# Fejlesztői dokumentáció

## Felhasznált technológiák

### PHP

A PHP (Hypertext Preprocessor) egy szkriptnyelv, amelyet tulajdonképpen webes alkalmazások és dinamikus weboldalak fejlesztésére használnak. A 90-es évek óta fejlesztik, a php-s kódot a webszerver futtatja le, tehát a klienstől érkező kéréseket dolgozza fel. Különböző keretrendszerek (Laravel, Symfony) és adatbázis-motorokkal (MySQL, MariaDB) kombinálhatjuk, amelyek megkönnyebíthetik a webes alkalmazások kialakítását. A nyelv támogatja az objektumorientált programozást, a szerveroldali adatbázis-kezelést, a fájlkezelést, a formafeldolgozást, a hálózati kommunikációt és még sok más funkciót. A PHP-t két fő módon használhatjuk: beágyazott módban és önállóan futó szkriptként. A beágyazott módban a PHP kódot a HTML vagy más szöveges dokumentumokba illeszthetjük, és az kiszámítja és generálja az adott oldal tartalmát. Az önállóan futó szkriptek segítségével pedig parancssorban vagy cron feladatként futtathatunk PHP-t, például adatok feldolgozására vagy rendszerműveletekre. Tehát elmondhatjuk, hogy a PHP egy sokoldalú és hatékony szerveroldali szkriptnyelv, amely lehetővé teszi a dinamikus weboldalak és webalkalmazások fejlesztését. A nyelv hatalmas felhasználói bázissal rendelkezik, és a webes fejlesztés egyik alapvető eszköze marad a jövőben is.

### Laravel

A laravel egy PHP alapú webes keretrendszer, amely segíthet a webfejlesztőknek a modern weboldalak kialakításához. k fejlesztését, a hatékonyság és a termelékenység növelése érdekében.

A Laravel számos funkcióval rendelkezik, amelyek megkönnyítik a webalkalmazások építését és karbantartását. Néhány fő jellemzője a következők:

**Eloquent ORM:** A Laravel beépített objektum-relációs leképzést (ORM) biztosít, amely lehetővé teszi az adatbázisokkal való könnyű kommunikációt. Az Eloquent ORM segítségével objektumokat és modelleket használhatunk az adatbázisok kezelésére, ami egyszerűsíti az adatmanipulációt és a lekérdezések írását.

**Routing:** A Laravel rendelkezik egy kényelmes routing rendszerrel, amely segít a webes útvonalak kezelésében. Ez lehetővé teszi, hogy könnyen definiálhassuk az alkalmazás útvonalait, és hozzárendeljük hozzájuk a megfelelő vezérlőket és műveleteket.

**Sablonok (Blade):** A Laravel saját sablonmotorral rendelkezik, amely Blade néven ismert. A Blade segítségével könnyedén készíthetünk dinamikus és újrafelhasználható sablonokat. A sablonmotor támogatja a feltételes logikát, ciklusokat és más funkciókat, amelyek lehetővé teszik a kódban való könnyebb szervezést és a szép, olvasható kimenet előállítását.

**Middleware-k**: A Laravel middleware-k segítségével kezelhetjük a HTTP kérések közötti műveleteket. Ez lehetővé teszi az autentikáció, az engedélyezés, a naplózás és egyéb műveletek egyszerű beállítását a kérések előtt vagy után.

**Adatbázis migrációk:** A Laravel migrációkat használ az adatbázis séma verziókezeléséhez és kezeléséhez. A migrációk lehetővé teszik az adatbázis változtatásainak verziókövetését, és könnyen visszaállíthatóvá vagy átvihetővé teszik azokat.

**Csatornák és események:** A Laravel lehetővé teszi az alkalmazásban történő valós idejű kommunikációt a csatornák és az események segítségével. Ez lehetővé teszi az élő frissítések küldését a kliensek felé, például az élő chat funkciókhoz vagy valós idejű értesítésekhez.

A Laravel keretrendszer általában nagy hangsúlyt helyez a kódolási konvenciókra, az egységtesztelésre és az olvasható, karbantartható kód írására. Emellett a Laravel széles körben támogatott és élénk közösséggel rendelkezik, amely segítséget nyújt a fejlesztőknek a dokumentáció, fórumok és közösségi hálózatok révén. Megemlíthető, hogy támogatja az MVC(Model-View Controller) modelt, a választott projektünk is ezt az architektúrát követi, a laravel projektstruktúra kialakításának köszönhetően könnyedén ki tudunk igazodni az MVC egyes részein. Tehát elmondhatjuk, hogy a laravel egy optimális keretrendszer a mai webalkalmazások elkészítéséhez.

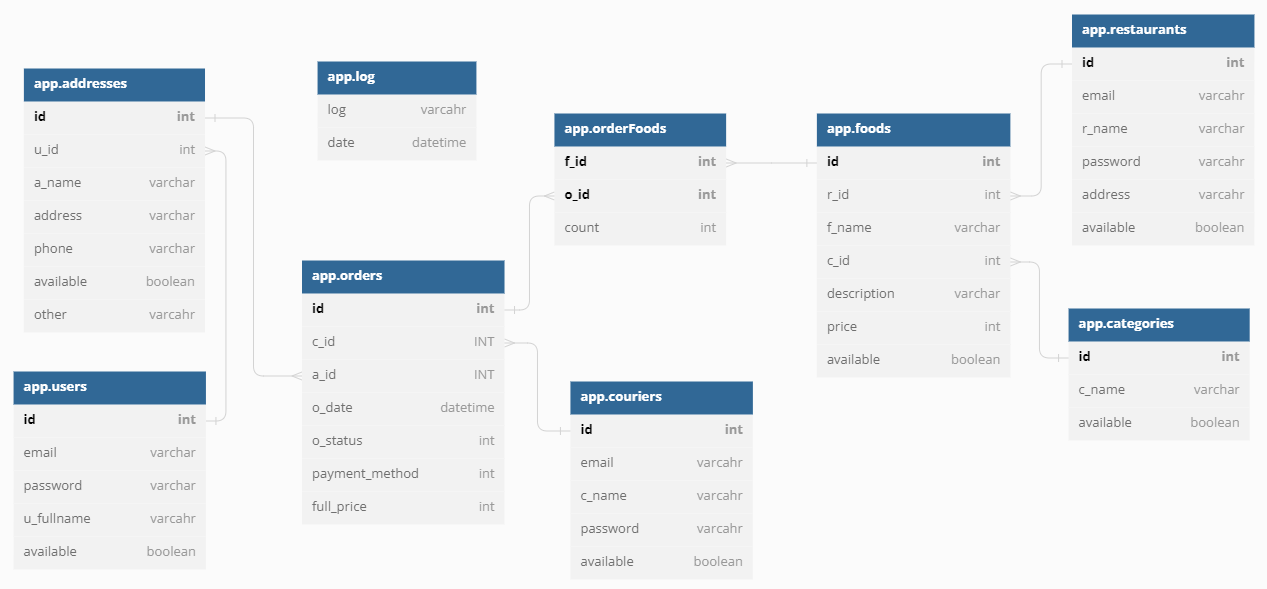
WebPUSH:[Webpush-php](https://github.com/web-push-libs/web-push-php) egy olyan php kiegészítő amit lehet használni push értesítések küldésére. A web-push protokollt használja, ami 2016 óta van szabványosítva.  
Nagyon konfigurálható, szinte mindenre képes bizonyos keretek között.  
Csak HTTPS kommunikációval vagy localhost-on működik a biztonság szempontjából.

### HTML

A HTML, avagy Hyper Text Markup Language egy weboldalak készítésére szolgáló leíró nyelv. A HTML jelenleg legfrissebb verziója a HTML 5. A HTML önmagában nem alkalmas komplex weboldalak létrehozására azon tulajdonsága miatt, hogy csak a weboldal megjelenéséért felel. A weboldalak formázásáért a CSS felelős, míg a kliens oldalon futó funkciókat jellemzően JavaScript segítségével oldhatjuk meg. Weboldalak esetén a szerveroldali funkciók kiszolgálására napjainkban már rengeteg nyelv biztosít lehetőséget, de a legismertebbek talán a PHP, a C# (ASP.NET keretrendszer), Java (Spring keretrendszer), Python, Node JS.

### MySql

A MySql az Oracle Corporation által fejlesztett SQL alapú relációs adatbázis szerver. A MySql mára az egyik legelterjedtebb adatbázis kiszolgáló olyan okok miatt, mint például a nyílt forráskód, a megbízhatóság, a stabilitás, valamint az, hogy platform független, tehát elérhető többek között Windows, Linux, MacOS X, FreeBSD, és szinte minden további operációs rendszeren. Az adatbázis kezeléséhez szükségünk van továbbá egy adatbázis kezelő szoftverre, mint például a MySql Workbench. MySql adatbázis kezelő szoftverek is szinte minden operációs rendszerre léteznek, még Androidra, vagy IOS-re is. Alternatívájaként érdemes megemlíteni az MSSQL-t, amely hasonló hatékonysággal képes ellátni az adatbázis feladatokat.



### JavaScript, JQuery

A JavaScript napjaink egyik legelterjedtebb weboldalakon használt script nyelve. Fontos tulajdonsága, hogy kliens oldalon fut le a kód, melyet a böngésző futtat, azaz a felhasználó által is kiolvasható. Ezen tulajdonságának betudható, hogy komplex alkalmazás fejlesztésére önmagában nem alkalmas, mivel nem futtatható szerver oldalon. Az elnevezése kisség megtévesztő lehet, mivel bár a szintaktikája kissé hasonló a Java-hoz, de semmilyen kapcsolatban nincs a két nyelv. A jQuery egy JavaScript könyvtár, melynek népszerűsége betudható annak, hogy gyorsan, egyszerűen használható, és rengeteg hasznos funkciót, eseménykezelést tartalmaz, melyeket így már JavaScriptben nem kell nekünk elkészíteni. Rengeteg publikus, más programozók által készített jQuery kód érhető el az interneten, melyekkel például bemeneti mezőket, és egyéb, a felhasználó számára látható elemeket tehetünk látványosabbá, használhatóbbá. A jQuery használatához script-ként be kell importálnunk azt az oldalunkra, viszont ez alapból elérhető az ASP.NET projektekben, így ezzel már nem kell foglalkoznunk. Le is tölthetjük, valamint hivatkozhatunk online elérhető verziójára is. Az online verzió mellett szól az, hogy naprakész, viszont hátránya lehet a rendelkezésreállás. További előnye lehet, hogy igény esetén Ajax segítsége mellett egyszerűen használhatjuk API-val.

### CSS, Bootstap

A CSS, vagy teljes nevén Cascading Style Sheets egy stílusleíró nyelv, mely főleg weboldalak esetén használatos. Célja, hogy a weboldalunk tartalmát és kinézetét elválassza egymástól. A CSS segítségével több módon is lehetőségünk van az oldalunk kinézetének csoportosítására. Hivatkozhatunk egy tag azonosítójára (id), nevére (name), vagy akár típusára is. Továbbá lehetőség van egy adott elem közvetlen, beágyazott stílus módosítására is, ám ez nem ajánlott, ugyanis sokkal olvashatatlanabb kódot eredményez, és nem utolsó sorban nem tudjuk több helyen is használni. A .NET keretrendszer által létrehozott projektekhez automatikusan társul a Bootstrap aktuális verziója. A Bootstrap egy nyílt forráskódú, ingyenes CSS keretrendszer. Alternatíva híján napjainkban a legelterjedtebb front-end keretrendszer. Első körben weboldalak kinézeteinek kialakításához használatos. Napról-napra növekszik a Bootstrap alapú sablonok száma, melyeket web designerek készítenek. Található ezek között ingyenes, valamint fizetős verzió is, de lehetőségünk nyílik saját sablon létrehozására is. Továbbá jelentős, hogy könnyen személyre szabható, és átlátható, erre azonban nem minden esetben van szükség, mivel alapból található benne a legtöbb webfejlesztés során használatos elemre (Például gombok, formok, beviteli mezők) megfelelő kinézet. Ezen kinézeteket leggyakrabban úgy használhatjuk fel, hogy az adott elemre alkalmazzuk a számunkra szükséges CSS osztályt. A Bootstrap használata mellett a legnagyobb érv a reszponzivitás. Reszponzív oldalról akkor beszélhetünk, amikor az azt megjelenítő kijelző méretétől, felbontásától, arányaitól függetlenül az alkalmazás úgy jelenik meg, hogy minden eleme megfelelően látszódjon, valamint olvasható és használható maradjon. Az előzőleg említett tulajdonság az egyik legfontosabb elvárás napjainkban a weboldalaknál, mivel a felhasználók egyre változatosabb eszközöket használnak változatos kijelző specifikációkkal kezdve a nagyfelbontású monitoroktól, az átlagos/nagyfelbontású mobiltelefonokon, okostelevíziókon, táblagépeken keresztül az alacsony felbontású okosórákig.

### GitHub

A GitHub a Microsoft egy leányvállalataként a verziókezelés, verziókövetés lehetőségét biztosítja. A diplomamunkám elkészítése közben számomra fontos volt, hogy szükség esetén az otthoni számítógépemen kívül máshonnan is elérhessem, fejleszthessem a kódomat. Ezen felül lehetőséget biztosít számunkra, hogy egy esetleges hibás fejlesztést követően visszaállítsuk a kódot egy régebbi verziójára, és nem utolsó sorban egy esetleges adatvesztés következtében is vissza tudom nyerni a diplomamunkámat. Segítségével nyomon követhetem, hogy miként fejlődik a program, mikor milyen fejlesztéseket, újításokat építek bele. Csoportos munkák esetén további létfontosságú funkciók is megnyílnak előttünk. Lehetőséget biztosít arra, hogy az általunk létrehozott projektet munkatársaink is elérhessék, megtekinthessék, módosíthassák jogosultságuktól függően. Ebben az esetben gyakran külön branch-et, úgynevezett munkaasztalt kapnak a fejlesztők, és a saját módosításaikat oda töltik fel, míg a fő munkaasztalt csak arra külön kijelölt személyek érik el, ők egyesítik a fejlesztők munkáit. Ez a lehetőség nagyvállalati környezetben elengedhetetlen. Bár legfőképpen forráskódjainkat tároljuk itt, de lehetőség van például dokumentáció, Wikipédia, integrációs könyvtárak (LIB-ek), és még rengeteg virtuális adat tárolására, nyomon követésére. A GitHub továbbá lehetőséget nyújt számunkra kisebb weboldalak tárolására, valamint kódjaink publikálására, és még megannyi hasznos funkcióra, melyeket szinte a végtelenségekig lehetne sorolni. Alternatívaként rengeteg lehetőséget felsorolhatnék, de talán a legismertebb az azonos alapokra épülő GitLab, valamint rengeteg cég mára már saját Git alapú verziókövető rendszert üzemeltet, és használ.

### MVC

A Model-View-Controller, vagy rövidítve MVC napjaink egyik legnépszerűbb, és leghasználgatóbb (jellemzően webes) program tervezési mintája. Célja, hogy az alkalmazásunkat három fő komponensre bontjuk, melyek a modell, a nézet (view), és a vezérlő (controller). Legfőbb előnyei többek között a párhuzamos fejlesztési lehetőség, vagy éppen a modularitás. Az egyes rétegek feladatai:

* Modell: Jellemzően az alkalmazás által használt adatok strukturált (osztályokba szervezett) tárolásáért, eléréséért felel.

Esetünkben az alkalmazás tartalmaz Adatbázis modelleket (DBModels), melyek az adott modellhez tartozó adatbázis-tábla szerkezetét írja le, és ennek segítségével Entity Frameworkön keresztül könnyen összekapcsolhatjuk az alkalmazásunkat és az adatbázisunkat.

Ezen túl az alkalmazás tartalmaz Request, és Response modelleket, melyek a Nézet- és vezérlő rétegek közötti rendszerezett adatcserét biztosítja.

* Nézet: A nézet réteg felel az alkalmazásunk megjelenéséért, valamint a felhasználóval történő „kapcsolattartásért”. A nézeten jelenítjük meg az adatokat a felhasználó számára, valamint ezen keresztül küld kéréseket a felhasználó a Vezérlő rétegnek (Pl.: egy gomb megnyomásának hatására adatokat küldünk a szervernek).

Vezérlő: Mint az a nevéből is adódik, az alkalmazásunk vezérléséért felel. Ez a (webes alkalmazások esetén szerveroldali) réteg felel azért, hogy feldolgozza a nézet rétegtől kapott (felhasználó által kiváltott) kéréseket, valamint a kérésekre valamilyen válasszal reagáljon. Ez a válasz webes alkalmazás esetén lehet egy elérési út (Nézet) visszaadása, vagy adatok visszaadása a megjelenítő számára, esetleg valamilyen hibakód.