

Debreceni Szakképzési Centrum
Baross Gábor Középiskolája és Kollégiuma
Debrecen, 4030
Budai Ézsaiás u. 8/A.

A SZAKKÉPESÍTÉS AZONOSÍTÓ SZÁMA:54 213 05
SZAKKÉPESÍTÉS MEGNEVEZÉSE: SZOFTVERFEJLESZTŐ

Záródolgozat címe:

Kong Burger

Záródolgozat készítője:

Miskolczi Dávid

Konzulens: Kovács István Zoltán

Konzultációs időpontok:

Debrecen, 2023.

Tartalomjegyzék:

1. Bevezetés:	3
Témaválasztás indoklása:	3
Elméleti és gyakorlati jelentőség:	3
2. Fejlesztői dokumentáció:	4
Weblapon használt adatszerkezetek:	4
Az adatbázis egyedi táblái és azok mezői:	5
A user tábla:	5
A termékek tábla:	6
A kosár tábla:	7
3. Egyedi algoritmusok az oldalon:	8
A regisztráció algoritmus:	8
A beléptetés algoritmus:	8
A kijelentkezés algoritmus:	9
A végösszeg számlálás algoritmus:	9
4. JavaScript funkciók:	10
Regisztráció funkció a regisztráció gombhoz:	11
Felhasználó és rang kiírása:	13
Navigációs fül:	13
Kosárba helyezés:	15
5. Teszt dokumentáció:	20
Az oldal tesztelése és a felmerülő hibák:	20

6. Fejlesztési lehetőségek:	21
7. Webes felület célja:	22
8. Hardware és szoftver igény:	23
Telepítés és indítás:	23
9. Az oldalon elérhető funkciók:	24
Bejelentkezés és regisztráció:	24
Főoldal:	25
Burgereink:	26
Kosár:	27
10. Admin felület:	30
Admin felület kinézete:	30
Új hamburger feltöltése:	30
11. Szakdolgozat célja:	32
Megvalósítás és megállapítások:	32
Célkitűzések, amik nem sikerültek teljességgel	32

1. Bevezetés:

Témaválasztás indoklása:

Tulajdonképpen azért választottam ezt a témát, mert a webfejlesztés közelebb áll hozzám, mint maga a programozás.

Választásom egy hamburgerező weboldalra esett, mert véleményem szerint manapság egyre elterjedtebb az online felületen történő ételrendelés. Szakdolgozatom célja, hogy az emberek jobban meg ismerjék az interneten történő ételrendelést és ennek menetét ennek érdekében szakdolgozatom egy internetes vásárlási felületet biztosít, melyben kedvükre válogathatnak hamburgerjeink közül. Jelenleg számítógépes rendszerüzemeltető szakmával rendelkezem ezért számomra egyértelmű volt, hogy valami nagyobb fába vágjam a fejszémet ennek köszönhetően jött egy webshop alapú ötlet.

Elméleti és gyakorlati jelentőség:

Elméletben az oldal elkészítése annyi jelentőséggel bírhat, hogy megtanulhatunk elkészíteni egy beléptető rendszert és egy regisztrációs felületet megmutatja, miként működik a bejelentkezés és az ezt követő munkamenet. Egy ilyen weboldal készítése során elsajátíthatjuk a Bootstrap kártyák használatát, az oldalak közötti navigációt, és akár az egyedi slideshow készítését is. A webes oldalak felépítésébe is bele láthatunk, illetve a háttérben álló adatbázis felépítésébe is. Elsajátíthatjuk a jelszavak biztonságos tárolását, mely egy ilyen weboldal készítésénél kihagyhatatlan. A weblap kódolása teljes oda figyelmet és megfelelő időbeosztást igényel, hogy mindenre elegendő időt és energiát tudjunk bele fektetni és weblapunk kinézete és backendje a lehető legtökéletesebb legyen. A legelső lépés a weblap témájának elképzelése volt, mely számomra igen rövid idő alatt egyértelművé vált. Miután a téma megvolt, neki is láttam a kódolásnak, ami számomra igazán izgalmas volt, a kinézet, a navigációs terület és kártyák megjelentítése egyaránt egy izgalmas és érdekes megméréstetés volt.

2. Fejlesztői dokumentáció:

Mivel egy weboldal készítését választottam, ezért elengedhetetlen az adatbázis kezelés, amelyben segítségemre volt :

- A **phpMyAdmin Database Manager 5.2.0**-es verziója, ami az adatbázis létrehozásához és kezeléséhez szükséges.
- A **MySQL Database 10.4.27**-es verziója, ami az oldalak háttérében lévő adatbázist szolgáltatja.

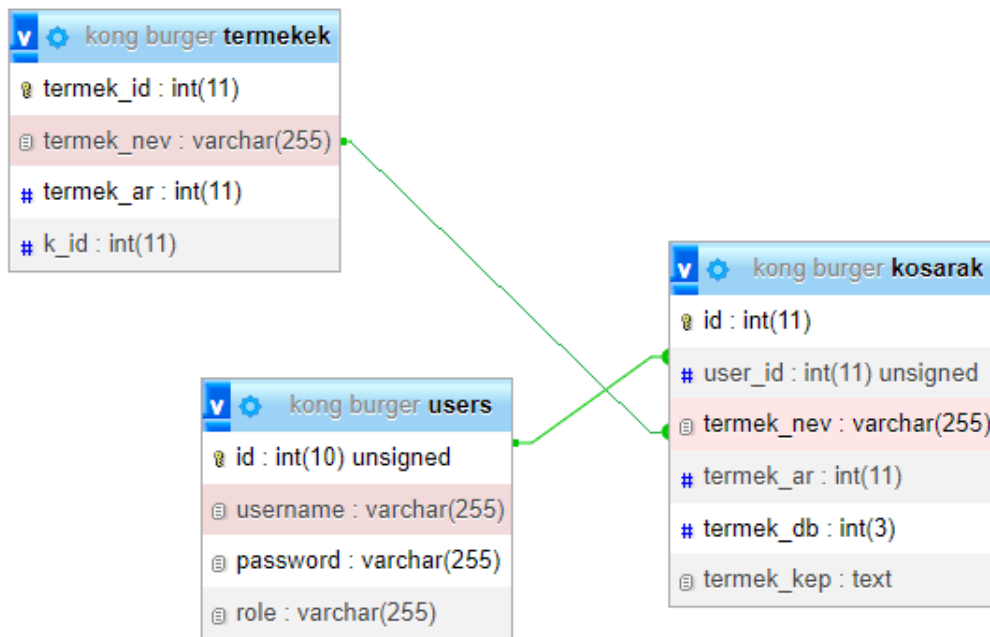
Továbbá a weblap során felhasználtam:

- **Bootstrap 5** elemeket
- **NodeJs** konzol parancsokat
- **CSS** stílus lapot
- **JavaScript** kódokat

Weblapon használt adatszerkezetek:

Mint minden olyan weboldalon, ahol regisztrációt várunk el felhasználóinktól használunk integer-t és string típusú adatokat. Ez alól az Kong Burger sem kivétel. A következő oldalon látható az adatbázis ábrája, ami a weboldal adattárolásáért felel benne a következő adattípusokkal integer illetve varchar. Maga az adatbázis egy összefüggő adatszerkezet, amely táblákból és azon belül mezőkből, vagyis különböző rekordokból áll.

Az adatbázis egyedi táblái és azok mezői:



1. ábra: Adatbázis

Az oldalon beérkező adatok tárolására a kong burger adatbázis nyújt lehetőséget. Az adatok külön feldolgozó programokon jutnak el ide.

A user tábla:

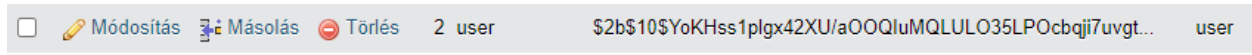
kong burger users	
id : int(10) unsigned	
username : varchar(255)	
password : varchar(255)	
role : varchar(255)	

2. ábra: users tábla

Elsődlegesen a felhasználók alap adatainak a tárolására szolgál. Első mezője az id, ami egész szám (integer) típusú, amely önmagától generálódik, valahányszor egy újabb felhasználó regisztrál az oldalon. Maga ez a mező a miatt releváns, mert a későbbiek folyamán nagyméretű

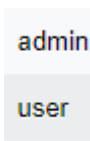
funkciója lesz a felhasználók beléptetésében az oldalon, ezenkívül ez lesz a felhasználó azonosítója is, és egyben ez különbözteti meg a rekordokat egymástól. A tábla második mezője a username, a felhasználók neveinek tárolására szolgál, rövid szöveg típusban (varchar).

A tábla harmadik mezője a password, a felhasználók jelszavainak tárolását segíti elő, mely titkosított formátumban jelenik meg az adatbázisban.



A tábla negyedik mezője a felhasználók rangját jelzi rövid szöveg típusban (varchar), mely rangok a következők lehetnek:

- user (Átlagos felhasználó)
- admin (Adminisztrátor)



Lényegében ez a tábla az alapja a felhasználók regisztrációjának és magának a bejelentkezésnek.

A termékek tábla:

kong burger termékek				
termek_id : int(11)	termek_id	termek_ar	termek_nev	k_id
termek_nev : varchar(255)	1	1790	Csibe Burger	1
termek_ar : int(11)	2	2190	Big Kong	2
k_id : int(11)				

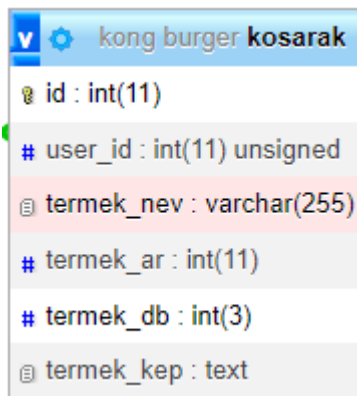
3. ábra: termékek tábla és adatai

A tábla első mezője szintén egy integer típusú mező, amely automatikusan generált azonosítót ad a terméknek. Mivel automatikusan generált, ezért ha új termék kerül a táblába, akkor az is egyedi azonosítóval lesz ellátva. A tábla második mezője nem más, mint maga a termék neve, amely egy varchar típusú mezőben lett el tárolva.

A tábla harmadik mezője a termék árát tartalmazza, amely egy integer típusú mezőben lett el tárolva.

Végül, de nem utolsó sorban, a tábla negyedik mezője a kosár egyedi azonosítóját menti, mely szintén integer típusú mezőben lett el tárolva.

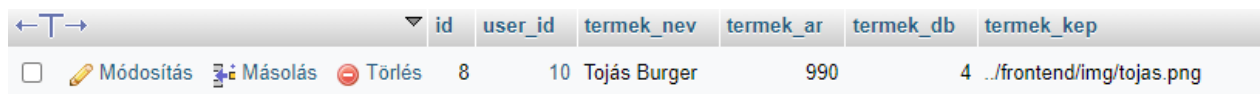
A kosár tábla:



kong burger kosarak	
id	int(11)
user_id	int(11) unsigned
termek_nev	varchar(255)
termek_ar	int(11)
termek_db	int(3)
termek_kep	text

4. ábra: Kosarak tábla

A kosarak tábla arra szolgál, hogy megtudhassuk, hogy egy adott felhasználó milyen terméket rendelt illetve, hogy az adott termék mennyibe kerül, és hány darabot rendelt az adott felhasználó. Az id mező a rendelés egyedi azonosítóját menti le, ami alapján könnyebben be azonosítható a rendelés. A user_id mező a felhasználó egyedi azonosítóját jelzi, amit regisztrációkor kapott. A termék_nev mező annak a terméknek a nevét menti az adatbázisba, amit a felhasználó rendelt. A termék_ar mező a rendelni kívánt termék árát menti le az adatbázisba. A termék_db a rendelni kívánt termék mennyiségét menti le, ahogy a példában is látszik, a felhasználó 4db tojás burgert rendelt. A termék_kep mező a megrendelt termék képének a nevét menti le az adatbázisba ez is a könnyebb be azonosítást szolgálja.



	id	user_id	termek_nev	termek_ar	termek_db	termek_kep
<input type="checkbox"/> Módosítás Másolás Törlés	8	10	Tojás Burger	990	4	../frontend/img/tojas.png

5. ábra: Rendelés adatbázisba

3 .Egyedi algoritmusok az oldalon:

A következő algoritmusok a program elkészítésének sorrendjében lettek leírva a szakdolgozatban.

A regisztráció algoritmus:

Célja, hogy a felhasználónak legyen lehetősége regisztrálni, majd ezt felhasználva a beléptetés algoritmusával be tudjon lépni és ténykedni az oldalon. Lépései sorrendben:

- A felhasználó által kötelezően kitöltendő mezők (felhasználónév, jelszó) ellenőrzése nem maradt-e üres, ha igen hibaüzenet küldése.
- A használni kívánt felhasználónév leellenőrzése egy eldöntés tétellel nincs-e már használatban az oldalon amennyiben már használt hibaüzenet küldése.
- A jelszó hosszának leellenőrzése, hogy az legalább 8 karakter. Amennyiben nincs, akkor felugró ablakban „A jelszó túl rövid” szöveg jelenik meg.
- Amennyiben a felhasználó mind a két kötelező mezőt kitöltötte, akkor a jelszó titkosítva fel kerül az adatbázisba, amelyhez **bcrypt** csomagot használunk.

A beléptetés algoritmus:

A beléptetés az adatbázisból lekéri a bejelentkezni kívánó személy felhasználó nevét és jelszavát, amennyiben valamelyik nem egyezik, akkor egy hiba üzenetet kapunk. Viszont ha mind a kettő megfelelő, akkor az oldal beléptet, és már a főoldalon találhatjuk magunkat.

A beléptetés kódja:

```
app.post('/login', (req, res) => {
  const { username, password } = req.body;
  connection.query('SELECT * FROM users WHERE username = ?', [username],
(error, results) => {
    if (error) throw error;
    if (results.length === 0) {
      return res.status(400).json({ message: 'A felhasználónév vagy jelszó
hibás!' });
    }
  });
});
```

```

    }
    const user = results[0];
    bcrypt.compare(password, user.password, (error, result) => {
        if (error) throw error;
        if (!result) {
            return res.status(400).json({ message: 'A felhasználónév vagy jelszó hibás!' });
        }
        req.session.username = user.username;
        req.session.role = user.role;
        res.json({ username: req.session.username, role: req.session.role });
    });
});
});

```

6. ábra: Beléptetés algoritmus

A kijelentkezés algoritmus:

A kijelentkezésnél a weboldal megvizsgálja, hogy van-e bejelentkezett felhasználó. Ha nincs, hiba üzenetet kapunk, viszont ha van, akkor a felhasználó sikeresen kijelentkezhet.

A kijelentkezés kódja:

```

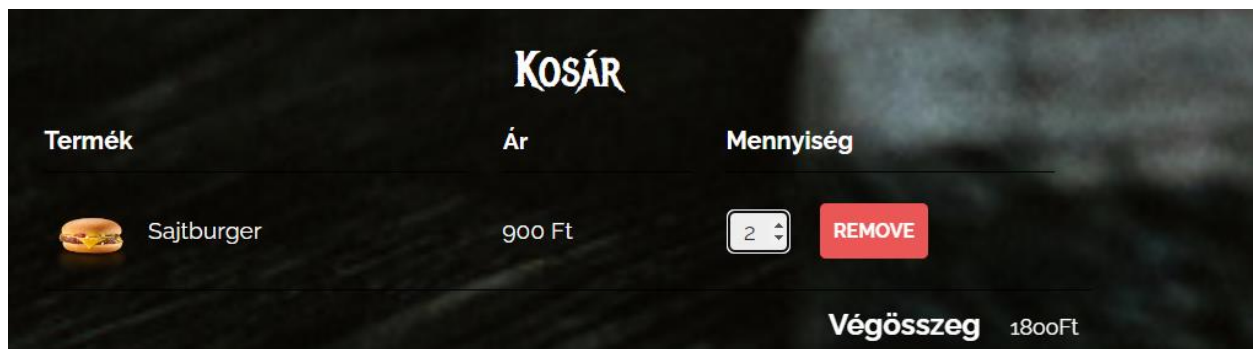
app.post('/logout', (req, res) => {
    if (!req.session.username) {
        return res.status(400).json({ message: 'Nincs bejelentkezett felhasználó' });
    }
    req.session.destroy(() => {
        res.json({ message: 'Sikeres kijelentkezés!' });
    });
});

```

7. ábra: Kijelentkezés kódja

A végösszeg számlálás algoritmus:

A végösszeg számlálás kódját JavaScriptben írtam. A lényege, hogy két különböző vagy akár több termék árát számolja ki (termékár + termékár = végösszeg). A végösszeg akkor is változik, ha egy adott termékből több darabot helyezünk a kosárba.



8. ábra: Kosár és végösszeg

A végösszeg számítás kódja:

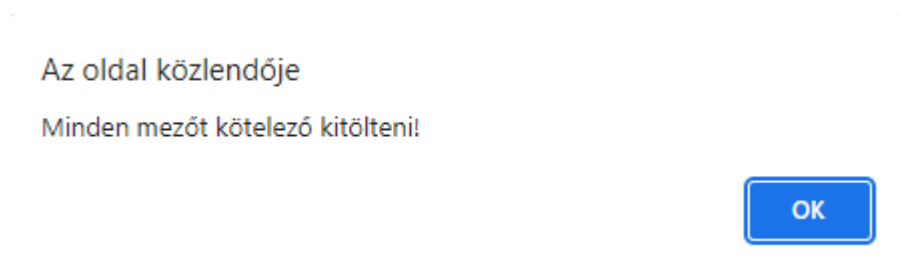
```
function updateCartTotal() {
    var cartItemContainer = document.getElementsByClassName('cart-items')[0]
    var cartRows = cartItemContainer.getElementsByClassName('cart-row')
    var total = 0
    for (var i = 0; i < cartRows.length; i++) {
        var cartRow = cartRows[i]
        var priceElement = cartRow.getElementsByClassName('cart-price')[0]
        var quantityElement = cartRow.getElementsByClassName('cart-quantity-
input')[0]
        var price = parseFloat(priceElement.innerText.replace('Ft', ''))
        var quantity = quantityElement.value
        total = total + (price * quantity)
    }
    total = Math.round(total * 100) / 100
    document.getElementsByClassName('cart-total-price')[0].innerText =
total+'Ft'}
```

9. ábra: Végösszeg számítás

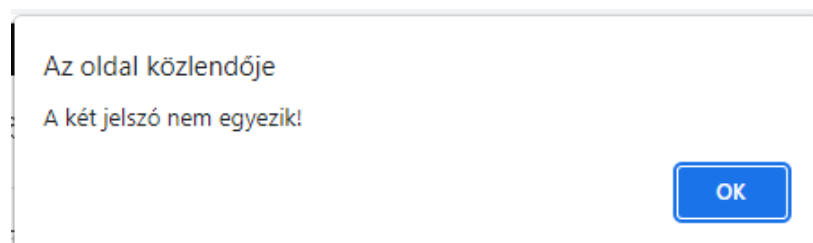
4. JavaScript funkciók:

Regisztráció funkció a regisztráció gombhoz:

A regisztráció gombra kattintáskor megjelenik egy ablak, melyben a program be kéri a felhasználónevet, a jelszót és a jelszót ismét. Ezután a program megvizsgálja, hogy a két be írt jelszó egyezik-e, illetve hogy minden mező ki van-e töltve. Amennyiben a mezők nincsenek ki töltve, vagy a két jelszó nem egyezik, abban az esetben ezek a hiba üzenetek fognak minket fogadni.

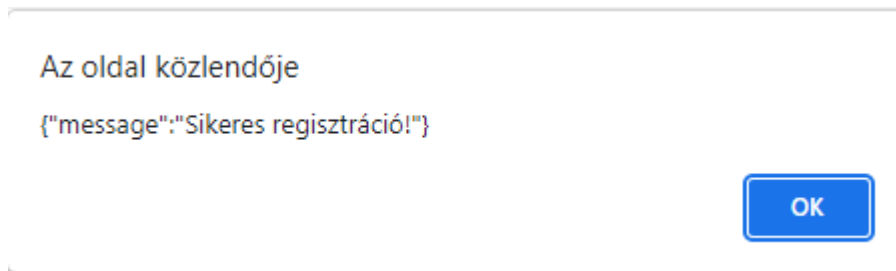


10. ábra: Üres mező alertje



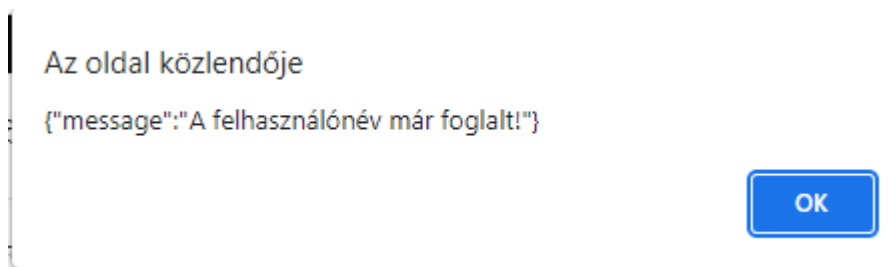
11. ábra: A jelszó nem egyezik alert

Viszont ha mindkét mezőt ki töltöttük, és a két jelszavunk is megegyezik, akkor már jöhet is a következő rendszer üzenet.



12. ábra: Sikeres regisztráció alert

Amennyiben viszont már felhasználónévünk foglalt ezt a kis hiba üzenetet kapjuk.



13. ábra: Foglalt felhasználónév

A regisztrációs felület rengeteg problémára fel van készítve, így útmutatást adhat a felhasználónak, hogy miért nem tud regisztrálni.

A regisztráció funkció kódja:

```
function registration() {
    if (document.getElementById('regUsername').value != "" &&
document.getElementById('regPassword1').value != "" &&
document.getElementById('regPassword2').value != "") {
        if (document.getElementById('regPassword1').value ==
document.getElementById('regPassword2').value) {
            let datas = {
                username: document.getElementById("regUsername").value,
                password: document.getElementById("regPassword1").value,
            };
            let url = "http://localhost:3000/register";
            let fetchOptions = {
                method: "POST",
                body: JSON.stringify(datas),
                headers: { "Content-type": "application/json; charset=UTF-8" }
            };
            fetch(url, fetchOptions)
```

```

        .then(x => x.text())
        .then(y => {
            alert(y);
            document.getElementById('regUsername').value = "";
            document.getElementById('regPassword1').value = "";
            document.getElementById('regPassword2').value = "";
        });
    }
    else {
        alert("A két jelszó nem egyezik!");
    }
}
else {
    alert("Minden mezőt kötelező kitölteni!");
}
}
}

```

14. ábra: Regisztráció kódja

Felhasználó és rang kiírása:

Véleményem szerint nagyon fontos, hogy egy weboldalon ki legyen írva a felhasználó nevünk, főleg ha mindezt egy jó dizájn mellé tesszük. A program kódom bejelentkezésnél lekéri a felhasználó adatait majd, ki íratja a weboldalra a felhasználónévét és rangját.

ddave (user)

admin (admin)

A felhasználó és a rang kiíratásának kódja:

```

function load() {
    let username = sessionStorage.getItem("username");
    let role = sessionStorage.getItem("role");
    if (username) {
        let loggedInUser = document.getElementById("loggedInUser");
        if (loggedInUser) {
            loggedInUser.innerHTML = ` ${username} (${role}) `;
        }
    }
}
};

```

15. ábra: Rang kiíratás

Navigációs fül:

A navigációs fül az oldalak közötti navigációt segíti elő, ez által lehet például a fő oldalról a kosár földre navigálni. Ez a folyamat egy nagyon egyszerű kis JavaScript kód amit bárki könnyen tud alkalmazni.

```
};  
function nextpage3() {  
    window.location.href = "kosar.html";  
};
```

16. ábra: Navigálás két oldal között

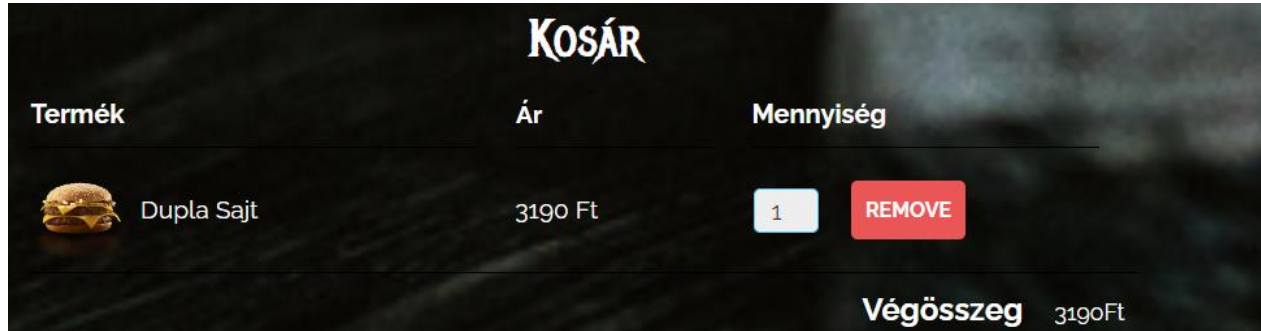


17. ábra: Navigációs fülek

Két oldal közti navigációhoz egy ilyen rövid function függvény szükséges.

Kosárba helyezés:

A webshopom egész kosara JavaScripttel épült fel, ez alól nem kivétel a kosárba helyezés sem. A programkód megvizsgálja, hogy melyik termékre lett kattintva, és azt a kosárba helyezi a hozzá megfelelő képpel, névvel illetve árral.



18. ábra: Kosárba helyezés

Ez a program kód számomra kissé bonyolult volt, de igazán büszke vagyok a végeredményre. A sok különböző osztály segítségével létre jöhetett életem első webshoppos kosara.

A kosárba helyezés kódja:

```
if (document.readyState == 'loading') {
    document.addEventListener('DOMContentLoaded', ready)
} else {
    ready()
}

function ready() {
    var removeCartItemButtons = document.getElementsByClassName('btn-danger')
    for (var i = 0; i < removeCartItemButtons.length; i++) {
        var button = removeCartItemButtons[i]
        button.addEventListener('click', removeCartItem)
    }
    var quantityInputs = document.getElementsByClassName('cart-quantity-input')
    for (var i = 0; i < quantityInputs.length; i++) {
        var input = quantityInputs[i]
        input.addEventListener('change', quantityChanged)
    }
    var addToCartButtons = document.getElementsByClassName('shop-item-button')
    for (var i = 0; i < addToCartButtons.length; i++) {
```



```

        var button = addToCartButtons[i]
        button.addEventListener('click', addToCartClicked)
    }
    document.getElementsByClassName('btn-purchase')[0].addEventListener('click',
purchaseClicked)
}

```

19. ábra: Kosárba helyezés kódja

Ez a programkód átadja a kosárnak egy adott termék nevét, árát és a termék képét.

Kosár ki ürítése:

```

var cartItemContainer = document.getElementsByClassName('cart-items')[0]
cartItemContainer.innerHTML = ''

localStorage.removeItem('cart')

alert('Köszönjük a vásárlást!')
}

function removeCartItem(event, termékTitle) {
    var buttonClicked = event.target
    buttonClicked.parentElement.parentElement.remove()

    // Delete item
    deleteItemFromCart(termékTitle)

    updateCartTotal()
}

function quantityChanged(event, termékTitle) {
    var inputValue = event.target.value
    if (isNaN(inputValue) || inputValue <= 0) {
        inputValue = 1
    }
}

```

20. ábra: Kosár kiürítés kódja

Ezzel a kóddal ki üríthetjük a kosár tartalmát.

Mennyiség frissítése és törlés a LocalStorageból:

```
const quantity = Number(inputValue)
  updateQuantityInCart(termekTitle, quantity)

  updateCartTotal()
}

// Ha frissül a lap töröljük ki az esetlegesen létező,
// "cart" nevű elemet a localStorage-ból
localStorage.removeItem('cart')

function addToCartClicked(event) {
  var button = event.target
  var shopItem = button.parentElement.parentElement
  var title = shopItem.getElementsByClassName('shop-item-title')[0].innerText
  var price = shopItem.getElementsByClassName('shop-item-price')[0].innerText
  // Ez a verzió nem megfelelő mert a következőt fogja vissza adni:
  // http://127.0.0.1:5500/frontend/img/gluten.png
  // var imageSrc = shopItem.getElementsByClassName('shop-item-image')[0].src

  // Ez már valóban a kép elérési útját adja vissza
  // Pl.: ../frontend/img/gluten.png
  var imageSrc =
    shopItem.getElementsByClassName('shop-item-image')[0].attributes.src.value

  setCartInLocalStorage(title, price, 1, imageSrc)

  addItemToCart(title, price, imageSrc)
  updateCartTotal()
}
```

21. ábra: LocalStorage frissítése és törlése

Ha frissül a lap, akkor a helyi tárolóból (LocalStorage) töröljük ki a cart nevű elemet, és annak minden adatát.

A „Cart” nevű Local Storage létrehozása és a termékek hozzá adása:

```
// localStorage cart
function setCartInLocalStorage(title, price, quantity, imageSrc) {
  const priceWithoutFt = Number(price.slice(0, price.length - 3))
  const newCartItem = { title, price: priceWithoutFt, quantity, imageSrc }
  const cartValue = localStorage.getItem('cart')

  // Ha a kosár már létezik
  if (cartValue) {
    // A cartValue egy string, ehhez nem tudunk elemet adni.
    // Atkell "parsolni" hogy az értékéből kinyerjük a tömböt.
    const cartArray = JSON.parse(cartValue)

    // Ne adjuk hozzá a terméket ha már benne van a kosárba
    const termékInCart = cartArray.find(
      ({ title: termékTitle }) => termékTitle === title
    )
    if (termékInCart) return

    // A tömbhöz hozzáadjuk az új elemet
    cartArray.push(newCartItem)

    // A tömböt ismét string-gé alakítjuk és elhelyezzük a
    // localStorage-en belül, "cart" néven.
    localStorage.setItem('cart', JSON.stringify(cartArray))

    console.log(cartArray)
  } else {
    // Üres kosár létrehozása
    const cartArray = []
    // A jelenleg hozzáadandó termék kosárba rakása
    cartArray.push(newCartItem)
  }
}
```

22. ábra: LocalStorage létrehozása és termékek hozzá adása

Ezzel a kóddal létre hozzuk a „Cart” nevű LocalStoraget, és eltároljuk benne a kiválasztott termék adatait.

Vásárlás és eltávolítás:

A vásárlás funkció azt a célt szolgálja, hogy rendelésünket le adhassuk a vásárlás gomb megnyomásával egy felugró ablakban köszönjük meg a vásárlást és az adatok mehetnek is az adatbázisba. Az eltávolítás funkció pedig pont a vásárlás ellentéte, ha valamit bele raktunk a

Vásárlás:



24. ábra: Rendelés mentése adatbázisba

KOSÁR		
Termék	Ár	Mennyiség
		Végösszeg 0Ft

A vásárlás és az eltávolítás kódja:

19

```

var cartItems = document.getElementsByClassName('cart-items')[0]
while (cartItems.hasChildNodes()) {
    cartItems.removeChild(cartItems.firstChild)
}
updateCartTotal()
}
function removeCartItem(event) {
    var buttonClicked = event.target
    buttonClicked.parentElement.parentElement.remove()
    updateCartTotal()
}

```

26. ábra: A vásárlás és az eltávolítás kódja

5. Teszt dokumentáció:

Az oldal tesztelése és a felmerülő hibák:

Az oldal le lett tesztelve 2 fajta böngészővel:

- Google Chrome
- Microsoft Edge

Változást a teszt alatt nem tapasztaltam. Minden teljesen ugyan olyan volt a két böngészőben. A funkciók egyesével tesztelve lettek, és minden egyformán működik a két böngészőben. Az oldal tesztelése során két ember segítségét kértem, egy az informatikai dolgokhoz kevésbé értő és egy számítógép szerelő ismerősömet.

A számítógép szerelő ismerősöm visszajelzése alapján elő került pár hiba, amit javítottam is.

Elő került hibák:

- Regisztráció után hiányzott a felugró sikeres regisztrációs kis ablak, amit egy javascript-es alert('sikeres regisztráció') utasítással tudtam orvosolni.
- Volt olyan kép, amelyet nem jelenített meg az oldal, így nem lehetett tudni, hogy is néz ki az adott hamburger. Ezt egy egyszerű HTML title tag-gel orvosolni tudtam, és újra megjelent a kép.

6. Fejlesztési lehetőségek:

A programozásban egy programot szinte a végtelenségig lehet bővíteni, itt felsorolok egy pár fejlesztési lehetőséget a szakdolgozatommal kapcsolatban:

- Több elérhető nyelv
- Keresés funkció
- Online fizetési lehetőség pl.: PayPal
- A regisztráción belül születési idő megadása
- Egyedi hamburger létrehozása
- Menürendelés
- Ital és több étel rendelése

7. Webes felület célja:

A weboldal célja, hogy egy olyan webes felületet biztosítson, a hamburgerimádók számára ahol tudnak:

- Kedvükre válogatni
- Rendelni
- Egy igazán jót enni

8. Hardware és szoftver igény:

Mivel ez egy webes felület ezért igényel egy eszközt, amivel lehet internetezni, ez lehet laptop vagy asztali számítógép. Továbbá szükséges egy szerver, amire a weboldal kerül, ezen felül szükséges internet hozzáférés, valamint a következő böngészők egyike: Google Chrome, Microsoft Edge.

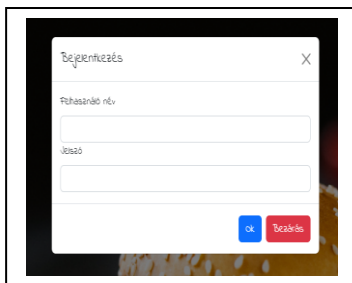
Telepítés és indítás:

Mivel a szoftver webes alapú ezért külön telepítési eljárás csak abban az esetben szükséges, ha még nem rendelkezünk böngészővel. Szükséges az oldal betöltéséhez internet hozzáférés is. Amennyiben rendelkezünk ezzel a két dologgal már csak a megfelelő címet kell beírunk

9. Az oldalon elérhető funkciók:

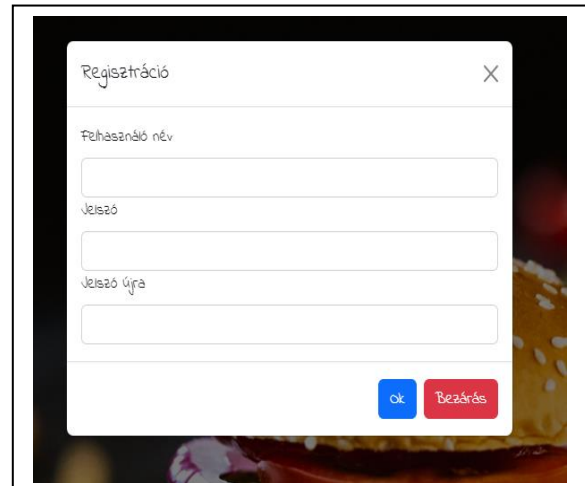
- A rólunk navigációs fülön való böngészés a helyi éttermek között
- Válogatás a különböző és lenyűgöző hamburgerek között
- Étel rendelés

Bejelentkezés és regisztráció:

A screenshot of a web application dialog box titled 'Bejelentkezés' (Login). It contains two input fields: 'Felhasználó név' (Username) and 'Jelszó' (Password). At the bottom right, there are two buttons: a blue 'OK' button and a red 'Bejelentkezés' (Login) button.

A regisztráció az oldalon való bejelentkezéshez és ez által az oldal funkcióinak használatához szükséges elengedhetetlen lépés. A funkció elérése a Bejelentkezési felületen keresztül történik, amely a weboldal jobb oldalán található, a Regisztráció gombon használatával. Erre kattintva a következő felületet kapjuk az oldal közepére tekintve.

A jelszónak 8 karakter hosszúságúnak kell lennie.

A screenshot of a web application dialog box titled 'Regisztráció' (Registration). It contains three input fields: 'Felhasználó név' (Username), 'Jelszó' (Password), and 'Jelszó újra' (Repeat Password). At the bottom right, there are two buttons: a blue 'OK' button and a red 'Regisztráció' (Registration) button.

Főoldal:



27. ábra: Főoldal

Bejelentkezés után a főoldal fogad minket, ahol egy kis reklám szerű diavetítést nézhetünk meg. A főoldalat nem csak arra használhatjuk, hogy a diavetítést nézzük, hanem arra is, hogy el navigáljunk bármelyik másik navigációs fülre.

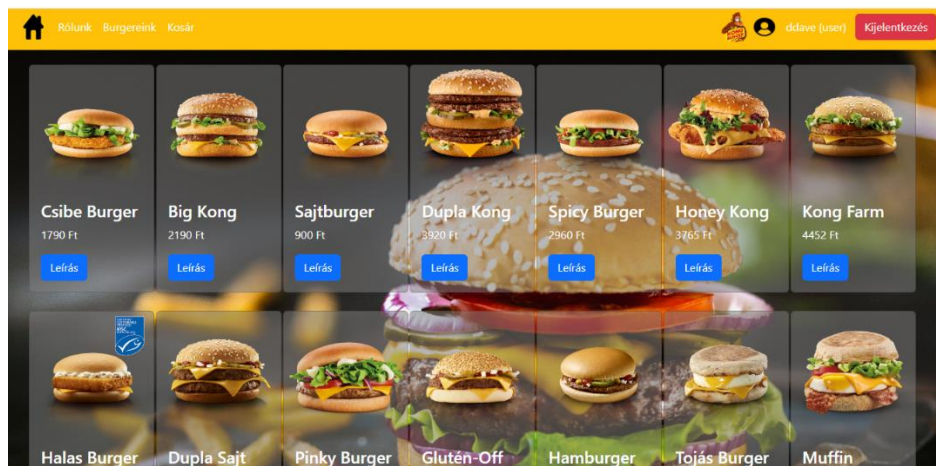
Rólunk:



28. ábra: Rólunk fül

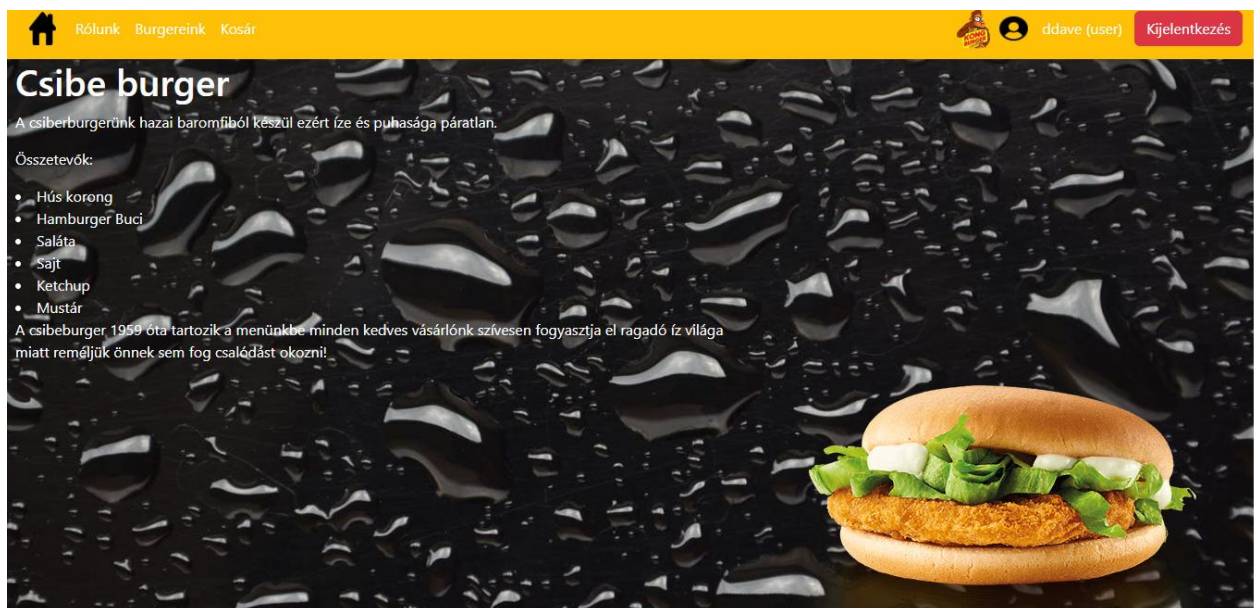
A rólunk fülnek egy igazán egyszerű feladata van. Mondja meg az új vásárlóknak és vendégeknek, hogy kik is vagyunk mi? Hol lehet minket megtalálni? A legfontosabb pedig, hogy mi az elérhetőségünk.

Burgereink:



29. ábra: Burgereink fül

A burgereink fülön megtekinthetik a jelenleg elérhető hamburgereinket. Ez az oldal kényelmes és rugalmas nézelődést kínál a vásárlóink számára. Minden hamburger mögött egy rövid leírás áll melyet, mindenki megtekinthet, ha rá kattint a leírás gombra.

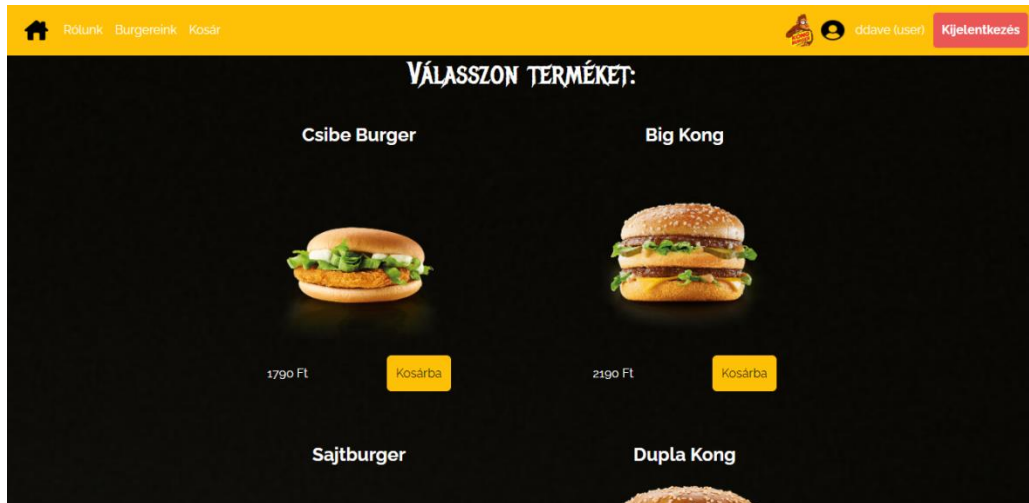


30. ábra: Egy hamburger leírása

A Kong Burger egyedi logóval lett el látva melyet megtalálhatnak a felhasználónév mellett illetve a böngésző címsorában.

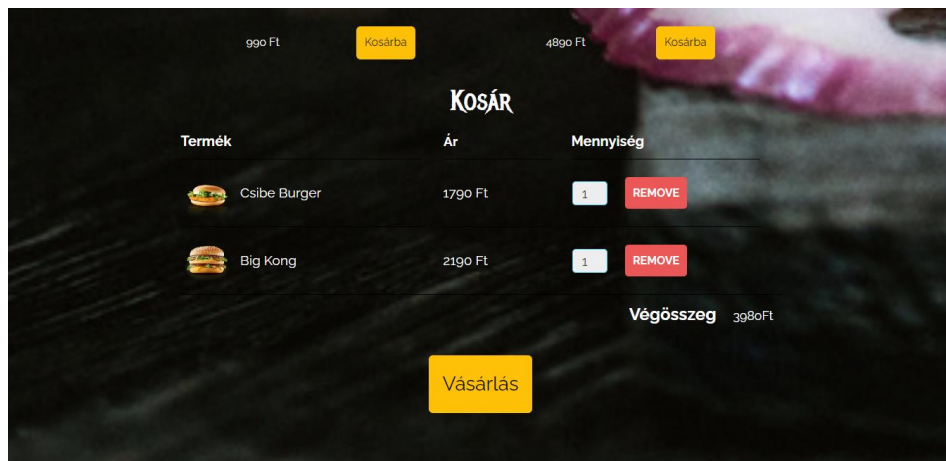


Kosár:



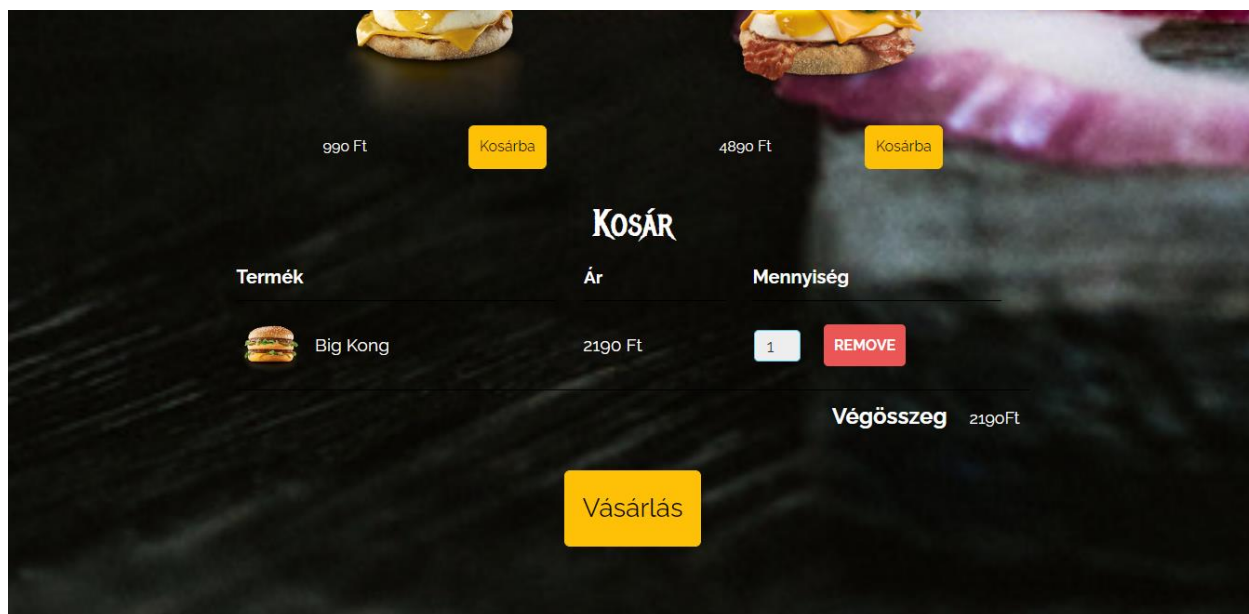
31. ábra: Kosár fül

A kosár fülön ki választhatjuk a számunkra legszimpatikusabb hamburgert, amelyet azonnal kosarunkhoz adhatunk és megrendelhetünk.



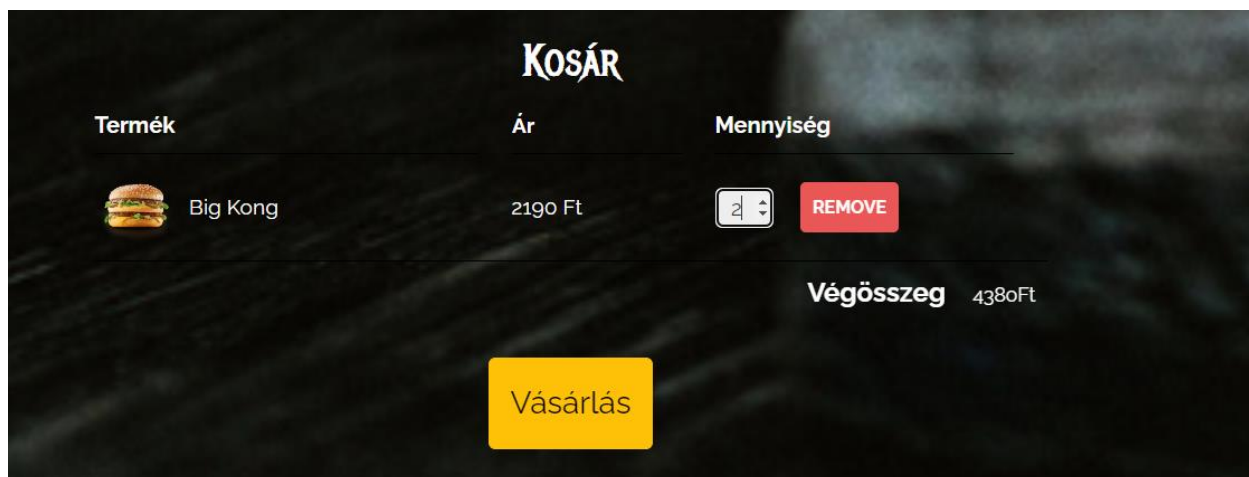
32. ábra: Kosár és rendelés leadás

Jaaj ne! Véletlen hozzáadtunk egy olyan terméket, amit nem szeretnénk megrendelni? Semmi gond a Kong Burgernek erre is van megoldása, hiszen a „REMOVE” gomb megnyomásával könnyen eltávolíthatjuk a nem kívánt terméket.



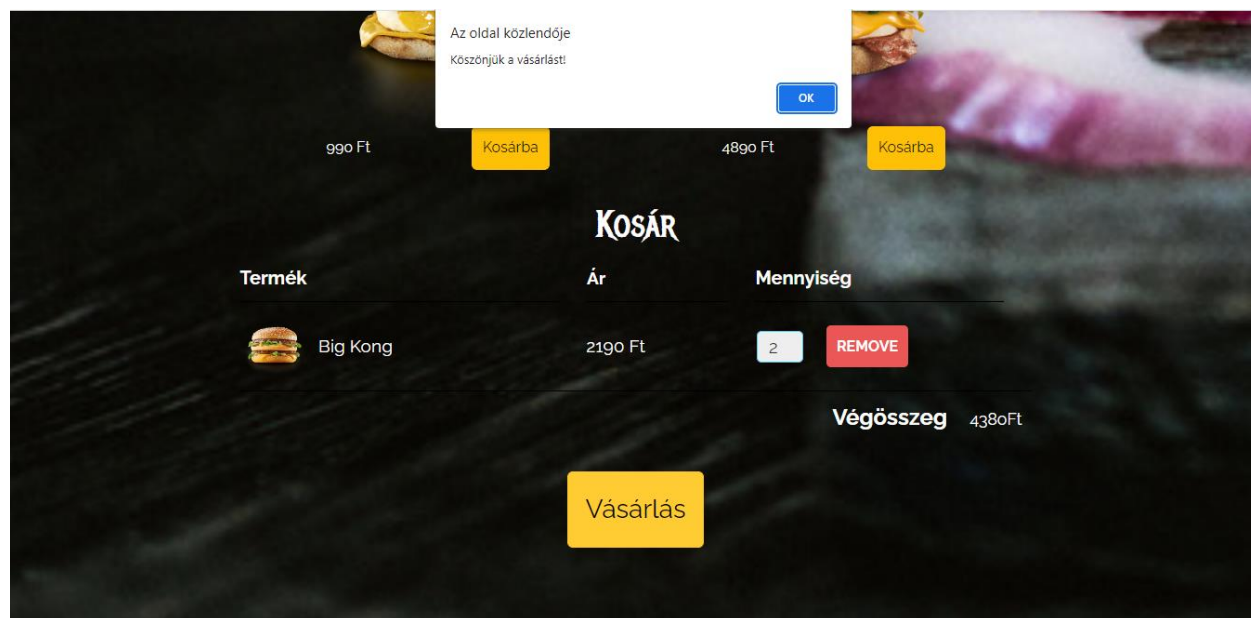
33. ábra: Termék a kosárban

A Kong Burger lehetőséget ad arra, hogy személyre szabott mennyiséget adhassunk meg egy kiválasztott termékből, és még azon sem kell gondolkoznunk, hogy mennyibe fog kerülni, hiszen a weboldal erre is fel van készítve és ki számolja nekünk a végösszeget.



34. ábra: Ki számolt végösszeg

A vásárlás gombra kattintva egy kedves üzenet fogja fogani a vásárlót amiben megköszönjük a vásárlást és a rendelés azonnal az adatbázisba kerül.



☐ ✎ Módosítás 📋 Másolás 🗑 Törlés 9 10 Muffin 4890 4 ../frontend/img/guacamole.png

10. Admin felület:

Admin felület kinézete:

Az admin felület kinézetéért egy belső style tag felel, melyben meg adott a háttérkép minden fontos információja.



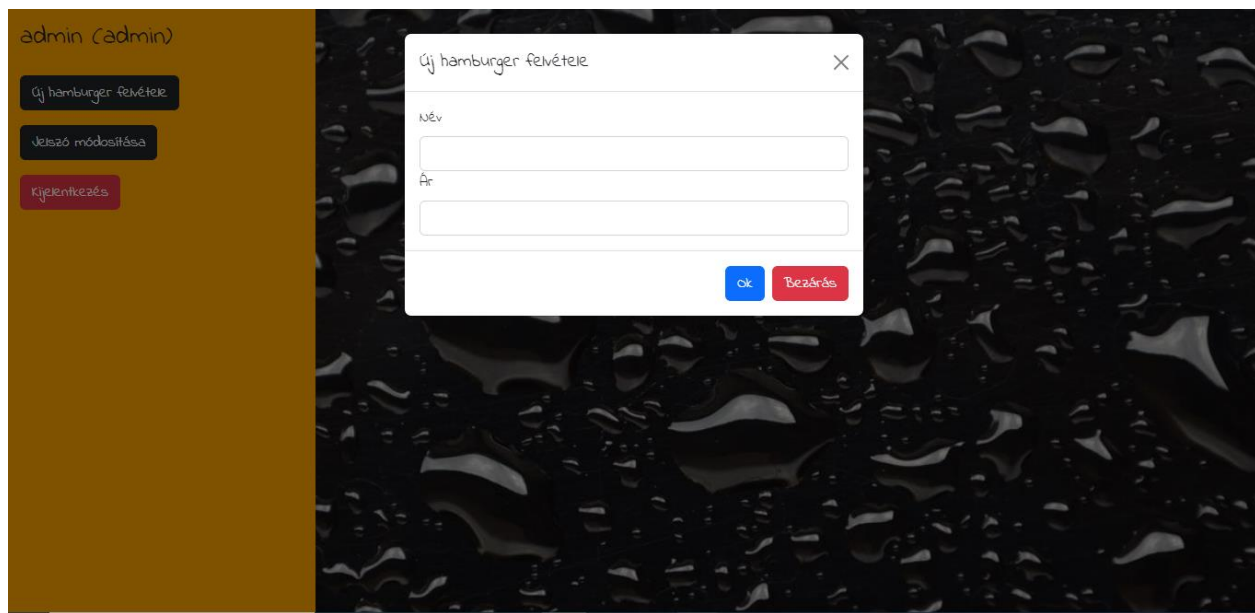
35. ábra: Admin felület főoldala

Mivel az alap felhasználói oldal is narancssárga és fekete dizájnt kapott, ezért úgy gondoltam, hogy az admin felület is legyen hasonló, de mégse ugyan olyan.

Az admin felület lehetővé teszi, hogy új hamburgert tegyünk a többi közé, és azt is, hogy jelszavat változtathassunk, ami valljuk be igazán jó dolog.

Új hamburger feltöltése:

Ha rá nyomunk az új hamburger felvitele gombra, akkor egy modális ablak jelenik meg, amelyben megtudjuk adni a termék nevét és árát.



36. ábra: Új hamburger felvitele

11. Szakdolgozat célja:

A szakdolgozatom célja egy rugalmas ételrendelő weboldal készítése, ahol az emberek kedvükre nézelődhetnek és válogathatnak a termékek közül.

Megvalósítás és megállapítások:

Adatbázis megtervezése és létrehozása, majd ezt követően felületek elkészítése HTML-ben minimális CSS beállításokkal. Ezt akkor érdemes, ha az ember látni szeretné előre mi és hol legyen az oldalon, ha szükséges ilyenkor még könnyebben tud rajta változtatni. Az első adatfeltöltéses felületek elkészítése, ami a beléptetés, majd ezt követően a regisztráció és az ehhez szükséges feldolgozó programok. Az adatfeltöltéses felületeken belül érdemes lehet inkább a regisztrációval kezdeni, hogy az embernek teszteléskor ne azzal kelljen kezdeni, hogy hozzon létre adatbázison belül magának egy teszteléshez szükséges profilt. A regisztrációt, és a beléptetést követően a regisztráció módosítására használt, és annak feldolgozó programja került elkészítésre. Erre azért volt szükség, mivel a regisztrációban nem várunk el sok adatot. Amikor egy weboldalban regisztrációt várunk felhasználóinktól, érdemes azt eleinte rövidre venni, mert az emberek nagy része nem kedveli a kötelező regisztrációt.

Célkitűzések, amik nem sikerültek teljességgel

- A weboldalon való regisztráció sikeres lett, ámbár a visszaigazoló e-mail-t nem igazán küldi el, ezt a jövőben javítani tervezem.
- Az oldalt igyekeztem teljesen alkalmazkodóvá (responsive) tenni. Sajnos ez maradéktalanul nem sikerült. Ennek az oka a bootstrap kártyák egyesével való felpakolása.
- Az admin felületen időhiány miatt nem tudtam megvalósítani az új hamburger hozzáadását.

12. Hivatkozásjegyzék:

- <https://www.w3schools.com/>
- <https://getbootstrap.com/>
- <https://nodejs.org/en>
- <https://expressjs.com/>
- <https://expressjs.com/en/resources/middleware/cors.html>
- <https://www.npmjs.com/package/nodemon>

13. Ábrajegyzék:

1. ábra: Adatbázis	5
2. ábra: users tábla	5
3. ábra: termékek tábla és adatai	6
4. ábra: Kosarak tábla	7
5. ábra: Rendelés adatbázisba	7
6. ábra: Beléptetés algoritmus	9
7. ábra: Kijelentkezés kódja	9
8. ábra: Kosár és végösszeg	10
9. ábra: Végösszeg számítás	10
10. ábra: Üres mező alertje	11
11. ábra: A jelszó nem egyezik alert	11
12. ábra: Sikeres regisztráció alert	12
13. ábra: Folgalt felhasználónév	12
14. ábra: Regisztráció kódja	13
15. ábra: Rang kiiratás	13
16. ábra: Navigálás két oldal között	14
17. ábra: Navigációs fülek	14
18. ábra: Kosárba helyezés	15
19. ábra: Kosárba helyezés kódja	16
20. ábra: Kosár kiürítés kódja	16
21. ábra: LocalStorage frissítése és törlése	17
22. ábra: LocalStorage létrehozása és termékek hozzá adása	18
23. ábra: Sikeres vásárlás	19
24. ábra: Rendelés mentése adatbázisba	19
25. ábra: Üres kosár	19
26. ábra: A vásárlás és az eltávolítás kódja	20
27. ábra: Főoldal	25
28. ábra: Rólunk fül	25
29. ábra: Burgereink fül	26

30. ábra: Egy hamburger leírása.....	26
31. ábra: Kosár fül.....	27
32. ábra: Kosár és rendelés leadás.....	27
33. ábra: Termék a kosárban.....	28
34. ábra: Ki számolt végösszeg.....	28
35. ábra: Admin felület főoldala	30
36. ábra: Új hamburger felvitele	31

Eredetiségnyilatkozat

Alulírott **Miskolczi Dávid** nyilatkozom, hogy záró dolgozatom a saját szellemi termékem.

Debrecen, 2023. április. 12.

Miskolczi Dávid

.

aláírás