BAJAI SZAKKÉPZÉSI CENTRUM KALOCSAI DÓZSA GYÖRGY TECHNIKUM ÉS KOLLÉGIUM

Készítette: Szeghalmi László, Murányi Marcell, Kovács Sándor

Tartalom

[Bevezetés 4](#_Toc190340997)

[Záródolgozat témájának megindoklása, projekt célja 4](#_Toc190340998)

[Célközönség 4](#_Toc190340999)

[Rendszerkövetelmény(frontend) 5](#_Toc190341000)

[Böngészők követelményei: 5](#_Toc190341001)

[Rendszerkövetelmény(backend) 6](#_Toc190341002)

[XAMPP: 6](#_Toc190341003)

[Visual Studio Code: 6](#_Toc190341004)

[Fejlesztői dokumentáció 7](#_Toc190341005)

[Szoftverek amiket használtunk 7](#_Toc190341006)

[Windows 10 7](#_Toc190341007)

[Visual Studio Code 7](#_Toc190341008)

[Xampp 7](#_Toc190341009)

[Programozási nyelvek, amiket használtunk 8](#_Toc190341010)

[HTML 8](#_Toc190341011)

[CSS 8](#_Toc190341012)

[Javascript (JS) 8](#_Toc190341013)

[MySQL 9](#_Toc190341014)

[Adatbázis szerkezete 9](#_Toc190341015)

[Az adatbázis táblái: 9](#_Toc190341016)

[Termékek tábla: 10](#_Toc190341017)

[Rendelések táblái 11](#_Toc190341018)

[Vásárlók táblái: 12](#_Toc190341019)

[Algoritmusok a weboldalon 12](#_Toc190341020)

[Regisztráció(controller): 12](#_Toc190341021)

[Regisztráció(router): 14](#_Toc190341022)

[Bejelentkezés(controller): 15](#_Toc190341023)

[Regisztráció(router): 17](#_Toc190341024)

[Felhasználói dokumentáció 17](#_Toc190341025)

[Források 17](#_Toc190341026)

[Köszönetnyilvánítás 18](#_Toc190341027)

# Bevezetés

## Záródolgozat témájának megindoklása, projekt célja

A mai világban az emberek ki vannak téve az egészségtelen ételeknek ezáltal egyre jobban szükségünk van az egészséges és mozgás dús életmódra. Ebben a szellemben a konditermek jelentős szerepet töltenek be a rendszeres testmozgás biztosításában, valamint az egészség megőrzésében. A vizsga remekünk célja hogy egy olyan weboldalt fejlesszünk, ami egy konditerem termékeit és az egészséges életmód fontos szükségleteit ajánlja a vásárlóknak és hasznos információkkal látja el őket. A témaválasztásunk alapja, hogy mi magunk is rendszeresen járunk edzeni, és ezt a tudásunkat akarjuk weboldallal bemutatni. Az oldal nem csak a konditerem szolgáltatásait mutatja be, hanem hasznos funkciókkal segíti a felhasználókat, mint például órarendek, edzői profilok és online bejelentkezési lehetőségek.

A projekt szakmai célja a modern webfejlesztési technológiák alkalmazásával egy felhasználóbarát, esztétikus és könnyen navigálható weboldal elkészítése. Ezzel nemcsak a felhasználói élményt kívánjuk növelni, hanem a konditerem népszerűségét is elősegíteni, hogy minél több embert érjenek el a fontos információk.

## Célközönség

Kinek ajánljuk a projektmunkánkat?

Azoknak az embereknek akik szeretnének egy kiegyensúlyozottabb életmódra váltani, mozogni, egészségesen táplálkozni és még segítséget is kérnének tapasztalt edzőktől.Akik szeretnének belemélyedni például a testépítés rejtelmeibe vagy éppen bármilyen sportot ki akarnak próbálni akár hosszútávon még űzni is.

## Rendszerkövetelmény(frontend)

(klienskövetelmény)

### Böngészők követelményei:

|  |  |  |
| --- | --- | --- |
| Böngészők | Operációs rendszerek | Hardver |
| Google Chrome | * **Windows:** 7-11(32 és 64bit) * **macOS:** 10.11 vagy újabb * **Linux:** legtöbb verziója alkalmas * **Android:** 4.4 vagy újabb * **iOS:** 12.0 vagy újabb | * **Processzor:** minimum 1GHz * **RAM:** Legalább 2GB * **Tárhely:** minimum 350MB szabad tárhely szükséges |
| Mozilla Firefox | * **Windows:** 7-11(32 és 64bit) * **macOS:** 10.12 vagy újabb * **Linux:** legtöbb verziója alkalmas * **Android:**5.0 vagy újabb * **iOS:** 13.0 vagy újabb | * **Processzor:** minimum 1GHz * **RAM:** Legalább 2GB * **Tárhely:** minimum 200MB szabad tárhely szükséges |
| Microsoft Edge | * **Windows:** 7-11(64bit) * **macOS:** 10.12 vagy újabb * **Linux:** legtöbb verziója alkalmas * **Android:** 4.4 vagy újabb * **iOS:** 12.0 vagy újabb | * **Processzor:** minimum 1GHz * **RAM:** Legalább 2GB * **Tárhely:** minimum 250MB szabad tárhely szükséges |
| Safari | * **macOS:** 10.12 vagy újabb * **iOS:** 11.0 vagy újabb | * **Processzor:** minimum 1GHz * **RAM:** Legalább 2GB * **Tárhely:** minimum 200MB szabad tárhely szükséges |
| Opera | * **Windows:** 7-11(32 és 64bit) * **macOS:** 10.12 vagy újabb * **Linux:** legtöbb verziója alkalmas * **Android:** 5.0 vagy újabb * **iOS:** 11.0 vagy újabb | * **Processzor:** minimum 1GHz * **RAM:** Legalább 2GB * **Tárhely:** minimum 200MB szabad tárhely szükséges |

## Rendszerkövetelmény(backend)

(szerverkövetelmény)

### XAMPP:

#### ****Operációs rendszer:****

* + **Windows:** 7, 8, 10, 11 (32 és 64 bit)
  + **Linux: bármely verziója alkalmas a XAMPP futtatásához**
  + **macOS:** (macOS 10.12 vagy újabb)
* **Hardver:**
  + **Processzor:** Minimum 1Ghz
  + **RAM:** Legalább 2GB, de inkább legyen 4GB-nál nagyobb
  + **Tárhely:** Nem kevesebb mint 400MB-os szabad tárhely csak XAMPP-hoz

### Visual Studio Code:

#### ****Operációs rendszer:****

* + **Windows:** 7, 8, 10, 11 (64 bit)
  + **Linux: sok verziója alkalmas a Visual Studio Code futtatásához**
  + **macOS:** (macOS 10.11 vagy újabb)
* **Hardver:**
  + **Processzor:** Legalább 1Ghz, de erősebb processzor ajánlott
  + **RAM:** Minimum 1GB, de ajánlott a 2GB-nál nagyobb
  + **Tárhely:** Legyen minimum 200MB-os szabad tárhely
* **Szoftver:**
  + **.NET Framework(Windows):** Legalább 4.5 vagy újabb verzió szükséges

# Fejlesztői dokumentáció

## Szoftverek amiket használtunk

### Windows 10

A Windows 10 az az operációs rendszer, ami a legismertebb és a legfelkapottabb most már nem a legújabb, de talán a legmegbízhatóbb. Más operációs rendszert nem használtunk mivel a Windows 10-hez szoktunk hozzá.

### Visual Studio Code

Egy elég felkapott kódszerkesztő szoftver és ez sem véletlen, mivel nagy segítséget nyújt a kódok írásában. Sok mintával mutatja az utat a megfelelő program felé és ezt az egyet használtuk a tanáraink által.

### Xampp

Lehet hogy nem a xampp az legjobb adatbázis kezelő, de mivel mi ezt ismerjük a legjobban mert már lassan 3 éve ezzel dolgozunk ezáltal erre esett a választásunk.

## Programozási nyelvek, amiket használtunk

### HTML

**H**yper**T**ext **M**arkup **L**anguage, vagyis HTML, egy programozási nyelv, amelyet weboldalak készítésére és struktúrájuk meghatározására használnak. A HTML segítségével a fejlesztők különböző elemeket, mint szövegek, képek és linkek, helyezhetnek el a weboldalon. A HTML dokumentumok címkékből, azaz tag-ekből állnak, melyek segítségével a böngészők értelmezik és megjelenítik a tartalmat. A HTML alapvető technológia a webfejlesztésben, és más eszközökkel, mint CSS és JavaScript, együttműködve hozzájárul a modern weboldalak létrehozásához.

### CSS

**C**ascading **S**tyle **S**heets egy stíluslap nyelv, amely a HTML weboldalak kinézetének formázását segíti. Segítségével meghatározhatjuk a betűszíneket, háttérképeket, margókat és egyéb dizájnelemeket. A CSS lehetővé teszi, hogy a weboldalak megjelenését elkülönítsük a tartalomtól, így könnyebbé válik a karbantartás és a dizájn módosítása.

### Javascript (JS)

 A JavaScript egy magas szintű programozási nyelv, amely weboldalak fejlesztését segíti, és lehetővé teszi a dinamikus elemek hozzáadását, például animációkat. A JavaScript segítségével interaktív funkciókat, választható tartalmakat és más látványos elemeket építhetünk be a weboldalakba, így javítva a felhasználói élményt.

### MySQL

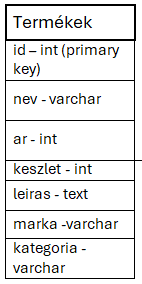
A MySQL nem programozási nyelv, hanem egy relációs adatbázis-kezelő rendszer, amelyet adatbázisok és webalkalmazások kezelésére használnak. A MySQL segítségével adatokat tárolhatunk és kérdezhetünk le, így lehetővé téve dinamikus weboldalak és alkalmazások működését. Az ilyen típusú rendszerek futtatásához gyakran alkalmazzuk a XAMPP-ot, ami egy ingyenes, nyílt forráskódú programcsomag, amely tartalmazza a MySQL-t és más szükséges eszközöket, mint a Apache webszervert, így segít a fejlesztésben és tesztelésben.

## Adatbázis szerkezete

### Az adatbázis táblái:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Termékek |  | Rendelések |  | Vásárlók |
| id – int (primary key) |  | id – int (primary key) |  | id – int(primary key) |
| nev - varchar | vasarloId – int(foreign key) | nev - varchar |
| ar - int | termekId – int(foreign key) |  | email - varchar |
| keszlet - int |  | egysegar - int |  | telszam - varchar |
| leiras - text |  | idopont - datetime |  | jelszo - varchar |
| marka -varchar |  | mennyiseg - int |  |  |
| kategoria - varchar |  |  |  |  |

### Termékek tábla:

id(Termékek azonosítója): Ez a mező az adott termék azonosítója minden hozzáadott termékkel arányosan növekszik. Ez segít megkülönböztetni, azaz azonosítani a termékeket.

nev(Termékek neve): Ez a mező felelős a táblában tárolt termékek neveinek a tárolására, hogy meglehessen őket különböztetni egymástol.

nev(Termékek neve): Ez a mező felelős a táblában tárolt termékek neveinek a tárolására, hogy meglehessen őket különböztetni egymástol.

ar(Termékek ára): Ez a mező a termékek árát tárolja a magyar valutában, az árakat próbáltuk valósághűre csinálni a vizsga időpontjához képest. Ez a mező nagyon fontos az adatbázisban.

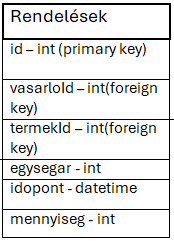
keszlet(Termékek mennyisége): Ez a mező a termékek mennyiségét tárolja például hogy hány darab elérhető még az adott áruból.

leiras(Termékek leírása): Ez a mező hosszab szöveges tartalmakra lett kitalálva amibe bele tudjuk írni a termékek leírásait például tápanyag tartalmát tömegét egyéb hasznos információkat.

marka(Termékek márkája): Ez a mező az adott termék típusokon belül még a márkákat is megkülönbözteti, így tud például a felhasználó alaposabban rákeresni a dolgokra.

kategoria(Termékek típusai): Ez a mező a termékek kategóriáját vagy típusát különíti el így tudunk rákeresni a táplálék kiegészítőkre a felszerelésekre és az edzéstervekre is külön.

### Rendelések táblái

id(Rendelések azonosítója): Ez a mező az adott rendelés azonosítója minden plusz rendeléssel arányosan növekszik. Ez segít megkülönböztetni, azaz azonosítani a különböző rendeléseket.

vasarloId(Vásárlók azonosítója): Ez a mező az adott vásárlót azonosítja minden egyes vásárlóval arányosan növekszik. Ez segít megkülönböztetni a különböző vásárlókat.

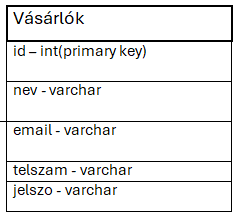
termekId(Termékek azonosítója): Ez a mező az adott termék azonosítója minden hozzáadott termékkel arányosan növekszik. Ez segít megkülönböztetni azaz azonosítani termékeket. A termékek tábla id azonosítójának az idegen kulcsa.

egysegar (Rendelések összára): Ez a mező felelős a rendelések összárának a tárolására. Így könnyebben tud vele számolni az algoritmus. A mező adatát vásárló számára is látható a rendelési kosárban

idopont (Rendelés ideje): Ez a mező felelős a leadott rendelések időpontjának a tárolására. Ez az adat fontos a vásárló, de még fontosabb a weboldalt üzemeltetőknek is.

mennyiseg (Rendelések mennyisége): Ez a mező a rendelések mennyiségéért felel így tudjuk tárolni a rendelések számát és ez miatt elengedhetetlen az egysegar mező kivitelezéséhez is.

### Vásárlók táblái:

id(Vásárlók azonosítója): Ez a mező felelős a vásárlók azonosításáért minden egyes vásárlóval arányosan növekszik az értéke. Ez segít megkülönböztetni azaz azonosítani a különböző vásárlókat.

nev (Vásárlók nevei): Ez a mező felelős az adott vásárló nevének a tárolására. A mező adatait a felhasználó tölti fel a regisztráció során

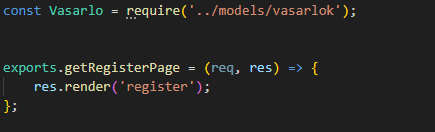
email (Vásárlók email címe): Ez a mező felelős az adott vásárló email címének a tárolásáért. Ezt a mezőt is a vásárló tölti meg a regisztráció idejében.

telszam (Vásárlók telefonszámai): Ez a mező az adott vásárló telefonszámának a tárolását végzi el. Szintén a vásárló tölti fel mikor regisztrál a weboldalra.

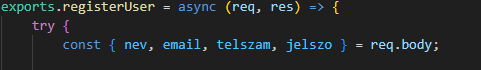
jelszo (Vásárlók jelszavai): Ez a mező az adott vásárló jelszavának a tárolását végzi el. Ugyanúgy a vásárló adja meg a mező adatait a regisztráció során.

## Algoritmusok a weboldalon

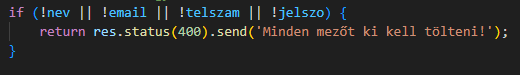
### Regisztráció(controller):



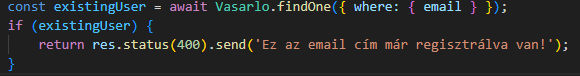
A regisztrációs controllerünkben az első sor oldotta meg a Vasarlo modell importálását. A másik kettő pedig a regisztrációs oldal megjelenítéséért felel GET kéréssel.



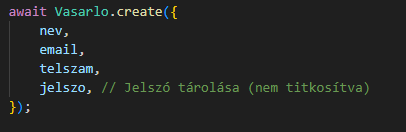
Ez a POST kérés dolgozza fel a regisztráció során megadott adatokat. A nevet, email címet, telefonszámot és a jelszót.



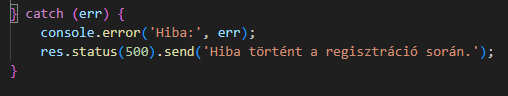
A szöveg felett lévő if egy feltételes ellenőrzés. Minden mező kitöltését ellenőrzi, így nem lesznek üres adatok az adatbázisban.



Ebben a kódban azt ellenőrizzük hogy az email cím vagy a felhasználó létezik-e, ha nem akkor 400-as hibát kapunk és kiírja a képernyőre a felhasználónak hogy az email címe már fellelhető az adatbázisunkban.

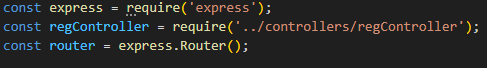


A fent lévő kódban a vásárlót adatait hozzuk létre az adatbázisban. A vásárló nevét, email címét, telefonszámát és a jelszavát viszont azt nem titkosítva, mivel nem ad hozzá olyan sokat a projektmunkánkhoz.

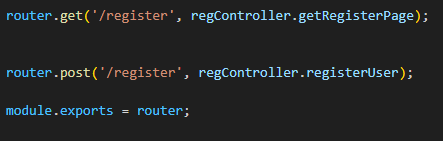


Az alábbi catch blokk a hibákat jelzi a felhasználónak ha történik valami nem odavaló a regisztráció során. Ha hibát talál akkor küld a felhasználónak egy 500-as általános hibaüzenetet.

### Regisztráció(router):

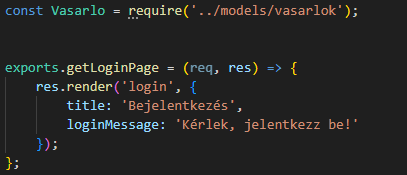


A routerrel a regisztrációs controller importálásával lehetővé teszi az oldal műveleteinek kezelését.

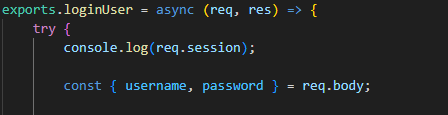


Az első sor a regisztrációs oldal megjelenítéséért felel GET kéréssel. A második pedig az adatok feldolgozását segíti POST kéréssel. A module.exports = router; pedig azért felelős hogy a router objektumot exportálja.

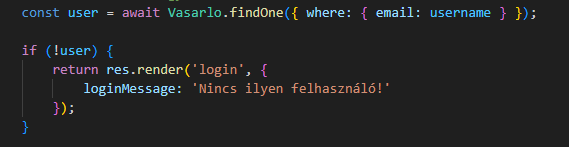
### Bejelentkezés(controller):



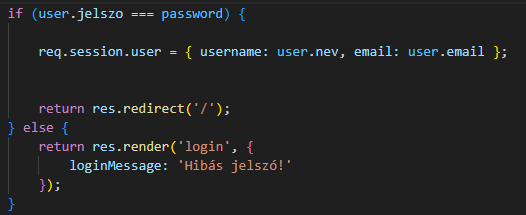
Az első sor a Vasarlo modell importálása. A képen látható maradék kódok a bejelentkezési oldal megjelenítéséért felel még egy üzenetet is küld a vásárlónak a loginMessage paranccsal.



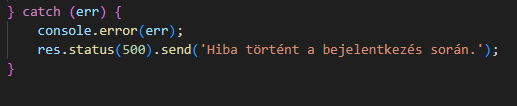
A fenti kód a bejelentkezés feldogozását csinálja meg, ami a felhasználónévből(email) és a jelszóból áll.



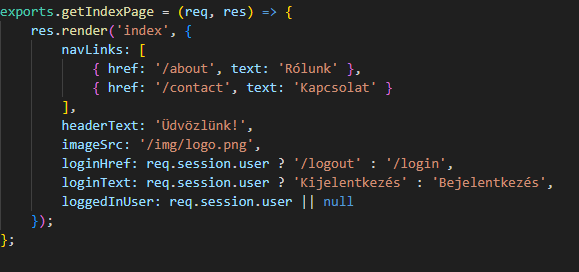
A user constructor megkeresi a beírt felhasználót és ha létezik akkor tovább megy a jelszó feldolgozására, viszont ha nem akkor kiír egy loginMessageet hogy „Nincs ilyen felhasználó!”.



A felső if else elágazásnál azt ellenőrizzük hogy a jelszó helyes-e. Ha helyes a jelszó akkor elmentjük a sessionbe a felhasználó nevet és az email címet. A return kód sorral pedig átirányítjuk a főoldalra. Viszont ha hamis a jelszó akkor kapunk gey hiba üzenetet hogy „Hibás jelszó!”.



A következő sor hibát jelez a bejelentkezés során 500-as hibakóddal.



A felső képen lévő kód részlet a bejelentkezés után új fület hoz létre, ami webshop névre hajaz. A webshop fülön keresztül lehet a termékeket megvásárolni a bejelentkezett felhasználóknak. A bejelentkezés után megjelenik a profil és a kijelentkezés opció is a weboldalon.

### Regisztráció(router):



A regisztrációs router felelős a bejelentkezések és a regisztrációs kérések kezelésééert. A router.get(’/login’) sor a bejelentkezés útvonalát szolgálja ki. A router.get(’/logout’) törli a sessiont és utána visszairányít a bejelentkezési weboldalra. Bejelentkezés feldolgozását a router.post végzi el, ugye POST kéréssel.

# Felhasználói dokumentáció

# Források

# Köszönetnyilvánítás

Szeretnénk kifejezni őszinte köszönetünket mindazoknak, akik hozzájárultak a projektünk sikeres megvalósításához.

Először is, hálásak vagyunk tanárainknak, Baksa László, Anisity Attila és Balogh Gábor osztályfőnök aki folyamatos támogatásával, iránymutatásával és értékes visszajelzéseivel segítette munkánkat a projekt során.

Munkánk során mindannyian hozzájárultunk a tervezéshez, fejlesztéshez és teszteléshez, és az együttműködésünk eredményeként egy sikeres projektet valósítottunk meg.

Végül szeretnénk megköszönni családjainknak és barátainknak a támogatást és a türelmet, amelyet a projekt során nyújtottak.

Hálásak vagyunk, hogy mindezek a segítségek lehetővé tették számunkra, hogy értékes tapasztalatokat szerezzünk a webszerkesztés területén.