



Szegedi SZC Tóth János Mórahalmi Szakképző Iskola és
Szilágyi Mihály Kollégium

VIZSGAREMEK

A **Padlás** applikáció fejlesztése

Készítette:

Ács Brigitta

Ádok Alex

Csányi Mónika

Szoftverfejlesztő és -tesztelő képzés

Mórahalom,

2024.

Tartalom

BEVEZETÉS	3
Főbb jellemzők:	3
1. Ruhák és cipők rendszerezése:	3
2. Képek hozzáadása:	3
3. Nem csak családoknak:	4
4. Bármely korosztály számára	4
Hogyan működik:	4
Termékek feltöltése:	4
Rendszerezés és kategorizálás	4
Keresés és használat:	4
FEJLESZTŐI DOKUMENTÁCIÓ	5
Fejlesztőkörnyezet	5
Hardver:	5
Szoftver:	7
Adatbázis	9
Felhasználók:	10
Tárolók:	10
Triggerek:	11
Fotók:	12
Az oldal felépítése és főbb algoritmusai	13
Regisztráció	13
Bejelentkezés	13
Tárolók listázása és új tároló létrehozása menüpont	14
Termékek / Új termék létrehozása menüpont	15
Termékek / Termékek keresése menüpont	16
Termék törlése:	17
Kép feltöltése:	17
Tároló választása:	17
Profil / Felhasználói adatok módosítása menüpont	17

Profil / Kijelentkezés menüpont	18
Tesztelés	18
Manuális tesztelés	18
FELHASZNÁLÓI DOKUMENTÁCIÓ	22
A Program céljának és lényegesebb funkcióinak összefoglalása	22
Szükséges hardvereszközök és szoftverek felsorolása	22
Telepítés és indítás lépéseinek ismertetése	23
A program részletes bemutatása	23
Helytelen használatból adódó hibajelzések magyarázata	23
Információkérés lehetőségeinek megadása	23
Felhasználók kezelése	24
Regisztráció	24
Bejelentkezés	24
Kijelentkezés.....	24
Profil menüpont.....	25
Termékek menüpont:	25
Tárolók listázása menüpont:	25
A „Tárolók listázása” menüpont a bejelentkezést követően válik elérhetővé a fejlécen lévő legördülő menük között elhelyezve. Kattintás után a Padlás alkalmazásban meglévő tárolók listája jelenik meg a tároló nevével, valamint bele/belőle (ki)helyezett termékek számával, illetve a tárolók hozzáadási és törlési lehetőségével.	25
FEJLESZTÉSI LEHETŐSÉGEK	26
ÖSSZEFOGLALÁS.....	27
IRODALOMJEGYZÉK	28

BEVEZETÉS

“Üdvözljük a Padlas nevű alkalmazásban, amelyet egyedülálló és családos személyek egyaránt használhatnak. Célunk az alkalmazással, hogy segítsünk rendszerezni és tárolni az átmenetileg nem használt tárgyakat, főként ruhákat és cipőket.

Szeretnénk segíteni alkalmazásunkkal, hogy könnyedén elérhetővé tegyünk azokat a holmikat, amelyeket időről időre elő kell venni, de nem szeretnénk, hogy elfoglalják az otthonunkban az értékes helyet. Használata egyszerű, csak néhány doboz és egy kis rendszerezési vágy szükséges környezetünk szervezettségéhez!”

Az alkalmazás ötlete igen egyszerű, sokak számára aktuális problémá(nk)ból eredt. Csapatunk 3 tagjának összesen 6 gyermeke van, így adott a feladat számunkra naponta, hogy a család minden tagjára jutó ruhákat, holmikat, cipőket rendben tartsuk. A napi használatban lévő kiegészítők mellett figyelniünk kell, hogy a gyerekek által aktuálisan nem használt, kinőtt, megunt holmiknak szintén helyet találjunk, mégpedig úgy, hogy az később gyorsan meg is legyen, ha mégis szükség lenne rá (akár kisebb gyermek számára, aki éppen belenő).

Sokak számára ismerős a helyzet, amikor a garázs/padlás/pince plafonig van dobozokkal, ám senki nem tudja, hogy melyik dobozban mi található, mivel a címkézés, rendszerezés sokszor elmarad. Ezt a problémát szeretnénk megelőzni és alkalmazásunk által átláthatóbbá, letisztultabbá és kezelhetőbbé tenni a folyamatot. A “Padlás” hosszútávon még számos irányba tovább fejleszthető, hiszen minden családnak egyedi igényei merülhetnek fel, de igyekeztünk a legáltalánosabb elvárásoknak megfelelően megtervezni a felületet.

Főbb jellemzők:

- Ruhák és cipők rendszerezése:** Az alkalmazás lehetővé teszi az aktuálisan nem használt ruhák és cipők méret, típus és más kategóriák szerint történő rendszerezését a felhasználó tárolók révén. Így évek múlva is könnyedén megtalálhatja rég nem látott „szekrénylakókat”
- Képek hozzáadása:** Csatoljon képeket a termékekhez, hogy még könnyebben azonosíthassa őket. Már az első pillantásra láthatja, hogy melyik ruha vagy cipő van a tárolóban.

3. **Nem csak családoknak:** Családok igényeinek is megfelel az alkalmazás, hogy segítsünk az olyan gyakori problémák megoldásában, mint a gyerekek közötti ruhák és cipők átörökítése, de remek segítség lehet bárki számára, aki gyakran frissíti gardróbját, ám néha rég elfeledett kincseket venne elő ismét. Egy szervezett gardrób, padlás, vagy garázs jó barátunk lehet fontos alkalmak előtt is.

4. **Bármely korosztály számára:** Az applikáció könnyen kezelhető, így bármely korosztály számára ideális, aki rendszert és szervezettséget szeretne teremteni környezetében.

Hogyan működik:

Termékek feltöltése: Egyszerűen töltsse fel a felesleges ruhákat és cipőket az alkalmazásba a megfelelő méret, típus és kép hozzáadásával.

Rendszerezés és kategorizálás: Rendezze és csoportosítsa a termékeket a legmegfelelőbb módon, hogy könnyen megtalálja őket később.

Keresés és használat: Amikor szüksége van valamire, egyszerűen keresse meg az alkalmazásban.

FEJLESZTŐI DOKUMENTÁCIÓ

Fejlesztőkörnyezet

Jelen fejezetben kerülnek bemutatásra azon hardverek és programozási eszközök, könyvtárak és beállítások csoportjai, amelyekkel a szoftverfejlesztési folyamat során a felhasznált programozási nyelveken létrehozott forráskódok futáskész állapotba kerültek.

Hardver: A fejlesztési folyamat alapvetően három számítógép segítségével zajlott, amelyek hardver konfigurációit az alábbiakban ismertetjük:

1. konfiguráció (Brigitta):

Számítógép: ACPI x64-alapú PC (Mobile)

Operációs rendszer: Windows 10 Home 64 bites

Google Chrome: Google Chrome 123.0.6312.123

DirectX: DirectX 12

Számítógép név: DESKTOP-8GACHRA

Felhasználó neve: DESKTOP-8GACHRA /acsbr

Dátum / idő: 2024.04.17. 20:56

Alaplap

Processzor típusa: Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz 2.71 GHz

Alaplap gyártója: ASUSTek COMPUTER Inc.

Alaplap termék: X405UA

Alaplap verziószáma: 1.0

BIOS típusa: AM Inc. X405UA.305, 2019.04.16.

Megjelenítés

Videokártya: Intel(R) HD Graphics 620

2. számítógép (Mónika):

konfiguráció:

Számítógép: ASUS VivoBook

Operációs rendszer: Windows 11 Pro 10.0.22631

Google Chrome: Google Chrome

DirectX: DirectX

Számítógép név: DESKTOP-APIHALK

Felhasználó neve: DESKTOP-APIHALK/szkcsb

Dátum / idő: 2024.04.22. 18:40

Alaplap

Processzor típusa: Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz, 1800Mhz

Alaplap gyártója: ASUSTeK COMPUTER Inc.

Alaplap termék: X542UN

Alaplap verziószáma: 1.0

BIOS típusa: UEFI 255.255

Megjelenítés

Videokártya: Intel(R) HD Graphics 620

3. számítógép (Alex)

konfiguráció:

Számítógép: DELL Precision T5610

Operációs rendszer: Windows 10 Pro 64 bit

Google Chrome: Google Chrome 123.0.6312.123

DirectX: DirectX 12

Számítógép név: DESKTOP-L2NUPVM

Felhasználó neve: DESKTOP-L2NUPVM /axxel.hu

Dátum / idő: 2024.04.30. 10:06

Alaplap

Processzor típusa: 2x Intel(R) Xeon(R) CPU E5-2667 0 @ 2.90GHz 2.90 GHz

Memória méret: 64.0 GB DDR3

Alaplap gyártója: Dell Inc.

Alaplap termék: 0WN7Y6

Alaplap verziószáma: A01

BIOS típusa: Dell Inc. A08, 2014.07.24.

Megjelenítés

Videokártya: NVIDIA Quadro P1000 - 4GB DDR5

Képernyő: Samsung S22E450

ASUS WH228

AOC 27B2G5

TCL55C715

Szoftver:

Az alkalmazás fejlesztéséhez választott szoftvereket a projekt kezdetén határoztuk meg. Mivel főként webalkalmazást készítettünk, az előző tapasztalatok és tanulmányok alapján döntöttünk a szükséges eszközökről. Az alkalmazott szoftverek listája a következő volt:

Az előző fejezetben említett hardvereken a Windows 10 operációs rendszer futott, ami biztosította a szükséges alapfunkciókat és kompatibilitást a fejlesztői munkához.

A kódoláshoz és a webalkalmazás készítéséhez a XAMPP nevű szoftvert használtuk, amely egy platformfüggetlen, nyílt forráskódú webservert tartalmazó szoftvercsomag. A XAMPP részét képezi az Apache webservert, a MariaDB (MySQL) adatbáziskezelő és elemei még a PHP értelmező és FileZilla FTP-kliens is.

A XAMPP 3.3.0 verzióját használtuk, ami biztosította a PHP kódok értelmezését, valamint a webservert és az adatbáziskezelő működését. Emellett összhangot biztosított a webservert, az adatbázisszert és a PHP szolgáltatásait, valamint a kommunikációs portok között.



Az alkalmazás adatbázis-kezeléséhez a PHPMyAdmin-t választottuk, ami egy webes adatbázis-kezelő alkalmazás MySQL adatbázisokhoz. Ennek segítségével könnyen és hatékonyan lehet kezelni az adatbázisokat böngészőn keresztül, így ideális megoldásnak bizonyult a projekthez.



A fejlesztés során a Visual Studio Code-ot választottuk, ami egy népszerű, ingyenes, nyílt forráskódú kódszerkesztő program. A szoftver támogatja a legelterjedtebb programozási nyelveket, és könnyen használható.

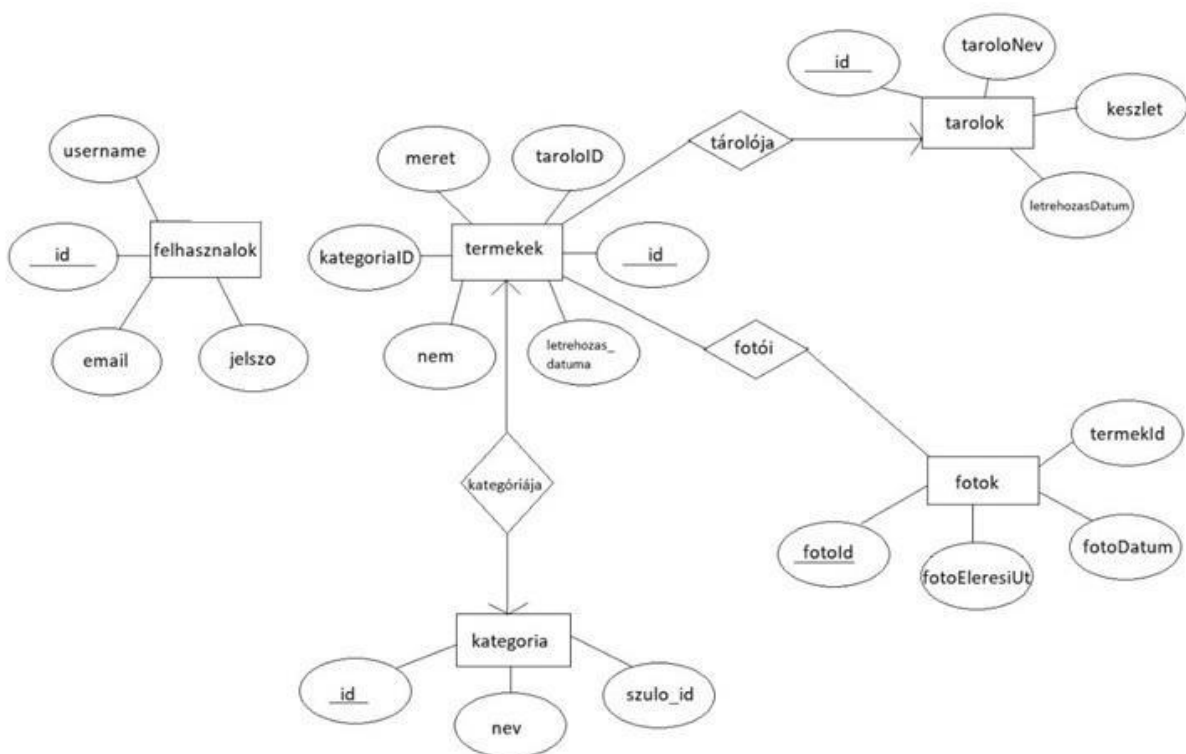


A webalkalmazás teszteléséhez a Google Chrome böngészőt használtuk, mivel ez a legnépszerűbb böngésző a felhasználók körében. A különböző funkciók és a CSS kódok megjelenítését is ebben a böngészőben teszteltük.



Adatbázis

Az adatbázis logikai modelljét egyed-kapcsolat jelölésrendszer segítségével vizualizáltuk, mely az 1. ábrán látható.



A munka során olyan adatbázist építettünk fel, mely alkalmas információt tárolni regisztrált felhasználókról, a rendszerben a regisztrált felhasználók által létrehozott tárolókról és az azokban elhelyezett termékekről, azok kategóriáiról és képeiről. Az alkalmazás

megvalósításához 5 adattáblára, 5 kapcsolatra és minden entitáshoz legalább három attribútumra volt szükség. A továbbiakban az imént felsorolt elemek kerülnek bemutatásra

Felhasználók:

Ez a tábla az alkalmazás felhasználóinak adatait tárolja. A felhasználók regisztrálás után kapnak egy egyedi azonosítót, meg kell adniuk egy felhasználónevet, email címet és jelszót. Ez utóbbi oszlop egy hash algoritmussal kódolt értéket tartalmaz.

Oszlop neve	Típus	Jellemzők
id	int(11)	Elsődleges kulcs, AUTO_INCREMENT
username	varchar(50)	unique
jelszo	varchar(200)	
email	varchar(100)	unique

Tárolók:

Ez a tábla tartalmazza az alkalmazásban a felhasználók által létrehozott tárolóknak az azonosítóját, a tároló nevét, az adott tárolóban lévő készletet, valamint nyilvántartja az adott tároló létrehozásának a dátumát is. A tárolók tábla rögzíti a raktárkészletet és annak elérhetőségét. Az adott táblához két trigger is kötődik, mely triggerok segítenek az adatintegritás fenntartásában és az adatbázis konzisztenciájának biztosításában.

Oszlop neve	Típus	Jellemzők
id	int(11)	Elsődleges kulcs, AUTO_INCREMENT
taroloNev	varchar(100)	
keszlet	int(11)	Alapértelmezett érték: 0
letrehozasDatum	datetime	

*Megjegyzés: A készlet oszlop tárolja a raktárban lévő termékek számát.

A termékek tábla lehetővé teszi a termékek rögzítését az alkalmazásban. Minden terméknek van egyedi azonosítója, létrehozás dátuma, kategóriája, mérete és tárolója. A külső kulcsok és a triggerok (termék rögzítés trigger, termék törlés trigger, törlés_termékhez_fotó trigger) segítenek az adatintegritás fenntartásában és az adatbázis konzisztenciájának biztosításában.

Oszlop neve	Típus	Jellemzők
id	int(11)	Elsődleges kulcs, AUTO_INCREMENT
letrehozás_datuma	datetime	
nem	tinyint(4)	
katégoriaID	int(11)	Külső kulcs a kategória táblához
meret	varchar(100)	
tárolóID	int(11)	Külső kulcs a tárolók táblához

Triggerok:

Termék_rogzites trigger: Ez a trigger az termékek táblához kapcsolódik és az új termék beszúrásakor „hívódik” meg. A trigger módosítja a tárolóban lévő termékek számát. Amikor egy új termék kerül beszúrásra a termékek táblába, a trigger megnöveli az adott termék tárolójában lévő termékek számát.

Termék törlés trigger:

Ez a trigger a termékek táblához kapcsolódik és a termék törlésekor „hívódik” meg. A trigger módosítja a tárolóban lévő termékek számát.

törles_termékhez_foto trigger:

Csakúgy, ahogyan az előző két trigger ez is a termékek táblához kapcsolódik és feladata, hogyha törlődik egy termék a termékek táblából, akkor a fotók táblából is törlődjenek a törölt termékhez kapcsolódó képek.

A kategória tábla hierarchikus struktúrában tárolja a termékkategóriákat. Minden kategóriának egyedi azonosítója van, a kategória neve, valamint az esetleges szülőkategória azonosítója. A szülőkategória azonosítója lehet null is, ami azt jelzi, hogy a kategória a legfelső szinten van a hierarchiában. Ez a struktúra lehetővé teszi a kategóriák könnyű csoportosítását és keresését a termékek között.

Oszlop neve	Típus	Jellemzők
id	int(11)	Elsődleges kulcs, AUTO_INCREMENT
nev	varchar(200)	
szulo_id	int(11)	

Megjegyzés: A szulo_id oszlop tárolja a szülőkategória azonosítóját, vagy 0-t, ha nincs szülőkategóriája.

Fotók:

A fotók tábla az egyes termékekhez feltölthető képek azonosítóját, elérési útját, a fotó feltöltésének dátumát, valamint annak a terméknek az azonosítóját tartalmazza, amelyhez a képet feltöltötték. Éppen ezért látható, hogy a táblában lévő termékId egy külső kulcs, egészen pontosan a termékek tábla id oszlopával egyeznek az értékei.

Oszlop neve	Típus	Jellemzők
fotoId	int(11)	Elsődleges kulcs, AUTO_INCREMENT
fotoEleresiUt	varchar(500)	
fotoDatum	datetime	
termekId	int(11)	Külső kulcs a termékek táblához

Az oldal felépítése és főbb algoritmusai

A program működését az alábbiakban bemutatásra kerülő, egy-egy részfeladat megoldását kivitelezését végző algoritmusok biztosítják.

Regisztráció

Az alkalmazás regisztrációs folyamata a registration.php fájlban kezdődik, ahol az első lépés a szükséges fájlok beimportálása és az adatbázis kapcsolat inicializálása. Ez magában foglalja a konfigurációs fájl betöltését, az adatbázis kapcsolat létrehozását és a felhasználó modell inicializálását.

Amikor a felhasználó kitölti és elküldi a regisztrációs űrlapot, az alkalmazás fogadja az adatokat. Ezeket az adatokat ellenőrizzük a validateRegistration.php segítségével. Ez a fájl felelős az adatok validálásáért és az esetleges hibák kezeléséért.

Ha a validáció sikeres, azaz az adatok megfelelnek a követelményeknek, a felhasználó létrehozása történik meg. Ehhez az új felhasználó adatait elmentjük az adatbázisba a User osztály newUserRegistration metódusának segítségével. Itt ellenőrizzük, hogy az adott felhasználónév és e-mail cím már foglalt-e az adatbázisban. Ha nem, akkor titkosítjuk a jelszót, és az összes adatot beillesztjük az adatbázisba.

Ha bármelyik lépés sikertelen volt (például hibás adatok, foglalt felhasználónév, vagy email cím), akkor egy hibaüzenet jelenik meg a felhasználónak, jelzésképpen hogy mi volt a probléma.

Végül, ha a regisztráció sikeres volt, a felhasználót visszairányítjuk a bejelentkezési oldalra, ahol már be tud jelentkezni az új fiókkal.

Bejelentkezés

A bejelentkezés felület működése két fő részből áll: az űrlap megjelenítéséből és a felhasználó bejelentkeztetéséből. Az űrlapot a loginForm.php fájl tartalmazza, melyben a felhasználó megadja a felhasználónevet és jelszavát. Ezeket az adatokat POST-tal adjuk át a user.php fájlban található getLoginUser nevű paraméteres függvénynek.

A user.php fájl felelős a felhasználókkal kapcsolatos SQL műveletek végrehajtásáért. A getLoginUser metódus az űrlapról kapott felhasználónév és jelszó alapján keres a felhasználók között az adatbázisban. Ellenőrzi, hogy minden adatot megadtak-e a felhasználók, és

figyelmeztet, ha hiányosak. Ha a megadott adatok hibásak, hibaüzenetet jelenít meg. Ha a megadott adatok helyesek, a felhasználót bejelentkezteti és átirányítja a padlasMainPage.php oldalra

Ez a folyamat biztosítja a felhasználók bejelentkezését az alkalmazásba, és lehetővé teszi számukra az adatbázisbeli adatok elérését és kezelését.

Tárolók listázása és új tároló létrehozása menüpont

Amikor a felhasználó a "Tárolók listázása és új tároló létrehozása" menüpontot választja, a rendszer először betölti a tárolók listáját az adatbázisból. Ezután ezeket a tárolókat egy táblázatban jeleníti meg a weboldalon.

Ha a felhasználó új tárolót szeretne létrehozni, ehhez egyszerűen csak rá kell kattintania az "Új tároló létrehozása" gombra. Ez a gomb egy createNewStorage() nevű JavaScript függvényt hív meg, ami egy AJAX kérést küld a szervernek és kéri a createStorage.php fájlt, amelyben egy létrejön az adatbázis kapcsolat valamint egy Storage egyed példányosítása történik.

A createStorage.php fájlban továbbá a Storage egyednek a create_new_storage() nevű metódusa kerül meghívásra, amely kezeli az új tároló létrehozását az adatbázisban. Egy véletlenszerű nevet generál az új tárolónak, majd ezt beszúrja az adatbázisba.

Ha a felhasználó termékeket szeretne kivenni egy tárolóból, csak annyit kell tennie, hogy rákattint az adott tárolóhoz tartozó "Termékek kivétele a tárolóból" ikonra. Ez a kattintás egy másik JavaScript függvényt (removeItemFromStorage(\$taroloid)) indít, ami ellenőrzi a tároló aktuális készletét. Ha a készlet nagyobb, mint 0, akkor egy confirm ablakban egy figyelmeztető üzenet jelenik meg, amit ha a felhasználó elfogad, akkor a program egy AJAX kérést küld a szervernek és kéri a removeItemFromStorage.php fájlnak a megnyitását és átadja a fájlnak a függvénynek az id paraméterét. A removeItemFromStorage.php fájlban történik meg az adatbázis kapcsolat létrehozása és a User osztály példányosítása, mely példányosítás után meghívjuk az osztály removeItemFromStorage(\$storageId) nevű metódusát. Abban az esetben, ha a készlet 0, akkor a rendszer jelez, hogy a tárolóból nem lehet terméket kivenni.

Ha a felhasználó egy tárolót szeretne törölni, csak annyit kell tennie, hogy rákattint az adott tárolóhoz tartozó "Tároló törlése" ikonra. A kattintás egy deleteStorage(\$taroloid) nevű

JavaScript függvényt hív meg, mely függvény először ellenőrzi, hogy vannak-e termékek a tárolóban. Ha van termék a tárolóban, akkor figyelmezteti a felhasználót, hogy először vegye

ki a termékeket a tárolóból és a tárolók törlése nem hajtodik végre. Amennyiben a tárolóban nincs termék, úgy a rendszer lehetővé teszi a tároló végleges törlését. A szerver oldali kommunikációt AJAX hívás valósítja meg.

Abban az esetben, hogyha a felhasználó sikeresen törölte a tárolót, úgy az AJAX frissíti a tárolók listáját, hogy az eltávolított tárolók ne jelenjenek meg többé a felhasználók számára. Ezzel a folyamattal a rendszer hatékonyan és biztonságosan kezeli a tárolók létrehozását és törlését, biztosítva a felhasználók számára a kényelmes használatot és az adatintegritást.

Termékek / Új termék létrehozása menüpont

A Termékek menüponton belül az Új termék létrehozása segítségével tudunk egy új terméket - esetünkben egy ruhadarabot, vagy egy lábbelit - felvinni az adatbázisba. Öt mezőt kell kitölteni - Nem, Kategória (ruha v cipő), Típus, Méret, Tároló. A Típus, és Méret mező csak akkor érhető el, ha a Nem, és a Kategória mező kiválasztásra került, mert így érhető el a megfelelő méret, és típus kiválasztás. Tehát a ruha kategóriához csak a ruha méretek, és típusok jelenjenek meg, a cipőhöz a cipő méretek, és típusok. Ezt a createNewProductForm.php-ben egy Javascript függvény végzi el valós időben. Tehát kategória váltáskor azonnal változik a másik két mező is. Miután az összes mezőt kitöltöttük a felvinni kívánt terméknek megfelelően a Termék létrehozása gombbal indítjuk el a tényleges létrehozási folyamatot, ellenkező esetben a gombra kattintva nem történik semmi. Amennyiben a tároló nem volt kiválasztva, egy figyelmeztetést kapunk, miszerint : Az előbb nem volt tároló kiválasztva! és újra próbálhatjuk a mezők kitöltését. Ha meggondoltuk magunkat, vagy csak véletlen navigáltunk erre az oldalra, a 'Létrehozás megszakítása' gombbal a Termékkeresőbe kerülünk. Sikeres termék létrehozás esetén a Products egyednek a createProduct() metódusát hívjuk meg, ahol is egy sql paranccsal beillesszük a terméket az adatbázisba, majd innen tovább ugrunk a createNewPhotos.php oldalra, ahol lehetőség van képeket feltölteni a termékhez. Amennyiben nem választottunk ki fájlt, úgy Nem lehet feltölteni a fájlt! üzenetet kapunk. A 'Még egy kép feltöltése' gombbal folytathatjuk a képek feltöltését, korlátlan mennyiségben. A 'Kép feltöltése gomb' esetén meghívódik a upload() függvény, amikor is a szerveren létrejön egy upload mappa - ha még nincs -, benne az aktuális év, hónap, nap mappa, benne a fájl-al, sikeres fájl feltöltés esetén a Products egyednek az uploadPhoto(\$filePath, \$id) metódusa megkapja a fájl szerveren található elérési útvonalát, és a termék azonosítóját, és ezek az adatok kerülnek bele a padlás adatbázis

fotok táblájába. Sikertelen fájl feltöltés esetén - pl jogosultsági problémák - ezek mind nem kerülnek végrehajtásra, Nem lehet feltölteni a fájlt üzenetet kapunk.

Termékek / Termékek keresése menüpont

Itt van lehetőség a szűrési feltételek alapján bármilyen terméket megkeresni. Kereshetünk termékekre, és kereshetünk tárolóra is.

Termék keresése:

A négy keresési mezőből bármilyen variációban szűrhetünk, de itt is alapfeltétel, hogy a kategória, és a típus, és a méret összefüggésben van egymással, tehát pl ruha kiválasztásánál itt sem tudunk cipő jellemzőkre szűrni, és fordítva. Ezt egy frissit() nevű Javascript függvény végzi valós időben. A szűrés után a kereső mező alatt találunk egy táblázatot, ID, NEM, KATEGÓRIA, MÉRET, FOTÓ, TÁROLÓ oszlopokkal, amiben az ID, és a TÁROLÓ oszlopokban található értékek egyben linkek is, amik a termék adatlapjára, vagy a tároló listázása menübe visz. A FOTÓ oszlopban amennyiben van fotó az egyben link is, ha nincs akkor nem ugró pont. A miniatürizált fotó a termék első elérhető fotóját tartalmazza (first_Photo()), rákattintva betöltődik a displayProductPhotos.php oldal, középen az aktuális fotó képével, tőle balra, jobbra miniatürizáltan az előző, vagy következő kép látható, lapozható gombokként tálalva. Amennyiben a nincs fotó képet látjuk ott, az természetesen nem linkelhető. Itt van lehetőség az aktuális kép törlésére is, ami csak egy felugró figyelmeztető ablak jóváhagyása után hajtódik végre.

Az ID értékre kattintva a modifyProduct.php töltődik be, itt van lehetőségünk a terméket módosítani, törölni, további képet feltölteni hozzá. Erre az oldalra a tároló listázó oldalról is érkezhetünk, ezért a Mégse gombra kattintva visszakerülünk az előző oldalra a bejelölt keresési mezőkkel együtt. Ezt a \$prev_site nevű segédváltozóval értjük el, ez tartalmazza az előző oldal típusát (product vagy storage). A termék módosításánál is fontos, hogy a már korábban említett kategória-típus-méret összefüggéseket ne tudjuk átlépni. Ha terméket módosítunk, meghívódik a modify_Product() eljárás, paraméterként átadjuk a _POST metódussal megkapott értékeket. Az eljárás először lekérdezi az adatbázisból elérhető az adott termékhez tárolt jellemzőket, majd különböző, összetett összehasonlítási algoritmusok alapján eldönti, hogy a módosítás végrehajtható-e vagy sem. Amennyiben végrehajtható, az adatbázisba felvezeti a változtatásokat.

Termék törlése:

Egy felugró figyelmeztető ablak jóváhagyása után törlődik a termék az adatbázisból, minden jellemzőjével, az esetleges fotóival együtt (`delete_Product()`). Mivel jelen alkalmazás egy gyermekruha raktárkészlet kezelő program, egy termék törlését úgy vesszük, mintha az a napi használatba lesz áthelyezve, vagy el lesz ajándékozva/értékesítve, ezért nem láttuk értelmét, hogy a törölt termék az adatbázisban történő további jelölését alkalmazzuk.

Kép feltöltése:

Itt van lehetőség az első, vagy további képek feltöltésére a termékhez kapcsolódóan. Hasonlóan, mint a termék létrehozása menüpontnál.

Tároló választása:

A tárolólistázó egy olyan része a szoftvernek, ahol a kiválasztott tárolóban található termékeket tudjuk kilistázni. A táblázat oszlopai: ID, NEM, KATEGÓRIA, MÉRET, FOTÓ. Az ID oszlopban levő érték egyben link a termék adatlapjához, illetve a miniatűrízált fotó is, ami a termék fotóihoz navigál.

Profil / Felhasználói adatok módosítása menüpont

A felhasználói adatok módosításának folyamata hasonlít a felhasználói adatok regisztrálásához szükséges folyamathoz. A `profil.php` oldalon oldalon két különböző űrlap található: az egyik a felhasználónév módosítását teszi lehetővé, míg a másik a jelszó módosítást hivatott elősegíteni.

A felhasználónév módosításához a felhasználónak meg kell adnia az új felhasználónevet, majd az „Új felhasználónév módosítása” gombra kattintva meghívódik egy `submitModifyUserName()` kétparaméteres függvény, mely egy AJAX kérést indít a `modifyUsername.php` felé és átadja a fájlnek a `submitModifyUserName()` két paraméterét. A `modifyUsername.php` fájlban az átvett paramétereket elmentjük egy-egy változóba, majd példányosítunk egy `user`-t a `User` osztályból és meghívjuk a `user` példány `modifyUserName` metódusát, ami elvégzi a felhasználónév módosítását.

A jelszót módosításához az új jelszót és annak megerősítését kell megadnia az erre a célra kialakított mezőkben.

Miután az űrlap kitöltése megtörtént, a felhasználó a „Módosítás mentése” gombra kattintva továbbítja az adatokat a szerver felé. A szerver fogadja az adatokat, majd ellenőrzi azokat. Üres

mezők vagy a jelszavak nem egyezése esetén hibaüzenet jelenik meg, figyelmeztetve a felhasználót a hiányzó vagy hibás adatokra.

A jelszó esetén a szerver ellenőrzi a régi jelszót az adatbázisban tárolttal, majd titkosítja az új jelszót és frissíti azt az adatbázisban.

Ha minden művelet sikeres volt, a szerver visszairányítja a felhasználót a főoldalra, ahol már az újonnan frissített adatok lesznek érvényesek. Ha bármelyik lépésben hiba történik, a szerver hibaüzenetet generál, és azt jeleníti meg a felhasználónak, felhívva figyelmét a problémára.

Profil / Kijelentkezés menüpont

A profil.php tartalmazza webalkalmazás profil oldalának vezérlőjét. A kódban több funkció található, például a felhasználói adatok módosítása, jelszóváltoztatás és a felhasználó törlése. Emellett egy kijelentkezési lehetőséget is biztosít a felhasználó számára.

Az alkalmazás kezdetén létrehoz egy munkamenetet (`session_start()`), majd ellenőrzi, hogy a felhasználó ki akar-e jelentkezni (`$_GET["submit"]`). Ha igen, akkor törli a munkamenetet és átirányítja a felhasználót az index.php fájlra. Ezt JavaScript segítségével teszi meg.

Tesztelés

Manuális tesztelés


Manuális tesztelést végeztünk a tervezett funkciók helyes működésének ellenőrzésére. Az egyes tesztesetek eredményét az alábbi táblázatban ismertetjük.






Regisztráció		
Tesztesetek	Elvárt kimenet	Sikeres
Regisztrálj! Klikkelése. 	Regisztrációs felület megjelenése	Igen

Regisztrálok gomb klikkelése üres input adatokkal	Hibaüzenet	Igen Regisztrációs hiba! Minden adatot kötelező megadni! Kérem, adja meg az összes adatot! Bezár
Adatbázisban már mentett email cím használata	Másik email cím választását kérő hibaüzenet	Igen Regisztrációs hiba! A megadott email cím már szerepel az adatbázisban! Kérlek, adj meg új email címet! Bezár
Helytelen formátumú e-mail cím megadása	Hibaüzenet	Igen
Helytelen megerősítő jelszó megadása	Hibaüzenet	Igen Regisztrációs hiba! A két jelszó nem egyezik! Kérem, adja meg újra a jelszavát! Bezár
Sikeres regisztráció	Átírányítás a bejelentkezé s felületre.	Igen

Bejelentkezés		
Tesztesetek	Elvárt kimenet	Sikeres
	Bejelentkezési felület megjelenése	Igen

Adatbázisban nem szereplő felhasználónév megadása	Hibaüzenet	Igen Bejelentkezési hiba! Hibás felhasználói azonosítót! Kérem, adja meg a helyes azonosítót! Bezár
Helyes felhasználónév-jelszó páros megadása	Üdvözlő szöveg megjelenése a padlasMainPage.php oldalon	
Nem a felhasználónévhez tartozó jelszó megadása	Hibaüzenet	Bejelentkezési hiba! Hibás jelszó! Kérem, adja meg újra a jelszavát! Bezár

Tárolók menüpont		
Tesztesetek	Elvárt kimenet	Sikeres-e?
	Tárolók felület megjelenése	Igen
Új tároló létrehozása gombra klikkelés	Új tároló beszúrása az adatbázisba és az új tároló megjelenése	Igen
Meglévő tároló törlése abban az esetben, ha tárolóban van termék	Hibaüzenet	Igen A(z) localhost közlendője A tárolóban termékek vannak, így azt nem lehet törölni! 

Meglévő tároló törlése, ha a tárolóban 0 db termék van	Felugró ablakban megerősítés kérés, a megerősítés után a tároló törlődik az adatbázisból.	Igen A(z) localhost közlendője Biztos törlöd a 31 tárolót?  
Termékek kivétele a tárolóból, ha a tárolóban van termék.	Felugró ablakban megerősítés kérés, melynek elfogadása után a tárolóban lévő termékek törlődnek az adatbázisból, a tároló készlete 0 lesz.	Igen A(z) localhost közlendője Ha kiveszed a termékeket a tárolóból, akkor a termékek törlődnek az alkalmazásból! Biztos, hogy folytatni akarsz a termékek kivételét?  
Termékek kivétele a tárolóból, ha a tárolóban 0 db termék van	Hibaüzenet	Igen A(z) localhost közlendője A tároló üres, onnan nem lehet terméket kivenni. 

FELHASZNÁLÓI DOKUMENTÁCIÓ

A Program céljának és lényegesebb funkcióinak összefoglalása

A Padlás egy olyan alkalmazás, amelyet kifejezetten saját elképzelése szerint rendszerezni és tárolni vágyó felhasználók számára terveztünk. Az átmenetileg nem használt ruhákat és cipőket sok esetben problémás éveken keresztül úgy tárolni, hogy később megtaláljuk egy pincében/garázsban/padláson éppen, amikor keressük.

Célunk, hogy könnyen elérhetővé tegyük azokat a nem használt holmikat, amelyeket időről időre elő kell venni, akár egy új családtag érkezése kapcsán, de addig is nem szeretnénk, hogy elfoglalják otthonunkban az értékes helyet.

Szükséges hardvereszközök és szoftverek felsorolása

A Padlás alkalmazás használatához szüksége lesz egy internetkapcsolattal rendelkező eszközre, mint például okostelefon, tablet vagy laptop.

Az alkalmazás elindításához először telepíteni kell a XAMPP-ot, majd a xampp htdocs mappájába el kell helyezni az alkalmazást tároló mappát.

Ha telepítettük a XAMPP-ot, akkor indítsuk el az Apache webszervert és a MySQL adatbázist. A MySQL Admin gombjára kattintva a webböngészőnk megnyitja a PHP Myadmin-t. Itt hozzunk létre egy új adatbázist „padlas” néven, majd ebbe az adatbázisba importáljuk a mellékelt SQL fájlt.

Ha elvégeztük a szükséges műveleteket, akkor a böngésző URL mezőjébe írjuk be az alábbiakat:

http://localhost/Padlas_alkalmazas/Padlas/index.php

Ekkor megjelenik a Padlás alkalmazás bejelentkezés oldala, amelyen található egy „Regisztrálj” gomb. Erre kattintva a rendszer a regisztrációs oldalra navigál. A sikeres regisztrációt követően a rendszer visszairányít a bejelentkezés oldalra, ahol a felhasználónév és

jelszó helyes megadása után tudunk belépni a rendszerbe. A belépést követően az alkalmazáson belül első lépésként új tárolók létrehozását javasolt elvégezni.

Telepítés és indítás lépéseinek ismertetése

Az alkalmazást egyszerűen telepítheti eszközére. Miután telepítette az alkalmazást, egyszerűen kattintson a Padlás ikonra az eszközön, hogy elindítsa a szoftvert. Kezdőoldalunkon keresztül könnyedén és gyorsan regisztrálhat, majd személyre szabhatja tárolóit és feltöltheti a rendezni kívánt holmikat.

A program részletes bemutatása

Az alkalmazás használata rendkívül egyszerű. Feltöltheti a felesleges ruhákat, kiegészítőket és cipőket, majd rendezheti és csoportosíthatja azokat különböző kategóriák szerint. A képek hozzáadása segít az azonosításban, így könnyen megtalálhatja a szükséges holmikat.

Helytelen használatból adódó hibajelzések magyarázata

Amennyiben helytelen adatokat ad meg például űrlapok kitöltésekor, vagy hibás regisztrációs, illetve bejelentkezési adatokat ír be, az alkalmazás hibaüzenetekkel jelezheti a problémát. A dokumentáció részletesen magyarázza ezeket a hibajelzéseket és javaslatokat ad azok kezelésére.

Információkérés lehetőségeinek megadása

Az alkalmazás fejlesztőihez történő információkérés céljából írjon a brigittaacs7@gmail.com email címre.

A részletes dokumentáció segítséget nyújt a problémák megoldásában és az alkalmazás hatékony használatában.

Felhasználók kezelése

Az alkalmazás különböző funkciókat tesz elérhetővé a felhasználók számára.

A kezdőoldalon érhető el a “Bejelentkezés” és „Regisztráció” menüpont (az oldalon középen elhelyezve). A felhasználónév és a jelszó megadására szolgáló input mezőket itt találja a felhasználó. Elsődleges célja, hogy a regisztrált felhasználók részére elérhető szolgáltatásokat biztosítsa.

A felhasználónév mezőhöz kapcsolódik egy Ajax hívás, amellyel ellenőrizzük, hogy a megadott felhasználónév szerepel-e az adatbázisban. Amennyiben sikeres a regisztráció, majd a bejelentkezés, egy erre kijelölt rész üdvözljük az oldalon az adott felhasználónév kiíratásával és egy üdvözlő üzenet megjelenítésével a felhasználót. Az üzenet az oldal közepén helyezkedik el, a felső menüsávban pedig a további lehetőségek.

Amennyiben regisztráció nélkül próbál a felhasználó bejelentkezni, automatikus üzenet navigálja tovább az érdeklődőt.

Regisztráció

A “Regisztráció” menüpont célja, hogy a felhasználónak legyen lehetősége regisztrálni, majd ezt felhasználva a beléptetés algoritmusával be tudjon lépni és különböző funkciókat elérni az oldalon. A felhasználónév meglétét ellenőrizzük az adatbázisban. Amennyiben már létező felhasználói email címmel kísérlük meg a regisztrációt, értesítő üzenetet küldünk, amely jelzi, hogy a megadott email cím már foglalt.

Bejelentkezés

A sikeres regisztrációt követően a rendszer automatikusan a “Bejelentkezés” oldalra navigál. Ezen az oldalon ismét szükséges a felhasználónév és jelszó megadása, majd a Bejelentkezés gombra kattintást követően a rendszer bejelentkezteti a felhasználót az alkalmazásba.

Kijelentkezés

A bejelentkezett felhasználók kijelentkezése a fejlécen lévő legördülő menüből érhető el a “Kijelentkezés” menüpont által, amelynek feladata. Sikeres kijelentkezés esetében a

felhasználót a bejelentkezés oldalra irányítjuk és ismételten elérhetővé válnak részére a bejelentkezéshez, regisztrációhoz kapcsolódó menüpontok.

Profil menüpont

A profil menüpont a bejelentkezést követően válik elérhetővé a fejlécen lévő legördülő menü megnyitását követően. Kattintás után a felhasználói adatok módosítására, illetve „Kijelentkezés”-re van lehetőség.

A „Felhasználói adatok módosítása menüpontra kattintva” A felhasználói email cím kivételével minden mező szerkeszthető, így a betöltött adatokban történő módosításokkal a felhasználó a saját adatait változtatni tudja. Az új jelszó és a jelszó megerősítő mezők egyezőségének ellenőrzéséről egy Ajax hívás segítségével dinamikus visszajelzést kap a felhasználó. A mentéssel az adatbázisban lévő adatok felülírására kerül sor.

Termékek menüpont:

A „Termékek” menüpont a bejelentkezést követően válik elérhetővé a fejlécen lévő legördülő menü megnyitását követően. Kattintás után a „Termékek keresése”, illetve az „Új termék létrehozása” menüpontok válnak elérhetővé a felhasználó számára.

A „Termékek keresése” menüpontot választva új felhasználói felület válik láthatóvá: Nem, kategória, típus, méret, tároló opciók válnak elérhetővé. Az egyes részek lenyíló menüvel kínálnak választási lehetőséget. A „nem” kategóriában választható: fiú és lány, a „kategória” fülnél: ruha, illetve cipő, a „típus” legördülő fülnél: felső, alsó, egyberuha, kiegészítő, fehérnemű, a „méret”-nél gyermekkori méretezéstől felnőttig találunk lehetőségeket, a „tároló” fülnél pedig lehetőségünk van tárolót választani.

Az „Új termék létrehozása” menüpontot választva a fentiekhez hasonló megjelenést láthatunk, azonban a keresés gomb helyett a termékek létrehozása, valamint a létrehozás megszakítására alkalmas gombokat használhatunk.

Tárolók listázása menüpont:

A „Tárolók listázása” menüpont a bejelentkezést követően válik elérhetővé a fejlécen lévő legördülő menük között elhelyezve. Kattintás után a Padlás alkalmazásban meglévő tárolók

listája jelenik meg a tároló nevével, valamint bele/belőle (ki)helyezett termékek számával, illetve a tárolók hozzáadási és törlési lehetőségével.

FEJLESZTÉSI LEHETŐSÉGEK

Ezen szoftverben - véleményünk szerint - hatalmas fejlesztési potenciál rejtőzik. Jelen állás szerint alkalmas egy bármekkora tagú család használaton kívüli ruhatermékeinek katalogizálására, és akár termék-egyed szintű összetett keresésre. Végző soron elmondhatjuk, hogy ez az általunk elkészített megoldás, egy célirányos raktár-kezelő szoftver. Egy célszoftver.

Az alkalmazás jelenlegi állapota azt feltételezi, hogy a család, aki ezt használja, csak egy tároló helyiséggel rendelkezik. Mint például a padlás, vagy a garázs, stb. Az egyik fejlődési irány lehet, hogy úgy bővítsük a programot, hogy több ilyen helyiséget feltételezünk, és a helyiségek között mozgathatóak például a tárolók. Ha már több helyiséget kezel a program, lehetne egy olyan bővítési lehetőség, amikor a terméket a meglévőnél nagyobb számú, és pluszban más jellegű minőségi jellemzővel látunk el. Például ilyen lehetne a beérkezés időpontja, a származási hely, beszerzési ár, és már közelítünk is egy használtruha kereskedés raktárbázisát kezelő alkalmazáshoz. Jöhet hozzá egyedi vonalkódok nyomtatása, beolvasása, segítő a készlet ellenőrzését.

A másik fejlődési irány úgy módosítani a szoftvert, hogy a felhasználó alakítsa ki a termék, a tároló minőségi, mennyiségi jellemzőinek a kategorizálását. Ezáltal bármilyen termék csoportra lehet használni a programot. Ez egy kezdő szoftverfejlesztőnek igazi kihívás lehetne, mert olyan megoldásokat kellene lefejleszteni benne, mint például, hogy egy termék jellemzőinek a felhasználó által meghatározott módon kellene az adatbázis struktúrát regenerálni, nem beszélve az esetleges külső kulcsok figyelembe vételét is. Nyilván korlátok nélkül ezt leprogramozni túl nagy falat, ezért lehetne, hogy meghatározott mennyiségű minőségi jellemzőt lehet csak használni, és az egyes jellemzők szűrési értékeit a felhasználó határozhatná meg. Ez azt jelentené, hogy a Padlás nevű alkalmazásban a méret, és a kategória jellemzőket a felhasználó határozhatja meg, és így már nemcsak gyermek ruhákat lehetne katalogizálni. Például egy judo sportklub a megfelelő menü ponttal felvehetne a ruha típusok közé judo-ruha típust, egy labdarúgó szakosztály pedig ruha kategóriába mez, sportszár, cipő kategóriába pedig teremcipő, edzőcipő, stoplis, azon belül éles, gumis, stb típusokat.

Következő fejlődési irány lehetne, amikor egy konkrét felhasználó alapján úgy módosítjuk a szoftvert, hogy a megrendelő elképzeléseit valósítjuk meg. Mivel ez egy speciális raktárkezelő

program, a szoftver írása közben is alakultak, módosultak a beépíteni kívánt funkciók. A program tervezésekor meghatároztuk azokat a funkciókat, amik a működéshez feltétlen szükségesek, és voltak olyanok is, amik úgynevezett extra funkciók voltak. A tervezéskor, és fejlesztéskor törekedtünk a moduláris felépítéshez, ami a lehetséges bővítés egyik fundamentuma. Ezt mutatja a nagyszámú fájl (~70 db), amiből a szoftver végül is felépül.

ÖSSZEFOGLALÁS

Összefoglalva, a Padlás nevű vizsgaremek alkalmas egy család ruhatárának pontos nyilvántartására. A menüpontokon keresztül könnyen navigálható, esztétikus felületet fejlesztettünk ki, mely több elektronikai eszközön is elérhető és egyszerűen kezelhető. A felhasználóbarát felület egyértelmű üzenetekkel és visszaigazolásokkal segíti a felhasználókat. Az adatok tárolására szolgáló adatbázis szerkezete megfelelőnek bizonyult a feladat elvégzéséhez, és további fejlesztési lehetőségeket is kínál.

IRODALOMJEGYZÉK

[1] <https://erdplus.com/>

[2] <https://cypress.org/>

[3] <https://www.w3schools.com/php/default.asp>

[5] Tóth Attila: Webes programozás jegyzet