# **Autó & Alkatrész Arzenál**

### Weboldal:Szíjgyártó János Máté

### Mobil Alkalmazás: Magyar Szabolcs

## 1.1 Bevezetés

Tervezünk egy weboldalt és ennek megfelelően egy mobilos alkalmazást, hogy az eddig használt oldalakat helyettesítsük és bővítsük mert szerintünk azok nem eléggé kielégítők, pl: hasznaltauto.hu vagy bontoplaza.hu; ezek a weboldalak tökéletesek egy hétköznapi ember számára, de nincs olyan weboldal ahol ezen oldalak tulajdonságait egyesítené és mi erre szeretnénk egy alternatívát készíteni.

## 1.2 Programok funkciói

###### 1.2.1 Login-Register

A weboldalon lehetőség lesz felhasználói fiók regisztrálására, és regisztráció után bejelentkezésre, hogy az oldal összes funkcióját kihasználhassa a felhasználó, azaz hirdetés közzététele, hirdetések mentése, saját hirdetések törlése-módosítása, felhasználói fiók szerkesztése, törlése. A weboldal a felhasználókat bejelentkezés után köszönti. Felhasználió fiók hiányában csak keresni lehet a weboldalon, és a publikusan megosztott adatokat megtekinteni, hirdetéseknél a telefonszám, vagy egyébb elérhetőség ha az eladó kívánja megosztani azokat.

###### 1.2.3 Upload

A weboldalra lehetőség lesz majd hirdetéseket feltölteni, ehhez szüksége lesz a felhasználónak egy fiókra amit pár kattintással már el is készíthet, és utána már töltheti is fel az általa eladni kívánt termékeket, legyen az egy használt autó, vagy egy adott alkatrész.

###### 1.2.2 Search

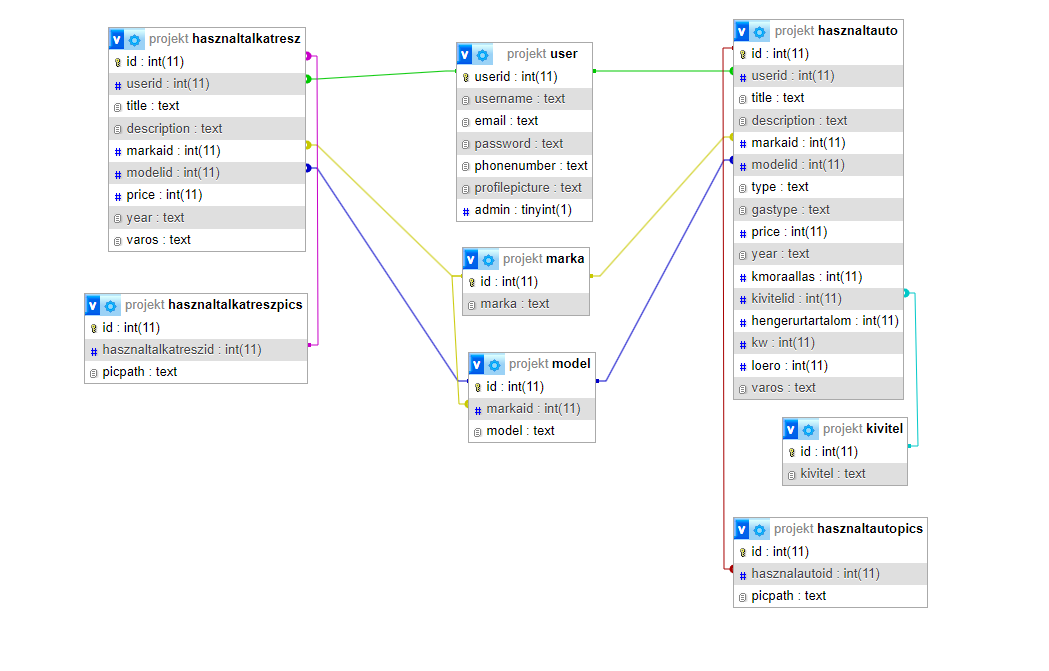
A weboldalon a hírdetéseket lehet majd szűrni, és közöttük keresi egyedi módon. A használtautó részlegen az előre meghatározott szűrőket lehet majd használni, viszont a használt alkatrész részlegen lehet majd keresővel is szűrni a hirdetések között.

1.3 Célközönség

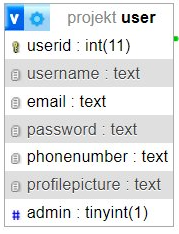
A weboldal az autós társadalmat célozza meg. Használtató kereséséhez és eladásához, használt alkatrészek autókhoz melyeket szintén a felhasználók adhatnak és vásárolhatnak meg. Fontos kiemelni hogy az oldal csak információt közvetít a felhasználók között mivel tranzakció lebonyolítására nem képes(hirdető felület).

2. Fejlesztői dokumentáció

2.1 Adatbázis

****

A weboldal egy MySQL típusú adatbázist fog használni, az alábbi adatok letárolására: felhasználói adat, autók alap adatai a további felhasználáshoz, hirdetések adatai.

**2.1.1** user tábla

Az adatbázis user táblájában vannak letárolva a felhasználók adatai, azaz a felhasználó id-ja, ami a weboldal számára fontos adat, felhasználónév, email cím, jelszó ami titkosítva van bcryptel backend oldalról, telefonszám, profilkép, és a felhasználó státusza, hogy admin vagy sima felhasználó.

###### 2.1.2 marka tábla

A marka tábla az egyik alap adat táblája lesz, ebből a táblából lehet majd kiválasztani, hogy a felhasználó milyen márkájú autót szeretne hirdetni, vagy keresni.

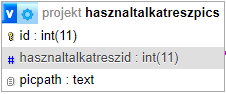
###### 2.1.3 model tábla

A model tábla kötődik a Márka táblához. Ez is egy alap adat, és a Márka kiválasztása után a felhasználónak ki kell választania az eladni kívánt, vagy keresendő autó modelljét.

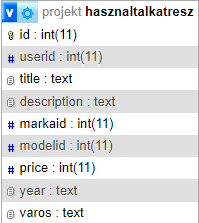
###### 2.1.4 kivitel tábla

###### A Kivitel tábla szintúgy egy alap adat lesz, ami által a felhasználó hirdetés feladása alatt, vagy keresés folyamán kitudja választani, hogy az eladni kívánt autó milyen kasztnival rendelkezik, pl: Lépcsőshátú (Szedán), Kombi, Kupé, vagy egyéb kivitelek.

###### 2.1.5 hasznaltalkatreszpics tábla

A hasznaltalkatreszpics tábla megadja az adott használtalkatrész hirdetésekhez feltöltött képeknek az útvonalát, hogy a szerver megtudja jeleníteni azokat a weboldalon.

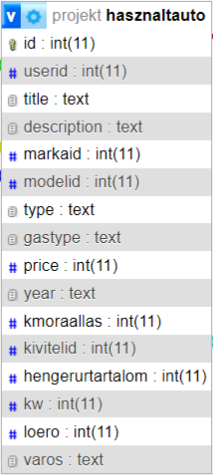
###### 2.1.6 hasznaltalkatresz tábla

A hasznaltalkatresz tábla a másik fő tábla ami eltárolja a feltöltött hirdetések adatait amit a felhasználók tesznek közzé.  
Az id a hirdetés azonosítója ami által az oldal tudja kezelni a hirdetéseket. A userid csatolja össze a felhasználót a hirdetéssel, ezzel tudjuk, hogy ki töltötte fel a hirdetést. A title az a hirdetés címe és a description a hirdetésben szereplő leírás. A markaid a modelid és a year meghatározza, hogy az adott alkatrész milyen autóból való. A price a meghirdetett termék árát jelzi. Végül a varos sor ami meghatározza, hogy a meghirdetett autót melyik városban árulják, és könnyebb keresést segít elő, hogy a potenciális vevő egy adott körzetben keressen alkatrészeket.

###### 2.1.7 hasznaltautopics tábla

A hasznaltautopics tábla megadja az adott használtalkatrész hirdetésekhez feltöltött képeknek az útvonalát, hogy a szerver megtudja jeleníteni azokat a weboldalon.

###### 2.1.8 hasznaltauto tábla

A Használtautó tábla lesz az egyik fő tábla, ahol le lesznek tárolva a weboldalon megjelenített hirdetések. Az ID a hirdetés azonosítója lesz, amit a weboldal fog felhasználni. A userid fogja összekötni a hasznaltauto táblát a user táblával, és a weboldal innen tudja meg, hogy melyik hirdetést ki töltötte fel, és a felhasználók is innen tudják majd meg. A title és description a hirdetések részét képzik, a cím és a leírás. A markaid és a modelid meghatározza az autó márkáját, és az autó márkán belüli modeljét. A type mező a hirdetett jármű egyedi típusát határozza meg, mint például egy gyártói tuning, vagy különböző felszereltségek. A gastype az üzemanyag fajtáját határozza meg. A price az a meghirdetett ár lesz, ezt már az eladó határozza meg. A year az előre meghatározott évek lesznek, amit a hirdetés feltöltésénél lehet majd kiválasztani. A kmoraallas az a hirdetett jármű km órájának az állása, ezt az adatot is a hirdető adja meg. A kivitelid a jármű kasztnijának a fajtája, előre meghatározott adat. A sorral, ami az autók motorjának a méretét határozza meg, erre rá lehet majd szűrni, mivel int típusú ezért kisebb vagy nagyobb szűrési lehetősége lesz. Bekerült még egy kw és egy loero sor is, ami az autó teljesítményét határozzák meg, ez ugyanúgy szürhető lesz mint a hengerültartalom sor. Végül a varos sor, ami meghatározza, hogy a meghirdetett autót melyik városban árulják, és könnyebb keresést segít elő, hogy a potenciális vevő egy adott körzetben keressen autókat.

2.2 Webboldal készítése

A weboldal Microsoft Windows 11 operációs rendszer alatt készült. A frontend html5, css3 és JavaScript (ES2022) felhasználásával, míg a backend elkészítése Node.JS 18.13.0 verziójával történt, a Microsoft Visual Studio Code kódszerkesztő segítségével.

2.3 app.js

Az index.js fájl a webáruház backend részének a főbeállító fájlja. Ez a fájl importál különböző modulokat, beállítja a middleware-eket, definiálja az útvonalakat és a logikájukat a HTTP metódusokhoz (GET, POST, PUT, DELETE).

2.4 regxlogin.js

**2.4.1 Regisztráció**

* A felhasználó beírja az adatokat a regisztrációs űrlapba.
* A kód ellenőrzi az adatokat (üres mezők, jelszóegyezés).
* Sikeres ellenőrzés esetén a kód elküldi az adatokat a szervernek.
* A szerver válaszol: sikeres regisztráció esetén üzenetet jelenít meg és átirányít a kezdőlapra, hiba esetén pedig hibaüzenetet jelenít meg.

**2.4.2 Bejelentkezés**

* A felhasználó beírja a bejelentkezési adatokat.
* A kód elküldi az adatokat a szervernek.
* A szerver válaszol: sikeres bejelentkezés esetén átirányít a felhasználó előző oldalára, hiba esetén pedig hibaüzenetet jelenít meg.

**2.4.3 Megjegyzések:**

* A kód Swal.fire()-t használ üzenetek megjelenítéséhez.
* A kód fetch()-et használ a szerverrel való kommunikációhoz.
* A kód hibákat kezel a try...catch blokkokkal.

2.5 home.js

**2.5.1 Változók**

* logineduserID: Üres string, a bejelentkezett felhasználó ID-ját tárolja.
* loginedusername: Üres string, a bejelentkezett felhasználó nevét tárolja.
* loginedemail: Üres string, a bejelentkezett felhasználó email címet tárolja.

**2.5.2 Adatlekérés a szervertől**

* A kód a fetch függvénnyel lekéri a /get-session-data végpontot a szervertől.
* A válasz érkezésekor a .then metódus segítségével a válasz szöveges tartalmát JSON formátumba alakítja (response.json()).
* Ezután ismét a .then metódust használja a konvertált JSON adat kezelésére.

**2.5.3 Bejelentkezési állapot vizsgálata**

* A JSON adatban ellenőrzi, hogy létezik-e user kulcs.
* Ha létezik user kulcs az adatban:
  + A kód a user objektum alatti userid, username, email tulajdonságokból kinyeri az értékeket és beállítja a megfelelő változókat (logineduserID, loginedusername, loginedemail).
* Ha nincs user kulcs az adatban:
  + A bejelentkezési változókat undefined értékre állítja.

2.6 bonto.js

**2.6.1 Bejelentkezett felhasználó ellenőrzése (az oldal betöltésekor)**

* A kód először egy eseményfigyelőt ad hozzá az ablak objektum "load" eseményéhez. Ez azt jelenti, hogy az ezen a funkción belüli kód akkor kerül végrehajtásra, amikor a teljes weboldal betöltése befejeződik.
* Az eseményfigyelőn belül meghívja a getpost függvényt.
* A getpost függvény valószínűleg a /get-session-data végponton keresztül a szervertől lekéri az aktuális felhasználói munkamenet adatait.
* A válaszadatok (felhasználó objektum) alapján feltölti a logineduserID, loginedusername, loginedemail és admin változókat a felhasználói információk tárolásához.
* Ezután meghívja a setbtnreg és setbtn függvényeket, hogy a weboldalon lévő gombok frissítésére szolgáljon, attól függően, hogy a felhasználó be van-e jelentkezve.

**2.6.2 Hirdetések lekérdezése és megjelenítése**

* A getpost függvény a hirdetésekről is lekéri az adatokat a szervertől a /get-alkatresz-ad végponton keresztül.
* Végigmegy a lekérdezett hirdetéseken, és minden hirdetéshez HTML tartalmat épít fel, amely részleteket tartalmaz, mint a cím, a leírás, az ár és a helyszín.
* Ellenőrzi az admin változót, hogy meghatározza, hogy további funkciókat, például törlés gombot kell-e megjeleníteni minden hirdetéshez.
* Végül a generált HTML-lel frissíti a weboldal egy szakaszának tartalmát (valószínűleg a "card-frame" azonosítóval).

**2.6.3 Márka- és modellválasztás**

Vannak loadbrand és loadModelSelect függvények, amelyek valószínűleg a szervertől lekérik az autókkal kapcsolatos márkákkal és modellekkel kapcsolatos adatokat, és kitöltik a weboldalon lévő legördülő menüket.

**2.6.4 Új hirdetés feltöltése**

* A kód kezeli az új hirdetés létrehozásához használt űrlap elküldését.
* Ellenőrzi, hogy a feltöltött fájlok képek-e, és érvényesíti a feltöltött fájlok számát.
* Ezután elküldi az űrlap adatait (beleértve a potenciálisan több képet is) a szervernek a /upload-alkatresz/ végponton keresztül (valószínűleg a hirdetési adatok tárolására).
* A szerverválasz alapján siker- vagy hibaüzenetet jelenít meg a felhasználónak.

**2.6.5  Hirdetés törlése**

* A delbtn függvényt valószínűleg akkor hívják meg, amikor a felhasználó rákattint egy hirdetés törlési gombjára.
* A függvény megidézi a delPost függvényt, átadva a törölni kívánt hirdetés azonosítóját.
* A delPost függvény lekéri annak a felhasználónak az azonosítóját, aki feltöltötte a hirdetést, majd megerősítést kér a felhasználótól, mielőtt törlési kérelmet küldene a szervernek a /delbonto/ végpont használatával.
* A szerver válaszától függően siker- vagy hibaüzenetet jelenít meg a felhasználónak

2.7 hasznalt.js

**2.7.1 Fő funkciók:**

* Hirdetések betöltése:   
  A weboldal betöltésekor a rendszer betölti a legújabb hirdetéseket a szerverről.
* Felhasználói profil:   
  Bejelentkezett felhasználók láthatják profiljukat, és szerkeszthetik hirdetéseiket.
* Hirdetésfeladás:   
  A felhasználók új hirdetéseket adhatnak fel képekkel és részletes leírással.
* Hirdetéskeresés:   
  A felhasználók típus, márka, modell és ár alapján kereshetnek hirdetéseket.
* Hirdetés szűrése:   
  A felhasználók szűrhetik a hirdetéseket további kritériumok alapján, mint például a futásteljesítmény, a gyártási év és a szín.
* Hirdetés törlése:   
  A hirdetés tulajdonosa bármikor törölheti a hirdetését.
* Részletes hirdetés:   
  A felhasználók megnyithatják a hirdetéseket, hogy megtekintsék a részletes leírást, a képeket és a térképet a hirdetés helyéről.

**2.7.2 Egyéb funkciók:**

* Bejelentkezés/Regisztráció
* Profil szerkesztése
* Jelszóváltoztatás
* Kapcsolatfelvétel

**2.7.3 Műszaki megvalósítás:**

A weboldal JavaScript, HTML és CSS nyelven íródott. Aszinkron kéréseket használ a szerverrel való kommunikációhoz a hirdetések betöltéséhez, a profil adatok lekéréséhez és a hirdetések feladásához/törléséhez.

**2.7.4 Összefoglalva:**

Ez a weboldal egy átfogó platformot nyújt a használtautó-vásárlók és -eladók számára. A felhasználók könnyedén böngészhetnek hirdetéseket, kereshetnek a kívánt autók alapján, feladhatják saját hirdetéseiket, és kapcsolatba léphetnek az eladókkal.

2.8 reszletes.js

**2.8.1 A GetPic funkció**

**2.8.1.1 Képadatok lekérése**

* A fetch API-t használja a /get-pic-to-reszletes/${id} végpontról történő aszinkron kéréshez, ahol az id a termék azonosítója.
* A szerverről JSON formátumban kapott képadatokat a poszt változóban tárolja.

**2.8.1.2 Slider HTML tartalom felépítése:**

* Ciklusonként végigmegy a képadatokon, és mindegyikhez:
  + Létrehoz egy "mySlides" osztályú elemet a nagy képhez, sorszámmal.
  + Létrehoz egy "demo cursor" osztályú elemet a miniatűr képhez.
  + Beállítja a képek forrását a szerverről.
  + Eseménykezelőket ad hozzá a navigációhoz (előző/következő, miniatűrök kattintása).

**2.8.1.3 A slider megjelenítése:**

* A beépített HTML-t az "imgslider" elem belső tartalmába beszúrja.
* Ha hiba van, a konzolra írja ki.

**2.8.1.4 Megjegyzések**

* A funkció aszinkron, azaz a szerverről származó adatokra várva fut.
* A poszt változó a szerverről kapott képadatokat tárolja.
* A postHTML változó a dinamikusan generált HTML tartalmat építi fel.
* A try...catch blokk a hibakezelést biztosítja.

### 1.4 Mobilos alkalmazás

A mobilos alkalmazás készítése még folyamatban van, eddig a frontend része egész jól áll, de a backend részével nem tudtunk még sokat foglalkozni. Jövőbeli tervek között van, hogy össze legyen kötve a teljes adatbázissal, és a weboldallal is, és emellett a backend rész kifejlesztése, és a teljesen mobilos alkalmazás továbbfejlesztése is.

Tervben van egy könnyen kezelhető felhasználói felület, egy felhasználó barát hirdetés feltöltés és egy részletes keresés is.

# 2.1 Összegzés

Összegzésként szeretnénk megemlíteni, hogy bár az elején nagyon lassan haladtunk és a határidőkkel is nagyon csúnyán megcsúsztunk emellett az időbeosztással is, de próbáltuk a maximumot kihozni belőle, a saját tudásunk és oktatóink segítségével is, ami miatt nagy köszönetet is szeretnénk mondani nekik.

Weboldal: Szíjgyártó János Máté  
Mobil alkalmazás: Magyar Szabolcs  
Adatbázis: Szíjgyártó János Máté  
Dokumentáció: Magyar Szabolcs

**Kisegítő tanárok : Szántó Benjámin, Kovács István Zoltán**