem így volt, tehát a felhasználó által megadott belépési adatok nem egyeztek egy, az adatbázisban már meglevő adatokkal, úgy egy felugró ablak figyelmezteti a felhasználó.

4 darab osztály létrehozása is szükség volt, illetve azoknak értkek átadása, hogy a későbbbiekben

# **Összefoglalás**

A záródolgozatomban létrehozott alkalmazással az volt a célom, hogy egy önállóan létrehott programot készítsek a tanulmányaim során tanultak segítésgével.

# **Irodalomjegyzék**

[[1](#_C#)] Illés, Z. (2005). *Programozás C# nyelven.* Budapest: Jedlik Oktatási Stúdió.