**Baranya Vármegyei Szakképzési Centrum**

**Simonyi Károly Technikum és Szakképző Iskola**

**Vizsgaremek**

**Pécs**

**2025**

Készítették: Pál Rajmund, Futó Csaba

**Baranya Vármegyei Szakképzési Centrum**

**Simonyi Károly Technikum és Szakképző Iskola**

**Pécs**

**2025**

Készítették: Pál Rajmund, Futó Csaba

Szakma megnevezése: Szoftverfejlesztő és –tesztelő

A szakma azonosító száma: 5 0613 12 03

**Vizsgaremek**

Wimu Webshop

# **1. Bevezető**

**Projekt Célja és Motiváció**

A Wimu Webshop létrehozásával célunk egy egyszerű, de hatékony online vásárlási platform fejlesztése volt, ahol elsajátíthattuk a webalkalmazások teljes életciklusát. A projektet azért választottuk, mert lehetőséget adott a gyakorlati ismeretek megszerzésére, kezdve az adatbázis-tervezéstől a felhasználói felület optimalizálásáig. A kezdeti kihívások (pl. kosárkezelés, session-kezelés) után komplex funkciókat valósítottunk meg, például fizetési módok integrálását és raktárkészlet-nyilvántartást.

**Tanultak és Új Ismeretek**

**A fejlesztés során elsajátítottuk**:

**Adatbázis-kezelés**: MySQL táblák tervezése (pl. termékek, felhasználók, rendelések).

**Biztonsági technikák**: Jelszavak titkosítása (Hash), SQL injection elleni védelem.

**Folyamatkezelés**: Felhasználói regisztráció, kosár tartalmának mentése, rendelés véglegesítése.

**Fizetési integráció**: PayPal SDK használata bankkártyás tranzakciókhoz.

**Reszponzív design**: Bootstrap segítségével mobilbarát felületek kialakítása.

**Későbbi Tervek**

**Jövőbeli fejlesztési céljaink**:

**Email-es regisztráció**: Jelszó-visszaállítás és regisztráció megerősítése.

**Rendeléskövetés**: Felhasználók számára áttekinthető szállítási állapot.

**Admin felület**: Termék- és rendeléskezelés intuitív vezérlőpultja.

**Értékelési rendszer**: Vásárlói visszajelzések megjelenítése.

**Csapatmunka és Szerepkörök**

A csapatban a feladatok így oszlottak meg:

**Csaba:** Backend fejlesztés (PHP, adatbázis logika, rendelésfeldolgozás).

**Rajmund:** Frontend tervezés (HTML/CSS, Bootstrap).

A kommunikációhoz Discordot használtunk. A kódot GitHubon tároltuk.

Projektszervezési Eszközök

**Discord (Kommunikáció):** Feladatok szétosztása és állapotuk nyomon követése (pl. "Kosár implementálása", "Fizetési módok hozzáadása").

**GitHub:** Kódmegosztás és együttműködés (branch-ek használata).

**Messenger:** Megbeszéltük, hogymikor csináljuk a vizsgaremeket.

**Összegzés**

A Wimu Webshop projekt lehetővé tette, hogy egy valós alkalmazás fejlesztésének minden szakaszában részt vegyünk. A nehézségek (pl. adatbázis-frissítések szinkronizálása) megoldása közben mélyebb betekintést kaptunk a webes rendszerek működésébe. A projekt továbbfejlesztésével célunk, hogy egy teljes körű, felhasználóbarát webshoppá nőjön ki.

# **Témaválasztás:**

Miért ezt választottuk?

Én Csaba azért ezt szerettem volna választani, mert nagyon sok webshop van a világban, de én is meg akartam nézni a folyamatát, hogy hogyan épül fel egy webshop.

Én Rajmund azért ezt választottam, mert mindig is érdekelt az üzleti oldala a dolgoknak, illetve mindig akartam egy saját boltot/webshopot nyitni.

**Közös cél:** Egy életképes prototípus létrehozása

Bár különböző motivációk vezettek minket, a cél egyértelmű volt: egy működő webshop prototípus, ami:

**Technikailag megalapozott**: Biztonságos, skálázható, könnyen bővíthető.

**Felhasználóbarát**: Intuitív navigáció, reszponzív design, gyors betöltés.

**Üzletileg releváns**: Valós igényeket szolgál (pl. címkezelés külföldre szállításhoz).

A projekt során megtanultuk, hogy a kommunikáció és a kompromisszumok nélkülözhetetlenek egy csapatban. Például amikor Rajmund egy összetett animációt akart a kosárhoz, de Csaba rámutatott, hogy az lelassítaná az oldalt, kompromisszumként egyszerűbb, de hatékony megoldást választottunk.

**Mi hoz a jövő?**

A webshop jelenleg egy **MVP** (Minimum Viable Product), de alapozóként szolgál további fejlesztésekhez:

**Email-es regisztráció**: Jelszó-visszaállítás és automatikus értesítések implementálása.

**Analitika**: Google Analytics integráció a vásárlói viselkedés nyomon követésére.

**SEO-optimalizálás**: A termékoldalak tartalmának finomhangolása keresőmotorok számára.

Csaba számára ez a projekt egy technikai referenciapont, Rajmundnak pedig egy üzleti alapkő. Mindketten abban reménykedünk, hogy a jövőben ezt a prototípust egy teljes értékű online vállalkozássá lehet fejleszteni.

# **3. Az alkalmazott fejlesztői eszközök:**

**1. Programozási Nyelvek & Keretrendszerek**

**PHP**: A backend logika (pl. kosárkezelés, felhasználókezelés) megvalósítása.

**MySQL:** Relációs adatbázis a termékek, felhasználók és rendelések tárolására.

**HTML/CSS/JavaScript:** Frontend felület reszponzív design-nal és interaktív elemekkel.

**Bootstrap 5**: Gyors és egységes felhasználói felület kialakítása (pl. kártyák, navigációs sáv).

**2. Fejlesztői Környezetek & Eszközök**

**Visual Studio Code**: Fő IDE kódírásra, debugolásra (PHP, HTML, CSS, JS).

**XAMPP**: Lokális szerver környezet (Apache, MySQL, PHP) teszteléshez.

**phpMyAdmin**: Adatbázisok kezelése, SQL lekérdezések futtatása.

**Git & GitHub**: Verziókövetés és csapatmunka koordinálása.

**3. Design & Médiaeszközök**

**PiktoChart**: Banner szerkesztése (pl. img/banner2.png).

**Leonardo.ai, MS Copilot**: Termék képek

**Bootstrap Icons**: Ikongyűjtemény a felhasználói felülethez.

**4. Könyvtárak & Függőségek**

**jQuery:** Dinamikus űrlapkezelés és AJAX kérések (pl. kosár frissítése).

**Popper.js & Bootstrap JS**: Interaktív elemek (dropdown menük, tooltip-ek).

**5. Egyéb Eszközök**

**Messenger/Discord**: Feladatok szétosztása és állapotkövetés.

**Google Chrome DevTools**: Frontend hibakeresés.

**6. Biztonsági Eszközök**

**BCrypt:** Jelszavak titkosítása (password\_hash() és password\_verify()).

**Prepared Statements**: SQL injection elleni védelem (pl. mysqli->prepare()).

**Session Hardening**: Cookie-k biztonságos beállításai (httponly, secure flag-ek).

**7. Kompatibilitás & Tesztelés**

**Cross-Browser Tesztelés**: Chrome, Firefox, Safari.

**Reszponzív Design Tesztelés**: Mobil-, tablet- és asztali nézet.

# **4. Tervezési módszer**

**4.1 Alkalmazás tervezése:**

A Wimu Webshop tervezése során a felhasználói igények és üzleti célok összehangolására fókuszáltunk. A tervezés három fő szakaszra oszlott:

**Követelményelemzés**: A vásárlók és adminisztrátorok igényeinek feltárása (pl. egyszerű regisztráció, kosárkezelés, rendeléskövetés).

**Rendszerfelépítés**: A háromrétegű modell (frontend, backend, adatbázis) kialakítása.

**Funkcionális specifikációk**:

Felhasználói szerepkörök (vendég, regisztrált felhasználó, admin).

Modulok (termékkatalógus, kosár, fizetési rendszer).

A tervezés során kiemelt szerepet kapott a reszponzív design, hogy a webshop minden eszközön (mobil, tablet, asztali) optimálisan működjön.

**4.2 Tervezési módszertan:**

A projekt agilis módszertant követett, amely lehetővé tette a folyamatos visszajelzés alapján történő iteratív fejlesztést.

**Csapatmunka**: Megbeszélés alapján összedolgoztunk, hogy mikor ér rá a másik kolléga.

**Felhasználó tesztelés**: User-ként bejelentkeztünk leteszteltünk főbb funkciókat.

**4.3 Vizuális modellezés:**

A rendszer vizuális reprezentációja kulcsfontosságú volt a tervezésben:

**Bootstrap-ek**:

Főoldal elrendezése (termékkártyák, navigációs sáv).

Kosár oldal felépítése (táblázatos nézet, gombok).

Adatbázis séma tervezése:

ER-diagram készült a táblák és kapcsolatok szemléltetésére (pl. arucikk ↔ kategoriak, ugyfel ↔ rendelesek).

Normalizálás az ismétlődések elkerülése végett.

**Folyamatábrák**:

Belépési és rendelési folyamatok vizualizációja.

**4.4 OOP megvalósulása:**

Bár a projekt főként PHP-t használ, objektumorientált elemeket is építettünk bele:

**Adatbázis kezelés OOP stílusban**:

A mysqli osztály használata prepared statement-ekkel:

$parancs = $kapcsolat->prepare("SELECT \* FROM ugyfel WHERE email=?");

$parancs->bind\_param("s", $email);

$parancs->execute();

Transaction kezelés a rendeléseknél

**Osztálytervek jövőbeli bővítéshez**:

**User osztály**: Felhasználói adatok és jogosultságok kezelése.

**Cart osztály**: Kosár műveletek (hozzáadás, törlés, frissítés).

**Product osztály**: Termékek lekérdezése és szűrése.

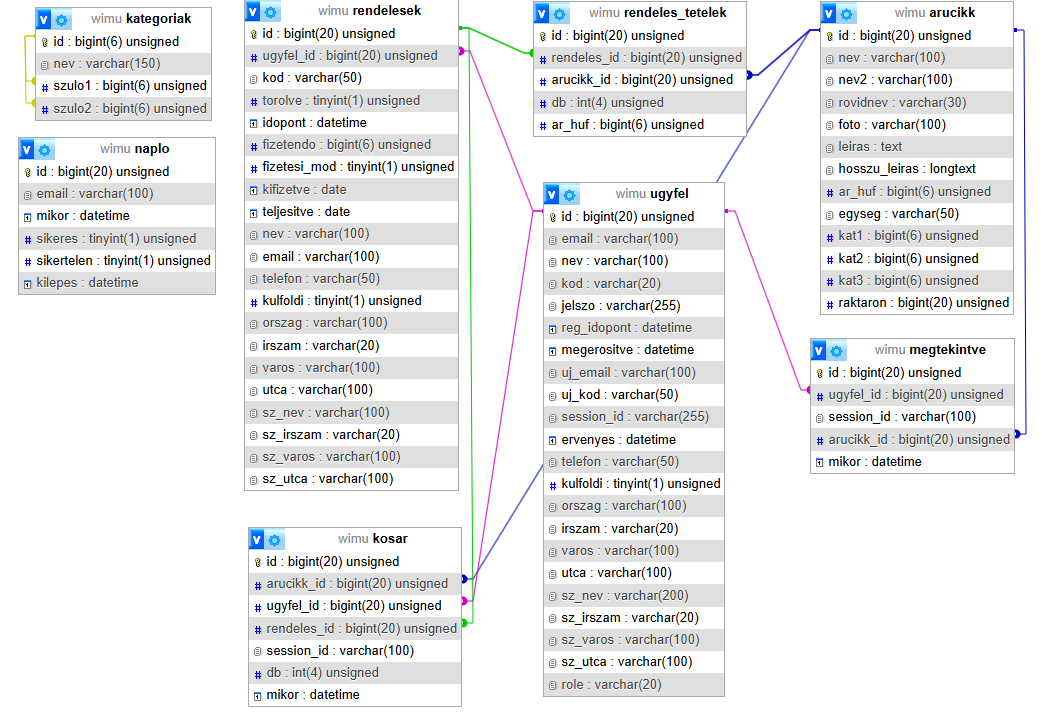
**Öröklés és egységbezárás**:

A Payment főosztályból származó alosztályok (pl. PayPalPayment, BankTransfer) a fizetési módok kezelésére. (fejlesztés alatt)

Privát metódusok biztonságos adatkezeléshez (pl. jelszó titkosítás).

MySql: Adatbáziskapcsolat egyszeres példányosítása.

# **5. Adatmodell leírása:**



**5.1 Főbb Táblák:**

* Termékek (arucikk)
  + Adatok: név, ár, készlet, kategóriák (3 szint), leírások.
  + Példa: Laptopok, workstationok.
* Kategóriák (kategoriak)
  + Hierarchikus struktúra (pl. Elektronika → Laptopok → Gaming).
  + Szülő-mezőkkel (szulo1, szulo2).
* Felhasználók (ugyfel)
  + Regisztráció, címek, titkosított jelszó (BCrypt), admin jogok (role).
* Kosár (kosar)
  + Session vagy felhasználóhoz kötött, ideiglenes adatok a rendelés előtt.
* Rendelések (rendelesek + rendeles\_tetelek)
  + Fizetési módok, címek, egyedi kód (67e7c1d056397).
  + Tételek: termék ID, darabszám, ár.

# 

**5.2 Fő Funkciók:**

* **Többszintű kategóriák**: Szűrés és navigáció egyszerűsítése.
* **Kosárkezelés**: Session-alapú vagy felhasználói fiókhoz kötött.
* **Rendelésfeldolgozás**: Szállítási/számlázási címek, külföldi támogatás.
* **Biztonság:** Titkosított jelszavak, bejelentkezési napló (naplo).

**5.3 Táblák Kapcsolatai**

Az adatbázis relációit idegen kulcsok (Foreign Keys) biztosítják:

Termék → Kategóriák:

ALTER TABLE arucikk ADD FOREIGN KEY (kat1) REFERENCES kategoriak(id);

Egy termék legfeljebb 3 kategóriába sorolható (kat1, kat2, kat3).

Kosár → Termékek:

ALTER TABLE kosar ADD FOREIGN KEY (arucikk\_id) REFERENCES arucikk(id);

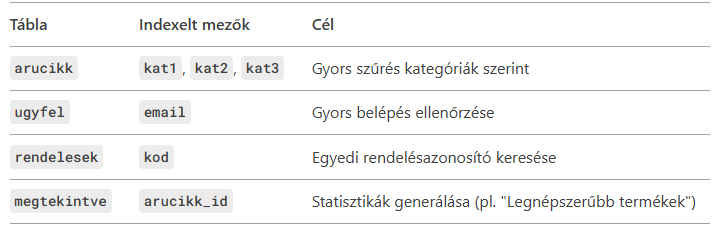
Egy kosár tétel mindig egy létező termékre hivatkozik.

Rendelések → Felhasználók:

ALTER TABLE rendelesek ADD FOREIGN KEY (ugyfel\_id) REFERENCES ugyfel(id);

Minden rendelés egy regisztrált felhasználóhoz tartozik.

**5.4 Indexek és Optimalizálás**

A teljesítmény növelése érdekében stratégiai indexek lettek létrehozva:

**5.5 Biztonsági Intézkedések**

**Jelszavak titkosítása**:

A ugyfel.jelszo mezőben BCrypt hashelés kerül használatra.

Példa: $2y$10$... – a 10 a titkosítási költség faktort jelöli.

**SQL Injection védelem**:

Prepared statement-ek használata PHP-ben:

$parancs = $kapcsolat->prepare("SELECT \* FROM ugyfel WHERE email=?");

$parancs->bind\_param("s", $email);

**Munkamenet kezelés**:

A ugyfel.session\_id és ervenyes mezők biztosítják a jogosulatlan hozzáférés kiszűrését.

**5.6 Jelentős Funkciók és Adatbázis Integráció**

**Többszintű kategóriák**

Megvalósítás: A kategoriak tábla hierarchikus struktúrája lehetővé teszi:

Akár 3 szintű kategória fa (pl. Elektronika → Laptopok → Gaming).

Rugalmas szűrés a terméklistákban (kat1, kat2, kat3 mezők).

**Kosárkezelés**

Vendég vásárlók: A session\_id menti a kosarat 7 napig (DELETE FROM kosar WHERE mikor<'$egyhete').

Készlet ellenőrzés: A kosar és arucikk táblák tranzakcióban frissülnek, hogy elkerüljük a túlértékesítést.

**Rendelésköveté**s

Státuszok: A rendelesek tábla fizetesi\_mod és teljesitve mezői nyomon követik a fizetést és szállítást.

**5.7 Későbbi Fejlesztések**

**Elavult kosarak törlése**: Heti automatikus törlés cron job segítségével.

**Full-text keresés**: A arucikk táblában a termékleírások gyors keresése.

**Többnyelvű támogatás**: Új mezők (pl. leiras\_en) a lokalizációhoz.

Ez az adatbázis szerkezet biztosítja a Wimu Webshop skálázhatóságát és a jövőbeli bővítések lehetőségét.

# **6. Részletes feladatspecifikáció, algoritmusok**

**6.1. belepes.php – Belépési Folyamat**

Funkció: Felhasználó hitelesítése e-mail és jelszó alapján.

**Paraméterek**:

$\_POST['email']: Felhasználó e-mail címe.

$\_POST['jelszo']: Felhasználó jelszava.



**Működés**:

JavaScript validálja az e-mail formátumot és a jelszó meglétét.

Szerver oldalon ellenőrzi az adatbázisban a felhasználó létezését (SELECT \* FROM ugyfel WHERE email='$email').

Hash-elt jelszó ellenőrzése password\_verify() függvénnyel.

Naplózza a sikertelen próbálkozásokat (napló tábla).

Ha 3 sikertelen próbálkozás van 30 percen belül, tiltja a belépést.

Sikeres belépés esetén beállítja a session-t és cookie-kat, átirányít a főoldalra.

Visszatérési érték: HTTP átirányítás hibákra (hiba=1, tilos=1) vagy sikeres belépés esetén index.php.

**6.2 Főbb függvények és metódusok specifikációja**

1. Kosár tartalmának betöltése (1\_kosar\_tartalma.php)

Funkció: A felhasználó kosarában lévő termékek megjelenítése, összegzés.



**Paraméterek**:

session\_id: A felhasználó session azonosítója (ha nincs bejelentkezve).

ugyfel\_id: A bejelentkezett felhasználó azonosítója.

**Adatbázis lekérdezés**:

SELECT \* FROM kosar WHERE session\_id = ? OR ugyfel\_id = ?;

Kimenet: HTML tábla a termékekkel, mennyiséggel és összesített árakkal.

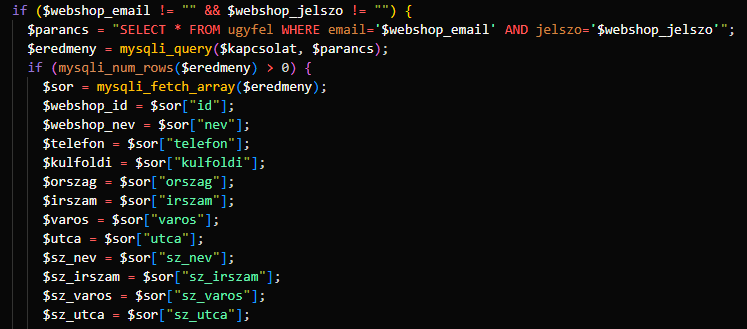
**6.3 Vásárlói adatok validálása (2\_vasarloi\_adatok.php)**

JavaScript függvény ellenoriz():

Funkció: Ellenőrzi, hogy minden kötelező mező ki van-e töltve.

Ellenőrzött mezők: Név, telefonszám, cím, számlázási adatok.

Hibakezelés: Alert üzenetek üres mezők esetén.



**Adatbázis frissítés**:

UPDATE ugyfel SET nev=?, telefon=?, ... WHERE id=?;

Paraméterek: Felhasználó által megadott adatok (pl. $\_POST['nev'], $\_POST['telefon']).

**6.4. Fizetési mód kiválasztása (3\_fizetesi\_modok.php)**

JavaScript függvény ellenoriz():

Funkció: Ellenőrzi, hogy legalább egy fizetési mód ki van-e választva.

Integráció: PayPal SDK használata bankkártyás fizetéshez.

Konverzió: HUF → USD fix árfolyammal (pl. osszeg / 350).

Adatátvitel:

<input type="hidden" name="fizet" value="1|2|3">

Értékek: 1 (banki átutalás), 2 (utánvét), 3 (PayPal).



**6.5. Rendelés összegzése (4\_rendeles\_osszegzes.php)**

Funkció: Megjeleníti a felhasználó kosarát és a kiválasztott fizetési módot.

**Adatbázis lekérdezés:**

SELECT arucikk.nev, kosar.db, arucikk.ar\_huf

FROM kosar

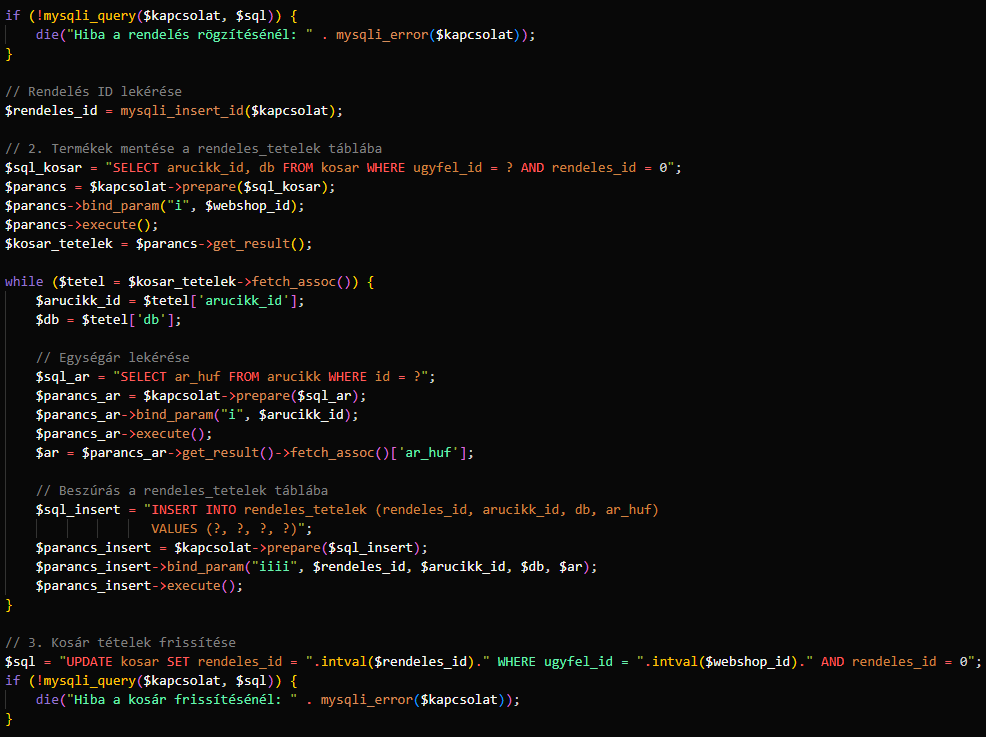
JOIN arucikk ON kosar.arucikk\_id = arucikk.id

WHERE ugyfel\_id = ?;

Kimenet: Tábla a termékek részleteivel és egy véglegesítő gomb.



**6.6 Rendelés feldolgozása (5\_rendeles\_elkuld.php)**

****

**Lépések**:

**Rendelés rögzítése**:

INSERT INTO rendelesek (ugyfel\_id, kod, fizetendo, ...);

**Termékek mentése**:

INSERT INTO rendeles\_tetelek (rendeles\_id, arucikk\_id, db, ar\_huf);

Raktárkészlet frissítése:

UPDATE arucikk SET raktaron = raktaron - ? WHERE id = ?;

**Kosár ürítése**:

DELETE FROM kosar WHERE ugyfel\_id = ?;

Kimenet: Sikeres rendelés üzenet és visszaigazoló e-mail szimuláció.

**6.7. Bejelentkezési rendszer (belepes.php)**

**Funkció**: Felhasználó hitelesítése és session kezelése.

**Ellenőrzések**:

**Jelszó ellenőrzés**: password\_verify($jelszo, $sor['jelszo']).

**Brute force védelem**:

SELECT COUNT(\*) FROM naplo WHERE email=? AND mikor > ? AND sikertelen=1;

Blokkolás, ha 3+ sikertelen próbálkozás van 30 percen belül.

**Cookie-k:** webshop\_email és webshop\_jelszo 7 napig érvényesek.

**6.8 Algoritmusok leírása**

**1. Kosár frissítési algoritmus**

Cél: A kosárban lévő termékek mennyiségének módosítása.

**Lépések:**

A felhasználó módosítja a mennyiséget a kosár oldalon.

Űrlap elküldése a mit=modosit paraméterrel.

**Adatbázis frissítés:**

foreach ($\_POST as $kosar\_id => $db) {

UPDATE kosar SET db = ? WHERE id = ?;

}

Összeg újraszámolása és átirányítás.

**6.9 Rendelésfeldolgozás folyamata**

**Struktogram**:

[START]

↓

[Adatbázis kapcsolat]

↓

[Rendelés létrehozása → rendelesek tábla]

↓

[Termékek másolása → rendeles\_tetelek tábla]

↓

[Raktárkészlet csökkentése]

↓

[Kosár ürítése]

↓

[Sikeres oldal megjelenítése]

[END]

**6.10. Termékkereső algoritmus (index.php)**

**Megjegyzések**

**Biztonsági hiányosságok:** SQL injection kockázat néhány lekérdezésben (pl. 5\_rendeles\_elkuld.php).

Jelszavak tárolása nyers formában helyett password\_hash() használata ajánlott.

**Optimalizálási lehetőségek:**

Tranzakciók használata rendelésfeldolgozásnál (ACID tulajdonságok biztosítása).

Cache-elés a gyakran lekérdezett termékadatokhoz.