**SAPIENTIA ERDÉLYI MAGYAR TUDOMÁNYEGYETEM**

**CSÍKSZEREDAI KAR**

**GAZDASÁGI INFORMATIKA SZAK**

**DIPLOMADOLGOZAT**

**Dolgozat címe**

**Végzős hallgató:**

**Kasza Sándor**

**Témavezető:**

**drd. Filep Levente, egyetemi tanársegéd**

**2022**

**Román kivonat**

**Cím románul**

Fusce a metus eu diam varius congue nec nec sapien. Vestibulum orci tortor, sollicitudin ac euismod non, placerat ac augue. Nunc convallis accumsan justo. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

Donec malesuada vehicula lectus, viverra sodales ipsum gravida nec. Integer gravida nisi ut magna mollis molestie. Nullam pharetra accumsan sagittis. Proin tristique rhoncus orci, eget vulputate nisi sollicitudin et. Quisque lacus augue, mollis non mollis et, ullamcorper in purus. Morbi et sem orci. Praesent accumsan odio in ante ullamcorper id pellentesque mauris rhoncus. Duis vitae neque dolor.

Duis sed purus at eros bibendum cursus nec a nulla. Donec turpis quam, ultricies id pretium sit amet, gravida eget leo.

**Angol kivonat**

**Cím angolul**

Fusce a metus eu diam varius congue nec nec sapien. Vestibulum orci tortor, sollicitudin ac euismod non, placerat ac augue. Nunc convallis accumsan justo. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

Donec malesuada vehicula lectus, viverra sodales ipsum gravida nec. Integer gravida nisi ut magna mollis molestie. Nullam pharetra accumsan sagittis. Proin tristique rhoncus orci, eget vulputate nisi sollicitudin et. Quisque lacus augue, mollis non mollis et, ullamcorper in purus. Morbi et sem orci. Praesent accumsan odio in ante ullamcorper id pellentesque mauris rhoncus. Duis vitae neque dolor.

Duis sed purus at eros bibendum cursus nec a nulla. Donec turpis quam, ultricies id pretium sit amet, gravida eget leo.

Tartalomjegyzék

[1. Bevezető 4](#_Toc101870867)

[2. Használt technológiák 5](#_Toc101870868)

[2. 1. HTML/CSS 6](#_Toc101870869)

[2. 2. Bulma 6](#_Toc101870870)

[2. 3. XAMPP 6](#_Toc101870871)

[2. 4. Apache 7](#_Toc101870872)

[2. 5. MySQL Workbench 7](#_Toc101870873)

[2. 6. Kliens szerver kommunikáció 8](#_Toc101870874)

[2. 7. PHP 9](#_Toc101870875)

[3. Alkalmazás bemutatása 10](#_Toc101870876)

[3. 1. Programozási környezet 10](#_Toc101870877)

[3. 2. Adatbázis 10](#_Toc101870878)

[3. 3. Az alkalmazás fontosabb funkciói 16](#_Toc101870879)

[3. 3. 1. Login 17](#_Toc101870880)

[3. 3. 2. Regisztráció 17](#_Toc101870881)

[4. Következtetések 18](#_Toc101870882)

[Irodalomjegyzék 19](#_Toc101870883)

# Bevezető

Pénzügyi portfolió nyilvántartása egy olyan weblap amin keresztül valós adatokkal nyomon követheti egy befektető vagyona jelenlegi értékét. Az ötletet a 2020 márciusában kialakult világjárvány , Gazdasági Informatika szakon eltöltött idő által szerzet tapasztalat , 2021 Erasmus+ által eltöltött 2 hónapos szakmai gyakorlat Svédországban és a legfőbb ok amiért ezt a témát választottam az az , hogy ez aktuális is számomra mivel ez a hobbim.

Nyilván tudjuk tartani a rendszerben felhasználó ingatlanját(ait) annak értékét(eit) és gondozó személy adatait ez hasznos lehet ha több ingatlanunk van könnyeben el tudunk igazodni köztük.

Nyilván tudjuk tartani hány munkahelyen dolgozunk munkahely adatait és a munkahelyen kiadott fizetést ez hasznos lehet akár arra is , hogy egységesen lássuk ingatlanunk értékét.

Nyilván tudjuk tartani kirptóvalutáinkat meg kell adjuk hány darabot vásároltunk az adott kriptódevizából és azt is hogy milyen értéken vásároltunk és a program reális időben kiszámolja mennyi a profitunk %-ban mennyi mennyi a profitunk $ ban és mennyi a kriptóvalutánk jelenlegi értéke.

Nyilván tudjuk tartani több mint 9000 Amerikai részvényünk reális árát , akár bővebb információt is tudunk olvasni a részvényünkről, meg kell adjuk részvényünket milyen áron vásároltuk meg és hány darabot vásároltunk és a rendszer automatikusan kiszámolja mennyi a profitunk %-ban mennyi mennyi a profitunk $ ban és mennyi a részvényünk jelenlegi értéke.

Nyilván tudjuk tartani miből mennyi nemesfémünk van és reálisan tudjuk követni ennek az árát, meg kell adjuk mennyiért vásároltuk és hány darabot a program kiszámolja mennyi a profitunk %-ban mennyi mennyi a profitunk $ ban és mennyi a részvényünk jelenlegi értéke. Felhasználó reálisan láthatja mennyi a teljes jövedelme és azt is tudja követni reálisan mennyi 1 dollár deviza már pénznemekhez viszonyítva.

Az oldal össze szedi mindazt a befektetést ami egy átlagos befektető számára szükséges és mindezt interaktívan reális időben vizuálisan színezve adja jeleníti meg a felhasználó számára.

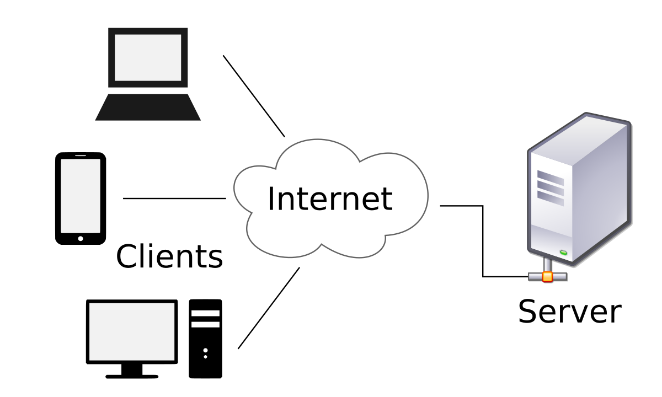
# Használt technológiák

A weboldalam elkészítéséhez több technologiára van szükség, amelyek segítenek megkönnyíteni front-end et ami a kinézetért felel és a back-end et ami a szerver oldalért felel.

A front-end az ami a felhasználó lát az oldalon feladata, hogy a felhasználó könnyen el tudjon igazodni a weboldal tartalmán.

A back-end amit a felhasználó nem lát “a hátérben történik meg” arról gondoskodik weboldal kliens oldalán minden a fejlesztő akarta szerint történjen. Több lehetőség is van back-end rendszert futtatni, én program megírása közben XAMPP használtam és a MySQL Workbench, a kész projektet saját debian szerveremre ami egy Raspbian pi 3 b+ gépen fut szükséges Apache-t telepíteni amin a PHP 7.1 fut és MariaDB is kellet telepíteni amin az oldalam adatbázisa működik. Mindezek miatt az oldalam felhasználó barát, elég gyors és megbízható mivel csak privát hálózatról lehet felcsatlakozni a szerverre.

Következő részekben kifejtem bővebben ezek technologiáját.



Kliens-szerver architektúra

(forrás: https://en.wikipedia.org/wiki/Client%E2%80%93server\_model /)

## HTML/CSS

A HTML[1] egy olyan nyelv, melyet weboldalak elkészítésére használhatunk. Minden weboldal mögött HTML található (de nem csak weboldalaknál lehet használni, hanem például e-könyveknél, vagy más dokumentumoknál is). A kifejezés a Hypertext Markup Language (hiperszöveges jelölőnyelv) rövidítése, és egy utasításkészlet a webböngésződ számára. Az utasítások betartásával a böngésző úgy jeleníti meg a weboldalad, ahogy azt a én szeretném.

A CSS[1] egy stíluslap, mely leírja egy weboldal megjelenését. A CSS azt mondja meg a böngészőnek, hogy egy HTML dokumentum egyes elemei miként jelenjenek meg a felhasználók számára. Legfontosabb jellemzője, hogy képes különválasztani a weboldal-tartalmat a designelemektől. A CSS lényege tulajdonképpen az, hogy az egész oldalra vonatkozóan kialakítasz egy megjelenést, és nem kell a HTML-ben mindenhol beállítani ugyanazt a stílust.

## Bulma

A Bulma egy ingyenes és nyílt forráskódú - Flexbox alapú CSS keretrendszer. HTML- és CSS-alapú tervezősablonokat tartalmaz ami megkönnyíti a programozó munkáját használata nagyon egyszerű, előre megírt CSS osztály nevét az adott elemben használjuk: például <input class="**input is-primary**" type="szöveg"> akár méretezi akár színezi az adott elemet.

## XAMPP

Ennek a szoftvernek a neve XAMPP[2] ami az Apache, MySQL, PHP kifejezést jelenti amí a Perl, míg az "X" betű olyan szoftverdarab, amelynek célja négy fő operációs rendszer futtatása: Windows, a Mac OS, a Linux és a Solaris. Ezt a kifejezést gyakran platformon keresztül (multi OS szoftver) nevezik.

Ahogy a neve erre utaltöbb olyan szoftver kombinációja, amely ugyanazt a funkciót támogatja, és támogatja azokat a webkészítőket, akik PC-re vagy laptopra szeretnék saját webszervert. Ezt a szoftvert a GNU is engedélyezi és ingyenesen letölthető az internetről, tekintettel annak létfontosságú szerepére, különösen a kezdő webes készítők számára.

## Apache

Ez egy nyílt forráskódú webkiszolgáló alkalmazás, melyet a modern operációs rendszerek, például a Microsoft Windows, a Netware, stb. Számára terveztek. Ezt az Apache Software Foundation fejlesztette ki. Ez a szerver olyan szolgáltatásokat kínál, mint a szerveroldali programozási nyelvek támogatása, stb. Az Apache teljesítménye csendes, mint a „más hatékony webszerverek nagy teljesítménye”. Támogatja a hitelesítési rendszereket. A virtuális tárhely lehetővé teszi, hogy egy apache-kiszolgáló különböző webhelyeket kiszolgáljon. A hibaüzenetek konfigurálhatók ezen a kiszolgálón, és számos grafikus felhasználói felület támogatja.

## MySQL Workbench

Az alkalmazás adatbázisaként a választás a MySQL Workbench-re [4] esett, amely biztonságos, rendszerek többségén futtatható, ingyenes. A többi relációs adatbázishoz hasonlóan a MySQL Workbench is sorokból és oszlopokból álló táblákban tárolja az adatokat. Számos adattípust támogat a pl számtípusokat szövegetípusoka.

A MariaDB [5] egy többfelhasználós, többszálú, SQL-alapú relációsadatbázis-kezelő szerver, egy MySQL-fork. Mivel kompatibilis a MySQL-lel, a MariaDB is lehet a LAMP adatbázisszervere.

## Kliens szerver kommunikáció

A jQuery[6] könyvtár támogatja az AJAX hívásokat, amik gyakorlatilag aszinkron HTTP kérések. Ez számunkra azt jelenti, hogy a háttérben történik egy HTTP kérés, és amikor a kérés eredménye megérkezik, akkor azt felhasználja, mindezt úgy, hogy az oldal nem töltődik újra.

Egy ilyen AJAX kéréshez szükség lesz egy jQuery.ajax() meghívására. Az .ajax() felparaméterezése szükséges a híváshoz. ezek a paraméterek egyszerű kulcs: érték párok. Az értékek lehetnek egyszerű értékek pl.: egy sztring, de gyakran előfordul, hogy egy névtelen (anonymous) függvényre lesz szükségünk.



Ábra. Ajax működése

(forrás: https://www.w3schools.com/xml/ajax\_intro.asp)

Egy HTTP kérés folyamata során egy kérést (Request) küldünk a kiszolgálónak (Server). Legtöbb esetben a küldéskor adatokat is hozzáfűzünk a kéréshez, ezen adatok küldésének több módja (Method) is van. Leggyakrabban a GET és a POST módot használjuk. A POST-ot legtöbbször űrlapok (form) küldésekor használjuk.

## PHP

Egy szerveroldali parancsnyelv, amit előszeretettel használnak például olyan backend feladatokhoz, mint a szervereken lévő adatok kezelése. Segítségével dinamikus weblapokat tudsz készíteni, ami azt jelenti, hogy nem ugyanazt a tartalmat fogja a keresett oldal minden esetben megjeleníteni a felhasználóknak, hanem bizonyos körülmények között mindig mást és mást. Rövidítése “Hypertext Preprocessor” kifejezésre utal, ami magyarul annyit tesz: hiperszöveg előfeldolgozó , a PHP-t be lehet ágyazni egy HTML-kódba.

A PHP[7]-t nagyon gyorsan be lehet üzemelni a számítógépen, a megfelelő környezet felállítása nem tart napokig mint néhány másik programnyelv esetében. Sőt, a legtöbb webhosting, azaz webhelyek tárolásával foglalkozó oldalnál már eleve telepítve van a szerverre a PHP. Ha pedig bármilyen problémába ütköznél a későbbiekben, egy hatalmas, aktív online közösség és rengeteg írásos anyag áll rendelkezésedre az interneten, ami segíthet megoldást találni.

# Alkalmazás bemutatása

Egy olyan alkalmazás elkészítéséhez, amelyben a felhasználó vagyonát tartjuk nyilván és ebből számolunk különböző statisztikai adatokat szükség van a fent említett technológiák összefüggő használatára, legfőképpen a PHP, az abba ágyazott AJAX komponensek és a szép külsőt adó Bulma CSS keretrendszer alkotja az alkalmazásom testét.

## Programozási környezet

Az alkalmazásom fejlesztése alatt a JetBrains PHPStorm[8] kódszerkesztőt használtam.A JetBrains PHPStorm viharos gyorsasággal vált a PHP-feladatok professzionális szakértőjévé.

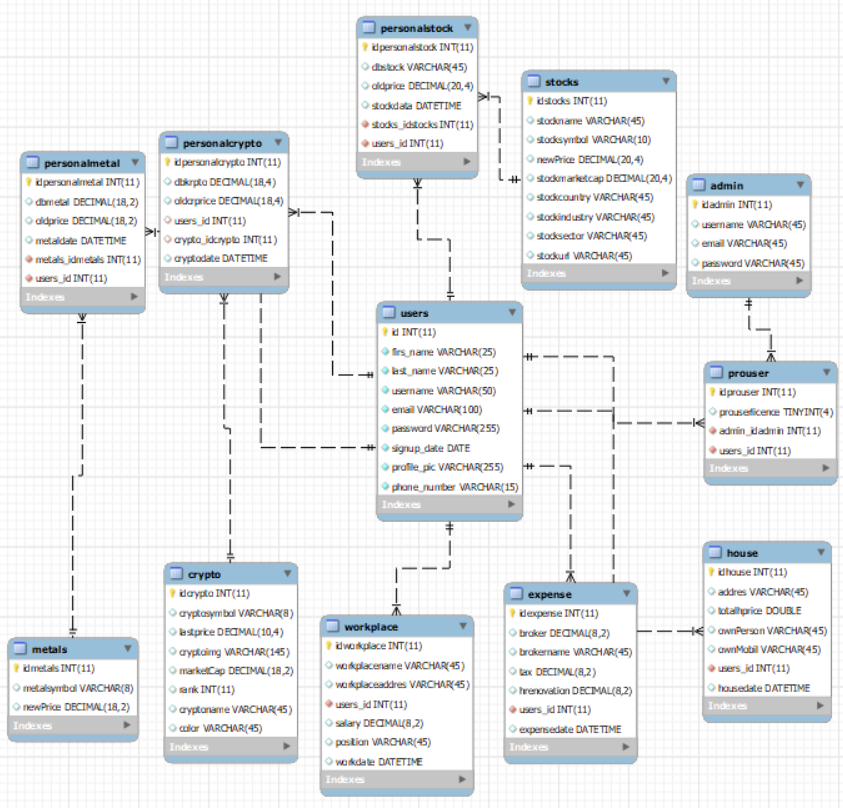
Ez egy intelligens PHP-fejlesztői környezet, amely széles eszköztárral rendelkezik valamennyi fejlesztési folyamat megvalósításához. A szoftvert a hatékonyság, a produktivitás jegyében alakították ki, minden korábbinál sikeresebbé teszi a PHP-fejlesztői munkát. Szerkesztője a begépelt kód mélyebb rétegeit is megérti és feldolgozza, valamennyi régi és új PHP-verziót támogatja.

Ez a szoftver a legjobb menet közbeni hibakeresést, kódkiegészítést és újragyártást nyújtja a világ piacon lévő szoftverek között. Mindent tud, ami a tökéletes kód kialakításához kell: (újra)igazítja, formázás stílus konfigurálással még tovább finomítja a kódot, majd vizsgálatok százaival analizálja azt a teljes projektben.

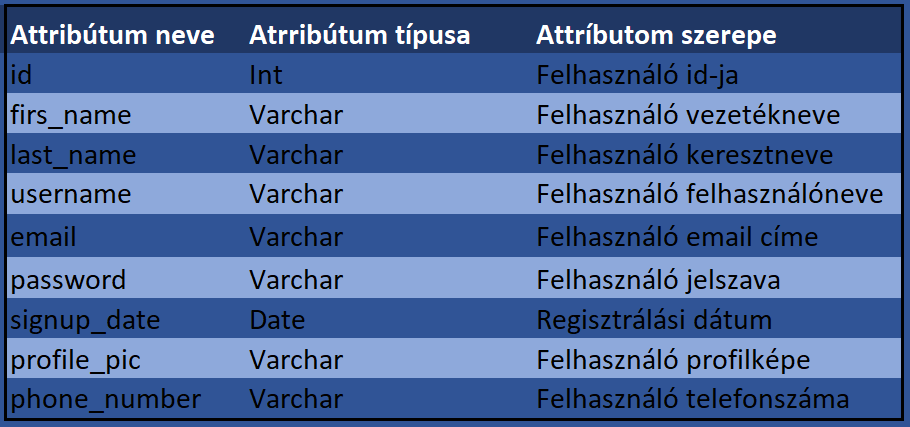
Az IDE funkciói között megtaláljuk az okos kódkiegészítést, a szintaxis kiemelést, a kiterjesztett kódformázási konfigurációt, a menet közbeni hibakeresést, a kódgöngyölítést, és a kevert nyelvek támogatását. Az automatikus refactoring nagy gondossággal vigyáz a kód biztonságára.

## Adatbázis

A 3.1. Ábra az alkalmazás sémáját mutatja be az adatbázis táblákkal és azok kapcsolataival együtt. Amint az ábrából is látszik az alkalmazás adatbázisa 12 táblázatból áll, amelyek között valamennyi kapcsolattípus előfordul.

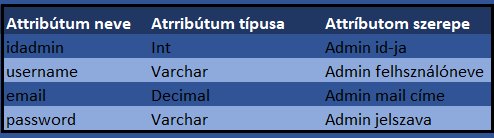


Ábra. Az alkalmazás adatbázis sémája

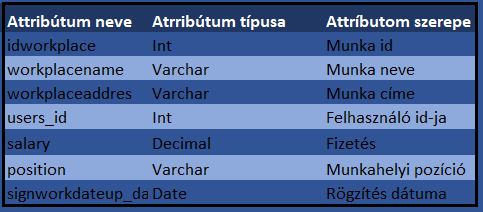


Ábra. Felhasználó tábla

A tábláim közül ez a legfontosabb tábla az egész rendszer gerincét adja, több táblám van ehhez a táblámhoz kötve 1:n kapcsolattal ha ez a táblám nem lenne nem tudnák egy felhasználóhoz vagyont esetleg kiadást hozzárendelni. Az id attrbútumhoz beállítottam primary key legyen más táblában tudjam használni beállítottam ne lehessen nulla és, hogy auto increment legyen. A felhaználónév is egyedi a felhasználó miután megkapja már nem tudja módosítani a személyes adatok módosítása oldalon. A telefonszám és az email cím is egyedi de azok tudja módosítani a felhasználó.

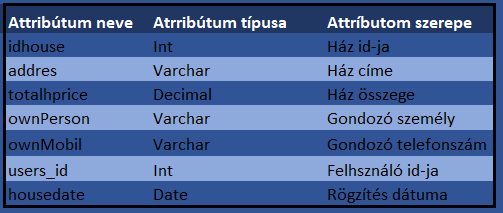


Az admin táblám szerepe felhasználók törlése , adminok törlése van egy super admin ami a weboldalon adminok listázásánal nem jelenik meg nem lehet törölni. Az adminok másik szerepe VIP tegyenek egy felhasználót. Az admin felhsználónév(username) egyedi és az email cím is.



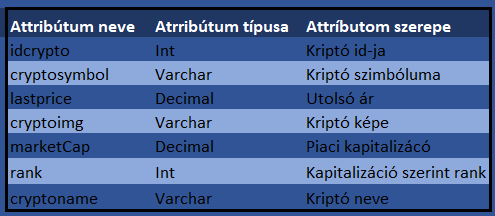
Ábra. Munka tábla

Munka(workplace) táblám szerepe a felhasználó rögzíteni tudja munkahely(eit)ét és követni tudja a felhőn keresztül, hogy mennyi fizetést kap és az aktuális pozícióját. Ahhoz hogy össze lehessen kötni users táblámmal szükség volt 1:n kapcsolatra a munka táblám megkapta users\_id attribútumot a felhasználó nem kell megjegyez saját ID ját mert a PHP- ban van egy úgynevezett munkament amikor a felhasználó rögzíti a munkahelyét a back-end kezeli users\_id alapján beszúrja.



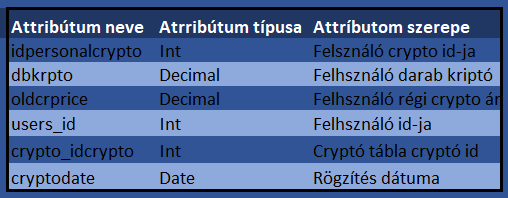
Ábra. Ház tábla

A ház(house) táblám szerepe , hogy a felhasználó rögzíteni tudja ingatlan(jait)ját elektronikus formában én nyomon tudja követni melyik ingatlana milyen címen van , a gondozó személy neve és telefonszáma , a háza teljes össze , mikor rögzítette a házát. A users táblámhoz 1:n kapcsolattal van összekötve , így mikor a felhasználó a weblapon rögzíti házát a back-end feldolgozza a kérést és users\_id alapján azt beszúrja.



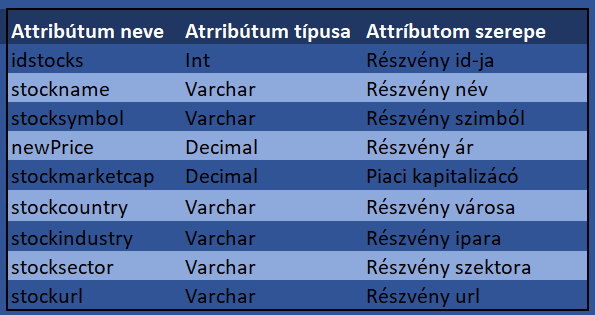
Ábra. Kriptó tábla

A kriptó(crypto) táblám feltöltése egy Élő api lekérdezésből történik meg és időközönként frissül az utolsó ár , ez ki listázásra kerül a weboldalon.



Ábra. Felhasználó kriptó tábla

A felhasználó kriptó táblám 1:n össze van kapcsola kriptó(crypto) táblámal crypto\_idcrypto által létre tudok hozni SQL lekérdezésben tábla összekapcsolást, és ugyancsak 1:n kapcsolattal van összekapcsolva users táblámal erre a dupla 1:n azért van szükség tudjam felhsználóhoz lehessen Kriptót rögziteni. Felhasználó az oldalon meg kell adja azt az árat amin vásárolta a kriptóvalutáját és azt, hogy hány darabot vásárolt és back-end feldolgozza és beszúrja a megfelelő adatot.



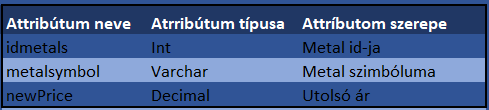
Ábra. Részvény tábla

A részvény(stock) táblám feltöltése egy Élő api lekérdezésből történik meg és időközönként frissül az utolsó ár , ez ki listázásra kerül a weboldalon.



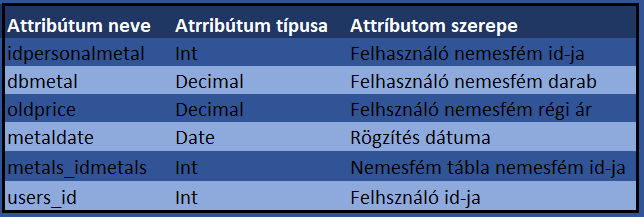
Ábra. Felhasználó részvény tábla

A felhasználó részvény táblám 1:n össze van kapcsolata részvény(stocks) táblámal stock\_idstocks által létre tudok hozni SQL lekérdezésben tábla összekapcsolást, és ugyancsak 1:n kapcsolattal van összekapcsolva users táblámal erre a dupla 1:n azért van szükség tudjam felhsználóhoz lehessen Részvényt rögziteni. Felhasználó az oldalon meg kell adja azt az árat amin vásárolta a részvényét és azt, hogy hány darabot vásárolt és back-end feldolgozza és beszúrja a megfelelő adatot.



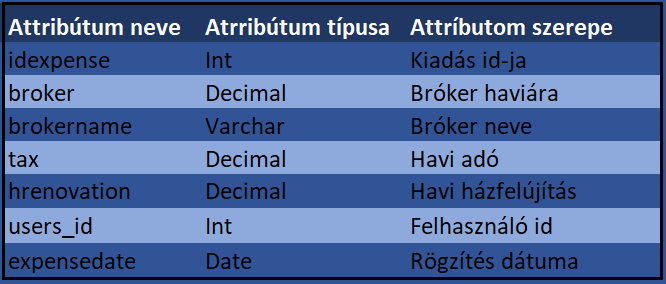
Ábra. Nemesfém tábla

A nemesfém(metals) táblám feltöltése egy Élő api lekérdezésből történik meg és időközönként frissül az utolsó ár , ez ki listázásra kerül a weboldalon.



Ábra. Felhasználó nemesfém tábla

A felhasználó nemesfém táblám 1:n össze van kapcsolata nemesfém(metals) táblámal metals\_idstocks által létre tudok hozni SQL lekérdezésben tábla összekapcsolást, és ugyancsak 1:n kapcsolattal van összekapcsolva users táblámal erre a dupla 1:n azért van szükség tudjam felhsználóhoz lehessen nemesfémet rögziteni. Felhasználó az oldalon meg kell adja azt az árat amin vásárolta a részvényét és azt, hogy hány darabot vásárolt és back-end feldolgozza és beszúrja a megfelelő adatot.



Ábra. Kiadás tábla

A kiadás táblám 1:n kapcsoltban van a users táblámal , felhasználó tudja rögzíteni brókere nevét , brókere havidíját , havi házfelujítás ára és a havi adó díja a felhasználó amint kitöltötte és elküldte az adott paramétereket és a (ebből a tábálából )főmenüben interaktívan megjelenik és azt is jelzi a renszer , hogy a felhasználü mikor rögzítette akár havonta be lehet írni az a felhsználó az adatait.

## Az alkalmazás fontosabb funkciói

### Login

### Regisztráció

# Következtetések

# Irodalomjegyzék

1. HTML/CSS, <https://webshark.hu/hirek/html/>
2. XAMPP <https://altitudetvm.com/hu/website/3096-pengertian-xampp-beserta-fungsi-dan-bagian-bagian-penting-pada-xampp.html>
3. Apache [Az Apache és a Nginx közötti különbség (spot-the-difference.info)](https://hu.spot-the-difference.info/difference-between-apache)
4. Mysql Workbanch <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>
5. MariaDB <https://hu.wikipedia.org/wiki/MariaDB>
6. jQuery <https://anon117blog.wordpress.com/tag/jquery-ajax-magyarul/>
7. PHP <https://codeberryschool.com/blog/hu/a-php-programozas-alapjai/>
8. PHPStorm https://szoftver.hu/szoftver/jetbrains/phpstorm-bemutatjuk-a-kodolas-kiralyat