** **

**ZÁRÓDOLGOZAT**

Készítették:

Oláh Márk - Szesztai Péter - Ambrus Kristóf

Konzulens:

Miskolc

2024.

Miskolci SZC Kandó Kálmán Informatikai Technikum

Miskolci Szakképzési Centrum

SZOFTVERFEJLESZTŐ- ÉS TESZTELŐ SZAK

Dokumentáció

Kandó Könyvtárnyilvántartó rendszer

Készítették:

Oláh Márk - Szesztai Péter - Ambrus Kristóf

Miskolc

2024-2025

Tartalomjegyzék

[1. Bevezetés 4](#_Toc164414894)

[2. Fejlesztői eszközök 5](#_Toc164414895)

[2.1. Adatbázis: 5](#_Toc164414896)

[2.1.1. WampServer: 5](#_Toc164414897)

[2.1.1.1. MySQL: 6](#_Toc164414898)

[2.2. BackEnd: 6](#_Toc164414899)

[2.2.1. ASP.NET 6](#_Toc164414900)

# Bevezetés

A könyvtárak az adat, továbbá a szakértelem dön͏t͏ő forrásai, a digitalizáció korában p͏e͏dig rendkívül fon͏tossá vált͏ ͏a haték͏ony, valamint ko͏rsze͏rű könyvtári nyilvántartási rendszerek ͏kialakítása͏.͏ Ez a dolgozat a mi általunk ki͏alakított könyvt͏á͏ri nyilvántart͏ási rends͏zerre ͏össz͏po͏ntosít, a napi kö͏nyvtári működés nehézségeinek ͏és követelményeinek megértése alapján.

Ebben a dolgozatban részletesen ismertetjük a szoftver létrehoz͏ásának processzus͏át, jellemzőit, továbbá előnyeit. Azt is megv͏itatjuk, hogy ez a pr͏ogra͏m hogyan javítja a könyvtári szolgáltatások hatékonyságát é͏s felhasználói͏ élményét.

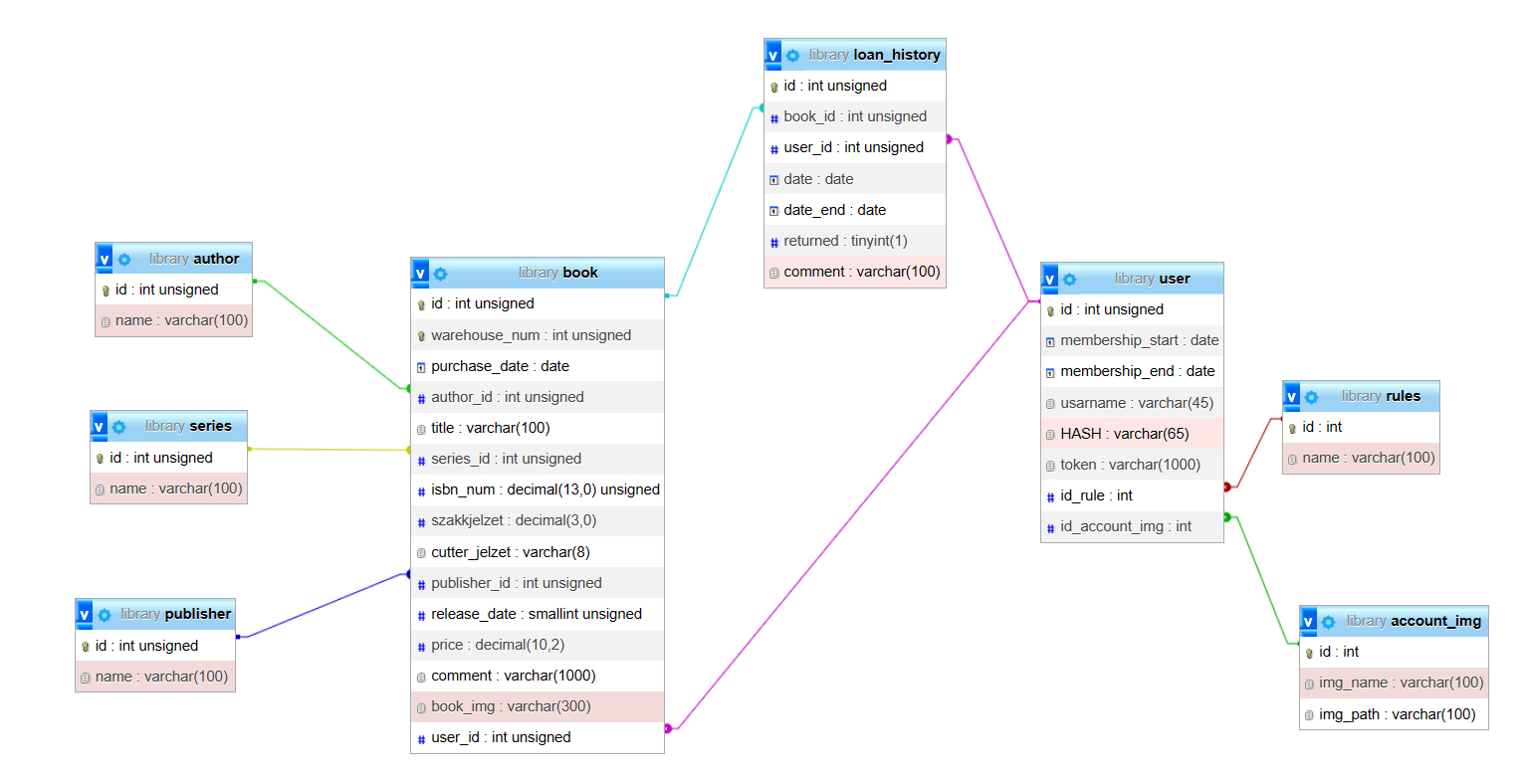
A dolgozatunk célja, hogy átfogó képet mutassunk arról, hogyan optimalizálhatók a könyvtárak a mai technológia segítségével, mik az előnyei, hátrányai és választ adunk arra is, hogy milyen kihívásokkal kellett szembesülnünk készítése közben.

Programunk az iskolánknak szeretne segíteni, hisz észrevettük, hogy iskolai könyvtárunknak még nincsen digitalizálva az adattárolása, egyben szeretne segíteni tanulótársainknak a könnyebb könyv választástásban is.

A project során tárolásra és verziókezelésre használtuk a [GitHub](https://github.com/ElitCovenant/osszes/tree/main)-ot, megbeszélésre a Discord-ot és Teamspeak-et, a feladatok számontartására pedig a [Trello](https://trello.com/b/cNCCjp6B/librarymanager)-t használtuk.

# Fejlesztői eszközök

# Adatbázis:

Adatbázisunk 8 relációból(táblából) áll. Látható a (1. Adatbázis\_szerkezet) képen. Egyik fő táblánk a **book** reláció, amely 15 mező-t tartalmaz, a másik fő táblánk az a **user**, aminek 8 mezője van. Ennek a két táblának vannak al-táblái, amelyek név szerint a: **rules**, **account\_img**, **loan\_history**, **publisher**, **series** és az **author**. Az utolsó három reláció az a book táblának az al-táblái.Az author tárolja az adott könyv íróját, a series tárolja a könyv szériáját, a publisher pedig a kiadókat tároljuk. A user-nek a rules al-táblájában tároljuk a felhasználók rangjait, az account\_img-be pedig az oldalunkon választható profil képeket tároljuk. A book és user táblának közös al-táblája pedig, a loan\_history. Itt tároljuk a felhasználók által kikölcsönzött könyveket.

1. Adatbázis szerkezet

# WampServer:

A WampServer egy Windows webfejlesztő platform dinamikus webes alkalmazásokhoz, amelyek Apache2 szervert, PHP szkriptnyelvet, MySQL adatbázist és MariaDB-t használnak. A PhpMyAdmin segítségével könnyebben kezelhetjük az adatbázisokat, valamint a MySQL Workbench GUI-t a MySQL és MariaDB szervereket.

Az évek során kipróbált localhost szerverkörnyezetek közül a Wampserver a legjobb. Könnyen használható, több lehetőség, könnyen konfigurálható. A legjobban kifort és a leghatékonyabb helyi webszerver szerintünk.

**Romain Bourdon** 2005-ben adta ki először a nagyvilágnak a programját. Azóta már nyílt forráskódú lett, és több fejlesztő is segítkezik a kódbázis fejlesztésében és karbantartásában. A WampServer a közösség által eléggé támogatott projekt, amely folyamatosan frissül és új funkciókkal bővűl.

Amit jó tudni, a WAMP-ról, hogy a LAMP-ból származik (az L a Linuxot jelenti). A kettő között az egyetlen különbség az, hogy a WAMP-ot Windowshoz, míg a LAMP-ot Linux alapú operációs rendszerekhez használják. A “WAMP” jelentése:

* „**W**” a Windows rövidítése, van még LAMP (Linux)-hoz és MAMP (Mac)-hez.
* „**A**” az Apache rövidítése. Az Apache a weblapok kiszolgálásáért felelős szerverszoftver. Amikor jön egy kérés felőled, hogy láss egy oldalt, az Apache HTTP-n keresztül teljesíti a kérést, és megmutatja a webhelyet.
* „**M**” a MySQL rövidítése. A MySQL feladata, hogy a szerver adatbázis-kezelő rendszere legyen. Tárolja az összes releváns információt, például a webhely tartalmát, felhasználói profiljait stb.
* „**P**” a PHP rövidítése. Ez az a programozási nyelv, amelyet a WordPress írásához használtak. Úgy működik, mint az egész szoftvercsomag ragasztója. A PHP az Apache-val együtt fut, és a MySQL-el kommunikál.

# MySQL:

A MySQL az Oracle által kifejlesztett relációs adatbázis-kezelő rendszer (**RDBMS**), amely strukturált lekérdezési nyelven (SQL) alapul. Az adatbázis az adatok strukturált gyűjteménye. A MySQL-t egy svéd cég, a MySQL AB hozta létre, amelyet a svédek David Axmark, Allan Larsson és a finn Michael "Monty" Widenius alapítottak. A MySQL első verziója 1995. május 23-án jelent meg. Ezt a Sun Microsystems 2008-ban vásárolta meg, és aztán az Oracle 2010-ben megvásárolta a Sun-t.

# BackEnd:

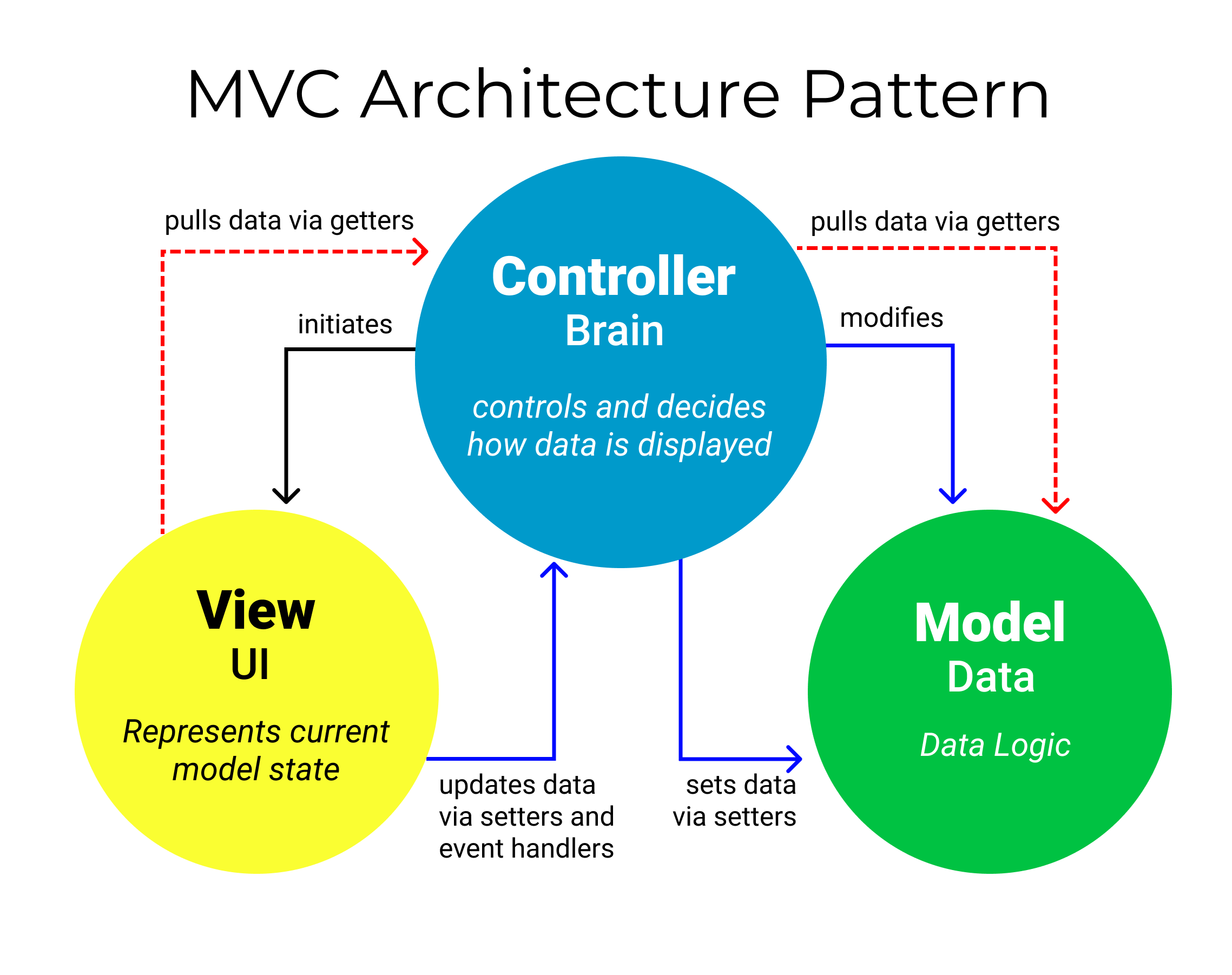
# ASP.NET

Az **ASP.NET** egy nyílt forráskódú webes keretrendszer, amelyet a Microsoft hozott létre modern webalkalmazások és -szolgáltatások a .NET segítségével történő létrehozására, az 1.0-s verzió pedig 2002-ben jelent meg, hogy a fejlesztők dinamikus webalkalmazásokat, szolgáltatásokat és webhelyeket készíthessenek. A keretrendszer a szabványos HTTP protokollal működik, amely az összes webalkalmazásban használt szabványos protokoll.

A .NET pedig egy olyan fejlesztői platform, amely széles eszköz kínálattal, sok programozási nyelv támogatással rendelkezik, és számos különböző típusú alkalmazás létrehozására alkalmas. Az alapplatform olyan komponenseket tartalmaz, amelyek több különböző típusú alkalmazás létrehozására alkalmasak. A további keretrendszerek, például az ASP.NET, kiterjesztik a .NET-et bizonyos típusú alkalmazások létrehozásához szükséges összetevőkkel.

Az ASP.NET az ASP (**Active Server Pages**) technológia utódja, és jelentős fejlesztést jelentett a rugalmasság és a teljesítmény tekintetében, kiterjeszti a .NET-et, így használható a csomagok és könyvtárak nagy ökoszisztémája, amely minden .NET-fejlesztő számára elérhető. Saját könyvtárak is létrehozhatóak, amelyek megoszthatók a .NET platformon írt alkalmazások között. A keretrendszer tartalmazza a .NET keretrendszer alapkönyvtárait, de a gyakori webes minták könyvtárait is.

Az egyik ilyen könyvtár a **Model View Controller** (MVC), amely lehetővé teszi az MVC tervezési minta használatát webalkalmazások és webhelyek tervezéséhez, amelyet a felhasználói felület (nézet, vagy angolul view), az adatok (modell) és az alkalmazáslogika (vezérlő, vagy angolul controller) szétválasztására használnak. A webhelyekhez használt MVC-mintát használva a kérések egy vezérlőhöz kerülnek továbbításra, amely felelős a Modellel való együttműködésért a műveletek végrehajtása és/vagy az adatok lekéréséért.

A Vezérlő kiválasztja a megjeleníteni kívánt nézetet, és ellátja a modellel. A Nézet a végső oldalt jeleníti meg a Modell adatai alapján. Megkönnyíti a “tiszta” modell osztályok létrehozását, és leegyszerűsíti a kapcsolódást az adatbázissal. Deklaratív módon határozza meg az érvényesítési szabályokat a C# attribútumok használatával, amelyeket a kliensen és a kiszolgálón alkalmaznak.  


Az ASP.NET továbbra is támogatott és frissített, ezért a legújabb fejlesztési eszközökkel lehet dolgozni benne, ami az egyik oka volt, hogy ezt a keretrendszert használjuk, a másik pedig, hogy az asztali alkalmazás programozási nyelve a C# volt, és mivel a keretrendszer támogatja azt, egyszerűbb megoldás volt egy programnyelvet használni a két projektrészhez, mint megtanulni egy teljesen újat.

# Web Api

Az API (application programming interface) vagy alkalmazásprogramozási felület olyan szabályok vagy protokollok összessége, amelyek lehetővé teszik a szoftveralkalmazások számára, hogy kommunikáljanak egymással az adatok, szolgáltatások és funkciók cseréje érdekében. Az API-k leegyszerűsítik és felgyorsítják az alkalmazás- és szoftverfejlesztést azáltal, hogy lehetővé teszik a fejlesztők számára, hogy más alkalmazások adatait, szolgáltatásait integrálják, ahelyett, hogy a semmiből fejlesztenék azokat.

Az API-ok az alkalmazástulajdonosok számára is egyszerű és biztonságos módot kínálnak arra, hogy alkalmazásadataikat és funkcióikat elérhetővé tegyék szervezetük részlegei számára. Az alkalmazástulajdonosok adatokat és funkciókat is megoszthatnak vagy forgalmazhatnak üzleti partnerekkel vagy harmadik felekkel.

Az API-k csak a szükséges információk megosztását teszik lehetővé, más belső rendszerrészleteket rejtve, amik segítik a rendszer biztonságában tartását. A kiszolgálóknak vagy eszközöknek nem kell teljes mértékben felfedniük az adatokat – az API-k lehetővé teszik kis adatcsomagok megosztását az adott kérés szempontjából.

Manapság a legtöbb API már WEB API. A webes API-ok a Remote API egy fajtája (ami azt jelenti, hogy az API protokollokat használ a külső erőforrások manipulálására), amelyek az alkalmazás adatait és funkcióit teszik közzé az interneten keresztül.

# Entity Framework

Az Entity Framework egy nyílt forráskódú ORM-keretrendszer a Microsoft által támogatott .NET-alkalmazásokhoz. Lehetővé teszi a fejlesztők számára, hogy tartományspecifikus osztályok objektumai segítségével dolgozzanak az adatokkal anélkül, hogy a mögöttes adatbázistáblákra és oszlopokra kellene összpontosítaniuk, ahol ezeket az adatokat tárolják. Az Entity Framework segítségével a fejlesztők magasabb absztrakciós szinten dolgozhatnak, amikor adatokkal foglalkoznak, és adatorientált alkalmazásokat hozhatnak létre és karbantarthatnak kevesebb kóddal a hagyományos alkalmazásokhoz képest.

Az Entity Framework API (EF6 és EF Core) magában foglalja a tartomány (entitás) osztályok adatbázissémához való leképezését, a LINQ-lekérdezések SQL-be ​​való lefordítását és végrehajtását, az entitásokon az élettartamuk során bekövetkezett változások nyomon követését és a változtatások mentését az adatbázisban.

