

**WatchLux Projekt Dokumentáció**

Készítette: Makra Martin, Szilágyi István, Berényi Gergő

**1. A szoftver célja**

A WatchLux projekt egy olyan óra webáruház, amelyhez egy háttérrendszer is tartozik, ahol az adminok tudják kezelni a termékeket, a rendeléseket és a felhasználókat. A célunk az volt, hogy egy átlátható, könnyen kezelhető, modern webshopot hozzunk létre, ami jól működik mind a vásárlók, mind az üzemeltetők számára.

**A rendszer három fő részből áll:**

**- Webes felület (frontend):** itt böngészhetők a termékek, lehet szűrni, keresni, regisztrálni, belépni, kosárba rakni és rendelni.

**- Szerveroldal (backend):** ez kezeli az adatokat, API-kon keresztül kommunikál a frontenddel, és gondoskodik arról, hogy minden működjön.

**- Asztali admin alkalmazás:** itt tudják az adminok szerkeszteni az adatokat, emellett a titkári rangokat beosztani a korlátozott hozzáféréshez és új órákat felvenni, megrendeléseket kezelni.

A backend különféle szűrést és kulcsszavas keresést is tud, és rugalmasan kezeli a különböző terméktulajdonságokat (mint például márka, típus, vízállóság stb.). A rendszer minden része MySQL adatbázist használ.

A cél tehát nem csak egy sima webshop létrehozása volt, hanem egy jól működő, komplex rendszeré, amit a valóságban is könnyen lehetne használni.

**2. A rendszer főbb komponenseinek technikai leírása**

A WatchLux projekt három fő részből épül fel: egy webes vásárlói felületből (frontend), egy szerveroldali háttérrendszerből (backend), valamint egy asztali adminisztrációs alkalmazásból. Ezek a komponensek egymással együttműködve biztosítják a rendszer teljes funkcionalitását.

**2.1 Frontend – Webes felület (React)**

Ez a rész az, amit a látogatók, vásárlók használnak. A React alapú frontend biztosítja a gyors, interaktív és modern felhasználói élményt. A felület mobilbarát (reszponzív), vagyis telefonon és asztali gépen is jól használható.

**A főbb funkciók:**

- Terméklista megjelenítése: képekkel, árakkal, részletes adatokkal.

- Szűrés és keresés: különböző szempontok (pl. márka, ár, nem, meghajtás) alapján.

- Felhasználói fiók: regisztráció, bejelentkezés, profiladatok kezelése.

- Kosár és rendelés: termékek hozzáadása, mennyiség módosítása, megrendelés véglegesítése.

A felület Axios segítségével kommunikál a backenddel (REST API-n keresztül), az adatokat JSON formátumban kapja meg.

**2.2 Backend – Szerveroldal (Node.js + MySQL)**

A szerveroldal felelős a kapcsolati logika és az adatbázis-kezelés lebonyolításáért. A backend Node.js környezetben készült, az adatbázis pedig MySQL alapú.

**Fő jellemzők:**

- REST API: minden frontend kéréshez külön végpont tartozik (GET, POST, stb.).

- Dinamikus szűrés: a backend képes tetszőleges kombinációjú szűrőfeltételek alapján lekérni a termékeket.

- Kulcsszavas keresés: több oszlopban történő keresés egyetlen mezőből (pl. megnevezés, típus, márka, stb.).

- Adatbiztonság: jelszavak SHA-256 hash-eléssel kerülnek eltárolásra.

- Árintervallum lekérdezés: automatikusan számolt minimum és maximum ár alapján.

A backend biztonságosan kapcsolódik az adatbázishoz a mysql2/promise csomag segítségével, és minden lekérdezés paraméterezett, hogy megelőzze az SQL injection támadásokat.

**2.3 Asztali admin alkalmazás – C# WPF**

A rendszer harmadik komponense egy Windows-os asztali alkalmazás, amelyet az adminisztrátorok és titkárok használnak. A program C# nyelven, WPF- ben készült.

**Fő funkciók:**

- Termékek kezelése: új órák felvétele, meglévők módosítása vagy törlése ugyan ezt a táblák kezelésével.

- Felhasználók és vásárlók kezelése: adatok megtekintése, módosítása, új felhasználók felvétele.

- Rendelések megtekintése: megrendelések listázása, részletek megtekintése.

- Szűrés és keresés: termékek és vásárlók között is lehet keresni és szűrni különböző feltételek alapján.

A felület átlátható, logikusan felépített, minden mezőhöz megfelelő validáció tartozik. Az alkalmazás közvetlenül a MySQL adatbázissal kommunikál.

**3. Működés műszaki feltételei**

A rendszer működéséhez a backendet és a frontendet külön kell elindítani Visual Studio Code-ban. Mindkettő Node.js alapú, ezért az indulás előtt telepíteni kell a szükséges csomagokat.

**Backend elindítása**

1. Visual Studio Code-ban nyisd meg a backend mappát.

2. Jobb kattintás az üres részre, majd válaszd: Open in Integrated Terminal.

3. A terminálba írd be:

elsőnek: npm install

majd az előzőt követően: npm start

4. Ez elindítja a backend szervert, ami az adatokat az adatbázisból fogja kiszolgálni.

**Frontend elindítása**

1. Nyisd meg a frontend mappát is Visual Studio Code-ban.

2. Ismét jobb kattintás → Open in Integrated Terminal.

3. A terminálba írd be:

elsőnek: npm install

majd az előzőt követően: npm start

4. Ez elindítja a webes felületet, amit a böngészőben lehet használni (http://localhost:3000 címen).

**Adatbázis beállítása**

A rendszerhez szükséges egy 'ora' nevű MySQL adatbázis. Ezt a .sql fájlt importálni kell például XAMPP segítségével.

Felhasználónév: root

Jelszó: root

Ez a beállítás azért ilyen egyszerű, hogy a vizsgáztatók gyorsan ki tudják próbálni a rendszert. Természetesen ez bármikor megváltoztatható a dbconfig.js vagy. env fájlban és akár létrehozható külön felhasználó is.

**Admin panel (WPF alkalmazás)**

Az admin felület egy különálló Windows alkalmazás, amit egy installer segítségével lehet elindítani. A program használatához is szükséges, hogy az adatbázis elérhető legyen, tehát az 'ora' adatbázisnak futnia kell.

**Fontos:** a rendszer minden része úgy lett kialakítva, hogy a vizsgáztatók számára könnyen tesztelhető legyen. A beállítások (pl. adatbázis felhasználónév, jelszó) egyszerűek, de biztonságosabb értékekre is lecserélhetők a fájlokban.

A működéshez elég egy alap Node.js és MySQL környezet, Visual Studio Code és Visual Studio használata ajánlott.

A rendszer modularitása lehetővé teszi, hogy később könnyen bővíthető vagy módosítható legyen.

**4. A rendszer használatának rövid bemutatása**

A WatchLux webáruház használata egyszerű és gyors. A látogatók a kezdőlapon a termékek fülre átkattintva azonnal böngészhetnek az órák között. Lehetőség van különböző szűrési feltételek beállítására, például márka, ár vagy típus szerint. A termékeket listában látják, képpel és alapinformációkkal. Egy termékre kattintva megjelenik a részletes oldal, ahol hozzá lehet adni a kosárhoz és a kedvencekhez is.

A kosárban szerkeszthető a darabszám azaz mennyiség és törölni is lehet a kosárból a terméket.

A vásárláshoz nem szükséges a regisztráció vagy bejelentkezés hiszen vendég vásárlóként is lehet vásárolni. A felhasználó a saját fiókjában módosíthatja adatait, és megnézheti korábbi rendeléseit is. A rendelés leadása után visszaigazolást kap.

Az admin panel egy különálló alkalmazás, amelyben az adminisztrátor vagy a office kolléga hozzáadhat új órákat, módosíthatja a meglévőeket, kezelheti a vásárlók adatait és nyomon követheti a megrendeléseket emellett fontos kiemelni hogy csak az admin jogosultságú alkalmazott/ tulaj tudja kezelni a felhasználókat. Az adminisztrációs felület a bejelentkezésnél leellenőrzi a jogosultságot az adatbázisban a felhasznalok táblában ez admin vagy office jogosultság lehet ennek megfelelően tölti be a rendszer a szerkeszthető ablakokat. Az admin felület egyszerűen kezelhető, minden funkció pár kattintással elérhető.