

# Fejlesztői napló

## 2023-10-09

Sokadik próbálkozásra sem tudtunk installálni a node csomagokat amik a a package.json fileban vannak megadva. Próbáltunk Windowson és próbáltunk linux környezetben is. Azt hisszük, a probléma azzal van, hogy közben update-elve lett a package.json és már nem találja a dependency-ket. Ezért holnap újra próbálkozunk, de más módszerrel. Ezt csak azért írtuk le, hogy ne felejtsek el, hogy hol tartottunk.

## 2023-10-11

Tegnap megpróbáltuk feltenni a node package-eket úgy, hogy felraktuk a linux operációs rendszerünkre egy Xampp-ot és ott hoztunk létre WordPress környezetet. Így már működött az npm install, bár az is kiderült, hogy egy 2019-es package.json fájlban foglalt csomagot telepítettünk, ami kicsit outdate-ed volt. Ezért módosítottuk a package.json fájlt. Ezután már sikeres volt az update.

A mai napon megpróbáltuk ugyanazt a környezetet létrehozni a DevKinsta nevű applikációval. Windows esetén az volt a probléma, hogy a Dockeres virtualizáció miatt egy virtuális elérési útvonalat kapott a node package installálásakor, így nem találta meg a "nem valós", vagyis virtuális helyet. Ezért nem tudta telepíteni a node modulokat. Viszont Linux rendszeren zökkenőmentesen feltelepültek a modulok.

Ezért a .gitignore fájlba be is vettük a node modulokat, hogy azok feleslegesen ne legyenek commitolva a repozitóriumba.

## 2023-11-11

A mai napon azon dolgoztunk, hogy létrehozzuk Vuepress-szel a programcsomag dokumentációs oldalát. Az oldal működőképes, de egyelőre kizárólag lokálisan. A deployálásnál valamiért elakadt a client compile. Nem tudtunk rájönni, mi lehet a probléma. Egyébként is kicsit sérülékenynek tűnik a Vuepress. 4-5 alkalommal próbáltuk, mire sikerült stabil verziót feltennünk a V.1-ből. Természetesen az újabb verziókat is kipróbáltuk, de azok még kevésbé voltak stabilak. Felvettünk magunknak új taskokat a Jira-ba. A programtervek szöveges megírását és a felületek tervezését terveztük meg.

Először leírtuk a működési folyamatokat szóban, aztán algoritmikusan ábráztuk. Ezután ezekhez a lépésekhez megterveztük a különböző képernyőképeket.

További terveink között szerepelt a mai és holnapi napra ezek kivitelezése, valamint a WordPress téma fejlesztése. Ezzel egy időben a bővítmény fejlesztésébe is bele akartunk fogni.

Úgy gondoltuk, hogy már most érdemes összeírni, hogy a WordPress adatbázisából mely táblákra lesz szükség a desktop applikációnál.

## 2024-02-05 - 2024-02-11

Ezen a héten létrehoztuk a vizsgaremek desktop alkalmazást nulláról. A korábbi Kréta applikáció visszafejlesztése nehézkes, időigényes és sok hibalehetőséget tartalmazó opció lett volna. Éppen ezért arra jutottunk, hogy nulláról hozzuk létre az egészet.

Jelenleg a következő oldalak (front end) készültek el:

Bejelentkezés Termék hozzáadása Termék keresése Üdvözlő oldal Már elkezdtek hozzáadni a színeket is, hogy a végső állapot a lehető legjobban hasonlítson a designtervben szereplőre.

Megkezdtek a bővítmény fejlesztését is. Mivel a Sparse Checkout továbbra sem működik megfelelően, ezért a saját Organization-ünk alá kezdtünk el commitokat készíteni. Majd ezt fogjuk átmásolni a vizsgaremek repositoriba. Tehát duplán fogunk dolgozni, de még nem találtuk megfelelő megoldást arra, hogyan lehetne a Sparse Checkoutot megfelelően működésre bírni. Egyelőre nem akartunk vele több időt elpazarolni.

## 2024-02-19

Nos, ez a fejlesztői sorozat tulajdonképpen a fejlesztői naplónk lesz a projekt feladatommal, amivel áprilisig végeznem kell.

A projekt során egy Desktop applikációt készítünk el, amely segít a felhasználóknak egyszerűsíteni termékek feltöltését egy WordPress/WooCommerce webshopba.

Három komponensből, három programból fog állni. Az első egy C# desktop applikáció lesz. Azért C#, mert a feladat az volt, hogy ebben kell elkészülnie. Bár mi Javában szerettünk volna dolgozni, a választás csak névlegesen volt lehetséges. A valóságban a C# volt a kötelező.

Vannak már korábbi naplóbejegyzéseink, amelyeket ide, a poszt végére fogunk beilleszteni, csak hogy tényleg minden itt szerepeljen.

A fejlesztést nehezíti, hogy alapvetően Linux alatt fejlesztünk, a C# alkalmazásnak viszont tartalmaznia kell olyan elemeket, amelyek csak Windows környezetben fejleszthetők.

További nehézséget jelent, hogy egy külön GitHub repóba kell feltöltenünk a kódokat, ezért külön user/email cím használata is kötelező.

A további nehézségekről is fogunk még írni, illetve megosztjuk, milyen furcsa tapasztalataink voltak. Mindenesetre ez most itt a bevezető.

A cél az, hogy minden nap 2 órát tudjunk rászánni a projektre (minimum), és hogy a fejlesztői naplót itt vezessük, amit később a fejlesztés repójába is fel fogunk tölteni. Így párhuzamosan tudunk haladni, és követhető lesz a fejlesztésünk is.

## 2024-02-20

Kezdjük az elején. Legyen ez az igazi intro. Tehát a feladat kritériumai. Desktop applikáció. Webes applikáció, ami HTML, CSS, Bootstrap és JavaScript-et használ. Legyen API kapcsolat, autentikáció, adatbázis és reszponzivitás.

Egy programcsomagra gondoltunk, ami egyszerűvé teszi asztali környezetből a termékek feltöltését egy WooCommerce webshopba.

Három részre bontottuk.

Desktop applikáció, amit sajnos csak C#-ban lehet elkészíteni. Itt az MVVM megközelítést alkalmazzuk. Ez lesz a Desktop applikáció, aminek lesz egy adatbázisa, ami tárolja az egyes termékeket. Lesz egy WordPress bővítmény, ami elvégzi az autentikációt, az adatbázis és képek szinkronizációját, valamint természetesen a cronjob-ot. Lesz egy Bootstrap alapú WordPress téma, ami minden képernyőméreten jól használható. Tehát extraként bejön még a PHP. Most már, hogy tisztázódtak a projekt kritériumai, elmesélem, mi történt ma. Létrehoztam a Desktop applikációhoz az AddProductView összes mezőjét. Létrehoztuk a ModifyProductView mezőit, valamint a GlobalMenu.xaml-t a globális menühöz.

Ezen kívül még a Login oldalt próbáltuk rendbe tenni, de amikor láttam a 11 hibát, akkor azt gondoltuk, hogy mára ennyi, majd folytatjuk.

Holnap ezeket szeretnénk rendbe tenni. A globális menüt, a hibákat, illetve szeretnék már létrehozni egy adatbázist is a szinkronizálendő menüvel együtt.

## 2024-02-26

Néhány napot lemaradtunk a fejlesztéssel. Egyszerűen annyi időt emésztett fel a munkánk és az iskolai teendők, hogy sehogy sem jutott időnk a projekt megvalósítására.

Mindenesetre...

A mai napon létrehoztuk az adatbázist a desktop alkalmazásunkhoz. Az adatbázis 3 táblát tartalmaz: User, Websites és Products. Hiszen a cél az, hogy a desktop alkalmazásban létrehozott termékeket szinkronizáljuk majd a weboldalunk adatbázisával.

Ezekhez rendelkezésünkre állnak az adattípusok és minden más, ami szükséges. Természetesen ez nem a végleges változat, de feladatként megkaptuk, hogy mutassuk be az adatbázisunkat.

Létrehoztuk a kapcsolódó EK diagrammot is. Ami azóta korrigálva lett, és két további tábla került beszúrásra, hogy a funkciók kezelése logikusabb és egyszerűbb legyen.

A célunk a hétre, hogy befejezzük a FrontEnd részét a Desktop alkalmazásnak és valamilyen alap kapcsolat is legyen már az adatbázissal.

Szeretnénk, ha a design már a helyére kerülne és hogy egy stíluslapon lehessen formázni a program elemeit. Részben ezért is szerettük volna Java-ban megírni a programot, mert akkor a Desktop alkalmazáshoz, a bővítményhez és a WordPress témához is BootStrap-et tudtunk volna használni. De most már ezzel kell tovább mennünk.

Holnap jelentkezünk a további fejlesztésekkel.

## 2024-03-02

A következő néhány bejegyzés (a 10. napig) gyors egymás utánban fog következni. Nem volt időnk megírni őket részletesen, csak bedobtunk néhány kulcsszót, hogy tudjuk, mit csináltunk aznap. De most, utólag sikerült kifejtenuk ezeket a bejegyzéseket, és ezeket fogjuk most feltenni.

A mai napon feláldoztuk az asztali gépeinket is, hogy azokon is fussanak a Windows. Mechanikus dual boot-hoz beletettük a korábbi laptopjaink SSD meghajtóit a gépekbe, ahol már Windows futott. Próbáltunk kialakítani a Laptopjainkon lévő fejlesztőkörnyezetet, és ott klónozni a vizsgaremek repository-t. Itt hibát írt ki, ezért a vasárnap többi részét arra fordítottuk, hogy töröljük és újra telepítsük a Visual Studio-t. Így már működőképesé vált a projekt.

Tehát a mai nap tulajdonképpen megint a Windows-os környezet rejtelseinek feltárására ment el. Nem véletlen, hogy nem szeretjük ezt az operációs rendszert. A későbbiekben tapasztaljuk majd (csak mert ezt utólag írjuk a bejegyzéshez), hogy arra is képes a Microsoft terméke, hogy míg az egyik gépen hiba nélkül sem fut le a fejlesztés alatt álló program, addig a másik gépen sok-sok hibával is elindul.

Csak a Linux.

2024-03-03

A mai napon mindent UserControl-lá alakítottunk, kivéve a LoginPage és MainWindow XAML fájlokat. Ma megpróbáljuk összekötni a különböző elemeket és a főmenüt, hogy már összelinkelve működhessen a navigáció. Ehhez utánanézzünk, hogy mit és hogyan kell.

Illetve a színeket szeretnénk véglegesíteni, valamint a mezőket, gombokat is. Szeretnénk, ha egységes lenne és jelentős mértékben hasonlítana ahhoz, ami a design tervben szerepel. Mind az oldalak tekintetében, az elrendezésben és a színeiben.

Erre egy style.xaml-ben fogjuk elhelyezni az ezekre vonatkozó stílusokat. Update: Az oldalak lekerekítésével megküzdöttünk, mert nehezen lehetett beállítani. Végül megtekintve a végeredményt, elég csúnyán nézett ki a UI. Ezért elvetettük, hogy az oldalak és általában a program sarkainál a rádiusz egy kicsit kerekdedebb lesz.

A gombok színeivel még küzdünk, ugyanis ha beállítunk egy fő színt a betűkhöz, akkor minden más is átveszi. Tehát a gombok betűi is. Ez azért nem jó, mert az alap betűszín sötét, míg a gombokon elhelyezett felirat pedig világosnak kellene lennie. Persze, ha lehetne használni CSS-t, azon belül is Bootstrap-ot, ami elvárás a programok írásakor, akkor simán meg lehetne oldani elég sok mindent.

Elég sokat olvastunk utána, és csak nagyon nehezen kivitelezhető, hogy "megegye" a Bootstrap osztályokat a WPF, sőt, ha jól tudunk, ezt csak a korábbi verziókban lehetett megoldani. Még egy indok, amiért Java-ban szerettünk volna megcsinálni a fejlesztést.

Mindegy, most már ez van, ezt kell szeretni. Tehát még a színekkel birkózunk, de az is lehet, hogy átmenetileg elengedjük a dolgot, és a végén, ha lesz még időnk, akkor visszatérünk a problémák kijavítására.

2024-03-05

A mai napon átstruktúráljuk az oldalt és összekötjük az elemeket. Ami nem érthető, az az, hogy Page-ek helyett elkezdünk UserControl-t használni, mert azt az információt kaptuk, hogy ezekkel megszűnnek a navigációs nyilak a bal felső sarokban. Ennek ellenére továbbra is megjelennek, így most azzal próbálkozunk, hogy a különböző View-kat Window alapokra helyezzük. Ez később egyébként hasznos is lehet, mert tervben van az is, hogy egy termék módosításakor ne történjen ablakváltás, hanem kényelmesen egy felugró ablakban lehessen szerkeszteni a termékeket.

Tehát ma beillesztjük a GlobalMenu-t a MainWindow-ba. A MainView-t töröljük, mert nem láttuk értelmét. Ugyanez történik a WelcomePageView-val is, mert felesleges volt egy külön oldalt használni. Innentől kezdve törekszünk arra, hogy a program minél ésszerűbb, jobban struktúrált és felesleges elemektől mentes legyen.

További probléma, hogy nem tudjuk megjeleníteni a logót, hiába próbáltunk mindenféle elérési útvonalat.

Tehát a problémák:

A logó nem jelenik meg. Az oldalakat UserControl-ról át kell helyoznünk Window alapokra, mert továbbra is megjelennek a navigációs nyilak. --> Ezekből Page-ek lettek, mert a Window-nál nem működött... Update: A logót sikerült beillesztenünk. Most már működik. Az egyszerű módszer az volt, hogy behívtuk a design felületen az elemre kattintva, majd kiválasztva az elérési útvonalat. Így már működött.

Az oldalak közti navigáció is működik. Illetve az adatbázist is sikerült csatlakoztatnunk, így már a termék felvitelekor le is tárolja az adatbázisban. A probléma csak az, hogy egyelőre felülírja a korábbi terméket. De már látjuk a megoldást.

Valamint az egész Search Product ablakot inkább beépítjük a kezdőlapba, hiszen ott egyébként sincs semmi. Az AddProduct pedig majd felugró ablakként fog megjelenni, így az "Add Product" gombhoz hozzá tudjuk adni az adatbázis lezárás funkciót. Tehát megszakítja majd minden egyes alkalommal az adatbázis kapcsolatot.

A design-ban is módosítottunk a színeken, a bordereken és az elrendezésen. Kezd végre alakot öltetni a program.

alt text

Holnap szeretnénk a Search Product ablakban megjeleníteni a rögzített termékeket is. És akkor már azt hiszem, elkezdődhet a bővítmény fejlesztése is.

## 2024-03-06

A mai napon egy kicsit átváriáltuk a Desktop applikációt. Létrehoztunk egy NewProductWindow-t az AddProductView helyett. Így egy felugró ablak jön létre a termék hozzáadásánál, ahelyett, hogy elnavigálnánk bárhova. Illetve a Search ProductView-nál szintén létrehoztunk egy EditProductWindow-t, ami a "Modify" gombra kattintva ugrik fel, és majd az adott terméket lehet módosítani vele. Így szintén nem kell elnavigálni a termékek listájáról, de egyszerűbbé teszi a felhasználást.

A mentéskor természetesen el is menti a módosításokat. Az SQLite egy tökéletes választás az átmeneti állapotra. Később majd szeretnénk a backendet is megírni hozzá az API kapcsolat és az összekötés miatt, de egyelőre tökéletesen kiszolgálja azt, amire szükség van.

Amíg teszteljük a NewProduct funkciót, addig egy plusz menüpontot hoztunk létre. Nem is törölöm a korábbi Add Product menüpontot, amíg ez nem működik.

Továbbá szeretnénk egy-egy címsort adni minden oldalhoz, ablakhoz, hogy tudja a felhasználó, hogy éppen hol jár.

Még egy extra dolog, hogy meg kell oldanunk, hogy a SearchProductView esetében legyenek apró checkboxok a termékek mellett, amik inkább radio buttonként funkcionálnak.

Ezekre azért van szükség, hogy a felhasználó kijelölhesse, melyik terméket szeretné szerkeszteni. Vagyis csak azt tudja kijelölni. Csoportos szerkesztéssel egyelőre nem szeretnénk bonyolítani a programot. Inkább legyen stabilan működő. De már majdnem a finishben.

## 2024-03-10

A mai napon ismét elővettük a Woosync WordPress bővítmény mappáját.

Ez a plugin fogja szolgálni az autentikációt és szinkronizációt a Desktop applikáció (C#) és a WordPress adatbázisa között.

Utoljára 2023 októberében commitoltunk a bővítmény repositoryjához, mert azóta rengeteg minden történt. Természetesen foglalkoztunk a bővítménnyel, de most az elkövetkező napokban aktívan szeretnénk fejleszteni. Így azzal kezdtük, hogy a korábbi kódot töröltük és átstrukturáltuk a repositoryban lévő mappákat, fájlokat.

Az index.php fájlal kezdtünk, ahol a főbb funkciókat hoztuk létre, majd ehhez kapcsolódóan a különböző osztályokat.

Erre azért volt szükség, mert így nulláról kezdve ennyi idő után végre képbe kerültünk a bővítménnyel, és nem lesz benne annyi sallang.

Lesz egy Autentikációs osztály, ahol az API kulcsot lehet legenerálni.

Lesz egy Cron osztály, ahol a szinkronizációval kapcsolatos beállítások lesznek, illetve lesz egy Intro rész, ahol a bővítményt és a működést mutatjuk be. Némi leírással és linkekkel, például az asztali alkalmazáshoz és a témához. Nagyjából ennyi.

Ma ezeket sikerült abszolválnunk, holnap megyünk tovább a többi részével.

## 2024-03-12

Ma létrehoztuk az autentikációs funkciót és osztályt. Nem volt könnyű működésre bírni, mert miután megírtuk a kódot az index.php-ban és a kapcsolódó js fájlban, rájöttünk, hogy érdemes lenne mindent külön szervezni. Tehát minden osztályt, js fájlt, css-t és minden mást külön-külön rendeztünk. Így a program struktúrájában és modularitásában is "szebb" lett volna.

A különszervezés okozta a legnagyobb gondot, mert mire összekapcsoltuk az osztályokat és funkciókat, addigra jó sok idő eltelt. Küzdelmes volt, de megérte. Az alábbiak szerint alakult a jelenlegi struktúra:

Woosync | index.php | README.md | uninstall.php | |——css | (CSS files) | |——js | (JavaScript files) | |——assets | (Asset files such as images, fonts, etc.) | |——includes (PHP include files)

Természetesen ez még változhat.

Az includes mappában található a woosync-authentication.php. A lényeg, hogy ez a funkció képes arra, hogy generáljon egy random API kulcsot, amit el is tárol az adatbázisban. Ezt a kulcsot meg is jeleníti a bővítmény admin felületén. Van lehetőség másolásra egy gomb segítségével. Ez a vágolapra helyezi a kulcsot.

Illetve van lehetőség arra, hogy Reset-eljük a kulcsunkat. Ez akkor lehet hasznos, ha valami miatt szeretnénk ha a kulcs nem működne tovább. Például jogosulatlan felhasználás esetén. Ekkor a kulcs az adatbázisból is törlődik.

Ma ennyit sikerült abszolválnunk. Holnap újult erővel folytatjuk.

## 2023-03-13

A Mai napon létrehoztuk a Cron osztályt, pontosabban a Settings menüpontot a bővítményben (woosync-settings.php). Ez arra szolgál, hogy itt lehet beállítani a Cron Job-ot, hogy mikor fusson le: havi, heti, napi vagy óránkénti intervallumokban. Illetve van egy "Minden percben" lehetőség is, amelyre kattintva megjelenik egy input mező, ahol számokkal lehet megadni, hány percenként fusson le a cronjob. Ez utóbbi természetesen csak akkor jelenik meg, hogy ha kiválasztottuk a legördülő select mezőből a percenkénti futtatást.

Ezeket a beállításokat el is menthetjük.

Még van egy Sync Now opció is, ahol azonnal futtathatjuk a szinkronizációt. Ez megnézi, hogy történt-e változás a termékek vagy felhasználók adattábláiban.

Ez a funkció még csak elő van készítve, de nem működik, mert ehhez még kell a kapcsolat a desktop applikációval. Minden esetre kiírja, hogy a szinkronizáció folyamatban van.

Illetve a mai napon hozzáadtunk a bővítményhez Bootstrap CSS és JS fájlokat is, hiszen ez is kérés volt. Valamint saját stíluslapot is kapott a bővítmény, mert néhány dolgot szeretnék egyénileg formázni.

Ami még megoldásra vár, hogy a PHP-val létrehozott gombokra is alkalmazhatóak legyenek a Bootstrap osztályok, illetve az API kulcs generálásakor minden gomb felvegye az osztályok attribútumait. Egyelőre nem sikerült megoldani, mert nem mind jelennek meg a betöltéskor. Van amelyik csak később egy-egy esemény hatására, ezért furcsa módon nem veszi át a CSS attribútumokat. Tehát ezt is meg kell oldanunk.

Reméljük, hogy sikerül tartani a heti tervet, és akkor a jövő héten már a témával is tudunk foglalkozni. Az is sok idő lesz még biztosan.



## 2024-03-14

A mai napon ellenőriztük az API kulcsok fellelhetőségét, mivel ezek által tudjuk majd validálni a kapcsolatot a desktop és a WordPress között. Azonban rájöttünk, hogy update option helyett más módszert kell alkalmaznunk.

Emellett dolgoztunk a WooSync-sync.php fájlban is. Kiegészítettük és felkészítettük a külső adatbázis használatára.

## 2024-03-18

A mai napon bemutattuk a vizsgaremeket az óránkon, és szinte teljesen jól működött. Azonban a visszajelzés alapján kiderült, hogy még több funkcionalításra lenne szükségünk. Több menüpontnak kellene lekérdezéseket, kereséseket és törléseket is tartalmaznia.

Ez kihívást jelent számomra, mert eredetileg azt terveztük, hogy a desktop applikáció egyszerűen és kevés felhasználói beavatkozással működjön. Most, a leadás előtt 1 hónappal, váratlanul új funkciókat kell implementálnunk, amelyek nem szerepeltek eredeti terveinkben. Bár kissé fejetlenséget érzek a dologban, de ha ez szükséges a sikerhez, akkor meg kell tennünk.

A következő napokban ezeket a módosításokat kell kitalálnunk és kiviteleznünk.

## 2024-03-19

A mai napon tisztáztuk és feltöltöttük a Fejlesztői naplókat, illetve elkezdtük a tesznaplók írását is.

Emellett az oktatóm javaslatára létrehozunk egy új menüpontot is a ProductBridge desktop applikációban, amelyben újabb hozzáadási, törlési és listázási lehetőségeket fogunk implementálni. Ezúttal a felhasználók kezelésére.

Létrehozunk egy Users.cs osztályt, majd egy UsersViewPage-et is.

## 2024-03-20

A mai napon a WooSync bővítménnyel foglalkoztunk kicsit. Elsőként arra próbáltunk rájönni, hogy miért nem vesznek át bizonyos Bootstrap osztályok bizonyos gombok. A Console-ban hibaüzenetként azt találtuk, hogy nem elérhető a bootstrap.min.map.css és a bootstrap.min.js.map fájlok. Ezeket pótoltuk a repositoryban. A hiba eltűnt, de továbbra sem oldódott meg a kérdés. Az !important attribútumot használva, már valamelyest működött a dolog, bár nem túl elegáns megoldás, de továbbra sem hibátlan az összkép, ezért tovább

nyomozunk. ID-ra alkalmazva a CSS attribútumokat rendben működik a dolog, de ez szintén nem túl elegáns megoldás, ezért ezt szeretnénk a lehető legjobban abszolválni ha lehet.

Tehát most ez került a hibanaplóba.

## 2024-03-21

A mai napon megterveztük, hogy miként fog majd működni az autentikáció és a beléptetés. A tervünk az, hogy első belépéskor (First Attempt) meg kell adni a WooSync bővítmény által generált API kulcsot és URL-t, majd a mentés gombra kattintva az adatok ellenőrzése után szinkronizálás történne a két adatbázis között (csak a felhasználók esetében). Ha lefutott a szinkronizálás, akkor a felhasználó már be tud lépni a WordPress felhasználónevével és jelszával.

Lesz az applikációban/programban egy RESET gomb is, ami pedig minden adatot töröl a lokális adatbázisból.

## 2024-03-24

A mai napon sikeres deploymentet végeztünk el a fejlesztői oldalon. Így már elérhető a Github Pages-ként a dokumentációs oldalunk, amit tartalommal is fel töltöttünk. Létrehoztuk az aloldalakat, elhelyeztük a letöltési és egyéb linkeket, valamint feltöltöttük a saját fejlesztési és tesztenaplókat. Ezt a naplóbejegyzést már élesben a VS Code-ból írom most.

Holnap folytatjuk a bővítmény fejlesztését.

## 2024-03-25

Ma folytattuk a dokumentációs oldal fejlesztését. Elakadtunk a logó megjelenítésénél a főoldalon. Furcsa módon dev módban megjelenik a logó, viszont build után preview módban sem. Már próbáltunk többféle módon beilleszteni a logót, de valahogy mindig elromlik valami. Több fórumon is utánanéztünk, de még nem találtunk működő megoldást.

Holnap megpróbáljuk a README.md fájlt együtt elhelyezni a root könyvtárban a dokumentációs projekttel, és onnan URL alapján hivatkozni rá, mintha egy külső fájl lenne. Reméljük, sikerül.

A Desktop applikáció kapcsán elkezdtük az átszervezést. MySQL adatbázist fogunk használni a projekt mögött. Az automatikus adatbázis létrehozása már működik a táblákkal együtt. A cél, hogy minden átkerüljön erre az alapra, mert úgy gondoljuk, hogy nagyban megkönnyíti majd a szinkronizációt.

Mivel már az API kulcsnak is van táblája az új adatbázisban, holnap szeretnénk létrehozni az applikációhoz azt az oldalt és osztályt, amelyben megjelenítjük az eltárolt API kulcsot, URL-t, és adott esetben resetelni is tudjuk a kapcsolatot és az adatbázis tartalmát.

## 2024-03-30

Ma elkezdtek az átszervezést a WooSync bővítményen. Erre azért volt szükség, hogy a Tiszta kód szellemében minél kisebb részekre bonthassuk a bővítményt. Ami csak lehet, azt kiszerveztük külön osztályokba, és át is neveztük a fájlokat, hogy beszédesebbek legyenek. Például az index.php-t átneveztük woosync.php-ra. Persze ennek során némi kavardás és merge conflict keletkezett, de végül megoldottuk.

## 2024-03-31

A mai napon sikerült egy kicsit átalakítanunk a WooSync kezelőfelületét. Létrehoztunk egy menüsort az Admin menüben elhelyezkedő menüsoron kívül, mert úgy gondoltuk, hogy így egyszerűbben használható és dizájnosabb lesz az applikáció. Az alap menüpontok mellett beágyaztuk például a Desktop applikáció letöltési linkjét és a dokumentációs oldal linkjét is. A gombokat és további elemeket is szebbé varázsoltuk, és minden lehetőséget külön classokba szerveztünk a bővítményben az egyszerűbb kezelhetőség és fejleszthetőség érdekében.

## 2024-04-02

Ma végighaladtunk a Desktop applikáción és átszerveztük itt is néhány osztályt. Többek között a teljes applikáció átállt a MySQL-re, SQLite helyett. Úgy gondoltuk, hogy így egyszerűbb ellenőrizni a tárolt adatok helyességét, és könnyebb lesz majd a szinkronizációs folyamatot is leködölni, mivel így a phpMyAdminban láthatunk minden változást. Némi probléma akadt a null értékekkel a termékek kapcsán, de azt is orvosoltuk.

## 2024-04-03

A mai napon sikerült megoldanunk a logo elhelyezését a dokumentációs oldalon. Egyszerűen a repository-ból hivatkoztunk be a fájl elérési útvonalára (mintha egy külső fájl lenne), és így már rendben megjelenik a dokumentációs oldal főoldalán. Továbbá elkészítettük a projektmunka prezentációjának prototípusát is.

## 2024-04-16

A mai napon kijavítottuk a hibákat, amelyek miatt nem működött a termék és felhasználó módosítás. Ezek után néhány apró hibát is javítottunk még. Mivel a DataAccess.cs fájl már túl nagy volt és sok funkciót tartalmazott egyben, ezért azokat kiszerveztük egy Repository

mappában lévő ProductRepository.cs-be, illetve egy UserRepository.cs-be. Ehhez kapcsolódóan módosításokat kellett végrehajtani az érintett fájlokon is, hogy már ne a Data Access-ben próbálják végrehajtani a feladatokat.

Ezután a DataAccess.cs-t kiszerveztük a DataAccess mappába a gyökeri könyvtárban. Szerettük volna szétbontani a hosszabb kódokat az olvashatóság és karbantarthatóság érdekében.

## 2024-04-17 - 2024-04-02

Ebben az időszakban a kód refaktorálását, apróbb módosításokat, a dokumentáció - fejlesztői dokumentáció, felhasználói dokumentáció, prezentáció, naplók - elkészítését eszközöltük, hogy a határidőre minden felkerüljön a repository-ba.