

**ZÁRÓDOLGOZAT**

Készítették:

Fidrus Adrián – Nagy Gábor

Miskolc

2024.

Miskolci SZC Kandó Kálmán Informatikai Technikum

Miskolci Szakképzési Centrum

SZOFTVERFEJLESZTŐ- ÉS TESZTELŐ SZAK

Fórum

közösségi fórum

2024-2025

Tartalomjegyzék

[Bevezetés 1](#_Toc165919195)

[Programozási nyelvek és technológiák 2](#_Toc165919196)

[MySQL adatbázis (Database) 2](#_Toc165919197)

[Az adatbázis fogalma 2](#_Toc165919198)

[Mi az SQL? (Strukturált lekérdező nyelv) 3](#_Toc165919199)

[PHPMYADMIN és a XAMPP 4](#_Toc165919200)

[HTML 5](#_Toc165919201)

[CSS 6](#_Toc165919202)

[Bootstrap 7](#_Toc165919203)

[React (webes frontend) 8](#_Toc165919204)

[Hogy működik a React? 9](#_Toc165919205)

[Miért kedvelik a fejlesztők a Reactot? 9](#_Toc165919206)

[Dokumentum Objektum Model (DOM) 10](#_Toc165919207)

[Backend 12](#_Toc165919208)

[ASP.NET 13](#_Toc165919209)

[A backend feladata 13](#_Toc165919210)

[Trello 14](#_Toc165919211)

[A weboldal elképzelt váza: 15](#_Toc165919212)

[Felhasználói dokumentáció 16](#_Toc165919213)

[Főoldal 17](#_Toc165919214)

[Termékek 17](#_Toc165919215)

[Kosár 18](#_Toc165919216)

[Regisztráció 19](#_Toc165919217)

[Bejelentkezés 19](#_Toc165919218)

[Rendelés 20](#_Toc165919219)

[Tesztelés 20](#_Toc165919220)

[Források: 22](#_Toc165919221)

[Mellékletek I](#_Toc165919222)

# Bevezetés

A fórumok már évtizedek óta az internetes közösségi élet egyik alappillérét képezik, lehetővé téve a felhasználók számára, hogy különböző témákban megoszthassák véleményüket, kérdéseiket és tapasztalataikat. A mai világban, amikor az információ gyorsan áramlik, a fórumok továbbra is fontos szerepet játszanak az online kommunikációban, hiszen közvetlen kapcsolatot biztosítanak a hasonló érdeklődésű emberek között.

A fórum oldalunk célja, hogy egy olyan interaktív platformot biztosítson, ahol a felhasználók könnyedén csatlakozhatnak beszélgetésekhez, új témákat indíthatnak, és segítséget kérhetnek vagy adhatnak a közösségnek. Az oldal lehetőséget biztosít arra, hogy bárki megoszthassa tudását és véleményét, miközben egy támogató, barátságos közösség részévé válhat.

Ebben a dokumentációban bemutatjuk, hogyan épült fel fórumunk, milyen funkciókat kínál, és hogyan használható a platform a legnagyobb hatékonysággal. A célunk, hogy minden felhasználó számára könnyen navigálható és értékes élményt nyújtó fórumot hozzunk létre. Kívánunk kellemes böngészést és aktív részvételt a közösségi diskurzusban!

# Programozási nyelvek és technológiák

## **MySQL adatbázis (Database)**

MySQL adatbázis (Database)

## **Az adatbázis fogalma**

Az adatbázis strukturált információk vagy adatok szervezett gyűjteménye, amelyeket jellemzően elektronikusan, számítógépes rendszerben tárolnak. Az adatbázisokat általában adatbázis-kezelő rendszer (DBMS-az a rendszer mely összekapcsolja az adatbázist a végfelhasználóval és kezelni tudják az adatokat benne) vezérli. Az adatokat és az adatbázis-kezelő rendszert, valamint a hozzájuk kapcsolódó alkalmazásokat együttesen adatbázis-rendszernek nevezzük, amit gyakran csak „adatbázis”-ként rövidítünk.

A ma használatos leggyakoribb adatbázis-típusokban az adatok jellemzően táblák soraiba és oszlopaiba vannak rendezve, ami megkönnyíti az adatok feldolgozását és lekérdezését. Az adatok így könnyen elérhetők, kezelhetők, módosíthatók, frissíthetők, ellenőrizhetők és rendszerezhetők. A legtöbb adatbázis strukturált lekérdezőnyelvet (SQL) használ az adatok írására és lekérdezésére

## **Mi az SQL? (Strukturált lekérdező nyelv)**

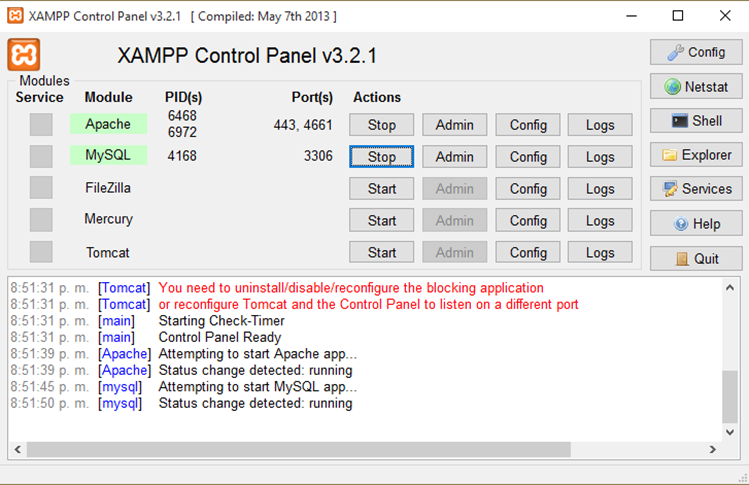
Az SQL egy programnyelv, amellyel műveleteket végezhet tényhalmazokkal és a közöttük fennálló kapcsolatokkal. A relációsadatbázis-kezelő programok, például az Access, az adatok kezelésére SQL nyelvet használnak. Számos programnyelvvel ellentétben az SQL könnyen olvasható és megérthető még a kezdő felhasználók számára is. Más programnyelvekhez hasonlóan az SQL is nemzetközi szabvány, amelyet az [ISO](https://www.iso.org/) és az [ANSI](https://www.ansi.org/) szabványozó testület is elismer.

Az SQL segítségével adathalmazokat írhat le, amelyekkel kérdéseket válaszolhat meg. Az SQL nyelv alkalmazásakor helyes szintaxist kell használni. A szintaxis az a szabálykészlet, amely biztosítja a nyelv elemeinek helyes összetételét. Az SQL szabályai az angol nyelv szabályain alapulnak, és a Visual Basic for Applications (VBA) nyelv számos szintaktikai elemét is tartalmazzák.

## **PHPMYADMIN és a XAMPP**

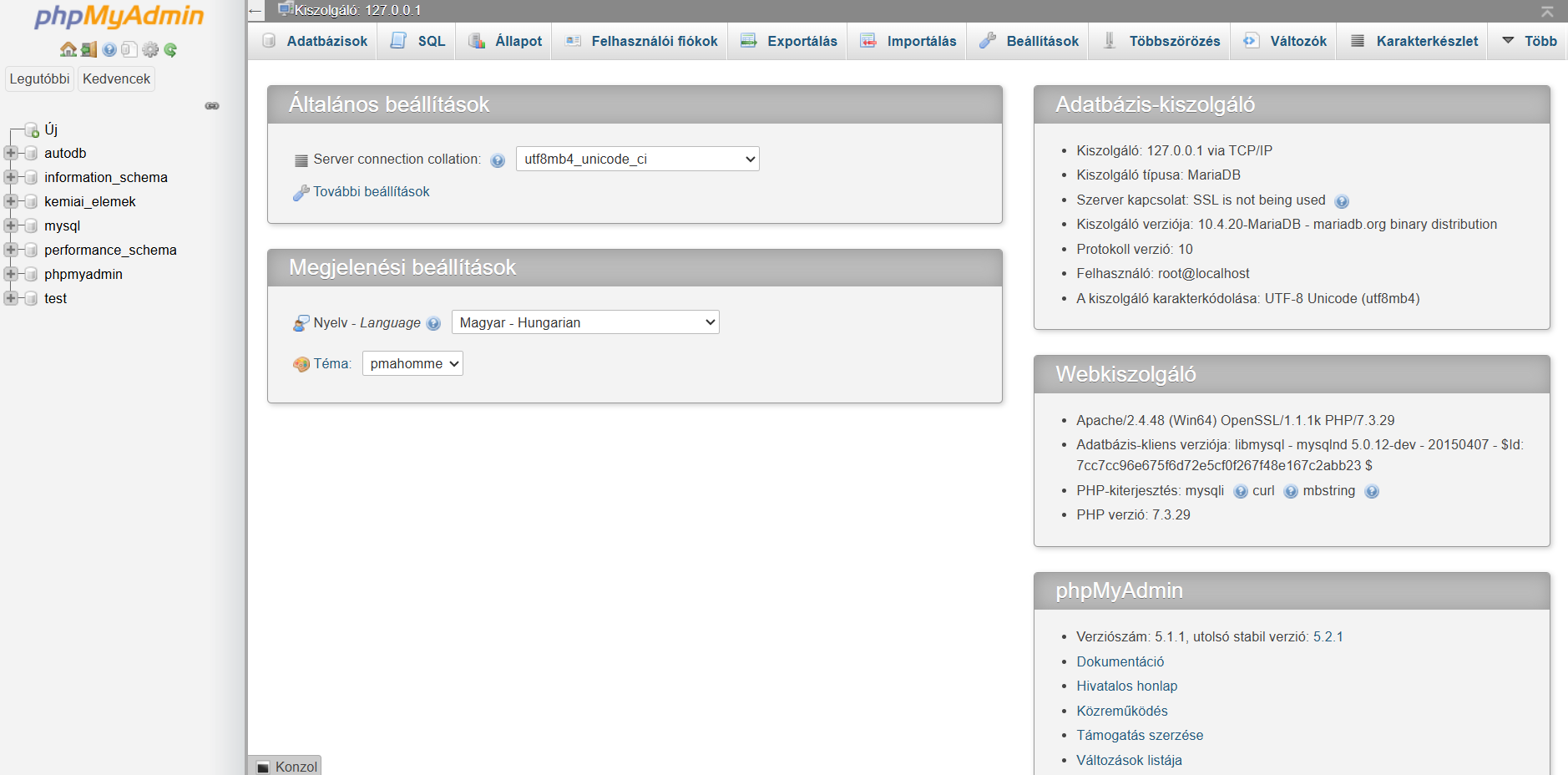
***Mi az a XAMPP?***

A XAMPP egy nyílt forráskódú, platformfüggetlen szoftvercsomag, amely tartalmazza az Apache webszervert, a MySQL adatbázist, valamint a PHP és Perl programozási nyelvek támogatását. A „X” a multiplatformot jelzi (Windows, Linux, Mac OS X), az „A” az Apache-ot, az „M” a MySQL-t, a „P” a PHP-t, és a másik „P” a Perl-t jelenti. A XAMPP célja, hogy könnyű és gyors megoldást nyújtson a fejlesztők számára, akik helyi környezetben szeretnék tesztelni és fejleszteni alkalmazásaikat.



***Mi az a PHPMYADMIN?***

A **phpMyAdmin** egy népszerű, ingyenes és [nyílt forráskódú](https://kiszervezettmarketing.hu/weboldal-keszites/forraskod/) eszköz, amit kifejezetten a **MySQL** és **MariaDB adatbázisok** webes kezelésére fejlesztettek ki. PHP programozási nyelven írták. Gyakorlatilag egy böngészőből elérhető felhasználói felületet biztosít, melyen keresztül könnyedén tudod adminisztrálni az adatbázisaidat.



## 

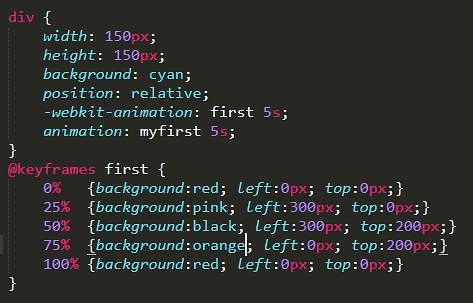
## **HTML**

A HTML (**H**yper**T**ext **M**arkup **L**anguage) egy leíró nyelv, amelyet weboldalak készítéséhez használunk. A HTML a W3C (World Wide Web Consortium) támogatásával internetes szabvánnyá vált.  
  
Egy HTML kód egyértelműen meghatározza a weboldal felépítését, kinézetét. Azaz a HTML nyelven megírt kódot a webböngésző értelmezi, majd a kód alapján megjeleníti a már általunk megszokott oldalakat.

## **CSS**

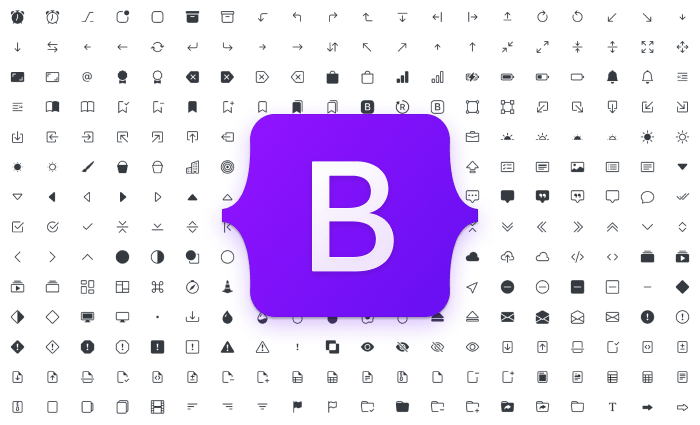
Ha ismerkedtél már a [weboldalfejlesztéssel, vagy olvastál a HTML-ről,](https://webiskola.hu/html-ismeretek/) akkor biztosan szembejött veled a CSS is. Hiszen a CSS az egyik legalapvetőbb nyelv a weboldal készítés során. Ebben a cikkben részletesen bemutatjuk, hogy mit jelent a CSS, mire lehet használni, illetve bemutatjuk azt is, hogy hogyan épül fel egy CSS kód. Egyszerűen fogalmazva bemutatjuk a CSS alapjait.

**A CSS-el formázhatjuk meg a weboldalunk megjelenését.** Például állíthatjuk be a szöveg méretét, kinézetét, színét, határozhatjuk meg, hogy egy oldalon belül hol helyezkedjen el egy kép, hogy hogy nézzen ki ez a kép, legyen-e kerete, le legyen-e kerekítve a sarka stb.



## **Bootstrap**

A [**Bootstrap**](https://getbootstrap.com/) egy nyílt forráskódú, ingyenes [kliens oldali keretrendszer](https://digikiad.gitbook.io/digitalis-kiadvanyok/fueggelek/kliens-oldali-keretrendszerek). A Twitter Inc. belső fejlesztésű rendszereként indult, 2011-ben nyilvánosan is elérhetővé tették. A jelenlegi 4-es verzió 2015-ben jelent meg. A Bootstrap számos előre elkészített felhasználói felület "darabkát" tartalmaz, úgymint navigációs menük, táblázatok, gombok, űrlapok, satöbbi ezek az úgynevezett komponensek. A komponensek jellemzően egy HTML kódrészletből, és a hozzá tartozó CSS stílus formázásból álnak, de egyes komponenseknek lehet javaScript része is.

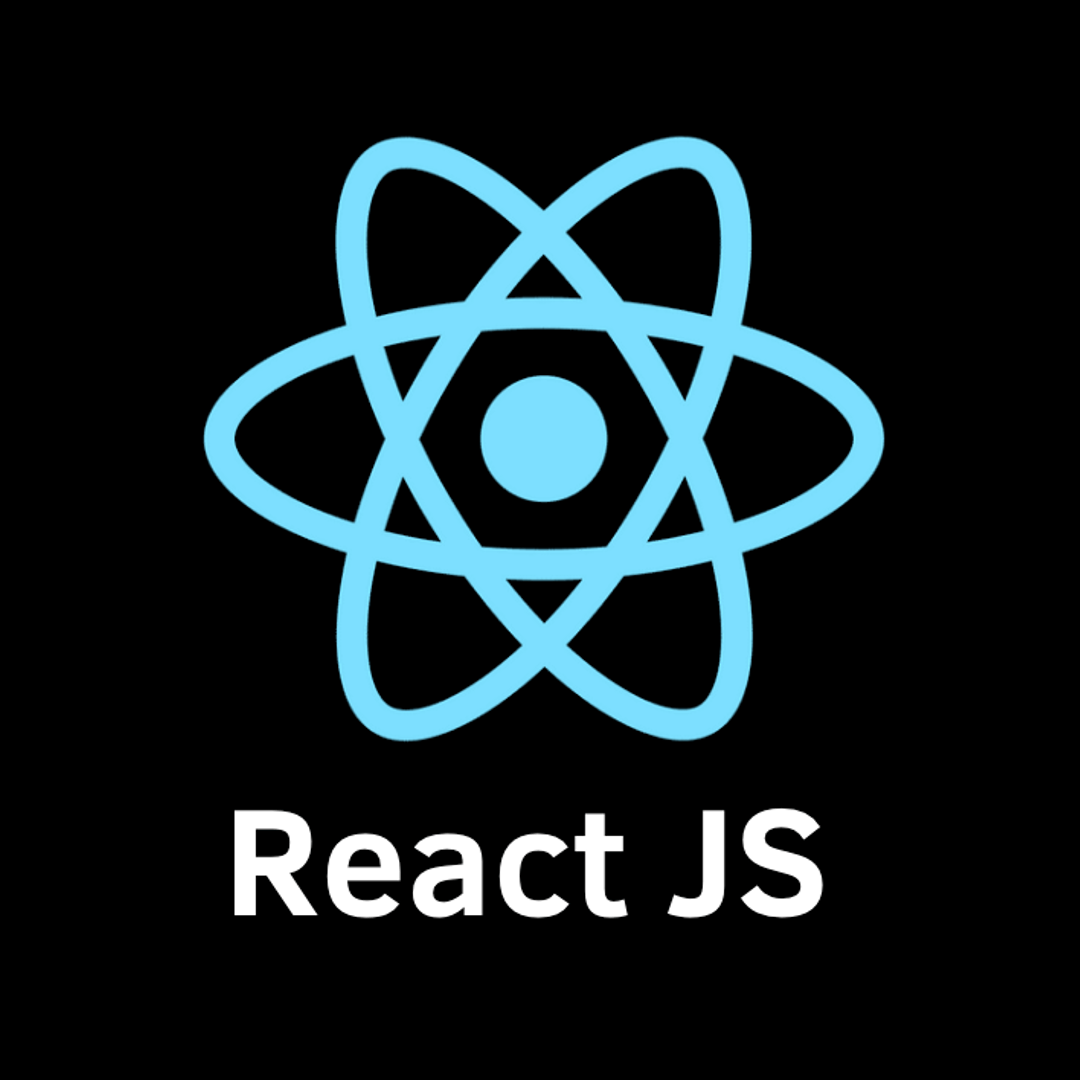


## **React (webes frontend)**

A React több lehetőséget nyit meg a szoftverfejlesztő szakemberek előtt, mint más keretrendszer. Egyszerűbbnek és kényelmesebbnek találják, mint az Angular-t. A könyvtár szinte teljes cselekvési szabadságot biztosítja.

A ReactJS egy TypeScript nyelven írt könyvtár. A [JavaScript](https://www.prooktatas.hu/hirek/minden-amit-tudnod-kell-javascriptrol-mielott-nekiallsz-megtanulni)modern verziójának szintaxisát használja, amelyet az ES6 és magasabb verziója ír le.

A React először a Facebook házon belüli projektje volt, és 2013-ban tették nyílt forráskódúvá. A ReactJS alkalmazása egy egyre növekvő görbét mutat, mivel egyre többen ismerik fel az előnyeit. Különböző startup-ok és már működő cégek is átveszik a technológiát. Ennek egyik oka, hogy a React egy olyan könyvtár, amely különösen segítségünkre van a one-page online alkalmazások front-end felületeinek összerakásában és létrehozásában, illetve kihasználja a virtuális DOM erejét.



## **Hogy működik a React?**

 A ReactJS segítségével épített alkalmazások az egyszeri újrafelhasználhatóság elvét alkalmazzák. Ez arra az elvre épül, hogy a weboldalak és alkalmazások komponensekből, egyirányú folyamattal épüljenek fel.  A Reactben létezik az állapotok és a megváltoztathatatlanság fogalma. A szülő és a gyermek komponensek tekintetében a komponensek szigorú hierarchiával rendelkeznek. A React esetében egy komponens egy meglévő kódrészlet, amely megírása a tiszta kód (Clean Code) elvén alapult.

## **Miért kedvelik a fejlesztők a Reactot?**

A React több lehetőséget nyit meg a szoftverfejlesztő szakemberek előtt, mint egy másik keretrendszer. Egyszerűbbnek és kényelmesebbnek találják, mint az Angular-t. A könyvtár szinte teljes cselekvési szabadságot biztosít, és kiváló minőségű felületet hoz létre bármilyen alkalmazáshoz.

* A React.js lehetővé teszi a könnyű fejlesztést és a kevésbé bonyolult elemekről a nehezebbekre való "áttérést".
* A React kód mind front-end, mind back-end komponenseknél alkalmazható - a könyvtár különböző eszközkészletekkel együtt használható robusztus, bonyolult alkalmazások építéséhez is.
* A React segítségével a fejlesztők egy-egy komponenst a többitől függetlenül módosíthatnak.
* Nincsenek követelmények. A React nem ír elő semmilyen követelményrendszert, így a fejlesztők teljes szabadságot kapnak a konkrét igényeknek megfelelő stack létrehozásában.
* Deklaratív programozás. Miután a fejlesztő beállítja a felületi elemek kinézetét meghatározó adatokat, a beállított paramétereket követve a könyvtár reagál az ezeken a komponenseken bekövetkező változásokra.
* Tanuld meg egyszer, programozz bárhova. A Facebook vállalati szlogenje utal arra, hogy a webes, mobil és cross-platform felhasználói felületek egyaránt használhatók ugyanazzal a React kóddal.

## **Dokumentum Objektum Model (DOM)**

A Dokumentum Objektum Modell (Document Object Model / DOM) egy platform - és nyelvfüggetlen standard objektummodell, amely a HTML, XHTML, XML, valamint rokon formátumaiknak a szerkezetét és az objektumaikkal történő interakciókat modellezi. A DOM egymással gyerek-szülő kapcsolatban álló objektumok rendszere. A dokumentum tartalmát, illetve a dokumentum valamennyi összetevőjét magában foglalja. A beépített objektumok kezelése böngészőnként eltérő lehet, továbbá plusz tulajdonságok is lehetnek különböző böngészők esetén.

Az 1994-ben alapított World Wide Web Consortium, amely nyílt szabványokat támogat a világháló számára, összehozta a Netscape Communications-t, a Microsoft-ot és más cégeket, hogy kifejlesszenek egy szabványt a webböngészőkben használatos szkript nyelvek számára, így alakult ki az „ECMAScript”. A szabvány első verziója 1997-ben látott napvilágot.

Az ECMAScript megjelenése után a W3C elkezdett dolgozni a DOM szabványosításán. A kezdeti eredmények voltak az ún. „DOM Level 1”, amelyet 1998 végén terjesztett elő a W3C. Ez a verzió már teljes modellt biztosított a teljes HTML és XML  dokumentum számára, beleértve a dokumentumok bármely részének megváltoztatásához szükséges eszközöket is.

2000 végén jelent meg a DOM Level 2. Bevezette a getElementById funkciót, valamint az eseménymodellt, és támogatta az XML névtereket és a CSS-t.

A DOM Level 3 (a DOM specifikáció jelenlegi kiadása) 2004 áprilisában jelent meg. Ez a verzió immár támogatja az XPath-t és a billentyűzet eseménykezelését, valamint egy interfészt a dokumentumok XML-lé szerializációjához.

2005-re a W3C-féle DOM nagy része jól támogatottá vált az átlagos ECMAScript támogatással rendelkező webböngészők, mint a Microsoft Internet Explorer 6, az Opera, a Safari, valamint a Gecko-alapú böngészők, mint a Mozilla Firefox, a SeaMonkey és a Camino körében.

Mivel a DOM mindkét irányú navigációt támogatja (szülő és az előző testvér felé is) és lehetővé teszi a változtatást tetszőleges helyen, az implementációban pufferelni kell a beolvasott dokumentumot (vagy annak értelmezett részét). Ebből következik, hogy a DOM leginkább olyan alkalmazások kezelésére alkalmas, ahol ismételt és nem szekvenciális sorrendű hozzáférés történik a dokumentumhoz.

A webböngészők rendszerint a DOM-hoz hasonló belső modellt használnak a dokumentumok (például egy HTML oldal) megjelenítéséhez. A DOM API-kat weboldalak, illetve egyes részeik JavaScript kódból történő vizsgálására, vagy módosítására is használják. Más szóval, a Dokumentum Objektum Modell az a mód, ahogyan a JavaScript látja a webböngésző állapotait és az általa megjelenített HTML oldalt.

Amikor egy böngésző megjelenít egy HTML oldalt, akkor legelőször betölti az oldalt a web-szerverről, majd értelmezi a jelöléseit (pl. HTML) és létrehozza a modellt a memóriában tárolt DOM formájában.

Minden dokumentum csomópontjai fa struktúrába vannak szervezve, ezt hívjuk DOM fának. A legfelső csomópont a DOM fában a Dokumentum (Document) objektum. Minden csomópont tetszőleges számú - nulla, vagy több - gyerekkel rendelkezhet. (Dom fa példa)

## **Backend**

**(Web API, ASP.NET CORE)**

**Az API jelentése és felhasználása**

Az API az angol Application Programming Interface rövidítést takarja, amit magyarra alkalmazásprogramozási felületnek fordíthatunk. Egyáltalán nem számít új találmánynak, hiszen már jóval a személyi számítógépek megjelenése előtt is létezett. Ekkor azonban még csak lokálisan, egy-egy szerveren látott el különféle feladatokat. Az internet elterjedésével vált az API igazán megkerülhetetlenné nagyjából a 2000-es évek közepetáján.



Ha szigorúan csak a definícióra vagyunk kíváncsiak, akkor a [Wikipediát hívjuk segítségül.](https://hu.wikipedia.org/wiki/Alkalmaz%C3%A1sprogramoz%C3%A1si_fel%C3%BClet) Eszerint az „alkalmazásprogramozási felület vagy alkalmazásprogramozási interfész egy program vagy operációs rendszer azon eljárásainak (szolgáltatásainak) és azok használatának dokumentációja – vagyis felsorolása és ismertetése egy szöveges dokumentumban –, amelyet más programok felhasználhatnak.”

Kicsit konyhanyelvebben megfogalmazva: az API teszi lehetővé, hogy a szoftverek egymás között kommunikáljanak, adatot cseréljenek, módosítsanak vagy előhívjanak egyes funkciókat. Fontos, hogy az API nem egy programozási nyelv, és nem is korlátozódik egyetlen nyelvre. Bármilyen programozási nyelven létrehozható és alkalmazható – és kismillió féle programban és appban használható.

Lássunk ezek alapján néhány tipikus példát, ahol API-ra van szükség:

* Mobilalkalmazások: ahhoz, hogy a telefonunkon futó alkalmazás használhassa a telefon funkcióit (mondjuk a kamerát vagy a híváslistát), API-t kell használni
* Közösségi média: Amikor bejelentkezünk egy honlapra a Facebook-, Google- vagy LinkedIn-fiókunkkal, egész biztosan az API áll a háttérben
* Webshopok: a webáruházak számos helyen alkalmazzák az API-t. Ez teszi lehetővé Simple vagy PayPal fizetést, de a készletkezelő és a webshop is így kommunikál egymással

Nézzük először fejlesztői szemszögből az API előnyeit. Nos, az alkalmazásprogramozási interfész egyszerűbbé teszi a munkájukat, és rengeteg időt megspórol nekik. Nem kell minden funkciót az alapoktól saját kezűleg megírniuk: egyszerűen csak használhatnak egy már elkészült API-t. Mondani sem kell, hogy ez jelentősen felgyorsítja a fejlesztési folyamatokat.

Az sem elhanyagolható, hogy az API által jobb felhasználói élményt biztosíthatnak. Lehet, hogy néhány fejlesztés nem kerülne be a programba, ha egyedileg kéne létrehozni, egy API-val azonban mindez nem jelent akadályt. Végül a fejlesztőknek könnyebbséget jelent, hogy az API platformfüggetlen, azaz bármilyen környezetbe, operációs rendszerbe integrálhatják.

Ami pedig a felhasználókat illeti, az API az ő életüket is jelentősen megkönnyíti, és ezt a cégünknél is kiaknázhatjuk. A felhasználók ugyanis nem szeretnek tucatnyi programot használni és megnyitogatni. Az API-val elérhetjük, hogy egy helyen, például a weboldalunkon érjenek el számos funkciót. Már említettük, hogy bejelentkezhetnek közösségi média fiókjukkal vagy fizethetnek külső alkalmazással.

Az API-val ezen túl arra is lehetőségünk nyílik, hogy valós raktárkészlet-információkat biztosítsunk az érdeklődőknek vagy éppen arra, hogy időpontot foglaljanak a Google Naptár segítségével – és a lefoglalt időpont azonnal bekerüljön a személyes naptárukba. Ha pedig egy ügyfél csetelni szeretne velünk, egy API támogatásával ezt megteheti az oldalunkon elérhető WhatsApp vagy Messenger használatával.

## **ASP.NET**

Az **ASP.NET** egy [nyílt forráskódú](https://hu.wikipedia.org/wiki/Ny%C3%ADlt_forr%C3%A1sk%C3%B3d%C3%BA_szoftver), [szerveroldali](https://hu.wikipedia.org/wiki/Szerver) [webalkalmazás](https://hu.wikipedia.org/wiki/Webalkalmaz%C3%A1s)-keretrendszer, amelyet webfejlesztésre terveztek dinamikus [weboldalak](https://hu.wikipedia.org/wiki/Weblap) előállítására. A [Microsoft](https://hu.wikipedia.org/wiki/Microsoft) fejlesztette ki, hogy lehetővé tegye a [programozóknak](https://hu.wikipedia.org/wiki/Szoftverfejleszt%C5%91) dinamikus webhelyek, alkalmazások és szolgáltatások létrehozását. A név az Active Server Pages Network Enabled Technologies rövidítése.

Először 2002 januárjában adták ki a .NET-keretrendszer 1.0-s verziójával, és a Microsoft Active Server Pages (ASP) technológiájának utódja. Az ASP.NET a Common Language Runtime-ra (CLR) épül, lehetővé téve a programozók számára, hogy ASP.NET kódot írjanak bármely támogatott .NET nyelv használatával. Az ASP.NET [SOAP](https://hu.wikipedia.org/wiki/SOAP) kiterjesztési keretrendszer lehetővé teszi az ASP.NET összetevők SOAP üzenetek feldolgozását.

Az ASP.NET utódja az ASP.NET Core. Ez az ASP.NET újbóli megvalósítása moduláris webes keretrendszerként, más keretrendszerekkel, például az Entity Frameworkkal együtt. Az új keretrendszer az új nyílt forráskódú .NET Compiler Platformot (kódnév: "Roslyn") használja, és több platformon működik. Az ASP.NET MVC, az ASP.NET Web API és az ASP.NET Web Pages (egy kizárólag Razor oldalakat használó platform) egységes MVC 6-ba egyesült.

## **A backend feladata**

## A backend fejlesztő egy olyan programozó, kinek elsődleges feladata az alkalmazás üzleti logika rétegének elkészítése az üzleti követelmények alapján. Ez gyakran kapcsolódik relációs adatbázisokhoz, ezért SQL tudással is rendelkezik. Backend fejlesztéshez erre alkalmas technológiákat, erre épülő keretrendszereket és könyvtárakat használ. Backend fejlesztésre különböző programozási nyelvek alkalmasak, ebből egyet vagy akár többet is ismer. Kapcsolatban van a frontend fejlesztővel, hiszen a backendhez a felhasználói felületet a frontend fejlesztő készíti el. Gyakran automatizált unit teszteket ír a munkája ellenőrzéséhez. Tisztában van a verziókezeléssel, frontend build folyamatokkal, a fejlesztési módszertanokkal, a fejlesztés menetével, és a continuous integration-nel. Munkáját képes dokumentálni. A bejelentett hibákat képes megkeresni és kijavítani. Az üzleti követelményeket sokszor az üzleti elemzőtől kapja, munkáját (manuális vagy automata) tesztelő ellenőrzi.

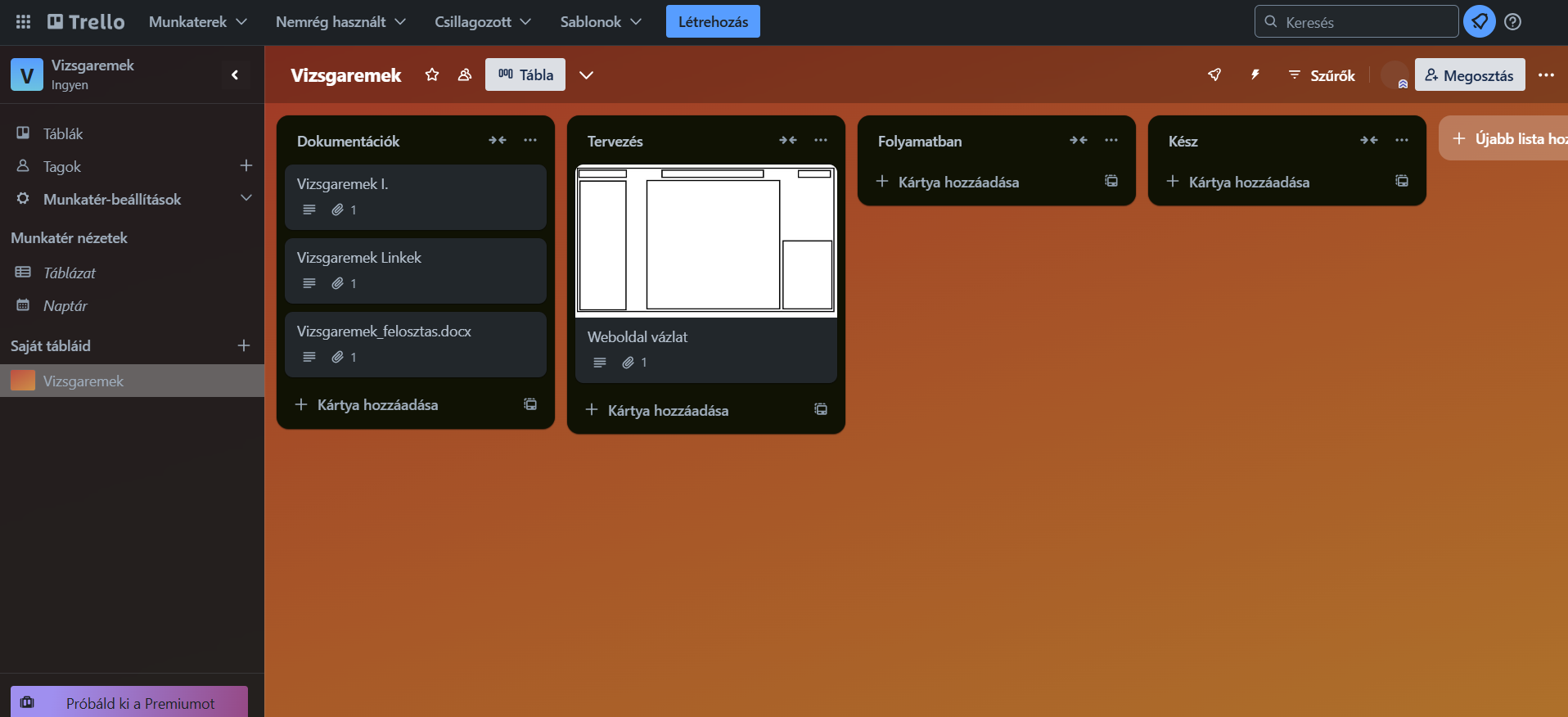
## **Trello**

Ahhoz, hogy minél egyszerűbb legyen a közös munkánk és a folyamatok nyomon követése, a Trello-t hívtuk segítségül.

Könnyedén használható felület, ahol munkateret hozhatunk létre, azon belül pedig táblákat, illetve listákat készíthetünk.

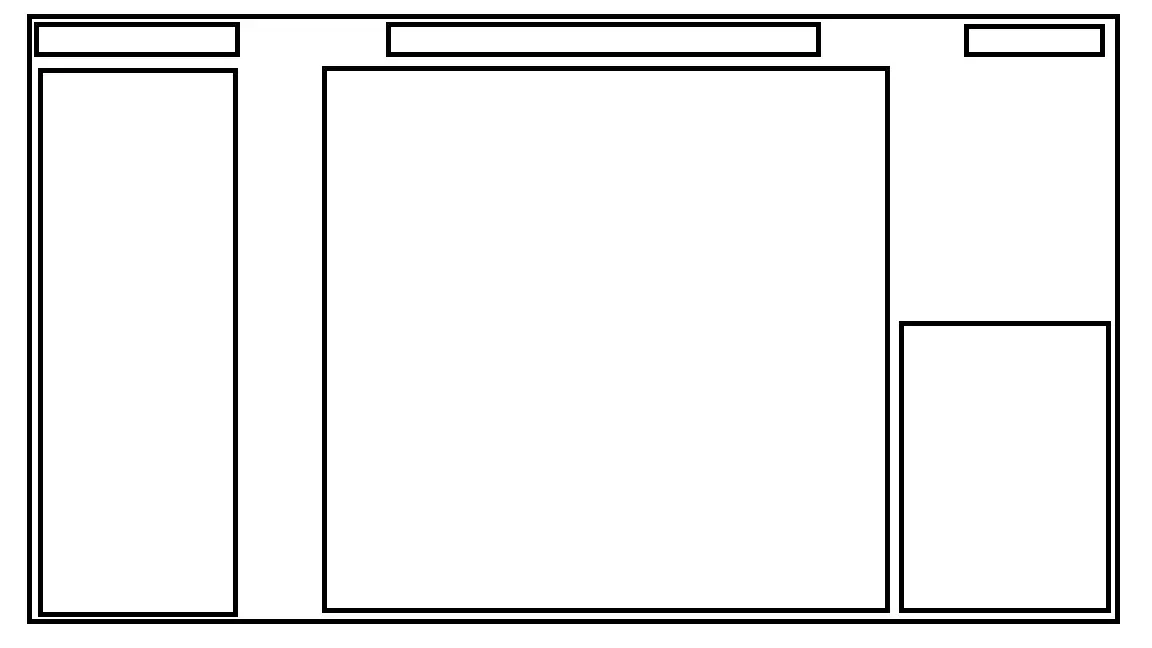
Ezzel a technológiával követtük nyomon magunkat, hogy hol tartunk, mi az, amit szeretnénk, mi az, ami épp megvalósítás alatt van, illetve mi az, amivel már készen vagyunk és fellélegezhetünk.

A felület bárkinek elérhető a Trello.com oldalról, viszont a saját munkatereket priváttá lehet tenni, és tudunk megadni tagokat, akik látják és tudnak benne szerkeszteni.

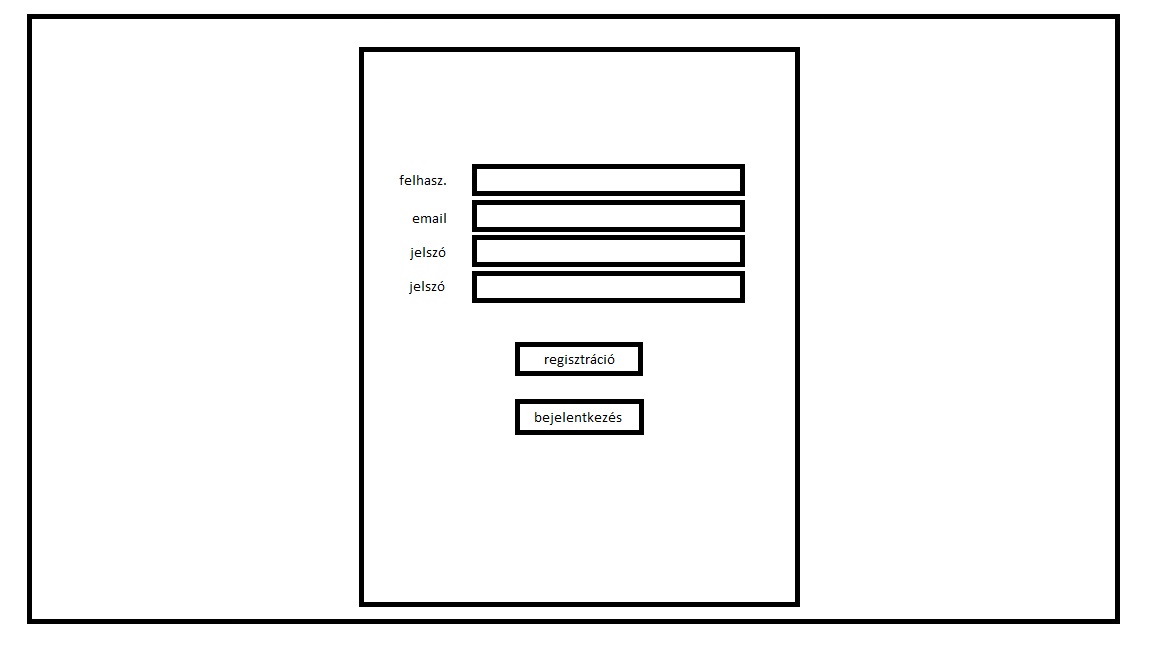


## **A weboldal elképzelt váza:**

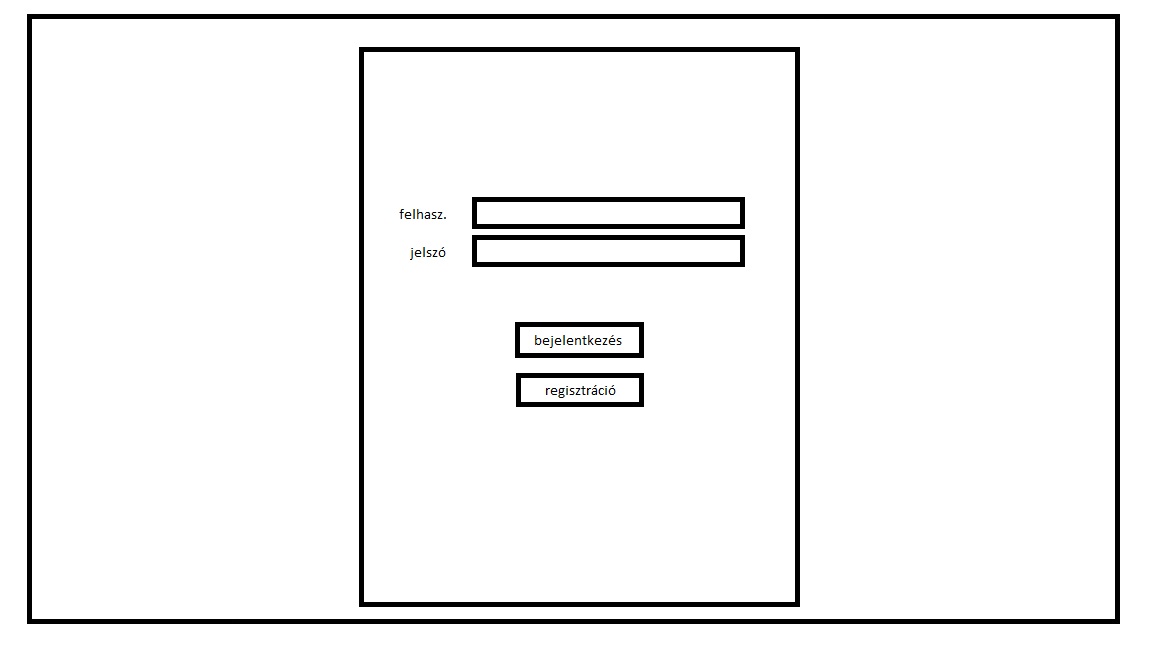
1. Főoldal



1. Regisztráció



1. Bejelentkezés



# **Felhasználói dokumentáció**

## 

## **Főoldal**

## **Regisztráció**

## **Bejelentkezés**

# **Források:**

<https://www.oracle.com/hu/database/what-is-database/> - Adatbázis fogalma - 2024. 12. 11.

<https://shorturl.at/BGHpy> - Mi az a SQL? - 2024. 12. 11.

<https://weblib.hu/a-xampp-hasznalatanak-elonyei/> - Mi az a XAMPP? – 2024. 12. 11.

<https://shorturl.at/KYarg> - Mi az a PHPMYADMIN? – 2024. 12. 11.

<https://webiskola.hu/css-ismeretek/mi-az-a-css-a-css-bemutatasa/> - CSS - 2024. 12. 11.

<https://shorturl.at/qOW1k> - Mi az a Bootstrap – 2024. 12. 11.

<https://shorturl.at/vnHHC> - React – 2024. 12. 11.

[https://hu.wikipedia.org/wiki/Document\_Object\_Model - DOM - 2024](https://hu.wikipedia.org/wiki/Document_Object_Model%20-%20DOM%20-%202024). 12. 11.

<https://www.minicrm.hu/blog/2023/07-06/mi-az-az-api/> - API – 2024.12.11

[https://hu.wikipedia.org/wiki/ASP.NET - ASP.NET-2024.12.11 - ASP.NET](https://hu.wikipedia.org/wiki/ASP.NET%20-%20ASP.NET-2024.12.11%20-%20ASP.NET) – 2024.12.11

<https://itmap.hu/pozicio/back-end-fejleszto-itmap-hu> - Backend feladata - 2024.12.11

# **Mellékletek**