

Miskolci Egyetem

Gépészmérnöki és Informatikai Kar

Általános Informatikai Intézeti Tanszék

# Beszámoló szakmai gyakorlatról

**Készítette:**

**Név:** Juhász Balázs

**Neptunkód:** ZUYISF

**Szak:** Mérnökinformatika BSc

Korszerű web technológiák szakirány

## Tartalomjegyzék

1. A cég bemutatása .....	4
2. Tevékenységeim a szakmai gyakorlatom során .....	5
2.1 Front-end fejlesztés: React és Next.js .....	5
2.2 Back-end fejlesztés: Spring Boot .....	5
3. Programnyelvek ismertetése.....	5
3.1 React.....	6
3.2 Next.js.....	7
3.3 Spring Boot .....	9
4. Webshop projekt.....	11
4.1 Home .....	11
4.2 Login .....	12
4.3 Registration .....	12
4.4 Webshop .....	13
4.5 Modify .....	14
4.6 Buy It.....	15
4.7 Cart .....	16
4.8 Client Site.....	18
4.9 Menubar .....	21
4.10 Projekt összegzés.....	22
5. Összegzés.....	22

## Ábrajegyzék

1. ábra: simpleSoft logó .....	4
2. ábra: Home page .....	12
3. ábra: Login page.....	12
4. ábra: Registration page .....	13
5. ábra: Webshop page .....	14
6. ábra: Modify page .....	15
7. ábra: Buy It page .....	15
8. ábra: Cart page .....	16
9. ábra: Cart page - Törlés.....	17
10. ábra: Cart page – Vásárlás adatok bekérésével .....	18
11. ábra: Cart - Vásárlás meglévő adatokkal .....	18
12. ábra: Client Site - Profil beállítások .....	19
13. ábra: Client Site - E-mail megadása.....	19
14. ábra: Client Site - Szállítási cím megadása .....	20
15. ábra: Client Site - Bankkártya adatok megadása.....	21
16. ábra: Menubar .....	22

## 1. A cég bemutatása

A simpleSoft Kft. egy 2007-ben alakult, stabil alapokkal rendelkező szoftverfejlesztő vállalat, amely kiemelkedő szaktudással rendelkezik a webes programozás minden területén. Ezt a tudást neves cégeknek és intézményeknek nyújtott sikeres projektjeik és referenciáik is igazolják.

Mottójuk, az „egyszerű megoldások egy bonyolult világnak”, jól tükrözi szakmájuk iránti elhivatottságukat és azt a törekvésüket, hogy a legösszetettebb kihívásokra is hatékony és egyszerű válaszokat adjanak. Legyen szó bármilyen egyedi igényről vagy elképzelésről, megvalósítják a szükséges webes alkalmazást, applikációt vagy szoftvert.

Szakértő munkatársaik folyamatos képzése és fejlesztése révén arra törekszenek, hogy ügyfeleik és partnereik problémáit a legmagasabb színvonalon oldják meg és a rájuk bízott feladatokat hatékonyan teljesítsék. A napjainkban elérhető legújabb internetes technológiákat (például PHP, .Net, Java, HTML5) alkalmazva egyedi szoftvereket és alkalmazásokat fejlesztenek, maximális szakértelemmel és figyelemmel.

Meggyőződésük, hogy a folyamatos fejlődés elengedhetetlen a sikerhez. Ezért kiemelt figyelmet fordítanak arra, hogy a legmodernebb infrastruktúrát biztosítsák munkájukhoz és szolgáltatásaikhoz, valamint folyamatosan fejlesztik technikai és technológiai hátterüket, valamint humán erőforrásaikat.



1. ábra: simpleSoft logó

## 2. Tevékenységeim a szakmai gyakorlatom során

A cégnél eltöltött 8 hetes szakmai gyakorlatom során lehetőséget kaptam arra, hogy mélyebb gyakorlati tapasztalatokat szerezzek a modern webfejlesztési technológiák alkalmazásában. Több fejlesztési környezet és eszköz megismerésével bővíthettem szakmai tudásomat, különösen a React, Spring Boot és Next.js technológiák terén. Ezek a technológiák a mai fejlesztési világ meghatározó elemei és a gyakorlat során közvetlen tapasztalatot szereztem a mindennapi használatukról, beleértve a projektek tervezését, implementációját és tesztelését is.

### 2.1 Front-end fejlesztés: React és Next.js

A React használatával alaposan megismertem a komponens alapú fejlesztést, ahol a felhasználói felületet különálló, újrahasználmható elemekből építettem fel. Megértettem, hogyan működik a React állapotkezelés és hogyan lehet az adatfolyamokat hatékonyan kezelni anélkül, hogy az egész alkalmazást újra kellene tölteni. Emellett elsajátítottam a React Hookok használatát, mint például a useState és useEffect, amelyek segítségével dinamikus, interaktív felületeket hoztam létre.

A Next.js keretrendszerrel bővítettem a React-ismereteimet és megtanultam, hogyan lehet szerveroldali renderelést alkalmazni a jobb teljesítmény és keresőoptimalizálás (SEO) érdekében. Ennek során részt vettem olyan projektekben, ahol az oldalak statikus előállítás és az API-k integrálása is fontos szerepet kapott. A Next.js beépített funkcionalitásai, mint az automatikus útvonalkezelés és a dinamikus tartalomkezelés, lehetővé tették, hogy hatékonyan hozzak létre reszponzív és gyors webes alkalmazásokat. Ezzel nemcsak a felhasználói élményt növeltem, hanem az alkalmazás teljesítményét is optimalizáltam.

### 2.2 Back-end fejlesztés: Spring Boot

A backend fejlesztés terén Spring Boot használatával dolgoztam, amely egy egyszerűsített, de erőteljes Java-alapú keretrendszer. Aktívan részt vettem olyan projektekben, ahol RESTful API-kat hoztam létre, amelyek kommunikáltak a front-end alkalmazásokkal. Elsajátítottam az adatbázis-kezelést a JPA segítségével, amely megkönnyítette az adatmodellek és az adatbázis-kapcsolatok kezelését. Emellett megtanultam, hogyan lehet a Spring Boot beépített eszközeivel gyorsan és hatékonyan skálázható, biztonságos szerveroldali alkalmazásokat fejleszteni.

## 3. Programnyelvek ismertetése

Ahogy már korábban is említettem a nyolc hét alatt 3 különböző nyelvet sikerült személyre vennem, valamilyen szinten elsajátítani és alkalmazni. A továbbiakban szeretném bemutatni ezeket a nyelveket.

### 3.1 React

A React egy JavaScript alapú, nyílt forráskódú front-end könyvtár, amelyet főleg felhasználói felületek (UI-k) fejlesztésére használnak. Létrehozásának célja az volt, hogy hatékony és egyszerű megoldást nyújtson az interaktív és komplex webes alkalmazások felhasználói felületeinek fejlesztésére, különösen olyan esetekben, amikor az adatok dinamikusán változnak. A React-ot a Facebook fejlesztette ki és 2013-ban tette nyílt forráskódúvá.

A React története:

- Kifejlesztés éve: 2011-ben kezdte el fejleszteni Jordan Walke, a Facebook mérnöke, belső használatra.
- Nyilvános megjelenés: A fentebb említett 2013 volt az az év, amikor a Facebook nyílt forráskódúvá tette a React-ot.
- Facebook és Instagram: A React-ot először a Facebook és az Instagram felhasználói felületeinek kezelésére használták, ahol dinamikusán változó adatokkal dolgoztak. A React azonnal sikeres lett, mivel jelentősen egyszerűsítette az interaktív UI-k készítését.

Mire épül a React:

A React alapvetően a JavaScript nyelvre épül és három kulcsfontosságú technológiát használ:

1. Component-based architecture: A React-ban az alkalmazásokat kis, újrahasználgatható komponensekből építik fel, amelyeket összefűzve egy nagyobb felhasználói felületet alkotnak. Minden egyes komponens egy logikai egységet képvisel (pl. egy gomb, egy űrlap) és ezek külön-külön kezelhetők és karbantarthatók.
2. JSX (JavaScript XML): A JSX egy szintaxisbővítés, amely lehetővé teszi, hogy HTML-szerű kódot írassunk JavaScript-be ágyazva. Ez leegyszerűsíti a komponensek létrehozását, mivel közvetlenül a felhasználói felület megjelenítéséhez szükséges HTML elemeket és az azokhoz tartozó logikát is tartalmazhatja.
3. Virtual DOM: A React hagyományos HTML DOM (Document Object Model) kezelés helyett a Virtual DOM-ot használja. A Virtual DOM egy könnyű másolata a valódi DOM-nak, amely jelentősen javítja a teljesítményt. A React minden adatváltoztatás után frissíti a Virtual DOM-ot, majd csak a minimális szükséges változtatásokat alkalmazza a valódi DOM-on, ezzel növelve a sebességet és a hatékonyságot.

Mire használják a React-ot:

A React-ot elsősorban az alábbiakra használják:

- Single-page applications (SPA): Olyan webes alkalmazások, amelyek egyetlen HTML oldalon futnak és az oldalak tartalma dinamikusan frissül.
- Interaktív UI-k: A felhasználói interakciókra gyorsan és hatékonyan reagáló webes alkalmazások fejlesztése.
- Mobilalkalmazások: A React Native segítségével, amely a React kiterjesztése, natív mobilalkalmazások fejleszthetők.
- Komplex adatvizualizáció: A React rugalmas komponensrendszere lehetővé teszi az adat intenzív, vizualizációs megoldások (pl. dashboardok, diagramok) hatékony megvalósítását.

React fontos tulajdonságai:

- Újrahasználhatóság: A komponensek újrahasználhatók, így csökkentve a fejlesztési időt és egyszerűsítve a karbantartást.
- Deklaratív programozás: A React deklaratív megközelítést alkalmaz, ami azt jelenti, hogy a fejlesztők meghatározzák, hogy mit szeretnének megjeleníteni és a React gondoskodik arról, hogy a megjelenés helyesen történjen.
- Nagy közösség és ökoszisztéma: A React körül hatalmas közösség alakult ki, rengeteg nyílt forráskódú bővítmény és kiegészítő érhető el hozzá, ami lehetővé teszi az egyszerű integrációt és fejlesztést más rendszerekkel.

Elterjedtsége és jövője:

A React napjainak egyik legnépszerűbb front-end könyvtára, amelyet számos nagyvállalat (pl. Facebook, Netflix) használ. A folyamatos fejlesztéseknek és frissítéseknek köszönhetően a React stabil, modern és nagy teljesítményű keretrendszer, amely még évekig meghatározó lesz a webfejlesztésben.

### 3.2 Next.js

Next.js egy nyílt forráskódú, React-alapú keretrendszer, amelyet kifejezetten a React alkalmazások kiszolgálói oldali rendereléséhez és a statikusan előállított oldalak kezeléséhez terveztek. Célja, hogy egyszerűbbé és hatékonyabbá tegye a React alkalmazások fejlesztését, különösen azokét, amelyeknél fontos a teljesítmény és a keresőoptimalizálás. A Next.js-t a Vercel fejlesztette ki és először 2016-ban adták ki.

A Next.js története:

- Kifejlesztés éve: A Next.js-t 2016-ban hozták létre a Vercel mérnökei.
- Nyilvános megjelenés: Az első verziót 2016-ban mutatták be.

- Célja: A React alkalmazások készítésének egyszerűsítése, különös tekintettel a teljesítményre és a SEO-ra, miközben kiszolgálói oldali renderelést és statikus oldalgenerálást biztosít.

Mire épült a Next.js:

A Next.js a React keretrendszerre épül és annak fejlesztési környezetét bővíti ki. A React ugyan hatékony a felhasználói felületek létrehozásában, de a Next.js számos további funkciót kínál, amelyek különösen hasznosak a modern webes alkalmazások fejlesztéséhez:

1. Server-side rendering (SSR): A Next.js lehetővé teszi, hogy a React alkalmazások szerveroldalon rendereljenek, mielőtt a böngésző megjeleníti azokat. Ez javítja a teljesítményt és a keresőoptimalizálást, mivel a felhasználók és a keresőrobotok már az első oldalletöltéskor látják a tartalmat.
2. Static site generation (SSG): A statikus oldalgenerálás segítségével a Next.js lehetőséget ad arra, hogy az oldalak tartalmát előre generálják build időben, így a statikus fájlokat közvetlenül lehet kiszolgáltatni, ami gyorsabb betöltési időket eredményez.
3. API Routes: A Next.js nemcsak a felhasználói felület kezelésében erős, hanem beépített API-kiszolgálási lehetőségeket is kínál. Ez lehetővé teszi, hogy ugyanazon a projektben backend API-kat hozzanak létre, így a fejlesztőknek nem kell különálló szervereket vagy rendszereket kezelniük.
4. Fullstack lehetőségek: A Next.js-ben a front-end és a back-end integráltan működik, így a fejlesztők könnyedén készíthetnek full-stack alkalmazásokat, ahol a front-end és az API-k közvetlenül egy projektben kezelhetők.

Mire használják a Next.js-t:

A Next.js-t széles körben használják különböző típusú alkalmazások fejlesztésére, különösen akkor, ha fontos a teljesítmény és a SEO:

1. Server-side rendered weboldalak: Olyan alkalmazások, amelyek valós időben generálják a tartalmat a szerver oldalon, majd a felhasználóknak továbbítják.
2. Statikus oldalak: Blogok, portfóliók, dokumentációs oldalak vagy marketing oldalak, ahol az előre generált tartalom gyors betöltést és jobb SEO-t eredményez.
3. E-kereskedelmi oldalak: Olyan projektek, ahol fontos a dinamikus változó tartalom és a gyors betöltési idő, miközben a SEO is kulcsfontosságú.
4. Progresszív webalkalmazások: Olyan alkalmazások, amelyek a webes alkalmazások funkcionalitását közelítik a natív mobilalkalmazásokéhoz.

Next.js fontos tulajdonságai:



1. Keresőoptimalizálás: A szerveroldali renderelés és a statikus generálás segítségével a Next.js-ben készült alkalmazások SEO szempontból kedvezőbbek, mivel a tartalom már az első betöltéskor megjelenik a keresőrobotok számára.
2. Teljesítmény: A statikus oldalgenerálás és a dinamikus kiszolgálói oldali renderelés gyorsabb felhasználói élményt eredményez, különösen nagyobb alkalmazások esetében.
3. Rugalmas adatkezelés: A Next.js támogatja mind az Incremental Static Regeneration technológiát, amely lehetővé teszi a statikus oldalak frissítését a háttérben, mind a dinamikus adatlekérdezéseket szerveroldalról.
4. Egyszerű útvonalkezelés: A fájlalapú útvonalkezelés egyszerűsíti a munkát, mivel a fájlrendszerben lévő fájlok automatikusan generálnak útvonalakat.

Elterjedtsége és jövője:

A Next.js napjaink egyik legnépszerűbb React-alapú keretrendszere, amelyet számos nagyvállalat használ, mint például a Twitch, TikTok és Netflix. A Next.js folyamatosan fejlődik és újabb funkciókkal bővül, így egyre elterjedtebb eszközzé válik a modern webalkalmazások fejlesztésében. Az erős közösségi támogatás és a folyamatos fejlesztés révén várhatóan még hosszú ideig kulcsfontosságú eszköz marad a front-end és full-stack fejlesztésben.

### 3.3 Spring Boot

Spring Boot egy nyílt forráskódú keretrendszer, amely a Spring Framework egyszerűsített változata, célja pedig a Java-alapú backend alkalmazások fejlesztésének felgyorsítása és megkönnyítése. A Spring Boot segítségével minimális konfigurálással, előre elkészített sablonok és automatikus beállítások segítségével lehet létrehozni robusztus, skálázható és gyártásra kész alkalmazásokat.

A Spring Boot története:

- Kifejlesztés éve: A Spring Boot-ot 2014-ben adták ki, a Pivotal Software fejlesztői hozták létre.
- Előzmények: A Spring Boot a 2003-ban megjelent Spring Framework-re épül, amely egy erős és rugalmas Java-alapú keretrendszer, de a Spring Boot azért jött létre, hogy egyszerűsítse a hagyományos Spring-alapú fejlesztési folyamatokat.
- Célja: A Spring Boot fő célja az volt, hogy a Spring Framework rugalmasságát és funkcionalitását úgy biztosítsa, hogy kevesebb konfigurációra legyen szükség, ezáltal gyorsabb legyen a fejlesztés és az alkalmazások bevezetése.

Mire épül a Spring Boot:

A Spring Boot a Spring Framework számos komponensére és alapelveire épül, de eltérően attól, automatizált beállításokkal és beépített funkciókkal rendelkezik, amelyek lehetővé teszik, hogy a fejlesztők kevesebb kóddal és konfigurációval hozzanak létre működő alkalmazásokat. Néhány kulcsfontosságú technológia és koncepció, amelyre épül:

- Inversion of Control (IoC): A Spring keretrendszer egyik alapelve az IoC, amely biztosítja, hogy az objektumok létrehozását és kezelését a keretrendszer végzi, nem pedig a fejlesztő.
- Dependency Injection (DI): A Spring Boot automatikusan kezeli a függőségek injektálását, ezáltal csökkentve a kézi konfiguráció szükségességét.
- Auto-configuration: A Spring Boot egyik legnagyobb előnye, hogy automatikusan konfigurálja az alkalmazást az alapján, hogy milyen függőségeket adtunk hozzá. Például, ha a projekt tartalmazza az adatbázis-kezelő függőségeket, akkor a Spring Boot automatikusan beállítja az adatbázis kapcsolódást.
- Spring Boot Starter Packages: Ezek előrecsomagolt függőséggyűjtemények, amelyek különböző technológiák integrálását teszik egyszerűvé.
- Embedded servers: A Spring Boot beépített webserverekkel érkezik, így nincs szükség külső szerverek telepítésére az alkalmazás futtatásához.

Mire használják a Spring Boot-ot:

A Spring Boot a Java backend fejlesztés egyik legnépszerűbb eszköze és számos különböző alkalmazástípushoz használják:

- Webes alkalmazások: A Spring Boot egyszerű és hatékony megoldást nyújt RESTful webszolgáltatások és webes alkalmazások készítésére.
- Microservice architektúra: A Spring Boot kiválóan támogatja a mikroszolgáltatás-alapú architektúrák fejlesztését, ahol több különálló, függetlenül skálázható szolgáltatás működik együtt.
- Adatfeldolgozó alkalmazások: A Spring Boot jól integrálható adatbázisokkal és adatfeldolgozó eszközökkel, így ideális választás adatintenzív alkalmazásokhoz.
- Felhőalapú alkalmazások: A Spring Boot könnyedén használható felhőplatformokkal, mint az Azure vagy Google Cloud, amelyek támogatják a felhőalapú, skálázható alkalmazások futtatását.

Spring Boot fontos tulajdonságai:

- Minimális konfiguráció: A Spring Boot egyik fő előnye a minimális konfiguráció szükségessége, mivel az automatikus beállítások és sablonok megkönnyítik a fejlesztést.
- Gyors fejlesztési folyamat: A beépített webserverek és a spring-boot-devtools funkció (amely támogatja az automatikus újraindítást fejlesztés közben) lehetővé teszi a gyors prototípus-fejlesztést és az iterációkat.
- Spring Ecosystem: A Spring Boot része a hatalmas Spring ökoszisztémának, amely rengeteg integrációt kínál különböző technológiákkal, például adatbázisokkal, üzenetkezelő rendszerekkel és felhőszolgáltatásokkal.
- Biztonság: A Spring Security integrációval könnyen készíthetők biztonságos alkalmazások, amelyek támogatják a hitelesítést és az autorizációt.

- Monitoring és menedzsment: A Spring Boot Actuator segítségével egyszerűen lehet monitorozni és menedzselni az alkalmazásokat, beleértve az egészségellenőrzéseket és a metrikák gyűjtését.

Elterjedtsége és jövője:

A Spring Boot napjaink egyik legnépszerűbb Java-alapú keretrendszere, amelyet széles körben használnak nagyvállalatok és startupok egyaránt, különösen azok, akik robusztus, skálázható backend rendszereket építenek. A Spring Boot közössége és fejlesztése folyamatosan bővül, így várhatóan hosszú távon is meghatározó eszköz marad a Java-alapú fejlesztések világában.

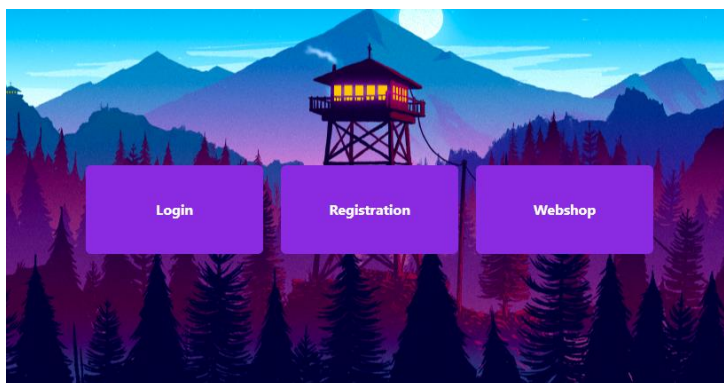
## 4. Webshop projekt

Ahhoz, hogy gyakorolni és rögzíteni tudjuk tudásunk, ki kellett találnunk egy saját projektet. Szakmai gyakorlatos hallgató társammal sokat gondolkoztunk, hogy mi lenne az a webalkalmazás, aminek az elkészítéséhez elegendő tudásunk van, de azért még fejlődni is tudunk vele. A sok ötletelés végén megszületett a döntés, egy webshop-ot fogunk elkészíteni.

Elkezdtek leírni az ötleteinket, hogy mi hogy nézzen ki, hogyan működjön, mi legyen a lényege. Ötletelést követően felosztottuk a feladatokat egymás között. Annyiban egyeztünk meg első körben, hogy övé a backend, enyém pedig a frontend rész. Továbbá nem utolsó sorban kiválasztottuk a programnyelveket, a backend részen nem volt sok választási lehetőségünk, viszont frontend-en kettő közül is tudtunk választani. Így választottuk ki a frontend nyelvnek a react-ot és a backend nyelvnek a spring boot-ot. A továbbiakban oldalakra bontva fogom bemutatni a webalkalmazás jelenlegi állapotát frontend oldalról. Azért emelem ki, hogy a jelenlegi állapotát, ugyanis nem lett teljesen kész, de a szakmai gyakorlatot követően is fogjuk folytatni a fejlesztését, amíg ötleteinken megfelelő állapotba nem kerül.

### 4.1 Home

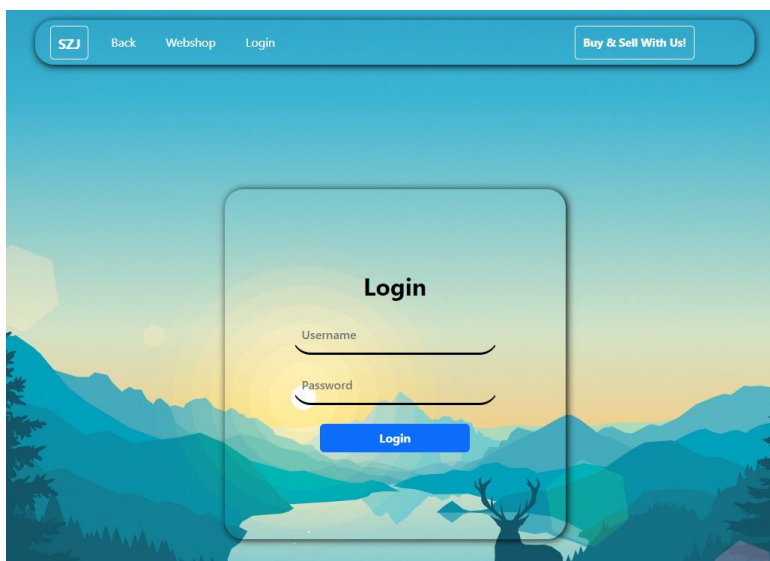
A kezdőlapra lépve három gomb található, amelyek funkciói és elnevezései a felhasználó bejelentkezési állapotától függően változhatnak. Kijelentkezett állapotban a gombok a következő funkciókat látják el: „Login”, „Registration” és „Webshop”. Ezek a felhasználói hitelesítéshez és a webshop böngészéséhez biztosítanak hozzáférést. Bejelentkezett állapotban a gombok „Client Site”, „Webshop” és „Logout” funkciókra módosulnak, így a felhasználók beléphetnek a saját ügyféloldalukra, folytathatják a webshop használatát, illetve kijelentkezhetnek. Az alkalmazás minden oldala reszponzív kialakítással rendelkezik, biztosítva, hogy különböző méretű kijelzőkön is megfelelően jelenjen meg. A CSS kód gondoskodik arról, hogy a felület könnyen használható maradjon mobil eszközökön és asztali számítógépeken egyaránt.



2. ábra: Home page

## 4.2 Login

A Login oldalon a felhasználók bejelentkezhetnek egy egyszerű felhasználónév és jelszó páros megadásával. Amennyiben a beírt adatok helyesek, a rendszer automatikusan átirányít egy másik oldalra, ahol a bejelentkezett funkciók elérhetővé válnak. Ha az adatok helytelenek, a rendszer hibaüzenetet küld, amely tájékoztat a sikertelen bejelentkezésről és lehetőséget ad a javításra.

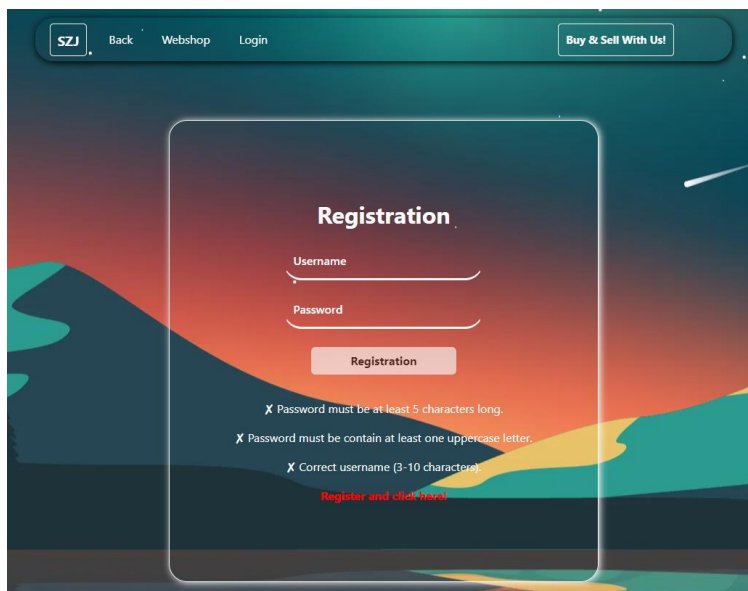


3. ábra: Login page

## 4.3 Registration

Amennyiben nem csupán a webshop termékeit szeretnénk böngészni, hanem azokat kosárba helyezni vagy megvásárolni, először regisztrálnunk kell. A regisztráció folyamata egyszerű, mindössze egy egyedi felhasználónévre és jelszóra van szükségünk. A felhasználónévnek minden esetben egyedinek kell lennie, míg a jelszónál egy dinamikus ellenőrző lista segít abban, hogy megfeleljen a biztonsági követelményeknek. Ez az ellenőrző lista folyamatosan figyeli a jelszó beviteli mezőjébe írt karaktereket és visszajelzést ad arról,

hogy a jelszó tartalmazza-e a szükséges feltételeket, például kis- és nagybetűket. Emellett a lista ellenőrzi azt is, hogy a felhasználónév megfelelő hosszúságú-e, biztosítva ezzel a zökkenőmentes regisztrációt.



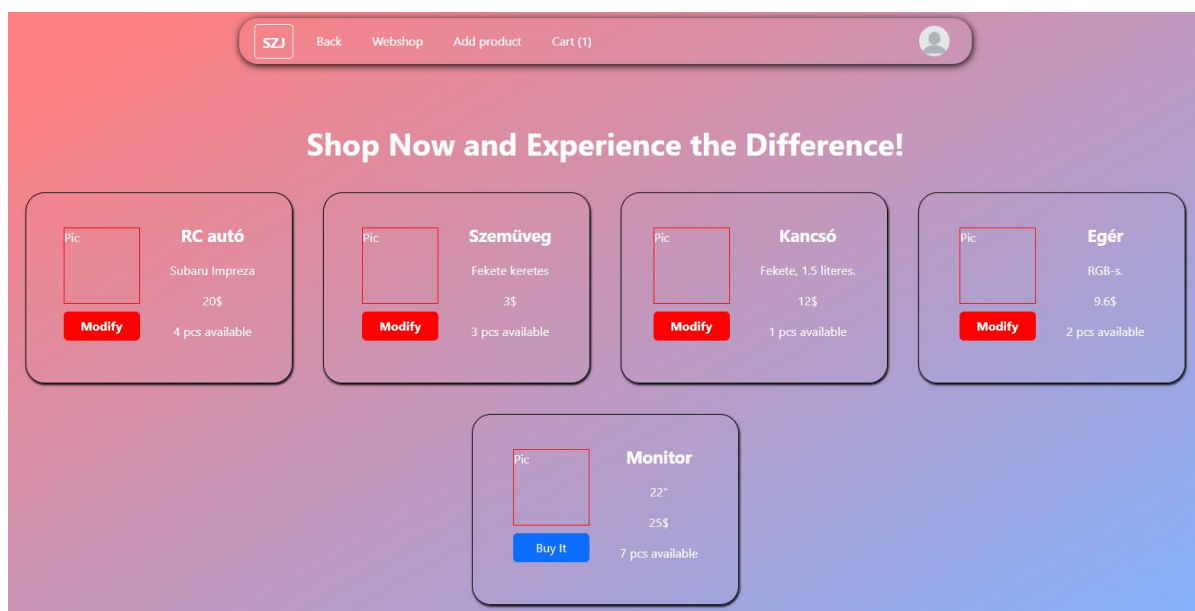
4. ábra: Registration page

#### 4.4 Webshop

A Webshop alapvetően különféle termékeket tartalmaz, amelyek meghirdetéséhez bejelentkezett fiókra van szükség. Ha kijelentkezett állapotban böngésszük az oldalt, csak a termékeket tudjuk megtekinteni, vásárlási lehetőség nélkül. Bejelentkezett felhasználóként viszont a termékek mellett elérhetővé válik a „Buy It” gomb, amelyre kattintva a termék kosárba helyezhető, miután megadtuk a kívánt darabszámot.

Amennyiben a meghirdetett terméket a saját bejelentkezett fiókunkon keresztül adtuk fel, akkor a „Buy It” gomb helyett a „Modify” gomb jelenik meg. Ezen keresztül lehetőségünk van a termék adatainak módosítására, például elírások javítására vagy kiegészítések hozzáadására, valamint a hirdetés törlésére is.

Az oldal jelenleg fejlesztés alatt áll, piros keretek jelzik, hogy hol lesznek láthatók a termékekhez feltöltött képek, amelyek egyelőre még nem kerültek megjelenítésre. A képek mellett megjelenik a termék neve, alatta a rövid leírása, az egységára, valamint a készleten lévő darabszám. Ezek az információk segítenek a vásárlóknak a döntéshozatalban és a rendelés leadásában.

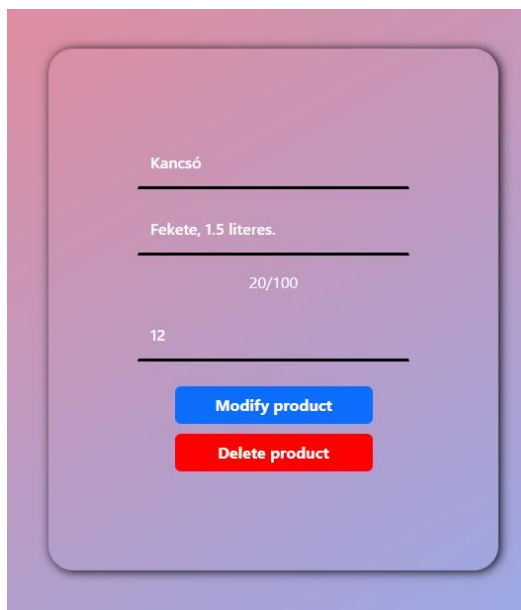


5. ábra: Webshop page

#### 4.5 Modify

A Modify oldal betöltésekor egy űrlapot kapunk, amely lehetőséget biztosít a meghirdetett termék adatainak módosítására. Az űrlap tartalmazza a termék nevét, leírását és árát, amelyeket könnyedén megváltoztathatunk. Az űrlap alatt két gomb található: az egyik a változtatások mentésére szolgál, a másik pedig a termék törlésére.

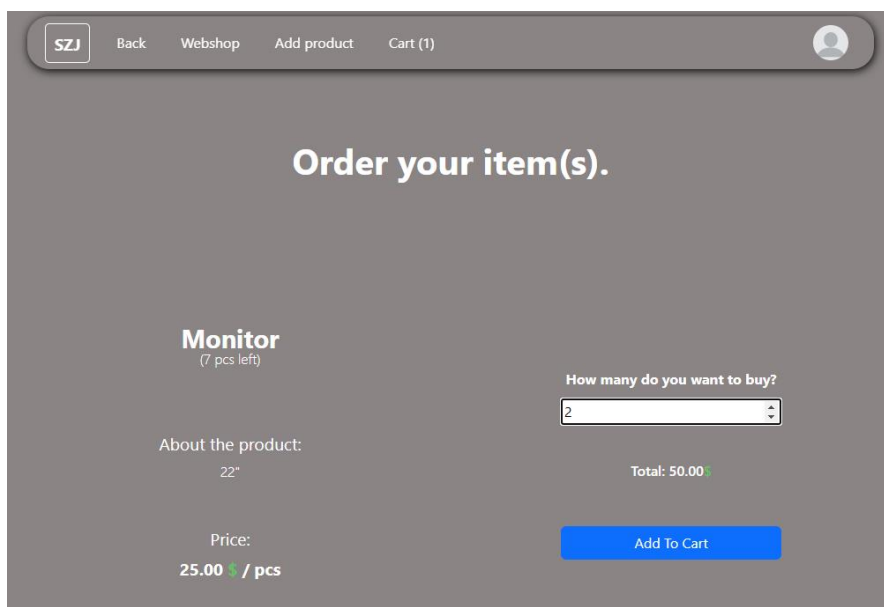
A leírás mező alatt egy számláló jelenik meg, amely figyeli a beírt karakterek számát és legfeljebb 100 karakterben korlátozza a leírás hosszát. Ez a szabály biztosítja, hogy a termék leírása tömör és könnyen áttekinthető legyen. A számláló valós időben jelzi, hogy hány karakter áll még rendelkezésre, így segítve a felhasználót az előírások betartásában.



6. ábra: Modify page

#### 4.6 Buy It

Ezen az oldalon megadhatjuk, hogy a kiválasztott termékből hány darabot szeretnénk a kosárba helyezni. Az oldalon láthatjuk a termék részletes adatait is: a termék nevét, az elérhető készletet, a termék rövid leírását, valamint a darabárát. Amikor kiválasztjuk a kívánt mennyiséget, a rendszer automatikusan kiszámítja és megjeleníti, hogy a kiválasztott mennyiség összesen mennyibe fog kerülni. Ez a funkció segít a felhasználónak a vásárlás előzetes kalkulációjában és abban, hogy egyszerűen ellenőrizhesse, mekkora lesz a termék költsége a kosárba helyezés előtt.



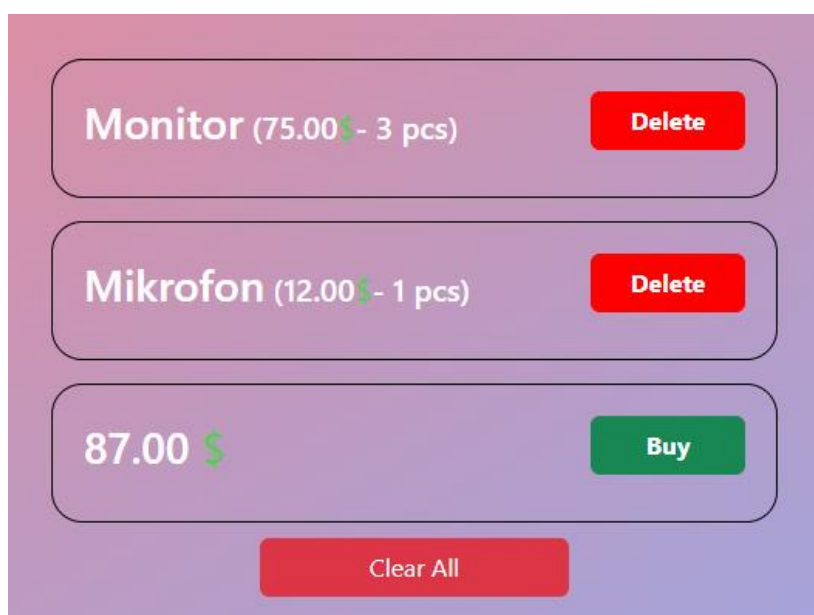
7. ábra: Buy It page

#### 4.7 Cart

A „Buy It” felületen található „Add To Cart” gomb megnyomásával a terméket a kosárba helyezhetjük. A menüsávon az alkalmazás zárójelben megjeleníti, hogy hány különböző termék található a kosarunkban, így könnyen követhetjük a kosarunk tartalmát. Amikor a Cart oldalra lépünk, részletesen láthatjuk az összes terméket, amelyeket eddig hozzáadtunk.

Az oldal összefoglalja a kosárban található termékek összárát és darabszámát, illetve azt is, hogy a teljes kosár megrendelése esetén mennyibe fog kerülni az összes termék. Ezen az oldalon háromféle gomb található:

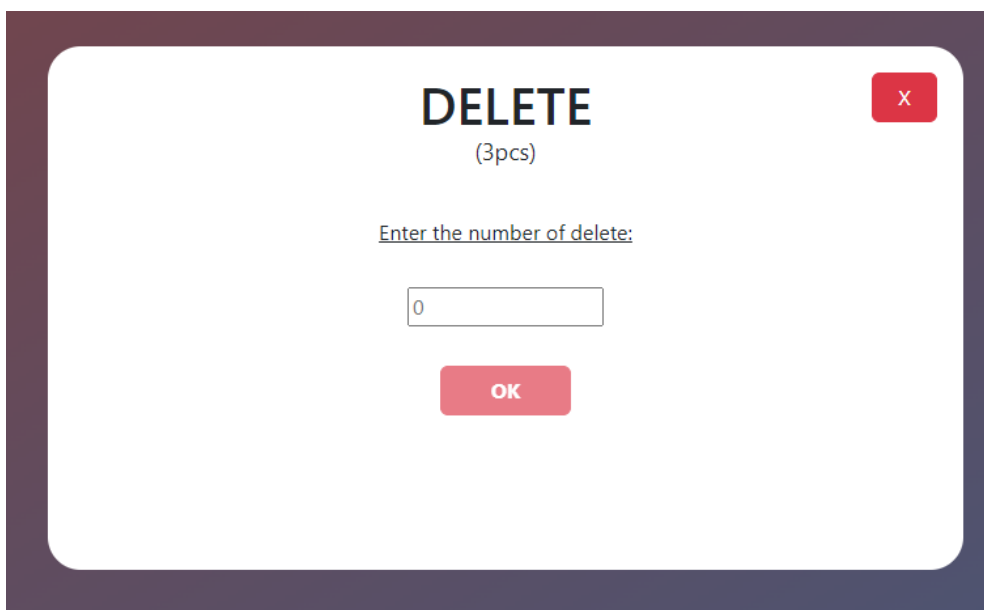
- Delete: Ezzel a gombbal egy-egy konkrét termék darabszámát tudjuk csökkenteni, akár tudjuk a terméket törölni a kosárból.
- Buy: A teljes kosár tartalmát megvásárolhatjuk ezzel a gombbal.
- Clear All: Ez a gomb lehetőséget ad arra, hogy az összes terméket egyetlen kattintással eltávolítsuk a kosárból.



8. ábra: Cart page

Amikor a „Delete” gombra kattintunk, egy felugró ablak jelenik meg, ahol pontosíthatjuk, hogy az adott termékből hány darabot szeretnénk törölni. Ez lehetővé teszi, hogy ne az egész terméket távolítsuk el a kosárból, hanem csak csökkentsük a megvásárolni kívánt mennyiséget. Az ablakban beírhatjuk a kívánt darabszámot, így könnyedén kezelhetjük, hogy mennyivel kevesebb termék maradjon a kosárban. Amennyiben kitöröljük a kosárban lévő termék teljes mennyiségét, akkor azzal kitöröljük a terméket a kosárból. Az ablak megjelenése segít abban, hogy rugalmasan és átgondoltan módosítsuk a kosár tartalmát a vásárlási folyamat során.



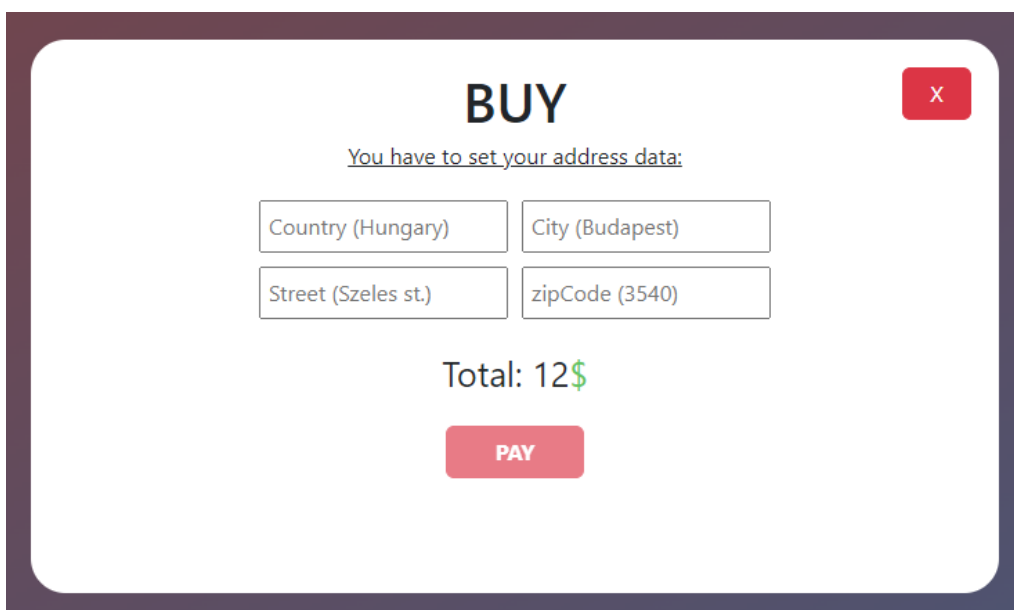


9. ábra: Cart page - Törlés

A „Buy” gombra kattintáskor szintén egy felugró ablak jelenik meg, amelynek elrendezése attól függően változik, hogy korábban már rendeltünk-e, illetve beállítottuk-e a postázási címünket a „Client Site”-on.

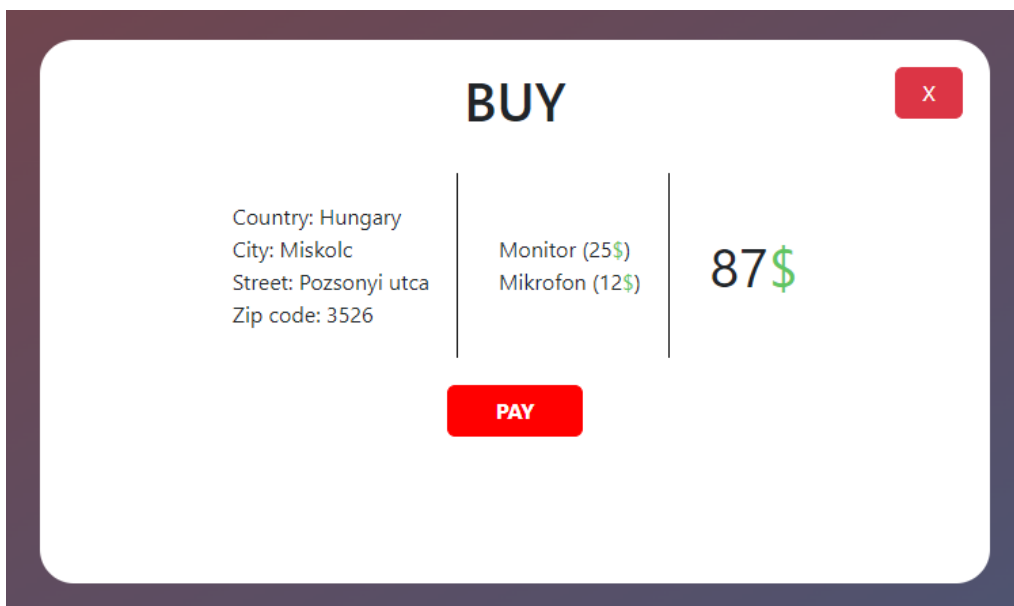
- Amennyiben még nem rendeltünk semmit és nincs megadva postázási címünk, egy űrlap jelenik meg, ahol megadhatjuk a szállítási adatokat. Az űrlap kéri a megrendelő országát, városát, utcáját és irányítószámát. Ezek az információk szükségesek ahhoz, hogy a rendelés később megfelelően teljesíthető legyen.
- Ha már megadtuk korábban a címet, ez az űrlap nem jelenik meg és csak egy összefoglalót fogunk látni a kiválasztott termékekről, illetve a végösszegről.

Miután a szállítási adatokat kitöltöttük, a „PAY” gombra kattintva fejezhetjük be a rendelést, feltéve, hogy van elegendő feltöltött pénzünk a rendszerben. Ha a fizetési feltételek teljesülnek, a rendelés sikeresen lezárul, az összeg levonásra kerül és a kosár tartalma törlődik.



The screenshot shows a 'BUY' modal window with a red close button in the top right corner. Below the title, a message states 'You have to set your address data:'. There are four input fields arranged in a 2x2 grid: 'Country (Hungary)', 'City (Budapest)', 'Street (Szeles st.)', and 'zipCode (3540)'. Below these fields, the text 'Total: 12\$' is displayed in green. At the bottom center is a red 'PAY' button.

10. ábra: Cart page – Vásárlás adatok bekérésével



The screenshot shows the 'BUY' modal window with pre-filled data. On the left, the address is listed: 'Country: Hungary', 'City: Miskolc', 'Street: Pozsonyi utca', and 'Zip code: 3526'. In the center, the items are listed: 'Monitor (25\$)' and 'Mikrofon (12\$)'. On the right, the total price '87\$' is shown in large green font. At the bottom center is a red 'PAY' button. A red close button is in the top right corner.

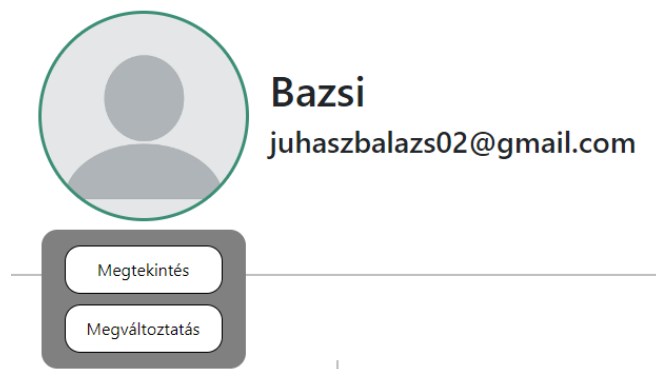
11. ábra: Cart - Vásárlás meglévő adatokkal

#### 4.8 Client Site

A Client Site oldalon a felhasználók személyre szabhatják a saját fiókjukat, és hozzáférhetnek különféle beállításokhoz. Jelenleg ez a rész fejlesztés alatt áll, de már van néhány működő funkció.

Az oldal tetején található egy egyszerű profilkép beállítási lehetőség. Az alapértelmezett profilképre kattintva egy felugró ablak jelenik meg, ahol a felhasználók megtekinthetik a jelenleg beállított profilképüket, vagy feltölthetnek egy újat. Jelenleg a profilkép feltöltése funkció már üzemel, azonban a feltöltött kép megjelenítése még nem működik teljes mértékben.

A profilkép mellett a fiók neve is látható, valamint alatta az e-mail cím, amennyiben az be van állítva. Az oldal további személyre szabási lehetőségei, mint például a jelszó módosítása és a személyes adatok frissítése várhatóan bevezetésre kerülnek a közeljövőben.



**12. ábra: Client Site - Profil beállítások**

Ahhoz, hogy beállítsuk a szállítási címünket, először meg kell adnunk az e-mail címünket a megfelelő formátumban. Az e-mail cím megadása fontos lépés, mivel ez az azonosítás alapjául szolgál. Az oldal egy validáló űrlapot biztosít, amely ellenőrzi, hogy az e-mail cím helyesen van-e megadva.

**13. ábra: Client Site - E-mail megadása**

Miután az e-mail cím beállítása megtörtént, az e-mail cím megadására szolgáló űrlap eltűnik, és helyette megjelenik a szállítási cím megadására szolgáló űrlap. Ezt az űrlapot is ki kell tölteni a megfelelő formátumok szerint, amely magában foglalja az országot, várost, utcát és irányítószámot.

Amint a szállítási cím űrlap kitöltése és mentése megtörténik, az adatok automatikusan elmentődnek. Ennek köszönhetően a későbbi vásárlások során, amikor a kosárban lévő termékeket megvásároljuk, már nem lesz szükség további szállítási adatok megadására. A rendszer automatikusan felhasználja a mentett címet, és csak a fizetési folyamatot kell elvégeznünk a vásárlás befejezéséhez. Ez egyszerűsíti a vásárlási folyamatot, és biztosítja, hogy a rendelésünk gyorsan és zökkenőmentesen teljesüljön.

## Add your address

City

Budapest

Country

Hungary

Street

Szeles st.

Zip code

3540

Save

**14. ábra: Client Site - Szállítási cím megadása**

Fontos kiemelni, hogy amennyiben már van beállítva szállítási címünk, az oldal az alsó „Save” gomb helyett „Modify” gombot fog megjeleníteni. Ez lehetőséget biztosít arra, hogy módosíthassuk a már megadott szállítási adatokat. Az űrlap kitöltésekor a korábban megadott adatok placeholder formájában jelennek meg, így könnyen ellenőrizhetjük és frissíthetjük a címünket az esetleges változások szerint.

Az oldal utolsó részlete a pénzfeltöltés funkció. Mivel valódi pénz tranzakciójára nincs lehetőség, a rendszer bankkártya adatok megadásával és a kívánt összeg megadásával végzi el a pénz feltöltését az ügyfél számára. A rendszer nem tárolja a bankkártya adatokat, így minden feltöltés során újra meg kell adni ezeket az információkat. Nincsenek előre kitöltött mezők az előző feltöltések adataival, ezzel biztosítva a pénzügyi adatok biztonságát és védelmét.

### Bank Card Details

**Card Number**

**Card Holder Name**

**Expiry Date**

**CVV**

**Withdraw Value (\$)**

**15. ábra: Client Site - Bankkártya adatok megadása**

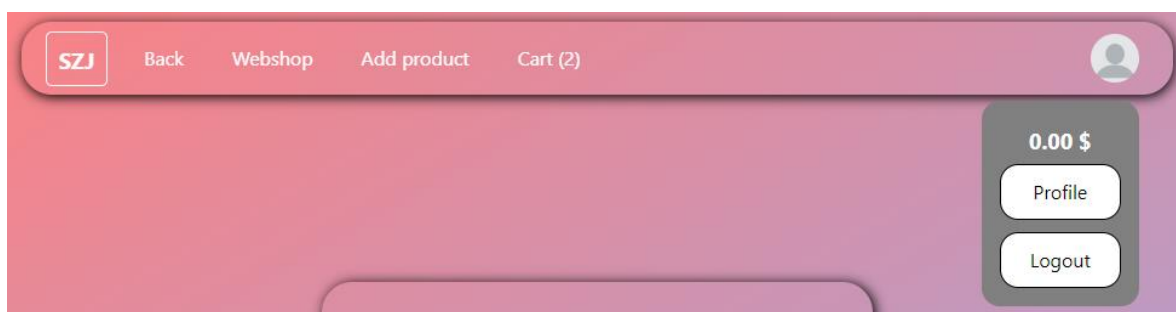
## 4.9 Menubar

A menüsáv kétféle kinézettel rendelkezik, attól függően, hogy bejelentkezett vagy kijelentkezett állapotban vagyunk.

- Kijelentkezett állapotban a menüsáv a következő gombokat tartalmazza:
  - Back: Visszatérés a korábbi oldalra.
  - Webshop: Átirányítás a webshop oldalára, ahol a felhasználók böngészhetnek a termékek között.
  - Login: Belépési lehetőség, amely átirányít a bejelentkezési felületre.
- Bejelentkezett állapotban a menüsáv az alábbi opciókat tartalmazza:
  - Add Product: Lehetőség új termékek hozzáadására a rendszerhez.
  - Cart: Hozzáférés a kosárhoz, ahol a felhasználók láthatják a kiválasztott termékeket és kezelhetik a vásárlási folyamatot.
  - Profilkép: A bejelentkezett felhasználó profilképe, amelyre kattintva egy legördülő menü jelenik meg.

A profilképre kattintva a legördülő menüben az alábbi opciók érhetők el:

- Egyenleg: Megjeleníti a felhasználó aktuális egyenlegét.
- Client Site: Átirányít a felhasználói fiók beállításainak oldalára, ahol további személyre szabási lehetőségek érhetők el.
- Logout: Kijelentkezés a fiókból, ami visszavisz minket a kijelentkezett állapotba, és az oldalon elérhető opciók is ennek megfelelően frissülnek.



**16. ábra: Menubar**

#### 4.10 Projekt összegzés

A projekt célja egy modern webalkalmazás kifejlesztése, amely lehetővé teszi a felhasználók számára, hogy termékeket böngésszenek, vásároljanak, és kezeljék a saját fiókjukat. Az alkalmazás több fontos funkciót tartalmaz, beleértve a felhasználói bejelentkezést, a termékek hozzáadását és vásárlását, valamint a felhasználói fiók személyre szabását.

A jövőbeli fejlesztések során a projekt további bővítése és finomítása várható. A következő lépések közé tartozik a felhasználói élmény továbbfejlesztése, például a profilkép megjelenítése és termékek képeinek megjelenítése. A pénzfeltöltési folyamat biztonságának növelése érdekében is tervezett új funkciók bevezetése, valamint a felhasználói fiók személyre szabási lehetőségeinek bővítése, mint például a jelszó módosítása és további személyes adatok kezelése. Ezek a fejlesztések célja, hogy a webalkalmazás még hatékonyabbá és felhasználóbarátabbá váljon, megfelelve a felhasználói igényeknek és a piaci elvárásoknak.

A projekt során mélyebb gyakorlati tapasztalatokat szereztem a modern webfejlesztési technológiák alkalmazásában, mint a React, Spring Boot. Jelentősen bővítette szakmai tudásomat és csapatmunkában való jártasságom.

### 5. Összegzés

Összességében a simpleSoft Kft.-nél töltött 8 hetes szakmai gyakorlat rendkívül értékes és tanulságos volt. Az itt szerzett tapasztalatok lehetőséget biztosítottak számomra, hogy mélyebb gyakorlati ismereteket szerezzek egy dinamikusan fejlődő szoftverfejlesztő cégnél. A gyakorlat során három programnyelvet is megismerhettem, és jelentős fejlődést értem el a React, Spring Boot és Next.js használatában. Emellett hálás vagyok a főállású kollégák folyamatos segítségéért, akik támogatást nyújtottak, amikor elakadtam egy-egy folyamatban. Az ő útmutatásuk és szakmai tudásuk hozzájárult a sikeres fejlődésemhez és a szakmai készségeim fejlesztéséhez. Az itt szerzett tapasztalatok nemcsak technikai tudásomat bővítették, hanem hozzájárultak a csapatmunkában való jártasságom fejlődéséhez is.