Békéscsabai SZC Nemes Tihamér Technikum és Kollégium

📚 Bookshop

Szakképesítés:  
Szoftverfejlesztő és -tesztelő

Készítette:  
Szabó Zoltán   
Szabó Bence Dániel

Békéscsaba, 2025

Tartalomjegyzék

[Bevezetés 1](#_Toc197097290)

[Témaválasztás indoklása 1](#_Toc197097291)

[Célkitűzés 1](#_Toc197097292)

[Fejlesztői dokumentáció 2](#_Toc197097293)

[Fejlesztői Környezet 2](#_Toc197097294)

[Felhasznált programok/eszközök 2](#_Toc197097295)

[Adatbázis felépítése, működés 3](#_Toc197097296)

[Adatbázis ismertetése 3](#_Toc197097297)

[Adatbázis feépítése 4](#_Toc197097298)

[books és kategóriák tábla 4](#_Toc197097299)

[Users tábla 5](#_Toc197097300)

[User\_tokens tábla 7](#_Toc197097301)

[game tábla 7](#_Toc197097302)

[Wishlist tábla 7](#_Toc197097303)

[orders tábla 8](#_Toc197097304)

[Ordered\_book tábla 8](#_Toc197097305)

# Bevezetés

### Témaválasztás indoklása

A témaválasztásunk célja egy online könyvesbolt megtervezése és megvalósítása volt. Jelenleg Magyarországon kevés korszerű könyves webshop működik, a meglévő áruházak többsége elavult megjelenésű, és funkcióik sem felelnek meg a mai vásárlói igényeknek. A mi BookShopunk könyveket kínál, amelyeket a felhasználók kényelmesen, online vásárolhatnak meg. A sablont a felhasználói élményre fókuszálva alakítottuk ki, modern, letisztult és könnyen kezelhető designt alkalmazva. Így webshopunk mind funkcionalitásában, mind megjelenésében megfelel a mai elvárásoknak. A felhasználóbarát felület lehetővé teszi a gyors és egyszerű vásárlást. Emellett regisztrált vásárlóink számára egy bónusz játékot is kínálunk, amelyben naponta egyszer játszhatnak, és értékes kedvezményeket, illetve kuponokat nyerhetnek.

### Célkitűzés

Nagyon fontos alappillérként kezeltük a SEO (Search Engine Optimization) alapelveinek alkalmazását, hogy a webáruház éles indulásakor a Google megfelelő helyezést biztosítson, ezáltal növelve az organikus keresések számát és a webshop értékét. Mivel Localhost környezetben dolgoztunk, a SEO-t egyelőre csak alap szinten tudtuk megvalósítani, de az élesítéskor ez könnyedén javítható lesz. Célunk volt egy széles termékkínálat létrehozása, így több mint 300 könyvet regisztráltunk a webshopba, amely a jövőben egyszerűen bővíthető. Külön admin felületet alakítottunk ki, ahol könyveket és kategóriákat tudunk hozzáadni, módosítani, törölni. Fontos szempont volt, hogy a webshop bejelentkezés nélkül is használható legyen, bizonyos korlátozásokkal, hogy a regisztráció előnyei is érvényesüljenek. Oldalunk nem gyűjt privát adatokat (például IP-címeket), így biztonságos felhasználói élményt biztosít. Vásárlás regisztráció nélkül is lehetséges, azonban kuponok így nem szerezhetők vagy válthatók be.

# Fejlesztői dokumentáció

### Fejlesztői Környezet

Webshopunk a projekthez méltó módon webes felületen érhető el. Teljes Frontend- és Backend-rendszert építettünk ki, hogy megkönnyítsük a fejlesztést, hiszen a különválasztott rétegek kezelése átláthatóbbá és gyorsabbá teszi a munkát. Emellett létrehoztunk egy asztali alkalmazást is, amely egy játék, és közvetlenül a weboldalról indítható. A fejlesztéshez a Visual Studio Code lehetőségeit, valamint különböző extension-öket használtunk, továbbá XAMPP környezetben is végeztünk módosításokat, hogy az útvonalkezelést egyszerűsítsük. Ezzel a fejlesztői munka hatékonyságát és pontosságát növeltük. Bár a közös munka miatt külön fejlesztői környezeteket is kialakíthattunk volna, mi végig Git verziókezelőt használtunk, így bármilyen gépen, bármikor tudtunk dolgozni a projekten.

### Felhasznált programok/eszközök

A teljes Frontend- és Backend-fejlesztést a **Visual Studio Code**-ban végeztük. Fontos szempont volt, hogy az általunk használt összes programnyelvet kezelje, és olyan beépített funkciókat biztosítson, mint a kódkiegészítés, Live Server, driverek, beépített Git-támogatás és a strukturált fejlesztést segítő bővítmények. A projekt körülbelül 60%-a PHP-alapú, ami egy webshop esetében teljesen megszokott, hiszen folyamatosan adatbázis-műveleteket végzünk: adatokat kérünk le és mentünk. A két réteg közötti kommunikációt több segédréteg támogatja, például a regisztráció során a telefonszámot különböző formátumokból (pl. +3630…, 0630…) egységesen, a magyar szabvány szerint alakítjuk át a rendszerben.

Az asztali alkalmazásunk, amely egy játék a regisztrált felhasználóknak van kialakítva **Visual Studio 2022**-ben készült C# nyelven. Fontos kitétel volt, hogy indítható legyen a weboldalról, amelyet meg is oldottunk, illetve az adatbázissal való teljes kommunikációt is képítettük, tehát bele írni, és kiolvasni is tud adatokat, amely lehetővé teszi a dinamikus adatkezelést és nyeremények könyvelését, a felhasználó játék elindításának idejét is lementjük, amellyel kritérium vizsgálatokat végzünk a továbbiakban.

**XAMPP** biztosítótta nekünk az adatbázis elindítását, vezérlését, és Apache szerver indítását. Az általunk használt verziók 8.0.30-8.2.12 problémamentesen működött minden, azonban eltérő verziók, amelyekeben elemi változások vannak bizonyos modulokban komplikációk léphetnek fel. Fontos, hogy a vizsga remekünkbe eltérő útvonalakat használtunk, illetve mappák neveit máshogy deklaráltuk, mint amelyen néven hivatkozunk rájunk, ez azért történt, hogy a strukturalitás megmaradjon, azonban a fejlesztés mégis egyszerűen, közérthetően menjen, egyszerűsített útvonalak pedig az ellenőrzést és a tesztelést is segítették.

A **GitHub-**ot használtuk, ugyanis globálisan szinte bárhonnan elérhető, illetve a repository-k klónozható bárki számára. Ahogy említettem korábban a Visual Studio Code és a Visual Studio 2022-es programokban is bele van építve, és egy egyszerű bejelentkezéssel hozzáférünk a fiókunkhoz. Ennél robosztusabb megoldást nem is találhattunk volna, teljesen kiszolgálta az igényeinket a fejlesztés során.

A weboldal logikája összefoglalva Front és Backend struktúra, amely dinamikusan képzi le az adatokat az adatbázisból, a kommunikáció asszinkron módon történik, hogy a dinamikusság megmaradjon, ne kelljen az oldalt újratölteni. Ez a **AJAX** (Asynchronous JavaScript and XML) mechanikájával történik, amely zökkenőmentesen, gyorsan tölti be az adatokat, az oldal újragenerálása nélkül. A rugalmasságot és a teljes dinamikusságot Javascript-el érjük el, ugyanis így sok időt spórulhatunk meg, illetve a kód ismétlést is elkerüljük. A JS kód részekben „Crazy JS” kódnak felel meg, amelyre nincsen pontos szakkifejezés, de valami ilyesmi lenne: „Advanced JavaScript Hacks vagy Creative Coding”. Ezt a decorate.js-be pontosan látni lehet, ahol egy változákony háttért generálunk le, amelynek alapja ebben az esetben matematika. Mindenek felett azonban az volt a lényeg, hogy érjük el a teljes kommunikációt, tehát minden-mindennel kommunikáljon, ez természetesen sikerült is.

# Adatbázis felépítése, működés

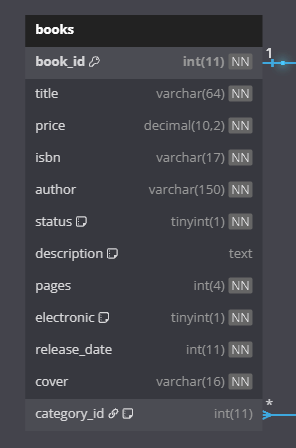
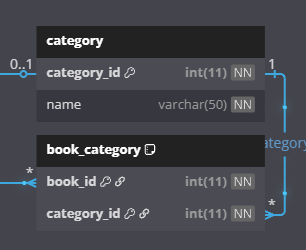
### Adatbázis ismertetése

A projektünk megtervezésekor a modellalapú gondolkodás mellett döntöttünk, és a Database First fejlesztési modellt választottuk. Ennek a módszernek a lényege, hogy először az adatbázis szerkezetét alakítjuk ki: megtervezzük a táblákat, definiáljuk a mezőket, és meghatározzuk az egyes entitások közötti kapcsolatokat. A programkód fejlesztését ezután az elkészült adatbázis-struktúrához igazítottuk.

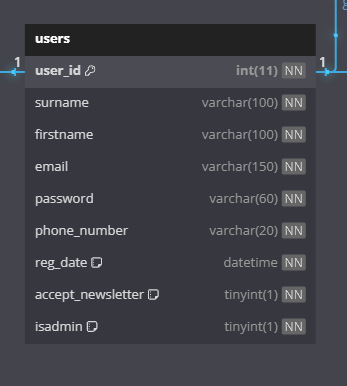
A fejlesztés során MySQL-t használtunk, amely kiválóan alkalmas gyors, megbízható és jól skálázható adatbázisok létrehozására. A Database First megközelítés nemcsak MySQL esetén, hanem más adatbázis-kezelő rendszerek (például PostgreSQL vagy Microsoft SQL Server) mellett is alkalmazható. Webáruház típusú rendszereknél ez a modell különösen hasznos, hiszen a jól strukturált adatbázis már a kezdetektől biztosítja a stabil működés alapjait. Így a fejlesztési idő jelentősen lerövidül, a hibalehetőségek csökkennek, és az esetleges befektetett költségek is minimalizálhatóak.

### Adatbázis feépítése

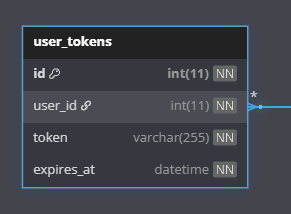
### books és kategóriák tábla

A **books** táblában találhatóak a könyvek alapvető információi. A book\_id egyedi kulcs, melyen keresztül csatlakozik és kapcsolatban áll a további táblákkal. A Cover tartalmazza a képet, amely át van konvertálva, egy 16 hosszú varcharra, így körülbelül 3000 könyvet is fel tudunk tölteni. A képek fellettek töltve file-ként és ezek nevei lettek eltárolva az adatbázisban. Próbálkoztunk blob-al, azonban azzal nem tudtuk kellően jól megjeleníteni a képeket, így az első verzióval dolgoztunk. Feltöltéskor az admin választhat, hogy a könyv elérhető-e vagy még nem (olyan eset, hogy elfogyott, vagy nincs raktáron, előre rendelhető stb.), illetve, hogy elektronikus-e vagy nem. Elektronikus könyveknek nincs isbn-je, azonban a mi weboldalunk esetében generálunk nekik, ugyanis minden könyv ’unique’ azonosítóval rendelkezik. A category\_id-n keresztül csatlakozik a kategóriához, amely a nagy kategória megnevezése, ilyen pl. Történelmi, Fantasy, Képregény stb. azonban ez még csatlakozik a **book\_category**-hoz, amely egy párosító tábla. Itt történik a kategóriához való könyv hozzárendelés, egy könyv több kategóriához is kapcsolódhat, amely fontos, hiszen sok könyv sorolható más-más kategóriába. A **category** tábla a kategória neveket tárolja el, ezeket említettem feljebb. Fontos a szeparáltság és strukturáltság, ezért amit lehet külön-külön tárolunk el táblákban.

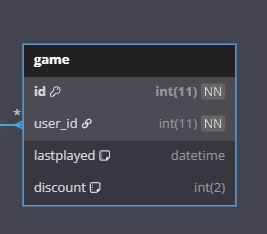
### Users tábla

A **users** tábla tartalmazza a felhasználói adatokat, amelyeket regisztrációnál megad. A jelszó természetesen hash-elve van, binhex függvény segítségével, amely röviden annyit takar, hogy a jelszó 60 karaktert hosszú és bycrypted. A telefonszám megadásánál minden esetben átformázzuk, hogy csakis magyar +36 (30) vagy +36 (20) formában jelenjenek meg, ugyanis regisztrációnál megadható ugyan ilyen formában, vagy esetleg 3630 vagy 0630 stb. Lényegében egy volt a cél, az egységesítés, amelyet a fentebb említett konvertálással oldottunk meg. A felhasználó bejelentkezése után, hogy pl. oldalváltáskor vagy tranzakcióknál, oldalon való interaktálásnál tokennel azonosítsuk. Mi nem a JWT tokent használtuk, hanem generálunk egy 16 számjegyből álló hexademical secure tokent, amely számokból (1-9) és hexaszámokból (10-A, 11-B, 11-C) stb. Fontos megemlíteni, hogy user\_id egy auto incrementes mező, mely növeli az értéket minden egyes új regisztrált felhasználó után. Ez nagyon hasznos, ugyanos a felhasználókat, tranzakcióknál, vagy beazonosító adatbázis műveleteknél könnyen, egyszerűen tudjuk lekérdezni, létrehozni. A regisztráció amikor megtörténik a pontos időt eltároljuk, így tudunk regisztrációs statisztikát is készíteni, amely SEO weboldalaknák nagyon jól jön, ugyanis látjuk melyek azok az időszakok amikor a webshopunk aktív. Ha a hírlevelet valaki elfogadja, ezzel jelezi számunkra, hogy nyitott az emailek fogadására. Ezt a funkciót nem építettük teljes mértékben ki, hiszen az „email magneting” egy külön szakterületként is felfogható, és nagyon időigényes lett volna, különböző, aktuális ajánlatokról üzeneteket küldeni, azonban a jövőre való tekintettel mégis elkészítettük ennek az alapját. Az utolsó mező az „isadmin” amely egy boolean típusú mező, és ennek az értéke mondja meg, hogy ki rendelkezik admin joggal. Ha az értéke 1 akkor admin, ha 0 akkor felhasználó. Ha admin jogot akarunk valakire ruházni, akkor az adatbázisban kell módosítanunk ennek a mező értékét. Nem akartuk, hogy ha valaki admin a weboldalon, akkor közvetlenül bárkire tudjon ruházni jogokat, ezért jogot csak az adminbázist vezérlő személy tudja. A webshopunknál szinte minden adatot dinamikusan az adatbázisból töltünk be, ezért a legnagyobb jogkör az adatbázis tulajdonosa, illetve kezelője.

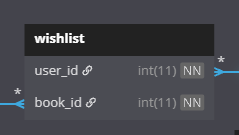
### User\_tokens tábla

A **user\_tokens** tábla a tokeneket kezeli, amikor valaki belép a weboldalra azonosítjuk. Nem névvel vagy emailel, hanem tokennel, ugyanis ez a legbiztonságosabb módszer, és valószínűleg a legpopulárisabb is. Bejelentkezésnél eltároljuk a tokent, amely egy bizonyos ideig érvényes, majd ha lejárt a token élettartama, akkor tölrésre is kerül az adatbázisban. A token bin2hex fügvénnyel generálódik le véletlenszerűen, és 64 karakter hosszú.

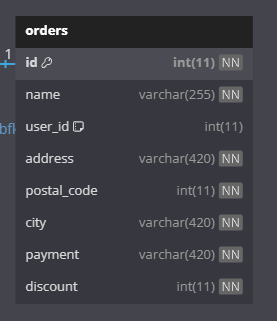
### game tábla

A **game** táblában a játék adatai, hogy a felhasználó pontosan mikor játszott, és nyert e leárazást. Minden felhasználó napi egyszer játszhat, és minden játszása log-olva lesz az adatbázisban, hogy pontosan lássuk, mikor játszott és mennyit nyert. Ez azért is jó, mert ha valami bug adódik véletlen, vagy egy hibát akarnak kihasználni esetlegesen, akkor látni tudjuk, hogy mikor ki játszott, így mindenki számára tisztességes játékot tudunk biztosítani. Ha valaki soha nem játszott még, csak regisztrált akkor egy alapértelmezett dátum kerül megadásra amely a „2000-01-01” ez azért is fontos, mert biztosítja, hogy a felhasználó nem játszott még, tehát 24 órán biztosan kívül esik. A „discount”/kupon amelyet szerezhet, lehet 10%, 25% de akár 50%-is. Ezt pontosan a játékba írt algoritmus határozza meg.

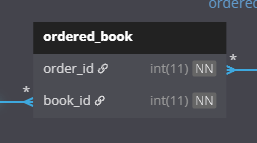
### Wishlist tábla

A **wishlist** táblába mentődik minden egyes kívánságlistához adás. Ez azért fontos, mert ha kijelentkezünk, és mondjuk másnap bejelentkezik a felhasználó, akár más eszközről, akkor megtalálja a kívánságlistájában az oda hozzáadott könyvet. Az adatbázisban csak a user\_id és a book\_id van eltárolva, azonban a kivánságlistában betölti a könyv teljes adatait pl. cím, kép, így egy kényelmes módszer minden felhasználó számára, aki csak nézelődik és ha megtetszik neki egy könyv hozzáadja, és le is tárolódik. Azonban ha nem vagyunk bejelentkezve, akkor számunkra a kosár mellett nem lesz elérhető a kívánságlista ikonja. Ez azért is fontos mert user\_id-t mentünk és legyen előnye annak, ha valaki beregisztált és felhasználóvá vált.

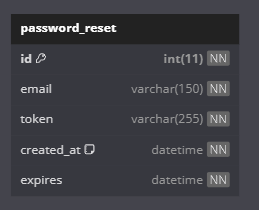
### orders tábla

Az **orders** táblában megtaláljuk a rendeléseket. A rendelések lehetnek bejelentkezetett felhasználókhoz kötve, az ilyen tagok tudnak kupont érvényesíteni, amelyet a rendelés leadása után már a pontos, kiszámolt árral kerül be az adatbázisba. Számlázási névhez mindig kell nevet adni, ugyanis vásárolhatunk és ha nem magunknak akarjuk, hanem egy családtagunknak, így tudjuk neki is számlázni, amely előny ilyen helyzetekben. Ha regisztráció nélkül vásárlunk, akkor egyértelműen fontos, hogy tudjuk kinek kell kiállítani a számlát, azonban fontos ilyen esetekben nem tudunk kupont érvényesíteni.

### Ordered\_book tábla

Az **ordered\_book** táblában az egyes felhasználókhoz megrendelt könyvek vannak beírva, egy felhasználóhoz nyílván több rendelés, és több könyv is társulhat.

### Password reset tábla

asdasda