**Dokumentáció**

**Téma: Használt hardver árusító webalkalmazás fejlesztése**

**Selyem Péter Ferenc (WOKEUE) - Radics Bence Zoltán (BTJ2TO)  
Témavezető: Nagy Zsuzsanna**

**1. Bevezetés**

A projekt labor feladatunk célja egy online platform létrehozása, amely lehetőséget biztosít használt számítástechnikai hardverek értékesítésére és vásárlására. Az alkalmazás hasonlóan fog működni, mint a már ismert hardverapro.hu, de egyedi funkciókkal bővítve, felhasználóbarát felülettel és modern technológiai megoldásokkal.

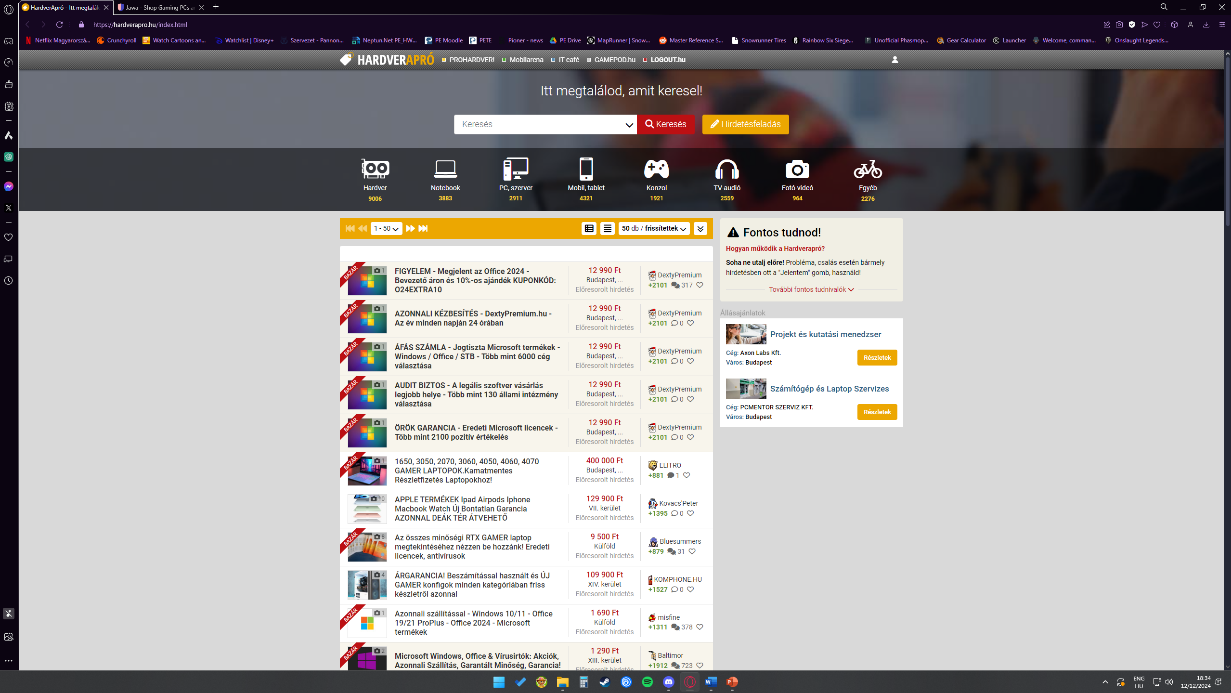
Ez a dokumentum specifikációkat, technikai követelményeket és egy Proof of Concept (POC) tervet tartalmaz a projekt megvalósításához.

**1.1. Hasonló alkalmazások**

Több magyar nyelvű, illetve magyar fejlesztésű használt hardver árusító oldal is megtalálható az interneten, mint például a HardverApro, Hardver Bazár, Gamer Ellátó, de még a Jófogás oldalán is lehetőség van használt hardver eladására. Ezeknek az oldalaknak a célja hirdetések összegyűjtése, ami megkönnyíti a használt eszközök eladását, azon személyeknek, akik megválnának régi, már nem használt hardvereiktől. Mindemellett, akik olcsóbban keresnek hardvert, vállalják, hogy inkább olcsóbb, használt árut vesznek az drágább, de új, bontatlan helyett, azoknak egy remek platformot nyújt az elérhetőségek böngészéséhez. A mi programunkhoz ezek közül legfőképpen a HardverApro oldalát vettük alapul.

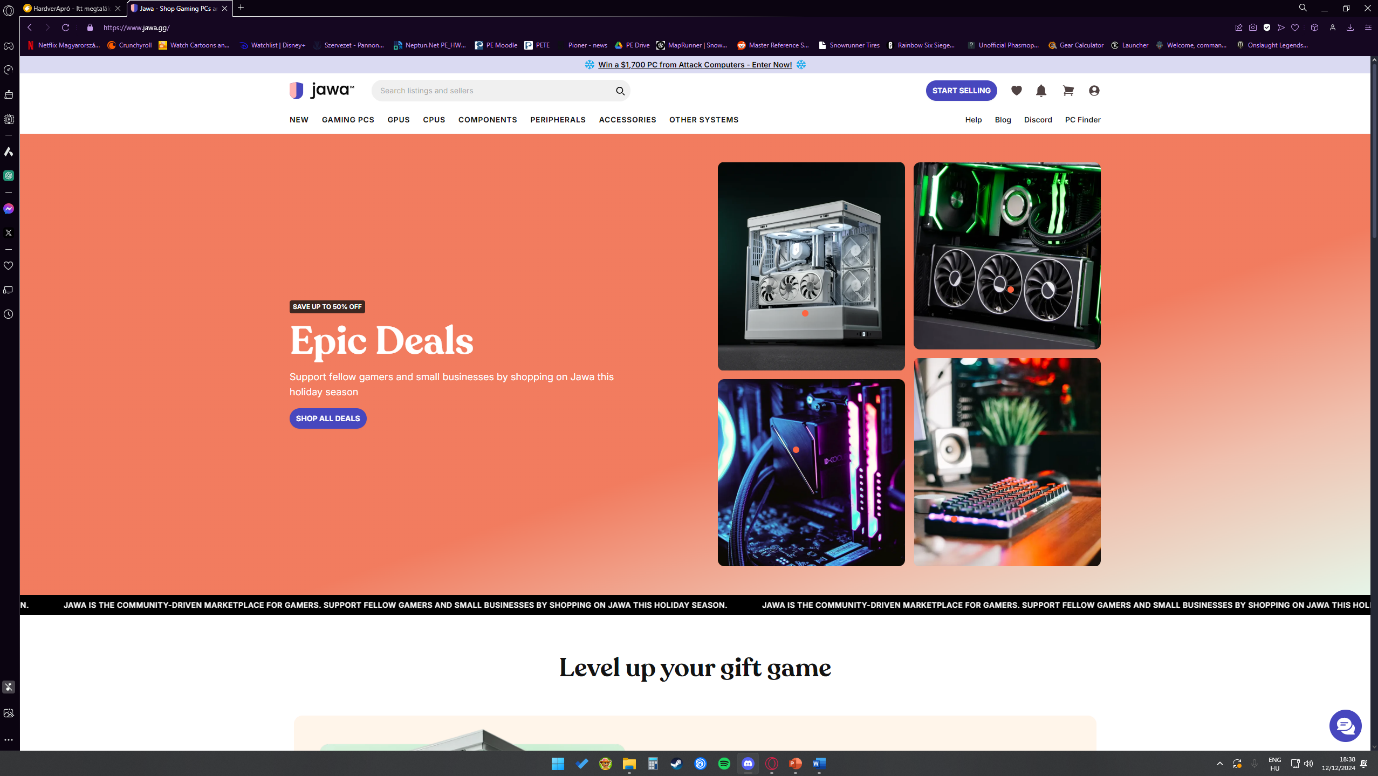
**1.1.1. HardverApro ismertetése**

A HardverApro is egy használt hardver árusító oldal. Az oldal kezelőfelülete nagyon egyszerű, letisztult, mégis minden fontos és elengedhetetlen funkció megtalálható rajta, ami egy ilyen program használatához kell. Be lehet regisztrálni az oldalra, regisztráció után bejelentkezni, a saját adatokat megtekinteni. A főoldalon láthatóak a legfrissebb, legújabb kiemelt hirdetések, ezek között böngészhetünk. Ha valami specifikusabbat keresnénk, akkor lehet szűrni kategória szerint, mint például hardver, notebook, PC, stb. Ezeken belül alkategóriák segítik a könnyebb keresést. Ha biztosan tudjuk milyen terméket keresünk, akkor pedig rendelkezésre áll az oldal tetején egy keresőmező, ahol kulcsszavakat megadva, vagy akár pontos termékneveket is beírva kereshetünk az általunk megtekinteni kívánt termékekre. A mi oldalunk is hasonló funkciókkal fog rendelkezni, szeretnénk egy hasonlóan jól felépített és minél több fontos funkcióval ellátott weboldalt létrehozni.



ábra 1. A hardverapro.hu főoldala

**1.1.2. Külföldi példa**

Külföldön is rengeteg hasonló program megtalálható, mi ezek közül a jawa.gg weboldalát vettük még alapul a HardverApro mellett. A weboldal egy nagyon esztétikusan, fejletten kidolgozott felülettel rendelkezik. Minden fontos funkció megtalálható rajta, ami egy ilyen program használatához szükséges, hasonlóan, mint a HardverApro esetében. Ami kiemeli, az a fejlettebb, modernebb felhasználói felület, illetve az internacionális támogatása.

ábra 2. A jawa.gg weboldala

**2. Funkcionális követelmények**

**2.1. Regisztráció és bejelentkezés**

* **Leírás**: A felhasználók regisztrálhatnak az oldalon egy felhasználónév és jelszó megadásával. Emellett megadják a teljes nevüket is, könnyebb, személyesebb azonosításhoz az oldal használata közben.
* **Követelmények**:
  + A felhasználónév alapú regisztrációhoz szükséges egy egyedi felhasználónév.

**2.2. Termékek böngészése**

* **Leírás**: A felhasználók böngészhetnek a meghirdetett termékek között.
* **Követelmények**:
  + Kategória szerinti szűrési lehetőség (pl. CPU, GPU, alaplap, stb.).
  + Ár szerinti szűrés és rendezés.
  + Termék állapot szerinti szűrés (új, használt, felújított).
  + Termékek képekkel és részletes leírással jelenjenek meg.

**2.3. Termékek feltöltése**

* **Leírás**: A regisztrált felhasználók feltölthetnek eladásra szánt termékeket.
* **Követelmények**:
  + Kötelező mezők: Termék neve, helye, kategóriája, állapota, ára, képek feltöltése, leírás.

**2.4. Adminisztrációs felület**

* **Leírás**: Az adminisztrátorok számára elérhető felület a felhasználók és hirdetések kezelésére.
* **Követelmények**:
  + Hirdetések felülvizsgálata és moderálása.
  + Felhasználói fiókok zárolása vagy inaktiválása szabályszegés esetén.

**3. Nem funkcionális követelmények**

**3.1. Teljesítmény**

* Az oldalnak gyors válaszidőt kell biztosítania, a termékek listázása és szűrése kevesebb, mint 1 másodperc alatt történjen.
* Az adatbázis 100,000 termék kezelésére legyen optimalizálva.

**3.2. Biztonság**

* HTTPS titkosított kapcsolat kötelező.
* Felhasználói adatok (beleértve a jelszavakat) titkosítása szükséges.

**4. Felhasznált technológiák**

A programunk elkészítéséhez korszerű technológiákat választottunk, amivel a fejlesztés menete megkönnyíthető, illetve hatékonnyá tehető. A Microsoft Visual Studio 2022 és Visual Studio Code fejlesztőkörnyezetei mellett döntöttünk, korábban megszerzett ismeretek és különböző ajánlások alapján. A programozási nyelvek, amikkel dolgozunk a frontend fejlesztéséhez a HTML, CSS, illetve JavaScript. Ezek mellett a Bootstrap nevű eszközkészlet segítségét is felhasználjuk a felhasználói felület átláthatóbbá és vizuálisan szebbé tételének érdekében. A backend programozásához a C# nyelvet választottuk, a .NET keretrendszerrel kiegészítve, illetve az SQLite programot is segítségül vettük az adatbázis struktúra kialakításához, és az adatbázis könnyebb kezeléséhez. Ezek mellett a GitHub nevű programot is felhasználtuk a könnyebb verziókezelés és jobb átláthatóság érdekében.

**4.1. Frontend**

Mint korábban említettük, a frontend megvalósításához a Microsoft Visual Studio Code fejlesztőkörnyezetét használtuk fel, a választott programozási nyelveink pedig a HTML, CSS, illetve JavaScript. A továbbiakban szeretnénk megindokolni a választásainkat, illetve kicsit bemutatni ezt a programot és az említett programozási nyelveket.

Kezdve a Visual Studio Code programmal, ez egy ingyenesen elérhető, manapság nagy népszerűségnek örvendő forráskódszerkesztő program. Előnyei, melyek számunkra is fontosak, hogy kis méretű, kevés tárhelyet foglal, ezáltal elfér több rendszeren is, kisebb tárhelyű laptopon is tudunk vele dolgozni, nem csak az otthoni számítógépeinken. Ebből adódóan alacsony a gépigénye, gyorsan indul és gördülékenyen fut szinte bármilyen felszereltségű számítógépen. Több ezer bővítmény elérhető hozzá, ami gyorsítja és egyszerűsíti a programozási feladatokat, ezzel megkönnyítve a fejlesztést, csökkentve a hibák számát is. Ehhez kapcsolódik az is, hogy tartalmazza az IntelliSense automatikus kódkitöltést, tartalmaz hibakeresést, Git integrációt is. Mellékes előny, hogy cross-platform, azaz több platformra is elérhető, mint például macOS vagy Linux rendszerekre is.

Programnyelvek nélkül egyértelműen nem működne a programozás, ezért fontos volt kiválasztanunk, hogy melyik nyelvet alkalmazva szeretnénk megírni a programunkat és melyik lenne legalkalmasabb és legoptimálisabb a céljaink eléréséhez. Emiatt esett a választásunk a HTML, CSS, illetve JavaScript nyelvek mellett. A HTML leíró nyelv a legelterjedtebb webes programozási nyelv, gyakorlatilag minden webfejlesztési projektnek ez az alapja. Ezt használjuk a weboldalaink elkészítésére, ezzel készült az oldalak alapja, funkcióiknak szerkezete. Számunkra célszerű volt a választás emiatt, illetve azokból az okokból is, hogy ez egy egyszerű, könnyen tanulható és értelmezhető programozási nyelv, mindemellett egyetemi tanulmányaink során tanórák keretein belül is volt lehetőségünk ezzel a nyelvel megismerkedni, illetve az alapjait elsajátítani.

A CSS nyelvet is hasonló okból választottuk, amellett, hogy tökéletesen kiegészíti a HTML programnyelvet és az azt felhasználva elkészített kódot remekül letisztítja, átláthatóbbá teszi. A CSS lehetővé teszi a weblapok bővebb dizájnolását, a stilisztikai elemek külön fájlban tárolását, finomhangolását. Emellett segít egy reszponzív dizájn létrehozásában is. Ezt egészíti ki a Bootstrap is, ami egy nyílt forráskódú CSS keretrendszer, ami kisegít a webfejlesztés gyorsításával és tovább egyszerűsítésével. Az implementációja nagyon egyszerű, szinte azonnal megy, csupán egy link vagy letöltés segítségével már hozzá is adható a weboldalhoz. Ez a keretrendszer lehetővé teszi, hogy a jövőben akár mobiltelefonos, illetve táblagép kijelzőkre is lehessen optimalizálni a weboldalainkat. Tartalmaz továbbá előre definiált alap stílusokat, amiket személyre szabással lehet saját igényekre alakítani, vagy akár alapértelmezetten azonnal felhasználni. Ezzel megkönnyíti, hogy a weboldalak egységesen, elemeik konzisztens stílusban jelenjenek meg. Dinamikus elemeket is nagyon egyszerűen és gyorsan lehet integrálni Bootstrapben, JavaScript vagy jQuery segítségével. Ezek mellett széles, nagy közösségi támogatással rendelkezik, illetve részletes dokumentáció rejlik mögötte. Legvégül, de nem utolsó sorban minden modern böngésző támogatja, így kompatibilitási problémák nem léphetnek fel használata közben.

A funkciók működését és a kapcsolatot a backend és frontend között JavaScript programkódok biztosítják. Emellett is azonos okból került a választás, mint a HTML és CSS mellet, miszerint korábbi alapismereteink vannak ebben a nyelvben az egyetemi éveink alatt látogatott webprogramozási tanórák révén. Ezt a programnyelvet felhasználva könnyen megvalósíthattuk a funkciók működését, illetve az adatok dinamikus kezelését, mindezzel lehetővé téve a program gördülékeny működését.

**4.2. Backend**

A backend fejlesztéséhez, mint korábban írtuk, a Microsoft által fejlesztett, széles körben elterjedt Visual Studio 2022 fejlesztőkörnyezetet választottuk. Ez a program különösen alkalmas akár nagyobb backend projektek fejlesztésére is, illetve elterjedtsége miatt sok segítség, tipp, oktatóanyag található hozzá az interneten. Emellett tanóráinkon is találkozhattunk már a fejlesztőkörnyezettel, így ezt volt a legkönnyebb alkalmazni.

A Visual Studio 2022 előnyei, hogy tartalmaz hasznos, kódolást nagymértékben segítő eszközöket, mint az IntelliCode, ami egy gépi tanuláson alapuló kódjavaslatokat adó eszköz, továbbá tartalmaz fejlett, modern hibakeresőt és tesztelési eszközöket is. Beépítetten tartalmaz .NET, illetve .NET Core keretrendszereket, amelyek a C# alapját képzik és nagymértékben segítik a fejlesztői munkát. Külön kiemelendő, hogy széleskörű integrációval rendelkezik, rengeteg plugin, kiegészítő elérhető a Visual Studio fejlesztőkörnyezetéhez, illetve Azure és GitHub integrációkat is tartalmaz.

Az általunk felhasznált programozási nyelv a C#, ezzel kevesebb tapasztalatunk van, Selyem Péternek több, ellenben sokan ajánlották és mivel egyéb nyelvekkel is kevés a tapasztalatunk, ezért esett emellett a választás a sok pozitívuma miatt. A C# egy Microsoft által fejlesztett objektumorientált programozási nyelv, amely a .NET keretrendszer alapja. Nagy előnye például a hatékony erőforrás- és memóriakezelés, amelyek meggyorsítják az elkészült alkalmazás működését, illetve lehetővé teszik a gazdaságos futtatását. A .NET Core-nak köszönhetően platformfüggetlen a nyelv, ezáltal több platformon is elérhető, több platformon nyújt támogatást. Webes backend, asztali alkalmazások és mobilappok fejlesztésére kiemelten ajánlják, emiatt számunkra egy egyértelmű választás volt.

Mint említettük a .NET keretrendszer alapja a C# és ezt többek között ezért is felhasználtuk a programunk elkészítésében. A .NET tartalmazza az ASP.NET Core rendszert, amely egy modern, gyors és rugalmas keretrendszer kifejezetten webes alkalmazások és API-k fejlesztésére. Programunk mivoltából ennek a rendszernek a használata szinte elengedhetetlen, ezzel a munkánk nagymértékben gyorsul és egyszerűsödik. A .NET továbbá tartalmaz magas szintű beépített biztonságot, az adatvédelem és hitelesítés támogatására. Ezek mellett tartalmaz előre elkészített funkciókat, könyvtárakat, amiket felhasználva tovább gyorsul és könnyebbedik a fejlesztési folyamat.

Mindezeket egybe véve kialakult az általunk használt technológiák rendszere a backend fejlesztés hatékonyságának és céljaink eredményes elérésének érdekében. A rendszer előnyei a gyors futási idő, alacsony erőforrás igény, letisztult, egységes, átlátható kódbázis, illetve a könnyű integrálhatósága a frontend megoldásoknak és az általunk választott SQLite adatbázis-kezelőnek.

**4.3. Kisegítő alkalmazások**

Kisegítő programoknak az SQLite adatbáziskezelő programot, illetve a GitHub nevű online eszközt használtuk. Ezekkel a programokkal átláthatóbbá és könnyebbé válik a fejlesztési folyamat.

Az SQLite program egy SQL alapú adatbáziskezelő rendszer, ami megfelel kisebb projektekhez, illetve helyi adatkezeléshez. Előnyei, hogy egyszerűen és gyorsan használható, hiszen nincs szükség külön adatbázis-szerverre, illetve az adatokat egy db fájlban tárolja, így azok könnyen mozgathatóak és könnyen megtalálhatóak. Emellett kompatibilis szinte minden programozási nyelvvel, ezzel biztosítva, hogy gördülékenyen fogunk tudni vele dolgozni, működni fog az alkalmazásunkkal is. A teljes funkcionalitás sem maradt ki belőle, támogatja az összes SQL szabványt, tranzakciókat, illetve az indexeket is.

A GitHub egy online platform, ahol a kódjainkat tárolhatjuk, több eszközről elérhetjük. A program lehetővé teszi a verziókövetést, így egyszerűbben elérhetőek a változtatások, amiket akár vissza is lehet vonni, vagy akár csak megtekinteni egy korábbi megoldást. Lehetővé teszi az együttműködést is, párhuzamosan dolgozhatnak a csapat tagjai. Ez a projekt csapatmunka, éppen ezért szinte elengedhetetlen a GitHub használata. Segítségével bármikor, bárhol elérhetjük a kódot, láthatjuk ki mit változtatott és mikor.

**5. Proof of Concept (PoC)**

A PoC célja, hogy egy minimálisan működő prototípust készítsünk a platform legfontosabb funkcióival. A következő funkciókat tartalmazza:

1. **Regisztráció és bejelentkezés**: Felhasználónév alapú regisztráció.
2. **Termék böngészés**: Alap termékkeresési és szűrési funkciók.
3. **Termékfeltöltés**: Egy egyszerű felület, ahol a felhasználók feltölthetnek egy terméket kép és leírás megadásával.

**5.1. Példa**

**Frontend:**

* A felhasználók a főoldalon kereshetnek termékek között.
* Egy felhasználó regisztrálhat, bejelentkezhet, majd feltölthet egy terméket a **„Termék feltöltése”** oldalon, ahol egy egyszerű űrlapot kell kitölteni.

**Backend:**

* Az API útvonalak segítségével biztosítja a termékek adatainak feltöltését, lekérdezését.

**6. Tesztelés és hibakezelés**

* **Unit tesztek**: Backend funkciók és API végpontok tesztelése.
* **Integrációs tesztek**: Felhasználói műveletek teljes folyamatának tesztelése (pl. termék feltöltése, majd annak megtekintése).

**6.1. Unit tesztek**

* **Cél**: A rendszer egyes komponenseinek tesztelése külön-külön annak érdekében, hogy megbizonyosodjunk arról, hogy a funkcionalitás helyesen működik.
* **Példa**:
  + Tesztelni kell a felhasználói bejelentkezési folyamatot, hogy a helyes felhasználónév és jelszó kombináció sikeres hitelesítést eredményezzen.
  + A termékfeltöltési funkció unit tesztje, amely biztosítja, hogy a kötelező mezők ki legyenek töltve, és az adatok megfelelően elérjék az adatbázist.

**6.2. Integrációs tesztek**

* **Cél**: Biztosítani, hogy a rendszer különböző részei együttműködjenek a várt módon. Például a frontend és a backend közötti adatáramlás zökkenőmentes legyen.
* **Példa**:
  + A felhasználó regisztrálása, bejelentkezése és egy termék sikeres feltöltése.
  + A termék szűrésének tesztelése kategóriák és ár alapján, hogy meggyőződjünk a szűrési algoritmus helyességéről.

**6.3. Hibakezelés**

* **Cél**: A hibák hatékony kezelése annak érdekében, hogy a felhasználók felé megfelelő hibaüzenetek jelenjenek meg, és a fejlesztők gyorsan azonosíthassák a problémákat.
* **Példa**:
  + Hibás adatbevitel esetén a rendszer azonnal figyelmeztető üzenetet küld a felhasználónak (pl. hiányzó mező vagy érvénytelen adat).
  + Az adatbázis-kapcsolati hibák vagy egyéb szerveroldali problémák esetén a rendszer egy általános hibaüzenetet küld a felhasználónak (pl. „Jelenleg nem elérhető, próbálja újra később.”).

**7. Felhasználói élmény (UX) és felület (UI)**

**7.1. Felhasználói felület tervezése**

* **Cél**: Az alkalmazásnak egy egyszerű, könnyen navigálható és vizuálisan vonzó felületet kell biztosítania.
* **Elvárások**:
  + Tiszta, modern dizájn.
  + A főoldalon jól elkülönített szűrési lehetőségek és gyors elérhetőség a különböző termékkategóriákhoz.
  + A felhasználói profil könnyen kezelhető, ahol az eladó egyszerűen nyomon követheti a feltöltött termékeket.
  + A bejelentkezési és regisztrációs oldalnak jól kell működnie, könnyen navigálhatónak kell lennie, hiszen egy új felhasználó ezt látja először, még a program igénybevétele előtt.

**7.2. Reszponzív dizájn**

* **Cél**: Az oldalnak különböző képernyőméreteken (mobil, tablet, asztali gép) is jól használhatónak kell lennie.
* **Példa**:
  + Kisebb kijelzőkön a terméklisták és szűrési lehetőségek harmonikusan illeszkednek, míg a nagyobb kijelzőkön több adat látható egyszerre.

**8. Projekt menedzsment**

**8.1. Fejlesztési szakaszok**

A fejlesztési folyamat három fő szakaszra osztható:

1. **Tervezés**: Ebben a szakaszban a követelmények pontos meghatározása és az architektúra tervezése történik meg.
   * Szükséges idő: 2 hét.
2. **Fejlesztés**: A platform funkcióinak lépésről lépésre történő fejlesztése, kezdve az alapvető funkcióktól (regisztráció, termékfeltöltés) a bonyolultabb rendszerekig (adminisztráció).
   * Szükséges idő: 8-10 hét.

**8.2. Projekt ütemezése**

A projekt becsült időtartama 10-12 hét, figyelembe véve a tervezést, fejlesztést, tesztelést.

**8.3. Erőforrások**

* **Technikai eszközök**: Laptopok, számítógépek, fejlesztői szoftverek (Visual Studio Code, Visual Studio 2022).

**9. Jövőbeli fejlesztési lehetőségek**

**9.1. Mobilalkalmazás fejlesztése**

Az oldalon elérhető funkciók mobilalkalmazás formájában történő megvalósítása, amely Android és iOS platformon is elérhető lenne. Az alkalmazás push értesítéseket biztosíthat a felhasználók számára, ezzel fokozva a felhasználói élményt. Emellett előnye lenne, hogy a felhasználók bárhonnan elérhetik az oldalt, bármikor megtekinthetik követett hirdetéseiket, nem csak otthonuk kényelméből.

**9.2. Aukciós rendszer bevezetése**

Az eladók számára lehetőséget biztosítani arra, hogy aukciós formában értékesítsék termékeiket, ahol a felhasználók licitálhatnak a termékekre. Ez megkönnyítené nehezen beárazható termékek eladását, vagy akár gyorsabb eladási módot nyújtana.

**9.3. Fizetési rendszerek integrálása**

Olyan online fizetési rendszerek, mint például a PayPal, integrálása, amelyek segítségével a vásárlók közvetlenül a platformon keresztül fizethetnek az eladóknak, ezzel is könnyítve a vásárlási folyamatot és biztonságosabbá téve azt.

**9.4. Gépi tanulás alapú ajánlórendszer**

Egy gépi tanulás alapú algoritmus kidolgozása, amely a felhasználói viselkedés alapján személyre szabott termékajánlásokat kínál, növelve ezzel az eladások esélyét és a felhasználói elégedettséget.

**10. Adatbázis felépítése**

**11. Frontend dokumentáció**

**11.1. Bevezetés**

A program frontendjének elkészítésénél felállítottunk pár alapvető célt, illetve szempontot. Elhatároztuk, hogy egy könnyen navigálható felületet szeretnénk létrehozni, ami támogatja az összes funkciót, amit implementálni tervezünk. Céljaink eléréséhez több weblapot is készítettünk, a program használata közben ezek között lehet váltogatni, hogy mindig azt lássuk, amit éppen használunk.

**11.2. Frontend felépítése, mappa struktúra**

Alább szemléletesen látható a mappaszerkezet.

TechTuri\_FrontEnd  
|-index.html  
|-item.html  
|-login.html  
|-postItem.html  
|-profile.html  
|-profileEdit.html  
|-register.html  
|+Images  
| -favicon.ico  
| -favicon.png  
| -LoginRegister.jpg  
| -MainBG.jpg  
| -ProfileBG.jpeg  
|+Scripts  
| -api.js  
| -auth.js  
| -item.js  
| -main.js  
| -postItem.js  
| -profile.js  
| -webSocket.js  
|+Styles  
| -login\_reg\_style.css  
| -style.css

Amint látható, a frontendünk hét weblapból, kettő CSS stílus fájlból, hét script fájlból és öt képből áll. A képek a weblapok hátteréhez lettek felhasználva, illetve a faviconhoz a weblapon. A beszédes fájlnevekkel és az átlátható mappaszerkezettel könnyebb volt a munkánk folyamata, egyszerűbben megoldható volt a mentések közötti keresés, mindenki megtalálta és megismerte az általa keresett fájlt.

**11.3. Oldalak leírása**

Mint fentebb látható volt, a programunk hét weblappal dolgozik. Ezeket logikai sorrendben ismertetjük.

A **login.html** oldal felelős a bejelentkezési felület megjelenítéséért. Az oldalnak van egy témához illő háttér, amin középen helyezkedik el a bejelentkezési felület. Két beviteli mezővel rendelkezik, ahol a felhasználó megadja a felhasználónevét, illetve jelszavát. A mezők alatt található „Bejelentkezés” gomb segítségével tud bejelentkezni. Ha még nem regisztrált a felhasználó, akkor a felület alján található link segítségével ugorhat át a regisztrációs felületre. Az oldal animációkkal, illetve változó elemekkel is el lett látva, bár ez mindegyik oldalunkra igaz.

A **register.html** weblapon található a regisztrációs felületünk. Dizájnban szinte teljesen egyezik a bejelentkezési felülettel, annyi különbséggel, hogy itt 4 beviteli mező található meg. A felhasználónév és jelszó pároson kívül a regisztrálni kívánó felhasználónak meg kell adnia teljes nevét, illetve még egyszer a jelszavát. Az utóbbi funkció olyan szempontból fontos, hogy megelőzzük a véletlen el írt jelszavak mentését, hiszen elég kicsi rá az esély, hogy a felhasználó egymás után kétszer is ugyanúgy elírja a megadni kívánt jelszavát.

Az **index.html** a főoldalunk, ezen az oldalon találhatóak meg az alap funkciók. Az oldalnak egy szintén témába vágó, de a bejelentkezési és regisztrációs felületekétől eltérő háttere van, a kezelőfelület pedig a háttér előtt szintén középen helyezkedik el. A felület tetején látható a bejelentkezett felhasználó neve, alatta pedig a navigáló linkek: egy link, ami a profil adataink megtekintésére, ellenőrzésére szolgáló lapra irányít, egy másik, ami segítségével új hirdetést adhatunk fel az erre a célra készített oldalon. A harmadik link pedig egy kijelentkezési link, erre kattintva az oldal kiléptet minket és visszanavigál a bejelentkezési felületre. Az oldalon még megtalálható minden feladott hirdetés, ezek között tudnak a felhasználók böngészni, nézelődni.

Az **item.html** oldal célja egy hirdetés minden adatának megjelenítése részletesebben, felépítése hasonló mind a főoldalnak, csak kevesebb funkcióval a könnyebb átláthatóság érdekében.

A **postItem.html** weblapon tudják a felhasználók a hirdetéseiket feladni. A főoldalról link segítségével tudnak ide navigálni. Az oldal alapja azonos a főoldaléval, azonban a felhasználói felület különböző. Az oldal tetején csak egy link van, ami segítségével a felhasználók visszatérhetnek a főoldalra. A link alatt található maga az űrlap, ami kitöltésével lehet feltölteni új hirdetést. Az űrlap hét kötelező mezőből áll. Az elején három kitöltendő szöveges mező található, ide kell megadni a termék nevét, árát, illetve a települést, címet, ahol a termék megtalálható. Ezek alatt található két rádiógombos beviteli mező, ahol kiválaszthatjuk a termékünk kategóriáját, illetve állapotát. Ezek után található egy nagyobb szöveges mező, ahova leírást tudunk rögzíteni a hirdetésünkhöz, itt meg tudunk adni plusz információkat, pontosítani tudjuk hirdetésünk adatait. Az utolsó mező pedig egy fájl feltöltő mező, ennek segítségével tudunk képeket feltölteni a hirdetésünkhöz. Az oldal legalján található a „Hirdetés feladása gomb”, melynek megnyomásával a hirdetésünket fel tudjuk adni, miután az oldal visszairányít minket a főoldalra.

A **profile.html** oldal lehetővé teszi a felhasználónak, hogy megtekinthesse, ellenőrizhesse az adatait. Az oldal kerete szintén egyezik a korábbiakkal. A felhasználói felületen található két navigációs link, az egyikkel visszatérhetünk a főoldalra, a másikkal pedig átléphetünk a profil adatok szerkesztésére készített oldalra, ahol módosíthatjuk adatainkat. A linkek alatt találjuk az adatainkat a következő sorrendben: Felhasználónév, Teljes név, Regisztráció ideje.

A **profileEdit.html** weboldal lehetővé teszi a felhasználóknak, hogy módosítsák a regisztrációkor megadott adataikat. A kezelőfelület tetején az aktuális adatok tekinthetőek meg, alatta pedig a módosítható adatok, beviteli mezőkkel, ahova megadhatja a felhasználó az új adatait. A felület alján két link segítségével lehet navigálni, az első linkre kattintva elmentjük az új adatokat és visszatérünk a profil oldalra, a másik linkkel pedig el tudjuk vetni a módosításokat és azok végrehajtása nélkül térhetünk vissza a profil oldalra.

A weblapok mellett a **login\_reg\_style.css** fájlban található a bejelentkezési és regisztrációs oldalakhoz tartozó stílus leírása, illetve a **style.css** fájlban található a többi oldalhoz tartozó stílus leírás. Ezekben a fájlokban van minden módosítás, formázás, ami a dizájnos és átlátható felhasználói felület biztosítása érdekében szükséges.

**11.4. Funkcionalitás**

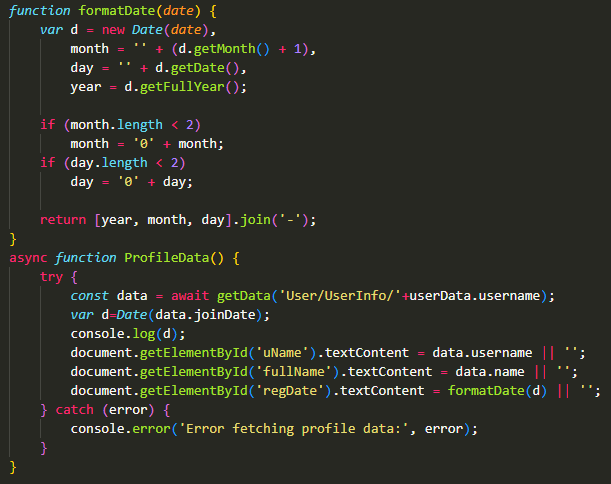
A funkcionalitást az oldalakhoz csatolt .js, avagy JavaScript fájlok biztosítják. Szinte minden oldalhoz tartozik egy saját script fájl, ezzel könnyítve a funkciók átláthatóságát.

Az **auth.js** fájl tartalmazza a bejelentkezési, illetve regisztrációs funkciókat, magát az autentikációt. Két futtatható funkció található benne, a login(), ami biztosítja a bejelentkezési folyamat gördülékenységét és működését, illetve a register(), ami gondoskodik a regisztráció sikerességéről, végbemeneteléről.

A **main.js** felelős az index.html főoldalunkon található funkciók biztosításáról, ezt a fájlt három funkció alkotja. Először is megtalálható két funkció, amik automatikusan végrehajtódnak az oldal betöltésekor, a displayUserInfo(), ami lehetővé teszi a felhasználó nevének kiírását az oldalra a könnyű azonosítás érdekében. A másik ilyen funkció a showItems() parancs, ami megjeleníti az oldalon a hirdetéseket. A harmadik funkciónk a logout(), ez fut le, amikor a felhasználó rákattint a kijelentkezés linkre, illetve automatikusan is lefut, hogyha az oldal úgy töltene be, vagy úgy lenne elérve, hogy nincs bejelentkezett felhasználó.

Az **item.js** script fájl biztosítja az item.html oldalon a hirdetés részletes adatainak megtekintésére szolgáló funkciókat.

A **postItem.js** tartozik a postItem.html oldalhoz, az ebben a fájlban található funkció segítéségével tudjuk feladni hirdetésünket. A fájl egy darab funkciót tartalmaz, azonban ez az egyik leghosszabb kódunk. Biztosítja a hirdetés feladásakor az adatok megfelelő feldolgozását, és átadását a backend részére, hogy tárolhassuk a hirdetés adatait, és később megjeleníthessük azokat. Itt érkeztünk el az első komolyabb problémánkhoz, ami a képek feldolgozása volt. Nehezen tudtuk megoldani, hogy a backend és frontend között a képek átadása jól működjön.

A **profile.js** tartalmazza a profiladatok megtekintéséhez, illetve módosításához szükséges funkciókat. Ebben a fájlban található egy ProfileData() funkció, ami nem csak a felhasználó nevét, hanem minden adatát visszaadja és megjeleníti a profile.html oldalon. A megjelenításbe csúszott egy kisebb probléma, ugyanis a regisztráció dátumát nem feldolgozható formátumban kapja meg a frontend. Ennek javítására létrehoztunk egy formatDate(date) funkciót, ami megkapja a regisztráció dátumát, már a korábbi ProfileData() lefutása közben *Date* változóvá konvertált állapotában, majd leformázza ezt jól megjeleníthető módon. Ezek mellett megtalálható egy ProfileDataChange() funkció, ami lehetővé teszi a felhasználó számára, hogy a profileEdit.html oldalon frissítse saját profiljának az adatait.

ábra 3. A fent említett megoldás a dátum feldolgozására

A **webSocket.js** fájl tartalmazza a WebSocket kapcsolathoz szükséges scripteket.

**11.5. Összegzés**

Összegezve látható tehát, hogy szinte minden eltervezett funkció megvalósult, és ezek egy átlátható, könnyen kezelhető struktúrában kerültek kialakításra. A funkciókat szétbontottuk több fájlba, ezzel értelmezhetőbben hozzácsatolva őket a hozzájuk tartozó oldalakhoz. A fájlokat, illetve képeket külön mappákba rendeztük az átláthatóság tovább növelése érdekében. Mindez megkönnyítette a csapatmunka folyamatát, és lehetővé tette, hogy mindenki könnyen elérje és megtalálja az általa keresett fájlokat.

**12. Backend dokumentáció**

**13. Összefoglalás**

A projekt labor keretében létrehozandó használt hardver árusító platform egy olyan webalkalmazás, amely modern technológiai megoldásokat használ a felhasználók számára gyors, biztonságos és kényelmes hirdetési lehetőséget biztosít. Az oldal biztosítja a termékek gyors feltöltését, keresését, és vásárlását, miközben lehetőséget ad a felhasználók értékelésére. A technikai specifikációk részletesen meghatározzák a felhasznált eszközöket és rendszereket, amelyek biztosítják az alkalmazás teljesítményét és biztonságát. A Proof of Concept (POC) segít abban, hogy az alapvető funkciók prototípusaként bizonyítsuk a projekt megvalósíthatóságát, és a későbbi fejlesztési szakaszok irányát. Az alkalmazás fő célja, hogy a felhasználók gyorsan és hatékonyan tudjanak használt számítástechnikai hardvereket hirdetni, keresni és vásárolni egy felhasználóbarát felületen.

A projekt szempontjából kiemelten fontos a biztonságos adatkezelés, a teljesítmény optimalizálása. A folyamatos tesztelés és hibakezelés kulcsfontosságú a stabil és megbízható működés érdekében.

A projekt labor zárásaként a platform fejlesztéséhez kapcsolódó tapasztalatokat és kihívásokat, valamint a jövőbeni fejlesztési lehetőségeket kell összegyűjteni. További javasolt fejlesztések lehetnek az új funkciók bevezetése, például aukciós rendszer vagy bővített keresési algoritmusok, illetve az integráció különböző fizetési rendszerekkel, hogy a felhasználók számára még kényelmesebbé váljon a vásárlás és eladás.

Ez a dokumentáció részletes áttekintést nyújtott a használt hardver árusító platform tervezéséről és fejlesztéséről. A platform modern technológiai megoldásokkal, átgondolt architektúrával és reszponzív felhasználói élménnyel kíván megfelelni a célcsoport igényeinek, biztosítva a könnyű kezelhetőséget és a biztonságos felhasználást.