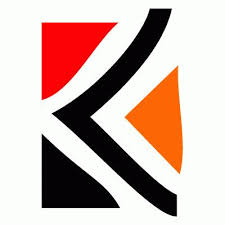
Záródolgozat

**Juhász Attila**



**OKJ 54 213 05**

**— Szoftverfejlesztő —**

**2/14/F**

**Kecskeméti Szakképzési Centrum**

**Kandó Kálmán Technikum**

**Kecskemét**

**2021**

FIGYELEM!

A program a fejlesztő tulajdona, amit a nemzetközi szerzői jogok védenek.  
A program használati engedélyét kizárólagosa**n** a verseny idejére biztosítom.  
Továbbadása, forráskódjának visszafejtése a szerző engedélye nélkül tilos.

**Számla iktatási rendszer**

**Tartalomjegyzék**

**Előszó 4  
Fejlesztői környezet 5  
Rendszerkövetelmények 5  
Alkalmazott fejlesztői eszközök, programnyelvek 5  
Használt technológiák 5  
Megvalósítás 9  
Az adatmodell leírása 9  
A program felépítése 11  
Fontosabb algoritmusok 14  
Felhasználói dokumentáció 16  
Tesztelési dokumentáció 21  
Továbbfejlesztési lehetőségek 22  
Források 23**

**Előszó**

A program egy, a munkahelyemen a valóságban létező problémát igyekszik orvosolni. Jelenleg a számlák igazolása teljesen manuálisan történik, ami azt jelenti, hogy a papír alapú bizonylatokat – amik vagy postán érkeztek, vagy pedig elektronikus számlák kinyomtatott példányai - utaztatunk a város különböző részén lévő szervezeti egységek között, a pénzügyi munkacsoporttól a számla igazolására illetékes vezető beosztású személyig. Mindez felesleges időbe, energiába és pénzbe kerül. A programom segítségével ez a folyamat kiváltható és teljesen elektronikus úton végezhető.

A program megvalósítja, hogy a gazdasági társaság nevére kiállított postán beérkező papír alapú, vagy pedig a valamilyen internetes csatornán keresztül érkező elektronikus számlákat iktatja, majd pedig az iktatott számlák teljesülését, és ezzel a kifizethetőségét egy arra jogosult személy leigazolhatja.

Első lépésként tárolni kell a számlák „képét”: a már eleve .pdf formátumban érkező elektronikus számlákat, és a papír alapú számlák szkennelésével nyert .pdf file-okat menteni kell egy nevesített mappába.

A következő lépés a számlák iktatása: a számlák adatai - számlaszám, dátumok, nettó érték, áfa érték, bruttó érték stb. - kézi adatrögzítés során bekerülnek egy adatbázisba, és szintén idekerül még a korábban letárolt .pdf file elérési útja is.

Ezután következhet a számlák igazolása: egy megfelelő jogosultsággal rendelkező személy, aki bír azzal az információval, hogy azok a gazdasági események, amelyekről a számlák készültek, valóban megvalósultak-e, lekérdezi az adatbázisból az iktatott, és még igazolatlan számlákat, majd pedig leigazolja, vagy pedig visszautasítja őket.

Végül az illetékes felhasználók lekérdezhetik a számlákat státuszuk szerint, és elvégezhetik velük a szükséges folyamatot; vagyis gondoskodnak az igazolt számlák kifizetéséről, a visszautasított számlákat pedig visszaküldik a számlát kiállító partnernek.

**Fejlesztői környezet**

Rendszerkövetelmények

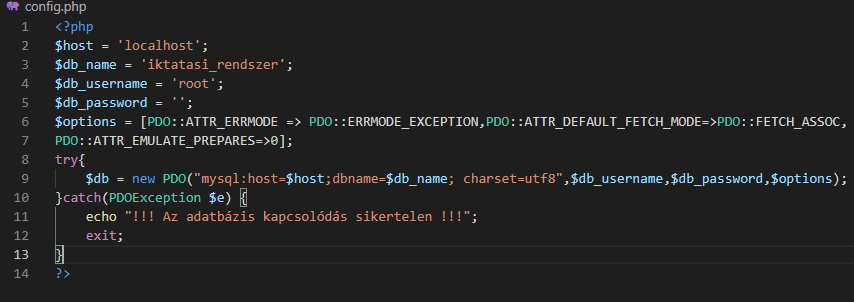
A program fejlesztése Intel(R) Core(TM) i7-8550U típusú, 1.80GHz-es x64-alapú processzorral és 8,00 GB RAM-mal rendelkező számítógépen történt, 64 bites Windows 10 Pro operációs rendszer alatt; másfajta konfigurációval nincs tapasztalat.

Alkalmazott fejlesztői eszközök, programnyelvek

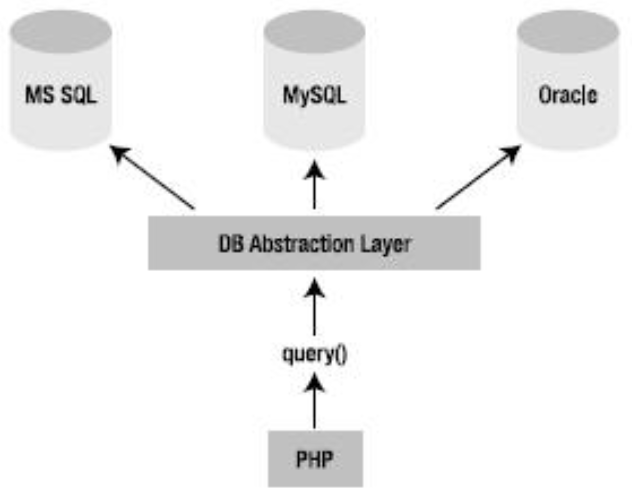
* Xampp webszerver
* Visual Studio Code
* Github
* PHP
* HTML
* CSS
* Bootstrap keretrendszer
* JavaScript

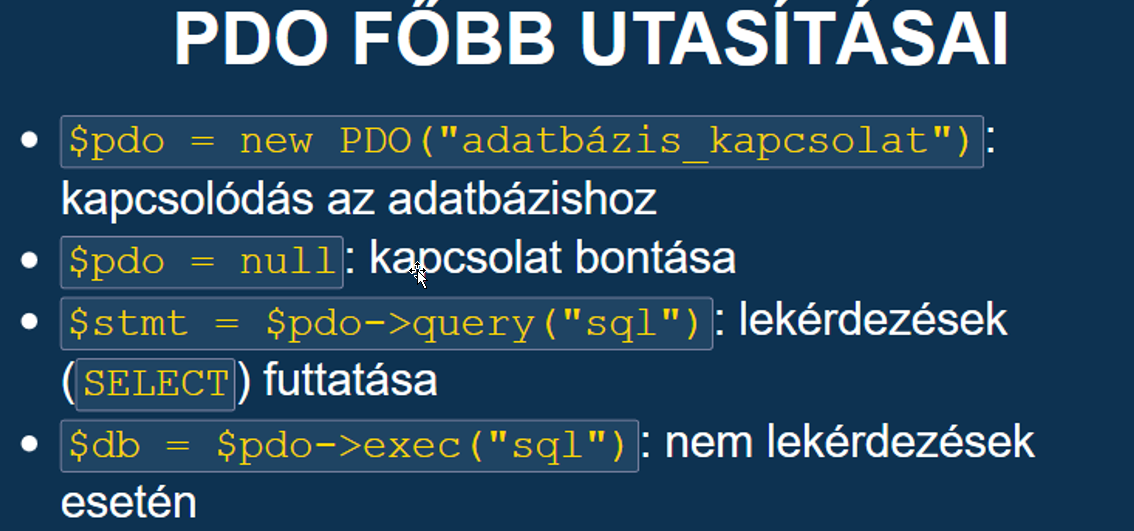
Használt technológiák

1. Adatbázis absztrakciós réteggel csatlakozunk az adatbázishoz, ami objektum orientált adatbázis kapcsolódás, így bármilyen típusú is az adatbázis, ugyanazzal a kóddal elérhető:



Az adatbázis absztrakciós réteg segítségével a kommunikáció központilag menedzselve valósulhat meg. Erre is létezik többféle megoldás, de talán a legelterjedtebb a PDO (PHP Data Objects).





1. Munkamenet kezelés:

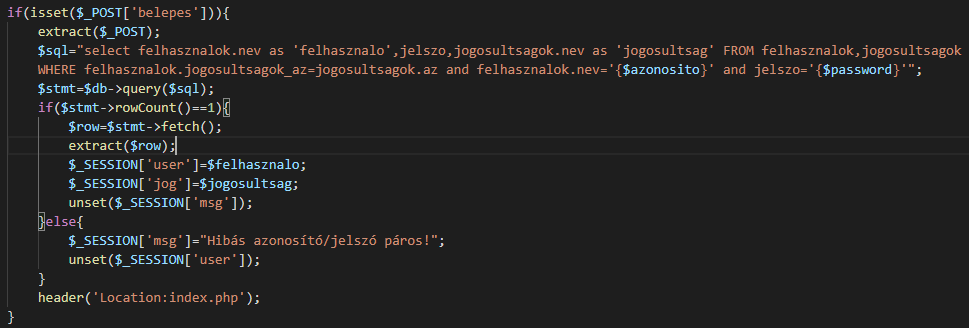
PHP-ban számos függvény és nyelvi elem segíti a munkamenet-kezelést. A munkamenet adatait a PHP a $\_SESSION szuperglobális asszociatív tömbben tárolja. A szkript kezdetekor ebbe tölti be, a szkript végén pedig ennek tartalmát menti ki (alapértelmezetten fájlba).

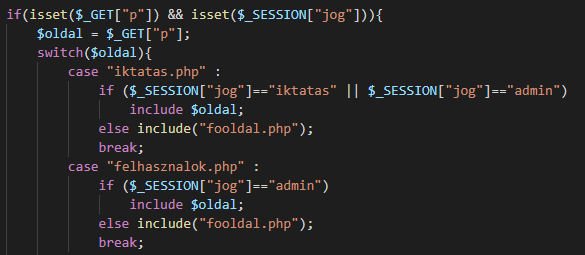
Hasznos függvények:

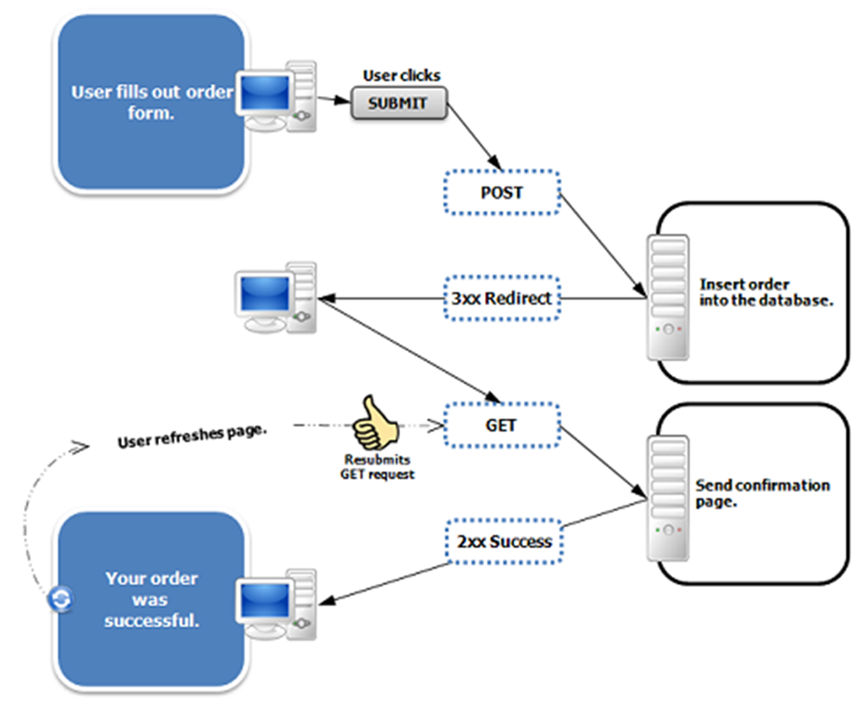
•session\_start(): a munkamenet indítását jelezzük. Minden olyan szkript elején el kell helyeznünk, ahol a munkamenet adataira van szükségünk. A parancs kiadása után a PHP ellenőrzi a sütiben vagy URL-ben érkező munkamenet-azonosítót, és ha van ilyen, akkor a hozzá tartozó adatokat betölti a $\_SESSION tömbbe. Ha nincs azonosító, akkor újat hoz létre.

•session\_destroy(): a munkamenet megszüntetésére szolgál.

Következzen egy példa a $\_SESSION tömb használatára. Az alábbi részlet a felhasználó bejelentkezését vezérlő login.php programból való. Sikeres autentikáció esetén eltároljuk (1. kép) a $\_SESSION tömbben a felhasználó nevét ($\_SESSION[’user’]) és jogosultságát ($\_SESSION[’jog’]), mely meghatározza, hogy a felhasználó a program mely menüpontjait használhatja (2. kép).





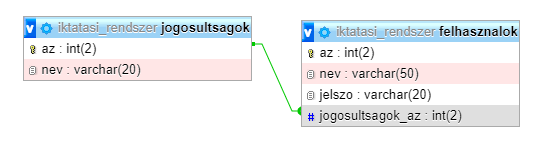
1. PRG programozási technika (post-redirect-get): ezzel akadályozzuk meg az ugyanazon űrlap adatok ismételt elküldését.
2. 

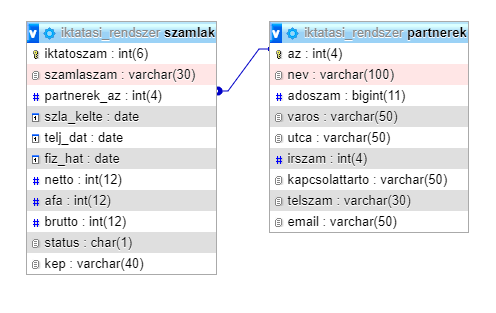
**Megvalósítás**

Az adatmodell leírása

A felhasználók tárolásához az alábbi 2 adattáblára van szükség. A „felhasznalok” tábla tartalmaz egy „az” nevű azonosítószámot, ami egy futó sorszám, az egyedi azonosító, a program használatának szempontjából nincs jelentősége; egy „nev” nevű mezőt, ami a felhasználó nevét tárolja; egy „jelszo” mezőt, amibe a jelszó kerül, és egy „jogosultsagok\_az” nevű mezőt, ami a felhasználóhoz tartozó jogosultság azonosítóját tárolja; megteremti a „felhasznalok” és a „jogosultsagok” táblák közötti kapcsolatot, és vezérli, hogy az egyes menüpontokhoz melyik felhasználóknak van hozzáférése.

A „jogosultsagok” tábla 2 mezőt tartalmaz: az „az” mező a jogosultság sorszáma, az egyedi azonosító; a „nev” mező pedig a jogosultság megnevezését tartalmazza.

A „partnerek” tábla tartalmazza a beszállító cégek adatait. Az „az” mező egy futó sorszám, egyben az egyedi azonosító, és a „számlák” táblával a kapcsolatot megteremtő mező is. A „nev” mező a partner nevét, az „adoszam” az adószámát, a „varos”, „utca” és „irszam” – irányítószám - mezőkben a partner címét tároljuk; a „kapcsolattarto” mezőben annak a nevét, aki a partner céget képviseli; a „telszam” mezőben a telefonszámát, az „email” mezőben pedig az e-mail címét.

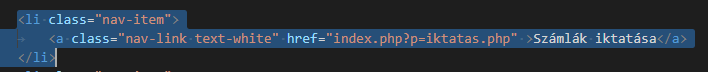
A „szamlak” tábla tartalmazza az iktatott számlák adatait. Az „iktatoszam” a tábla egyedi azonosítója. A „partnerek\_az” a számla kiállítójának azonosítója az adatbázisban; emellett ez a mező teremti meg a kapcsolatot a „partnerek” táblával. A következő mezők a számláról kapnak értéket: „szla\_kelte” a számla kiállításának dátuma; a „telj\_dat” a teljesítés dátuma; a „fiz\_hat” a fizetési határidő; a „netto”, „afa” és „bruttó” mezők pedig a számlán található érték adatokat tárolja. A „status” mező a számla „állapotát” tartalmazza, ami lehet: „N” iktatott, „I” igazolt és „V” visszautasított. A „kep” mező a számlához tartozó pdf file nevét tartalmazza.

A program felépítése

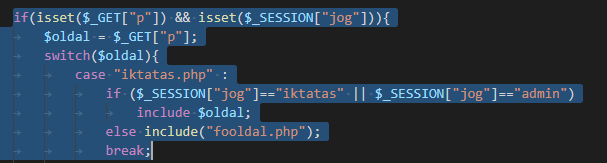
Az oldal betöltődésekor az index.php nevű program indul el, ami önmagában vezérli az egész program működését. Itt történnek meg az alapvető beállítások, ez felelős a menüpontok megjelenítéséért, tartalmazza azt a jogosultság rendszert, amely engedi, vagy tiltja egy-egy menüpont használatát, és vezérli az egyes menüpontok mögött futó php programok betöltését. Az index.php betöltés után a login.php-ra irányít át, ami a felhasználó bejelentkezését végzi. A felhasználónévhez tartozó jogosultság egészen a kijelentkezésig meghatározza, hogy a felhasználó mely funkciók használatára jogosult. Ennek megtörténte után a felhasználó már „válogathat” a menüpontok közül.

A menüpontok kiválasztásakor történteket a következő programrészek irányítják (tételezzük fel, hogy a felhasználó a „Számlák iktatása” menüpontot választotta):

Ez maga a menüpont, ami valójában egy hivatkozás, ami egy lista elemeként jelenik meg a képernyőn. Kattintáskor az index.php hívódik meg úgy, hogy a „p” változó az iktatas.php értéket kapja meg.



A $\_GET tömb „p” eleme ($\_GET["p"]) - aminek értéke átadódik a $oldal változónak -, tartalmazza annak a programnak a nevét, amit a felhasználó a kiválasztott menüponttal aktiválni szeretne (esetünkben ez az iktatas.php). A $\_SESSION tömb „jog” nevű eleme ($\_SESSION["jog"]) tárolja a felhasználó jogosultságát. Egy feltételes elágazás vizsgálja, hogy az egyes, a $oldal változó által meghatározott php programok futtatthatók-e (példánkban az iktatas.php akkor futtatható, ha a bejelentkezett felhasználó jogosultsága „iktatas”, vagy pedig „admin”). Ha a felhasználónak nincs megfelelő jogosultsága, akkor a fooldal.php töltődik be.



A „rendszer” több nagyobb egységre osztható fel a hozzá tartozó php programok alapján. A login.php-ről már fentebb esett szó. A felhasznalok.php – mely a felhasználók kezelését teszi lehetővé -, a partnerek.php – ahol a beszállító cégek adatait tarthatjuk nyilván - és az iktatas.php – minek segítségével rögzíthetjük a számlákat az adatbázisba – hasonló elven működnek, mindhárom a CRUD programozási technikát alkalmazza (az iktatas.php-t láthatjuk).

<?php

ob\_start();

require\_once 'config.php';

if(isset($\_GET['editId']))

    include "iktatas\\edit.php";

else if(isset($\_GET['deleteId']))

    include "iktatas\\delete.php";

else if(isset($\_GET['insert']))

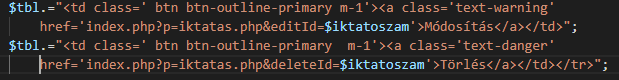
    include "iktatas\\insert.php";

else

    include "iktatas\\iktatasView.php";

?>

Mindhárom programhoz tartozik saját edit, delete, insert és …View „alprogram”, melyek természetesen témánként külön mappákban tárolódnak. Az egyes „alprogramok” attól függően hívódnak meg, hogy a $\_GET tömb milyen értéket tartalmaz. Ha a felhasználó egy Módosít „gombra” kattint, akkor az editId nevű változó értéket kap – ez az érték annak a rekordnak az egyedi azonosítója, amit az adatbázisban módosítani kell -, és ezzel együtt elindul a megfelelő edit.php program. Amikor egy Törlés „gombon” történik kattintás, akkor a deleteId megkapja a törlendő rekord egyedi azonosítóját, és ezzel a paraméterrel hívódik meg a megfelelő delete.php.



Új elem felvitele „gombon” történt klikkelés esetén az insert.php indul el, paraméter nélkül.



Egyéb esetben a …View.php hívódik meg, ami kiírja a képernyőre táblázatos formában a megfelelő adattábla tartalmát.

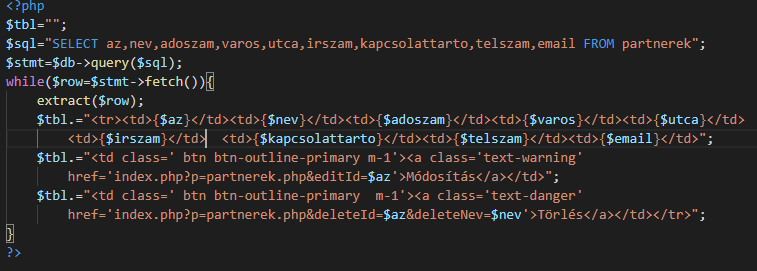
Az igazolas.php – mely az iktatott számlák igazolását valósítja meg -, ettől eltérően működik. A menü futtatásakor képernyőre kerül az összes olyan számla, aminek a státusza iktatott. Ez azt jelenti, hogy az adatbázisban a „status” mező értéke ’N’. A felhasználó két lehetőség közül választhat, és ehhez két „gomb” áll rendelkezésére minden számlánál, ami két külön php programot hív meg: az egyik az „Igazolás” – igazol.php -, mely hatásásra a „status” értéke ’I’ lesz, a másik pedig a „Visszautasítás” – visszautasit.php -, minek hatására a „status” mező értéke ’V’-re változik.

Azokat a számlákat, amik az előbbi igazolási folyamaton átestek, két külön menüpontban láthatjuk, ezek az: „Igazolt számlák” – igazolt.php – és a „Visszautasított számlák” – visszautasitott.php.

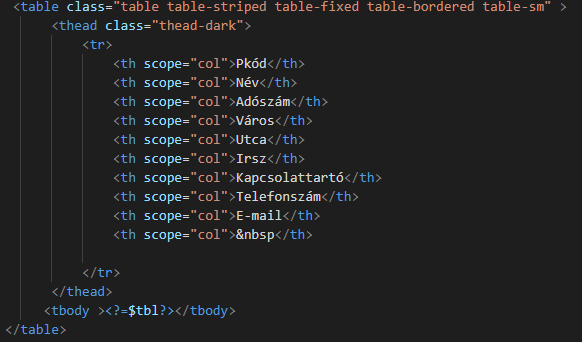
Fontosabb algoritmusok

A következő algoritmus, programrészlet több helyen is előfordul a programban. Feladata egy, az adatbázisból lekérdezett eredményhalmaz táblázatos megjelenítése. Az alábbi részlet a partnerekView.php programból származik.

A program elején deklarált $tbl változó tartalmazza majd magát a táblázatot, annak soraival, oszlopaival, celláival és adataival. Az $sql nevű változóban tároljuk le azt az sql parancssort, amivel lekérdezhetjük a kívánt eredményhalmazt az adatbázisból – ezesetben a partnerek tábla minden rekordját. A lekérdezett eredményhalmaz a $stmt változóban tárolódik, ennek tartalmát olvassuk végig egy while ciklus segítségével. A ciklus minden egyes futásakor a fetch() függvény segítségével bekerül a $row változóba az $stmt-ben tárolt adathalmaz éppen következő sora. Ennek a sornak a tartalmát „bontjuk ki” az extract utasítással azért, hogy az egyes adatokra az adatbázisbeli nevükkel tudjunk direkt módon hivatkozni. Következik a táblázat megalkotása: a $tbl változóhoz hozzáadjuk a táblázat következő sorát, vagyis a <tr> és <td> html tagekkel létrehozzuk a következő sort és annak celláit, a cellákba pedig betöltjük az adatbázis megfelelő mezőinek értékét a nevükre, mint változókra hivatkozva. Végül minden táblázatsor végére kerül egy Módosítás és egy Törlés nevű, bootstrap-pel formázott gomb.

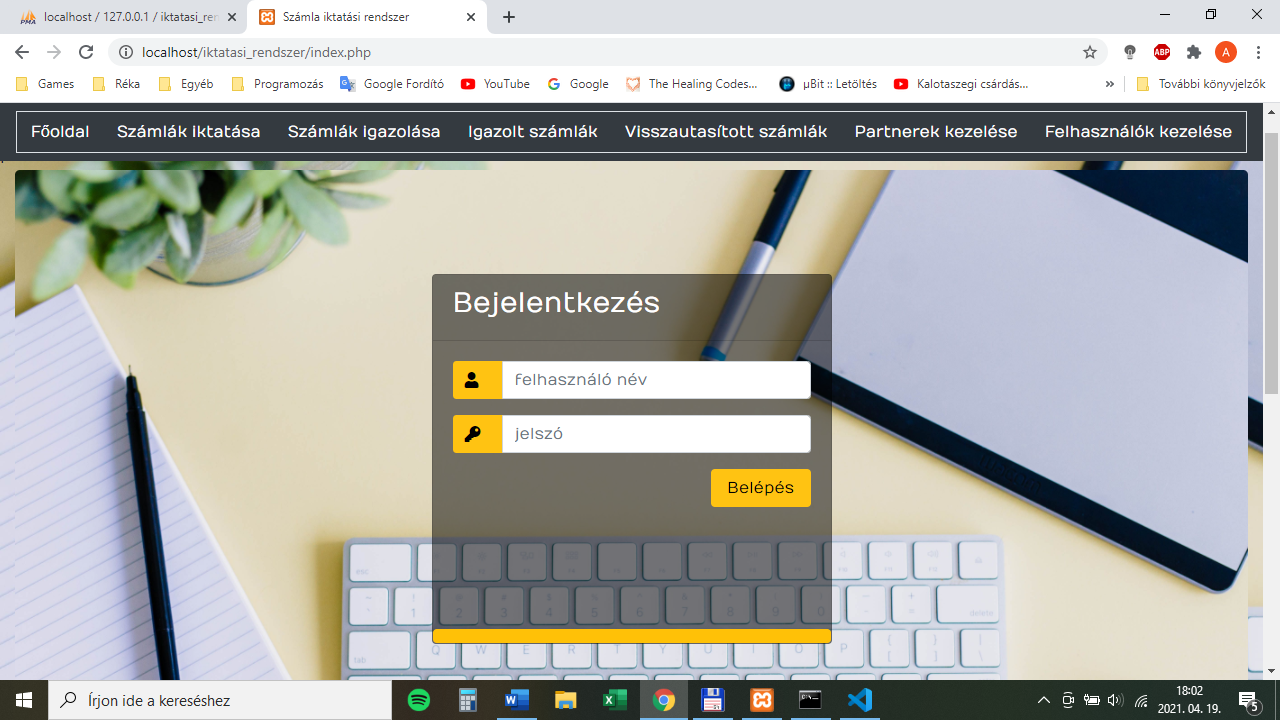


A következőkben láthatjuk azt a html kódot, ami „megrajzolja” az előbbi táblázat fejlécét, majd pedig megjeleníti a táblázat sorait, vagyis a táblázat <tbody> tagébe betölti annak a $tbl változónak a tartalmát, amiben fentebb már futásidőben létrehoztuk és letároltuk az egyes sorokat.



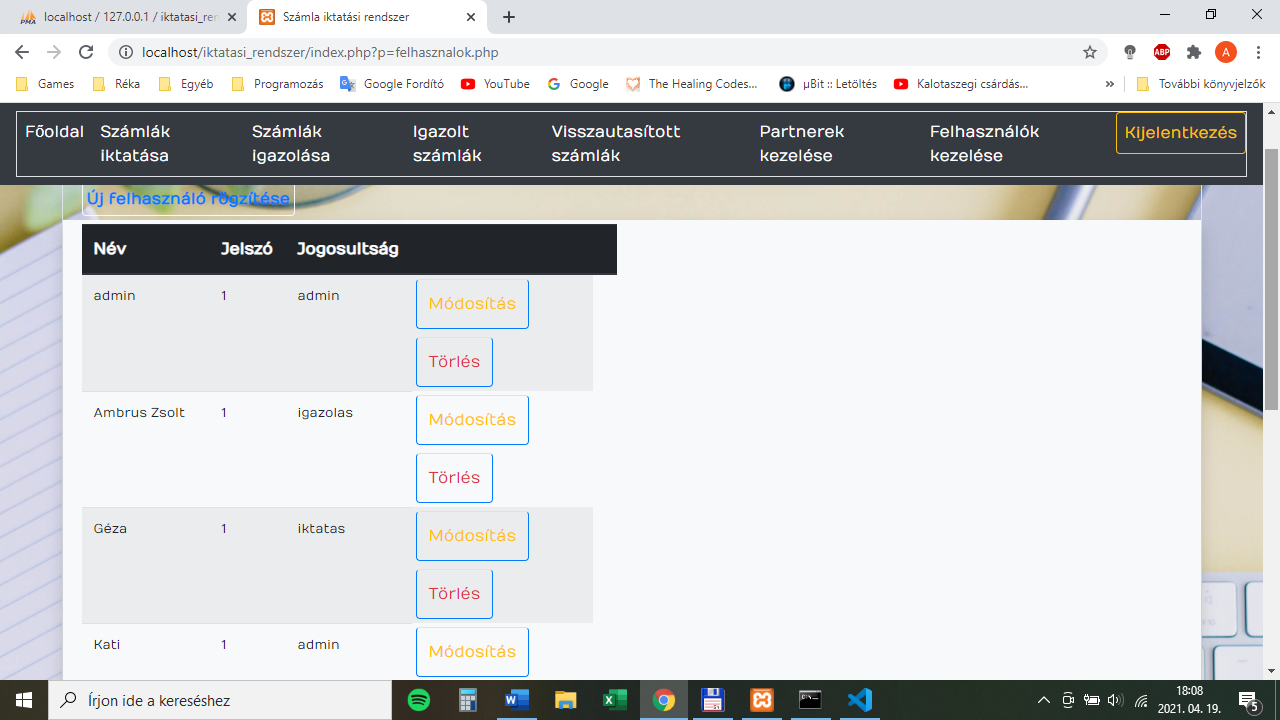
Felhasználói dokumentáció

A program indítása után a bejelentkezési képernyő fogadja a felhasználót, itt kell megadnia a felhasználónevét és a jelszavát. Minden felhasználónévhez hozzá van rendelve egy jogosultság, ami meghatározza, hogy a bejelentkezés után melyik menüpontok használhatók.

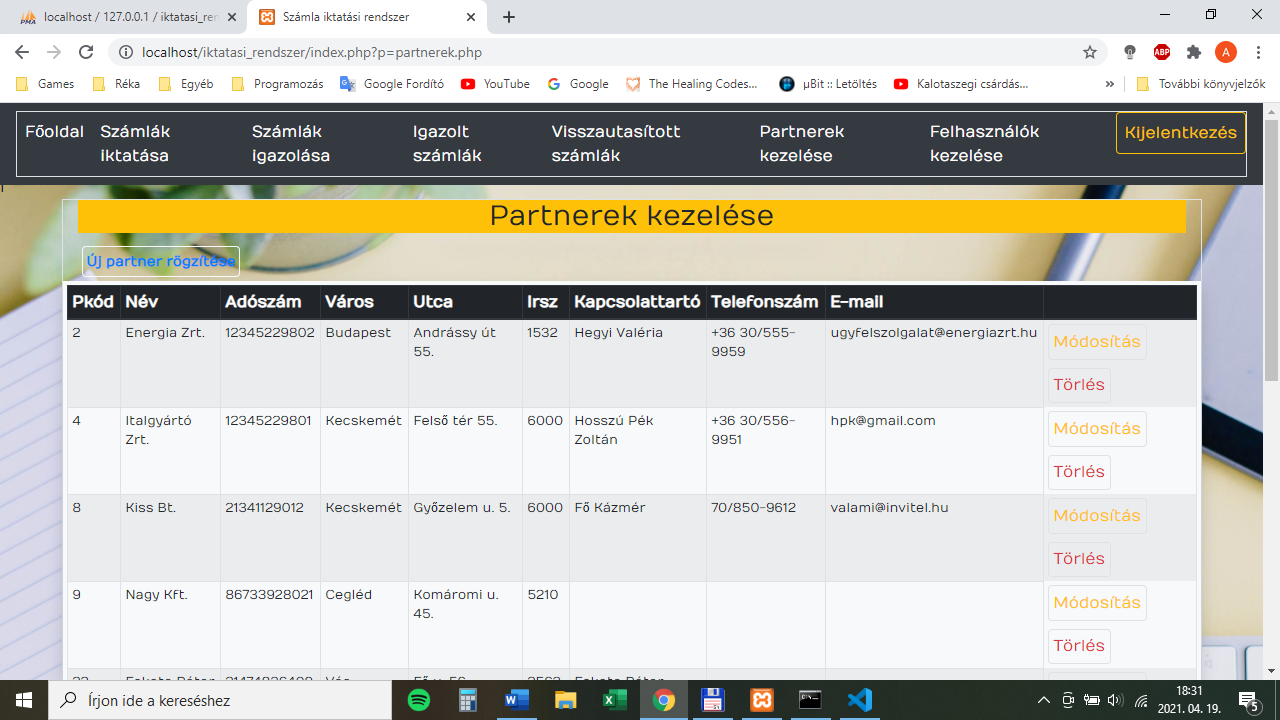


Bejelentkezés után jelenik meg a Kijelentkezés gomb, ezzel lehet a programból kilépni. Hatására a felhasználó visszakerül a bejelentkezési ablakra.

A Felhasználók kezelése menüpont csakis admin jogosultsággal használható. Itt láthatjuk a rendszerben rögzített felhasználóneveket a hozzájuk tartozó jelszóval és jogosultsággal. A Módosít gomb segítségével lehet egy felhasználó adatait megváltoztatni, a Törlés gombbal törölni, az Új felhasználó rögzítése gombbal pedig egy új felhasználó adatait berögzíteni.

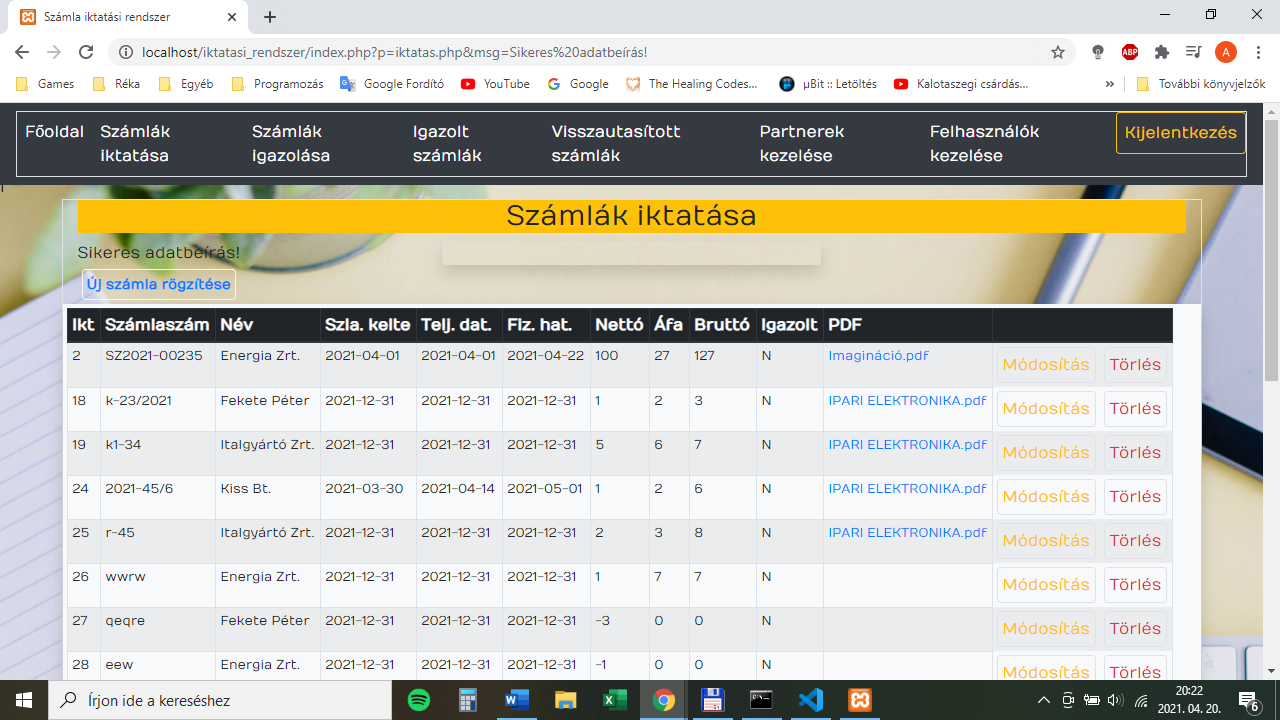


A Partnerek kezelése menüpontban lehetséges a szállító partnerek adatait feltölteni. Belépéskor láthatjuk a létező partnerek adatait. Lehetőség van a Módosításra, Törlésre – kivéve, ha a partner azonosítójával már rögzítettünk számlát, mert ebben az esetben a partner nem törölhető -, és tudunk új partnert is rögzíteni. Rögzítéskor a partner azonosítóját a program automatikusan osztja ki. Rögzítés vagy Módosítás esetén a program ellenőrzi, hogy a megadott adószám létezik-e már az adatbázisban; ha igen, akkor figyelmeztet, és nem engedi a felvitelt.

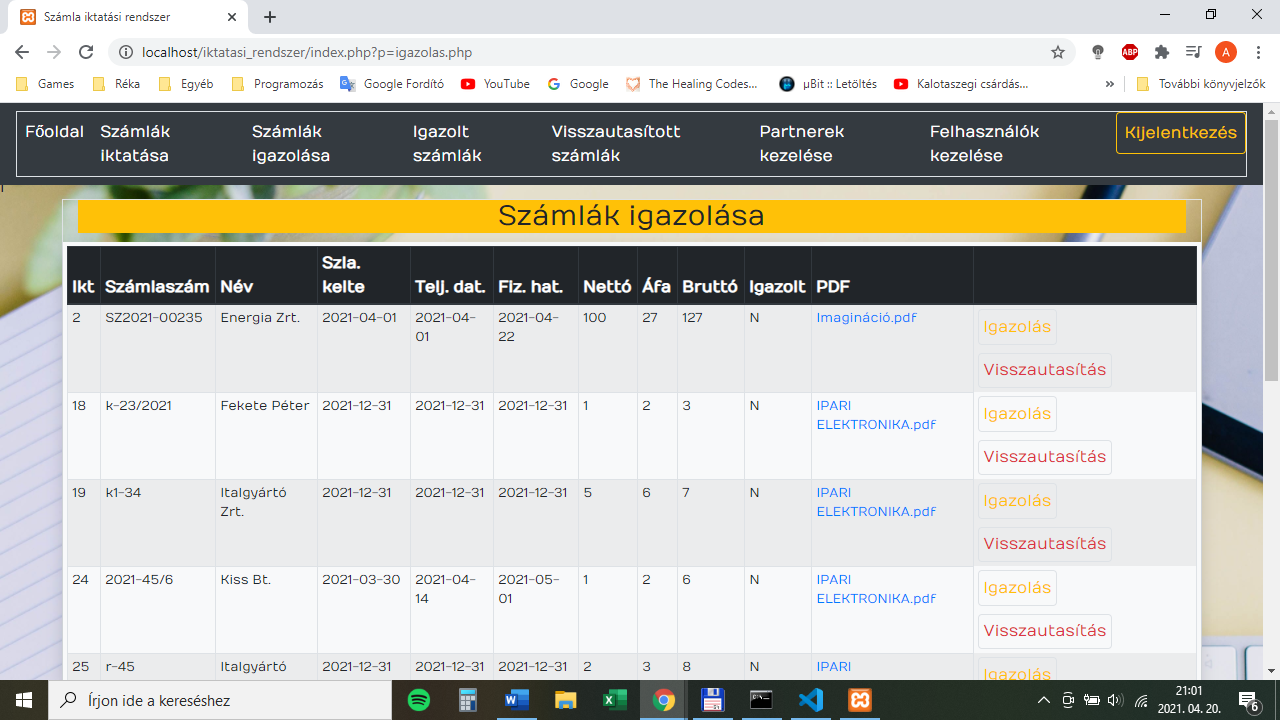


A Számlák iktatása menüpontba való belépéskor az összes, még nem igazolt és nem visszautasított számlát láthatjuk, ezeket van lehetőség módosítani, vagy törölni a megfelelő gombokkal - az igazolt, vagy visszautasított számlák nem törölhetők és tartalmuk sem megváltoztatható; ezeket két külön menüpont alatt tudjuk megnézni.

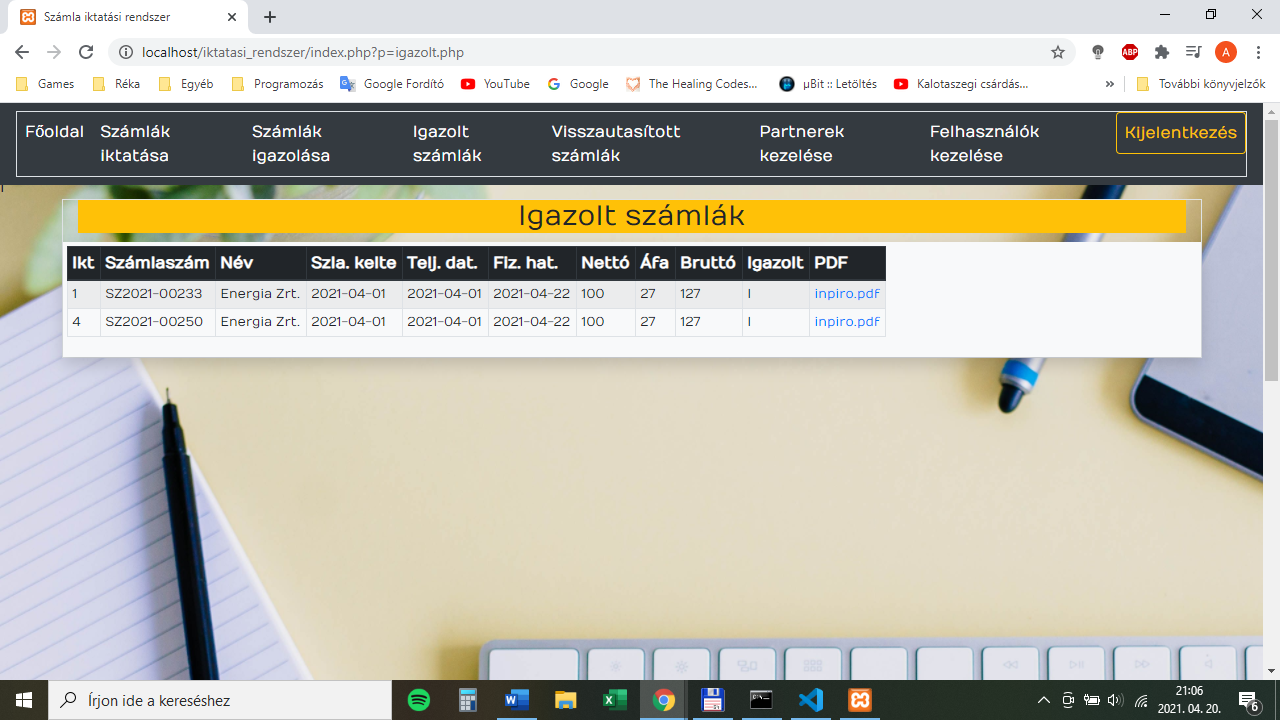
Az iktatószám a számlák egyedi azonosítója, ezt a program osztja automatikusan, értéke nem megváltoztatható. Új számla felvitelekor, vagy pedig meglévő módosításakor a partner nevét egy legördülő listából választhatjuk ki. A PDF mezőbe egy pdf file neve kerül be, amit a felhasználónak kell tallózással kiválasztania. A listában lévő file nevére kattintva ez a pdf file egy új böngésző ablakban nyílik meg. Az Igazolt mező értéke iktatáskor automatikusan ’N’ lesz, ez jelenti azt, hogy a számla nincs igazolva. Az összes többi mezőt a számláról kell feltölteni.



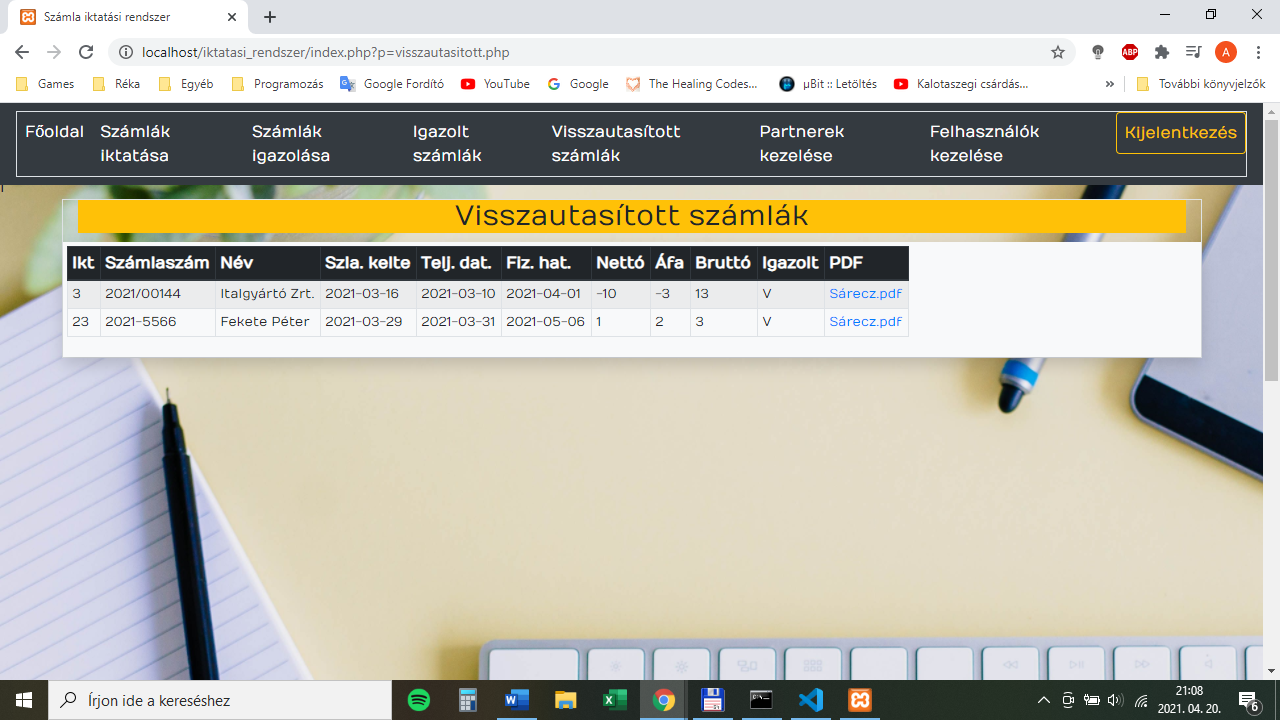
A számlák igazolása menüpont alatt az összes olyan számlát láthatjuk, amelyeknek a státusza ’N’, vagyis nem igazolt. A menüpont használatához külön jogosultság szükséges. A számla igazolója a PDF mező tartalmára kattintva előhívhatja és megnézheti a számla képét, és eldöntheti, hogy a számla tartalma megvalósult-e; ha igen, akkor az Igazolás gombbal leigazolja – ezzel a számla státusza ’I’-re változik -, vagy pedig valamilyen kifogása van a számla ellen, akkor a Visszautasítás gombbal elutasítja a bizonylatot, aminek a státusza ezzel ’V’-re változik.



Az igazolt számlák menüpont egy lekérdezés, ami az ’I’, igazolt státuszú számlákat hozza fel egy listára. Az itt látható számlák kerülhetnek kifizetésre.

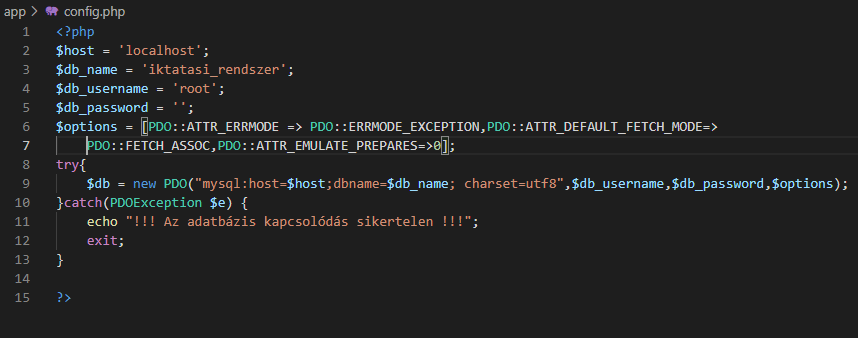


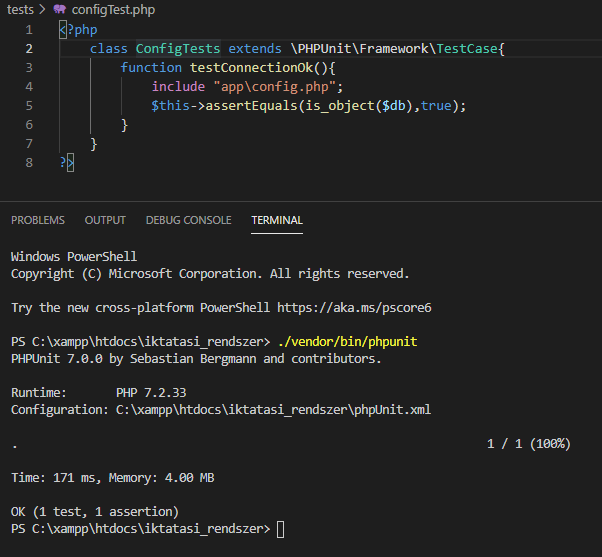
A Visszautasított számlák menüpont szintén egy lekérdezés, itt léthatók azok a bizonylatok, melyeket a számlaigazoló visszautasított. Ezeket a számlákat vissza kell küldeni a partnernek, aki kiállította.



Tesztelési dokumentáció

Az alábbiakban az adatbázis kapcsolódásért felelős config.php sikeres tesztelését láthatjuk egy egységteszt segítségével, ami a PhpUnit keretrendszerrel készült.





**Továbbfejlesztési lehetőségek**

1. A program bővítése nyomtatás funkcióval azok számára, akik papíron szeretik az adatokat elemezni. Ezt az Igazolt számlák és a Visszautasított számlák menüpontoknál tudom elképzelni.
2. Az igazolt számlákról készülhetne adatexport egy .csv file-ba, ami importálható lenne egy pénzügyi, számviteli rendszerbe további felhasználásra.
3. A visszautasított számlákról automatikus e-mail küldése az érintett partnernek, egy előre meghatározott sablon e-mail szövegtörzzsel.

**Források**

* <https://www.php.net/>
* <https://www.w3schools.com/>
* <https://github.com/>
* <http://webprogramozas.inf.elte.hu/webprog/ea/02/?print-pdf#/>
* https://mek.oszk.hu/09700/09713/