Veszprémi Szakképzési Centrum Ipari Szakgimnáziuma

**Szakképzés neve: Szoftverfejlesztő**

**OKJ száma: 54 213 05**

Szakdolgozat

Hungarian United Army blog

*Gyurik Pál* *Sokhegyi Amadé*

témavezető 13.C

Veszprém, 2020, április 1

# Tartalom

[Tartalom 2](#_Toc36640120)

[Bevezetés 4](#_Toc36640121)

[Felhasználói dokumentáció 5](#_Toc36640122)

[Általános specifikáció 5](#_Toc36640123)

[Rendszerkövetelmények 5](#_Toc36640124)

[XAMPP szoftvercsomag telepítése és az adatbázis létrehozása 5](#_Toc36640125)

[A weboldal megnyitása 8](#_Toc36640126)

[A webalkalmazás használatának részletes leírása 8](#_Toc36640127)

[Cikkek megtekintése 9](#_Toc36640128)

[A jelentkezés folyamata 10](#_Toc36640129)

[Az admin felület 12](#_Toc36640130)

[Jelentkezők listája 13](#_Toc36640131)

[Taglista 15](#_Toc36640132)

[Tag hozzáadása 16](#_Toc36640133)

[Cikkek listája 17](#_Toc36640134)

[Cikk írása 18](#_Toc36640135)

[Tankok listája 19](#_Toc36640136)

[Repülők listája 20](#_Toc36640137)

[Új jármű felvitele 21](#_Toc36640138)

[Bejelentkezés 21](#_Toc36640139)

[Fejlesztői dokumentáció 24](#_Toc36640140)

[Témaválasztás indoklása 24](#_Toc36640141)

[Nyelvek, fejlesztői eszközök 24](#_Toc36640142)

[HTML 24](#_Toc36640143)

[CSS 24](#_Toc36640144)

[Bootstrap 24](#_Toc36640145)

[Apache 25](#_Toc36640146)

[PHP 25](#_Toc36640147)

[MySQL 25](#_Toc36640148)

[MariaDB 25](#_Toc36640149)

[Visual 25](#_Toc36640150)

[XAMPP 25](#_Toc36640151)

[JavaScript 26](#_Toc36640152)

[AJAX 26](#_Toc36640153)

[JQuery 26](#_Toc36640154)

[Rich Text-Editor 26](#_Toc36640155)

[Adatbázis 26](#_Toc36640156)

[Az adatbázis táblái táblázatos formában 28](#_Toc36640157)

[Jelentkezo tábla 29](#_Toc36640158)

[Jelentkezobirtokol tábla 29](#_Toc36640159)

[Jarmu tábla 29](#_Toc36640160)

[Felhasznalo tábla 29](#_Toc36640161)

[Cikk tábla 30](#_Toc36640162)

[Algoritmusok 31](#_Toc36640163)

[Táblázatba helyezés 31](#_Toc36640164)

[Törlés 32](#_Toc36640165)

[Tagfelvitel 33](#_Toc36640166)

[A mező(k) ürítése 35](#_Toc36640167)

[A cikkek főoldalra helyezése és működő linkek 35](#_Toc36640168)

[Bejelentkezés 38](#_Toc36640169)

[.htaccess használata 39](#_Toc36640170)

[Jelszótárolás 39](#_Toc36640171)

[Tesztelés dokumentálása 40](#_Toc36640172)

[Fejlesztési lehetőségek 40](#_Toc36640173)

[Összefoglaló 41](#_Toc36640174)

[Ábrajegyzék 42](#_Toc36640175)

[Irodalomjegyzék 44](#_Toc36640176)

# Bevezetés

Szakdolgozatom témája egy internetes játékbeli század (klán, játékosközösség) blog-szerű weboldala, illetve egy játékon kívüli jelentkezési felületet biztosító webes alkalmazás.

Az internetes játék neve War Thunder. A War Thunder egy free-to-play (ingyenesen játszható), de prémium előfizetés lehetőségét megadó MMO harci játék, amelyet a Gaijin Entertainment fejleszt és jelentetett meg Microsoft Windows, Mac OS X, PlayStation 4 és Xbox One platformokra. A játék lehetőséget ad arra, hogy a játékos különböző harci járműveket próbáljon ki egészen az 1930-as évektől kezdődően a ’90es évekig bezárólag.

A War Thunderben megtalálható egy kissé kompetitívebb részleg – a kláncsata vagy századcsata. Ehhez a felhasználónak csatlakozni kell egy századhoz, amelyet bizonyos feltételek mellett tehet meg (pl.: minimum szintű jármű, minimum tapasztalat, csaták száma). A mi századunk a HUA – Hungarian United Army. Magyar tagokkal rendelkezünk, de van néhány határon túli társunk is.

A klánban 1 éve vagyok jelen, és a vezetőknek felajánlottam, hogy csinálok egy webes felületet, mert a Facebook oldal mellett más platformon nem vagyunk jelen. Az oldal csak egy adott rétegnek nyújt szolgáltatást, hisz akit nem érdekel a játék, az nem fogja látogatni.

Az alkalmazás elsősorban egy közösségi platform, ahol a látogatókkal klánnal kapcsolatos információkat közlünk, a járműnyilvántartó csak a szakasz tagjai által elérhető. Lehetőség van a klánba való jelentkezésre is, melyet ugyan meg lehet tenni a játékon belül is, de a weboldalon kicsit több információhoz juthatunk a jelentkezővel kapcsolatban.

# Felhasználói dokumentáció

## Általános specifikáció

A webalkalmazásom célja a század számára egy médiumforrás biztosítása. Cikkeket írhatnak az arra jogosultak, amelyek kikerülnek a főoldalra, és a látogatók el tudják olvasni. Az alkalmazás csak egy kisebb közösség számára készült, hisz a WarThunder magyar vonatkoztatásban nem számít híres játéknak.

## Rendszerkövetelmények

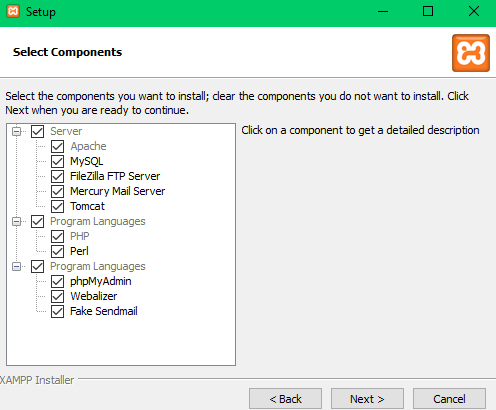
Hardver követelmények:

* Háttértár: 600MB szabad hely
* Memória: 1GB RAM

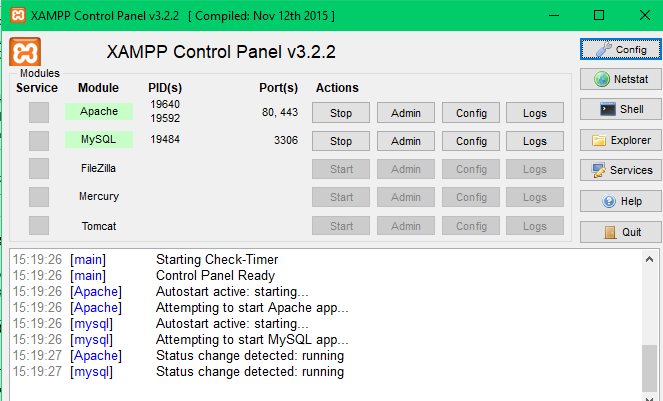
Szoftver követelmények:

* Operációs rendszer
* Böngésző
* XAMPP szoftvercsomag

## XAMPP szoftvercsomag telepítése és az adatbázis létrehozása

1. Első lépésként töltsük le az alkalmazást a [megadott oldalról](https://www.apachefriends.org/download.html). Ajánlott a legfrissebb verziót letölteni (jelenleg 7.4.3 – 2020.03.29.).
2. Futtassuk a letöltött installer.exe-t
3. 

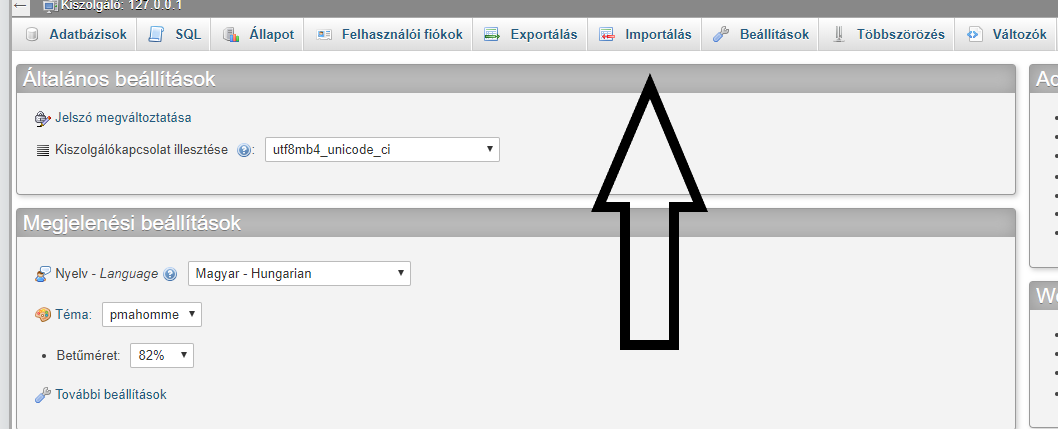
1. ábra XAMPP telepítő felület

1. Válasszuk ki hova szeretnénk telepíteni. Az alapértelmezett telepítési útvonal (C:\xampp\) megfelelő.
2. Telepítés után ha futtatjuk az alkalmazást, ez a képernyő fogad minket:
3. 

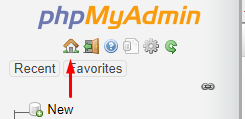
2. ábra XAMPP Kontrol ablak

1. Kattintsunk rá a MySQL sorban az Admin gombra
2. 

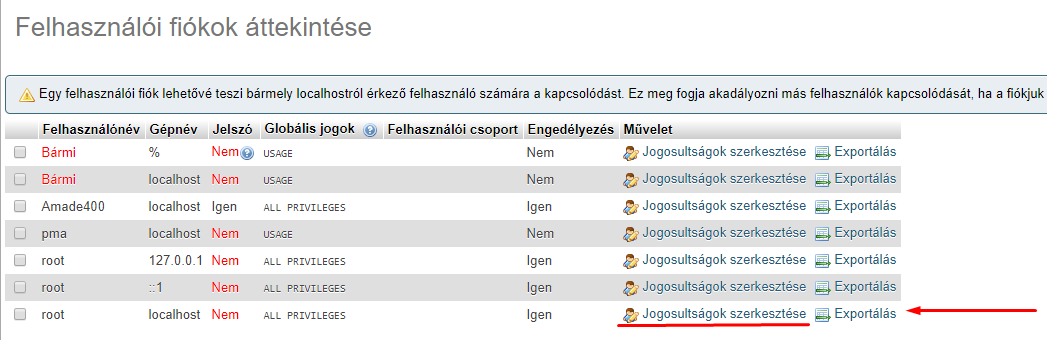
3. ábra localhost/phpmyadmin bejelentkező felülete

1. Jelentkezzünk be, alapértelmezetten nincs a rootnak jelszava.
2. Kattintsunk az importálás fülre, és a fájl kiválasztása gombra.
3. 

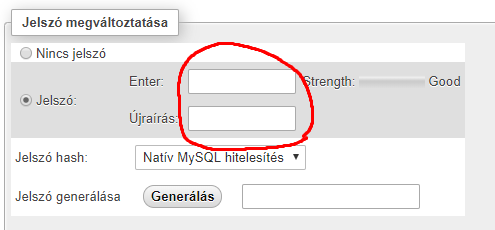
4. ábra XAMPP felület 1

1. Válasszuk ki az adatbázis mappából a huateam.sql fájlt és kattintsunk az indítás gombra az oldal alján.
2. Sikeresen felraktuk az adatbázisunkat a MySQL szerverre.
3. Most állítsuk be a root jelszavát.
4. Kattintsunk a ház ikonra a bal oldali menüsor tetején
5. 

5. ábra XAMPP felület 2

1. Kattintsunk a felhasználói fiókok fülre.
2. Keressük meg a 6. ábrán látható sort és kattintsunk a Jogosultságok szerkesztése gombra
3. 

6. ábra XAMPP felhasználói fiókok felület

1. Fenti menüsorban kattintsunk a Jelszó megváltoztatása gombra
2. A mezőkbe írjuk be a „rootroot” kifejezést
3. 

7. ábra XAMPP felhasználói felület

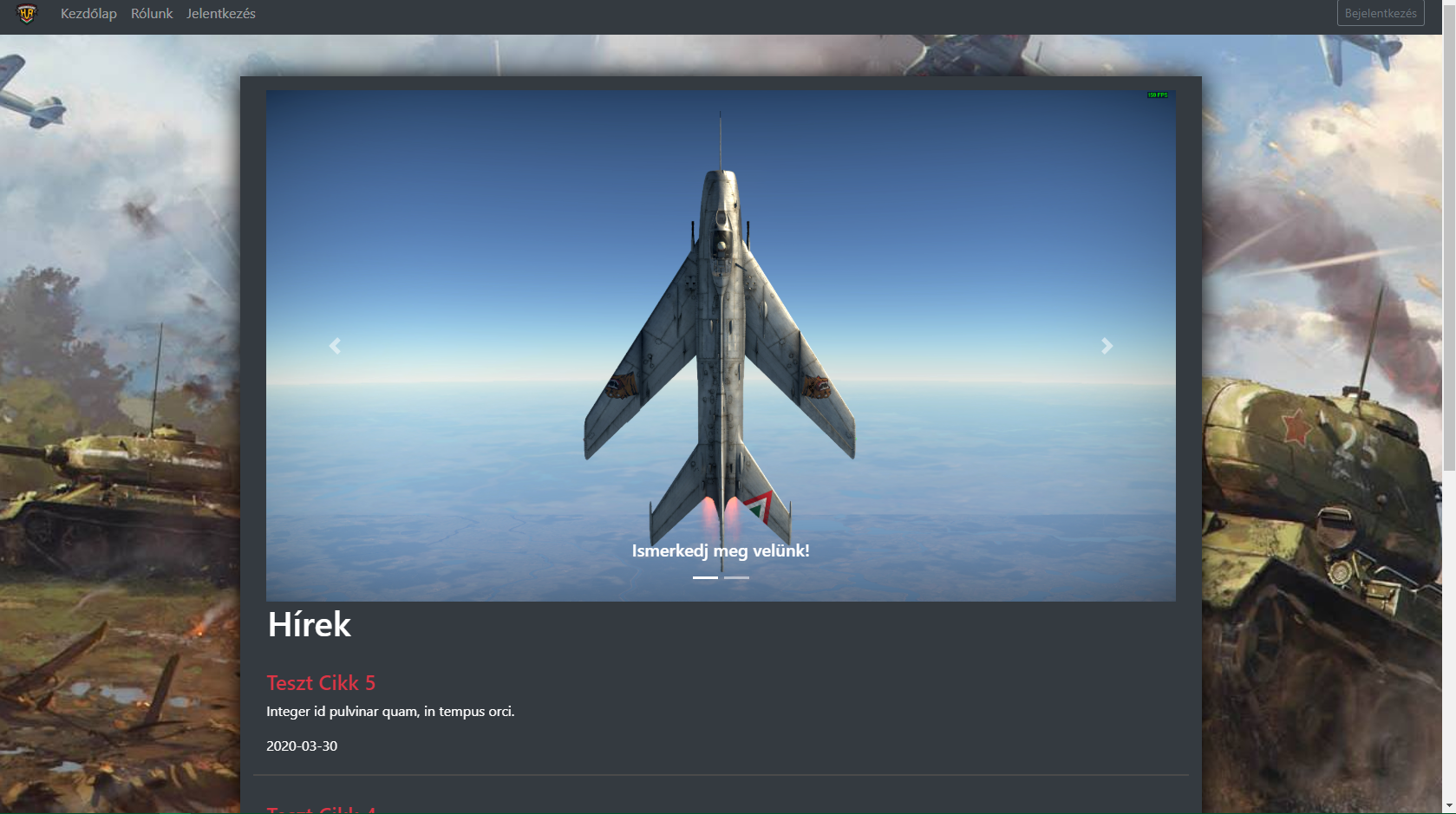
1. Jobb oldalunk kattintsunk az Indítás gombra. A program visszadob a bejelentkezési felületre (3. ábra). A jelszóhoz írjuk be a rootroot jelszót, és ha sikeresen be tudunk jelentkezni, akkor a jelszóváltoztatás megtörtént.

## A weboldal megnyitása

1. A weboldal mappát másoljuk át a \htdocs mappába (ezt a telepített útvonalon lesz, alapértelmezetten C:\xampp\htdocs.
2. Nyissunk meg egy böngészőt. A címsorba gépeljük be a következőt: localhost/weboldal
3. Ezzel a weboldal főoldalára kerültünk.

## A webalkalmazás használatának részletes leírása

A webalkalmazás fő oldalának tetején látható a navigációs sáv, a következő pontokkal: Kezdőlap (ahol most vagyunk), Rólunk, Jelentkezés és Bejelentkezés. Ezek elérhetőek a látogatónak, kivéve a bejelentkezés, mert az csak a század tagjainak érhető el, és a funkciója nem a hozzászólások írása a közleményeink alá, hanem a század tagjainak nyilvántartása.



. ábra index.php

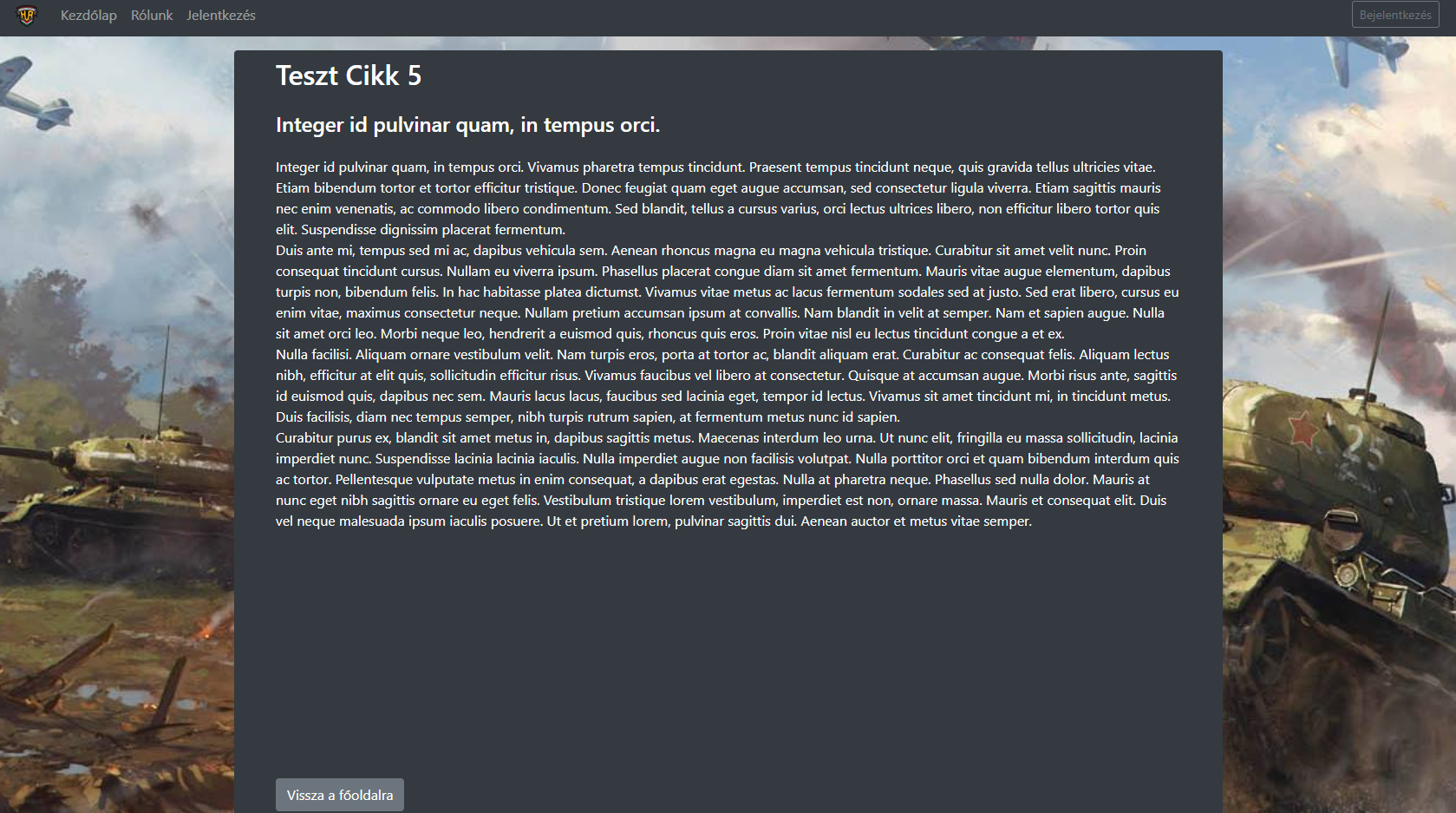
A rólunk oldalon egy rövid bemutatkozót olvashat a látogató a klánnal kapcsolatban. A jelentkezés oldalon a látogatónak jelentkezésre van lehetősége.



. ábra rolunk.php tartalma

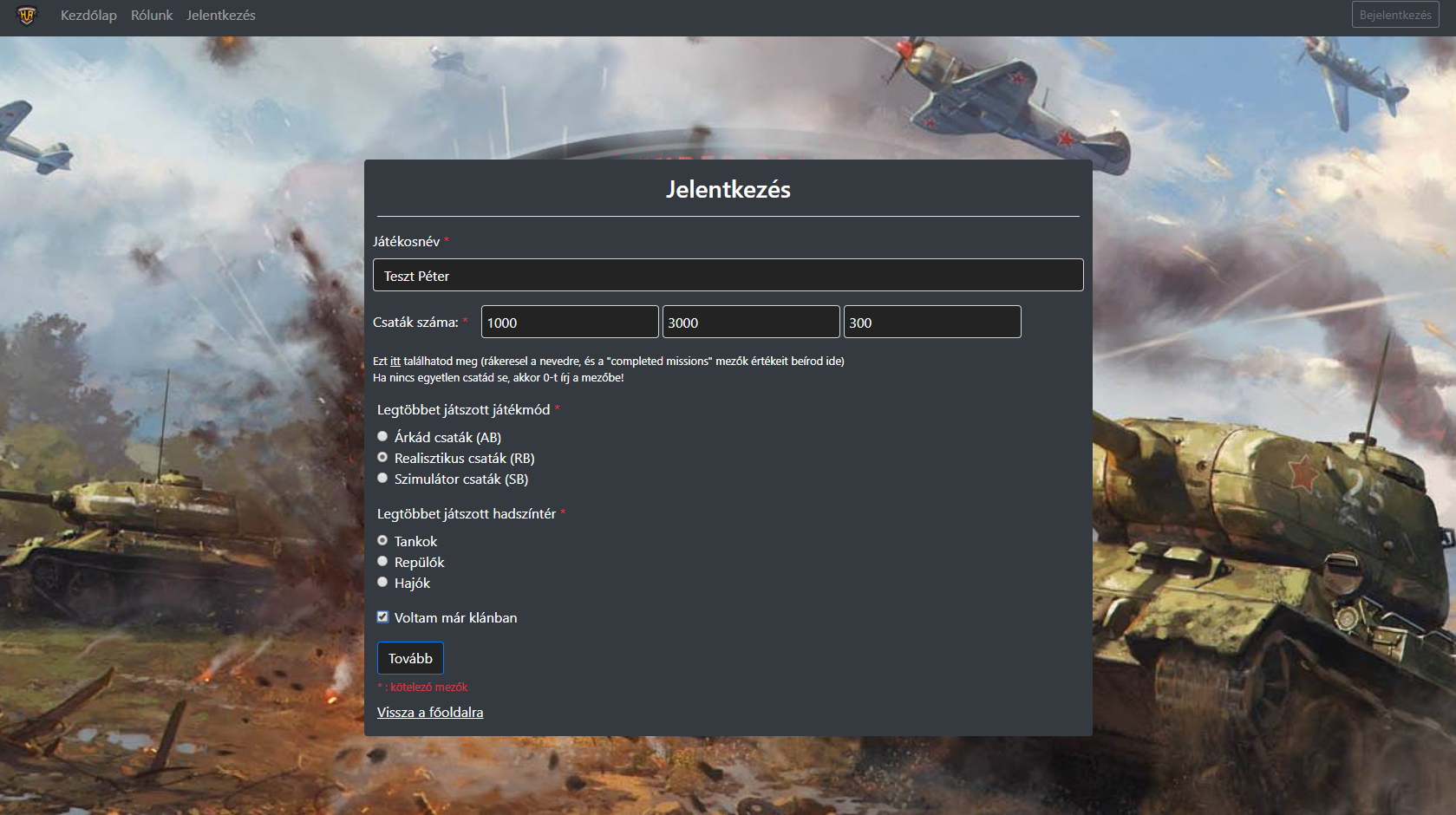
### Cikkek megtekintése

A cikk címére kattintva megnyílik a cikk. A cikk címe, alatta a rövid leírás, ezután pedig a szöveg. A cikk végén van egy gomb, amivel visszamehetünk a főoldalra.



. ábra cikk.php

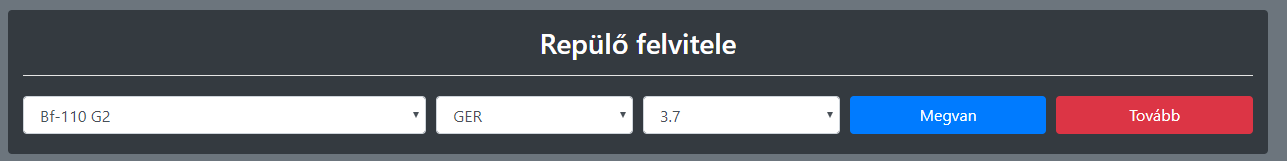
### A jelentkezés folyamata



. ábra jelentkezes.php

A látogatónak meg kell adnia a játékosnevét, a csatáinak számát, a mostanában legjátszottabb játékmódját, a hadszínteret, amelyen szívesen játszik, és azt, hogy volt-e már században. Ezekre azért van szükség, hogy a felvételizőről egy előzetes képet kapjunk, hogy mennyire tapasztalt a játékban.

Ha kitöltötte az összes mezőt, a tovább gombra kattintással átirányítja a repülőgépek felvitele oldalra.



. ábra jelentkezorepcsik.php

Itt a játékos megadhatja, milyen repülőkkel rendelkezik, ezeknek megvan-e az összes modifikációja, illetve a repülő legénységének szintje mennyi. Ha egy repülője sincs, a Tovább gombra kattintva tovább léphet, ekkor egy üzenetben az oldal megkérdezi, szeretne-e továbblépni. Ha egy repülőt felvisz a játékos, a tovább gombra már nem lehet kattintani.



. ábra jelentkezorepcsik.php felvitt repülőkkel

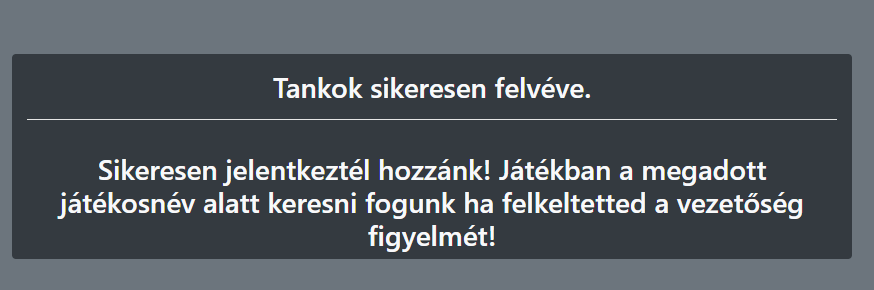
A felvitel gombra kattintva átirányítja a tankok felvitele oldalra, amelynek a lényege ugyan ez, csak tankokkal, nem repülőkkel.



. ábra jelentkezotankok.php felvitt tankokkal

Fontos megjegyezni, hogy az itt megadott járműveknél sokkal több van a játékban – ezeket a járműveket a mi századunk várja el a játékosoktól.

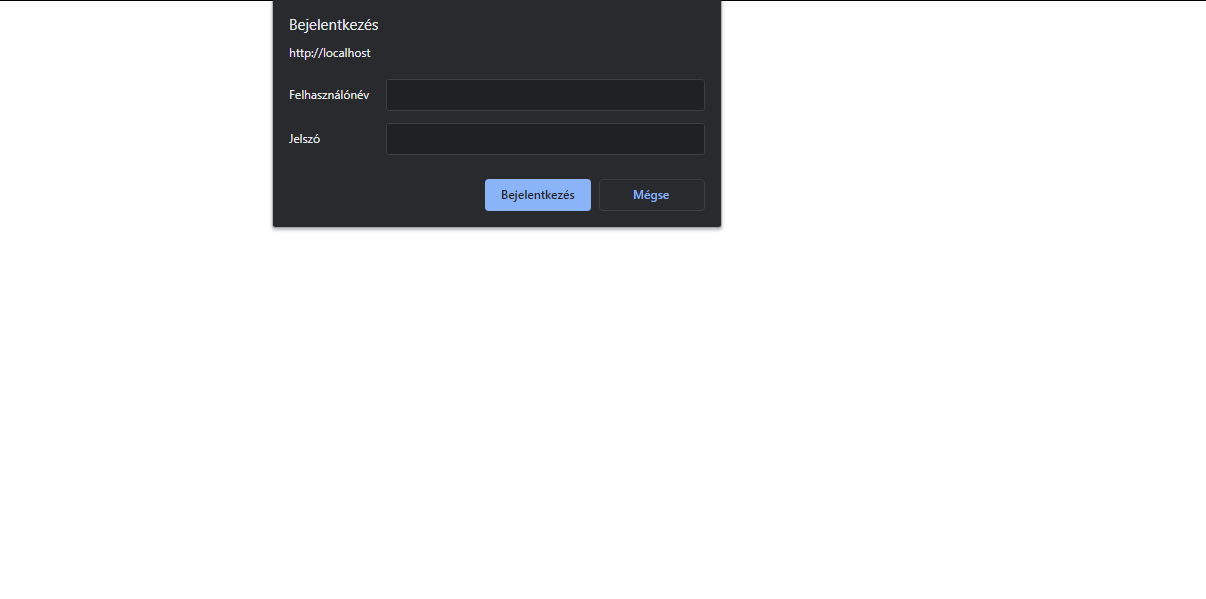
A jelentkezési folyamat pedig ezzel az üzenettel zárul:



. ábra sikerestankfelvitel.php

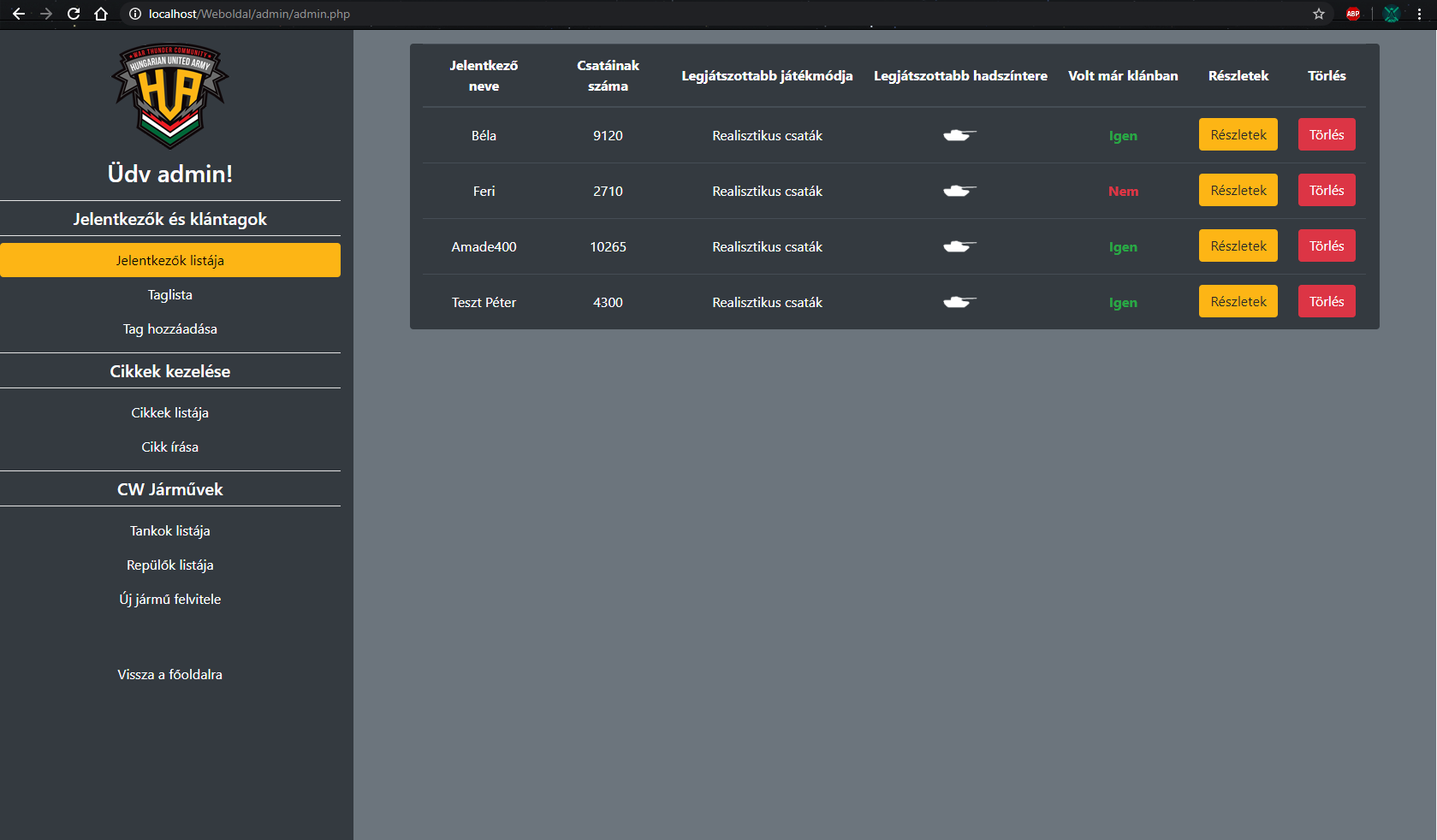
## Az admin felület

Az oldal jelentős részét az admin felület képezi. Ezt alapból nem találjuk a menüpontok alatt. Az admin felület eléréséhez a címsorba a következőt kell beírni: localhost/weboldal/admin/admin.php. Ekkor a következő oldal fogad minket



16. ábra admin.php .htaccess-es bejelentkezési felület

A felhasználónév, amellyel ehhez hozzáférhetünk admin, a jelszó pedig HUAbest. A megfelelő bejelentkezési adatokkal már elérhetjük az oldalt.



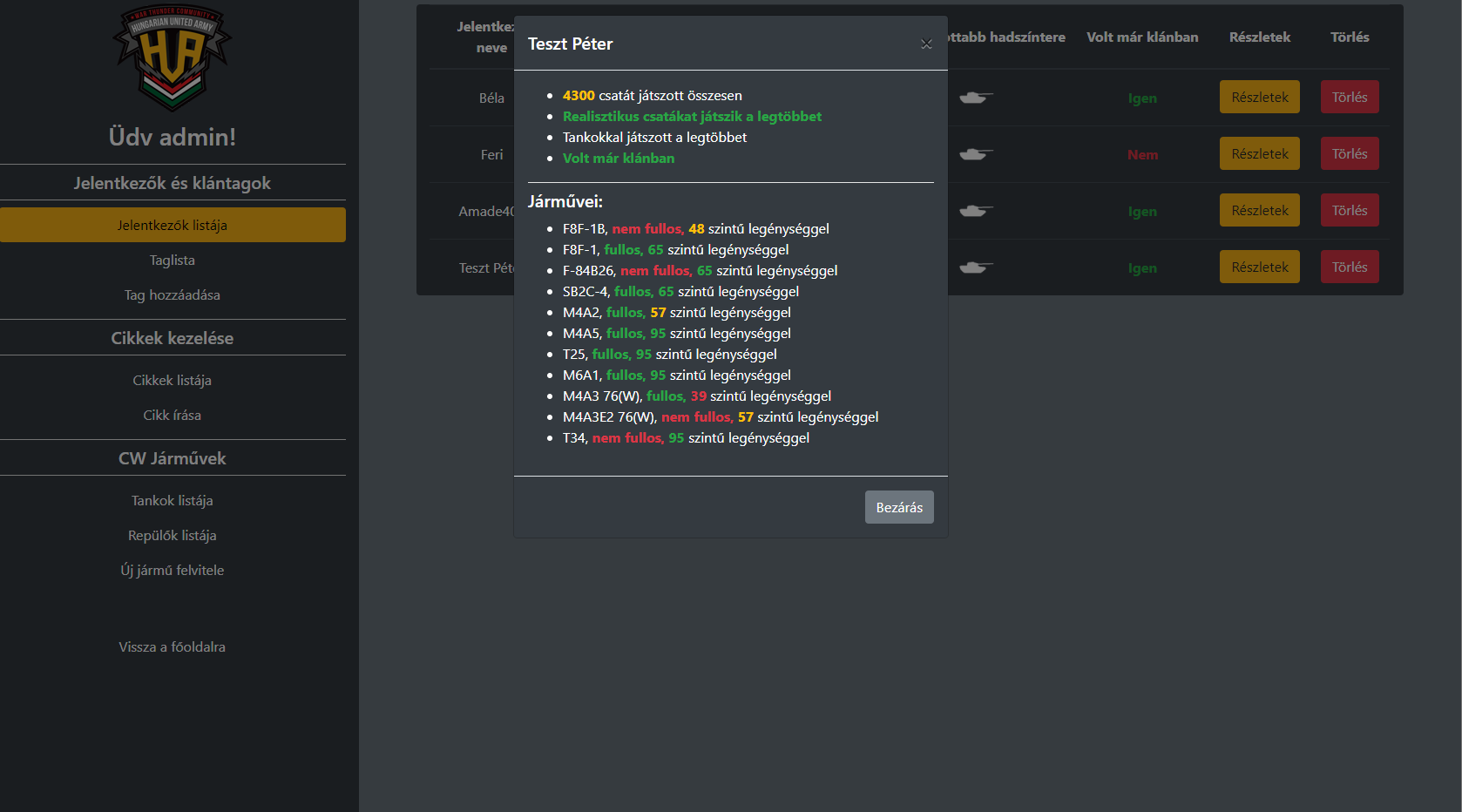
. ábra admin-php Jelentkezők listája

Bal oldalon a következő menüpontok láthatók:

* Jelentkezők listája
* Taglista
* Tag hozzáadása
* Cikkek listája
* Cikk írása
* Tankok listája
* Repülők listája
* Új jármű felvitele
* Vissza a főoldalra

### Jelentkezők listája

Alapértelmezetten a jelentkezők listája jelenik meg. A részletek gombra kattintva részletesebb leírást kapunk a jelentkezőről.

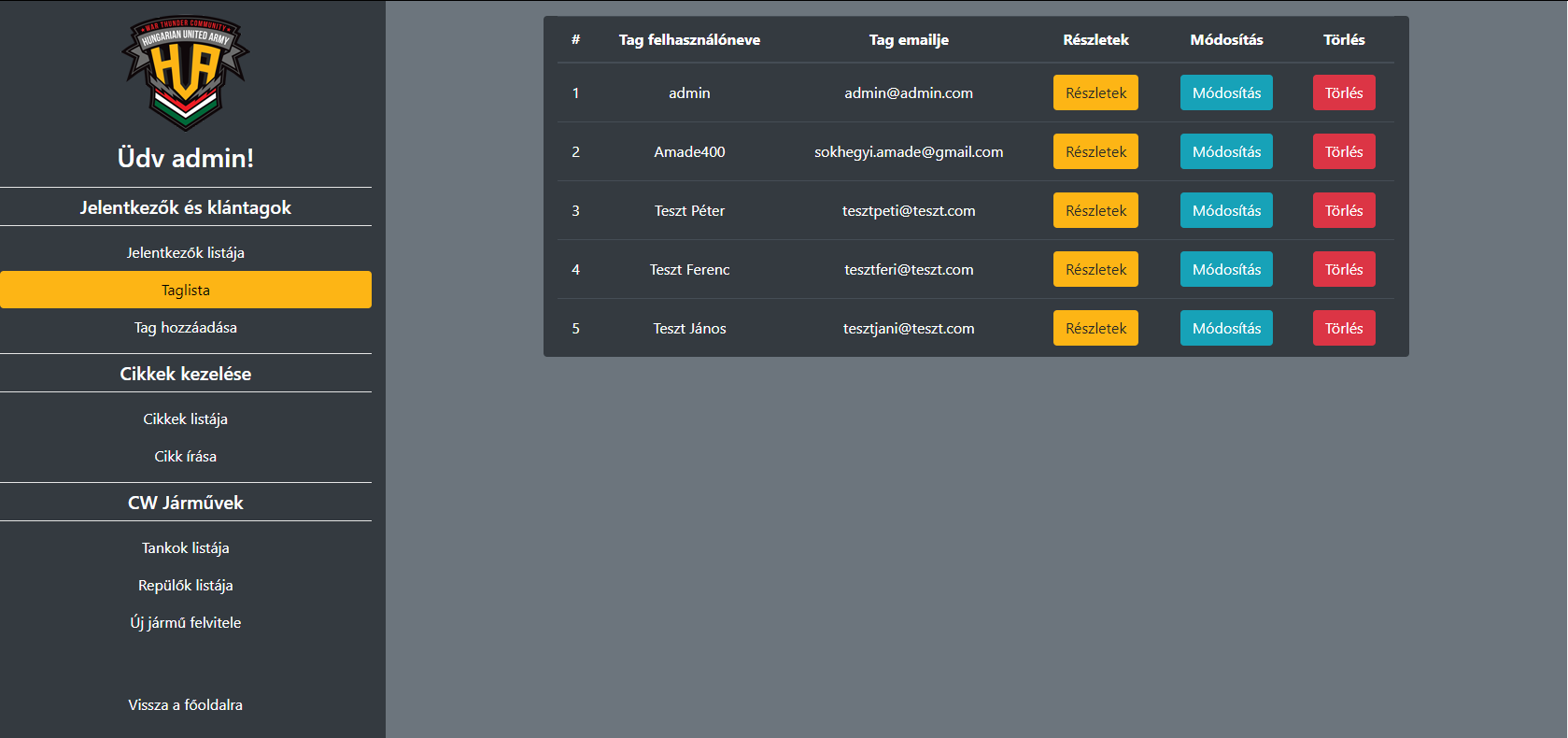


. ábra admin.php jelentkezők listája egy jelentkező részletei

Itt a jelentkezés során felvitt adatok színkódolva jelennek meg. A csaták száma 2500 alatt piros, 2500 és 7500 között narancssárga, 7500 felett zöld. 7500 csatával egy játékos már kifejezetten tapasztaltnak mondható. Játékmódok közül az árkád piros, mert egy árkádot játszó játékos nem fogja jól érezni magát a realisztikus csaták környezetében, míg a realisztikus és szimulátor mód zölden jelenik meg. Ha a játékos volt klánban, azt zölddel, ha nem, azt pirossal jelöljük. A jármű legénységénél, ha a jármű repülő, akkor 30-ig piros, 30-55 között sárga, 55 felett zöld színnel jelöljük, ha a jármű tank, akkor 50-ig piros, 50-95 között sárga, 95 felett zöld. Ha egy jármű nincs kifejlesztve teljesen, azaz nincs meg rá mindegyik modifikáció, akkor piros színű, ellenkező esetben zöld.

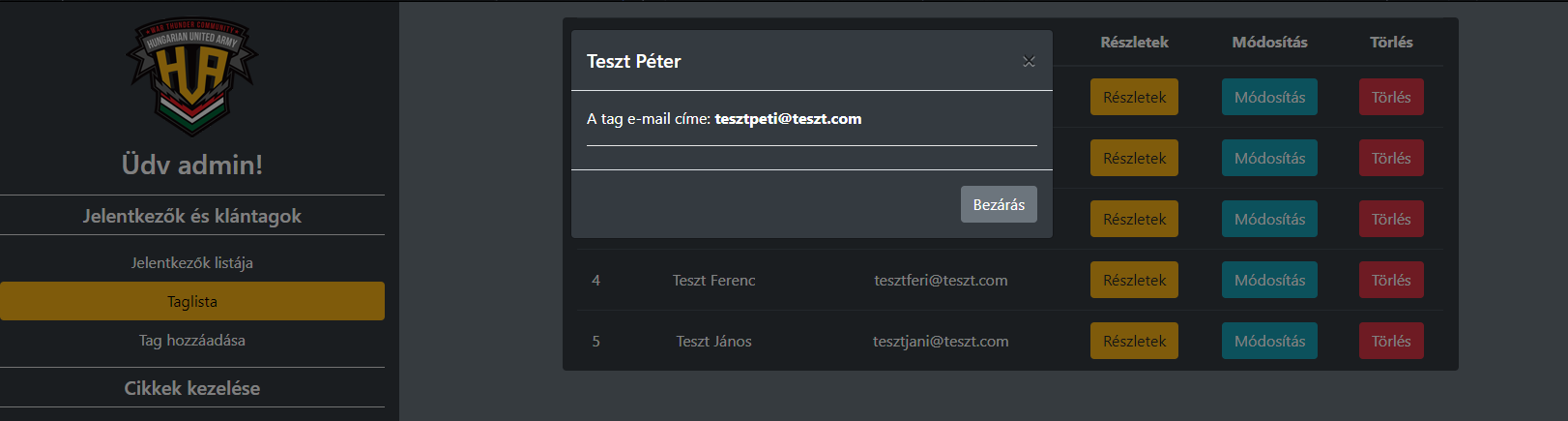
A törlés gomb megnyomásával felugrik egy ablak egy megerősítő kérdéssel arról, hogy biztos töröli akarjuk-e a jelentkezőt. Ha az igenre nyom az admin, a jelentkező törlődik az adatbázisból.

### Taglista



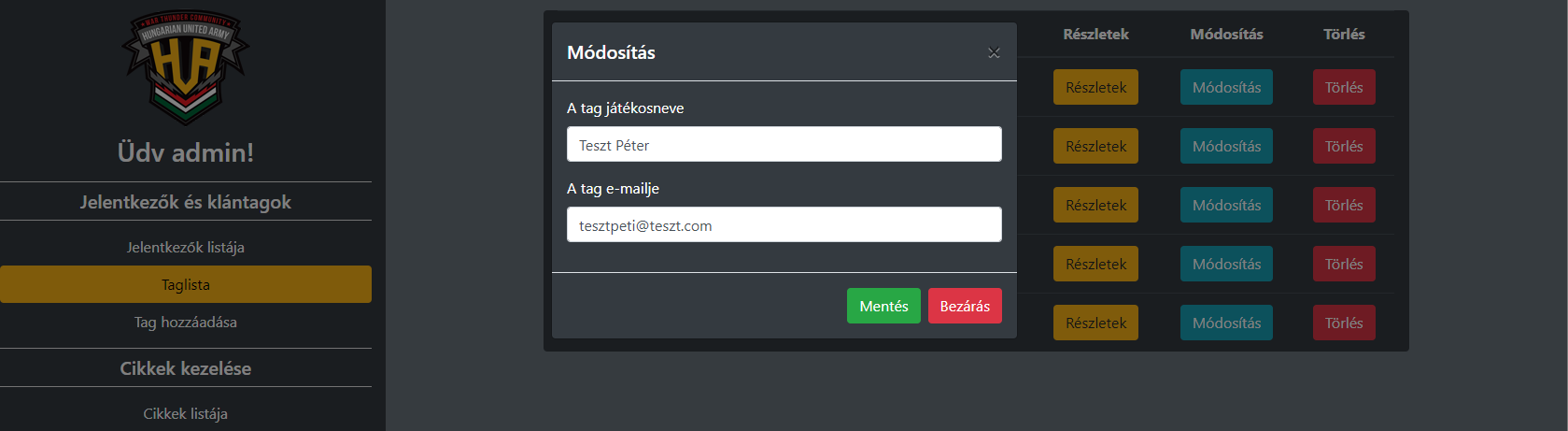
. ábra admin.php taglista

A tagok hasonló formában jelennek meg, mint a jelentkezők. A részletek gomb megnyomásával az adott tagról kapunk információkat.



. ábra admin.php jelentkezők listája egy tag részletei

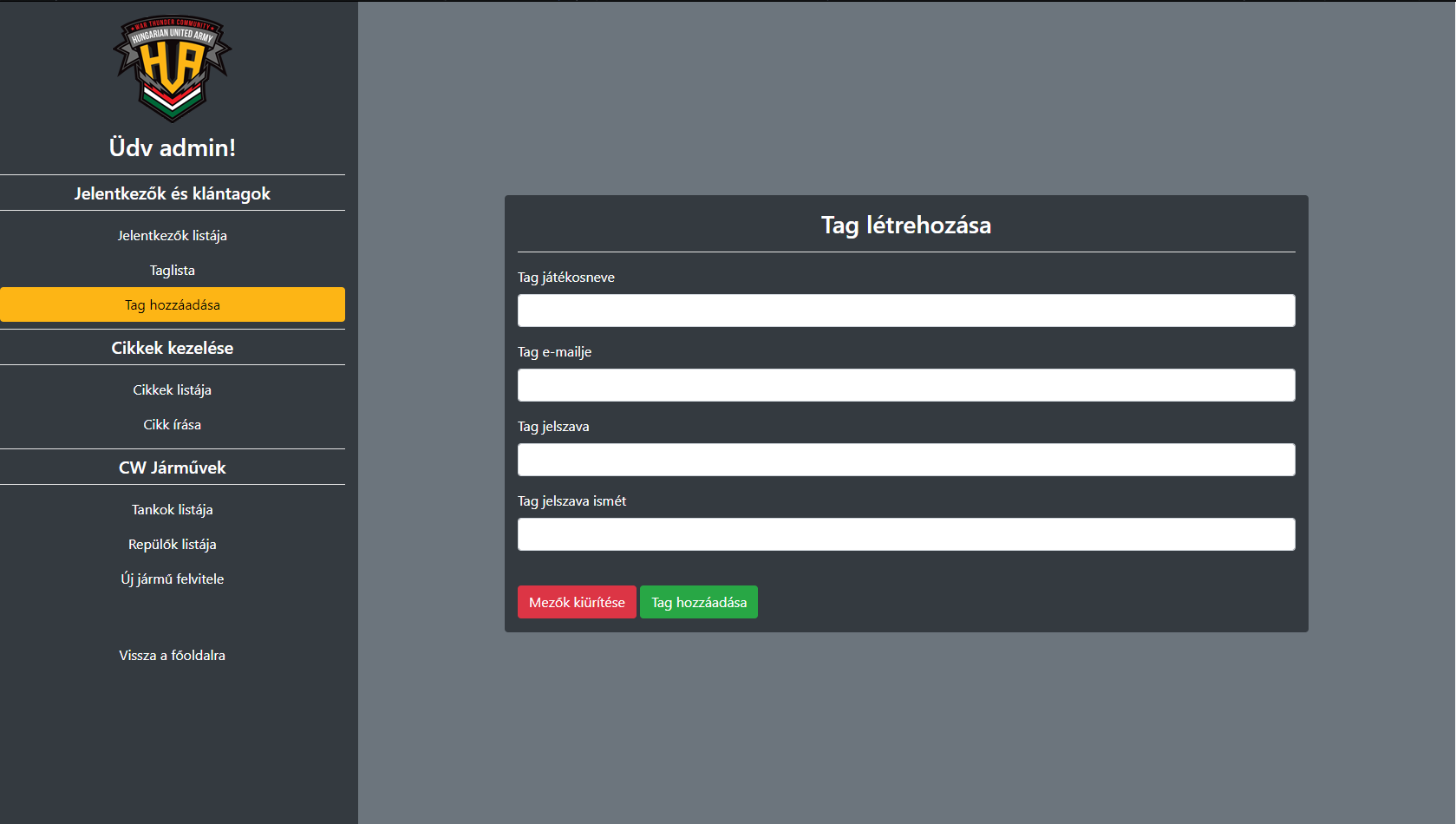
A módosítás gombbal a felhasználó nevét tudjuk módosítani. Az e-mailt biztonsági okok miatt nem lehet módosítani.



. ábra admin.php egy tag módosítása

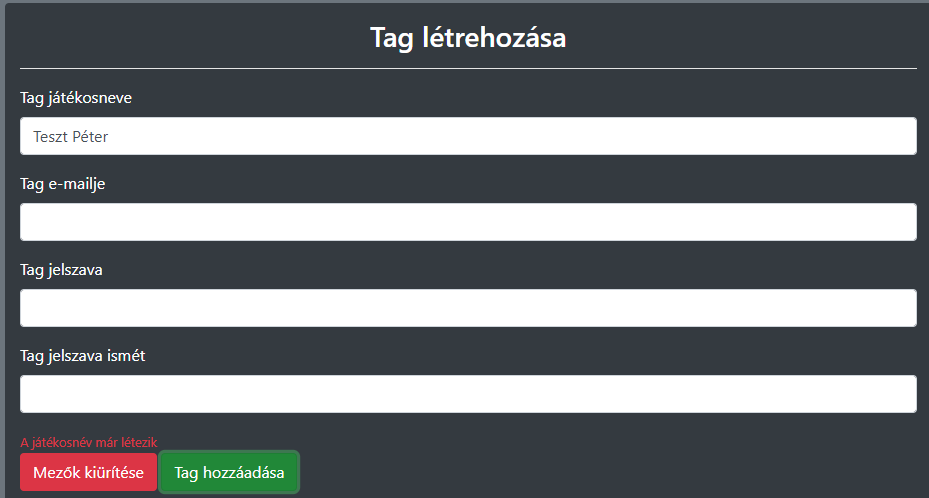
Törölni a felhasználót pedig a törlés gombbal lehet, ezesetben megjelenik egy üzenetablak, ami megkérdezi, hogy biztos töröljük-e a felhasználót. Az igenre kattintással a felhasználó törlődik a rendszerből.

### Tag hozzáadása



. ábra admin.php tag létrehozása

Az admin itt adhat hozzá felhasználókat. Ezt azért így csináltam, hogy egy átlagos felhasználó ne tudjon regisztrálni, ezzel felesleges munkát adva az adminoknak. Az admin megadja az alapértelmezett jelszót a játékosnak, aki bejelentkezés után meg tudja változtatni azt. Az e-mail megadásánál fokozott figyelem ajánlott, ugyanis ezt később nem lehet módosítani!

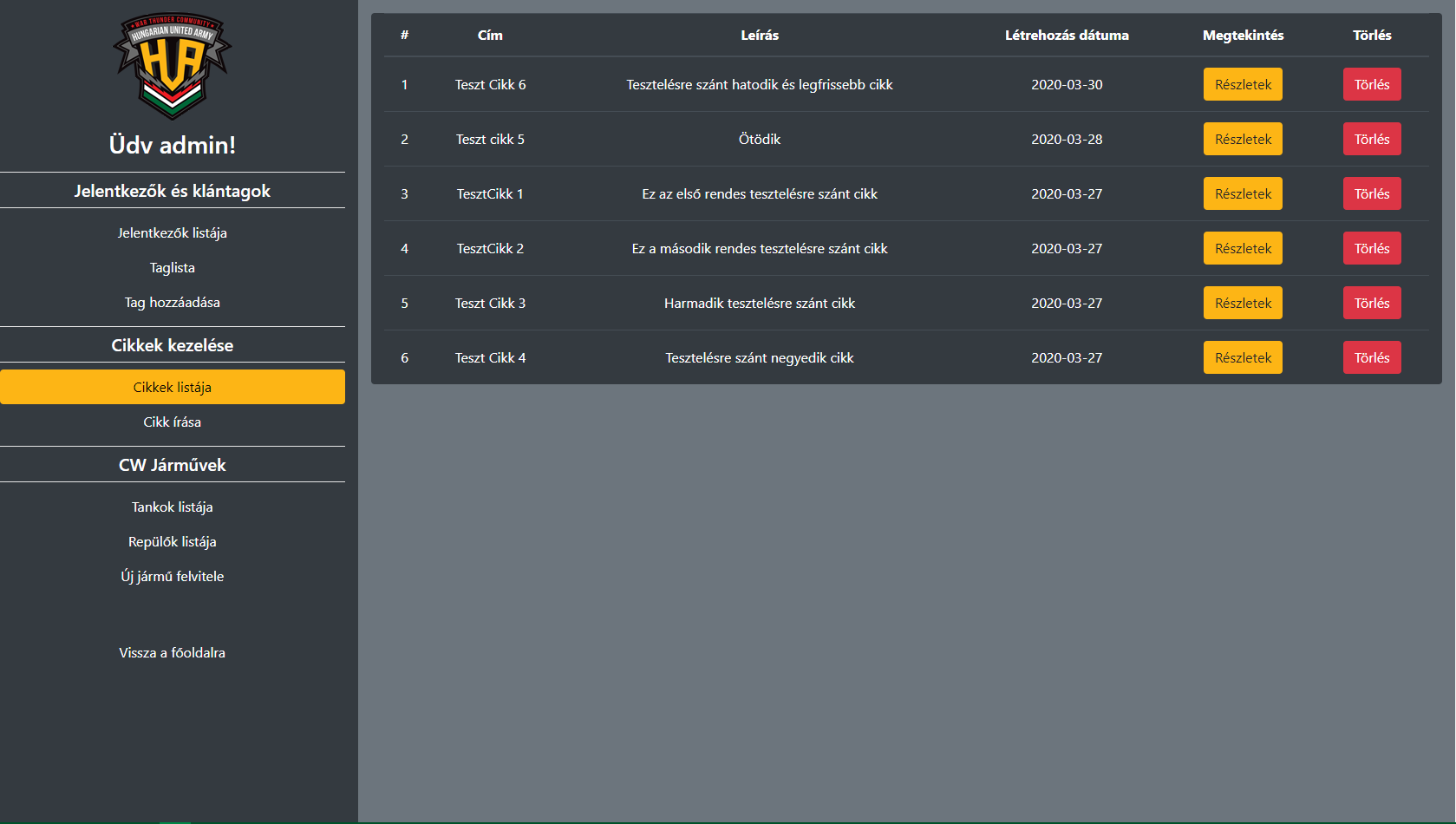


. ábra admin.php tag létrehozása hibaüzenet

A felhasználónevet és az e-mailt ellenőrzi a rendszer, hogy kétszer ugyanaz az adat ne legyen felvéve az adatbázisba.

A tag hozzáadása gombra kattintva megjelenik a sikeres hozzáadás üzenet, a mezők kiürülnek, és a Taglista oldalon ellenőrizhetjük, hogy sikeres volt-e a felvitel.

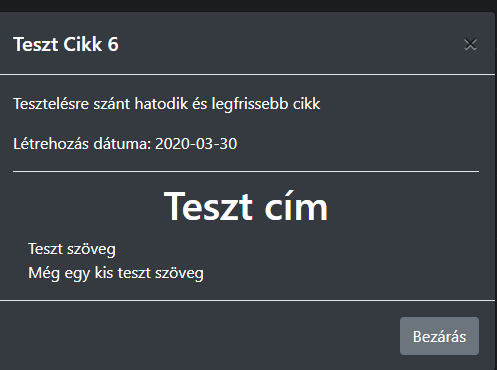
### Cikkek listája



. ábra admin.php cikkek listája

A cikkek listája oldalon a cikkeket láthatjuk kilistázva, dátum szerint rendezve.

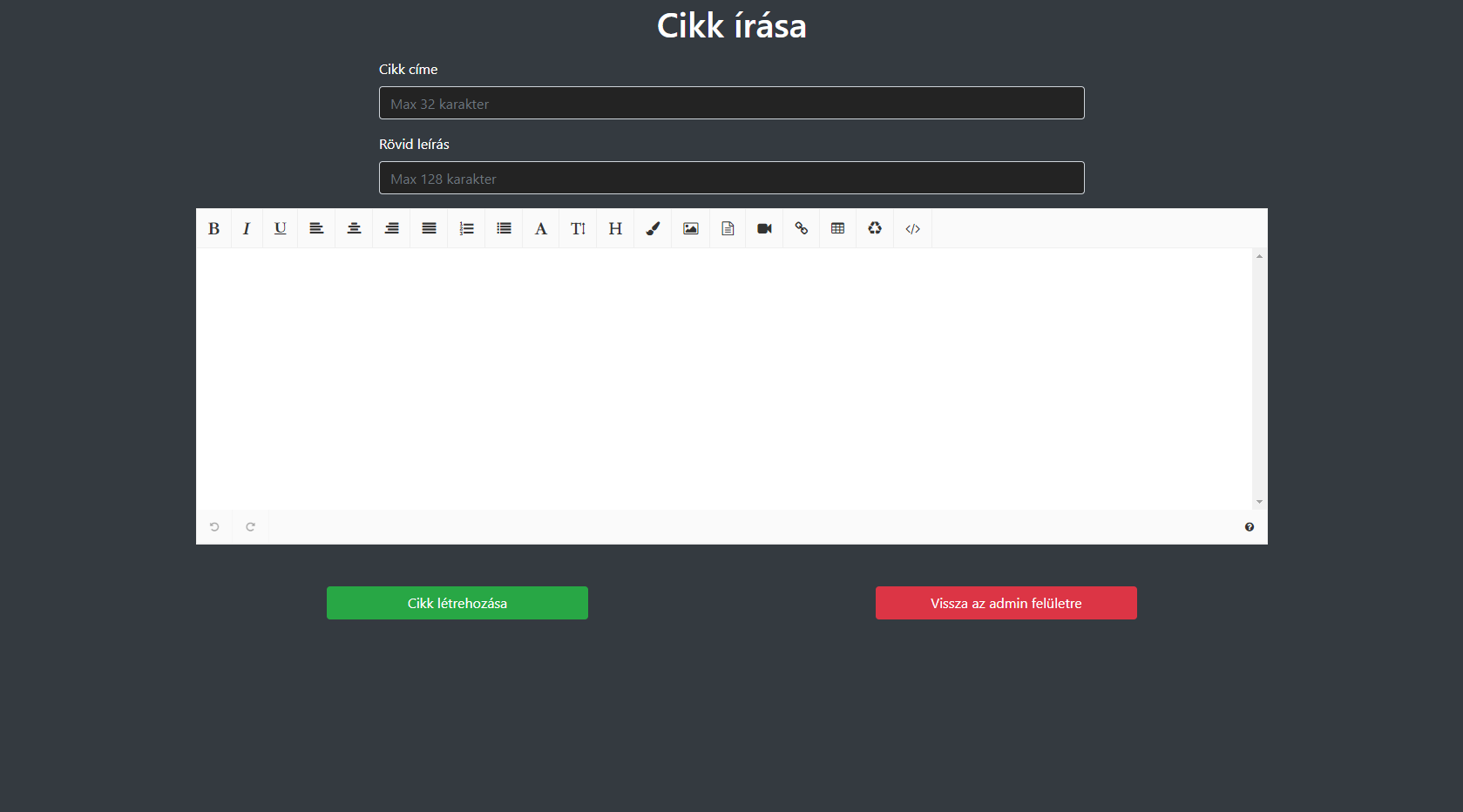
A részletek gombra kattintva megnézhetjük, mi a tartalma 1-1 cikknek pontosan.



. ábra admin.php cikk részletek

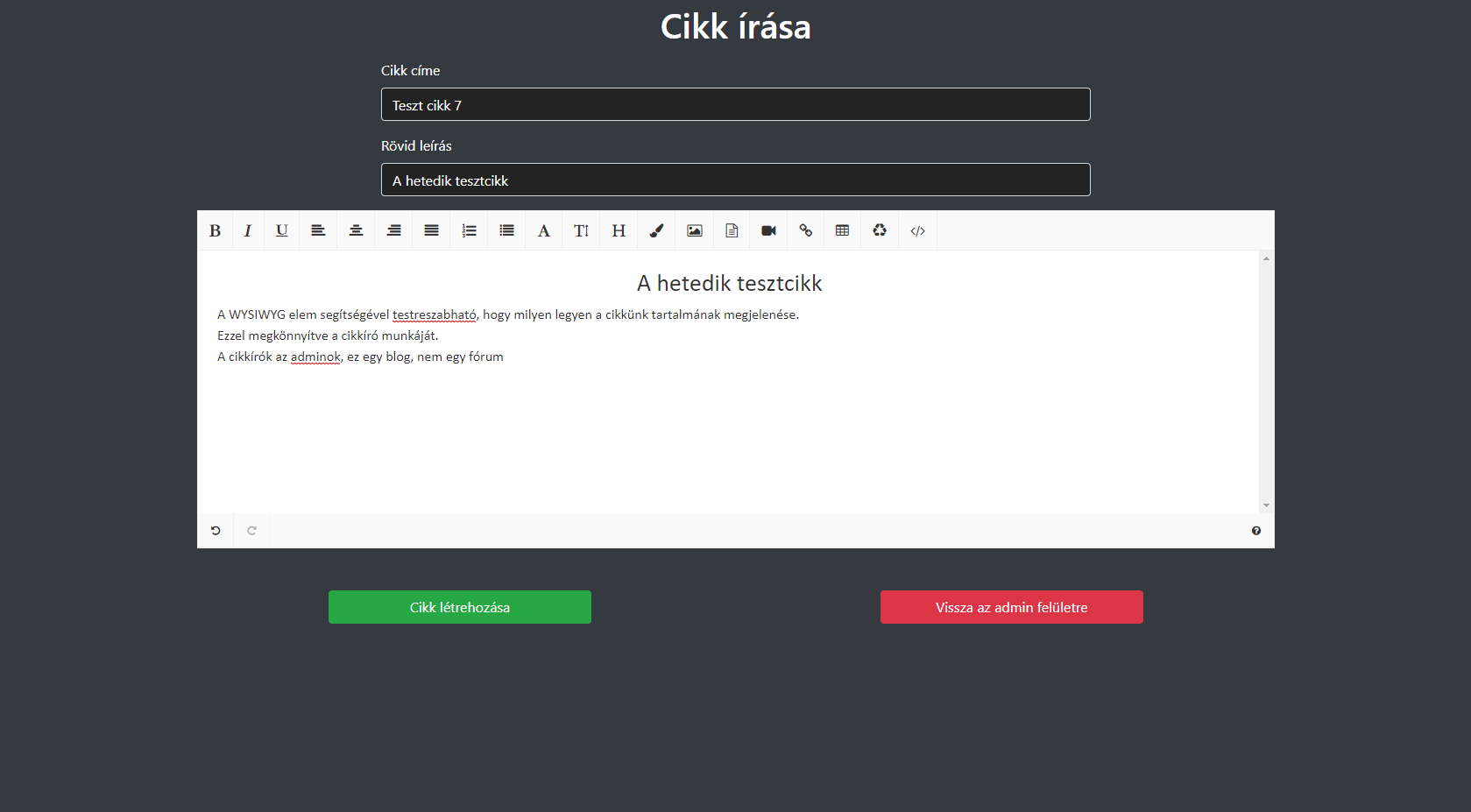
A törlés gombra kattintva pedig a rendszer a megszokott módon egy üzenetablakban rákérdez, hogy biztosan töröljük-e a cikket.

### Cikk írása



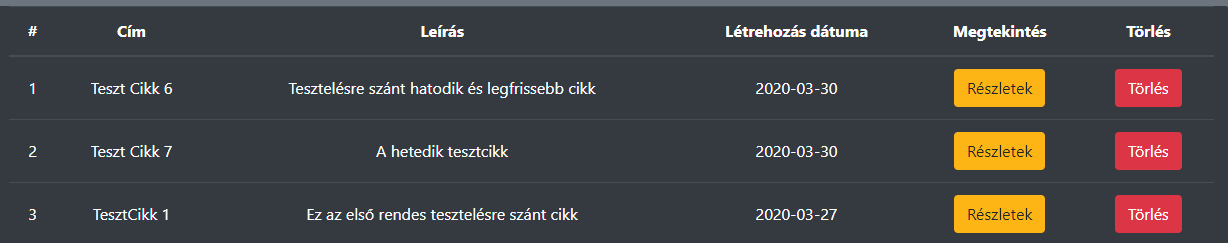
. ábra editor.php cikk írása

A cikk írása menüpont egy másik oldalra irányít át minket. Be kell írni a címet, egy rövid leírást megadni, és utána a nagy szöveges mezőben begépelhetjük a cikk tartalmát.

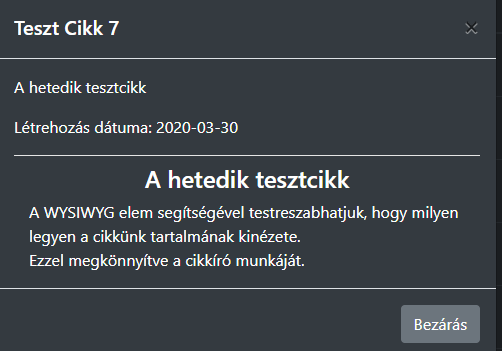


. ábra editor.php cikk megírva

A cikk létrehozása gombra kattintva 1 másodperc múlva átirányít minket az oldal vissza az admin felületre, ahol a Cikkek listája menüpontra kattintva láthatjuk is az új cikket

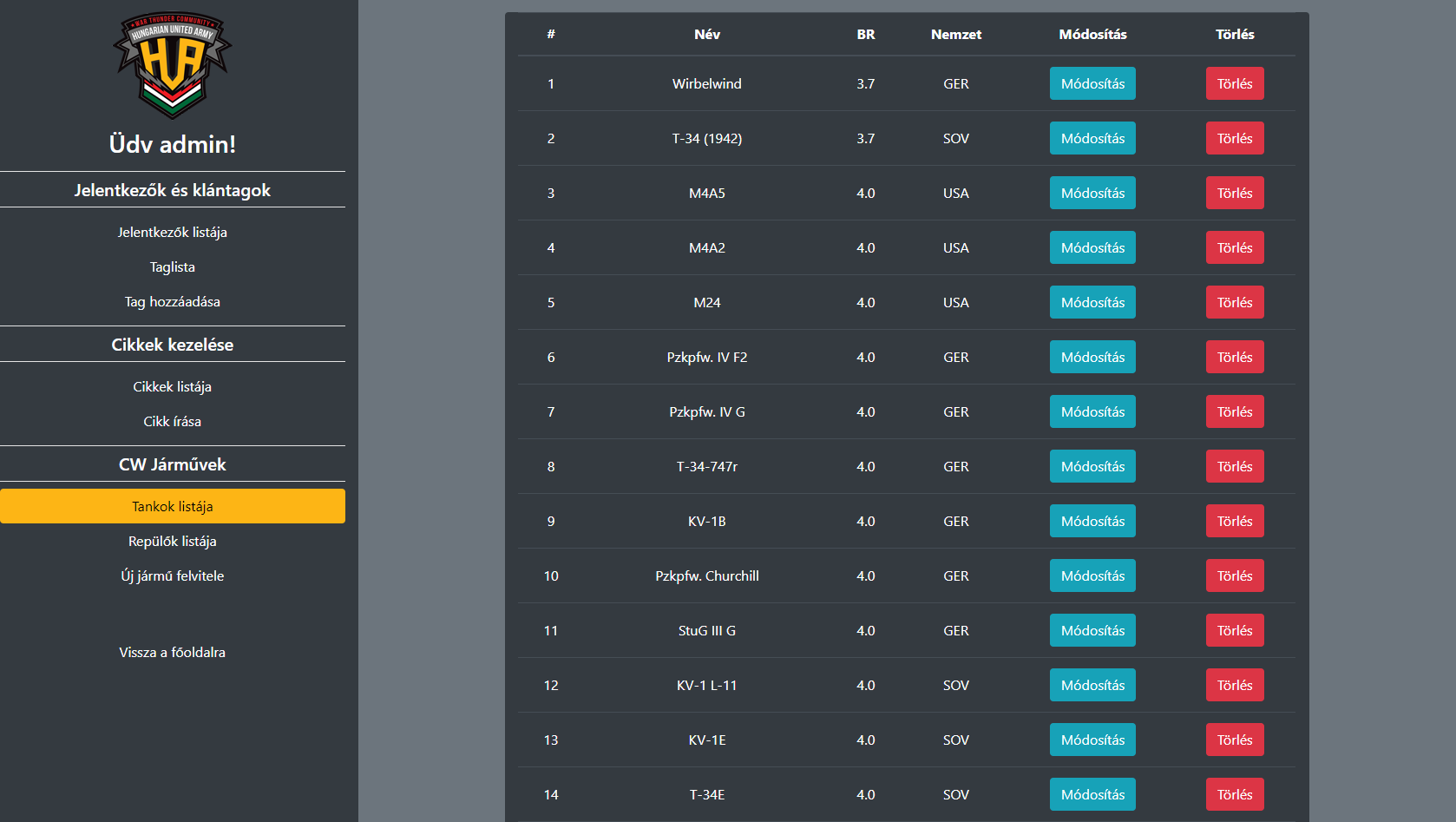


. ábra admin.php cikk létrejött, bent van a listában



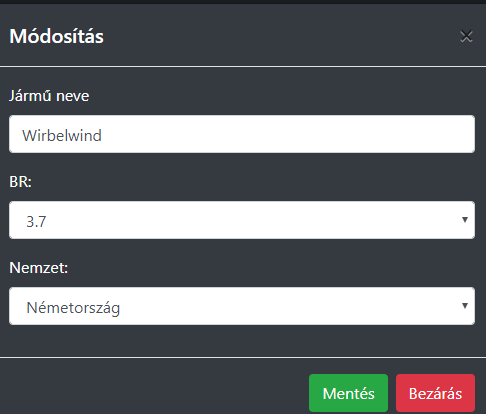
. ábra admin.php cikk részletek

### Tankok listája



. ábra admin.php tankok listája

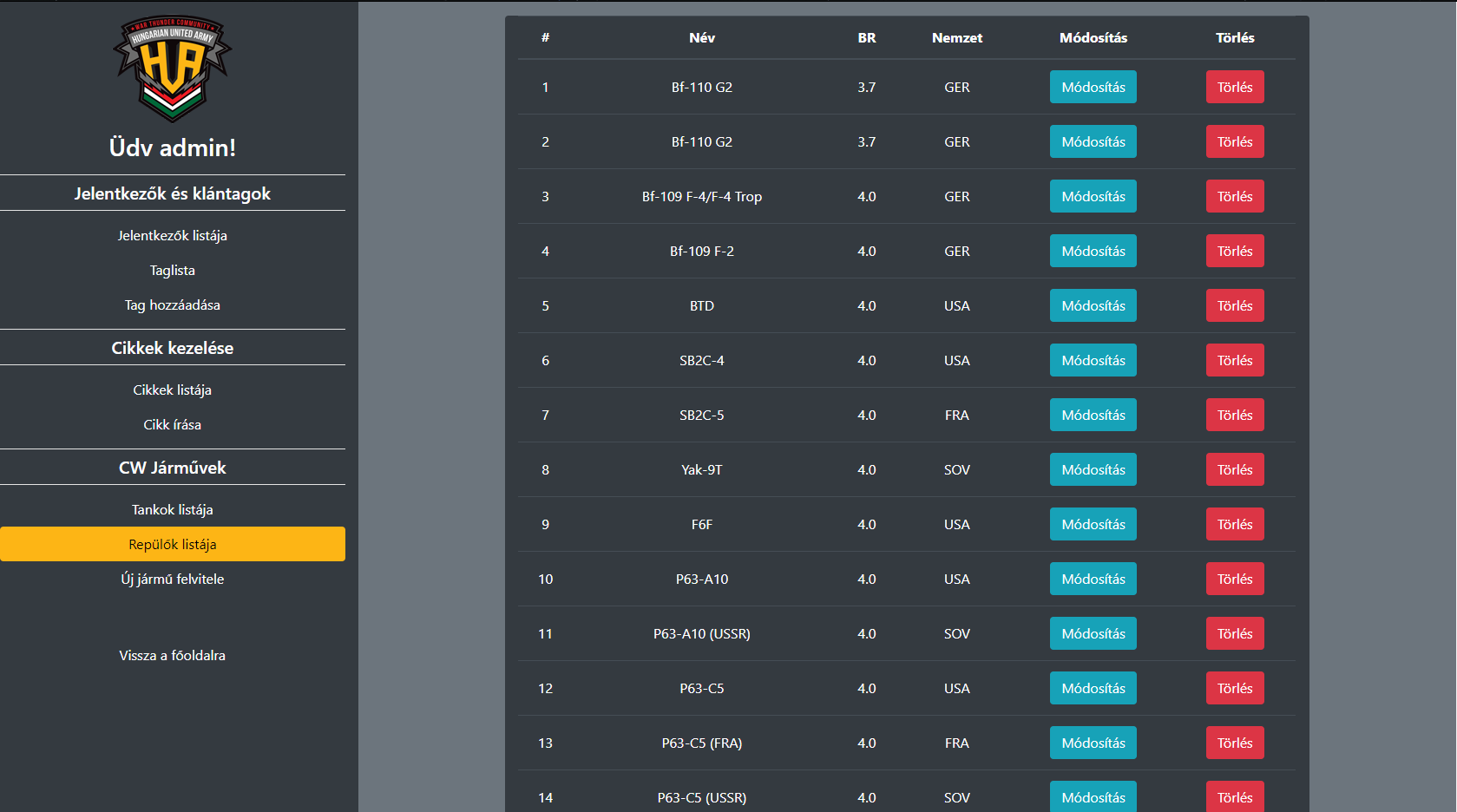
A tankokat, melyek a jelentkezési felületen elérhetők, itt találjuk meg. Lehet módosítani őket a módosítás gombbal.



. ábra admin.php tank módosítása

Illetve törölhetjük azokat a megszokott módon, üzenetablakkal értesítve az admint.

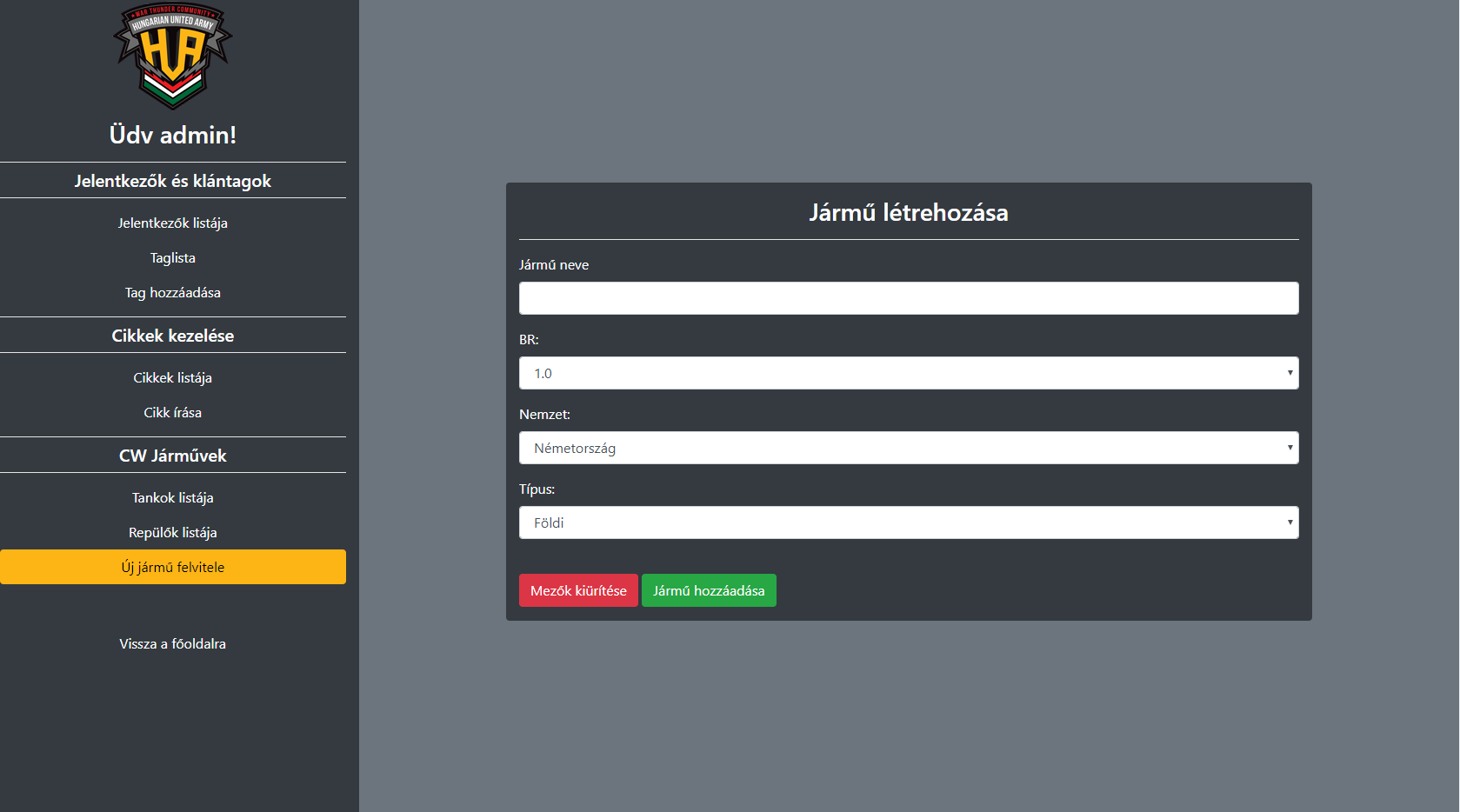
### Repülők listája



. ábra admin.php repülők listája

A repülők listája ugyan úgy épül fel, mint a tankok listája.

### Új jármű felvitele

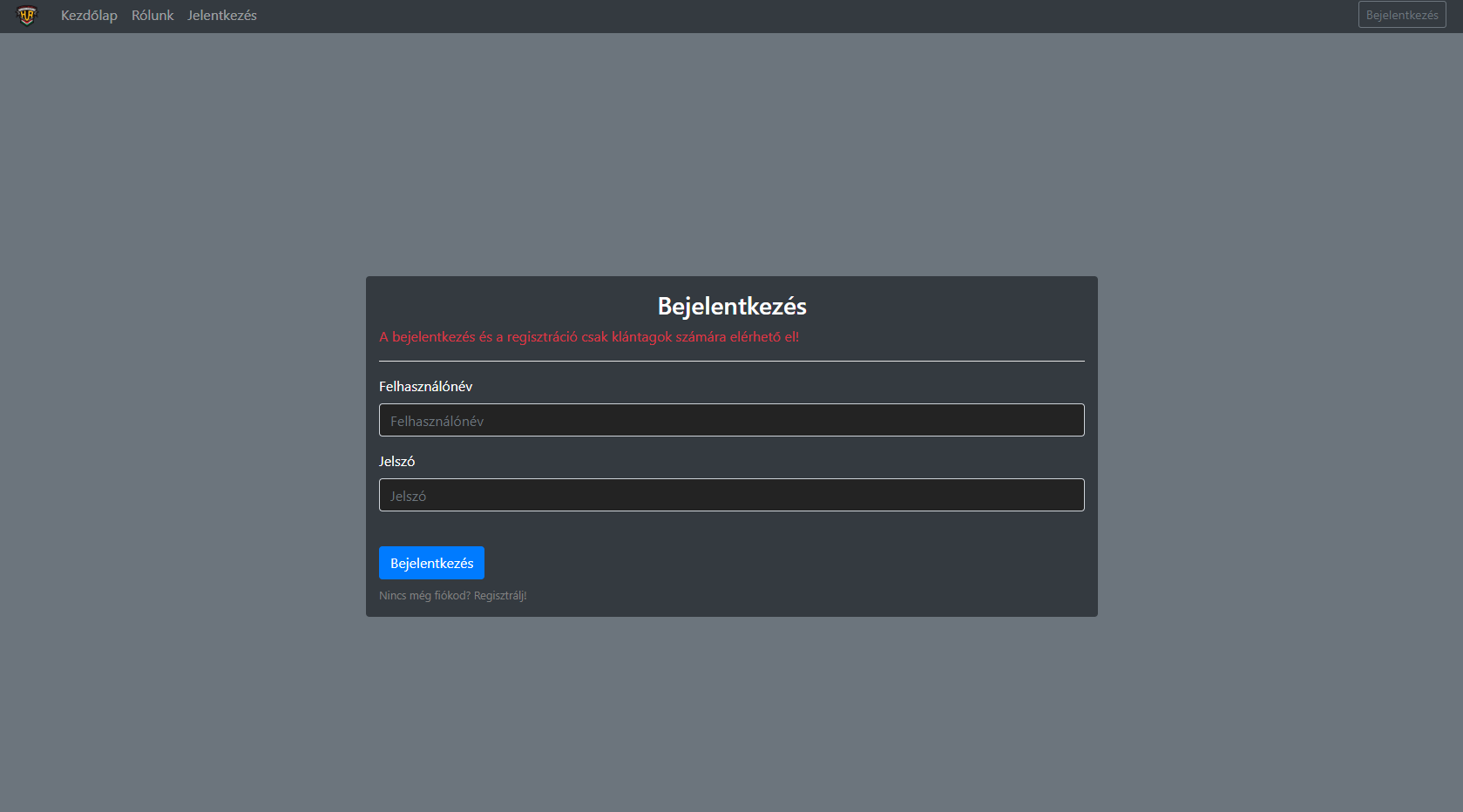


. ábra admin.php új jármű felvitele

Itt megadhatjuk a jármű nevét, típusát, BR-jét (Battle Rating, bővebben az adatbázis leírásnál) és nemzetét. Ha már létező járművet próbálunk felvinni, hibaüzenetet kapunk

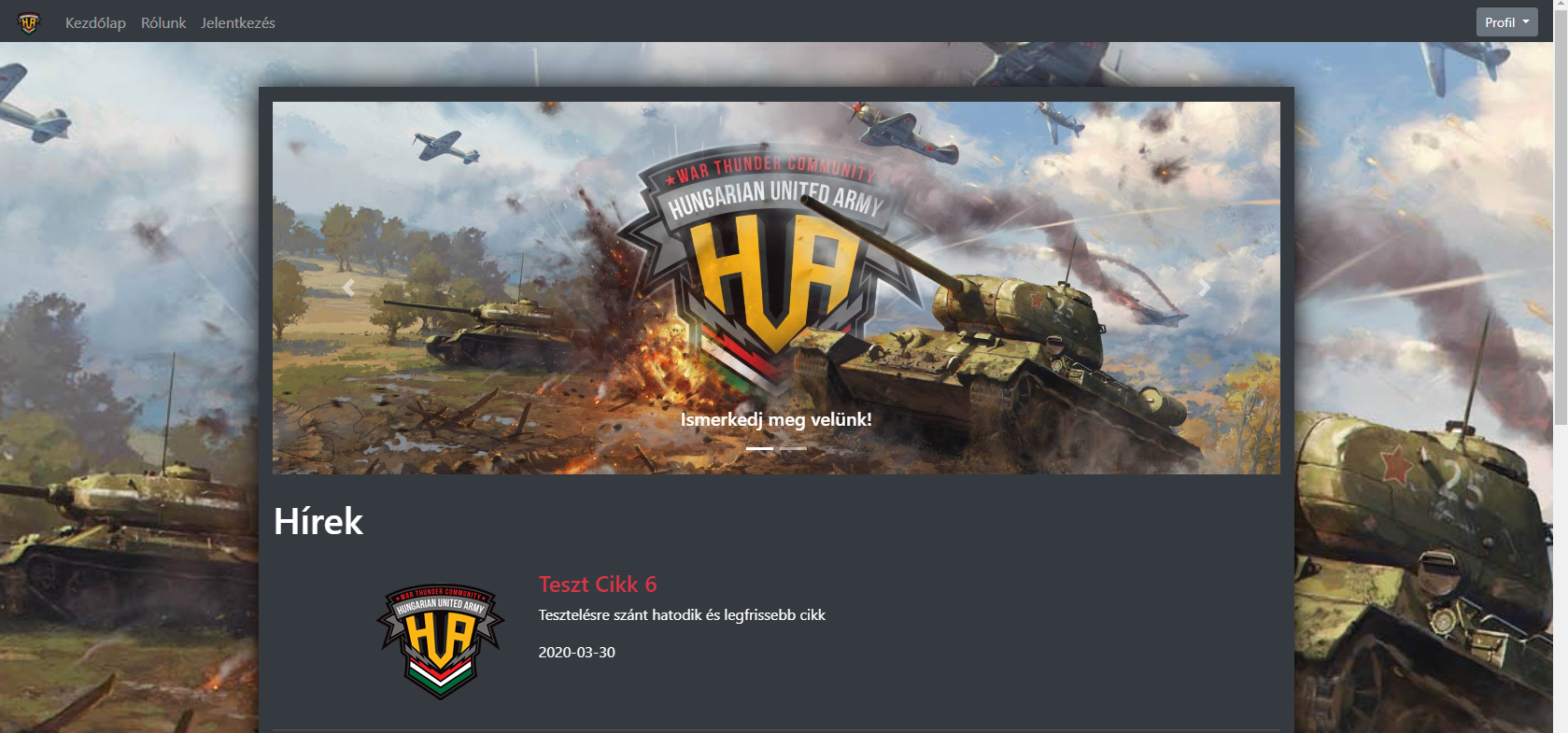
## Bejelentkezés

A bejelentkezés gombra kattintva a következő oldal fogad minket:

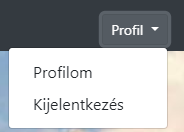


. ábra login.php

Az admin által megadott felhasználóval és jelszóval bejelentkezhetünk, ekkor a kezdőlapra irányítjuk át a felhasználót. A különbség annyi, hogy a Bejelentkezés gomb helyett egy profil feliratú legördülő lista jelenik meg



. ábra index.php bejelentkezett állapotban



. ábra index.php bejelentkezett, legördülő menü

A profilom gombra kattintva a következő felület fogad minket:



. ábra profil.php, profil szerkesztése

Itt megváltoztathatjuk a játékosnevünket és a jelszót. Fontos hozzátenni, hogy a sikeres jelszóváltoztatáshoz szükséges, hogy a Játékosnév mezőben a megfelelő adatok legyenek! Ha a régi jelszóhoz rossz értéket írunk be, vagy a két új jelszó mező értéke nem egyezik, vagy valamelyik mező nincs kitöltve, akkor hibaüzenetet kapunk. Az e-mail címet nem lehet megváltoztatni.

A kijelentkezés gombra kattintva pedig az fogad minket, ahol elkezdtük az oldal bejárását.

# Fejlesztői dokumentáció

## Témaválasztás indoklása

Ezt a témát azért választottam, mert a század vezetői szerettek volna egy platformot a Facebook mellett, amelyen keresztül kommunikálhatunk a magyar WarThunder közösséggel. Ez tökéletes lehetőséget nyújtott nekem arra, hogy létrehozzak egy weboldalt a századnak. Emiatt viszont a szoftver célközönségének létszáma alacsony. Nem egy széleskörű weboldal, de akik részt vesznek a magyar klánéletben, egy ilyen újítást örömmel fogadnak.

## Nyelvek, fejlesztői eszközök

HTML**:** A HTML (Hypertext Markup Language) egy leíró nyelv, melyet weboldalak készítéséhez fejlesztettek ki. Az én weboldalam alapját is ez képezi. A HTML nem programozási, hanem jelölőnyelv, hiszen nem tud ellátni számítási, matematikai, és adattárolási feladatokat. Erre a weboldalak esetében például a JavaScript hivatott, a HTML csupán az elemek megjelenítéséért felelős. Beépített elemeit „<” és „>” karakterek, úgynevezett „tag”-ek között hozhatjuk működésbe. Számos elemnek záró „tag”-je is van, ezek általában valamilyen konkrét megjelenítendő dolgot (pl. szöveget) tartalmaznak.

Ezekből a leggyakoribbak:

* **<p>Ez egy bekezdés</p>**
* **<img src=”kép\_relatív\_elérési\_útja.jpg”>**
* **<h1>1-es szintű címsor. (Összesen 6 szint van.)</h1>**
* **<title>Az oldal címe</title>**
* **<button>Gomb</button>**

Forrás: [[HTML5]](#_Irodalomjegyzék)

CSS**:** A CSS (Cascading Style Sheet) egy jelölőnyelv, melyet az oldalunk formásához használhatunk. Segítségével egyszerűen módosíthatjuk a HTML-ben szerkesztett weboldal elemeinek színét, méretét, betűtípusát, stb. Ezt megtehetjük külső stílusap (.css kiterjesztés), **<style></style>** „tag”-eg, vagy **style=** tulajdonsággal is. Külső stíluslap esetében fontos hozzárendelni a CSS file-t a HTML file-hoz, ellenkező esetben nem fog működni a formázás. Ezt a **<link rel="stylesheet" href="css\_relatív\_elérési\_útja" type="text/css">** taggel tudjuk megtenni.

Forrás: [[CSS]](#_Irodalomjegyzék)

Bootstrap**:** A Bootstrap egy keretrendszer, amelyet a Twitter fejlesztésekor hoztak létre. Segítségével egyszerűen, osztálykijelölőkön keresztül szerkeszthetjük a HTML oldalunkat, ehhez a Bootstrap weboldalán rengetek segédletet, kész kódrészleteket is találunk. Használatával könnyedén készíthetünk professzionális megjelenésű, reszponzív oldalt, amely akár kis nézetben, például mobilon böngészve is áttekinthető marad.

Gyakran használt jelölők:

* <div class=" container"></div>: egy div elemet rácsrendszer-kompatibilissé tesz
* <div class=" container-fluid"></div>:: egy div elemet reszponzívvá tesz alapbeállítások szerint
* <table class="table">: egy reszponzív és szép táblázat
* <div class="carousel slide" data-ride="carousel">: egy „körhinta” elem, amin különböző képeket lehet megjeleníteni
* <nav class="navbar">: egy sokoldalúan testreszabható navigációs elem
* <div class="modal">: dialógus ablak, mellyel könnyen tudunk a felhasználóval kommunikálni

Források: [[BOST1]](#_Irodalomjegyzék), [[BOST2]](#_Irodalomjegyzék)

Apache**:** Nyílt forráskódú webkiszolgáló alkalmazás, mellyel működésbe hozhatunk egy webalkalmazást. A XAMPP (szintén nyílt forráskódú) webszerver-szoftvercsomag részeként használtam. Az Apache erős és rugalmas webszerver, ami kompatibilis a HTTP1.1 protokollal. Azért is ezt használtam, mert az iskolában ezzel a webszerverrel ismerkedtünk meg, így a használata egyszerű volt számomra.

Forrás: [[APCH]](#_Irodalomjegyzék)

PHP**:** A PHP (Hypertext Preprocessor) egy általános szerver oldali script nyelv, mellyel dinamikus weboldalakat készíthetünk és tarthatunk karban. Használatával futásidőben végezhetünk műveleteket a weboldalon, adatbázis-lekérdezéseket futtathatunk, HTML elemek tulajdonságait, láthatóságát változtathatjuk.

Forrás: [[PHP]](#_Irodalomjegyzék)

MySQL: A MySQL egy SQL alapú, többfelhasználós, nyíltforráskódú, platformfüggetlen adatbázis-kezelő szerver, mely sok adat tárolására alkalmas. Ezek az adatbázisok könnyen kezelhetőek a PHP vagy az AJAX nyelv segítségével.

Forrás: [[SQL]](#_Irodalomjegyzék)

MariaDB**:** A MariaDB egy többfelhasználós, többszálú, SQL-alapú relációsadatbázis-kezelő szerver. Mivel kompatibilis a MySQL-lel, a MariaDB is lehet egy LAMP adatbázisszervere. A LAMP (Linux Apache Php Mysql) egy megoldásgyűjtemény, általában szabad és nyílt forráskódú szoftver, melyet dinamikus weboldalak és szerverek futtatásához használnak, Linux operációs rendszeren.

Forrás: [[MDB]](#_Irodalomjegyzék), [[LAMP]](#_Irodalomjegyzék)

VisualStudioCode**:** AMicrosoft terméke, ingyenesen letölthető fejlesztő környezet, a legtöbb nyelvben lehet vele dolgozni, sok hasznos funkcióval, segédlettel rendelkezik. Nagymértékben testreszabható, Windows, Linux, és OS X rendszerekre is elérhető. Ezzel a programmal írtam a kódokat. Sokat segített az alkalmazás elkészítésében, hisz az egyik legrugalmasabb szövegszerkesztő alkalmazás a piacon. Rengeteg plugin segíti a felhasználó rugalmas kódírási lehetőségét, és sokat segít a kód kiegészítésével is.

Forrás: [[VSCODE]](#_Irodalomjegyzék)

XAMPP**:** A XAMPP egy szabad és nyílt forrású platformfüggetlen webszerver-szoftvercsomag, amelynek legfőbb alkotóelemei az Apache webszerver, a MariaDB adatbázis-kezelő, valamint a PHP és a Perl programozási nyelvek értelmezői. Ez a szoftvercsomag egy integrált rendszert alkot, amely webes alkalmazások készítését, tesztelését és futtatását célozza, és ehhez egy csomagban minden szükséges összetevőt tartalmaz. A rendszer egyik nagy előnye az összehangolt elemek könnyű telepíthetősége. Azért ezt használtam, mert az iskolában ezzel a szervercsomaggal találkoztunk először, és ennek a használata és beállítása egyszerűbb volt számomra, mint megtanulni egy másik szerver szoftvercsomag hatékony használatát.

Forrás: [[XAMPP]](#_Irodalomjegyzék)

JavaScript: A JavaScript programozási nyelv egy objektumorientált, prototípus alapú szkriptnyelv, amelyet weboldalakon elterjedten használnak. A JavaScript kód vagy a html fájlban vagy külön (jellemzően .js kiterjesztésű) szövegfájlban van. Ezek a fájlok tetszőleges szövegszerkesztő (nem dokumentumszerkesztő) programmal szerkeszthetők.

Forrás: [[JS]](#_Irodalomjegyzék)

AJAX**:** Asynchronous JavaScript and XML interaktív webalkalmazások létrehozására szolgáló webfejlesztési technika. A weblap kis mennyiségű adatot cserél a szerverrel a háttérben, így a lapot nem kell újratölteni minden egyes alkalommal, amikor a felhasználó módosít valamit. Ez növeli a honlap interaktivitását, sebességét és használhatóságát. Az AJAX XHTML, DOM, XMLHttpRequest, XML technikákból áll össze. A Google Web Toolkit is ezt használja. A weboldalamon az admin felület rugalmasságát oldottam meg AJAX-szal.

Forrás: [[AJAX]](#_Irodalomjegyzék)

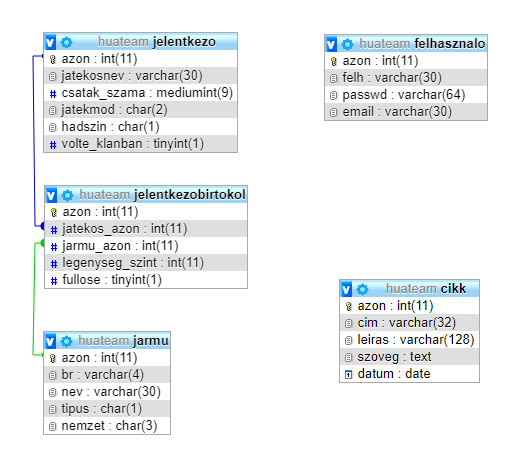
JQuery**:** A jQuery egy népszerű JavaScript könyvtár, mely a HTML kód és a kliensoldali JavaScript közötti kapcsolatot hangsúlyozza. 2006 januárjában jelentette meg a Mozilla Alapítvány népszerű tagja, John Resig. A függvénykönyvtár MIT és GNU kettős licenc alatt jelent meg. A jQuery ingyenes, nyílt forrású szoftver. Számos ismert IT cég is alkalmazza a jQuery-t saját projektjeiben, például a Microsoft erre építette a Visual Studioban is elérhető ASP.NET AJAX platformját. A jQuery célja, hogy segítsen minél inkább leválasztani a JavaScript kódot a HTML-ről, és kényelmes kommunikációt biztosítson a weblap elemeivel – eseményvezérlők és azonosítók (ún. CSS szelektorok) használatával.

Források: [[JQ1]](#_Irodalomjegyzék), [[JQ2]](#_Irodalomjegyzék)

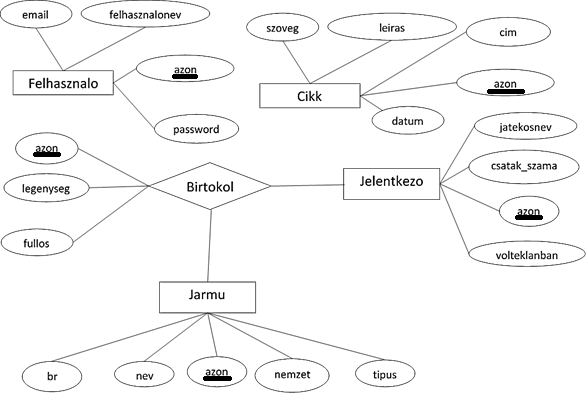
Rich Text-Editor: Egy GitHub-ról letöltött WYSIWYG elem. A WYSIWYG angol nyelvű rövidítés, feloldása: „what you see is what you get” (amit látsz, azt kapod). Ezt a cikkek egyszerű megírására alkalmazom. Működéséhez JQuery és a FontAwesome betűtípus szükséges. Források: [[RTE]](#_Irodalomjegyzék)

## Adatbázis

Az adatbázisom 5 táblából épül fel, melyek az alábbiak:



38. ábra



. ábra, az adatbázis ER modellje

|  |  |  |
| --- | --- | --- |
| Az adatbázis táblái táblázatos formában | | |
| Jelentkező tábla | | |
| Mező neve | Típus | Leírás |
| azon | int(11) | A jelentkező azonosítója, automatikusan növelt mező |
| jatekosnev | varchar(30) | A jelentkező játékosneve |
| csatak\_szama | mediumint(9) | A jelentkező összes csatájának száma |
| jatekmod | char(2) | A jelentkező legjátszottabb játékmódja, lehet AB, RB és SB |
| hadszin | char(1) | A jelentkező legjátszottabb hadszíntere, lehet A, G, S |
| volte\_klanban | tinyint(1) | Volt-e a jelentkező már században, lehet 1 vagy 0 |
| Jelentkező birtokol tábla | | |
| azon | int(11) | A birtoklás azonosítója, automatikusan növelt mező |
| jatekos\_azon | int(11) | A jelentkező azonosítója |
| jarmu\_azon | int(11) | A jármű azonosítója |
| legenyseg\_szint | int(11) | A birtokolt jármű legénységének szintje |
| fullose | tinyint(1) | A birtokolt járművön meg van-e minden modifikáció, lehet 1 vagy 0 |
| Jármű tábla | | |
| azon | int(11) | A jármű azonosítója, automatikusan növelt mező |
| br | varchar(4) | A jármű battle rating értéke, 1.0-10.3 közötti érték |
| nev | varchar(30) | A jármű játékban használt neve |
| tipus | char(1) | A jármű típusa, lehet A vagy G |
| nemzet | char(3) | A jármű nemzete, 3 karakteres rövidítéssel |
| Felhasználó tábla | | |
| azon | int(11) | A felhasználó azonosítója, automatikusan növelt mező |
| felh | varchar(30) | A felhasználó neve |
| passwd | varchar(64) | A felhasználó jelszava |
| e-mail | varchar(30) | A felhasználó e-mail címe |
| Cikk tábla | | |
| azon | int(11) | A cikk azonosítója, automatikusan növelt mező |
| cím | varchar(32) | A cikk címe |
| leiras | varchar(128) | A cikk leírása |
| szoveg | text | A cikk szövege, maximum 65535 karakter |
| datum | date | A létrehozás dátuma |

### Jelentkezo tábla

A weboldal lehetőséget ad a játékosok jelentkezésére. A tába elsődleges kulcsa az „azon”, egy automatikusan növelt mező. A jelentkezőnek meg kell adnia a:

* War Thunderben használt nevét („jatekosnev”)
* A lejátszott csatáinak számát („csatak\_szama”)
* A legtöbbet játszott játékmódját („jatekmod”, lehet ’AB’ ’RB’ vagy ’SB’)
* A leginkább játszott hadszíntért („hadszin”, lehet ’A’ = Air vagy ’G’ = Ground)
* Illetve azt, hogy volt-e már században („volteklanban).

### Jelentkezobirtokol tábla

Egy jelentkező egy adott listából kiválaszthatja azokat a járműveket, melyeket elvárunk a században. Az elsődleges kulcs „azon” egy automatikusan növelt mező. A „klantag\_azon” és „jarmu\_azon” idegen kulcsok a „felhasznalo” és „jarmu” táblákból. A legénységi szint is fontos egy adott járműnél (tanknál 150, repülőnél 75), ezt a „legenyseg\_szint” mező tárolja. Egy jármű modifikációkkal rendelkezik, melyeket kifejlesztve az eredetinél jobb teljesítményt nyújt a jármű. A századcsatákra érdemes teljesen kifejlesztett járművet vinni, ezért a „fullose” mezőben tárolom az erre vonatkozó adatot (igen/nem).

### Jarmu tábla

A jármű táblában a századunkban elvárt járműveket tárolom. Ez egy folyamatosan módosuló lista, így adminként ezt is lehet módosítani a weboldalon. Az elsődleges kulcs „azon” néven itt is egy automatikus növekedésű mező. Minden járműnek van egy „BR”-je azaz battlerating (= harci minősítés). Ez lehet 1.0-tól 10.3-ig minden tizedes törtszám, ami .0-ra, .3-ra és .7-re végződik. A jármű nevét a „nev”, típusát a „tipus” (lehet ’A’ = Air vagy ’G’ = Ground) mező tárolja. Minden jármű egy adott nemzethez („nemzet”, 3 karakter) tartozik.

### Felhasznalo tábla

A felhasználók adatai a regisztráció során a „felhasznalo” táblába kerülnek. A felhasználó adataihoz az elsődleges kulcs („azon”), a felhasználó neve („felh”), az e-mail címe („e-mail”) valamint jelszava („passwd”) tartozik. Utóbbit a biztonság érdekében titkosítva, úgynevezett hash-függvény segítségével tárolom az adatbázisban. Sóként az e-mail címet használom. A sózás egy olyan művelet, ami biztonságosabbá teszi a jelszavak tárolását. Egyszerűen fogalmazva egy szövegdarabot illesztünk hozzá a már meglévő jelszóhoz, és ezzel hasheljük, így növelve a biztonságot.

### Cikk tábla

A cikkeket tárolom itt. Az „azon” mező automatikusan növelt, a cikk elsődleges kulcsát tárolja, annak azonosítására szolgál. A cím egy 32 karakter hosszúságú szöveg lehet, a leírás 128, míg a szöveg 65535. A dátum mező pedig a cikk létrejöttének idejét jelöli.

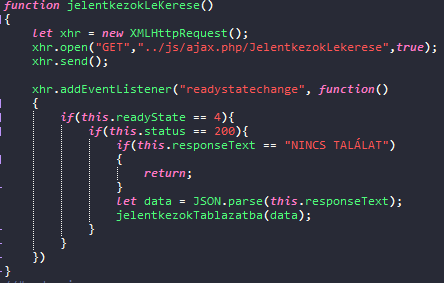
## Algoritmusok

### Táblázatba helyezés

Az első algoritmus, amit szeretnék bemutatni, a táblázatba helyezés. Az admin oldalon rengeteg adatot jelenítek meg táblázatos formában, így ez egy visszatérő algoritmus. Ezt AJAX-szal írtam meg, ahogy az admin oldal nagy részét is, a felhasználói kényelem növelése érdekében. Alapból a táblázat, amelybe adatot fogok helyezni már statikusan létezik fejléccel együtt. Ha azonban nincs még adat az adatbázisban, akkor a táblázat helyett egy szöveg jelenik meg arról, hogy nincs ebből az adatból egy sem. Ha viszont már van felvéve adat, akkor megjelenítem a táblázatot, és dinamikusan elhelyezem benne az adatokat Minden futás alkalmával ki kell üríteni a táblázat testét, hogy ne az eddig kiadott adatok után fűzze a módosított adatokat, hanem mindig frissen kérje le az adatbázisból.. A bootstrapnek köszönhetően itt a class értékekhez adhatok különböző értékeket, amik az eredmény megjelenését befolyásolják. Például a „class”, „align-middle” minden mezőre ki van adva, hogy vertikálisan is középre legyenek igazítva az elemek. A létrehozás során mindegyik gombot feliratkoztatom arra az alprogramra, amit később el kell végeznie, átadva az adott mező azonosítóját. Modalokat használok a részletek megjelenítéséhez. A modal egy beépített bootstrap elem, ami alapból a háttérben létezik, és egy megadott gomb megnyomására kerül előtérbe. Ezt a „data-target” érték változtatásával érem el. Mivel én tisztábbnak éreztem ezt az egyesével összefűzős programkódot, ezért ezt használtam, nem a **table.addRow()** függvényt. Ennek egyetlen hátránya az, hogy a programkód hosszabb, hisz mindegyik elemet egyesével kell a megfelelő helyre fűzni, megadott sorrendben.

Forrás: [[AJAXCODE]](#_Irodalomjegyzék)

A kód a következőképpen néz ki:



. ábra, részlet az admin.js állományból, 259. sor



. ábra, részlet az admin.js állományból, 281. sor

### Törlés

Egy adott sor törlése az adatbázisból minden alkalommal úgy történik, hogy az oldal megkérdezi a kezelőt, hogy biztosan végre akarja-e hajtani a törlés műveletet. Erre a JavaScript **window.confirm()** beépített függvényét használtam.



. ábra, részlet az admin.js állományból, 369. sor

### Tagfelvitel

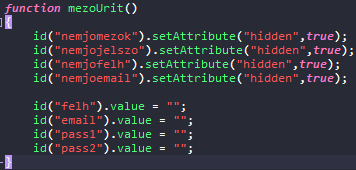
A tagfelvitel 3 lépésből áll: Az első lépésben megvizsgáljuk, hogy kitöltött-e minden mezőt, illetve elrejtjük az összes hibaüzenet mezőt, amik esetleg az előző futásnál megjelentek. Az első lépésnél egy lekéréssel megnézzük, van-e már ilyen felhasználónév az adatbázisban, itt megy tovább a program a második lépésre. A második lépésben ellenőrizzük az adatbázistól kapott adatot, és az alapján folytatjuk a folyamatot vagy megszakítjuk, kiírva a hibás felhasználó üzenetet. A második lépés végén lekérdezzük, hogy van-e ilyen e-mail cím az adatbázisban. A harmadik lépésben az adatbázistól kapott értéket ellenőrizzük, illetve a két jelszót, hogy egyeznek-e. Ha minden megfelelő, akkor felvisszük az adatbázisba a felhasználót, és 3 másodperc múlva kiürítjük a mezőket.



. ábra, részlet az admin.js állományból, 641. sor

### A mező(k) ürítése

Pár helyen ezt az egyszerű algoritmust is használom. Egy gombra rakom az alprogram lefutását, ami nem csinál mást, mint a megadott id-jű elemeket kiüríti és a hibaüzeneteket elrejti.



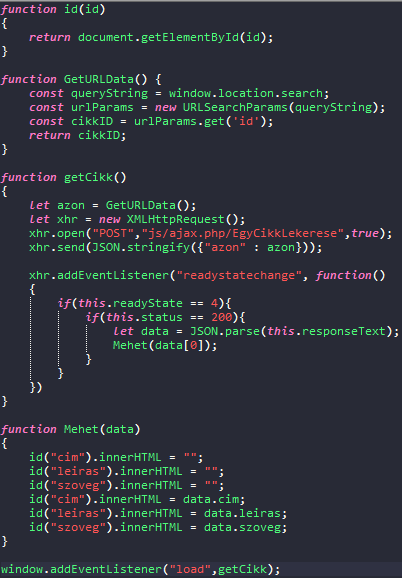
. ábra, részlet az admin.js állományból, 696. sor

### A cikkek főoldalra helyezése és működő linkek

A cikkek a főoldalra dinamikusan kerülnek ki, hasonlóan, mint a táblázatba kihelyezős függvények esetében. A fontosabb kérdés az, hogy hogyan oldottam meg, hogy minden cikk külön oldalon legyen elérhető. A link href mezőjébe a következő módon raktam az értéket: „cikk.php?id=”, tehát link segítségével adtam át adatot. A cikk.php oldalon pedig az oldal betöltésénél ellenőrzi a JavaScript a linkben az id értékét, és az alapján kéri le az adatbázisból a cikkhez tartozó értékeket.



. ábra, részlet az index.js állományból



. ábra cikk.js

### Bejelentkezés

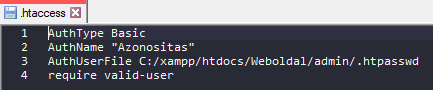
A többi kódom közül kitűnik, mivel ez az egyetlen művelet, amit tisztán php-val végzek. Ezt nehezebbnek találtam megoldani AJAX-szal, így az egyszerűbb php megoldást választottam. Hátránya, hogy minden alkalommal, amikor a felhasználó rossz jelszót ad meg, az oldal újratölt, és újra meg kell adnia a felhasználónevét is.



. ábra loginfeldolgoz.php

### .htaccess használata

A .htaccess fájlt 404-es hibaüzenet megjelenítésére és az admin.php oldal levédésére használtam. Érdekes volt megismerni ezt a módszert, és ezzel kicsit felhasználó barátabbá tenni az oldalamat. **FONTOS!** A mappa nevét (HUAblog) **NEM** szabad megváltoztatni, mert ha nem pontosan ez az elérési út, akkor a .htaccess nem fog működni!



. ábra, admin/.htaccess állomány

### Jelszótárolás

A jelszavakat az adatbázis passwd mezőjében tárolom. A jelszavakat titkosítás céljából sha256-os függvénnyel hashelve tárolom, és a biztonság növelése érdekében tárolás előtt a jelszóhoz só gyanánt az e-mail címet fűzöm hozzá. Azért így oldottam meg, hogy ne kelljen egy plusz mezőben a sót tárolni, hisz így már tárolva van az e-mail mezőben.

### E-mail módosítás

Az e-mail módosítására nincs lehetőség. Ennek az oka az, hogy az e-mail egyben a só is, ezért ha megváltoztatjuk, a fiók elérhetetlenné válik.

## Tesztelés dokumentálása

A szakdolgozatom mérete és bonyolultsága nem éri el egy átlagos blog oldal méretét, ezért a tesztelés nagyját fehér doboz módszerrel teszteltem. A fehér doboz módszer mindig a program írásával párhuzamosan zajlik. A hibák nagyja ezekben a rövid tesztelésekben került elő, és amint előfordultak, ki is javítottam őket. A tesztelés során AJAX és JavaScript kódot a console.log() függvény segítségével teszteltem az adataimat – ez a parancs kiír a böngésző Console felületére egy üzenetet. Ezt a felületet az F12 megnyomásával lehet elérni. A console.log()-okat mindig olyan helyekre raktam, ahol nem voltam biztos, hogy eljut-e odáig a program, vagy már előtte hibát dob. Arra is használtam ezt a függvényt, hogy ellenőrizzem, hogy az adott helyen megfelelő adatokkal dolgozik-e a kód. Erre jó példa az admin oldalon végzett munkám, ahol szinte minden művelet JavaScript kódot igényel. A táblázatba írás műveletnél például mindig ellenőriztem, hogy az adatbázistól kapott adatok milyen formában érkeznek meg. Ha nem megfelelő formátumot használtam, akkor a táblázatba rossz adatok kerültek. Ezt kijavítva már jó volt az alprogram. Amikor a funkciókkal végeztem, megkértem 2 barátomat, akik szintén informatikusok, hogy teszteljék az alkalmazásomat. Ennek a tesztelésnek a során derült ki egy hiba, miszerint a bejelentkezésnél bármilyen jelszóval beenged a rendszer. Ezt hamar kijavítottam, és ezután a végső tesztek szerint a programom nem rendelkezett több hibával. A végső tesztelési fázisban édesanyámat ültettem le a gép elé, hogy tesztelje a programomat, mint egyszerű felhasználó. Nem talált hibát, így magabiztosan fejeztem be a program írását. Rengeteg JSON konvertálási hibával mentem szembe a tesztelés során, amik nagyrésze hibás Ajax lekérdezés meghívása vagy rossz metódus (GET vagy POST) megadása miatt volt. A sokadik alkalommal már rutinosan tudtam, melyik hibát hol kell keresni, mit hol kell kiíratni. A php kódok tesztelésénél az echo és a print\_r() függvényeket használtam, ezzel írattam ki azokat az értékeket, amikben nem voltam biztos, hogy jók.

## Fejlesztési lehetőségek

Sok dolog van, amivel tovább tudnám fejleszteni a webalkalmazásomat.

Az egyik, hogy a cikkek megjelenése kicsit letisztultabb és szebb legyen. A cikkekhez jelenleg nem lehet feltöltött képeket fűzni, csak internetről linkelt képet. Ezt mindenképp szeretném a továbbiakban elkészíteni, hogy lehetőség nyíljon képek feltöltésére is, melyeket a cikk megírásakor lehet linkelni.

Azt is érdemes lesz kialakítani a továbbiakban, hogy lehessen hozzászólást fűzni a cikkek alá. Ehhez meg kell csinálnom a regisztrációt, melynek során e-mail-ben kapnak a felhasználók egy alapértelmezett jelszót, amivel bejelentkeznek, módosítják a jelszavukat, és szöveges kritikát is megfogalmazhatnak a közleményeinkkel kapcsolatban.

Még egy fejlesztési lehetőség, hogy egy médiatárat adok az oldalhoz. Itt a századunk néhány tagja, akik videókészítéssel foglalkoznak, megoszthatnák a munkáikat. Ehhez a profilbeállításokhoz kellene hozzányúlni, hozzáadni egy média felvitele felületet, ahova behelyezheti a videó vagy kép linkjét, ami az oldalon dinamikusan jelenik meg.

Az utolsó tervezett továbbfejlesztési lehetőség a honlap kinézetét érinti. Minden felhasználó más színvilágot szeret, más színek azok, amelyeket szívesen lát egy weboldalon. Ezért gondoltam, hogy egy témaváltó gombot hozzáadok a weboldalhoz, ami az alapszíneket (a fejléc és a tartalom színét) változtatja. A jelenlegi színekkel sötét az oldal, így kicsit borongós lehet néhány felhasználó számára, de a szemet nem terheli annyira, mint a világosabb színek.

# Összefoglaló

Dolgozatom fő célja a webfejlesztés részletesebb megismerése és különböző megoldásmódok találása volt egy adott problémára. Ez sikerült is, hisz kezdetben a php-t részesítettem előnyben, de végül áttértem az AJAX és JS gyakoribb használatára. A piaci megmérettetést - és ezek közül a Wordpress az, ami a legnagyobb konkurencia bármilyen szabadon dolgozó webalkalmazás-fejlesztőnek - szerintem nem bírná jelenlegi formájában az alkalmazásom. Nem tud elég mindent ahhoz, hogy sikeresen teljesítsen a piaci környezetben, de megpróbálom a továbbiakban olyan mértékűre fejleszteni, hogy sikeresen helyt tudjon állni a piacon is. Az alkalmazás fejlesztése során sokat fejlődött a weboldal programozói képességem, a projekt vége felé már tulajdonképpen könnyű szerrel tudtam minden hibát elhárítani. Döntéseim során figyelemmel voltam arra, hogy a jelenleg elterjedt és hozzáférhető megoldásokkal dolgozzak, és kerüljem az egyedi megoldások használatából származó inkompatibilitási és fejlesztési problémákat.

# Ábrajegyzék

[1. ábra XAMPP telepítő felület 5](#_Toc36509638)

[2. ábra XAMPP Kontrol ablak 6](#_Toc36509639)

[3. ábra localhost/phpmyadmin bejelentkező felülete 6](#_Toc36509640)

[4. ábra XAMPP felület 1 7](#_Toc36509641)

[5. ábra XAMPP felület 2 7](#_Toc36509642)

[6. ábra XAMPP felhasználói fiókok felület 7](#_Toc36509643)

[7. ábra XAMPP felhasználói felület 8](#_Toc36509644)

[8. ábra index.php 8](#_Toc36509645)

[9. ábra cikk.php 9](#_Toc36509646)

[10. ábra jelentkezes.php 9](#_Toc36509647)

[11. ábra jelentkezorepcsik.php 10](#_Toc36509648)

[12. ábra jelentkezorepcsik.php felvitt repülőkkel 10](#_Toc36509649)

[13. ábra jelentkezotankok.php felvitt tankokkal 11](#_Toc36509650)

[14. ábra sikerestankfelvitel.php 11](#_Toc36509651)

[15. ábra admin.php .htaccess-es bejelentkezési felület 12](#_Toc36509652)

[16. ábra admin-php Jelentkezők listája 12](#_Toc36509653)

[17. ábra admin.php jelentkezők listája egy jelentkező részletei 13](#_Toc36509654)

[18. ábra admin.php taglista 13](#_Toc36509655)

[19. ábra admin.php jelentkezők listája egy tag részletei 14](#_Toc36509656)

[20. ábra admin.php egy tag módosítása 14](#_Toc36509657)

[21. ábra admin.php tag létrehozása 14](#_Toc36509658)

[22. ábra admin.php tag létrehozása hibaüzenet 15](#_Toc36509659)

[23. ábra admin.php cikkek listája 15](#_Toc36509660)

[24. ábra admin.php cikk részletek 16](#_Toc36509661)

[25. ábra editor.php cikk írása 16](#_Toc36509662)

[26. ábra editor.php cikk megírva 17](#_Toc36509663)

[27. ábra admin.php cikk létrejött, bent van a listában 17](#_Toc36509664)

[28. ábra admin.php cikk részletek 17](#_Toc36509665)

[29. ábra admin.php tankok listája 18](#_Toc36509666)

[30. ábra admin.php tank módosítása 18](#_Toc36509667)

[31. ábra admin.php repülők listája 19](#_Toc36509668)

[32. ábra admin.php új jármű felvitele 19](#_Toc36509669)

[33. ábra login.php 20](#_Toc36509670)

[34. ábra index.php bejelentkezett állapotban 20](#_Toc36509671)

[35. ábra index.php bejelentkezett, legördülő menü 20](#_Toc36509672)

[36. ábra profil.php, profil szerkesztése 21](#_Toc36509673)

[37. ábra 25](#_Toc36509674)

[38. ábra, az adatbázis ER modellje 25](#_Toc36509675)

[39. ábra, részlet az admin.js állományból, 259. sor 29](#_Toc36509676)

[40. ábra, részlet az admin.js állományból, 281. sor 30](#_Toc36509677)

[41. ábra, részlet az admin.js állományból, 369. sor 31](#_Toc36509678)

[42. ábra, részlet az admin.js állományból, 641. sor 32](#_Toc36509679)

[43. ábra, részlet az admin.js állományból, 696. sor 33](#_Toc36509680)

[44. ábra, részlet az index.js állományból 34](#_Toc36509681)

[45. ábra cikk.js 35](#_Toc36509682)

[46. ábra loginfeldolgoz.php 36](#_Toc36509683)

[47. ábra, admin/.htaccess állomány 37](#_Toc36509684)

# Irodalomjegyzék

|  |  |  |
| --- | --- | --- |
| Név | Forrás | Megtekintve |
| [HTML5] | <https://hu.wikipedia.org/wiki/HTML> | 2020.03.30. |
| [CSS] | <https://hu.wikipedia.org/wiki/Cascading_Style_Sheets> | 2020.03.30. |
| [BOST1] | <https://hu.wikipedia.org/wiki/Bootstrap> | 2020.02.15. |
| [BOST2] | [https://szit.hu/](https://szit.hu/doku.php?id=oktatas:web:front-end_framework:bootstrap) | 2020.03.30. |
| [APCH] | <https://hu.wikipedia.org/wiki/Apache_HTTP_Server> | 2020.02.15. |
| [JS] | <https://hu.wikipedia.org/wiki/JavaScript> | 2020.02.15. |
| [VSCODE] | <https://azure.microsoft.com/hu-hu/products/visual-studio-code/> | 2020.03.30. |
| [XAMPP] | <https://hu.wikipedia.org/wiki/XAMPP> | 2020.02.15. |
| [PHP] | <https://hu.wikipedia.org/wiki/PHP> | 2020.02.15. |
| [SQL] | <https://hu.wikipedia.org/wiki/MySQL> | 2020.02.15. |
| [MDB] | <https://hu.wikipedia.org/wiki/MariaDB> | 2020.02.15. |
| [AJAX] | [https://hu.wikipedia.org/wiki/Ajax](https://hu.wikipedia.org/wiki/Ajax_(programoz%C3%A1s)) | 2020.02.15. |
| [JQ1] | <https://hu.wikipedia.org/wiki/JQuery> | 2020.02.15. |
| [JQ2] | [https://adamlaki.com/](https://adamlaki.com/hu/jquery-alapok-bevezetes/) | 2020.03.30. |
| [RTE] | <https://github.com/webfashionist/RichText> | 2020.03.30. |
| [AJAXCODE] | órai munka/órai jegyzet | 2019. |
| [LAMP] | <https://hu.wikipedia.org/wiki/LAMP_(szoftvercsomag)> | 2020.03.31. |