MTG CARDSHOP

Az MTG azaz Magic The Gathering egy gyűjtögetős kártyajáték, ami 1993-óta van kint a piacon és a mai napig is nagy érdeklődés van, mint a gyűjtők, mint a játékosok részéről. Közben Újabb és újabb “Sett”-ek (kiadasok) érkeznek a játékba (úgy, mint a Gyűrűk ura kiadas) és új játékmódok, hogy friss maradjon a játék mindenki számára. A legnépszerűbb formátum (jatek mod) a Commander ahol 4 játékos küzd meg egymással előre elkészített 100 különféle lapokból álló paklikkal, de vannak versenyformátumok is mint a Standard vagy a Pioneer ahol szemtől szemben mérhetik össze a játékosok tudásukat, ezek a versenyek igen magas pénz nyereménnyel is bírhatnak.

Ez a projekt egy vállalkozó részére készül, a megbízás arról szól, hogy az éppen most nyíló kártyaboltjához szeretne egy weboldalt, ahol egyenként el tudja adni a kártyákat és különböző termékeket. Ezen kívül a vásárlók és játékosok meg tudják tekinteni az eseményeket egy naptárban az oldalon, ahol különböző versenyek, új kiadásokat és közösségi programok tekinthetők meg.

A projekt célja egy online kártya webáruház létrehozása, amely lehetővé teszi a felhasználók számára, hogy a kártyák kinézetét, kiadási számát és egyéb adatait összegyűjtsék, és egyetlen platformon rendelhessék meg azokat. A feladatot ketten dolgozzátok, és a feladatokat a következőképpen osztottátok fel:

* **Beni** a webes felületen dolgozik (HTML + CSS + JavaScript) --> indoklás
* Azért a HTML-t választottam a webdokumentum megjelenítésére, mert ez volt, amit tanultunk elsősorban. A formázásra a CSS-t választottam, mivel számomra ez könnyebben használható, mint a Bootstrap vagy a másik különféle formázási lehetőségek. Az oldal interakcióihoz JavaScriptet alkalmazok, mivel ez a legmegfelelőbb szerintem.
* **Marci** az asztali alkalmazást fejleszti (C# WinForms) miert keszul az alkalmazas
* Azért van szükség asztali alkalmazásra, mivel a megrendelésben külön kérve volt egy alkalmazás, ahol az adminisztratív dolgokat tudja a tulaj kezelni, és ehhez csak ő tud hozzáférni. Itt tudja ellenőrizni a termékeket, és szintén itt tudja hozzáadni az újabb termékeket.
* Az adatbázist és a backendet közösen készítitek (MySQL + Node.js)
* Az adatbázist mi MySQL adatbázissal készítjük el, amit a Node által nyújtott Express backend szolgáltatással fogunk elkészíteni, és ennek a végpontjait fogja elérni egyrészt a webes, másrészt az asztali alkalmazásunk.

### **1. Problémaelemzés és követelmények összegyűjtése**

* **Felhasználói igények**:
  + Felhasználók számára könnyen kereshető kártyák, melyek információkat tartalmaznak (kinézet, kiadási szám, egyéb jellemzők).
  + A kártyák megrendelése egyetlen platformon keresztül történjen.
* **Technológiai követelmények**:
  + Webes és asztali alkalmazás integrációja.
  + Adatbázis létrehozása MySQL-ben.
  + Backend fejlesztés Node.js és Express segítségével.
  + Webes felület fejlesztése (Beni).
  + Asztali alkalmazás fejlesztése (Marci).

### **2. Rendszertervezés**

* **Adatbázis tervezés (közösen)**:
  + **MySQL** adatbázist terveztek, amely tartalmazza a kártyák adatait, felhasználói adatokat és rendelési információkat.
  + A fő táblák: -- fontosab entitások
    - **Kártyák**: kártya ID, név, kiadási szám, kép, kategória, ár, készlet, stb.
    - **Felhasználók**: felhasználó ID, név, email, jelszó, stb.
    - **Rendelések**: rendelés ID, felhasználó ID, rendelési dátum, kártyák, mennyiség, rendelési státusz.
* **Backend fejlesztés (Node.js + Express, közösen)**:
  + Készítsetek API-kat az adatbázissal való kommunikációra.
  + API-k példái:
    - **Kártyák kezelése** (lekérés, hozzáadás, módosítás)
    - **Rendelések kezelése** (új rendelés létrehozása, rendelés státusz frissítése)
    - **Felhasználói fiók** (regisztráció, belépés, profilkezelés)

### **3. Feladatok felosztása**

* **Beni (Webes felület)**:
  + **Weboldal tervezés**:
    - Kártyák keresése, szűrése és megjelenítése.
    - Felhasználói fiók létrehozása, bejelentkezés, rendeléskezelés.
    - Kosár és rendelés funkciók.
    - Reszponzív dizájn: biztosítani kell, hogy a weboldal minden eszközön jól működjön.
  + **Technológia**: HTML, CSS, JavaScript (React vagy más front-end keretrendszer javasolt, ha szükséges).
* **Marci (Asztali alkalmazás)**: táblazatok grafikonok
  + **Asztali alkalmazás fejlesztés** (C# WinForms):
    - Az asztali alkalmazás képes tablazatok vagy grafikonok formajaban megmutatni a termekek kulonbozo atlagait eredmenyeit
    - A termékeke kezelese az adatbazisban
  + **Adatkommunikáció a backenddel**:
    - Az asztali alkalmazásnak API-hívásokat kell indítania a Node.js backendhez a kártyák, rendelésadatok és felhasználói adatok eléréséhez.

### **4. Fejlesztési lépések+**

1. **Adatbázis létrehozása (közösen)**:
   1. MySQL adatbázis létrehozása és táblák tervezése (eladok, kartyak, kartyak\_tipusok, tipusok, vevok, vevo\_kartyak\_vasarok).
   2. Adatbázis kapcsolódás a backendhez.
2. **Backend fejlesztés (Node.js + Express)**:
   1. API végpontok létrehozása a kártyák és rendelésadatok kezelésére.
   2. API végpontok a felhasználói autentikációhoz és rendelési folyamatokhoz.
   3. A backend folyamatos tesztelése és dokumentálása.
3. **Webes felület fejlesztése (Beni)**:
   1. Frontend oldal tervezése és implementálása (HTML, CSS, JavaScript).
   2. Az API-al való kommunikáció biztosítása, kártyák és rendelési információk megjelenítése.
   3. Kosár és rendelési funkciók megvalósítása.
   4. Felhasználói fiók létrehozása, belépés, regisztráció.
4. **Asztali alkalmazás fejlesztése (Marci)**:
   1. Asztali alkalmazás tervezése és implementálása **C# WinForms** segítségével.
   2. Az alkalmazás kommunikáljon a backenddel API hívásokon keresztül.
   3. Felhasználói interakciók tervezése, rendeléskezelés a felhasználói felületen.
5. **Tesztelés**:
   1. **Funkcionális tesztelés**: Teszteljétek le az összes funkciót (kártyák keresése, rendelés leadása, szűrés).
   2. **Integrációs tesztelés**: A webes felület és az asztali alkalmazás közötti adatkommunikáció tesztelése.
   3. **Biztonsági tesztelés**: Az API végpontok biztonsági tesztelése, belépési és hozzáférési mechanizmusok.
   4. **Felhasználói tesztelés**: A rendszer valós felhasználókkal történő tesztelése.
6. **Indítás és karbantartás**: felhasznalasi dokumentacio
   1. Weboldal és asztali alkalmazás élesítése.
   2. Folyamatos hibajavítások és frissítések a felhasználói visszajelzések alapján.
   3. Felhasznaloi dokumentacio