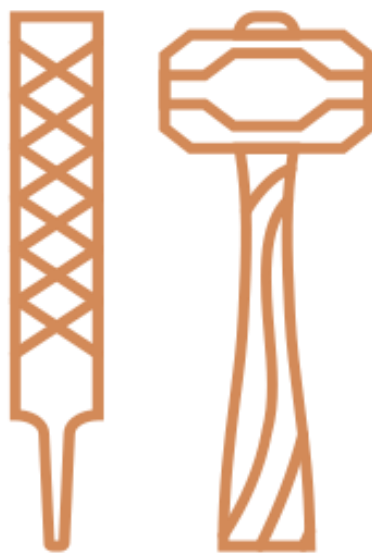


Kreatív hobbi- és barkács webáruház

Adatmodell dokumentáció



Készítette:

Kovács Edit

Tóthpál István



Tartalomjegyzék

1	Felhasználók.....	3
1.1	Felhasználó tábla	3
1.2	Felhasználó adatok tábla	4
2	Termékek.....	5
2.1	Termékek tábla	5
2.2	Összekötő táblák a termékekhez	5
2.3	Kedvencek tábla	5
3	Termék megrendelése	6
3.1	Rendelések tábla.....	6
3.2	Rendelt termékek.....	6
4	Blog	7
4.1	Posztok tábla.....	7
4.2	Kommentek tábla	7
4.3	Kiegészítő táblák a bloghoz	8
5	Kiegészítő táblák.....	9
5.1	Címkék tábla.....	9
5.2	Kategóriák tábla	9
5.3	Színek tábla	9
5.4	Képek tábla.....	9
5.5	Városok tábla.....	9



1 Felhasználók

1.1 Felhasználó tábla

A weboldalunk központi eleme a **felhasználók** tábla, amelynek elsődleges célja, hogy minden látogatónak zökkenőmentes, biztonságos és személyre szabott élményt nyújtson. Ez a tábla a rendszer gerincét képezi: ennek köszönhetően tudjuk az egyes látogatókat beazonosítani, kezelni és nyomon követni a különféle funkciók használata során. A tábla felépítése és működése teszi lehetővé, hogy a vendégek egyszerű regisztráció után teljes jogú tagjaivá váljanak a közösségnek, hozzászóljanak a blogbejegyzésekhez, értékeléseket írjanak, személyes ajánlásokat kapjanak, valamint gördülékenyen intézzék vásárlásaikat.

felhasznalok	
id	int
felhasz_nev	varchar(100)
email	varchar(100)
jelszo	varchar(255)
profilkep_id	int
statusz	enum(aktiv,inaktiv)
letrehozás_Datuma	timestamp
utolso_Belepes	timestamp

Az itt eltárolt adatok többségét a felhasználó adja meg a regisztrációs űrlap által. A **felhasz_nev** a felhasználó általa választott beceneve, ami csak egyedi lehet, így elkerülve a félreértéseket.

Az **id** egy automatikusan hozzárendelt szám a felhasználóhoz, amit ők nem látnak, viszont ennek a segítségével tudunk felhasználónevet kapcsolni a posztokhoz és kommentekhez.

Az **email** a kapcsolattartás miatt fontos, mivel innen kaphat híreket az akciókról, a bejegyzéseire kapott kommentekről, illetve a vásárlás után érkező értesítéseket is azon keresztül kapja a megrendelő.

A **jelszo** leginkább a biztonságot szolgálja, amivel a felhasználó be tud lépni, és több funkcióhoz hozzá tud férni mint egy vendégként látogató.

A **profilKep_id** egy kevésbé lényeges, ám olyan funkció, ami könnyebbé tudja tenni hogy a felhasználók felismerjék egymást. Ezt a képet a többi képpel egy táblában tároljuk, ugyanis ha a felhasználó nem választ, akkor mi egy automatikus képet utalunk ki neki.

A **statusz** mezővel a fiók állapotát tudjuk változtatni, ami automatikusan aktív, ám ha túl sok ideig nincs aktivitása a fióknak inaktívvá válik, majd egy idő után törlődik a rendszerből.

A **letrehozás_Datuma** az **utolso_Belepes**-el együtt egy időpontot tárol el. A profil létrehozásának ideje statisztikai és kedvezmény ajánlasi okokból lehet fontos, az utolsó belépés ideje pedig a fiók állapotának meghatározásában segít.



1.2 Felhasználó adatok tábla

A rendszerünk egy másik, különálló táblában kezeli azokat az adatokat, amelyek nem feltétlenül szükségesek a regisztrációhoz, de a felhasználó kényelme és a vásárlási folyamat megkönnyítése érdekében elmenthetők. Ez a funkcionális szétválasztás növeli a rendszer hatékonyságát és az adatbiztonságot, mivel az érzékenyebb személyes adatok csak akkor kerülnek előtérbe, amikor arra valóban szükség van.

A két tábla közötti összeköttetést természetesen a **felhasznalo_id** szolgáltatja, ahol az azonosító számhoz kötjük az adatokat, így felhasználó név módosítás esetén is könnyedén hozzá tudjuk kapcsolni a felhasználóhoz az adatait.

felhasznalokAdatai	
felhasznalo_id 🔗 🔗	int
vezeteknev	varchar(100)
keresztnev	varchar(100)
varos 🔗	int
utca	varchar(255)
hazszam	int
emelet_ajto	varchar(10)
telefonszam	varchar(20)
kartyaszam	int

Ilyen kiegészítő adatok például a **vezeteknev** és **keresztnev**, amelyek alapvetően a számlázási és szállítási folyamatokhoz nélkülözhetetlenek. A címadatok (**varos**, **utca**, **hazszam**, **emelet_ajto**) segítségével a rendszer pontosan képes meghatározni a kiszállítás helyét. Itt a könnyebb átláthatóság érdekében a városokat egy külön táblában tároljuk, ezzel is elkerülve a hibás vagy inkonzisztens adatokat, illetve a felhasználó számára is egyszerűbb adatbeviteli lehetőséget nyújt ez a fajta megoldás.

A **telefonszam** szintén fontos szerepet játszik, hiszen alternatív azonosítási eszközként is szolgálhat biztonsági ellenőrzések során.

A **kartyaszam** mező pedig opcionális, de jelentősen javítja a felhasználói élményt: ha a vásárló úgy dönt, hogy elmenti a bankkártya adatait, a rendszer a következő vásárlás során automatikusan kitölti a szükséges mezőket, így a fizetés sokkal gyorsabbá és kényelmesebbé válik.



2 Termékek

2.1 Termékek tábla

A **termékek** táblába kerül felvételre minden árucikk, amit a weboldalon meg lehet venni. Itt is egy számmal, az **id**-val azonosítjuk elsősorban a terméket, és a továbbiakban ennek a segítségével kötjük a többi táblához.

A **nev** és a **leiras** a termék egyik legfontosabb jellemzői, ide vesszük fel a pontos nevét, a felhasználó pedig a leírásból tudja eldönteni hogy ezt a terméket keresi e.

Az **ar** a termék jelenlegi árát mutatja, a **darab** pedig hogy mennyi elérhető még a termékből. A **kategoria_id** egy másik táblából hivatkozva mutatja, hogy melyik kategóriába sorolható a termék, a **fo_kep_id** pedig arra a képre mutat, ami az oldalon elsőnek megjelenik a képről.

termékek	
id 🔗 🔗	int
nev	varchar(255)
leiras	text
ar	decimal(10,2)
darab	int
kategoria_id	int
fo_kep_id 🔗	int

2.2 Összekötő táblák a termékekhez

A **termekKepek**, **termekCimkek** és **termekSzine** táblák egy másik táblából adnak tulajdonságokat az árucikkeinknek. A redundancia elkerülése miatt jobbnak láttuk külön táblába elhelyezni a többször előforduló adatokat, így csak egy helyen kell módosítanunk őket.

2.3 Kedvencek tábla

A tábla segítségével a felhasználók el tudják menteni, kedvencekbe teheti a terméket, így csak a profiljából máskor is egyszerűen el tudja érni őket.

termekKepek 🔗	
termek_id 🔗	int
kep_id 🔗	int
rendezés	int

termekCimkek 🔗	
termek_id 🔗	int
cimke_id 🔗	int

termekSzine 🔗	
termek_id 🔗	int
szin_id 🔗	int

kedvencek 🔗	
felhasznalo_id 🔗	int
termek_id 🔗	int



3 Termék megrendelése

3.1 Rendelések tábla

Itt az alapvető adatokat tároljuk a rendelésről, mint a felhasználó, a rendelés állapota, a végső összeg, illetve a megrendelés időpontja.

A **felhasznalo_id** által tudjuk lekérni a regisztrált vevőnek az adatait, illetve ha már korábban vásárolt, és elmentette az adatait, akkor azt is automatikusan ki tudjuk neki tölteni, így meggyorsítva a rendelési folyamatot.

rendelesek	
id 	int
felhasznalo_id	int
statusz	enum
osszeg	decimal(10,2)
rendeles_Datuma	timestamp

A rendelés státusza lehet: függőben, szállítás alatt, kiszállítva és törölve. Ezt az információt a vevő is látja a profiljában.

Amennyiben még nincs regisztrálva a vásárló, nem tud vásárolni amíg nem regisztrál az oldalra.

Az **osszeg** a rendelésért fizetendő teljes összeg, ami az alábbi táblából kerül kiszámításra.


3.2 Rendelt termékek

A termékeket a jobb átláthatóságért itt tároljuk el.

A **rendeles_id** köti az aktuális tételt egy megrendeléshez, a **termek_id** pedig hozzáköti a megvásárolt terméket.

Az **egysegar** a termék ára a megrendelés pillanatában, így lehetséges, hogy ugyan az az adat jelenik meg egyszerre két helyen, ám ha változik az ár a megrendelés után, akkor ezáltal a felhasználó ugyanúgy az eredeti árat fizeti ki.

A mennyiség segítségével nem csak a megrendelő láthatja, hogy pontosan mennyit is rendelt, hanem a rendelés végösszegét is segíti ez kiszámolni. A **szin_id** visszaigazoló célt szolgál, a vevő a vásárlás után is leellenőrizheti, hogy jó terméket vett-e.

rendeltTermek	
id 	int
rendeles_id	int
termek_id	int
mennyiseg	int
egysegar	decimal(10,2)
szin_id	int



4 Blog

4.1 Posztok tábla

A poszt tábla az oldal tartalmi részének alapját képezi. A poszthoz közvetlen tartozó adatokat tároljuk itt, vagyis a bejegyzések minden alapvető adatait. Az egyszerűbb átláthatóság és redundancia csökkentése miatt a további tartalmakat, például a képek, címkék vagy a kommentek, pedig külön táblában helyeztük el. Ezzel egy logikusabb adatstruktúrát kapunk, és az adatfeldolgozás is egyszerűbbé válik.

Az **id**, mint mindenhol máshol, a poszt egyedi azonosítója, ennek a segítségével kötjük a poszthoz a kommenteket, a képeket és a címkéket.

A **cim** egy rövid cím, ami bemutatja hogy miről fog szólni a poszt, a **kivonat** pedig egy rövid összefoglaló ami megjelenik a posztok böngészésénél, így a felhasználónak nem kell rákattintania a posztra hogy megtudja miről is szól.

A **tartalom** mező a poszt tényleges szövegét tárolja, amit csak azután válik teljesen láthatóvá miután a felhasználó a posztra kattint.

A **szerzo_id** segítségével tudjuk megjeleníteni a poszt íróját, ez a felhasználók táblával van összekötve.

A **fo_kep_id** a poszt borítóképét tartalmazza, amely vizuálisan is kiemeli a bejegyzést. Amennyiben a szerző nem ad saját képet, a rendszer automatikusan hozzárendel egy alapértelmezett borítóképet.

A **letrehozás_datuma** a poszt eredeti publikálása, a **modositas_datuma** pedig hogy mikor szerkesztették utoljára a posztot.

A **statusz** mutatja a poszt állapotát, ez lehet: piszkozat, közzétett, archivált.

posztok	
id	int
cim	varchar(255)
kivonat	varchar(255)
tartalom	text
szerzo_id	string
foKep_id	int
letrehozás_Datuma	timestamp
modosítás_Datuma	timestamp
statusz	enum

4.2 Kommentek tábla

A **kommentek** tábla a posztokhoz kapcsolódó hozzászólásokat tárolja. Minden rekord egy-egy felhasználói interakciót képvisel, amelyet az olvasók a bejegyzés alá írnak.

Az **id** itt is azonosító szerepet tölt be, illetve ha van alkomment a kommenthez, akkor erre hivatkozva kapcsoljuk őket egymáshoz.

kommentek	
id	int
komment	text
poszt_id	int
kommentelo	varchar(255)
elozo_komment_id	int
letrehozás_Datuma	timestamp
statusz	enum(aktiv,törölt)



A **komment** mező maga a komment szövegét tárolja, a **kommentelo**-vel pedig megjelenítjük a kommentelő felhasználónevét, ami a felhasználók táblából kéri az adatot.

Ahogy fentebb is említettük, a **poszt_id** köti a kommentet a megfelelő poszthoz. További kommenteket az **elozo_komment_id** segítségével tudunk megjeleníteni.

A `letrehozás_datuma` egy alapvető funkció kommenteléskor, a közzététel dátumát mutatja, a `statusz` pedig a komment állapotát, ami lehet aktív, vagy törölt.

4.3 Kiegészítő táblák a bloghoz

A **posztReakciok** tábla egy egyszerű, mégis hatékony rendszer a felhasználói visszajelzések kezelésére. Itt tároljuk, hogy ki hogyan reagált egy bejegyzésre: tetszik (like) vagy nem tetszik (dislike). Ehhez a **poszt_id**-t és a **felhasznalo_id**-t kötöttük, így egy felhasználó csak egyszer reagálhat egy posztra.

A **posztKeppek** a posztokhoz hozzáadott képek megjelenítéséért felelős, így a **poszt_id**-t és a **kep_id**-t kapja meg, így összekötve a két táblát. A **sorrend** segítségével pedig befolyásolni lehet a képek megjelenítését, így a sorba rendezése nem csak automatikus lehet. Ez a rugalmasság a tartalmat készítők számára fontos, mivel így maguk dönthetik el, hogy melyik kép jelenjen meg elsőként.

Végül pedig a **posztCimkek** segítik a tartalom rendszerezését és kereshetőségét. Ennek a segítségével könnyebben rátalálni a posztokra, mivel a termékekhez hasonlóan a posztok is címkéket kapnak, így egy általános képet adva a poszt témájáról, és segítve hogy a felhasználó könnyebben talál a tetszésének megfelelő hasonló témájú posztokat.

posztReakciok	
poszt_id	int
felhasznalo_id	int
reakcio	enum(tetszik,nemtetszik)

posztKeppek	
poszt_id	int
kep_id	int
sorrend	int

posztCimkek	
poszt_id	int
cimke_id	int



5 Kiegészítő táblák

Az itt található táblák mindegyikére már hivatkoztunk fentebb, most pedig részletesen ismertetjük ezeket is.

5.1 Címkék tábla

A többi táblához kötéskor az **id**-ra hivatkozunk, így az összekötés után le tudjuk kérni a címke nevét, és azt jelenítjük meg a termékeknel és a posztoknál is.

címkék	
id	int
nev	varchar(255)

5.2 Kategóriák tábla

A termékek kategória adásához az **id**-val való összekötést használjuk, így a címkékhez hasonlóan a lekérdezett kategória nevét ennek a segítségével tudjuk megjeleníteni. Hasonlóan a kommentekhez, az alkategóriák úgy kapcsolódnak a szülő kategóriájukhoz hogy a **fo_kategoria_id**-val megadjuk annak az azonosítóját.

kategóriák	
id	int
nev	varchar(255)
fo_kategoria_id	int

5.3 Színek tábla

Idegen kulcsként használjuk az **id**-t hogy a termékekhez, illetve a rendelt tételekhez színt kössünk. A szín neve a könnyebb azonosítást segíti a felhasználók számára, a **hex_kod**-al pedig eltároljuk a szín pontos paramétereit, így az oldalon meg tudjuk jeleníteni a színt is a neve mellett.

színek	
id	int
szin_nev	varchar(50)
hex_kod	varchar(7)

5.4 Képek tábla

Több táblában is idegen kulcsként használjuk az **id**-t, ennek a segítségével rendeljük a képeket a helyükre. Az **url_link** segít a képet megtalálni, az **alt_szoveg** pedig akkor jelenik meg, amikor a kép valamiért nem tud. A **leiras** segít megérteni a képet amikor szükség van rá.

kepek	
id	int
url_Link	varchar(255)
alt_Szoveg	varchar(255)
leiras	text

5.5 Városok tábla

Az átláthatóbb adattárolás érdekében a városokat külön táblában tároljuk el, így elkerülve a redundanciát és a pontatlanságokat.

varosok	
id	int
iranyitoszam	int
nev	varchar(100)