

**BÉKÉSCSABAI SZC
TREFORT ÁGOSTON TECHNIKUM,
SZAKKÉPZŐ ISKOLA ÉS KOLLÉGIUM**

***SZAKKÉPESÍTÉS NEVE: SZOFTVERFEJLESZTŐ
OKJ SZÁMA:5421305***

**ZÁRÓ DOLGOZAT
ÍZLETES ÉTTEREM**

**SZABÓ RICHÁRD
TÉMAVEZETŐ**

**KREPSZ BALÁZS ANDRÁS
13/A**

Békéscsaba,2024

Irodalomjegyzék

Bevezető	3
A program általános meghatározása :.....	4
Témaválasztás indoklása:	6
Az alkalmazott fejlesztői eszközök:	7
3.Rendszerkövetelmények.....	8
4.Adatmodell leírása.....	9
Program Használati útmutató	13
A program telepítése.....	24
Továbbfejlesztése	28
Irodalomjegyzék:	29
Plágium-nyilatkozat.....	30

Bevezető

Üdvözljük éttermünk online rendelési platformján! Rugalmas és gyors megoldást kínálunk az ételrendelésre, legyen szó munkahelyről vagy otthonról. Felhasználóbarát felületünkön könnyedén válogathat ételkínálatunkból és egyszerűen leadhatja rendelését. Az online rendszerünk lehetővé teszi számukra, hogy hozzáférjenek saját adataikhoz és korábbi rendeléseikhez. Tapasztalataikat megoszthatják, segítve másokat a választásban és fejlesztve szolgáltatásainkat.

Regisztrációra buzdítjuk ügyfeleinket, hiszen ez számos előnnyel jár. Gyorsabbá és egyszerűbbé teszi az ételrendelést, előre rögzítve személyes adataikat. Személyre szabott ajánlatokban és promóciókban részesülhetnek, valamint könnyen nyomon követhetik rendeléseiket és beállíthatják kedvenc ételeiket.

Az adatvédelem kiemelt fontosságú számunkra. Biztosítjuk a biztonságos bejelentkezést és kijelentkezést, valamint lehetőséget biztosítunk a rendelés nyomon követésére. Ügyfeleink mindig tájékozottak lehetnek a rendelésük állapotáról, és könnyen megtekinthetik saját adataikat és korábbi rendeléseiket.

Az Étterem Online Rendelési Rendszere könnyedén és gyorsan használható, hogy egyszerűsítse és felgyorsítsa az ételrendelési folyamatot. Ha Ön is könnyedén és gyorsan szeretne ételt rendelni, és megosztani véleményét az étteremről, akkor online rendszerünk ideális választás lehet Önnek!

Reméljük, hogy élvezni fogja az ételrendelést az új online rendszerünkön keresztül, és várjuk visszajelzéseit és rendeléseit! Köszönjük, hogy választotta éttermünket!

Célunk, hogy az online rendelési platformunkkal az ügyfeleknek mélyebb betekintést nyújtsunk az ételrendelés folyamatába, lehetővé téve számukra, hogy teljeskörűen megismerjék az éttermünk által kínált lehetőségeket és szolgáltatásokat. Ezen keresztül az ügyfelek jobban megérthetik az ételkínálatot, az elkészítési folyamatot, valamint az árakat és az műveleteket.

Az online rendelési platformunk segítségével az ügyfelek könnyen böngészhetik az étlapot

A program általános meghatározása :

Az éttermünk online rendelési rendszere lehetővé teszi, hogy kényelmesen és gyorsan rendeljen ételt a saját otthonából vagy bárhol. A felhasználóbarát felületen könnyen böngészhet az ételkínálatunkban és egyszerűen leadhatja rendelését. Legyen szó egy gyors ebédről munkahelyén vagy egy romantikus vacsoráról otthonában, az online rendszerünk segítségével egyszerűen és hatékonyan rendelhet.

Rendelési előzmény:

Az online rendszerünkben lehetősége van megtekinteni saját adatait és rendelési előzményeit is. A felhasználóknak biztosítjuk az eszközöket ahhoz, hogy egyszerűen hozzáférhessenek személyes információikhoz és korábbi rendeléseikhez.

Vélemény:

Az oldalon lehetőség van a felhasználóknak, hogy saját véleményüket és tapasztalataikat megosszák, ezáltal segítve másokat a választásban és nekünk abban, hogy továbbfejlesszük szolgáltatásainkat. Ügyfeink véleménye fontos számunkra! Az online rendszerünk lehetővé teszi, hogy értékelje és véleményezze az általunk kínált ételeket és szolgáltatásokat. Ezzel segíthet másoknak a választásban és nekünk abban, hogy folyamatosan javítsuk szolgáltatásainkat. Legyen szó az étel minőségéről, az ételkiszállítás sebességéről vagy az ügyfélszolgálatról, minden visszajelzés számít, és hozzájárul ahhoz, hogy még jobban szolgálhassuk Önt.

Regisztráció:

A felhasználó az oldal megkéri, hogy regisztráljon és hozzon létre saját felhasználói fiókot az online rendszerünkben, hogy kényelmesebb és hatékonyabb legyen az ételrendelés! A regisztráció során számos előnyt élvezhet. Először is, egyszerűbben és gyorsabban tud majd rendelni azáltal, hogy az Ön személyes adatait már előre rögzítjük. Ezenkívül a regisztrált felhasználók részesülnek személyre szabott ajánlatokban és promóciókban, amelyek csak rájuk vonatkoznak, így még kedvezőbb áron juthatnak hozzá az ételkülönlegességeinkhez. Emellett a felhasználói fiók segítségével könnyen nyomon követheti rendeléseit, valamint beállíthatja kedvenc ételeit és visszatekintheti korábbi rendeléseit.

Bejelentkezés:

Az online rendszerünk kiemelkedően fontosnak tartja ügyfeink biztonságát és adatvédelmét. Ennek érdekében biztosítjuk a biztonságos bejelentkezés és kijelentkezés lehetőségét. Amikor bejelentkezik a fiókjába, garantáljuk, hogy csak Ön fér hozzá személyes adataihoz és előzményeihez. Ez lehetővé teszi számára, hogy könnyen és kényelmesen nyomon követhesse rendeléseit.

Kijelentkezés:

A kijelentkezés funkció ugyancsak kiemelkedő fontosságú az adatvédelem szempontjából. Amikor befejezte a böngészést és már nem használja az oldalt, a kijelentkezés lehetőséget biztosítja, hogy a rendszer teljesen kilépjen a felhasználói fiókjából. Ez az intézkedés megakadályozza, hogy mások hozzáférjenek adataihoz, és maximalizálja a biztonságot. Így Ön mindig biztonságban érezheti magát.

Rendelés:

Miután Ön leadott egy rendelést az online rendszerünkben, biztosítjuk számára a lehetőséget, hogy egyszerűen és gyorsan nyomon kövesse annak állapotát. Ennek köszönhetően mindig tájékozott lehet a rendelés aktuális státuszáról. Így Ön könnyedén és gyorsan követheti rendelésének útját az étel megrendelésétől egészen a kézbesítésig, és biztos lehet abban, hogy mindig naprakész információk állnak rendelkezésére.

Saját adatok:

Bármikor bejelentkezve a felhasználói fiókjába, könnyen és gyorsan megtekintheti saját adatait. Ide tartoznak a regisztráció során megadott név, cím, telefonszám és egyéb kapcsolatfelvételi információk. Ezek az adatok segítenek abban, hogy gyorsan és pontosan kezeljük rendeléseit, valamint, hogy biztosítsuk az Ön elégedettségét. A felhasználók számára lehetőség van nyomon követni korábbi rendeléseiket is. Az "Rendelési Előzmények" szekcióban láthatja az összes korábbi rendelését azokat is, amik már teljesítve lettek.

Témaválasztás indoklása:

Ez egy olyan téma, amely közel áll hozzám, mert érdekesnek találom az éttermek működését és az online rendelési rendszerük mögött rejlő folyamatokat. A mai világban sokan nem látnak bele az éttermek online jelenlétének és rendelési lehetőségeinek működésébe, és úgy éreztem, hogy fontos lenne ezt a témát jobban megérteni. Az étterem online rendelési rendszere fontos és aktuális kérdéseket érint mind az étterem üzemeltetői, mind pedig az ügyfelek szempontjából. Az éttermek online rendelési rendszerei ma már alapvető fontosságúak a digitális világban. Személyesen is izgalmasnak találom ezt a témát, mert érdekel, hogy az éttermek hogyan alkalmazzák az innovatív megoldásokat a digitális térben. Sok étterem kínál már online rendelési lehetőséget, ami gyorsaságot és kényelmet nyújt az ügyfeleknek. Mégis, sokan nem értik pontosan ezeket a folyamatokat és rendszereket. Fontos lenne, hogy jobban megértsük és feltárjuk ezt a területet.

Az online rendelési lehetőségek nemcsak az ügyfeleknek kedveznek, hanem az éttermeknek is nagy segítségül szolgálnak a versenyben maradásban. Az online jelenlét és a rendelési rendszerek használata lehetővé teszi az éttermek számára, hogy elérjék és szolgálják ki az ügyfeleiket, valamint hatékonyabban kezeljék a rendeléseket. Az online rendelési folyamatoknak és technológiáknak sok előnye van, például a hatékonyabb rendeléskezelés, az automatizált fizetési folyamatok és a személyre szabott szolgáltatások. Az online rendelési lehetőségek a vásárlók számára is kényelmet és rugalmasságot jelentenek. A gyors és egyszerű rendelési folyamatok segítenek abban, hogy az ügyfelek könnyedén juthassanak hozzá az ételükhöz, bárhol is legyenek éppen. Ez a téma érdekes mind az éttermek, mind pedig az ételrendelést végző vevők számára.

Az éttermek online rendelési rendszereinek jobb megértése lehetővé teszi, hogy mind az üzleti, mind pedig a fogyasztói oldalon még magasabb szintű szolgáltatásokat kínáljunk. Ezért fontos, hogy folyamatosan foglalkozzunk ezzel a témával, és keressük azokat az új lehetőségeket és fejlesztéseket, amelyek elősegíthetik az éttermek digitális jelenlétének és hatékonyságának növelését.

Az alkalmazott fejlesztői eszközök:

Frontend:

A frontend fejlesztése során alkalmazott fő technológia a JavaScript volt, mivel ez dinamikus és sokoldalú nyelv, tökéletes a modern webalkalmazásokhoz. A Visual Studio Code fejlesztői környezetet választottam a frontend kód szerkesztéséhez és fejlesztéséhez, mivel ez könnyen kezelhető és hatékony eszköz, amely támogatja a JavaScript fejlesztést. Tapasztalatom van a Visual Studio Code használatában, és korábbi projektek során sokat használtam ezt a környezetet, így magabiztosabban tudtam vele dolgozni. A weboldal tervezéséhez HTML-t, CSS-t és JavaScriptet alkalmaztam. Az HTML az alapvető szerkezetet adta a weboldalnak, a CSS segített a stílusok és elrendezések megvalósításában, míg a JavaScript a dinamikus funkciók és interaktivitás hozzáadásában játszott fontos szerepet. Emellett a Tailwind CSS-t is használtam, amely egy moduláris CSS keretrendszer. A Tailwind segítségével könnyedén kialakíthatóak az egyedi stílusok és komponensek, ami gyors és hatékony frontend fejlesztést tesz lehetővé. Ezeket a technológiákat együtt alkalmazva sikerült létrehoznom egy modern és felhasználóbarát webes felületet, amely megfelel az elvárásoknak és könnyen használható a felhasználók számára.

• Backend:

A backend fejlesztése során az Express.js, a Prisma és a MySQL kombinációját választottam, hogy hatékony és megbízható webalkalmazást hozzak létre. Az Express.js-t választottam a webalkalmazás és API kialakításához, mivel ez egy könnyen tanulható és alkalmazható keretrendszer, amely lehetővé teszi a gyors prototípusok készítését és a fejlesztési idő minimalizálását. A Prisma az adatbázis-kezelésre szolgál, és segít egyszerűsíteni az adatbázis műveleteket, miközben objektumorientált kód írását teszi lehetővé, anélkül, hogy mélyreható ismeretekre lenne szükség az SQL nyelvvel kapcsolatban. Emellett a MySQL-t alkalmaztam az adatok tárolására és kezelésére, mivel ez egy népszerű és megbízható relációs adatbázis-kezelő rendszer, amely hatékonyan kezeli az adatokat. Ennek eredményeként ez a kombináció lehetővé teszi az emberi hibák minimalizálását és az optimális fejlesztési folyamatot.

3.Rendszerkövetelmények

Hardver

Követelmények:

- Processzor: Intel Core i3 vagy ekvivalens
- Memória: Legalább 4 GB RAM
- Tárhely: Legalább 100 MB szabad tárhely
- Internetkapcsolat: Szükséges

Ajánlott konfiguráció:

- Processzor: Intel Core i5 vagy annál erősebb
- Memória: Legalább 8 GB RAM vagy annál több
- Tárhely: Legalább 500 MB szabad tárhely
- Internetkapcsolat: Erősen ajánlott

Szoftver követelmények:

- A program kompatibilis a legfrissebb Windows 7 vagy annál modernebb operációs rendszerekkel.
- Node.js 16+ szerverkörnyezet kötelező Egyéb

szoftverkomponensek:

- Webböngésző: A legfrissebb verziója ajánlott a Chrome, Firefox, vagy Safari böngészőknek
- Adatbázis-szerver: MySQL

Adatbázisállományok: A szükséges adatbázis-állományokat a program mellékeli, és telepítés után automatikusan kerülnek beállításra

Fontos: Az internetkapcsolat szükséges az étterem weboldalának megfelelő működéséhez, mivel az online rendelési és foglalási folyamatokhoz szükséges adatokat online kell elérni.

4. Adatmodell leírása

```
model users {
  id          Int          @id @default(autoincrement())
  nev         String
  email       String       @unique
  telefonszam String       cim      String
  jelszo      String       rendelések
  rendelések[]
}
```

```
@ @map("felhasznalok")
```

3	feke kispista	fekefekispista@nemfizetek.hu	0630123124	nemakarokfizetni utca 12	\$argon2id\$=19\$m=65536,t=3,p=4\$ogpTpwfrdrbEWt1IPB9...
---	------------------	------------------------------	------------	-----------------------------	--

1. ábra

Az adatbázisomban megtalálható egy felhasználók tábla, amelyet Prisma segítségével hoztam létre. Ez az azonosító mező felelős minden felhasználó egyedi azonosítójának tárolásáért. Az `@id` dekorátorral és az `"@default(autoincrement())"` beállítással az adatbázis automatikusan növeli az azonosító értékét minden új étel létrehozásakor. A táblában a felhasználó neve került, amit bejelentkezésnél is meg kell adni a rendelő nevét, hogy az oldal tudja ki rendelte az ételt és ki írta le a véleményét az oldalban. Ezt egy string típusú adat típusként hozom létre. Valamint a felhasználókhoz `userToken` használtam. Ez a token lehetőséget ad az alkalmazásnak, hogy azonosítsa és azonosítson egy felhasználót például bejelentkezési műveletek során. Ennek az az előnye, hogy megkönnyíti a felhasználói munkamenetek kezelését, valamint az engedélyek és jogosultságok kezelését az alkalmazásban. Az Email mező `@unique` beállítása biztosítja, hogy minden felhasználó egyedi e-mail címmel regisztráljon, ezáltal megelőzve a duplikációt és biztosítva adataid integritását. Az oldal lekezelet, hogy `@` nélkül nem lehet email címet létrehozni. A telefonszám mező számokat fogad csak el, ezáltal garantálva, hogy csak érvényes telefonszámokat lehessen megadni. A lakcím kötelező megadása szintén jó megközelítés, mivel ez biztosítja, hogy minden rendeléshez rendelkezésre álljon a szállításhoz szükséges információ.

A jelszavak titkosítása fontos lépés a felhasználók biztonságának és adatvédelmének biztosítása érdekében. Biztonságos jelszó tárolás céljából az Argon2-t használtam, mivel ez a jelenlegi ajánlott és elismert jelszó hashelési algoritmus. Az Argon2 kifejezetten tervezett a jelszavak biztonságos tárolására és védelmére, és ellenálló a különféle támadásokkal szemben és a jelszó adatbázisok kiszivárgása. A főbb okok, amiért az Argon2-t használtam mert biztonságos és ellenálló a különböző támadásokkal szemben. Az algoritmus célja az, hogy megnehezítse a támadók számára a jelszavak feltörését, még akkor is, ha azoknak hosszú és erős jelszavak. Az Argon2 lehetővé teszi egyéb paraméterek beállítását, amelyekkel finomhangolható a hashelési munka terhelése és memóriahasználata. Ez lehetővé teszi az alkalmazás számára, hogy szabályozza a hashelés sebességét és a biztonság közötti egyensúlyt teremtsen.

```

model etelek {
    id Int @id @default(autoincrement())
    ar Int
    kategoria String
    leiras String
    nev String
    rendeles rendeles[]
    @@map("etelek")}

```

2. ábra

Az ételek táblában az **id** mező felelős minden étel egyedi azonosítójának tárolásáért. Az `@id` dekorátorral és az `@default(autoincrement())` beállítással az adatbázis automatikusan növeli az azonosító értékét minden új étel létrehozása esetén. Az `ar` tárolja az étel árát. Ez lehetővé teszi az étel árának követését az alkalmazásban. Az étel kategóriáját a `kategoria` mező tárolja el, például "Amerikai konyha", "Pizza", "Ázsiai" stb. Ez lehetővé teszi az ételek könnyű csoportosítását és rendezését a felhasználók számára. A `leiras` mező lehetővé teszi egy rövid leírás hozzáadását az ételhez, a rendelésről vagy maga az étteremről. Ez segíthet a felhasználóknak a választásban, valamint részletesebb információkat nyújthat az ételről. A `nev` mező tárolja az étel nevét. Ez az egyedi név azonosítja az ételt az alkalmazásban és lehetővé teszi a könnyű azonosítást és keresést. A `rendeles` ételek kapcsolatot képez az "etelek" és a "rendeles" táblák között, ami lehetővé teszi az egy ételhez tartozó rendelések nyomon követését. Ez az összekapcsolás segíti az alkalmazást abban, hogy tárolja és kezelje az ételrendeléseket az adott ételhez.

```

model rendeles {
    id Int @id
    @default(autoincrement()) ar Int
    datum DateTime @default(now())
    etelid Int
    mennyiseg Int @default(1)
    felhasznalo_id Int
    megjegyzes String
    statusz String @default("Felvéve")
    etelek etelek @relation(fields:
[etelid], references: [id])
    felhasznalok users @relation(fields:
[felhasznalo_id], references: [id])
    velemeny velemeny[]

    @@index([etelid], map: "rendelesek_etelid_fkey")
    @@index([felhasznalo_id], map: "rendelesek_felhasznalo_id_fkey")
    @@map("rendelesek")
}

```

3. ábra

Az azonosító mező felelős minden rendelés egyedi azonosítójának tárolásáért. Az @id dekorátorral és az "@default(autoincrement())" beállítással az adatbázis automatikusan növeli az azonosító értékét minden új rendelés létrehozásakor. Ár tárolja a rendelés teljes árát. Ez lehetővé teszi az adott rendelés költségének nyomon követését. A dátum mező árulja a rendelés dátumát és időpontját. Az "@default(now())" beállítás lehetővé teszi, hogy az alapértelmezett érték automatikusan a rendelés létrehozásának időpontja legyen. Ételid mező az étel azonosítóját tárolja, amelyhez a rendelés tartozik. Ez a mező kapcsolatot teremt az ételek tábla egy ételével. Mennyiség mező tárolja el az adott ételből rendelt mennyiséget. Az @default(1) beállítás azt jelenti, hogy alapértelmezés szerint minden rendelés egy darab ételt tartalmaz.

Felhasználó id tárolja a rendelést leadó felhasználó egyedi azonosítóját. Ez a mező kapcsolatot teremt a users táblával, ami a felhasználók adatait tartalmazza. A Megjegyzés lehetővé teszi a felhasználók számára, hogy megjegyzést adjanak hozzá a rendelésükhöz. Státusz mező a rendelés státuszát mutatja, például "Felvéve", "Feldolgozás alatt", "Kiszállítás alatt" stb. Ételek és felhasználók kapcsolata az ételek és users táblákkal való kapcsolatot jelölik, és az adott rendeléshez tartozó étel és felhasználó adatait biztosítják. Vélemény kapcsolata mező kapcsolatot teremt a vélemény táblával, ami a rendeléshez kapcsolódó vásárlói véleményeket tárolja.

```
model vélemény {
```

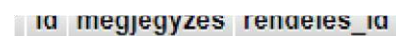
```
  id      Int    @id @default(autoincrement())
```

```
  megjegyzes String  rendeles_id Int  rendeles  rendeles @relation(fields: [rendeles_id],
references: [id])
```

```
  @@index([rendeles_id], map: "velemenyek_rendeles_id_fkey")
```

```
  @@map("velemenyek")
```

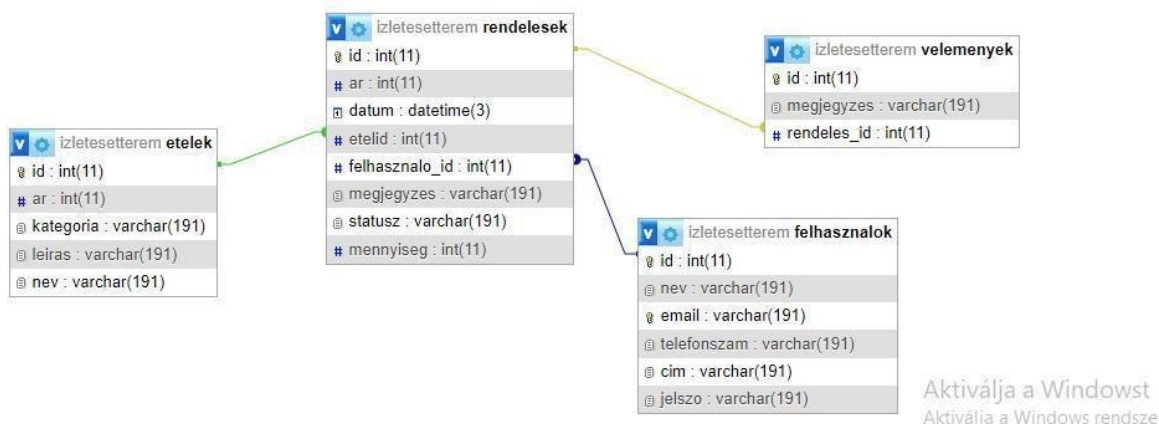
```
}
```



4. ábra

Azonosító mező itt is felelős minden vélemény egyedi azonosítójának tárolásáért. Az "@id" dekorátorral és az "@default(autoincrement())" beállítással az adatbázis automatikusan növeli az azonosító értékét minden új vélemény létrehozásakor. Megjegyzés mező tárolja a felhasználók által hozzáadott véleményeket vagy megjegyzéseket a rendelésekkel kapcsolatban. A rendeles id mező tárolja a kapcsolódó rendelés egyedi azonosítóját, amelyhez a vélemény tartozik. A Rendeles kapcsolat mező a rendeles táblával való kapcsolatot jelöli. A rendeles táblában tárolt adatokhoz való hozzáférést biztosítja a vélemények táblában. Az @@index dekorátor segítségével indexeket hozhatsz létre az adatbázisban, amelyek segíthetik az adatok hatékonyabb keresését és lekérdezését. Az @@index dekorátorral a megfelelő mezőkre mutató indexeket hoztunk létre az adatbázisban. A "velemenyek_rendeles_id_fkey" index a rendeles_id mezőre mutat, amely a velemeney tábla külső kulcsaként szolgál a rendeles táblához. Ez az index segíthet az adatok integritásának és konzisztenciájának fenntartásában, valamint a kapcsolódó adatok gyorsabb lekérdezésében.

Kapcsolatok:



5. ábra

Az adatmodellben szereplő kapcsolatok segítenek az adatok összekapcsolásában és strukturálásában az adatbázisban. Felhasználók és Rendelések kapcsolata a felhasználók modell és a rendelés modell között egy-több kapcsolat áll fenn, mivel egy felhasználóhoz több rendelés is tartozhat. A felhasználók modellben a rendelések tömbként vannak tárolva, ami lehetővé teszi a felhasználókhoz tartozó összes rendelés egyszerű lekérdezését. Az ételek és Rendelések kapcsolata az ételek modell és a rendelés modell között is egy-több kapcsolat van, mivel egy ételhez több rendelés is tartozhat. A rendelés modellben az ételeket reprezentáló mező egy másik ételek objektumra mutat, amely lehetővé teszi az étel adatainak hozzáférését minden rendelés esetén. Rendelések és Vélemények kapcsolat a rendelés modell és a vélemény modell között egy-több kapcsolat áll fenn, mivel egy rendeléshez több vélemény is tartozhat. A vélemény modellben a rendeléshez tartozó rendelés azonosítóra mutató mező segítségével lehetővé teszi a rendeléshez kapcsolódó vélemények összegyűjtését és megjelenítését. Ezek a kapcsolatok lehetővé teszik az adatok egyszerű kezelését és strukturálását az adatmodellben, és segítenek a különböző adatok közötti összefüggések fenntartásában.

Program Használati útmutató

1. Bevezetés:

Kedves Felhasználó!

Nagyon örülök, hogy itt vagy, és érdeklődsz a Program Használati Útmutatója iránt! Ahogy a technológia egyre inkább behálózza mindennapi életünket, az informatika területe is folyamatosan fejlődik és új lehetőségeket teremt mindennapi tevékenységeink könnyebbé és hatékonyabbá tételére. Az alkalmazások és szoftverek által nyújtott kényelem és funkcionalitás egyre nélkülözhetetlenebbé válik a mindennapi munkavégzés, tanulás és szórakozás során.

Ez az útmutató arra szolgál, hogy segítsen neked az első lépéseknél és felfedezéseknél ebben a programban. Lehet, hogy új vagy az ilyen típusú szoftverek használatában, vagy csak szeretnéd jobban kihasználni az általa kínált lehetőségeket - bármi is legyen a helyzet, itt vagyok, hogy segítssek neked az utadon.

Ez a dokumentáció egy teljes körű útmutatót kínál a program telepítéséről, beállításáról és használatáról, és minden olyan információt tartalmaz, amire szükség lehet ahhoz, hogy teljes mértékben kihasználd a program által nyújtott előnyöket. Kezdve a legapróbb részletekkel a rendszerkövetelményekről, egészen a program használatának legbonyolultabb funkcióinak elemzéséig, mindenre kiterjedünk.

Az útmutató lépésről lépésre vezet be a program használatába, és részletesen ismerteti a bejelentkezés, a funkciók használata és a lehetséges problémák megoldása folyamatát. Emellett gyakori kérdésekkel és problémákkal is foglalkozunk, hogy segítsünk neked, ha bármilyen nehézségbe ütköznél a program használata során.

Bármilyen további kérdésed vagy észrevételed van, bátran oszd meg velem! A véleményed nagyon fontos számomra, és mindent megteszek annak érdekében, hogy az útmutató minél informatívabb és segítőkészebb legyen.

Köszönöm, hogy elkötelezted magad a program használata mellett, és bízom benne, hogy ez az útmutató segíteni fog abban, hogy sikeresen és örömmel használd a programot!

2. Regisztráció:

Az űrlap az oldal frissítését a **preventDefault()** metódus segítségével kadályozza meg.

A kód a felhasználó által megadott adatokat beszerzi az űrlap mezőiből, amelyek tartalmazzák a felhasználó által megadott információkat.

A regisztrációs űrlap a következő mezőket tartalmazza:

- Felhasználónév (**#username**)
- E-mail cím (**#email**)
- Telefonszám (**#phonenumber**)
- Lakcím (**#cim**)
- Jelszó (**#password**)

A kód egy objektumot hoz létre, amelyet 'regForm' néven azonosít, és amely tartalmazza a felhasználó által a regisztráció során megadott adatokat, mint például a felhasználónév, e-mail cím, telefonszám, lakcím és jelszó. Ez az objektum összegyűjti és ennek a vázlatát építi fel ezeket az adatokat, hogy könnyen kezelhető formában lehessenek továbbítva a későbbi feldolgozáshoz és adatbázisban történő tároláshoz.

```
e.preventDefault();
const nev = document.querySelector('#username').value;
const email = document.querySelector('#email').value;
const telefonszam = document.querySelector('#phonenumber').value;
const cim = document.querySelector('#cim').value;
const jelszo = document.querySelector('#password').value;

const regForm = {
  nev: nev,
  email: email,
  telefonszam: telefonszam,
  cim: cim,
  jelszo: jelszo
}
```

6. ábra

Majd egy POST kérést küld az űrlap által összegyűjtött adatokkal az '/user/register' végpontra, ahol a szerver fogadja és feldolgozza ezeket az adatokat. Ez a folyamat lehetővé teszi, hogy a felhasználók regisztráljanak az alkalmazásba vagy weboldalra, és a szerver oldalon történő feldolgozás során az adatokat tárolják az adatbázisban vagy más kezelőrendszerben, hogy később hozzáférhetőek legyenek. Ez az űrlap elküldése és adatok feldolgozása az első lépés a felhasználói regisztráció folyamatában, amely lehetővé teszi a felhasználók számára a hozzáférést a platform kínálta szolgáltatásokhoz vagy tartalmakhoz.

Amikor regisztráltam az online platformon, fontos volt számomra, hogy azonnal értesüljek arról, hogy a regisztráció sikeres volt-e vagy sem. Ezért döntöttem úgy, hogy megvalósítom a regisztráció utáni visszajelzést toast üzenet formájában. Úgy éreztem, hogy ez egy hatékony és közvetlen módja annak, hogy a felhasználót azonnal tájékoztassam a folyamat eredményéről, anélkül, hogy túl sokat kellene belépnie a részletekbe vagy navigálnia az oldalon.

A sikeres regisztráció esetén egy pozitív üzenet jelenik meg, amely gratulál nekem a sikerhez, és tájékoztat arról, hogy most már hozzáférhetek az összes funkcióhoz. Úgy gondoltam, hogy ez segít abban, hogy pozitív érzéseket keltsen az új felhasználókban, és bátorítást ad a további használatra.

Ha a regisztráció sikertelen volt, egy negatív üzenet jelenik meg, amely tájékoztat arról, hogy mi volt a probléma, és hogyan lehet azt orvosolni. Úgy éreztem, hogy ez segít abban, hogy a felhasználók ne érezzék magukat elveszettnek vagy frusztráltak a regisztráció során, és segít nekik a probléma megoldásában.

Az üzenetek megjelenítése toast formában rövid és tömör, ami megkönnyíti azok gyors felfogását és nem terheli feleslegesen az oldal felületét. Összességében úgy érzem, hogy a toast üzenetek hatékonyan hozzájárulnak a felhasználói élményem javításához és az oldal intuitívabbá tételéhez.

Amennyiben a regisztráció sikeres volt, a felhasználónak hozzáférési tokenre lehet szüksége a későbbi belépésekhez vagy más biztonsági ellenőrzésekhez. A hozzáférési token egy egyedi azonosító, amely lehetővé teszi a felhasználó azonosítását és hitelesítését a szerveren, anélkül, hogy ismételten be kellene jelentkeznie. A token általában egy hosszú karakterlánc, amelyet a szerver generál, és a felhasználóhoz rendel.

```
let backgroundColorForToast = "red";
console.log(data);
if(data.message.includes("Sikeres")){
  backgroundColorForToast = "green";
  localStorage.setItem("userToken", JSON.stringify(data.access_token));
  Toastify({
    text: data.message,
    duration: 3000,
    newWindow: true,
    close: true,
    gravity: "top", // 'top' or 'bottom'
    position: "center", // 'left', 'center' or 'right'
    stopOnFocus: true, // Prevents dismissing of toast on hover
    style: {
      background: backgroundColorForToast,
    },
    onClick: function(){} // Callback after click
  }).showToast();
  setInterval(() => {
    location.href = "../weboldalam.html"
  }, 5000);
}
```

7. ábra

A hozzáférési token tárolása a böngésző localStorage-jában lehetővé teszi annak megőrzését az aktuális munkamenet (session) során. Ez azt jelenti, hogy ha a felhasználó frissíti az oldalt vagy bezárja a böngészőt, majd később visszatér az oldalra, a token továbbra is elérhető marad, és a felhasználó bejelentkezettként kezeli.

A regisztráció sikeres lebonyolítása után a

felhasználót azonnal átirányítom a weboldalam.html oldalra. Ezáltal biztosítom, hogy az újonnan regisztrált felhasználók azonnal hozzáférjenek az oldal további tartalmához és funkcióihoz.

3.Bejelentkezés:

Amikor a jelentkezés gombra kattintasz, megakadályozom az alapértelmezett űrlapküldési viselkedést. Összegyűjtöm az input mezők értékeit, beleértve a felhasználónevet, e-mail címet, telefonszámot, címet és jelszót. Ezután létrehozok egy JSON objektumot, amit **loginForm**-nak nevezek el, és ebbe a JSON objektumba belempekölöm ezeket az adatokat. Ezt követően pedig egy POST kérést küldök egy helyi kiszolgálónak a **http://localhost:8000/user/login** címre, és ezzel a JSON adattal.

A szerver válaszáól függően egy felugró értesítést jelenítek meg, ami jelzi, hogy sikerült-e a jelentkezés vagy sem. Ha a jelentkezés sikeres (**data.message** tartalmazza a "Sikeres" szöveget), akkor a felugró értesítés háttérének színét zöldre állítom, elmentem a felhasználói tokent a helyi tárolóba, majd némi késleltetés után átirányítalak egy másik oldalra (**../weboldal.html**). Ha a jelentkezés nem sikerül, akkor egy hibaüzenet jelenik meg piros háttérrel.

8. ábra

A loginButton arra szolgál, hogy kiválassza azt a HTML elemet az oldalon, amelynek azonosítója a "login". Ez az elem valószínűleg a jelentkezés gombot jelenti, vagyis az a gomb, amelyre kattintva a felhasználó be szeretne jelentkezni a rendszerbe. Tehát, amikor a felhasználó rákattint erre a gombra, a hozzárendelt eseménykezelő függvény fut le, ami kezeli a további lépéseket a jelentkezés folyamatában.

Amikor egy eseményfigyelőt hozzáadunk a jelentkezés gombhoz, az eseményfigyelő funkció arra szolgál, hogy figyelje a 'click' eseményt, ami azt jelenti, hogy érzékeli, amikor rákattintok a gombra. Ezen esemény bekövetkeztével pedig végrehajtódik a addEventListener függvény által átadott függvény, amelyet én adtam meg. Ez a függvény tartalmazza azokat a műveleteket és

utasításokat, amelyeket végrehajtok, amikor a gombra

kattintok. Így amikor a gombra kattintok, a addEventListener által hozzárendelt függvény fut le, és ez a függvény tartalmazza az általam meghatározott műveleteket és folyamatokat, amelyeket szeretnék végrehajtani a gombra kattintás után. Ezen keresztül indul el a jelentkezési folyamat, és az általam előírt műveletek és utasítások végrehajtása.

Az `e.preventDefault()` hívás egy olyan funkció, ami megakadályozza az alapértelmezett viselkedést, amely az eseményhez kapcsolódik. Például, amikor a bejelentkezés gombra kattintasz, az alapértelmezett működés az lenne, hogy az űrlap elküldik a szervernek, és az oldal újra töltődik a válasz alapján.

Azonban, ha szeretnéd, hogy a bejelentkezés folyamata a JavaScript kóddal vezérelve történjen, akkor használhatod az `e.preventDefault()` függvényt. Ez azt jelenti, hogy miután a gombra kattintasz, az esemény továbbra is bekövetkezik, de az alapértelmezett űrlap elküldési viselkedését megakadályozzuk. Ehelyett a JavaScript kód, amit én írtam, fog lefutni, és így dönthetsz arról, hogyan szeretnéd folytatni a bejelentkezés folyamatát. Ez lehetőséget ad arra, hogy aszinkron módon kommunikálj a szerverrel, adatokat küldj vagy fogadj, és reagálj a válaszokra anélkül, hogy az oldal újra töltődné.

A JavaScript kód az űrlap különböző mezőiből (például felhasználónév, e-mail, telefonszám, cím, jelszó) összegyűjti a felhasználó által megadott értékeket. Ehhez a mezők azonosítóját használja, mint például `#username`, `#email`, `#phonenumber`, `#cim` és `#password`. Az összegyűjtött értékeket egy JavaScript objektumba tároljuk. Ez az objektum neve `loginForm`, és a gyűjtött adatokat kulcs-érték párokban tartalmazza.

Amikor a felhasználó be szeretne jelentkezni az alkalmazásba, először is a kliens oldalon egy POST kérést küldök a szervernek. Ehhez a `fetch` API-t használom, és a kérés tartalmazza a felhasználó által megadott adatokat, amelyeket a `loginForm` objektumban tároltam JSON formátumban. Ezáltal biztosítom, hogy a szerver megfelelően értelmezhesse és feldolgozhassa a bejelentkezési adatokat.

Miután a szerver fogadta és feldolgozta a kérést, visszaküld egy választ a kliensnek. Ezt a választ én JavaScript kódban a kliens oldalon JSON formátumban értelmezem, hogy az általam megjelenített felhasználói felületen hasznos információkkal tudjak dolgozni.

A szerver választól függően egy felugró értesítést jelenítek meg a felhasználónak a `Toasty` könyvtár segítségével. Ha a bejelentkezés sikeres volt, a felugró értesítés háttérének színét zöldre állítom, és egy pozitív üzenetet jelenítek meg, amely megerősíti a sikeres bejelentkezést. Ellenkező esetben, ha a bejelentkezés nem sikerült, egy hibaüzenet jelenik meg piros háttérrel, amely tájékoztatja a felhasználót a probléma okáról.

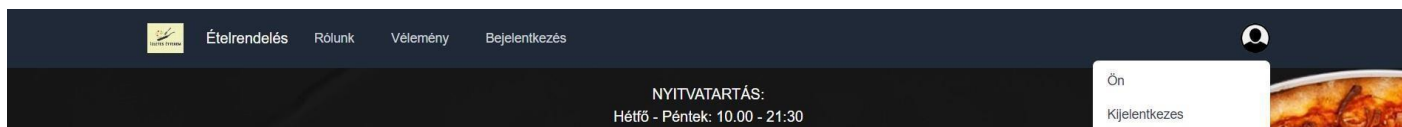
Végül, ha a bejelentkezés sikeres volt, elmentem a felhasználói tokent a helyi tárolóba a további használat céljából. Ezután egy kis késleltetést követően a felhasználót átirányítom egy másik oldalra vagy tevékenységre, például a bejelentkezett felhasználói felületre, hogy további tevékenységeket végezhesen az alkalmazásban.

4. Főképernyő áttekintése:

A weboldalam felső részén található navigációs sáv nagyon fontos szerepet tölt be az alkalmazásban történő navigációban. Ennek az eszköznek a segítségével könnyen és gyorsan eljuthatok az alkalmazás fontos részeihez és funkcióihoz. A navigációs sáv tartalmaz egy logót. Ez segít azonosítani az oldalt, és lehetővé teszi számomra, hogy könnyen visszatérjek a főoldalra. Az oldal felső részén elhelyezett menüpontok segítségével könnyen navigálhatok az alkalmazás különböző részei között. Ezek a menüpontok általában fontos funkciókat vagy oldalakat jelölnek, mint például a főoldal, a rólunk oldal, a vélemény oldal vagy a bejelentkezés oldal. Amikor kisebb kijelzővel rendelkező eszközt használok, például okostelefont vagy táblagépet, a navigációs sáv egy speciális mobil menü gombot tartalmaz. Ez a gomb egy kis ikon, amelyet kattintással vagy érintéssel lehet aktiválni. Amikor aktiválom, egy kis menü jelenik meg, ahol további navigációs lehetőségek találhatók. Ez segít abban, hogy könnyen navigálhassak még kisebb kijelzőkön is, mivel a menüpontok kevesebb helyet foglalnak el, és könnyebben elérhetők.

Az "Ételrendelés" gomb vagy link az alkalmazás fontos része, amely lehetővé teszi a felhasználók számára, hogy gyorsan és egyszerűen hozzáférjenek az ételrendelési lehetőségekhez. Ennek a vezérlőelemnek a segítségével könnyen és kényelmesen böngészhetik az étlapot, válogathatnak az ételek között, és végül leadhatják a rendelésüket.

Amikor rákattintok az "Ételrendelés" gombra vagy linkre, azonnal elérhetővé válnak az ételrendelési lehetőségek, a menük, ételek és italok. Ez a funkció kiemelkedő fontossággal bír az alkalmazás szempontjából, mivel ez teszi lehetővé a felhasználók számára az ételrendelést, amely valószínűleg az alkalmazás egyik legkeresettebb szolgáltatása.



Az "Ételrendelés" gomb vagy link nagy hangsúlyt kap a felületen, hogy könnyen megtalálható

legyen a felhasználók számára. Általában élénk színekkel vagy jól látható formatervezéssel rendelkezik, hogy felkeltse a figyelmet és ösztönözze a felhasználókat a kattintásra. Ezáltal biztosítva a felhasználók számára a gyors és zökkenőmentes ételrendelési élményt az alkalmazásban.

A jobb felső sarokban elhelyezkedő felhasználói menü egy fontos és gyakran használt része az alkalmazásnak. Ez a menü általában egy kis profilképet vagy ikont tartalmaz, amely lehetővé teszi a felhasználó számára, hogy azonosítsa magát vagy az aktuálisan bejelentkezett felhasználót. Amikor a felhasználó rákattint erre a profilképre, egy lenyíló menü jelenik meg. Ebben a lenyíló menüben számos funkció és lehetőség található. A felhasználó itt megtekintheti saját profiladatait, beleértve a nevet, e-mail címet, profilképet stb. Ezáltal lehetősége van

frissíteni vagy módosítani az adatait az alkalmazásban. Ezenkívül a felhasználói menüben általában megtalálható a Kijelentkezés lehetőség is. Ez a gomb lehetővé teszi a felhasználók számára, hogy biztonságosan és egyszerűen kilépjenek a rendszerből, amikor már nem szeretnék tovább használni az alkalmazást.

A modal ablak valójában az ételek rendelését szolgálja, és a felhasználók számára lehetővé teszi, hogy kiválasszák és elküldjék a rendelésüket. Az ablakban található egy beviteli mező, amely lehetővé teszi a felhasználók számára, hogy megadják a nevüket. Ez a névmező fontos információ a rendelés feldolgozása során, mivel lehetővé teszi az étterem számára, hogy azonosítsa, hogy ki küldte be a rendelést.

Az érdekesség az, hogy a névmező letiltott (disabled) állapotban van, tehát alapvetően nem szerkeszthető. Ez azt jelenti, hogy a felhasználók nem tudják megváltoztatni a nevüket az ablakban. Ezen a ponton a rendelő neve jelenik meg a névmezőben, mivel a felhasználó már be van jelentkezve. Ez a funkció biztosítja, hogy csak bejelentkezett felhasználók tudjanak rendelni az alkalmazáson keresztül.

Így tehát az ételek rendelése kizárólag a bejelentkezett felhasználók számára lehetséges, és a nevüket az automatikusan kitöltött és letiltott névmezőben adják meg. Ez egyszerűsíti és gyorsítja a rendelési folyamatot, miközben biztosítja a rendszer biztonságos működését és az egyértelmű felhasználói azonosítást.

Ezenkívül a modal ablakban megjelennek az elérhető ételek nevei, árai és a rendelhető mennyiségek. A felhasználók kiválaszthatják az ételeket a listából, és beállíthatják, hogy mennyi darabot szeretnének belőlük rendelni. Ehhez általában checkbox-okat használnak, amelyekkel a felhasználók kijelölhetik az adott ételt a rendeléshez, valamint beviteli mezőket, amelyekbe beírhatják a kívánt mennyiséget.

Válassz ételt:		
Rendelő neve: undefined	Ételek:	
<input type="checkbox"/> Lasagne Bolognese	Ár:2500-1	+
<input type="checkbox"/> Spaghetti Carbonara	Ár:2200-1	+
<input type="checkbox"/> Penne all Arrabbiata	Ár:1900-1	+
<input type="checkbox"/> Tortellini Panna e Prosciutto	Ár:2400-1	+
<input type="checkbox"/> Fettuccine Alfredo	Ár:1890-1	+
<input type="checkbox"/> Margherita Pizza	Ár:2800-1	+
<input type="checkbox"/> Pepperoni Pizza	Ár:2900-1	+
<input type="checkbox"/> BBQ Chicken Pizza	Ár:3200-1	+
<input type="checkbox"/> Capricciosa Pizza	Ár:3100-1	+
<input type="checkbox"/> Meat Lovers Pizza	Ár:3500-1	+
<input type="checkbox"/> Veggie Supreme Pizza	Ár:3600-1	+
<input type="checkbox"/> Vegan Margherita Pizza	Ár:3200-1	+
<input type="checkbox"/> Pad Thai tofuval és rizstésztával	Ár:1900-1	+
<input type="checkbox"/> Vegán zöldséges curry kókusztejjel	Ár:1500-1	+
<input type="checkbox"/> Steak sült burgonyával és zöldbabokkal	Ár:2650-1	+
<input type="checkbox"/> Sertésszelet borral párolva és párolt zöldségekkel	Ár:2490-1	+
<input type="checkbox"/> Lazac filé citromos mártással és sült édesburgonyával	Ár:2700-1	+
<input type="checkbox"/> Garnélarák és quinoa saláta	Ár:2380-1	+
<input type="checkbox"/> Gyros Tál	Ár:2090-1	+
<input type="checkbox"/> Sajtos Gyros Tál	Ár:2100-1	+

Ezáltal a felhasználók egyszerűen és kényelmesen választhatnak az elérhető ételek közül, és meghatározhatják, hogy mennyit szeretnének belőlük rendelni, mindezt egyetlen modal ablakban. Ez a felépítés segíti az egyszerű és intuitív rendelési folyamatot, és lehetővé teszi a felhasználók számára, hogy gyorsan és hatékonyan leadják rendelésüket.

Fontos információkat tartalmaz az étteremről. A fejléc részei közé tartozik egy háttérkép (bgimg és grayscale-min osztályok), amely lehetővé teszi az oldal vizuális vonzerejét, és segít kiemelni az étterem jellemzőit. A w3-display-container osztály egy keretet biztosít a fejléc tartalmának, és ezen belül helyezkedik el az információk és linkek.

Az időbeosztás (Hétfő - Péntek: 10.00 - 21:30, szombat - vasárnap: 10.00 - 22.00) fontos információ az ügyfelek számára, amely megjelenik a fejléc alsó részén. Ez lehetővé teszi számukra, hogy könnyen ellenőrizzék az étterem nyitvatartási idejét, és időben látogassanak el oda. Az "Ízletes Étterem" szöveg, valamint az "Étlap" link (a navigációs menü része) középen található. Ezek segítik az ügyfeleket abban, hogy gyorsan elérjék az étterem főoldalát, valamint az étlapot, ahol megnézhetik az elérhető ételek listáját. A "Étlap" link egy nagy gomb formájában jelenik meg, ami vonzóvá és könnyen kattinthatóvá teszi az étterem kínálatának megtekintését.

A főoldal következő részében egy fotógaléria található, ami kiemelt szerepet kap az oldal látványosságában. Ez a galéria a Főoldal elnevezésű szakaszban található, ami az oldal fő részét, azaz a főoldalt jelöli. A fotógaléria hat különálló képből áll, amelyek mindegyike saját dia formájában jelenik meg. Ezeket a diákat a mySlides osztály azonosítja, és mindegyik egyedi számozással és egy képpel rendelkezik.

A diák közötti



navigációhoz könnyen használható gombok szolgálnak: egy prev és egy next gomb. Ezek a gombok lehetővé teszik az előző vagy a következő dia megjelenítését. A dia váltásokat a

plusSlides függvény végzi, ami az előző vagy a következő diát jeleníti meg, a felhasználó kényelmét szolgálva.

Az alsó sorban pedig egy sor kis előnézeti kép található, amelyek gyorsan áttekintést nyújtanak a galéria tartalmáról. Minden egyes képet egy oszlop reprezentál, és mindegyikük kattintható, amit a demo osztály jelöl. A képek kattintásával a currentSlide függvény aktiválódik, ami megfelelő diára váltja a nagyobb méretű képet, így részletesen megtekinthetővé válnak az ételek vagy fogások. A képek címkéi az "alt" attribútumban adják meg az adott étel nevét, ami segíti a felhasználókat az azonosításukban és a választásban.

Ez után az étlap tekinthető meg a felhasználó számára. A div elemek, amelyek az étlapon szereplő ételeket reprezentálják, egy sor stílust alkalmaznak, hogy vonzó megjelenést biztosítsanak. A rounded-lg osztály lekerekített sarkokat ad az elemeknek, ami lágyabb, barátságosabb kinézetet kölcsönöz nekik. Abg-white osztály fehér háttért biztosít az ételek számára. Az ételek úgy jelennek meg, mintha kiemelkednének a lap felületéből. Végül, a "dark:bg-neutral-700" osztály sötét témában egy szürke hátteret biztosít az ételeknek, ami további kontrasztot és hangsúlyt ad nekik. Ezek az összetett stílusok együttesen a felhasználó figyelmét a kínált ételekre irányítják. Az átlátszó réteg az ételképek alatti részen helyezkedik el, és akkor jelenik meg, amikor a felhasználó az egérrel ráhúzza az adott ételre. Ez a megjelenési forma egyfajta felhasználói visszajelzést nyújt, hogy az adott elem interaktív, és lehetőséget biztosít a felhasználónak az étellel kapcsolatos további információk vagy akár megrendelés elérésére. A réteg átlátszó volta lehetővé teszi, hogy ne takarja el az ételképet, így könnyen észlelhető marad az étel, még akkor is, ha a réteg megjelenik. Ezáltal a felhasználói élmény fokozódik, mivel könnyebben navigálhatóvá válik az étlap. megjelenítés reszponzív, és könnyen áttekinthető a felhasználók számára.

A Főoldal alján kiemelt Elérhetőségeink cím hangsúlyozza és vastagítja a szöveget, így könnyebben megtalálható és érthetővé válik a szakasz célja. Az <address> elemet HTML5-ben szemantikus jelölésnek tekintik, ami segít a címek megfelelő megjelenítésében és az azonosításukban. Ez az elem nemcsak a címek formázására szolgál, hanem jelezheti a kapcsolatfelvétellel vagy elérhetőségekkel kapcsolatos információkat is. Az étterem címét, telefonszámát és e-mail címét általában az <address> elemen belül helyezik el, ami lehetővé teszi ezeknek az adatoknak a böngészők és keresőmotorok számára való megfelelő azonosítását. Ezek az adatok fontosak, mivel segítik az érdeklődőket abban, hogy felveszék a kapcsolatot az étteremmel vagy kérdéseiket és észrevételeiket közöljék. A footer rész az oldal alján található, gyakran nem közvetlenül hasznos információkat tartalmaz, de gyakran megtalálhatók benne jogi nyilatkozatok, mint például az © 2024 Minden jog fenntartva.

Elérhetőségeink:



123 Gasztronómia utca

Városközpont

Telefonszám: +36 1 234 5678

E-mail: info@izletesetterem.hu

Aktiválja a Windowst

Aktiválja a Windows rendszert a Gépházban.

© 2024 Minden jog fenntartva.

Ha a navbáron a "Rólunk" az oldalra kattintva egy másik oldalra viszi, amely arra a kérdésre válaszol, hogy miért érdemes minket kipróbálni, valójában azt szerettem volna elérni, hogy az emberek már az első pillanattól kezdve egyértelműen értsék és érezzék az étterem hangulatát és értékeit. Az "Üdvözöljük az Ízletes Étteremben!" üzenet számomra több mint egyszerű köszöntés: ez egy meghitt bevezető az étterem világába, amely azonnal felkelti az látogatók figyelmét és érzelmeit. A részletes történet bemutatása célja az volt, hogy árulkodjon az étterem alapjairól és értékeiről, ezáltal segítve a bizalom és elkötelezettség kialakítását az étterem és az látogatók között. Az "Miért érdemes minket meglátogatni?" rész további érdeklődést kelt az étterem iránt, részletesen kiemelve azokat a különleges élményeket, amelyekre számíthatnak az étterem látogatói. Ez az aprólékos leírás segíti az látogatókat abban, hogy jobban megértsék az étterem vonzerőjét és megtapasztalják azt a gasztronómiai kalandot,

Főoldal

Üdvözöljük az Ízletes Étteremben!



Szenvedélyesen hiszünk az étel és az élmény erejében. Éttermünk nem csupán egy hely, ahol étkezni lehet, hanem egy olyan közeg, ahol az ízek, az illatok és a kulináris élmények egyedülálló kombinációja vár minden vendéget.

Történetünk:

Az Ízletes Étterem története a minőségi alapanyagok, a kreatív szakácsok és a vendégszeretet köré fonódik. Minden ételünk egy kulináris mestermű, amelyet a legfrissebb és legjobb minőségű összetevőkből készítünk el. Büszkék vagyunk arra, hogy helyi termelőktől származó alapanyagokat használunk, támogatva ezzel a közösségünket és teremtve az igazi, autentikus ízeket.

Miért érdemes meglátogatni minket?

Éttermünk nem csupán az ízek helyszíne, hanem egy barátságos és meghitt közeg, ahol az emberek találkozhatnak és emlékezetes pillanatok élnek át. Szívélyesen várjuk Önt, hogy részese lehessen ennek a gasztronómiai utazásnak, és megoszthassuk Önnel a szenvedélyünket az étel iránt.

Köszönjük, hogy választotta az Ízletes Éttermet. Reméljük, hogy hamarosan személyesen üdvözölhetjük Önt éttermünkben!

amit az
étterem
ígér. A
footer
részben az
étterem logójának és
a jogi
nyilatkozatnak a
megjelenítése az
oldal hitelességének
és

megbízhatóságának fontos eleme, mivel ez segít megerősíteni az látogatók bizalmát és megnyugtatja őket arról, hogy az oldal hivatalos és jogilag védett. [linkre](#) mennek az oldal átvizsgálás egy másik oldalra, ami olyan kérdésre ad választ, hogy miért érdemes minket kipróbálni.

Az "Üdvözljük az Ízletes Étteremben!" üzenet nem csupán egy egyszerű köszöntés, hanem egy meghitt és invitáló bevezető az étterem világába. Ez a szöveg azonnal megragadja az látogató figyelmét és érzelmét, elősegítve a pozitív benyomás kialakítását már az oldal első pillanataiban.

A történet részletes bemutatása árulkodik az étterem alapjairól és értékeiről, ami fontos a bizalom és elkötelezettség kialakításában az étterem és az látogatók között. Ez a rész kifejezi az étterem elkötelezettségét a minőségi és friss alapanyagok felhasználása mellett, valamint bemutatja a konyhaművészetüket.

A footer rész, amely az étterem logóját és a jogi nyilatkozatot tartalmazza, az oldal hitelességének és megbízhatóságának fontos eleme. Ez a rész a bizalom erősítéséhez és az étterem jogainak védelméhez szükséges, biztosítva az látogatókat arról, hogy az oldal hivatalos és jogilag védett. A `<a>` elemmel az "Főoldal" szövegre kattintva az oldalam tetején könnyen visszatértem a főoldalra. Ez az egyszerű navigációs funkció lehetővé tette számomra, hogy gyorsan és hatékonyan böngésszek az oldal különböző részei között, miközben mindig könnyen visszatérhettem az alapvető tartalmakhoz vagy információkhoz az oldal kezdetén.

Ha a navbáron a "Vélemény" az oldalra kattintva egy másik oldalra viszi, amely arra szolgál, hogy a felhasználó elmondhassa a véleményét az adott rendelésről, ételről. Az általam megosztott vélemények és visszajelzések célja az, hogy egy sokkal mélyebb és személyesebb kapcsolatot teremtsen az étterem és az érdeklődő látogatók között. Ahelyett, hogy csak egy hagyományos bemutatkozó oldalt látnának, ahol az étterem saját magát dicsőíti, ezek a posztok valódi emberek tapasztalatait tükrözik, akik már megtapasztalták az étterem kulináris kínálatát és vendégfogadását.

Azt a szándékot akartam ezekkel a bejegyzésekkel, hogy az emberek ne csak az étterem nevében, hanem közvetlenül más vendégek által ismerjék meg az ételek ízvilágát, az étterem kiszolgálását és az ott átélt élményeket. Úgy gondolom, hogy ez a megközelítés sokkal hitelesebb és meggyőzőbb lehet, mint bármilyen hirdetés vagy bemutató, mert valós emberek valós tapasztalatait közvetíti. Emellett ezek a posztok lehetővé teszik az érdeklődők számára, hogy azonosuljanak más vendégekkel, akik hasonló élményeket éltek át az étteremben. Ez az emberi kötődés, ami segíthet az étterem iránti bizalom megerősítésében és az étterem látogatásának döntésében.

A weboldal legalján található egy rész, ahol véleményt lehet megosztani, azonban csak bejelentkezett felhasználók számára elérhető ez a lehetőség. Amikor valaki rákattint a "Posztolás" gombra, a `checkLogin()` függvény fut le, ami azt vizsgálja, hogy a felhasználó már be van-e jelentkezve a rendszerbe. Ha igen, akkor lehetőség nyílik az üzenet beküldésére, ha viszont nem, akkor valamilyen bejelentkezési folyamatot kezdeményez a felhasználó számára. Az üzeneteket a post-container nevű blokkban jelenítik meg. Ez a blokk lenne felelős

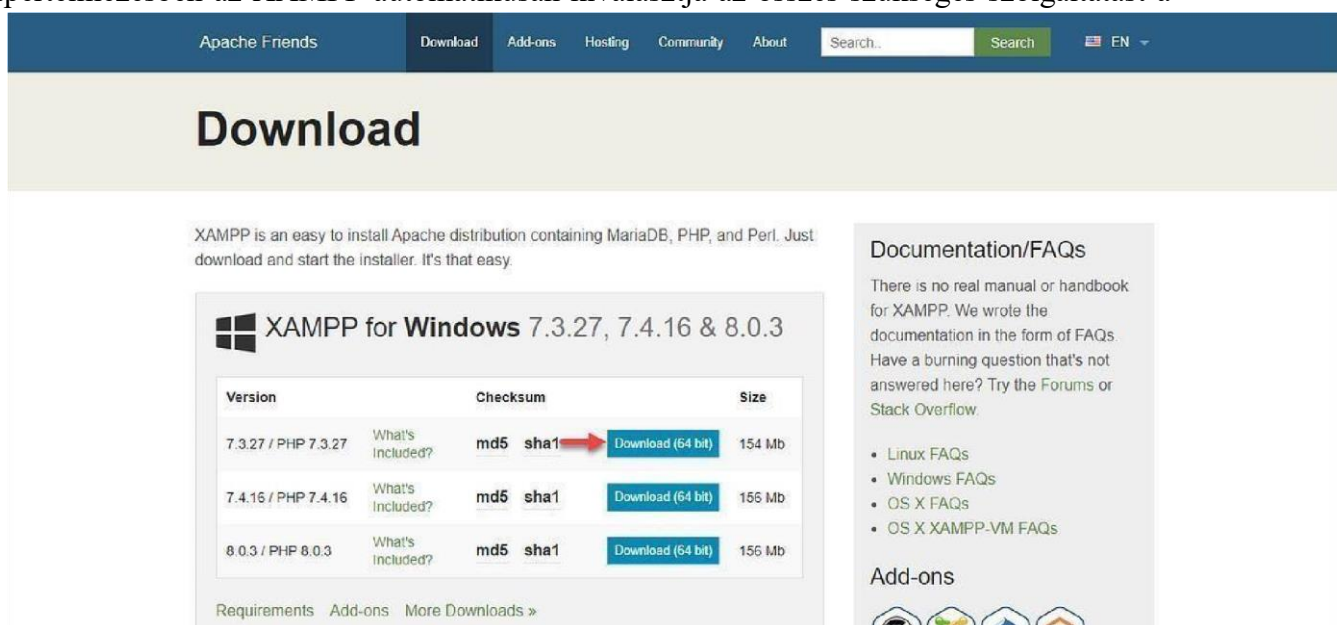
a posztok dinamikus megjelenítéséért, amelyek általában adatbázisból vagy más forrásból származó információk alapján generálódnak és rendelődnek ki.



A program telepítése

Az XAMPP telepítése során először is a hivatalos Apache Friends weboldalára kell navigálni <https://www.apachefriends.org/index.html> és innen kell letölteni az XAMPP szoftvert, egyszerűen a "Download" gombra kattintva. Fontos, hogy a letöltendő verziót az operációs rendszernek megfelelően válasszuk ki, legyen szó Windowsról, macOS-ről vagy Linuxról. Amint a telepítőfájl letöltődött, elég egy kattintással elindítani, majd amennyiben szükséges, adni kell az adminisztrátori jogosultságokat.

A telepítő felületén a felhasználónak lehetősége nyílik a kívánt komponensek kiválasztására. Alapértelmezésben az XAMPP automatikusan kiválasztja az összes szükséges szolgáltatást a



webfejlesztéshez, de ha szükség van specifikusabb konfigurációra (például csak a MySQL vagy a PHPMyAdmin szolgáltatások telepítésére), azt is beállíthatjuk.

A telepítési folyamat során szükség lesz a telepítési útvonal megadására, azaz azt a mappát kell kiválasztani, ahova az XAMPP-t telepíteni szeretnénk. Ezt követően egyszerűen folytathatjuk a telepítést, csupán egy gombnyomással.

Miután a telepítés befejeződött, elérhető lesz az XAMPP Vezérlőpanelje, ahol az Apache és a MySQL szolgáltatások indíthatók és leállíthatók. Fontos, hogy mindkét szolgáltatás futása aktív

legyen. Ha minden rendben van, akkor csak nyissa meg a webböngészőt, és látogasson el a <http://localhost> címre annak érdekében, hogy ellenőrizze az XAMPP sikeres telepítését és működését.

A Node.js telepítése egy viszonylag egyszerű folyamat, a Node.js telepítése során a kezdő lépés az, hogy felkeresse és böngéssze a hivatalos Node.js weboldalt <https://nodejs.org>, ahol elérhetővé válnak a legfrissebb verziók és a telepítőfájlok. Fontos megemlíteni, hogy a letöltendő verziót az operációs rendszernek megfelelően kell kiválasztani, legyen az Windows, macOS vagy Linux rendszer. Miután a megfelelő verziót kiválasztotta és letöltötte, elindíthatja a telepítőfájlt, és kövesse az ablakokban megjelenő utasításokat. A telepítési folyamat során elfogadhatja a licencszerződést és választhatja ki a telepítési helyet az alapértelmezett vagy saját preferenciái szerint. A telepítés befejezése után ellenőrizheti, hogy a Node.js sikeresen telepítve lett-e a rendszerére, például egy terminál vagy parancssor nyitásával és a `node -v` parancs kiadásával, amely megmutatja a telepített Node.js verziót. Amennyiben a telepítés sikeres volt, a Node.js készen áll az alkalmazásfejlesztésre, lehetőséget biztosítva a szerveroldali JavaScript alkalmazások létrehozására és futtatására.

A Prisma telepítése és konfigurálása számos lépésből áll, amelyek együttesen lehetővé teszik a hatékony adatbázis-interakciókat és a modellkezelést a Node.js-alapú alkalmazások fejlesztése során. Először is, a Prisma CLI telepítése szükséges, amelyhez szükség van a Node.js környezetre, ezáltal a telepítés előtt meg kell győződni arról, hogy a Node.js megfelelően vane telepítve a rendszerre. Ezt követően inicializálni kell egy új Prisma projektet, amely során a `prisma` mappa és a `schema.prisma` fájl létrejön, és ebben a fájlban lehet konfigurálni az adatbázis kapcsolatot és a modellt. A `schema.prisma` fájlban definiált sémát azután migrálni kell az adatbázisba a Prisma migrációs eszköz segítségével. Ezután telepíteni kell a Prisma Clientet a projektbe, hogy lehetővé tegye az adatbázissal való kommunikációt és a modellkezelést a kódban. Végül, miután ezek a lépések befejeződtek, a fejlesztő készen áll az alkalmazás kódjának írására és a Prisma által biztosított funkciók használatára az adatbázisinterakciók során.

A `@prisma/client` egy kulcsfontosságú csomag a Node.js alapú alkalmazásokhoz, amely lehetővé teszi az adatbázis-kezelés egyszerűsítését és hatékonyabbá tételét a Prisma ObjectRelational Mapping (ORM) keretrendszer segítségével. Az ORM-ek olyan eszközök, amelyek lehetővé teszik az alkalmazások számára, hogy objektumorientált módon kommunikáljanak az adatbázissal, ezzel minimalizálva a kézi SQL-lekérések és a kapcsolódó adatbázis-specifikus kód mennyiségét. A `@prisma/client` telepítése során általában az npm csomagkezelőt használjuk, mivel ez a Node.js projektek legelterjedtebb csomagkezelője. A telepítés rendkívül egyszerű, és csak egy egyszerű npm `install @prisma/client` parancs kiadását igényli a projekt gyökérkönyvtárában.

Ez a parancs automatikusan letölti és telepíti a Prisma klienskönyvtárat, amely szükséges az alkalmazásodhoz az adatbázisból való adatok lekérdezéséhez, módosításához és kezeléséhez. A Prisma ORM egy intuitív és hatékony interfészt biztosít az adatbázissal való kommunikációhoz, amely segít az alkalmazásfejlesztőknek gyorsan és hatékonyan működő adatbázis-kezelő megoldást létrehozni. Ezáltal az `@prisma/client` használata lehetővé teszi az

adatbázishoz való hozzáférést a Node.js projektjeidben, ami nélkülözhetetlen a modern webalkalmazások fejlesztéséhez.

A nodemon telepítése fejlesztési függőségként nagy előnyt jelent a helyi alkalmazásfejlesztés során. A nodemon segítségével könnyedén követhetjük az alkalmazásunk módosításait, például egy Express szerver esetén. Telepítése során az `npm install --save-dev` parancsot használjuk, amelyet a terminálba kell beírni. Miután telepítettük, a nodemon automatikusan figyeli a megadott könyvtárat vagy fájlokat. Amint bármelyik változik, az alkalmazást automatikusan újratölti, minimalizálva ezzel a fejlesztési időt és a manuális beavatkozás szükségességét. Ezáltal a fejlesztők gyorsabban és hatékonyabban dolgozhatnak, mivel nem kell kézzel újra indítaniuk az alkalmazást minden egyes változtatás után.

Az argon2 egy kiváló könyvtár, mely segítségével jelszavakat lehet biztonságosan kódolni. Ez a kódoló algoritmus kifejezetten a jelszavak védelmére és titkosítására lett tervezve, és nagy biztonságot nyújt a jelszavak tárolásához és ellenőrzéséhez. Az argon2 használata különösen ajánlott az olyan alkalmazásokban, amelyek szigorú biztonsági követelményeket támasztanak a felhasználói adatok kezelésével kapcsolatban.

Általában az npm használata ajánlott az argon2 telepítéséhez, különösen, ha Node.js alapú alkalmazást fejlesztesz. Az npm (Node.js Package Manager) egy megbízható és elterjedt csomagkezelő, amely lehetővé teszi a Node.js projektek számára a külső csomagok könnyű és hatékony kezelését.

Ezzel a paranccsal letöltöd és telepíted az argon2 csomagot a projektedbe. Miután telepítetted, könnyedén használhatod az argon2 funkcionalitását a jelszavak biztonságos kódolásához és ellenőrzéséhez az alkalmazásodban.

Az argon2 használata jelentősen növeli a jelszavak biztonságát, mivel a jelszavakat erősített módon kódolja, így nehezebbé téve azoknak, akik illetéktelenül hozzáférnének az adatokhoz. Ezáltal az argon2 nagyobb védelmet nyújt a különböző támadásokkal, például a brute force támadásokkal és a jelszavak feltörésével szemben. A terminálba tudod telepíteni az argon2 az `npm install argon2` paranccsal.

A CORS egy közbeépülő, amelyet gyakran alkalmaznak az Express.js keretrendszerben annak érdekében, hogy lehetővé tegyék a különböző eredetű HTTP kérések biztonságos kezelését. Ezáltal a szerver engedélyezheti vagy korlátozhatja az erőforrások hozzáférését a kliensek részéről, biztosítva ezzel a szükséges biztonságot az adatcseréhez.

Amikor egy kliensoldali alkalmazás kérést küld egy Express.js szerverhez, a böngésző biztonsági korlátozásai miatt a szerver csak akkor fogadja el a kérést, ha a kérés eredete megegyezik a szerver eredetével vagy ha az szerveroldal kifejezetten engedélyezi azt.

A CORS közbeépülő segítségével az Express.js alkalmazások könnyen konfigurálhatók úgy, hogy válaszoljanak a CORS fejlécekre és engedélyezzék a különböző eredetű kéréseket. Így a

szerver szabályozhatja, hogy mely kérések érkezhetnek meg, melyeket fogad el, és milyen feltételekkel. Ezáltal biztosítva van az alkalmazás biztonsága és a felhasználói adatok védelme a különböző eredetű kérésekkel szemben.

A CORS közbeépülőt az Express.js alkalmazásba való telepítéséhez először is szükség van a Node.js projekt környezetére, ahol az Express.js alkalmazásod fut. Ha még nem tetted meg, telepítsd a Node.js-t a számítógépedre. A `cors` az `npm install cors` paranccsal tudjuk telepíteni. Azután nyisd meg a parancssor vagy terminál ablakot, és navigálj az Express.js projekt főkönyvtárába. `npm install cors` paranccsal tudjuk telepíteni. Ez a parancs letölti és telepíti a CORS közbeépülőt a projektbe, amely után már használhatod azt az Express.js alkalmazásodban. Fontos megjegyezni, hogy a telepítés előtt győződj meg róla, hogy az Express.js alkalmazásodhoz szükséges fájlban általában az **app.js** vagy **server.js** nevű fájlban beimportáltad-e a CORS közbeépülőt és konfiguráltad-e a használatát az alkalmazásodban.

Az Express egy népszerű webalkalmazás-keretrendszer Node.js-hez, amely lehetővé teszi a gyors és hatékony webalkalmazások fejlesztését. A telepítés során használható csomagkezelők könnyen elérhetővé teszik az Express keretrendszert a projekt számára. Ha az `npm` csomagkezelőt használjuk, a telepítés egy egyszerű `npm install express` parancs segítségével végezhető el a terminálban. Ez a parancs letölti és telepíti az Express keretrendszert a projekt függőségei közé, így lehetővé téve az alkalmazásunk számára, hogy az Express nyújtotta funkciókat és lehetőségeket felhasználja. A telepítés után az Express segítségével könnyedén definiálhatjuk a különböző útvonalakat, kezelhetjük a kéréseket és válaszokat, és építhetünk erőteljes, dinamikus webalkalmazásokat a Node.js platformra.

A telepítés után az Express használatával könnyedén definiálhatunk különböző útvonalakat, kezelhetjük a kéréseket és válaszokat, valamint építhetünk erőteljes, dinamikus webalkalmazásokat a Node.js platformra. Az Express segítségével könnyedén implementálhatunk `middleware`-eket, kezelhetjük az adatbázis-kommunikációt, és kialakíthatjuk az alkalmazásunk struktúráját a legjobb gyakorlatoknak megfelelően. Ezáltal az Express egy hatékony eszköz a Node.js alapú webalkalmazások fejlesztéséhez, amely lehetővé teszi számunkra, hogy hatékonyan és rugalmasan dolgozzunk az alkalmazásunkon, és kihozzuk belőle a legtöbbet.

A `jsonwebtoken` egy olyan könyvtár, amely kifejezetten a JSON Web Token generálására és ellenőrzésére lett tervezve. A JWT egy kriptográfiai alapú token, amelyet gyakran használnak a webalkalmazásokban és API-kban az autentikációhoz és a felhasználói engedélyek kezeléséhez. A `jsonwebtoken` könyvtár telepítése során az `npm` használata ajánlott. Az `npm` használata esetén a telepítés egy egyszerű `npm install jsonwebtoken` parancs segítségével végezhető el a terminálban vagy a parancssorban.

Ez a parancs letölti és telepíti a `jsonwebtoken` könyvtárat a projekt függőségei közé, így lehetővé téve az alkalmazásod számára, hogy az JWT-kat használja a felhasználók azonosítására és az autentikációhoz szükséges információk hitelesítésére. Miután telepítetted a `jsonwebtoken` könyvtárat, könnyedén kezelheted a JWT-kat az alkalmazásodban.

Továbbfejlesztése

Szeretnék tovább fejleszteni az alkalmazást néhány új funkcióval és fejlesztéssel:

Először is, szeretném kibővíteni a felhasználói profilokat és beállításokat, hogy a felhasználók személyre szabhassák a profiljukat, beleértve az avatárokat. Ez segítene a felhasználóknak az alkalmazásban való egyéni azonosításban és a személyes élményük testre szabásában.

Másodszor, tervezem egy közösségi platform létrehozását az alkalmazásban, ahol a felhasználók megoszthatják saját tartalmaikat, véleményüket, és közvetlenül interakcióba léphetnek egymással. Ez magában foglalná a posztolást, kommentelést és üzenetküldést, ami lehetővé tenné a közösségi interakciók bővítését és a felhasználók közötti kapcsolatok erősítését.

Ezen kívül be szeretném vezetni az alkalmazásba valós idejű értesítések rendszerét, hogy a felhasználók azonnal értesüljenek az eseményekről, például új üzenetek, követők vagy tartalomfrissítések esetén. Szintén fontosnak tartom az adatvizualizáció és analitika fejlesztését, hogy a felhasználók könnyen követhessék az alkalmazásban való tevékenységüket, és megfigyelhessék a trendeket egy intuitív dashboard vagy analitikai felület segítségével.

Az alkalmazás funkcionalitásának további bővítése érdekében tervezem kiegészítő alkalmazások integrálását, például egy fizetési rendszer vagy harmadik fél API-jának beillesztését, hogy fokozzam a felhasználói élményt. A mobilalkalmazás fejlesztése az alkalmazás mellé egy új platformot kínálna, amely lehetővé teszi a felhasználóknak, hogy a funkciókat mobil eszközökről is elérjék.

Végül, szeretném biztosítani a globális lokalizációt és többnyelvű támogatást az alkalmazás számára, hogy kiterjedjen a nemzetközi felhasználókra is, és hogy a felhasználók különböző nyelveken tudjanak kommunikálni az alkalmazásban.

Kosár

A kosarad üres.

Irodalomjegyzék:

W3Schools:

- Link: <https://www.w3schools.com> ○ Oldal címe: W3Schools Online Web Tutorials

Tailwind CSS: ○ Link:

<https://tailwindcss.com/docs>

- Oldal címe: Tailwind CSS Documentation

W3Schools Photogalary:

- Link: https://www.w3schools.com/howto/howto_js_slideshow.asp ○ Oldal címe: W3Schools Online Web Diavetítés Galéria

Tailwind CSS login és regisztrációs példák: •

Link:

<https://tailwindcss.com/docs/examples>

- Oldal címe: Tailwind CSS Documentation - Examples

Plágium-nyilatkozat

Alulírott.....Krepsz Balázs András.....(név)

..... Gyula (szül.hely)

..... 2005.Január.27(szül.idő)

..... Krepszné Molnár Zsuzsanna Katalin.....(anyja neve) jelen nyilatkozat aláírásával kijelentem, hogy a

.....Ízletes Étterem..... című záródolgozat (a továbbiakban: dolgozat) önálló munkám, a dolgozat készítése során betartottam a szerzői jogról szóló 1999. évi LXXVI. tv. szabályait, valamint a Békéscsabai Szakképzési Centrum által előírt, a dolgozat készítésére vonatkozó szabályokat.

Tudomásul veszem, hogy a dolgozat esetén plágiumnak/szerzői jogsértésnek számít:

- szó szerinti idézet közlése idézőjel és hivatkozás megjelölése nélkül;
- tartalmi idézet hivatkozás megjelölése nélkül;
- más szerző publikált gondolatainak saját gondolatként való feltüntetése

Kijelentem továbbá, hogy a dolgozat készítése során az önálló munka kitétel tekintetében a konzulenszt, illetve a feladatot kiadó oktatót nem tévesztettem meg.

Jelen nyilatkozat aláírásával tudomásul veszem, hogy amennyiben bizonyítható, hogy a dolgozatot nem magam készítettem vagy a dolgozattal kapcsolatban szerzői jogsértés ténye merül fel, a Békéscsabai Szakképzési Centrum a dolgozatot elégtelennek minősíti.

Békéscsaba, 2024.03.....hó ...26...nap

Tanuló aláírása