

SZAKDOLGOZAT

Sóti József Máté

2023

Magyar Agrár- és Élettudományi Egyetem
Károly Róbert Campus

Nagyon jó sportegyesület
webalkalmazás

Konzulens

Sike Zoltán

mesteroktató

Készítette

Sóti József Máté

gazdaságinformatikus

nappali tagozat

2023

Tartalomjegyzék

1.	Bevezetés.....	4
1.1.	Köszönetnyilvánítás.....	4
1.2.	Témaválasztás.....	4
1.3.	Elméleti jelentőség.....	4
2.	Irodalmi áttekintés.....	5
2.1.	Fejlesztői környezet	5
2.1.1.	Notepad++ v8.5.2	5
2.1.2.	Visual Code	6
2.1.3.	XAMPP v3.2.4	7
2.1.4.	phpMyAdmin v5.0.3	8
2.2.	Webprogramozás	8
2.3.	Felhasznált technológiák.....	9
2.3.1.	HTML	9
2.3.2.	CSS	10
2.3.3.	JavaScript	11
2.3.4.	Bootstrap.....	12
2.3.5.	jQuery	12
2.3.6.	Node.js.....	13
3.	Anyag és módszertan	15
3.1.	Weblap készítés kritériumai.....	15
3.2.	Módszer	15
4.	Weblap bemutatása	16
4.1.	Adatbázis	16
4.1.1.	Adatbázis fogalma	16
4.1.2.	Adatbázisban használt változók.....	16
4.1.3.	Szakdolgozat adatbázis.....	17
4.1.4.	Users adattábla.....	18
4.1.5.	Profiles adattábla	19
4.1.6.	Genders adattábla	21
4.1.7.	Products adattábla.....	22
4.1.8.	Orders adattábla.....	22
4.1.9.	Comments adattábla	24
4.2.	Frontend	25
4.2.1.	Mi is az a frontend.....	25

4.2.2.	Index.html.....	25
4.2.3.	Regist.html.....	27
4.2.4.	Login.html	28
4.2.5.	Home.html	29
4.2.6.	Pride.html	30
4.2.7.	Contacts.html.....	31
4.2.8.	Comments.html	32
4.2.9.	Webshop.html.....	33
4.2.10.	Purchase.html	34
4.2.11.	Profile.html	35
4.3.	Backend	36
4.3.1.	Mi is az a backend	36
4.3.2.	Index.php	36
4.3.3.	Db_connect.php.....	37
4.3.4.	Jump.php	37
4.3.5.	Regment.php.....	38
4.3.6.	Login.php.....	39
4.3.7.	Logout.php.....	39
4.3.8.	Insert.php	39
4.3.9.	Order.php	40
4.3.10.	Update.php.....	40
5.	Eredmények, következtetések és javaslatok.....	41
6.	Összefoglalás.....	42
7.	Irodalomjegyzék.....	43
8.	Ábrajegyzék	44
9.	Nyilatkozatok	45

1. Bevezetés

1.1. Köszönetnyilvánítás

Ezúton szeretnék köszönetet mondani konzulensemnek és témavezetőmnek, Sike Zoltán tanár úrnak, aki egyetemi éveim során és különösen a szakdolgozati konzultációk alatt, nagyon sokat hozzájárult szakértelmével, magyarázataival, tanácsaival és segítőkészségével e szakdolgozat elkészítéséhez. Ezen felül még hálával tartozom kollégáimnak, szaktársaimnak és barátaimnak, akik tartották bennem a lelket és folyamatosan támogattak. Köszönöm nekik, hogy tanulmányaim és munkám során türelemmel és megértéssel járultak hozzá célom eléréséhez.

1.2. Témaválasztás

Szakdolgozatom témájaként azért választottam a webprogramozást, mert mindig is foglalkoztatott a programfejlesztés, viszont a HTML, PHP és a Java nyelvek közelebb állnak a szívemhez, mint a C#, ezen felül ezt érdekesebbnek és látványosabbnak is találom. Legelőször Jászberényben a Gróf Apponyi Albert Általános Iskolának készítettem volna egy saját honlapot, de a vezetőség az utolsó pillanatban sajnálatos módon meggondolta magát, és leállította a projektet. Ezután a konzulensem segítségét kértem témaválasztás céljából, fel is sorolt párat, amiből az egyik nagyon megnyerte a tetszésem, így végül egy darts sportegyesületnek készítettem weblapot. Alapból egy létező sportegyesületnek készítettem volna weblapot, viszont nem túl sokat ismerek, amelyeknek a létezéséről tudok, azoknak pedig már van saját honlapja, ezért végül egy fiktív egyesületnek készítettem.

1.3. Elméleti jelentőség

Ennek az oldalnak az elkészítése talán annyi elméleti jelentőséggel bírhat, hogy webprogramozás témáján belül tudhatjuk, a regisztrációs felületek elkészítésének mikéntjét, felhasználók belépését, avagy miként működik a bejelentkezés és az ezt követő SESSION, magyarul munkamenet. Minimálisan a fórumok működésébe is bele láthatunk, hiszen írhatunk kommenteket az egyesületről, amelyek meg is jelennek a komment fülön. Mindezekon felül, a weboldalak admin felületébe is belátást nyerhetünk alap szinten, és a háttérben lapuló adatbázisok felépítésébe is.

2. Irodalmi áttekintés

2.1. Fejlesztői környezet

Mivel témámként weboldalfejlesztést választottam, ezért az ehhez szükséges alapvető eszközöket használtam. A következők a felhasznált eszközök: Notepad++ a kódolás folyamatához, XAMPP Control Panel v3.2.4-et pedig localhosting-ra.

2.1.1. Notepad++ v8.5.2

A Notepad++ egy ingyenes, nyílt forráskódú szövegszerkesztő Windowshoz, amelyet úgy terveztek, hogy hatékony és könnyű eszköz legyen a programozók és fejlesztők számára. Ez a Microsoft beépített Jegyzettömb alkalmazásának továbbfejlesztett változata, és olyan funkciók széles skáláját kínálja, amelyek hatékonyabbá és produktívabbá teszik a kódolást és a szövegszerkesztést.

Egyik legfontosabb jellemzője a programozási nyelvek széles skálájának támogatása, beleértve a C, C++, Java, HTML, PHP, CSS stb. Ez ideális választássá teszi azoknak a fejlesztőknek, akik több nyelven dolgoznak, vagy akiknek olyan eszközre van szükségük, amely képes kezelni a különböző szintaxis kiemelési és automatikus kiegészítési lehetőségeket.

Másik figyelemre méltó tulajdonsága a nagy fájlok gyors és egyszerű kezelése. Sok más szövegszerkesztővel ellentétben, amelyek lelassulhatnak, és nem reagálnak nagy fájlok kezelésekor, a Notepad++ gyors és reszponzív marad még akkor is, ha több százezer sornyi kódot tartalmazó fájlokkal dolgozunk.

Számos egyéb funkciót is tartalmaz, amelyek sokoldalú és hatékony szövegszerkesztővé teszik, beleértve a több lap támogatását, a fogd és vidd funkciót, a makró rögzítést, valamint a szerkesztő testreszabásához használható bővítmények széles skáláját. hogy megfeleljen az mindenki egyedi igényeinek.

Kódolási funkciói mellett általános szövegszerkesztési feladatokra is használható, mint például jegyzetelés, dokumentumok írása és konfigurációs fájlok szerkesztése. Számos szövegkódolást támogat, beleértve az UTF-8-at is. Nagy és aktív felhasználói közösséggel rendelkezik, amely számos bővítményt hozott létre a szerkesztő funkcionalitásának javítása érdekében. Ide tartoznak az FTP- és SFTP-integráció bővítményei, a kódrészletek, valamint a népszerű kódtárolókkal, például a GitHub-bal való integráció.

Összességében a Notepad++ egy rendkívül hatékony és rugalmas szövegszerkesztő, amely minden készségszintű programozó és fejlesztő számára kiválóan alkalmas. Akár

csak most kezdi a kódolást, akár tapasztalt fejlesztő, aki hatékony és hatékony eszközt keres.

2.1.2. Visual Code

A Visual Studio Code, közismert nevén VS Code, a Microsoft által fejlesztett, népszerű nyílt forráskódú kódszerkesztő. Úgy tervezték, hogy könnyű, gyors és rugalmas legyen, a funkciók széles skálájával, amelyek minden készségszintű fejlesztőt kielégítenek.

A VS Code egyik legfontosabb előnye a programozási nyelvek széles skálájának beépített támogatása, beleértve az olyan népszerű nyelveket, mint a Python, a JavaScript és a C++. Ez azt jelenti, hogy a fejlesztők a VS Code-ot használhatják minden kódolási igényük kielégítésére anélkül, hogy több alkalmazás között kellene váltaniuk.

A VS Code másik előnye a kiterjesztések hatalmas ökoszisztémája, amely lehetővé teszi a fejlesztők számára, hogy testreszabják és bővítsék a szerkesztő funkcióit sajátos igényeiknek megfelelően. Ezek a bővítmények a területek széles skáláját fedik le, a szöszöléstől és hibakereséstől a kód formázásáig és a szintaxis kiemeléséig.

A VS Code egyik kiemelkedő tulajdonsága az integrált terminál, amely lehetővé teszi a fejlesztők számára, hogy parancsokat hajtsanak végre és szkripteket közvetlenül a szerkesztőből indítsanak el. Ez leegyszerűsíti a fejlesztési folyamatot, és szükségtelenné teszi a fejlesztőknek a szerkesztő és a parancssor közötti váltást.

Mindemellett a VS Code aktív felhasználói és fejlesztői közösséggel rendelkezik, akik hozzájárulnak a szerkesztő fejlesztéséhez és támogatást nyújtanak más felhasználóknak. Ez biztosítja, hogy a szerkesztő folyamatosan fejlődjön és javuljon, és rendszeresen megjelennek új szolgáltatások és hibajavítások.

Összefoglalva, a Visual Studio Code egy hatékony és sokoldalú kódszerkesztő, amely számos szolgáltatást és bővítményt kínál a fejlesztők igényeinek kielégítésére. Intuitív kezelőfelülete és robusztus funkcionalitása kiváló választássá teszi minden szintű fejlesztő számára, aktív közössége pedig biztosítja, hogy folyamatosan fejlődjön, hogy megfeleljen a felhasználók igényeinek.

Többen is ajánlották figyelmembe a VS Code-ot és ki is próbáltam, viszont végül visszatértem a Notepad++-ra. Jogosan merülhet fel a kérdés, hogy miért is. A válaszom nagyon egyszerű, számomra szimpatikusabb volt a Notepad letisztultabb megjelenése. Funkcionalitás szempontjából bőven megfelelt a szükségeimnek. Míg a másik fejlesztői környezet valóban többet tud, azokat a funkciókat úgysem vettem volna igénybe.

2.1.3. XAMPP v3.2.4

Az XAMPP egy népszerű szoftvercsomag, amely tartalmazza a webszerver helyi számítógépen történő futtatásához szükséges összes összetevőt. A "Cross-Platform, Apache, MySQL, PHP és Perl" rövidítése, és elérhető Windows, Mac és Linux operációs rendszerekhez.

Elsődleges célja, hogy helyi fejlesztői környezetet hozzon létre a webalkalmazások számára, amelyeket azután tesztelni lehet, mielőtt egy élő szerverre telepítenék. Különösen hasznos azoknak a webfejlesztőknek, akiknek tesztelniük kell PHP-szkriptjeiket vagy MySQL-adatbázisaikat, mielőtt feltöltenék őket egy távoli szerverre.

Egyik legjelentősebb előnye, hogy leegyszerűsíti a helyi szerverkörnyezet beállításának folyamatát. Ahelyett, hogy különálló összetevőket, például Apache-t, MySQL-t, PHP-t és Perl-t kellene telepítenie és konfigurálnia, az XAMPP ezeket egyetlen, könnyen telepíthető csomagba csomagolja. Ez megkönnyíti a felhasználók számára a helyi fejlesztés megkezdését anélkül, hogy a különböző szoftverösszetevők közötti kompatibilitási problémák miatt kellene aggódniuk.

Másik előnye, hogy számos további eszközt és segédprogramot tartalmaz, amelyek hasznosak a webfejlesztéshez. Például tartalmazza a phpMyAdmin programot, amely egy népszerű eszköz a MySQL adatbázisok kezelésére. Az XAMPP számos példaalkalmazást és szkriptet is tartalmaz, amelyek kiindulási pontként használhatók egyéni webalkalmazások fejlesztéséhez.

Egyik legfontosabb jellemzője, hogy lehetővé teszi a fejlesztők számára, hogy webalkalmazásaikat helyi szerverkörnyezeten, úgynevezett "localhost"-on futtassák. Ez azt jelenti, hogy a fejlesztők internetkapcsolat vagy élő webszerver nélkül tesztelhetik alkalmazásaikat. Ez különösen hasznos lehet azoknak a fejlesztőknek, akik olyan összetett alkalmazásokkal dolgoznak, amelyek gyakori tesztelést és hibakeresést igényelnek.

Összefoglalva, a XAMPP egy hatékony eszköz a webfejlesztők számára, amely leegyszerűsíti a webalkalmazások tesztelésére szolgáló helyi szerverkörnyezet beállításának folyamatát. Könnyű kezelhetősége, csomagban lévő összetevői és további eszközei kiváló választássá teszik a munkafolyamat egyszerűsítésére.

A XAMPP Control Panel v3.2.4 ezeket tartalmazza:

- Apache, ami a php futtatásához elengedhetetlen,
- a MySQL Database, ami az oldalak háttérében lévő adatbázist szolgáltatja,

- phpMyAdmin Database Manager 5.0.3, ami az adatbázis létrehozásához és kezeléséhez szükséges,
- FileZilla
- Mercury
- Tomcat

2.1.4. phpMyAdmin v5.0.3

A phpMyAdmin egy ingyenes és nyílt forráskódú eszköz a MySQL és MariaDB adatbázisok kezelésére és adminisztrálására. PHP nyelven íródott, és lehetővé teszi a felhasználók számára, hogy különféle feladatokat hajtsanak végre, beleértve adatbázisok, táblák és mezők létrehozását és szerkesztését, adatok importálását és exportálását, SQL lekérdezések végrehajtását és felhasználói fiókok kezelését.

A phpMyAdmin egyik legfontosabb előnye a web alapú felület, amely bárhol elérhetővé teszi internetkapcsolattal. Ez azt jelenti, hogy a felhasználók bármilyen eszközről kezelhetik adatbázisaikat, legyen szó asztali számítógépről vagy mobiltelefonról.

Az alapvető funkciókon kívül a phpMyAdmin egy sor speciális szolgáltatást is kínál, például tárolt eljárások és triggerek létrehozását és futtatását, a szerver állapotának és változóinak megtekintését, valamint a szerverbeállítások konfigurálását.

A phpMyAdmin másik előnye az aktív felhasználói és fejlesztői közösség, akik támogatást nyújtanak, és folyamatosan frissítik és fejlesztik a szoftvert. Ez biztosítja, hogy a felhasználók mindig hozzáférjenek a legújabb szolgáltatásokhoz és biztonsági frissítésekhez.

Összességében a phpMyAdmin egy hatékony és rugalmas eszköz a MySQL és MariaDB adatbázisok kezelésére. Intuitív kezelőfelülete és átfogó szolgáltatáskészlete kiváló választássá teszi kezdő és tapasztalt felhasználók számára egyaránt.

2.2. Webprogramozás

A webprogramozás a webhelyek, webalkalmazások és egyéb web alapú szoftverek létrehozásának folyamatát írja le. Különféle programozási nyelveket, keretrendszereket és eszközöket foglal magában, amelyek lehetővé teszik a fejlesztők számára, hogy interaktív és dinamikus webhelyeket hozzanak létre, amelyek a világ bármely pontjáról elérhetőek.

A webprogramozás egyik kulcseleme a HTML (Hypertext Markup Language), amely a weboldalak alapvető szerkezetét és tartalmát adja. A CSS (Cascading Style Sheets) egy másik fontos nyelv, amelyet a weboldalak vizuális stílusának és elrendezésének

meghatározására használnak, míg a JavaScriptet interaktív és dinamikus elemek, például animációk, űrlapok és multimédiás tartalmak létrehozására használják.

A webes programozás magában foglalja a különféle keretrendszerek és könyvtárak használatát is, mint például a jQuery, a React és az Angular, amelyek előre elkészített összetevőket és funkciókat biztosítanak a fejlesztőknek, amelyek könnyen integrálhatók webes alkalmazásaikba. Ezek az eszközök felgyorsíthatják a fejlesztési időt, és biztosíthatják, hogy a webhelyek és alkalmazások megbízhatóak, méretezhetőek és hatékonyak legyenek.

A webes programozás másik fontos aspektusa a szerveroldali programozás, amely a szerveren futó kód írását foglalja magában, és kölcsönhatásba lép a felhasználó webböngészőjében futó kliens oldali kóddal. Ez a következő programozási nyelveket foglalhatja magába, például a PHP, a Python vagy a Ruby, valamint a webalkalmazás-keretrendszerek, például a Django, a Flask vagy a Ruby on Rails használatát.

Összességében elmondható, hogy a webes programozás elengedhetetlen készség mindazok számára, akik webhelyeket, webes alkalmazásokat vagy más web alapú szoftvereket szeretnének létrehozni. Szükséges a programozási nyelvek, keretrendszerek és eszközök alapos ismerete, valamint a tervezőkkel, fejlesztőkkel és más érdekelt felekkel való együttműködés képessége a felhasználók igényeinek megfelelő, kiváló minőségű webes élmények létrehozásához.

2.3. Felhasznált technológiák

2.3.1. HTML

A HTML (HyperText Markup Language) egy alapvető technológia, amelyet webhelyek és webes alkalmazások létrehozására használnak. Ez egy jelölőnyelv, amely meghatározza a weboldalak szerkezetét és tartalmát, lehetővé téve a fejlesztők számára, hogy szabványos módon formázzák és jelenítsék meg az információkat a weben. A HTML az internet gerince, és nélküle a modern web nem létezne.

Címkék rendszerét használja az olyan elemek meghatározására, mint a címsorok, bekezdések, képek és hivatkozások. Ezek a címkék szögletes zárójelben vannak, és a címke neve jelzi a definiált elem típusát. Például a <h1> címke egy legfelső szintű címsort, míg a <p> címke egy bekezdést határoz meg. A HTML-címkék attribútumokat is tartalmazhatnak, amelyek további információkat szolgáltatnak egy elemről, például a kép forrásáról vagy a hivatkozás célhelyéről.

Egyik legfontosabb erőssége a rugalmassága. Segítségével statikus weboldalak, dinamikus weboldalak és teljes webalkalmazások hozhatók létre. A fejlesztők a HTML-t más webes technológiákkal, például a CSS-sel (Cascading Style Sheets) és a JavaScript-kóddal együtt használhatják, hogy gazdag, interaktív felhasználói élményt hozzanak létre.

Másik fontos szempontja a hozzáférhetősége. Mivel a HTML szabványosított nyelv, sokféle eszköz és platform által olvasható és értelmezhető, így hatékony eszköz a globális közönség elérésére. A HTML olyan funkciókat is biztosít, mint például az alternatív szöveg és a szemantikai jelölés, amelyek a webes tartalmakat könnyebben hozzáférhetővé teszik a fogyatékkal élők számára.

Összefoglalva, egy alapvető technológia, amely a modern internetet vezérli. Lehetővé teszi a fejlesztők számára, hogy olyan weboldalakat és webalkalmazásokat hozzanak létre, amelyek rugalmasak, hozzáférhetőek, eszközök és platformok széles skálájáról elérhetők. Ahogy az internet folyamatosan fejlődik, a HTML továbbra is a jövő webjének építésének alapvető eszköze marad.

2.3.2. CSS

A Cascading Style Sheets vagy röviden CSS egy alapvető technológia, amelyet a webfejlesztésben használnak. A CSS egy stíluslapnyelv, amellyel meghatározható, hogy a HTML-elemek hogyan jelenjenek meg a weboldalon. Lehetővé teszi a webfejlesztők számára a webhely elrendezésének, színének, betűtípusának és egyéb vizuális szempontjainak szabályozását.

Egyik legfontosabb előnye, hogy elválasztja a weboldal megjelenítését a tartalmától. Ez azt jelenti, hogy a webfejlesztő létrehozhat egy webhelyet tiszta HTML-kóddal, majd CSS-stílusokat alkalmazhat annak érdekében, hogy látványosan nézzen ki. A prezentáció és a tartalom elkülönítése a weboldal egyszerűbb karbantartását és frissítését is lehetővé teszi.

Lépcsőzetes stílushierarchiát használ a stílusok HTML-elemekre történő alkalmazásának meghatározására. Ez a hierarchia lehetővé teszi a fejlesztők számára, hogy stílusokat alkalmazzanak adott elemekre vagy elemcsoportokra, miközben tartalék stílusokat is biztosítanak arra az esetre, ha egy adott stílus nincs meghatározva.

Lehetővé teszi a reszponzív tervezést is, ami azt jelenti, hogy a weboldalak úgy alakíthatók ki, hogy alkalmazkodjanak a különböző képernyőméretekhez és eszközökhöz. Ez médialekérdezéseken és rugalmas elrendezéseken keresztül érhető el, amelyek a képernyő mérete alapján állítják be az elemek méretét és helyzetét.

A vizuális stílusok mellett a CSS segítségével animációkat és interaktív elemeket is létrehozhatunk egy weboldalon. Ez magában foglalhat lebegtetési effektusokat, átmeneteket és akár összetett animációkat is olyan CSS-könyvtárak használatával, mint a CSS-animációk és átmenetek.

Összességében a CSS alapvető technológia a modern, tetszetős webhelyek létrehozásához. Hatékony eszközkészletet biztosít a webfejlesztőknek a tartalom megjelenítésének szabályozásához és vonzó felhasználói élmény megteremtéséhez.

2.3.3. JavaScript

A JavaScript egy magas szintű programozási nyelv, amelyet dinamikus és interaktív weboldalak létrehozására használnak. 1995-ben fejlesztette ki Brendan Eich, és azóta a világ egyik legszélesebb körben használt programozási nyelvéné vált.

Egyik legfontosabb jellemzője, hogy képes kliens és szerver oldalon is futni. Ez azt jelenti, hogy a fejlesztők komplex webalkalmazások létrehozására használhatják, amelyek képesek kommunikálni szerverekkel és adatbázisokkal, valamint valós időben kommunikálni a felhasználókkal.

Rugalmaságáról és sokoldalúságáról is ismert. Alkalmazások széles körének létrehozására használható, a weboldalak funkcionalitását fokozó egyszerű szkriptektől a teljes webhelyeken futó összetett alkalmazásokig. Ez egy rendkívül adaptálható nyelv is, számos keretrendszerrel és könyvtárral segíti a fejlesztőket az alkalmazások hatékonyabb felépítésében.

Az egyik ok, amiért a JavaScript olyan népszerűvé vált, az integrációja a HTML-lel és a CSS-sel, a másik két fő webfejlesztési technológiával. A JavaScript használható a HTML és a CSS manipulálására, lehetővé téve dinamikus és interaktív weboldalak létrehozását, amelyek reagálnak a felhasználói bevitelre és valós időben frissülnek.

Másik előnye a széles körű alkalmazása és a közösségi támogatás. Számos online forrás áll rendelkezésre, beleértve a dokumentációt, az oktatóanyagokat és a fórumokat, amelyek segíthetnek a fejlesztőknek a nyelv elsajátításában és a problémák megoldásában.

Számos előnye ellenére a JavaScriptnek vannak hátrányai is. Nehéz lehet nagy kódbázisokat írni és karbantartani, és gyakran vannak olyan problémák a böngésző kompatibilitásával kapcsolatban, amelyek problémákat okozhatnak a fejlesztőknek.

Összességében a JavaScript egy erőteljes és sokoldalú nyelv, amely elengedhetlenné vált a modern webfejlesztéshez. Dinamikus és interaktív weboldalak létrehozására való képessége a felhasználói élmény fokozásának fontos eszközévé teszi, széles körű

elterjedése és közösségi támogatása pedig vonzó választássá teszi a webalkalmazásokat építeni szándékozó fejlesztők számára.

2.3.4. Bootstrap

A Bootstrap egy széles körben használt nyílt forráskódú keretrendszer a front-end webfejlesztéshez, amelyet Twitter fejlesztők hoztak létre. Előre elkészített HTML-, CSS- és JavaScript-összetevőket kínál, amelyek megkönnyítik a reszponzív és mobil-első webhelyek készítését.

Használatának egyik legfontosabb előnye, hogy időt és erőfeszítést takarít meg a webhely készítése során. A fejlesztők gyorsan létrehozhatnak egy professzionális megjelenésű elrendezést előre megtervezett összetevők, például gombok, űrlapok és navigációs menük segítségével, ahelyett, hogy a nulláról kezdenék. Ezenkívül a Bootstrap reszponzív kialakítása biztosítja, hogy a webhely bármilyen eszközön való megtekintésre optimalizálva legyen, az asztali számítógépektől a mobiltelefonokig.

Másik előnye, hogy jól dokumentált, és a fejlesztők nagy közössége támogatja. Ez azt jelenti, hogy a felhasználók gyorsan választ találhatnak kérdéseikre, és kihasználhatják a legújabb funkciókat és frissítéseket. Ezenfelül a Bootstrap rendszeresen frissítésre kerül, hogy kompatibilis maradjon a legújabb böngészőkkel és eszközökkel.

Egyik lehetséges hátránya azonban az, hogy népszerűsége némileg hasonló megjelenést eredményezhet a webhelyeken. Bár lehetséges az összetevők testreszabása, egyes fejlesztők számára kihívást jelenthet olyan egyedi design létrehozása, amely kiemelkedik a tömegből.

Összefoglalva, a Bootstrap egy hatékony eszköz a webfejlesztők számára, amely számos előre elkészített komponenst és reszponzív kialakítást kínál, amely időt és erőfeszítést takaríthat meg. Népszerűsége és kiterjedt dokumentációja vonzó lehetőséget kínál mind az új, mind a tapasztalt fejlesztők számára, bár szükség lehet némi testreszabásra egy igazán egyedi dizájn létrehozásához.

2.3.5. jQuery

A jQuery egy népszerű JavaScript-könyvtár, amely leegyszerűsíti a HTML-dokumentumok kezelését és az események kezelését. 2006-ban hozta létre John Resig, és gyorsan az egyik legszélesebb körben használt könyvtár lett az interneten. Most megvizsgálunk néhány olyan funkciót, amelyek miatt a jQuery olyan népszerű választás a webfejlesztők számára.

Egyik fő előnye az egyszerűsége. Ahelyett, hogy bőbeszédű JavaScript-kódot írna a DOM-mal való interakcióhoz, a jQuery tömör és intuitív API-t biztosít, amellyel könnyedén manipulálhatók a HTML-elemek. Például egy adott osztályú elem kiválasztása a jQuery-ben olyan egyszerű, mint a `$('.class')` metódus meghívása. Ez az egyszerűség megkönnyíti a fejlesztők számára a kód írását, olvasását és karbantartását.

Másik kulcsfontosságú jellemzője az események kezelésére való képessége. Az eseménykezelés fontos szempont a modern webfejlesztésben, és a jQuery hatékony eseményrendszerrel biztosít, amely felhasználható a felhasználói interakciók figyelésére és reagálására. Segítségével könnyedén hozzárendelheti az eseménykezelőket HTML-elemekhez, és reagálhat olyan eseményekre, mint például a kattintások, a billentyűleütések és az űrlapok beküldése.

Számos hasznos segédfunkciót is biztosít, amelyek segítségével manipulálhatók és bejárhatók a HTML-elemek. Például az egyes függvények lehetővé teszik az elemek gyűjteményének iterációját, és mindegyiken feladat végrehajtását. Hasonlóképpen, az `append` és `prepend` függvények használhatók tartalom hozzáadására egy elemhez.

Végül a jQuery rendkívül bővíthető, élénk plugin-ökoszisztémával, amely további funkciók széles skáláját kínálja. Az űrlapellenőrzéstől a képcsúszkákig mindenhez rendelkezésre állnak beépülő modulok, amelyek megkönnyítik a speciális funkciók hozzáadását weboldalaihoz anélkül, hogy sok egyedi kódot kellene írni.

Összefoglalva, egy erőteljes és sokoldalú JavaScript-könyvtárról van szó, amely a modern webfejlesztés alapelemévé vált. Egyszerűsége, eseménykezelési képességei, segédfunkciói és bővíthetősége kiváló választássá teszi a fejlesztők számára, akik egyszerűsíteni szeretnék kódjukat, és fejlett funkciókat szeretnének hozzáadni weboldalaikhoz. Akár kezdő, akár tapasztalt fejlesztő, a jQuery egy olyan eszköz, amelyet érdemes megtanulni.

2.3.6. Node.js

A Node.js egy hatékony, nyílt forráskódú szerveroldali platform, amely lehetővé teszi a fejlesztők számára, hogy JavaScript használatával rendkívül méretezhető és hatékony alkalmazásokat építsenek. Először 2009-ben mutatták be, és azóta az egyik legnépszerűbb webfejlesztési technológia lett.

Egyik legfontosabb jellemzője a nem blokkoló I/O modell, amely lehetővé teszi nagyszámú egyidejű kapcsolat kezelését a teljesítmény feláldozása nélkül. Ezt egy eseményvezérelt architektúra használatával éri el, ahol a kéréseket aszinkron módon

kezelik, lehetővé téve más kérések feldolgozását, miközben az I/O műveletek befejezésére várnak.

Nagy és élénk fejlesztői közösséggel is rendelkezik, amely hozzájárult a harmadik féltől származó modulok és könyvtárak hatalmas ökoszisztémájának kifejlesztéséhez. Ez lehetővé teszi a fejlesztők számára, hogy gyorsan és egyszerűen funkcionalitással bővítsék alkalmazásaikat anélkül, hogy újra fel kellene találniuk a kereket.

Másik előnye, hogy kompatibilis különféle platformokkal, beleértve a Windowst, a macOS-t és a Linuxot. Ez rendkívül sokoldalú és hozzáférhető platformot jelent a fejlesztők számára.

Ezen kívül nagymértékben méretezhető, lehetővé téve nagy mennyiségű forgalom kezelését teljesítményproblémák nélkül. Ez azáltal érhető el, hogy képes a munkaterhelést több magra osztani, valamint támogatja a horizontal scaling-et a clusterezés használatával.

Összességében a Node.js egy erőteljes és sokoldalú platform rendkívül méretezhető és hatékony alkalmazások létrehozásához. Eseményvezérelt architektúrája, harmadik féltől származó modulok nagy ökoszisztémája és méretezhetősége kiváló választássá teszi a modern és robusztus alkalmazásokat építeni szándékozó webfejlesztők számára.

3. Anyag és módszertan

3.1. Weblap készítés kritériumai

Számos kritériumnak kell megfelelnie egy weboldalnak ahhoz, hogy pozitív felhasználói élményt nyújtson, és hatékonyan tudja elérni a kitűzött célt. Néhány a legfontosabb kritériumok közül:

- **Felhasználóbarát kialakítás:** A weboldal kialakításának könnyen navigálhatónak, tetszetősnek kell lennie, és világos információhierarchiát kell biztosítania.
- **Hozzáférhetőség:** A weboldalnak minden felhasználó számára elérhetőnek kell lenniük, beleértve a fogyatékkal élőket is, követve a web-akadálymentesítési irányelveket, például a WCAG 2.1-et.
- **Betöltési sebesség:** A weboldalak gyorsan, ideális esetben 2-3 másodpercen belül betöltődjenek, hogy a felhasználók ne veszítsék el érdeklődésüket vagy hagyják el az oldalt.
- **Mobilra optimalizálás:** A weboldalakat mobileszközökre kell optimalizálni, mivel egyre több felhasználó éri el az internetet okostelefonon és táblagépen keresztül.
- **Tartalomminőség:** A weboldalnak jó minőségű, releváns, könnyen olvasható és érthető tartalommal kell rendelkezniük. A tartalom legyen jól szervezett és logikus sorrendben kerüljön bemutatásra.
- **Keresőoptimalizálás:** A weboldalakat releváns kulcsszavak, metacímkék és leírások hozzáadásával kell optimalizálni a keresőmotorokhoz.
- **Kompatibilitás:** A weboldalnak kompatibilisnek kell lenniük számos böngészővel és operációs rendszerrel, hogy a lehető legszélesebb közönség számára elérhetők legyenek.

Ezen kritériumok teljesítésével a weboldalak pozitív felhasználói élményt nyújthatnak, bizalmat és hitelességet építhetnek a felhasználók körében, és hatékonyan elérhetik a kitűzött célt.

3.2. Módszer

A honlap frontend részét, Notepad++-ban készítettem el, HTML nyelven írtam, bizonyos részeket, mint például a navbar-t (a felső menüsört) és a carouselt-t (a főoldalon található képes csúszkát) bootstrap-ből importáltam be. A megjelenést továbbá egy css nyelven íródott stylesheet felel.

A backend rész php nyelven készítettem el. Vannak esetek, amikor ezt beleágyaztam a HTML fájlba, viszont van, amikor különálló fájlként található meg.

4. Weblap bemutatása

4.1. Adatbázis

XAMPP egyik beépített funkcióját, a phpMyAdmin-t használom, hogy a localhost-on adatbázist tudjak tárolni. Létrehoztam egyet „szakdolgozat” néven, amely több adattáblát és azokon belül mezőket, rekordokat foglal magába. Ha már szóba jöttek az adatbázisok, akkor tisztázzuk mi is az az adatbázis.

4.1.1. Adatbázis fogalma

Az adatbázis olyan strukturált adatgyűjtemény, amely úgy van megszervezve, hogy lehetővé tegye az információk hatékony tárolását, visszakeresését és kezelését. Ez egy digitális adattár, ahol az adatokat tárolják, és szükség szerint hozzáférhetnek, módosíthatók és lekérdezhetők.

Az adatbázisok mérete a kis, csak néhány rekordot tartalmazó adatbázisoktól a nagy, összetett, több millió rekordot tartalmazó adatbázisokig terjedhet. Sokféle adat tárolására használhatók, beleértve a szöveget, számokat, képeket és multimédiás fájlokat.

Az adatbázisokat számos alkalmazásban használják, beleértve az üzleti, pénzügyi, egészségügyi, oktatási és sok más területet. Használhatók vásárlói adatok, pénzügyi nyilvántartások, készletek, termékkatalógusok és sok más tárolására és kezelésére.

Az adatbázisban tárolt adatok kezelésére adatbázis-kezelő rendszert (DBMS - database management system) használnak. A DBMS egy olyan szoftveralkalmazás, amely lehetővé teszi a felhasználók számára adatbázisok létrehozását, módosítását és kezelését, valamint olyan feladatok elvégzését, mint az adatbevitel, adatvisszakeresés és adatelemzés.

A népszerű DBMS-ek például a MySQL, az Oracle, a Microsoft SQL Server és a MongoDB. Ezek a rendszerek a szolgáltatások és képességek széles skáláját kínálják, és testreszabhatók a különböző felhasználók és alkalmazások speciális igényeinek megfelelően.

4.1.2. Adatbázisban használt változók

Az adatbázisokban a változókat mezőknek vagy oszlopoknak is nevezik. Arra szolgálnak, hogy az adatbázisban minden rekordhoz vagy bejegyzéshez meghatározott adatokat tároljanak. Az adatbázisban lévő változók típusa és száma az adatbázis szerkezetétől és az általa használt céltól függ. Íme néhány gyakori változótípus, amely megtalálható az adatbázisban:

- Szöveg (text) vagy karakterlánc (string): Ez a változó alfanumerikus karakterek, szimbólumok és egyéb karakterek tárolására szolgál szöveg formájában.
- Numerikus (numeric): Ez a változó numerikus értékek, például egész számok, tizedesjegyek vagy lebegőpontos számok tárolására szolgál.
- Dátum/idő (date): Ez a változó a dátum és idő információk tárolására szolgál, mint például egy esemény dátuma, időpontja vagy egy határidő dátuma.
- Logikai érték (boolean): Ez a változó igaz/hamis vagy igen/nem értékek tárolására szolgál.
- Bináris (binary): Ez a változó bináris adatok, például képek, hang- vagy videofájlok tárolására szolgál.
- Objektum (object): Ez a változó összetett adatstruktúrák, például tömbök, struktúrák vagy objektumok tárolására szolgál.
- Null: Ez a változó az érték hiányát vagy az ismeretlen értéket jelzi.

Az adatbázisban lévő változókat az adatmodell határozza meg, amely meghatározza az adatok szerkezetét, kapcsolatait és megszorításait. A változók táblázatokba vannak rendezve, amelyek az adatbázis alapegységei. A táblázat minden sora egyedi rekordot vagy bejegyzést jelöl, minden oszlop pedig egy olyan változót, amely az adott rekordhoz tartozó adatot tárolja.

4.1.3. Szakdolgozat adatbázis

szakdolgozat users id : int(3) username : varchar(50) email : varchar(50) password : varchar(50)	szakdolgozat profiles id : int(3) username : varchar(50) gender : varchar(6) full_name : varchar(50) birth_date : date country : varchar(50) town : varchar(50) postcode : int(4) street : varchar(50) house_number : int(3) phone_number : int(9)	szakdolgozat orders id : int(3) date : date username : varchar(50) email : varchar(50) phone_number : int(9) product : varchar(50) quantity : int(3) card_number : varchar(19) expiration_date : varchar(5) security : int(3) country : varchar(50) town : varchar(50) postcode : int(4) street : varchar(50) house_number : int(3)	szakdolgozat products id : int(2) product : varchar(50) manufacturer : varchar(50) quantity : int(2) description : text price : int(6)
	szakdolgozat genders id : int(2) gender : varchar(50)		szakdolgozat comments id : int(3) date : date name : varchar(50) message : text

1. ÁBRA, SZAKDOLGOZAT ADATBÁZIS (SAJÁT FORRÁS, 2023)

A *szakdolgozat* adatbázist, kifejezetten ehhez a honlaphoz készítettem. Viszonylag egyszerű adatbázisról beszélünk, ezt alátámasztja az is, hogy mindössze 6 adattáblát tartalmaz, és ezekből csak ötöt használ fel végül, mert az egyik adattábla helyett találtam egy sokkal egyszerűbb, hatékonyabb megoldást.

4.1.4. Users adattábla

				id	username	email	password
☐		Módosítás		Másolás		Törölés	1 ADMIN admin@kicsi.hu ADMIN
☐		Módosítás		Másolás		Törölés	2 asd asd@asd.asd asd
☐		Módosítás		Másolás		Törölés	3 SotiJMate sotijozsefmate@gmail.com Krunker.io69
☐		Módosítás		Másolás		Törölés	4 Biggie Cheese biggie@cheese.com cheese
☐		Módosítás		Másolás		Törölés	5 Rick Astley rick@astley.com rickroll
☐		Módosítás		Másolás		Törölés	6 Lajos szia@lajos.hu hány fokos
☐		Módosítás		Másolás		Törölés	7 qwe qwe@qwe.qwe qwe

2. ÁBRA, USERS ADATTÁBLA (SAJÁT FORRÁS, 2023)

Ez az adattábla a *users*, avagy felhasználók tárolására szolgál, viszonylag kicsi tábláról beszélünk ugyanis összességében 4 mezőt tartalmaz.

Sorban a legelső az *id* avagy az azonosító. Ez a tábla elsődleges kulcsa, ebből kifolyóan ez unique (egyedi) minden egyes rekordnál. A harmadik felhasználó esetében az azonosító értéke 3. Ez egy integer, ami az egész szám megfelelője és maximálisan 3 karakter hosszú lehet, ami azt jelenti, hogy ezen adattáblában összesen 999 felhasználót lehet elmenteni. Ezen felül auto_increment-es (automatikusan növekvő) ez a mező, ami azt jelenti, hogy amikor új rekordot veszünk fel ebbe a táblába, akkor automatikusan saját magának határozza meg ennek a mezőnek az értékét és minden esetben eggyel nagyobb elődjénél. A negyedik felhasználó esetében ezért lett 4 az azonosító értéke, mert az eggyel nagyobb, mint a második felhasználó azonosítója, aminek az értéke 3 volt.

A második mező a *username*, ez a felhasználó felhasználónevét rögzíti. Ez egy string, ami egy karakterlánc vagy egy olyan lista, amely karaktereket tartalmaz. Ezen esetben az adattáblán belül varchar-nak nevezik, de ugyanazt a szerepet tölti be, mint egy string. Maximális karakter számnak 50-et határoztam meg, azt megfelelő mennyiségnek találtam. Eddigi tapasztalataim alapján, más honlapokon, alacsonyabb értéket szoktak meghatározni, 20 körüli karakterszámot, viszont azokon a honlapokon sokkal több felhasználó található, mint az enyém esetében, ezért is döntöttem, úgy hogy megengedhetem a hosszabb felhasználóneveket. Harmadik felhasználónk esetében a felhasználónév: *SotiJMate*.

A felhasználók adattábla harmadik mezője ismételt egy string, ami pedig nem más, mint az *email*. Ezen oszlopban a felhasználók email címét tárolom, ezen esetben is 50 karaktert határoztam meg limitnek. Eddig vizsgált felhasználónk esetében ez a karakterlánc: *sotijozsefmate@gmail.com*.

Negyedik mező pedig a *password*, a felhasználó jelszava, ha ez egy élesben működő honlap lenne, nem csak egy bemutató, teszt változat, akkor ezt a mezőt titkosítani kellene, hogy ténylegesen csak a felhasználó tudhassa. Viszont ezen esetben erre nem volt szükség, hiszen nem valós adatokkal van feltöltve egyik tábla sem. Harmadik felhasználónk esetében a jelszó karakterlánc: *Kruner.io69*.

4.1.5. Profiles adattábla

				id	username	gender	full_name	birth_date	country	town	postcode	street	house_number	phone_number			
		Módosítás		Másolás		Törlés	1	ADMIN	other	ADMIN	2023-04-17	Magyarország	-	0	-	0	0
		Módosítás		Másolás		Törlés	2	asd	male	Asd Asd	1969-06-09	Asdország	Asdasd	420	Asd utca	69	694206969
		Módosítás		Másolás		Törlés	3	SotiJMate	male	Sóti József Máté	2000-11-17	Magyarország	Jászberény	5100	Medve utca	15	209548594
		Módosítás		Másolás		Törlés	4	Biggie Cheese	-	-	2000-01-01	-	-	0	-	0	0
		Módosítás		Másolás		Törlés	5	Rick Astley	-	-	2000-01-01	-	-	0	-	0	0
		Módosítás		Másolás		Törlés	6	Lajos	-	-	2000-01-01	-	-	0	-	0	0
		Módosítás		Másolás		Törlés	7	qwe	-	-	2000-01-01	-	-	0	-	0	0

3. ÁBRA, PROFILES ADATTÁBLA (SAJÁT FORRÁS, 2023)

Második adattáblánk a *profiles*, magyarul profilok tábla, ahol a felhasználók tárolhatják személyes adataikat. Ennek használata nem kötelező, csak beleraktam extra funkciónak. A felhasználók adattáblához hasonlóan itt is átnézzük egyesével a mezőket, és példának a harmadik felhasználót vesszük majd elő.

Első mezőnk ismét az *id*, mint minden esetben, ez a felhasználó azonosítója, a táblában elsődleges kulcs, és a felhasználók táblához idegen kulcs. Ez a mező megint *auto_increment*-tel van felruházva, tehát magától növeli az előzőhöz képest eggyel, ha új sort mentünk el. Harmadik felhasználónk esetében, ennek a mezőnek az értéke 3, ezt egyfajta indexként is lehet értelmezni.

Második oszlop a *username*-t, a felhasználó felhasználónevét tárolja, ez kimondottan a szerkeszthetőség miatt került bele. Ismét egy 50 karakterszámos limit lett beállítva erre a *varchar*-ra is. Harmadik sorban a felhasználónév: *SotiJMate*.

Harmadik mezőben, a *gender* oszlopban a felhasználó nemét tárolja az adatbázis. Ez egy rövidebb string, mert csak négy opció közül lehet válogatni, amelyek: --, férfi, nő, egyéb. Viszont ezek csak a megjelenítendő szövegek, a háttérben angolul dolgozik a program. Így a választható értékek: --, *male*, *female*, *other*. Ezek közül a leghosszabb a *female* így a maximális karakterszám nem több mint 6 karakter. Harmadik felhasználónk neme férfi, ezért a táblában *male* értéket találunk.

Negyedik oszlopban *full_name*, avagy a felhasználó teljes neve található. Ez ismét egy 50 karakter limités varchar, amely tartalmazhat egyszerű betűktől kezdve, speciális karaktereken keresztül akár még számokat is. Hármas indexű felhasználónak a teljes neve: *Sóti József Máté*.

Következő oszlopunk a *birth_date*, avagy a felhasználó születési dátuma. Ez date formátumban van, tehát ez egy dátum. [éééé-hh-nn] formátumban tárolja a születési dátumot, ezen esetben nem lehetett limitet szabni a karakterszámnak. Megszokottan a harmadik felhasználónak vizsgáljuk meg az értékét, ami ez esetben *2000-11-17*.

Hatodik oszlopunk a *country* mező, ami a felhasználó származási országát tárolja. Ismételttem 50-es limittel ellátott varchar-ról beszélünk. A szóban forgó felhasználó származási országa: *Magyarország*. Először az országoknak terveztem készíteni egy külön adattáblát, viszont csak magyar felhasználók vannak, és csak őket kínoznám, a legördülő listával, így végül úgy döntöttem, hogy egy sima textbox-ot használok és mindenki beírja magának.

Ezután vizsgáljuk meg a *town* mezőt, amely annak a városnak a nevét tárolja, ahol a felhasználó jelenleg lakik. Ismételttem varchar, aminek a karakterszáma le lett limitálva 50-re. Hármas sorban található felhasználónak ez az értéke: *Jászberény*.

A városhoz természetesen irányítószám is tartozik, amelyet a *postcode* nevű mező tartalmaz. Itt 4 karakteres integereket lehet tárolni, avagy egész számokat. Harmadik felhasználó lakcímének irányítószáma: *5100*.

A kilencedik oszlop, *street*, az útnak, utcának vagy térnek a nevét tárolja, ahol a felhasználó lakik. Ismételttem varchar-ként tárolja ezt és a karakterek számának limitje 50. Harmadik felhasználónk a *Medve utcában* lakik.

Természetesen a házszámot is rögzíti ez a tábla, *house_number* néven. Itt 3 karakteres integereket lehet tárolni, ami azt jelenti, hogy a legnagyobb tárolható házszám 999. Tudomásom szerint nem létezik 4 jegyű házszám, kivéve persze a /A, /B és hasonlókat, de azokkal most nem foglalkozunk. Harmadik felhasználónk a *15-ös* számú házban lakik.

Végezetül a telefonszám rögzítésére is van lehetőség, a *phone_number* mezőben. Ismételttem integert tárol, viszont ennek a maximális hossza 9 karakter. A telefonszám elején található 06-ot vagy a +36-ot nem kell leírni az úgymint egységes mindenkinek. Magát a telefonszámot, pedig szóköz nélkül egyben kell leírni, hogy beleférjen a limitbe. Harmadik felhasználó telefonszáma +36(20)954-8493 ezért a táblában: *209548493* szerepel.

4.1.6. Genders adattábla

				id	gender
☐	 Módosítás	 Másolás	 Törlés	1	Abinary
☐	 Módosítás	 Másolás	 Törlés	2	Agender
☐	 Módosítás	 Másolás	 Törlés	3	Ambigender
☐	 Módosítás	 Másolás	 Törlés	4	Androgyne
☐	 Módosítás	 Másolás	 Törlés	5	Androgynos
☐	 Módosítás	 Másolás	 Törlés	6	Androgynous
☐	 Módosítás	 Másolás	 Törlés	7	Aporagender
☐	 Módosítás	 Másolás	 Törlés	8	Autigender
☐	 Módosítás	 Másolás	 Törlés	9	Bakla
☐	 Módosítás	 Másolás	 Törlés	10	Bigender
☐	 Módosítás	 Másolás	 Törlés	11	Binary
☐	 Módosítás	 Másolás	 Törlés	12	Bissu

4. ÁBRA, GENDERS ADATTÁBLA (SAJÁT FORRÁS, 2023)

A *genders* tábla mindössze 2 mezőből áll, amelyből az egyik az az *id*, avagy azonosító, a másik pedig a hozzátartozó érték. Első oszlop egy 2 karakter hosszú integer, amely csakis indexként szolgál.

A második oszlop az maga a nem (*gender*), ez egy varchar amely 50 karaktert engedélyez maximálisan. Első ötletem az lett volna, hogy felsorolom a létező összes nemet, hiszen 2023 van, és berakom egy legördülő listába majd, de elvettem az ötletet, mert túl sok munkával járt volna és nem sok értelmét láttam. Így egy szimpla férfi, nő, egyéb listát készítettem, egy sokkal egyszerűbb módszerrel, amelyet a későbbiekben részletezek majd. Mivel, elég sokáig tartott feltölteni ezt a táblát és a későbbiekben még akár jól is jöhet, ezért meghagytam de a weblap nem használja semmilyen formában sem.

4.1.7. Products adattábla

← →			id	product	manufacturer	quantity	description	price				
		Módosítás		Másolás		Törölés	1	Darts nyíl	ismeretlen	5	Egyszerű darts nyíl, semmi extrát nem tud, de arra...	4990
		Módosítás		Másolás		Törölés	2	Prémium Darts nyíl	BA Szanyó Kft.	3	Prémium minőségű darts nyíl, precíziós anyagokból ...	8990
		Módosítás		Másolás		Törölés	3	Darts tábla	ismeretlen	1	Elengedhetetlen, minden háztartásban kötelező fels...	19990

5. ÁBRA, PRODUCTS ADATTÁBLA (SAJÁT FORRÁS, 2023)

Készítettem egy webshopot is a honlapomra, amihez természetesen szükség volt egy újabb adattáblára. Ebből lett a *products* tábla, amely a termékeket tárolja. Mindössze 6 mezőt tartalmaz, ezek rendre: *id*, *product*, *manufacturer*, *quantity*, *description*, *price*.

Az *id* mező ismételt auto_increment és elsődleges kulcs, ezen a táblán belül, tehát ez alapján egyértelműen meg lehet határozni minden egyes rekordot a táblán belül. Maximális hossza az azonosítónak 2 karakter, tehát maximum 99 terméket lehet tárolni ebben az adatbázisban.

A *product* mező, a vásárlásra kívánt termék nevét tárolja varcharban és megszokottan itt is 50 a karakterek maximális száma.

A *manufacturer* mező tulajdonságai teljes mértékben megegyeznek a *product* mezőével, tehát ugyanúgy varchar(50), csak ezen esetben a termék neve helyett, a termék gyártóját tárolja.

A *quantity* mező a termék mennyiségét tárolja darabban kifejezve egy csomagon belül. Ezt egy maximálisan 2 karakter hosszúságú integerben tárolja, tehát egy csomagban maximálisan 99 termék lehet. Persze erre nem kerül sor, de a lehetőséget fenn tartom ilyen esetekre is.

A *description* mezőben egy gyors leírást lehet írni a termékről. Ezt text-ként tárolja az adattábla, ezzel megadva a lehetőséget a hosszabb szövegeknek is, így nincs lekorlátozva 50-re a karakterszám.

A *price* mezőben a termék árát tárolhatjuk, egy maximum 6 karakter hosszú integerként. Nem tudom pontosan, milyen árkategóriában mozognak az ilyen termékek, ezért egy százezer forintos kategóriát adtam meg határnak. Lehet, hogy léteznek milliós tételek is ezen a témán belül, de azokat majd nem lehet megvenni ezen a honlapon keresztül.

4.1.8. Orders adattábla

				id	date	username	email	phone_number	product	quantity	card_number	expiration_date	security	country	town	postcode	street	house_number		
☐		Módosítás		Másolás		Törölés	1	2023-04-29	asd	asd@asd.asd	201234567	Darts nyíl	1	1234 1234 1234 1234 01/02	123	Magyarország	Gyöngyös	3200	Mátrai út	36
☐		Módosítás		Másolás		Törölés	2	2023-04-29	SotiMate	sotijozsefmate@gmail.com	209548493	Prémium Darts nyíl	1	1234 5678 8765 4321 11/22	369	Magyarország	Jászberény	5100	Méde utca	15

6. ÁBRA, ORDERS ADATTÁBLA (SAJÁT FORRÁS, 2023)

Ha vannak termékek, akkor logikusan kell léteznie *orders*, avagy rendelések adattáblának is. Ez az egyik legnagyobb adattábla az egész adatbázisban, összesen 13 mezőt tartalmaz. Sorrendben ezek a mezők találhatóak meg ebben a táblában: *id*, *date*, *username*, *email*, *phone_number*, *product*, *quantity*, *card_number*, *expiration_date*, *security*, *country*, *town*, *postcode*, *street*, *house_number*.

Az *id* mező megszokottan a tábla elsődleges kulcsa, maximum 3 karakter hosszú integer, azaz egész szám, ami azt jelenti, hogy összesen 999 rendeltet tud tárolni az adatbázis, *auto_increment*-es tehát magától növeli mindig egyel nagyobbra, mint az előző.

A *date* oszlop a megrendelés idejét rögzíti dátumként, melynek formátuma [éééé-hh-nn]. Ennek értékét is saját magának határozza meg a weblap.

A *username* mező a megrendelő felhasználónevét vagy, ha nem regisztrált felhasználó szeretne rendelni, akkor neki külön kell megadnia a nevét. Ezt egy maximum 50 karakteres *varchar*-ként menti le ebbe a mezőbe.

Az *email* mező a megrendelő email címét tárolja, azon célból, hogy a kiszállítás során informálni lehessen a megrendelőt a csomagja jelenlegi státuszáról és esetleg helyéről is. Ezt az információt maximum 50 karakter hosszúságú *varchar*-ként tárolhatjuk ezen mezőben.

A *phone_number* mező a megrendelő telefonszámát tárolja 06 vagy +36 nélkül, mert az úgyszintén ugyanaz minden esetben. Ezt egy 9 számjegyű *string*-ként menti el. Azért szükséges a telefonszámot megadni, mert bármi közbejöhethet a kiszállítás alatt és erről egyből értesíteni tudják a megrendelőt. A *profiles* adattáblában már egyszer szerepel a telefonszám mező, de az csak a regisztrált felhasználókra vonatkozik, viszont rendelést bárki adhat fel, nem szükséges hozzá regisztrálni.

A *product* mező a megrendelendő termék nevét tartalmazza egy 50-es karakterlimitű *varchar*-ként.

A *quantity* mező a termék darabszáma vagy csomagszáma. Ezt egy integerként rögzíti és 3 számjegyű lehet ez az érték maximum tehát 1-999 közötti számokat képes tárolni.

A *card_number* mező a megrendelésnél a fizetéshez szükséges, ez a megrendelő bankkártyájának a számát rögzíti egy 19 karakter hosszú *varchar*-ként, hiszen a bankkártya száma 16 karakter és négyes egységekre van bontva szóközzel.

Az *expiration_date* mező a bankkártyán szereplő lejáratidő tárolására szolgál. Ezt egy 5 karakter hosszú *string*-ként menti el.

A *security* mező a bankkártya hátoldalán található biztonsági kód tárolására szolgál, amelyet 3 számjegyű integer-ként tárol.

A *country* oszlop a kiszállítási cím országát tárolja maximum 50 karakter hosszú varchar-ként. Ezt a *profiles* adattábla is rögzíti, viszont nem csak a regisztrált felhasználók rendelhetnek, ezért szükség volt itt külön rögzíteni. Ráadásul fenn kell hagyni annak is a lehetőségét, hogy nem a lakcímére kívánja a kiszállítást a megrendelő.

A *town* mező a kiszállítási címből a város nevét tárolja varchar-ként, melynek a maximális hossza 50 karakter.

A *postcode* mező a kiszállítási cím városának az irányítószámát tárolja 4 karakter hosszúságú integer-ként.

A *street* mező az utcának a nevét tárolja, ahova a kiszállítást kérte a megrendelő. Ezt ismételten maximum 50 karakter hosszú karakterláncként rögzíti.

A *house_number* a kiszállítási címnek megadott ingatlan ház számát tárolja legfeljebb 3 számjegyű integer-ként.

4.1.9. Comments adattábla

							id	date	name	message
☐		Módosítás		Másolás		Törlés	1	2023-04-29	ADMIN	Minden felhasználót szeretettel üdvözlünk egyesüle...
☐		Módosítás		Másolás		Törlés	2	2023-04-29	asd	asd asd asd Csak teszteltem a funkció működését ...
☐		Módosítás		Másolás		Törlés	3	2023-04-29	Rick Astley	We're no strangers to love, You know the rules a...

7. ÁBRA, COMMENTS ADATTÁBLA (SAJÁT FORRÁS, 2023)

Alapból terveztem készíteni egy blog fület, viszont átgondoltam és több értelmet láttam egy kommentek fülnek. Így született meg a *comments* adattábla, amely 4 oszlopból áll mindössze. Ezek az oszlopok az *id*, a *date*, a *name* és a *message* mezők.

Az *id* mező ismételten a táblán belüli elsődleges kulcs, tehát ez alapján azonosíthatjuk be az egyes kommenteket. Auto_increment-es tehát saját magának határozza meg ezen oszlop értékét és mindig eggyel növeli az előző rekordhoz képest. Ezt az azonosítót egy szimpla integerként tárolja, amelynek a maximális hossza 3 karakter, tehát 999 kommentet lehet tárolni az adatbázisban.

A következő *date* mező a komment megírásának idejét tárolja. Ennek van külön adattípusa, ami ugyanúgy date és a formátuma [éééé-hh-nn]. Ezt automatikusan a komment megírása után a küldés gombra való kattintáskor meghatározza és el is menti magának a honlap.

A *name* mező a kommentelő nevét tárolja, ha regisztrált felhasználóról van szó, akkor ezt automatikusan kitölti a felhasználó felhasználónevével, de persze meghagyja a lehetőséget

ennek módosítására, ha valaki nem szeretné felvállalni, amit ír. Ha nem regisztrált felhasználóról van szó, akkor neki manuálisan kell megadnia a megjelenítendő nevet. Ez ismételt maximum 50 karakter hosszú varchar-ként tárolja.

A *message* mező, magát a kommentet tárolja text-ként, így nincs megszabva ebben az oszlopban a karaktereknek maximális darabszám, ez lehetővé teszi hosszabb, összetettebb kommentek írását és azoknak rögzítését.

4.2. Frontend

4.2.1. Mi is az a frontend

A webes programozásban a frontend kifejezés a webalkalmazás azon részére utal, amellyel a felhasználó közvetlenül kapcsolatba lép. A webalkalmazás kliens oldalának is nevezik. A frontend tartalmazza a felhasználói felületet (UI) és a felhasználói élmény (UX) kialakítását, valamint az alkalmazás vizuális és interaktív elemeit vezérlő kódot.

A frontend webes programozásban használt egyes technológiák közé tartozik a HTML, a CSS és a JavaScript. A HTML (Hypertext Markup Language) a weboldal tartalmának strukturálására és az alapvető elrendezés meghatározására szolgál. A CSS-t (Cascading Style Sheets) használják a vizuális elemek (például színek, betűtípusok és térközök) stílusozására és tervezésére. A JavaScript segítségével interaktivitást és funkcionalitást adnak a weboldalhoz, például űrlapellenőrzést, animációkat és dinamikus frissítéseket.

4.2.2. Index.html



Főoldal Búszkeségeink Kapcsolatok Kommentek Webshop Bejelentkezés Regisztráció

8. ÁBRA, NAVBAR (SAJÁT FORRÁS, 2023)

```

14 <nav class="navbar navbar-expand-lg navbar-dark text-white bg-dark position-fixed">
15 <div class="container-fluid">
16 <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
17 </button>
18 <div class="collapse navbar-collapse" id="navbarSupportedContent">
19 <ul class="nav navbar-nav">
20 <li class="nav-item">
21 <a class="nav-link" href="#">Főoldal</a>
22 </li>
23 <li class="nav-item dropdown">
24 <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-bs-toggle="dropdown" aria-expanded="false">Büszkeségeink</a>
25 <ul class="dropdown-menu">
26 <li class="dropdown-item">Aladár</li>
27 <li class="dropdown-item">Csaba</li>
28 <li class="dropdown-item">Dömötör</li>
29 </ul>
30 </li>
31 <li class="nav-item">
32 <a class="nav-link" href="index.php?oldal=contacts">Kapcsolatok</a>
33 </li>
34 <li class="nav-item">
35 <a class="nav-link" href="index.php?oldal=comments">Kommentek</a>
36 </li>
37 <li class="nav-item">
38 <a class="nav-link" href="index.php?oldal=webshop">Webshop</a>
39 </li>
40 <li class="nav-item">
41 <a class="nav-link" href="#">Profil</a>
42 </li>
43 <li class="nav-item">
44 <a class="nav-link" href="#">Kijelentkezés</a>
45 </li>
46 </ul>
47 </div>
48 </nav>

```

9. ÁBRA, NAVBAR MÖGÖTTI KÓD (SAJÁT FORRÁS, 2023)

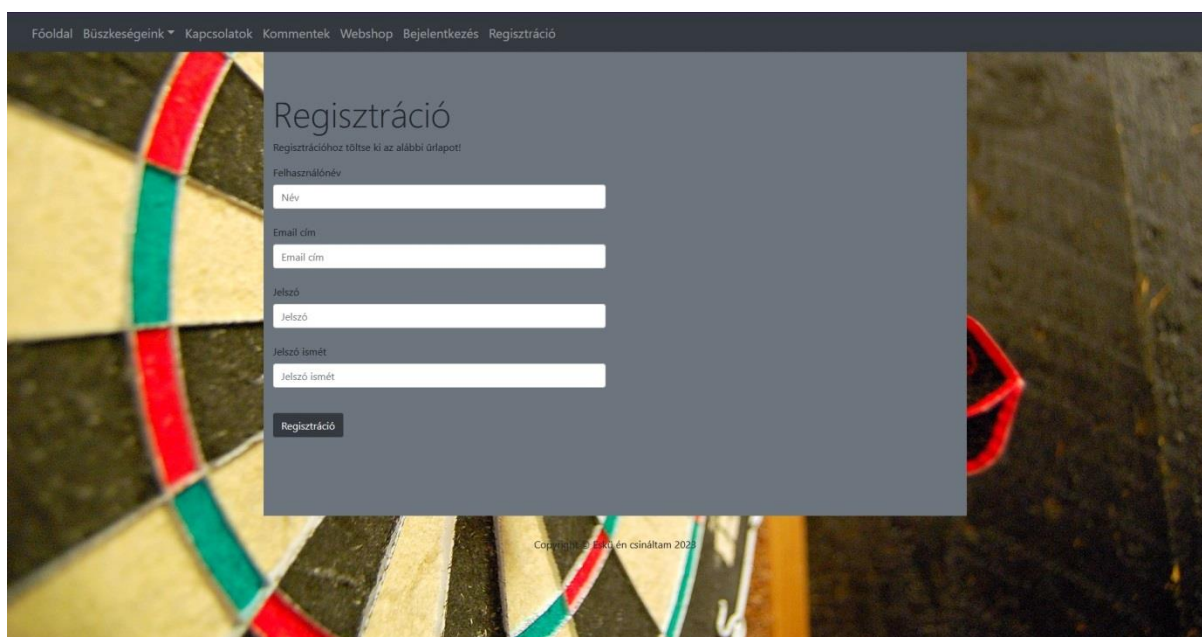
Az index.html természetesen az egész honlap kezdőpontja, mint ahogy annak kell lennie. Ebben az esetben csak egy keretet szolgált a honlap számára. Ez azt jelenti, hogy csak azokat a részeket tartalmazza, amelyek minden fülnél megegyeznek, ez a navbar-t (felső menüsört) és a footer-t (lábazatot) jelenti.

A navbar-ban található egy kis beágyazott php kód is, ami azt a célt szolgálja, hogy a regisztrált felhasználó bejelentkezés után láthassa a kifejezetten nekik szóló menüpontokat is (például: profil adatok szerkesztése, kijelentkezés), abban az esetben, ha nincs bejelentkezve senki, akkor pedig a bejelentkezés és regisztráció gombok lesznek elérhetőek. Továbbá az ADMIN felhasználó számára az admin felület gombja is hasonló a módszerrel készült el, nem készítettem külön admin felületet, hiszen a phpMyAdmin kézhez állt, viszont a honlapról egy gombnyomással elérhetővé tettem, amelyre ha ráklikkelünk, külön lapon megnyitja azt az ADMIN felhasználó számára. A menüsor legelső pontja a Főoldal, ami az alapértelmezett, tehát a honlap megnyitásakor ez töltődik be. A következő pont, a Büszkeségeink, egy lenyitható menü, avagy klikkeléskor nem egy másik lapot tölt be, szimplán négy másik lehetőséget ajánl fel. Ezt követő pont a Kapcsolatok, ahol az egyesület kapcsolatait tekinthetjük meg, emellett kommentet írhatunk, ami a következő menüpontban, a Kommentek lapon meg is jelenik. Ötödik menüpont egy webshop, ahol megtekinthetjük a termékeket és a jövőben majd akár meg is rendelhetjük őket. Ha be vagyunk jelentkezve akkor találhatunk még kettő extra menüpontot, amelyek a Profil és Kijelentkezés. A Profil oldalon a személyes adatait

szerkesztheti a felhasználó, a Kijelentkezés ponttal pedig kijelentkezhet a jelenlegi felhasználóból.

A fejléc is ebben a fájlban található, bár ez láthatatlan a felhasználó számára, kifejezetten a keresőmotorok kisegítésére szolgál. Ezen felül itt kerül meghívásra a css fájl is, ami a megjelenést határozza meg és a beimportált scriptek is ebben a fájlban kerülnek meghívásra. A tartalom rész üresen lett hagyva, mert azt egy php fájl határozza meg, hogy melyik html fájl lesz betöltve az index.html keretbe.

4.2.3. Regist.html

The image shows a web browser window displaying a registration page titled "Regisztráció". The page has a dark header with navigation links: "Főoldal", "Büszkeségeink", "Kapcsolatok", "Komentek", "Webshop", "Bejelentkezés", and "Regisztráció". The main content area is a light gray box with the title "Regisztráció" and a subtitle "Regisztrációhoz töltsd ki az alábbi űrlapot!". Below the subtitle are four input fields: "Felhasználónév" (Name), "Email cím" (Email), "Jelszó" (Password), and "Jelszó ismét" (Password confirmation). Each field has a placeholder text. At the bottom of the form is a "Regisztráció" button. The background of the page is a blurred image of a dartboard.

10. ÁBRA, REGISZTRÁCIÓS FELÜLET (SAJÁT FORRÁS, 2023)

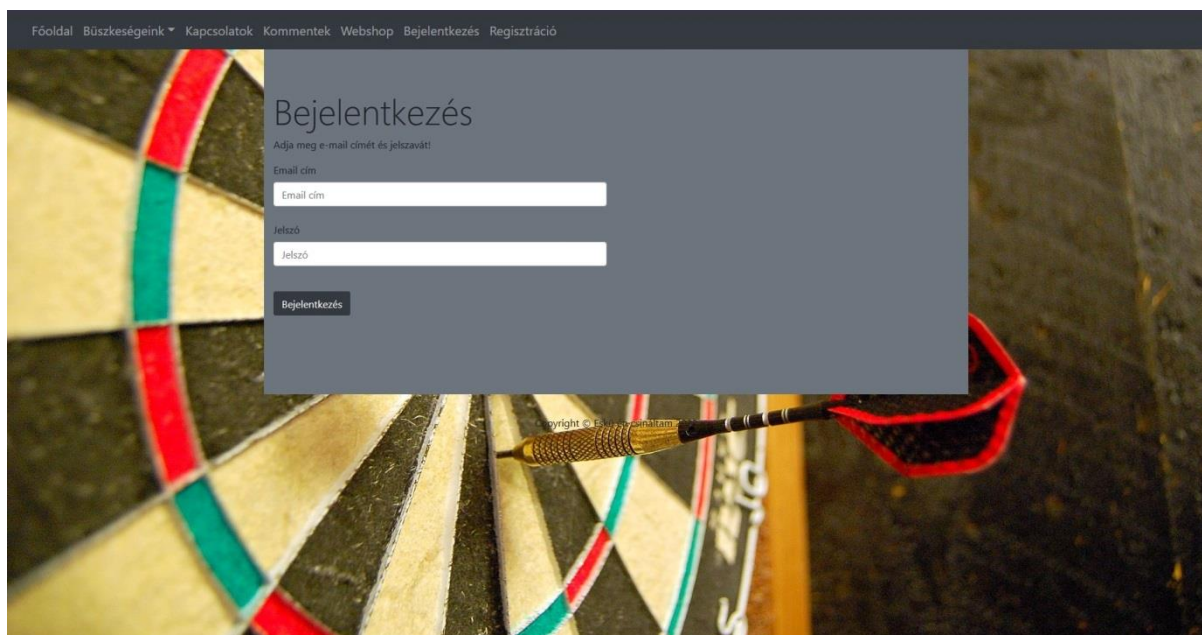
```
1 <div class = "jumbotron-jumbotron-fluid bg-secondary">
2 <div class = "container-fluid wrapper">
3 <h1 class = "display-4"> Regisztráció </h1>
4 <div class = "row">
5 <div class = "col-12">
6 <p>Regisztrációhoz töltsd ki az alábbi űrlapot!</p>
7 <form action = "index.php?oldal=regment" method = "POST">
8 <div class = "form-group">
9 <label for = "username">Felhasználónév</label>
10 <input type = "text" class = "form-control" id = "username" name = "username" required placeholder = "Név"><br>
11 <label for = "email">Email cím</label>
12 <input type = "email" class = "form-control" id = "email" name = "email" aria-describedby = "emailHelp" required placeholder = "Email cím"><br>
13 <label for = "password">Jelszó</label>
14 <input type = "password" class = "form-control" id = "password" name = "password" required placeholder = "Jelszó"><br>
15 <label for = "password2">Jelszó ismét</label>
16 <input type = "password" class = "form-control" id = "password2" name = "password2" required placeholder = "Jelszó ismét"><br>
17 </div>
18 <button type = "submit" class = "btn btn-dark">Regisztráció</button>
19 </form>
20 </div>
21 </div>
22 </div>
23 </div>
24
```

11. ÁBRA, REGISZTRÁCIÓS FELÜLET MÖGÖTTI KÓD (SAJÁT FORRÁS, 2023)

A regist.html csakis akkor elérhető, ha a felhasználó nincs bejelentkezve és ez nem más, mint egy regisztrációs űrlap. A cím alatt négy darab szövegdozoz található, amelyből három speciális. A legelső szimplán egy felhasználónevet igényel, és csakis annyit ellenőriz, hogy ne legyen üres. A következő egy email címet vár és meg is vizsgálja, hogy valójában egy email címet adtunk-e meg neki vagy csak valami karaktersort. Ez azt jelenti,

hogy leellenőrzi tartalmaz-e '@' karaktert és a végén található-e '.' karakter és utána még legalább egy darab betű. A következő kettő szövegdoz password típusú, tehát jelszavakat kezelnek, ezt onnan tudhatjuk, hogy a beírt karakterek helyett csak pöttyöket jelenít meg. Mindegyik textbox kitöltése kötelező, és csakis akkor tudunk lépni, ha az előbb említett feltételeknek megfelelünk. Az oldal alján található még egy Regisztráció gomb is, aminek a szerepe, hogy meghívja a regment.php fájlt és átadja neki a szövegdozba írt karakterláncokat.

4.2.4. Login.html



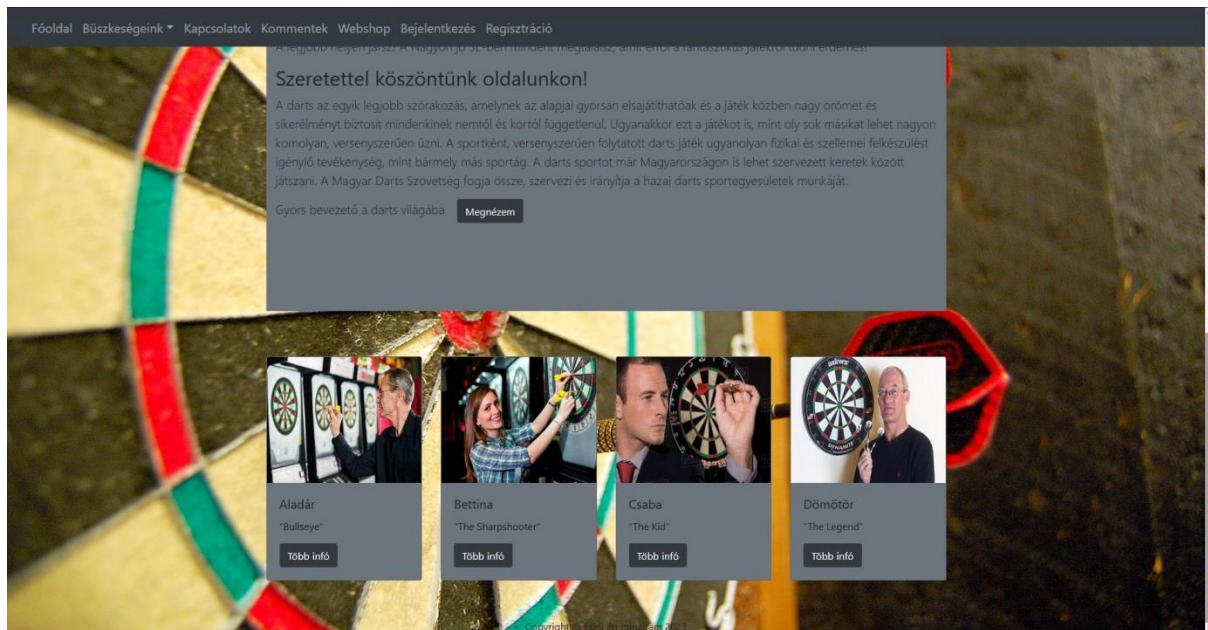
12. ÁBRA, BEJELENTKEZÉSI FELÜLET (SAJÁT FORRÁS, 2023)

```
1 <div class = "jumbotron-jumbotron-fluid bg-secondary">
2 <div class = "container-fluid wrapper">
3 <h1 class = "display-4"> Bejelentkezés </h1>
4 <div class = "row">
5 <div class = "col-lg-6">
6 <p>Adja meg e-mail címét és jelszavát!</p>
7 <form action = "index.php?oldal=loginvalid" method = "POST">
8 <div class = "form-group">
9 <label for = "exampleInputEmail">Email cím</label>
10 <input type = "email" class = "form-control" id = "exampleInputEmail" name = "email" aria-describedby = "emailHelp" required placeholder = "Email cím"><br>
11 <label for = "exampleInputPassword">Jelszó</label>
12 <input type = "password" class = "form-control" id = "exampleInputPassword" name = "password" required placeholder = "Jelszó"><br>
13 </div>
14 <button type = "submit" class = "btn btn-dark">Bejelentkezés</button>
15 </div>
16 </div>
17 </div>
18 </div>
19 </div>
```

13. ÁBRA, BEJELENTKEZÉSI FELÜLET MÖGÖTTI KÓD (SAJÁT FORRÁS, 2023)

A login.html nagyon sokban hasonlít a regist.html-hez és az elmélete körülbelül meg is egyezik. A regisztrációs űrlappal ellentétben itt csak kettő szövegdozot találunk. Az felsőbe az email címet kell megadni, amit ugyanúgy levizsgál, mint regisztráció esetén a másodikba pedig a felhasználó jelszavát kell megadni, amit ugyanúgy nem jelenít meg csak azonos számú pöttyöt jelenít meg. A lap alján található Bejelentkezés gombbal a login.php-t hívjuk meg és adjuk át neki a szövegdozok tartalmát.

4.2.5. Home.html



14. ÁBRA, FŐOLDAL (SAJÁT FORRÁS, 2023)

```
1 <!-- Carousel wrapper -->
2 <div class="container-fluid wrapper">
3   <div id="#carouselExampleIndicators" class="carousel slide" data-bs-ride="carousel">
4     <ol class="carousel-indicators">
5       <li data-bs-target="#carouselExampleIndicators" data-bs-slide-to="0" class="active"/></li>
6       <li data-bs-target="#carouselExampleIndicators" data-bs-slide-to="1"/></li>
7       <li data-bs-target="#carouselExampleIndicators" data-bs-slide-to="2"/></li>
8     </ol>
9     <div class="carousel-inner">
10      <div class="carousel-item active">
11        
12      </div>
13      <div class="carousel-item">
14        
15      </div>
16      <div class="carousel-item">
17        
18      </div>
19    </div>
20    <div class="carousel-control-prev" href="#carouselExampleIndicators" role="button" data-bs-slide="prev">
21      <span class="carousel-control-prev-icon" aria-hidden="true"/></span>
22      <span class="visually-hidden" Previous />
23    </div>
24    <div class="carousel-control-next" href="#carouselExampleIndicators" role="button" data-bs-slide="next">
25      <span class="carousel-control-next-icon" aria-hidden="true"/></span>
26      <span class="visually-hidden" Next />
27    </div>
28  </div>
29 <!-- Content -->
30 <div class="jumbotron jumbotron-fluid by-secondary">
31   <div class="container-fluid wrapper">
32     <ol class="display-4">Magyar jó darts sport egyesület/</ol>
33     <li>Munkatársak/</li>
34     <li>Magyar darts sport/</li>
35     <li>Magyar darts sport/</li>
36     <li>Magyar darts sport/</li>
37     <li>Magyar darts sport/</li>
38     <li>Magyar darts sport/</li>
39     <li>Magyar darts sport/</li>
40     <li>Magyar darts sport/</li>
41   </div>
42 </div>
43 <!-- Content -->
44 <div class="row">
45   <div class="col-md-6 col-sm-4 col-3">
46     <div class="card">
47       
48       <div class="card-body">
49         <h4>Aladár Károly</h4>
50         <p>Magyar darts sport/</p>
51         <a href="#">Több info</a>
52       </div>
53     </div>
54   <div class="col-md-6 col-sm-4 col-3">
55     <div class="card">
56       
57       <div class="card-body">
58         <h4>Bettina Károly</h4>
59         <p>Magyar darts sport/</p>
60         <a href="#">Több info</a>
61       </div>
62     </div>
63   <div class="col-md-6 col-sm-4 col-3">
64     <div class="card">
65       
66       <div class="card-body">
67         <h4>Csaba Károly</h4>
68         <p>Magyar darts sport/</p>
69         <a href="#">Több info</a>
70       </div>
71     </div>
72   <div class="col-md-6 col-sm-4 col-3">
73     <div class="card">
74       
75       <div class="card-body">
76         <h4>Dömötör Károly</h4>
77         <p>Magyar darts sport/</p>
78         <a href="#">Több info</a>
79       </div>
80     </div>
81   </div>
```

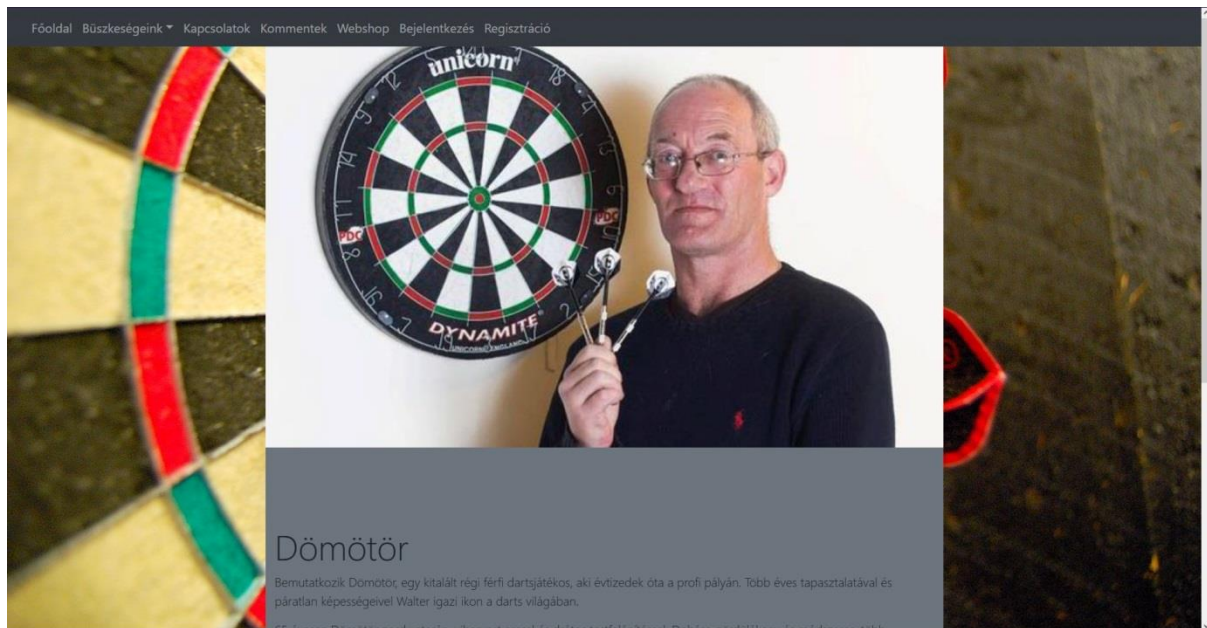
15. ÁBRA, FŐOLDAL MÖGÖTTI KÓD (SAJÁT FORRÁS, 2023)

A home.html a honlap főoldalának tartalmát foglalja magába. Ennek a lapnak a tetején egy carousel-t találunk, avagy egy dia showt. Három hangulateltető képet tartalmaz, amelyek maguktól időzítve lépnek, viszont felhasználó igényre lehet külön is léptetni a képek szélein található nyilakkal, vagy a képek alján található lapozó elemmel. Ezután egy rövid bemutatkozást találhatunk a sportegyesületről, és egy kis köszöntő beszédet. A szöveg után találhatunk egy gombot, ami egy rövid wikihow youtube videót nyit meg

nekünk, egy gyorstalpalóként kezdőknek vagy olyan felhasználóknak, akik még csak ismerkednek a dartssal.

A lap alján négy kis kártyát találhatunk az egyesület büszkeségeiről. Ezek a kártyák csak egy képet a sportolóról és a nevüket tartalmazzák. Mindegyik kártyán található egy-egy gomb, amivel további információkat olvashatunk az adott sportolóról.

4.2.6. Pride.html



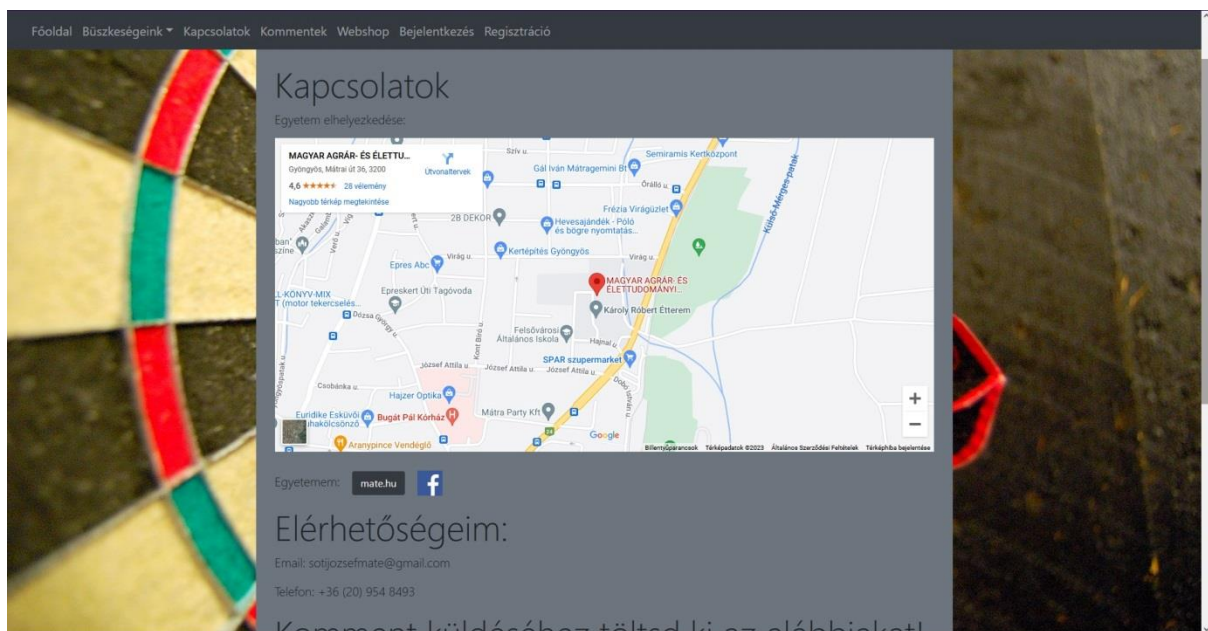
16. ÁBRA, DÖMÖTÖR, A BÜSZKESÉGÜNK (SAJÁT FORRÁS, 2023)



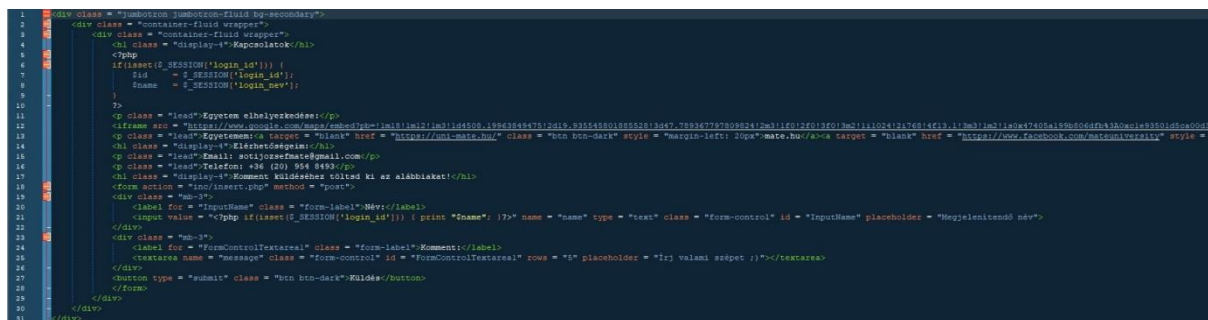
17. ÁBRA, BÜSZKESÉGÜNK MÖGÖTTI KÓD (SAJÁT FORRÁS, 2023)

A pride.html-ből négy darab is van, külön egy mindegyik sportolónak, akikre büszke az egyesület. Egyetlen kép és egy kis leírás található mindössze ezeken a lapokon.

4.2.7. Contacts.html



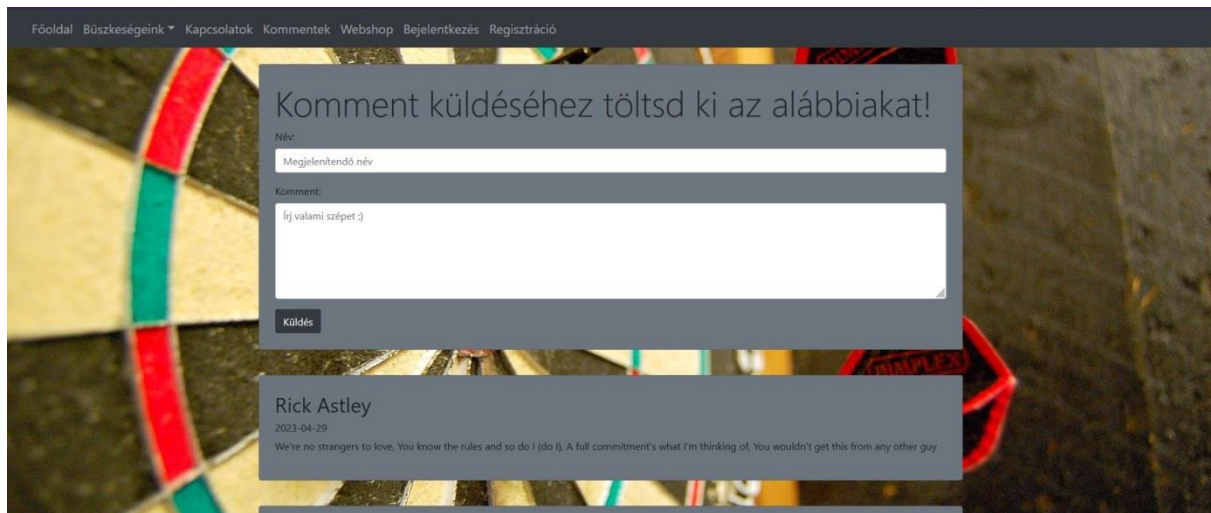
18. ÁBRA, KAPCSOLATOK (SAJÁT FORRÁS, 2023)



19. ÁBRA, KAPCSOLATOK MÖGÖTTI KÓD (SAJÁT FORRÁS, 2023)

A contacts.html-ben a sportegyesület kapcsolatait nézhetnénk meg, ha ez egy létező egyesület lenne. Mivel valami adatot a formaiságok miatt szolgáltatni kell, ezért a saját és az egyetemünk elérhetőségeit adtam meg. Először láthatjuk az egyetem elhelyezkedését egy importált Google térkép segítségével. Ez után 2 linket találhatunk, az első az egyetem hivatalos honlapjára, míg a második az egyetem hivatalos Facebook oldalára küldi a felhasználót. A linkek alatt találhatóak az email és telefonos elérhetőségeim. A lap alján található az egyik lehetőségünk kommentelni. Itt kettő szövegdobozt találunk, az egyik a kommentelő neve, a másik, pedig maga a komment. Beágyazott php gondoskodik arról, hogy ha a felhasználó be van jelentkezve, akkor automatikusan kitöltse a felhasználónevével a név cellát. A szövegdobozok alatt találjuk a Küldés gombot, amivel a cellák tartalmát elküldhetjük, ezzel az adatbázisba feltöltve azt és a következő lap gondoskodik a kommentek megjelenítéséről.

4.2.8. Comments.html



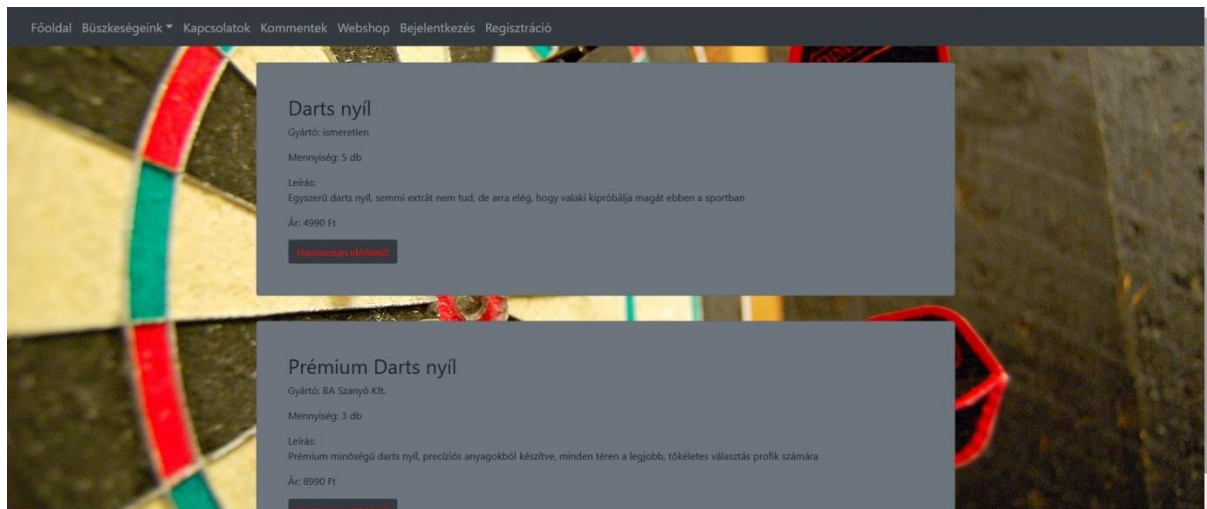
20. ÁBRA, KOMMENTEK (SAJÁT FORRÁS, 2023)

```
1 <div>
2 <div>{--div2c0br-->
3 <br>
4 </br>
5 <?php
6 if(isset($_SESSION['login_id'])) {
7     $id = $_SESSION['login_id'];
8     $name = $_SESSION['login_name'];
9 }
10 </?php>
11 <div class = "card bg-secondary" style = "padding:25px">
12 <div class = "display-4">Komment küldéséhez töltsd ki az alábbiakat!</div>
13 <form action = "index.php?oldal=insert" method = "POST">
14 <div class = "mb-3">
15 <div class = "form-label">Név:</div>
16 <input type = "text" value = "{$_SESSION['login_name']}">
17 </div>
18 <div class = "mb-3">
19 <div class = "form-label">Komment:</div>
20 <div class = "form-control">Írj valami szépet :)</div>
21 </div>
22 <div class = "text-center">
23 <button type = "submit" class = "btn btn-dark">Küldés</button>
24 </div>
25 </div>
26 <?php
27 $result = mysqli_query($conn, "SELECT * FROM comments ORDER BY id DESC");
28 while ($row = mysqli_fetch_assoc($result)) {
29     print "
30 <div class = \"card bg-secondary\" style = \"padding:25px\">
31 <div class = \"display-4\">{$row['name']}</div>
32 <div class = \"text-center\">{$row['date']}</div>
33 <div class = \"text-center\">{$row['message']}</div>
34 </div>";
35 }
```

21. ÁBRA, KOMMENTEK MÖGÖTTI KÓD (SAJÁT FORRÁS, 2023)

A comments.html-ben legfelül ismét találkozunk a komment küldés lehetőségével. Ez egy az egyben megegyezik a Kapcsolatok fülön találhatóval, azzal a különbséggel, hogy itt egy kártyába van beillesztve, ott meg a tartalom fő részébe volt beágyazva. Alatta a kommenteket láthatjuk egyesével külön kártyában, időrendben a legújabb van legfelül, a legrégebbi pedig legalul. Egy ilyen kártya a kommentelő nevét tartalmazza, a komment megírásának dátumát és magát az üzenetet.

4.2.9. Webshop.html



22. ÁBRA, WEBSHOP (SAJÁT FORRÁS, 2023)

```
1 <div><br><br>
2 <?php
3 $result = mysqli_query($conn, "SELECT * FROM products ORDER BY id ASC");
4 while ($row = mysqli_fetch_assoc($result)) {
5     print "
6     <div class = \"card bg-secondary\" style = \"padding:50px\">
7         <h2>$row[product]</h2>
8         <p>Gyártó: $row[manufacturer]</p>
9         <p>Mennyiség: $row[quantity] db</p>
10        <p>Leírás:<br>$row[description]</p>
11        <p>Ár: $row[price] Ft</p>
12        <form method = \"POST\" action = \"index.php?oldal=purchase\">
13            <button type = \"submit\" name = \"buy_button\" value = \"$row[id]\" class = \"btn btn-dark\" style = \"color:red\">Hamarosan elérhető!</button>
14        </form>
15    </div>";
16 }
17 <?>
18 </div>
```

23. ÁBRA, WEBSHOP MÖGÖTTI KÓD (SAJÁT FORRÁS, 2023)

A webshop.html egy kezdetleges terméklista, amiket meg lehet rendelni, de egyelőre még csak egy adattáblában elmenti a rendeléseket. A termékeket egy-egy kártyában tekinthetjük meg, amin belül megtaláljuk a termék megnevezését, gyártóját, kiserelését (csomagon belüli darabszámát), leírását és az árát, e termékek az azonosítójuk alapján növekvő sorrendben jelennek meg, tehát amilyen sorrendben hozzá lettek adva az adatbázishoz. Ezeket az adatokat a termékek adattáblából szedi a weblap. Minden kártya legalján megtalálható a 'Hamarosan elérhető!' gomb, amire ha rákattintunk, átvizsz minket a megrendelés lapra.

4.2.10. Purchase.html

Főoldal

Büszkeségeink

Kapcsolatok

Kommentek

Webshop

Bejelentkezés

Regisztráció

Megrendelés

Megrendelő

Minta Márton

Email cím:

Email cím

Telefonszám (06 vagy +36 nem kell leírni)

201234567

Termékkel kapcsolatos adatok

Termék

Prémium Darts nyíl

Mennyiség (darabban kifejezve)

Darabszám

Fizetési adatok

Kártya szám

1234 5678 8765 4321

24. ÁBRA, MEGRENDELÉS ŰRLAP (SAJÁT FORRÁS, 2023)

```
<!--div class = "pagination pagination-fluid bg-secondary"></div><br>  
<div class = "container-fluid wrapper">  
    <div class = "display-4">Megrendelés</div>  
        <div>  
            <div>  
                <ifisset($_SESSION['login_id'])> {  
                    $id  
                        = $_SESSION['login_id'];  
                    $name  
                        = $_SESSION['login_name'];  
                }  
                <ifisset($_POST)> {  
                    <ifproduct_id = $_POST['buy_button']>  
                        $sql = "SELECT * FROM products WHERE id = '$product_id'";  
                        $res = mysqli_query($conn, $sql);  
                        $row = mysqli_fetch_assoc($res);  
                        <if($row) {  
                            <$product_name = $row['product'];>  
                        }  
                    }  
                }  
            </div>  
  
            <div class = "row">  
                <div class = "col-lg-6">  
                    <form action = "index.php?id=order" method = "POST">  
                        <div class = "form-group">  
                            <label for = "username">Megrendelő</label>  
                            <input type = "text" value = "<?php if(isset($_SESSION['login_id'])) { print "$name"; } ?>" class = "form-control" name = "username" aria-describedby = "emailHelp" required placeholder = "Hívd meg az értékesítőt!" />  
                            <label for = "email">Email cím</label>  
                            <input type = "email" class = "form-control" id = "email" name = "email" aria-describedby = "emailHelp" required placeholder = "Email cím"/>  
                            <label for = "phone_number">Telefonszám (06 vagy +36 nem kell leírni)!</label>  
                            <input type = "text" class = "form-control" id = "phone_number" name = "phone_number" aria-describedby = "emailHelp" placeholder = "7012345678"></div>  
                            <div><checkbox checked="" type = "checkbox">Térmetéki expedíciós áradék!</div>  
                            <label for = "product">Termék</label>  
                            <input type = "text" value = "<?php print '$product_name'; ?>" class = "form-control" id = "product" name = "product" aria-describedby = "emailHelp" required placeholder = "Termék neve"/>  
                            <input type = "quantity">Mennyiség (darabban kifejezésben)!</div>  
                            <input type = "text" class = "form-control" id = "quantity" name = "quantity" aria-describedby = "emailHelp" required placeholder = "Darabszám"></div><div><button type = "submit" value = "Készítsd el!">Készítsd el!</div>  
                            <label for = "card_num">Kártya szám</label>  
                            <input type = "text" class = "form-control" id = "card number" name = "card number" aria-describedby = "emailHelp" required placeholder = "1234 5678 9012 3456"/>  
                            <label for = "expiration_date">Lejárat dátuma</label>  
                            <input type = "text" class = "form-control" id = "expiration date" name = "expiration date" required placeholder = "dd/mm/yy"/>  
                            <label for = "security">Biztonsági kód</label>  
                            <input type = "text" class = "form-control" id = "security" name = "security" required placeholder = "420"></div>  
                            <div><checkbox checked="" type = "checkbox">Árnyalatok!</div>  
                            <label for = "country">Ország</label>  
                            <input type = "text" class = "form-control" id = "country" name = "country" required placeholder = "Magyarország"/>  
                            <label for = "town">Település</label>  
                            <input type = "text" class = "form-control" id = "town" name = "town" required placeholder = "Gyöngyös"/>  
                            <label for = "postcode">Irányítószám</label>  
                            <input type = "text" class = "form-control" id = "postcode" name = "postcode" required placeholder = "3200"/>  
                            <label for = "street">Utca/Tér</label>  
                            <input type = "text" class = "form-control" id = "street" name = "street" required placeholder = "Matrai út"/>  
                            <label for = "house number">Hazai lakóhely</label>
```

25. ÁBRA, MEGRENDELÉS ÚRLAPJA MÖGÖTTI KÓD (SAJÁT FORRÁS, 2023)

A purchase.html a megrendelő felület, ami tizenhárom szövegdobozt tartalmaz, amelyek négy csoportra bonthatóak.

- A megrendelő adatai: megrendelő neve, megrendelő email címe, megrendelő telefonszáma. A megrendelő nevét automatikusan kitölti a felhasználó nevével, ha regisztrált felhasználó rendel, ezt beágyazott php kód végzi el.
- Termék adatok: termék neve, termék darabszáma.
- Fizetési adatok: bankkártya száma, bankkártya lejárat dátuma, bankkártya hátoldalán szereplő biztonsági kód.
- Kiszállítási adatok: Kiszállítási cím országa, települése, irányítószáma, útja/utcája/tere, házszáma.

4.2.11. Profile.html

[illegible]

26. ÁBRA, PROFIL ADATOK (SAJÁT FORRÁS, 2023)

```

1 <div class="jumbotronumbotron-fluid bg-secondary color-white">
2 <div class="container-fluid wrapper">
3 <div class="display-4">Profil adatok</div>
4 <div class="row">
5 <div class="col-lg-6">
6 <p>Profil adatok szerkesztéséhez kattintson a textbekezdőre</p>
7 <form action="{index.php?oldal=adatok}" method="post">
8 <input type="button" value="Profil szerkesztés" />
9 </form>
10 <div class="row">
11 <div class="col-lg-6">
12 <div class="form-group">
13 <div class="form-control">
14 <input type="text" value="{$_SESSION['login_id']}" />
15 </div>
16 </div>
17 <div class="form-control">
18 <input type="text" value="{$_SESSION['password']}" />
19 </div>
20 </div>
21 </div>
22 </div>
23 </div>
24 <div class="form-group">
25 <div class="form-control">
26 <input type="text" value="{$_SESSION['full_name']}" />
27 </div>
28 </div>
29 <div class="form-group">
30 <div class="form-control">
31 <input type="text" value="{$_SESSION['birth_date']}" />
32 </div>
33 </div>
34 <div class="form-group">
35 <div class="form-control">
36 <input type="text" value="{$_SESSION['country']}" />
37 </div>
38 </div>
39 <div class="form-group">
40 <div class="form-control">
41 <input type="text" value="{$_SESSION['town']}" />
42 </div>
43 </div>
44 <div class="form-group">
45 <div class="form-control">
46 <input type="text" value="{$_SESSION['postcode']}" />
47 </div>
48 </div>
49 <div class="form-group">
50 <div class="form-control">
51 <input type="text" value="{$_SESSION['street']}" />
52 </div>
53 </div>
54 <div class="form-group">
55 <div class="form-control">
56 <input type="text" value="{$_SESSION['house_number']}" />
57 </div>
58 </div>
59 </div>
60 </div>
61 </div>
62 </div>
63 </div>
64 </div>
65 </div>
66 </div>
67 </div>
68 </div>
69 </div>
70 </div>
71 </div>
72 </div>
73 </div>
74 </div>
75 </div>
76 </div>
77 </div>
78 </div>
79 </div>
80 </div>
81 </div>
82 </div>
83 </div>
84 </div>
85 </div>
86 </div>
87 </div>
88 </div>
89 </div>
90 </div>
91 </div>
92 </div>
93 </div>
94 </div>
95 </div>
96 </div>
97 </div>
98 </div>
99 </div>
100 </div>
101 </div>
102 </div>
103 </div>
104 </div>
105 </div>
106 </div>
107 </div>
108 </div>
109 </div>
110 </div>
111 </div>
112 </div>
113 </div>
114 </div>
115 </div>
116 </div>
117 </div>
118 </div>
119 </div>
120 </div>
121 </div>
122 </div>
123 </div>
124 </div>
125 </div>
126 </div>
127 </div>
128 </div>
129 </div>
130 </div>
131 </div>
132 </div>
133 </div>
134 </div>
135 </div>
136 </div>
137 </div>
138 </div>
139 </div>
140 </div>
141 </div>
142 </div>
143 </div>
144 </div>
145 </div>
146 </div>
147 </div>
148 </div>
149 </div>
150 </div>
151 </div>
152 </div>
153 </div>
154 </div>
155 </div>
156 </div>
157 </div>
158 </div>
159 </div>
160 </div>
161 </div>
162 </div>
163 </div>
164 </div>
165 </div>
166 </div>
167 </div>
168 </div>
169 </div>
170 </div>
171 </div>
172 </div>
173 </div>
174 </div>
175 </div>
176 </div>
177 </div>
178 </div>
179 </div>
180 </div>
181 </div>
182 </div>
183 </div>
184 </div>
185 </div>
186 </div>
187 </div>
188 </div>
189 </div>
190 </div>
191 </div>
192 </div>
193 </div>
194 </div>
195 </div>
196 </div>
197 </div>
198 </div>
199 </div>
200 </div>
201 </div>
202 </div>
203 </div>
204 </div>
205 </div>
206 </div>
207 </div>
208 </div>
209 </div>
210 </div>
211 </div>
212 </div>
213 </div>
214 </div>
215 </div>
216 </div>
217 </div>
218 </div>
219 </div>
220 </div>
221 </div>
222 </div>
223 </div>
224 </div>
225 </div>
226 </div>
227 </div>
228 </div>
229 </div>
230 </div>
231 </div>
232 </div>
233 </div>
234 </div>
235 </div>
236 </div>
237 </div>
238 </div>
239 </div>
240 </div>
241 </div>
242 </div>
243 </div>
244 </div>
245 </div>
246 </div>
247 </div>
248 </div>
249 </div>
250 </div>
251 </div>
252 </div>
253 </div>
254 </div>
255 </div>
256 </div>
257 </div>
258 </div>
259 </div>
260 </div>
261 </div>
262 </div>
263 </div>
264 </div>
265 </div>
266 </div>
267 </div>
268 </div>
269 </div>
270 </div>
271 </div>
272 </div>
273 </div>
274 </div>
275 </div>
276 </div>
277 </div>
278 </div>
279 </div>
280 </div>
281 </div>
282 </div>
283 </div>
284 </div>
285 </div>
286 </div>
287 </div>
288 </div>
289 </div>
290 </div>
291 </div>
292 </div>
293 </div>
294 </div>
295 </div>
296 </div>
297 </div>
298 </div>
299 </div>
300 </div>
301 </div>
302 </div>
303 </div>
304 </div>
305 </div>
306 </div>
307 </div>
308 </div>
309 </div>
310 </div>
311 </div>
312 </div>
313 </div>
314 </div>
315 </div>
316 </div>
317 </div>
318 </div>
319 </div>
320 </div>
321 </div>
322 </div>
323 </div>
324 </div>
325 </div>
326 </div>
327 </div>
328 </div>
329 </div>
330 </div>
331 </div>
332 </div>
333 </div>
334 </div>
335 </div>
336 </div>
337 </div>
338 </div>
339 </div>
340 </div>
341 </div>
342 </div>
343 </div>
344 </div>
345 </div>
346 </div>
347 </div>
348 </div>
349 </div>
350 </div>
351 </div>
352 </div>
353 </div>
354 </div>
355 </div>
356 </div>
357 </div>
358 </div>
359 </div>
360 </div>
361 </div>
362 </div>
363 </div>
364 </div>
365 </div>
366 </div>
367 </div>
368 </div>
369 </div>
370 </div>
371 </div>
372 </div>
373 </div>
374 </div>
375 </div>
376 </div>
377 </div>
378 </div>
379 </div>
380 </div>
381 </div>
382 </div>
383 </div>
384 </div>
385 </div>
386 </div>
387 </div>
388 </div>

```

27. ÁBRA, PROFIL ADATOK MÖGÖTTI KÓD (SAJÁT FORRÁS, 2023)

35

feltölti a cellákat, amiket, ha módosítunk, akkor a lap alján lévő Módosítás gombbal rögzíthetünk is. Ilyenkor csak lefrissíti az adatokat az adatbázisban, nem hoz létre új rekordot.

4.3. Backend

4.3.1. Mi is az a backend

A webes programozásban a backend kifejezés a webalkalmazás azon részét jelenti, amely az alkalmazás adatainak és logikájának feldolgozásáért és kezeléséért felelős. A webalkalmazás szerveroldalának is nevezik.

A háttérrendszer általában tartalmaz egy webszervert, egy alkalmazáskiszolgálót és egy adatbázist. A webszerver fogadja és feldolgozza a kéréseket az alkalmazás kliens oldaláról (frontend), az alkalmazásszerver pedig végrehajtja a szükséges logikát és feldolgozást a válasz generálásához. Az adatbázis tárolja és kezeli az alkalmazás által használt adatokat.

4.3.2. Index.php

```
1 <?php
2 session_start();
3 include('inc/db_connect.php');
4 include('inc/jump.php');
5 if(isset($_GET['oldal'])) {
6     switch($_GET['oldal']) {
7         case 'comments' : $content = 'comments.html'; break;
8         case 'contacts' : $content = 'contacts.html'; break;
9         case 'home' : $content = 'home.html'; break;
10        case 'insert' : $content = 'insert.php'; break;
11        case 'login' : $content = 'login.html'; break;
12        case 'loginvalid' : $content = 'login.php'; break;
13        case 'logout' : $content = 'logout.php'; break;
14        case 'order' : $content = 'order.php'; break;
15        case 'pridel' : $content = 'pridel.html'; break;
16        case 'pride2' : $content = 'pride2.html'; break;
17        case 'pride3' : $content = 'pride3.html'; break;
18        case 'pride4' : $content = 'pride4.html'; break;
19        case 'profile' : $content = 'profile.html'; break;
20        case 'purchase' : $content = 'purchase.html'; break;
21        case 'regist' : $content = 'regist.html'; break;
22        case 'regment' : $content = 'regment.php'; break;
23        case 'update' : $content = 'update.php'; break;
24        case 'webshop' : $content = 'webshop.html'; break;
25        default : $content = 'home.html';
26    }
27 }
28 else {
29     $content = "home.html";
30 }
31 include("index.html");
32 ?>
```

28. ÁBRA, INDEX.PHP (SAJÁT FORRÁS, 2023)

Az index.php határozza meg, mi legyen a tartalom az index.html-en belül. Először is elindít egy SESSION-t avagy munkafolyamatot. Magába foglalja a db_connect és jump metódusokat, ez azt jelenti, hogy ameddig a SESSION tart, addig bárhol elérhetőek lesznek anélkül, hogy újra meg kelljen őket írni. \$_GET-tel megkapjuk, hogy az adott pillanatban, mit kell betölteni tartalomként, vagy éppen melyik php parancssort kell lefuttatni. Ezt egy switch-case-el érjük el. Ez azt jelenti, hogy minden egyes esetre meghatározzuk egyesével, hogy mit kell tenni. Ha bármi probléma felmerülne, külön meg lett határozva, hogy a főoldal kerüljön betöltésre.

4.3.3. Db_connect.php

```
1 <?php
2 $server      = 'localhost';
3 $user        = 'root';
4 $password    = '';
5 $db          = 'szakdolgozat';
6 $con         = mysqli_connect($server, $user, $password, $db);
7 if($con === false) {
8     die("ERROR: Nem sikerült csatlakozni :(" . mysqli_connect_error());
9 }
10 ?>
```

29. ÁBRA, DB_CONNECT.PHP (SAJÁT FORRÁS, 2023)

A db_connect.php csak egyetlen parancssort tartalmaz, ami meghatározza a csatlakozáshoz szükséges információkat. Igazából ezt nem szükséges külön fájlban lementeni, viszont így csak egyszer kell megírni és mivel include-olva lett ezért ameddig a SESSION tart addig bármelyik kódrészlet eléri.

4.3.4. Jump.php

```
1 <?php
2 function jump($where, $when) {
3     print '
4         <script>
5             setTimeout("window.location.href = \''.$where.'\'", '.$when.');
6         </script>
7     ';
8 }
9 ?>
```

30. ÁBRA, JUMP.PHP (SAJÁT FORRÁS, 2023)

A jump.php az előzőhöz hasonlóan csak egy metódust tartalmaz, aminek a célja, hogy bizonyos műveletek elvégzése után átugorjon egy másik fülre. Például sikeres regisztráció után három másodperc után átugrik a honlap a bejelentkezés űrlapjára.

4.3.5. Regment.php

```
1 <?php
2 if(isset($_POST)) {
3     $username = $_POST['username'];
4     $email = $_POST['email'];
5     $password = $_POST['password'];
6     $password2 = $_POST['password2'];
7     if($password == $password2) {
8         $sql = "SELECT * FROM users WHERE username = '$username' OR email = '$email' LIMIT 1;";
9         $res = mysqli_query($con, $sql);
10        $row = mysqli_fetch_assoc($res);
11        if($row) {
12            if($email == $row['email']) {
13                print "<h2><br><br>Ezzel az email címmel már regisztráltak!</h2>";
14                jump("index.php?oldal=regist", 3000);
15            }
16            if($username == $row['username']) {
17                print "<h2><br><br>Ezzel a felhasználó névvel már regisztráltak!</h2>";
18                jump("index.php?oldal=regist", 3000);
19            }
20        }
21        else {
22            $sql_1 = "INSERT INTO users VALUES ('', '$username', '$email', '$password')";
23            mysqli_query($con, $sql_1);
24            $sql_2 = "INSERT INTO profiles VALUES ('', '$username', '-', '-', '2000-01-01', '-', '-', '0', '-', '0', '0')";
25            mysqli_query($con, $sql_2);
26            print "<h2><br><br>Sikeres regisztráció!</h2>";
27            jump("index.php?oldal=login", 3000);
28        }
29    }
30    else {
31        print "<h2><br><br>A két jelszónak meg kell egyeznie!</h2>";
32        jump("index.php?oldal=regist", 3000);
33    }
34 }
35 >
```

31. ÁBRA, REGMENT.PHP (SAJÁT FORRÁS, 2023)

A regment.php a regisztrációs folyamatért felelős. Először a szövegdobozokból kiszedi a tartalmakat és elmenti azokat változóba. Ez után megvizsgálja, hogy a kettő megadott jelszó megegyezik-e, ha nem, akkor hibaüzenetet dob és visszaugrik a regisztrációs űrlapra. Ha megegyezik a kettő megadott jelszó, akkor megvizsgálja, hogy ezzel az email címmel és/vagy felhasználónévvel regisztráltak-e már. Ha igen akkor ismételt hibaüzenetet ír ki, viszont ha egyik sem volt még használva, akkor a regisztráció sikeres volt. Ilyenkor 2 adattáblába is létrehoz egy új rekordot, egyik a users tábla ahova a megadott információkat írja, a másikban, a profiles-ben csak lefoglalja a helyet, hogy későbbiekben ha a felhasználó szeretné el tudja menteni személyes adatait majd. Sikeres regisztráció esetén is kapunk visszajelzést, majd átdobja a felhasználót a bejelentkezési űrlaphoz.

4.3.6. Login.php

```
1 <?php
2 foreach($_POST as $name => $value) {
3     $$name = $value;
4 }
5 if(isset($email)) {
6     $result = mysqli_query($con, "SELECT * FROM users WHERE email = '$email' AND password = '$password'");
7     $row = mysqli_fetch_assoc($result);
8     if(mysqli_num_rows($result) == 1) {
9         print "<h2><br><br>Sikeres bejelentkezés</h2>";
10        print "<h2>Üdvözöllek $row[username]!</h2>";
11        $_SESSION['login_id'] = $row['id'];
12        $_SESSION['login_nev'] = $row['username'];
13        $_SESSION['login_email'] = $row['email'];
14        jump("index.php?oldal=kezdolap", 3000);
15    }
16    else {
17        print "<h2><br><br>Hibás felhasználónév vagy jelszó</h2>";
18        jump("index.php?oldal=login", 3000);
19    }
20 }
21 else {
22     print "Bejelentkezéshez kattintson a Bejelentkezés menüpontra!!!";
23 }
24 ?>
```

32. ÁBRA, LOGIN.PHP (SAJÁT FORRÁS, 2023)

A login.php a bejelentkezést végzi el, az űrlap celláiból változókba tárolja a bejelentkezés űrlapján beírt tartalmakat. Megvizsgálja, hogy létezik-e ezzel az email cím és jelszó kombinációval felhasználó, ha igen akkor bejelentkezik, és SESSION változókba elmenti a felhasználó azonosítóját, felhasználónevét és email címét. Ha nem talál ezzel a kombinációval felhasználót az adatbázisban hibaüzenetet dob.

4.3.7. Logout.php

```
1 <?php
2 session_destroy();
3 jump("index.php?oldal=kezdolap", 1);
4 ?>
```

33. ÁBRA, LOGOUT.PHP (SAJÁT FORRÁS, 2023)

A logout.php szimplán leállítja a SESSION-t tehát kijelentkezteti a felhasználót.

4.3.8. Insert.php

```
1 <?php
2 $date = date("Y-m-d");
3 $name = mysqli_real_escape_string($con, $_REQUEST['name']);
4 $message = mysqli_real_escape_string($con, $_REQUEST['message']);
5 $sql = "INSERT INTO comments (id, date, name, message) VALUES ('', '$date', '$name', '$message')";
6 if(mysqli_query($con, $sql)) {
7     print "<h2><br><br>Komment elküldve sikeresen :)</h2>";
8     jump("index.php?oldal=comments", 3000);
9 }
10 else {
11     print "<h2><br><br>ERROR: Komment elküldése sikertelen :(</h2>" . mysqli_error($con);
12     jump("index.php?oldal=contacts", 3000);
13 }
14 mysqli_close($con);
15 ?>
```

34. ÁBRA, INSERT.PHP (SAJÁT FORRÁS, 2023)

Az insert.php a komment elküldésére készült, változóba elmenti a komment megírásának dátumát, a kommentet megíró felhasználó nevét és magát az üzenetet. Majd ezen változók tartalmát feltölti az adatbázisba, majd erről visszajelzést ír. Ha valami nem jönne össze, akkor viszont a hibaüzenetet írja ki.

4.3.9. Order.php

```

1 <?php
2 $date = date("Y-m-d");
3 $username = mysqli_real_escape_string($conn, $_REQUEST['username']);
4 $email = mysqli_real_escape_string($conn, $_REQUEST['email']);
5 $phone_number = mysqli_real_escape_string($conn, $_REQUEST['phone_number']);
6 $product = mysqli_real_escape_string($conn, $_REQUEST['product']);
7 $quantity = mysqli_real_escape_string($conn, $_REQUEST['quantity']);
8 $card_number = mysqli_real_escape_string($conn, $_REQUEST['card_number']);
9 $expiration_date = mysqli_real_escape_string($conn, $_REQUEST['expiration_date']);
10 $security = mysqli_real_escape_string($conn, $_REQUEST['security']);
11 $country = mysqli_real_escape_string($conn, $_REQUEST['country']);
12 $town = mysqli_real_escape_string($conn, $_REQUEST['town']);
13 $postcode = mysqli_real_escape_string($conn, $_REQUEST['postcode']);
14 $street = mysqli_real_escape_string($conn, $_REQUEST['street']);
15 $house_number = mysqli_real_escape_string($conn, $_REQUEST['house_number']);
16 $sql = "INSERT INTO orders (id, date, username, email, $phone_number, product, quantity, card_number, expiration_date, security, country, town, postcode, street, house_number) VALUES ('', '$date', '$username', '$email', '$phone_n";
17 if(mysqli_query($conn, $sql)) {
18     print "<h2>Úr-úrnő! Rendelését sikeresen rögzítettük :)</h2>";
19     jump("index.php?oldal=home", 3000);
20 }
21 else {
22     print "<h2>Úr-úrnő! Rendelés feladása sikertelen :(</h2>" . mysqli_error($conn);
23     jump("index.php?oldal=webshop", 3000);
24 }
25 mysqli_close($conn);
26 >

```

35. ÁBRA, ORDER.PHP (SAJÁT FORRÁS, 2023)

Az order.php a rendelés rögzítését végzi, a többihez hasonlóan változóba kimenti a cellákban megadott értékeket, majd azokat beilleszti az orders adattábla megfelelő mezőibe. A sikeres rendelésről pozitív visszajelzést kapunk, viszont ha bármilyen probléma felmerül, azon esetben negatív visszajelzést kapunk.

4.3.10. Update.php

```

1 <?php
2 if(isset($_POST)) {
3     $sender = $_POST['sender'];
4     $full_name = $_POST['full_name'];
5     $birth_date = $_POST['birth_date'];
6     $country = $_POST['country'];
7     $town = $_POST['town'];
8     $postcode = $_POST['postcode'];
9     $street = $_POST['street'];
10    $house_number = $_POST['house_number'];
11    $phone_number = $_POST['phone_number'];
12    $id = $_SESSION['login_id'];
13    $sql = "UPDATE profiles SET sender = '$sender', full_name = '$full_name', birth_date = '$birth_date', country = '$country', town = '$town', postcode = '$postcode', street = '$street', house_number = '$house_number', phone_num";
14    mysqli_query($conn, $sql);
15    print "<h2>Úr-úrnő! Adatok sikeresen frissítve :)</h2>";
16    jump("index.php?oldal=home", 3000);
17 }
18 >

```

36. ÁBRA, UPDATE.PHP (SAJÁT FORRÁS, 2023)

Az update.php a személyes adatok lefrissítésére szolgál, nem ad hozzá új rekordot a táblához, hanem már a meglévő rekordok közül a bejelentkezett felhasználóhoz tartozó sort módosítja. A szövegdobozok tartalmát kimenti változóba és azokkal az adatokkal írja majd felül a már meglévő adatokat a Módosítás gombra történő kattintás után és itt is kapunk visszajelzést a feladat sikeres elvégzéséről.

5. Eredmények, következtetések és javaslatok

Fel lehet tenni a kérdést, hogy mi értelme, hogy elkészítsem ezt a honlapot. Mint a bevezetőben is említettem, legelőször a jászberényi Gróf Apponyi Albert Általános Iskolának készítettem volna egy honlapot, de sajnos a megrendelő, avagy az igazgatóság lemondta és inkább valaki mást kértek meg, hogy készítse ezt el nekik. Ezután tanácstalan voltam és konzulensemhez, Sike Zoltánhoz fordultam segítségért a témaválasztással kapcsolatban. Fel is sorolt nekem legalább öt lehetőséget és egyből megragadta a figyelmem ez a téma. Elméletileg egy létező sportegyesületnek készítettem volna honlapot, de út közben úgy döntöttem, hogy inkább szabadra engedném fantáziámat és egy fiktívnek készítenék honlapot. Így nincs meghatározva miknek kell lennie a weblapon és megtapasztalhatom több webes funkció elkészítését is. Viszont a honlap elkészítése során végig szem előtt tartottam, hogy nagyon jól olvasható és értelmezhető legyen a kód és ha majd valaki a későbbiekben meg szeretné valósítani az alapból megbeszélte sportegyesület honlapját, akkor ha nem is az egész munkámat, de legalább részeit fel tudja majd használni. Igazából egy sablont szerettem volna készíteni, ami jó alapul szolgálhat majd az éles honlap elkészítése során. Ezen felül alaposan megismertem a webprogramozás lépéseit és tapasztalatot szereztem ebben a témakörben is.

Természetesen ez még csak egy úgynevezett „alpha” változat, tehát még csak tesztelés céljából készült el. Ez azt jelenti, hogy ez még nem a végleges változat, szóval még igény szerint ráférne még egy-két módosítás és kiegészítés néhány extra funkcióval. Volt néhány ötletem, amit meg szeretnék majd valósítani a jövőben, ezek közé tartozik az elfelejtett jelszó funkció, avagy ha a felhasználó elfelejti a jelszavát, akkor legenerál egy újat a program és azt elküldi emailben vagy a rendelés úrlapon egy funkció, amely a kiszállítási címnek a szövegdozsoit kitölti a regisztrált felhasználó lakcímével, ha mondjuk egy pipát berak a megrendelő.

6. Összefoglalás

Röviden összefoglalva, ez a webalkalmazás egy darts sportegyesületnek készült volna, viszont mivel nem vagyok kellően jártas a témában, ezért úgy döntöttem, hogy inkább egy fiktív sportegyesületről készítek weblapot. Több indokom is volt a döntésemre, mint például, nem szeretek olyanról írni, amiről nincsenek alapos információim, ezért találtam ki egy fikcionális sportegyesületet, mert amit én álmodtam meg, azt muszáj ismernem. A másik indokom pedig az volt, hogy így nem volt megszabva miket kell leprogramoznom, magyarul nem voltam lekorlátozva. Pontosan ezen okból kifolyóan, kipróbálhattam olyan funkciók elkészítését, amik alaptól nem kerültek volna be a weblapba. Az alap ötletet, ha már nem valósítottam meg legalább egy kis segítséget szerettem volna biztosítani, ezért a honlap készítése közben végig törekedtem arra, hogy a kód egyszerű, jól átlátható és könnyen értelmezhető legyen. Ha nem is használják majd fel az egész kódot, legalább részeket át tudjanak majd emelni, vagy legalább sablonként tudjon szolgálni és ez alapján könnyebb legyen elkészíteni majd a fejlesztőnek a létező sportegyesület hivatalos honlapját.

Ez a webalkalmazás, az alapvető funkciókon felül néhány érdekesebb, nagyobb kihívásként ismert dolgot is magába foglal. Az alapok funkciók alatt, a regisztrációs adatlapba kitöltött adatok, adatbázisba illesztésére, avagy a regisztrációs folyamatra, a bejelentkezési űrlapba beírt információk adatbázisban történővel való összehasonlítása és siker esetén SESSION indítása, tehát bejelentkezésre és munkafolyamat leállítására más néven kijelentkezésre. Mindezen funkciókon felül, a felhasználónak meg van a lehetősége kommenteket írni és olvasni is. Az előbb említettek mellett, még egy webshopot is készítettem, bár a termékek nem valódiak, a rendelési funkciót is elkészítettem hozzájuk. Viszont mivel ez még csak teszt fázisban van, ezért a termékek alatt nem „Megrendelem”, hanem „Hamarosan elérhető!” felirat olvasható a gombokon. Mint említettem, próbáltam az egyszerűsége törekedni, ezért nem lett felhasználói jogosultsági szint meghatározva külön, de az ADMIN felhasználót, a felhasználóneve alapján beazonosítja és neki egyedi a felső menüsora (navbar), hiszen egy admin felület gombbal lett kiegészítve. Ugyan nem készítettem neki egyedi admin felületet, de belinkeltem a XAMPP-nak a beépített pypMyAdmin felületét, amely bőven tud mindent, amire a honlap adminjának szüksége lehet.

7. Irodalomjegyzék

- [1] Notepad++: <https://www.infoartnet.hu/html-forras-szerkesztese-funkcio>
- [2] Visual Studio Code:
https://videotanfolyam.hu/blog/1886/Visual_Studio_Code_az_egyik_legjobb_ingyenes_kodszerkeszto
- [3] XAMPP: <https://webiskola.hu/php-ismeretek/php-apache-xampp-letoltese-telepitese/>
- [4] phpMyAdmin:
- [5] HTML: <https://webiskola.hu/html-ismeretek/mi-az-a-html-a-html-bemutatas/>
- [6] CSS: <https://matebalazs.hu/css.html>
- [7] JavaScript: <https://webiskola.hu/javascript-ismeretek/mi-az-a-javascript-a-js-bemutatas/>
- [8] Bootstrap: <https://gremmedia.hu/mi-az-a-bootstrap-4-hogyan-hasznaljuk>
- [9] jQuery: <https://gremmedia.hu/mi-az-a-jquery-a-legjobb-jquery-widgetek>
- [10] Node.js: <http://www.inf.u-szeged.hu/~tarib/javascript/nodejs.html#irodalomjegyzek>
- [11] Adatbázis: <http://www.inf.u-szeged.hu/~tarib/javascript/nodejs.html#irodalomjegyzek>
- [12] Frontend: <https://lexiq.hu/front-end>
- [13] Backend: <https://lexiq.hu/back-end>

8. Ábrajegyzék

1. ÁBRA, SZAKDOLGOZAT ADATBÁZIS (SAJÁT FORRÁS, 2023)	17
2. ÁBRA, USERS ADATTÁBLA (SAJÁT FORRÁS, 2023)	18
3. ÁBRA, PROFILES ADATTÁBLA (SAJÁT FORRÁS, 2023)	19
4. ÁBRA, GENDERS ADATTÁBLA (SAJÁT FORRÁS, 2023)	21
5. ÁBRA, PRODUCTS ADATTÁBLA (SAJÁT FORRÁS, 2023)	22
6. ÁBRA, ORDERS ADATTÁBLA (SAJÁT FORRÁS, 2023)	22
7. ÁBRA, COMMENTS ADATTÁBLA (SAJÁT FORRÁS, 2023)	24
8. ÁBRA, NAVBAR (SAJÁT FORRÁS, 2023)	25
9. ÁBRA, NAVBAR MÖGÖTTI KÓD (SAJÁT FORRÁS, 2023)	26
10. ÁBRA, REGISZTRÁCIÓS FELÜLET (SAJÁT FORRÁS, 2023)	27
11. ÁBRA, REGISZTRÁCIÓS FELÜLET MÖGÖTTI KÓD (SAJÁT FORRÁS, 2023)	27
12. ÁBRA, BEJELENTKEZÉSI FELÜLET (SAJÁT FORRÁS, 2023)	28
13. ÁBRA, BEJELENTKEZÉSI FELÜLET MÖGÖTTI KÓD (SAJÁT FORRÁS, 2023)	28
14. ÁBRA, FŐOLDAL (SAJÁT FORRÁS, 2023)	29
15. ÁBRA, FŐOLDAL MÖGÖTTI KÓD (SAJÁT FORRÁS, 2023)	29
16. ÁBRA, DÖMÖTÖR, A BÜSZKESÉGÜNK (SAJÁT FORRÁS, 2023)	30
17. ÁBRA, BÜSZKESÉGÜNK MÖGÖTTI KÓD (SAJÁT FORRÁS, 2023)	30
18. ÁBRA, KAPCSOLATOK (SAJÁT FORRÁS, 2023)	31
19. ÁBRA, KAPCSOLATOK MÖGÖTTI KÓD (SAJÁT FORRÁS, 2023)	31
20. ÁBRA, KOMMENTEK (SAJÁT FORRÁS, 2023)	32
21. ÁBRA, KOMMENTEK MÖGÖTTI KÓD (SAJÁT FORRÁS, 2023)	32
22. ÁBRA, WEBSHOP (SAJÁT FORRÁS, 2023)	33
23. ÁBRA, WEBSHOP MÖGÖTTI KÓD (SAJÁT FORRÁS, 2023)	33
24. ÁBRA, MEGRENDELÉS ŰRLAP (SAJÁT FORRÁS, 2023)	34
25. ÁBRA, MEGRENDELÉS ŰRLAPJA MÖGÖTTI KÓD (SAJÁT FORRÁS, 2023)	34
26. ÁBRA, PROFIL ADATOK (SAJÁT FORRÁS, 2023)	35
27. ÁBRA, PROFIL ADATOK MÖGÖTTI KÓD (SAJÁT FORRÁS, 2023)	35
28. ÁBRA, INDEX.PHP (SAJÁT FORRÁS, 2023)	36
29. ÁBRA, DB_CONNECT.PHP (SAJÁT FORRÁS, 2023)	37
30. ÁBRA, JUMP.PHP (SAJÁT FORRÁS, 2023)	37
31. ÁBRA, REGMENT.PHP (SAJÁT FORRÁS, 2023)	38
32. ÁBRA, LOGIN.PHP (SAJÁT FORRÁS, 2023)	39
33. ÁBRA, LOGOUT.PHP (SAJÁT FORRÁS, 2023)	39
34. ÁBRA, INSERT.PHP (SAJÁT FORRÁS, 2023)	39
35. ÁBRA, ORDER.PHP (SAJÁT FORRÁS, 2023)	40
36. ÁBRA, UPDATE.PHP (SAJÁT FORRÁS, 2023)	40

9. Nyilatkozatok

NYILATKOZAT

a szakdolgozat nyilvános hozzáféréséről és eredetiségéről

A hallgató neve: Sóti József Máté
A Hallgató Neptun kódja: M829RZ
A dolgozat címe: Nagyon jó sportegyesület webalkalmazás
A megjelenés éve: 2023
A konzulens tanszék neve: Alkalmazott informatikai tanszék

Kijelentem, hogy az általam benyújtott szakdolgozat egyéni, eredeti jellegű, saját szellemi alkotásom. Azon részeket, melyeket más szerzők munkájából vettem át, egyértelműen megjelöltem, s az irodalomjegyzékben szerepeltettem.

Ha a fenti nyilatkozattal valótlan állítottam, tudomásul veszem, hogy a Záróvizsga-bizottság a záróvizsgából kizár és a záróvizsgát csak új dolgozat készítése után tehetek.

A leadott dolgozat, mely PDF dokumentum, szerkesztését nem, megtekintését és nyomtatását engedélyezem.

Tudomásul veszem, hogy az általam készített dolgozatra, mint szellemi alkotás felhasználására, hasznosítására a Magyar Agrár- és Élettudományi Egyetem mindenkori szellemi tulajdon-kezelési szabályzatában megfogalmazottak érvényesek.

Tudomásul veszem, hogy dolgozatom elektronikus változata feltöltésre kerül a Magyar Agrár- és Élettudományi Egyetem könyvtári repozitori rendszerébe.

Kelt: Gyöngyös 2023 év 05 hó 02 nap



Hallgató aláírása

KONZULTÁCIÓS NYILATKOZAT

A Sóti József Máté (hallgató Neptun azonosítója: M829RZ) konzulenseként nyilatkozom arról, hogy a szakdolgozatot áttekintettem, a hallgatót az irodalmi források korrekt kezelésének követelményeiről, jogi és etikai szabályairól tájékoztattam.

A szakdolgozatot a záróvizsgán történő védeésre javaslom / nem javaslom¹.

A dolgozat állam- vagy szolgálati titkot tartalmaz: igen nem²

Kelt: Gyöngyös 2023 év 05 hó 02 nap


Belső konzulens

¹ A megfelelő aláhúzendő.

² A megfelelő aláhúzendő.

10. Mellékletek

Soti_Jozsef_Mate_M829RZ_2023_B-GYO-N-HU-GAZDI_Melleklet elnevezésű rar kiterjesztésű fájl, amely a következőket tartalmazza:

Adatbázis mappa: a webalkalmazáshoz tartozó szakdolgozat nevű exportált adatbázist tartalmazza

Honlap mappa: a webalkalmazást felépítő fájlokat tartalmazza, beleértve a HTML, PHP, CSS fájlokat és a felhasznált képeket.