Noel Candy Shop

Vizsgaremek Dokumentáció

Készítette:

Szamosfalvi Szonja,

Gazdik Dorina,

Varga Lilla

Tartalomjegyzék

1. Bevezetés

A **Noel Candy Shop** egy online webshop, amely amerikai és japán édességek értékesítésére specializálódik. A projekt célja, hogy egy már működő fizikai üzlet számára biztosítson egy digitális értékesítési csatornát, így a vásárlók kényelmesen, online is elérhessék a termékeket.

A webshop modern technológiákat alkalmaz a fejlesztés során. A frontend Angular keretrendszerrel készül, amely biztosítja a gyors és dinamikus felhasználói élményt. Az adatok tárolásához és backendként Firebase adatbázist használunk, amely skálázható és valós idejű adatszinkronizálást tesz lehetővé.

Ez a dokumentáció bemutatja a webshop fejlesztési folyamatát, technikai részleteit, a rendszer működésének feltételeit, valamint a használati és karbantartási útmutatókat.

2. A szoftver célja

A **Noel Candy Shop** webshop célja, hogy a fizikai üzlet kínálatát online elérhetővé tegye a vásárlók számára. Jelenleg a bolt termékei csak helyben vásárolhatók meg, ami földrajzilag és időben is korlátozza a vásárlási lehetőségeket. A webáruház lehetővé teszi, hogy a vásárlók bárhonnan böngészhessék a kínálatot, kiválaszthassák a kívánt termékeket, és online rendelést adhassanak le.

A webshop főbb célkitűzései:

- **Elérhetőség bővítése**: Az édességek szélesebb vásárlói körhöz juthatnak el, nemcsak a helyi bolt látogatóihoz.
- **Kényelmes vásárlás**: Az ügyfelek otthonról vagy útközben is böngészhetik és megrendelhetik a termékeket.
- **Automatizált készletkezelés**: A rendszer nyomon követi a raktárkészletet, csökkentve az adminisztrációs terheket.
- **Felhasználóbarát felület**: Az Angular keretrendszerrel készült modern és reszponzív weboldal könnyen kezelhető és gyors felhasználói élményt biztosít.
- **Biztonságos adatkezelés**: A Firebase adatbázis gondoskodik az adatok biztonságos tárolásáról és a valós idejű szinkronizációról.

A webshop nemcsak az eladásokat növelheti, hanem hozzájárulhat az ügyfélélmény fejlesztéséhez és az üzlet digitális jelenlétének megerősítéséhez.

3. Fejlesztői dokumentáció

3.1. Felhasznált technológiák

Frontend:

- Angular (frontend keretrendszer)
- TypeScript (programozási nyelv)
- Bootstrap (felhasználói felület stílus, reszponzivitás)
- Firebase (adatok tárolása, valós idejű szinkronizáció)
- HTML, CSS (a HTML az oldal szerkezetét, tartalmát biztosítja, a CSS pedig az oldal megjelenítését és stílusát alakítja)

Backend:

- Firebase Authentication (felhasználói azonosítás és jogosultságkezelés)
- Firebase Realtime Database (adatok tárolása)
- Firebase Storage (képek tárolása)
- Node.js + Express.js (backend szerver futtatása és API végpontok kezelése)
- JWT (hitelesítés és jogosultságkezelés)
- bcrypt (jelszavak titkosítása és biztonságos tárolása)

3.2. Fejlesztőkörnyezet

Frontend:

- Fejlesztői eszközök: Visual Studio Code, Node.js, Angular CLI
- Függőségkezelés: npm
- Verziókezelés: Git
- Böngésző: Chrome

Backend:

- Fejlesztői eszközök: Visual Studio Code, Postman (API teszteléshez)
- Föggőségkezelés: npm
- Verziókezelés: Git

3.3. Komponensek és szervizek

Frontend komponensek:

AppComponent - Az alkalmazás fő komponense, amely az egyes oldalak megjelenítéséért felel.

Osztályváltozók:

- title: A komponens címe, amely az alkalmazás nevét tárolja. Jelen esetben 'webshop'.

NavbarComponent - A navigációs menü, amely a felhasználók számára biztosítja az egyes oldalak közötti váltást.

Osztályváltozók:

- cart: Ez az objektum tartalmazza a kosár adatait, amelyeket a CartService-en keresztül frissítünk.
- searchQuery: A keresési lekérdezés sztringje, amelyet a felhasználó beír a kereső mezőbe.
- searchResult: A keresési eredmények listája, amely a keresési lekérdezés alapján jelenik meg.

Termékek:

- products: Ez egy statikus lista a termékekről, ahol minden terméknek van egy neve és kategóriája. Ezt használhatnád a keresési funkcióhoz, hogy a felhasználó megtalálja a kívánt terméket.

Metódusok:

- setSearch(car: string): Ez a metódus a SearchService-et hívja meg, és átadja a keresési lekérdezést. Az SearchService-nek gondoskodnia kell a keresés feldolgozásáról.
- addOrder(): Hozzáadja az aktuális kosár tartalmát a rendeléshez a CartService-en keresztül. A metódus jelenleg a kosár tartalmát rendelésbe helyezi, de az üzleti logika szerint további részletek is szükségesek lehetnek.
- onSearch() (kommentelve): Ez a metódus végzi el a keresést a products tömbben. Ha nincs találat, figyelmeztetést ad.

Konstruktor:

- constructor(private cartServ: CartService, public searchServ: SearchService): Az osztály konstruktorában a CartService és a SearchService injektálva van. A CartService-t használjuk a kosár adatainak lekérésére.

FooldalComponent - A kezdőoldal tartalmának megjelenítéséért felelős komponens.

Osztályváltozók:

- products: Ez az objektum tömbje, amely tartalmazza a termékek adatokat, mint például az id, a name, a image (a termékhez tartozó képfájl neve), és a link (ahová a felhasználó kattintva elérheti a terméket).

Metódusok:

- redirectToUrl(url: string): Ez a metódus átirányítja a felhasználót a megadott URL-re. A window.location.href használatával történik az átirányítás.
- scrollToTop(): Ez a metódus az oldal tetejére görgeti a nézetet, sima animációval (behavior: 'smooth').

Konstruktor:

- constructor(private productsServ: ProductlistService): Az osztály konstruktorában a ProductlistService szolgáltatás injektálva van, de úgy tűnik, hogy a szolgáltatás használata jelenleg nem történik meg a kódban. Ha szükséges, a termékek listáját a szolgáltatáson keresztül is kezelhetjük, például adatlekéréshez.

ProductListComponent - Az összes termék listázását végzi.

Osztályváltozók:

- products: A termékek listáját tároló változó, amelyet a ProductlistService szolgáltatás segítségével tölt be.
- newProduct: Az új termék adatait tároló objektum, amelyet a felhasználó a termék hozzáadásakor kitölthet.
- searchText: A keresési szöveg, amely alapértelmezésben a "Half" szót tartalmazza.
- columns: A termékek listáján megjelenítendő oszlopok neveit tartalmazó tömb.

Metódusok:

- constructor(private pserv: ProductlistService, private cartServ: CartService): Az osztály konstruktora, amelyben a ProductlistService és CartService szolgáltatásokat injektálja. A getProducts() metódus segítségével lekéri a termékeket és feltölti a products változót.
- addProduct(): Hozzáad egy új terméket a terméklistához a newProduct objektum alapján, majd visszaállítja a newProduct objektumot egy üres állapotba.
- sortProducts(order: string): A termékek listáját rendezi az ár alapján, az order paramétertől függően (növekvő vagy csökkenő sorrend).
- editProduct(product: any): Frissíti egy meglévő termék adatait a ProductlistService segítségével.
- deleteProduct(product: any): Törli a megadott terméket a terméklistából a ProductlistService segítségével.

CandyListComponent - Az édességek kategóriájának megjelenítésére szolgáló komponens.

Osztályváltozók:

- candies: Tárolja a cukorkákat (termékeket), amelyek a ProductlistService-től érkeznek. A getProducts() metódussal lekéri a szolgáltatásból.

Metódusok:

- constructor(private prodServ: ProductlistService): Az osztály konstruktorában a ProductlistService szolgáltatás segítségével lekéri a termékeket és azokat a candies változóba menti.

Szolgáltatók:

- ProductlistService: Ez a szolgáltatás felelős a termékek adatainak kezeléséért, lekéréséért a backendről. A getProducts() metódusával kérdezi le a termékeket.

CandiesComponent - Egy adott édességcsoport (pl. cukorkák) részleteinek megjelenítése.

Osztályváltozók:

- candies: A @Input() dekorátorral jelölt változó, amely tartalmazza a termékeket (cukorkákat), amelyeket a szülő komponens ad át.
- showNotification: Boolean típusú változó, amely azt jelzi, hogy egy termék sikeresen hozzáadásra került a kosárhoz. Alapértelmezett értéke false.
- searchText: A felhasználó által beírt keresési szöveget tároló változó.
- order: A termékek rendezésére használt változó, amely tartalmazza a kívánt rendezési sorrendet.

Metódusok:

- kosarbaRak(candy: any, quantity_in: string):
  - Ez a metódus hozzáadja a cukorkát a kosárhoz a CartService segítségével.
  - Az quantity_in változóból létrehozza a terméket, amely tartalmazza az árut és a mennyiséget.
  - Az értesítési állapotot aktiválja és a kosár tartalmát naplózza.
  - 500 ms után eltűnik az értesítés.
- sortProducts(order: string): A termékeket rendezi az order változóban megadott mód szerint (például: "low-to-high", "high-to-low").
- scrollToTop(): A weboldal tetejére görgeti az oldalt, ha a felhasználó ezt kérte, sima animációval.

Szolgáltatók:

- CartService: A kosár kezeléséhez, termékek hozzáadásához és a kosár tartalmának lekéréséhez használt szolgáltatás.

- SearchService: A keresési szöveget kezeli, és figyeli a szülő komponens által küldött frissítéseket.

ChipsekComponent - A chips kategória termékeinek megjelenítésére szolgáló komponens.

Osztályváltozók:

- candies:
  - A cukorkák (vagy chipsek) listáját tartalmazza, amelyeket a ProductlistService-től kapunk.
  - Az @Input() dekorátorral van jelölve, így kívülről is átadható.
- showNotification: Egy boolean változó, amely az értesítések láthatóságát szabályozza. Kezdetben false, de amikor egy termék hozzáadásra kerül a kosárhoz, true értékre változik, majd rövid idő elteltével visszaáll false-ra.

Metódusok:

- constructor(private cartService: CartService, private cdr: ChangeDetectorRef, private prodServ: ProductlistService):
  - A komponens konstruktorában a ProductlistService-től lekérdezi a termékeket (candies), és a CartService segítségével kezeli a kosarat.
  - Az adatokat a prodServ.getProducts()-on keresztül szerzi be és frissíti a candies változót.
- kosarbaRak(candy: any, quantity_in: string):
  - A metódus hozzáad egy terméket (candy) a kosárhoz. A quantity_in a termék mennyisége, amit a kosárhoz adunk.
  - A kosárba való hozzáadás után az értesítés (showNotification) láthatóvá válik, majd egy idő után eltűnik.
- sortProducts(order: string): A termékek rendezése árat szerint (price) történik. Ha az order értéke 'high-to-low', akkor csökkenő sorrendbe rendezi őket, ha 'low-to-high', akkor növekvő sorrendbe.
- scrollToTop(): A képernyőt a tetejére görgeti sima animációval.

Szolgáltatók:

- CartService: A kosár működését kezeli, beleértve a termékek hozzáadását a kosárhoz és a kosár tartalmának lekérését.
- ProductlistService: A termékek listáját szolgáltatja a komponens számára. A getProducts() metódussal lekéri a termékeket a backendről.

CsiposChipsekComponent - A csípős chipsek listázása és megjelenítése.

Osztályváltozók:

- candies: A chipsek listáját tartalmazza, amelyeket a ProductlistService-től kapunk. Az @Input() dekorátorral van jelölve, így kívülről is átadható.

- showNotification: Egy boolean változó, amely az értesítések megjelenítését szabályozza. Kezdetben false, de amikor egy termék hozzáadásra kerül a kosárhoz, true-ra változik, majd rövid idő elteltével visszaáll false-ra.

Metódusok:

- constructor(private cartService: CartService, private cdr: ChangeDetectorRef, private prodServ: ProductlistService):
  - Az komponens konstruktorában a ProductlistService segítségével lekérdezi a termékeket (candies), és a CartService-t használja a kosár kezelésére.
  - A termékek a prodServ.getProducts() metódus meghívásával kerülnek lekérésre, és a választás alapján frissíti a candies változót.
- kosarbaRak(candy: any, quantity_in: string):
  - A metódus egy terméket (candy) ad hozzá a kosárhoz. A quantity_in az a mennyiség, amelyet a kosárhoz adunk.
  - Az értesítés (showNotification) láthatóvá válik, amikor a termék hozzáadódik a kosárhoz, majd 500 ms után eltűnik.
- sortProducts(order: string): A termékek rendezése történik az áruk szerint (price). Ha az order értéke 'high-to-low', akkor csökkenő sorrendbe rendezi őket, ha 'low-to-high', akkor növekvő sorrendbe.
- scrollToTop(): A képernyőt a tetejére görgeti sima animációval.

Szolgáltatók:

- CartService: A kosár működését kezeli, beleértve a termékek hozzáadását és eltávolítását.
- ProductlistService: A termékek listáját szolgáltatja a komponens számára. A getProducts() metódus segítségével lekérjük a termékeket.

EdesGumicukrokComponent - Édes gumicukrok megjelenítése.

SavanyuGumicukrokComponent - Savanyú gumicukrok megjelenítése.

UditokComponent - Az üdítők kategóriájának megjelenítése.

EdesUditokComponent - Az édes üdítők kategóriájának megjelenítése.

SavanyuUditokComponent - A savanyú üdítők kategóriájának megjelenítése.

CartComponent (KosarComponent) - A bevásárlókosár kezelése.

Osztályváltozók:

- cartItems: any[]: Kosárban lévő termékek.
- totalPrice: number: Kosár teljes ára.

Metódusok:

- ngOnInit(): Betölti a kosár tartalmát.
- loadCart(): Kosár frissítése.
- removeFromCart(productId: string): Termék eltávolítása a kosárból.
- calculateTotalPrice(): Kosár végösszegének kiszámítása.
- goCheckOut(): Fizetési oldalra navigálás.

CheckoutComponent - A rendelés véglegesítéséért felelős komponens.

Osztályváltozók:

- order: Ez egy objektum, amely a rendelési információkat tartalmazza (pl. név, cím, email, telefonszám stb.).
- isSubmitting: Egy logikai változó, amely jelzi, hogy folyamatban van-e a rendelés leadása. Ez segít elkerülni a duplázott rendeléseket.
- successMessage: A rendelés sikerességét jelző üzenet, amely csak akkor jelenik meg, ha a rendelés sikeresen lezajlott.

Metódusok:

- submitOrder(form: any): Ezt a metódust hívják meg a rendelés leadásakor. Először ellenőrzi, hogy a form valid-e és hogy nincs-e éppen folyamatban rendelés leadás. Ha minden rendben van, a rendelést hozzáadja a cartService segítségével, és megjeleníti a sikeres rendelés üzenetét. A rendelés sikeres leadása után a formot visszaállítja és átirányítja a felhasználót a főoldalra.

OrdersListComponent - A felhasználó által leadott rendelések listázását végzi.

Osztályváltozók:

- orders: A rendeléseket tároló változó, amely a CartService-ből származó adatokat tárolja.

Metódusok:

- constructor(private cart: CartService): Az osztály konstruktorában a CartService-t injektálja, és az getOrders() metódus segítségével lekéri a rendeléseket, amelyeket az orders változóba helyez el.

KapcsolatComponent - Kapcsolati űrlap és elérhetőségi információk megjelenítése.

Osztályváltozók:

- address: Ez a változó tárolja az üzlet címét, amelyet később a Google Maps-en való kereséshez használunk.

Metódusok:

- getGoogleMapsUrl(): Ez a metódus visszaadja a Google Maps URL-jét, amely tartalmazza az üzlet címét, így a felhasználók egy kattintással elérhetik a Google Maps-en a cím helyét.

FiokomComponent - A felhasználói fiók adatait kezeli és jeleníti meg.

Osztályváltozók:

- userEmail: string – A felhasználó email címe, amelyet a localStorage-ból tölt be.

Metódusok:

- ngOnInit() – Az inicializálás során beolvassa a localStorage-ból a bejelentkezett felhasználó email címét.
- editProfile() – Profil szerkesztési funkció (jelenleg csak konzolba ír ki üzenetet).
- logout() – Kijelentkezési funkció, amely eltávolítja az emailt a localStorage-ból és a bejelentkező oldalra navigál.

LoginComponent - A bejelentkezési felület.

Osztályváltozók:

- loginForm: FormGroup – A bejelentkezési űrlapot reprezentáló FormGroup.

Metódusok:

- constructor(fb: FormBuilder, router: Router) – Inicializálja az űrlapot és az útválasztót.
- onSubmit() – Ellenőrzi az e-mail és jelszó helyességét a localStorage alapján, és sikeres bejelentkezés esetén a felhasználót a fiók oldalára navigálja, ellenkező esetben hibaüzenetet jelenít meg.

RegistrationComponent - A regisztrációs űrlap megjelenítése és kezelése.

Osztályváltozók:

- registrationForm: FormGroup – A regisztrációs űrlapot reprezentáló FormGroup.

Metódusok:

- constructor(fb: FormBuilder, router: Router) – Inicializálja az űrlapot és az útválasztót.

- passwordsMatchValidator(group: FormGroup) – Ellenőrzi, hogy a megadott jelszó és megerősítő jelszó megegyezik-e.
- emailValidator(control: any) – Ellenőrzi, hogy az e-mail cím tartalmazza-e az @ és . karaktereket.
- onSubmit() – Ha az űrlap érvényes, menti a felhasználó adatait a localStorage-be, és átirányítja a bejelentkezési oldalra.

AdminLoginComponent - Adminisztrátori bejelentkezés felülete.

Osztályváltozók:

- loginForm!: FormGroup - Az admin bejelentkezési űrlap adatait tárolja.
- errorMessage: string - A bejelentkezési hibák üzeneteit kezeli.
- adminEmail: string - Az adminisztrátor e-mail címe (admin@example.com).
- adminPassword: string - Az adminisztrátor jelszava (admin123).

Metódusok:

- ngOnInit() - Inicializálja az adminlogin űrlapot a megfelelő validációkkal.
- onSubmit() - Ellenőrzi, hogy az admin belépési adatok helyesek-e, és hibaüzenetet jelenít meg, ha nem egyeznek.

WishlistComponent (KivansaglistamComponent) - A kívánságlista kezelése.

Osztályváltozók:

- termekek: A kívánságlistán szereplő termékeket tartalmazó tömb. Minden elem egy Termek típusú objektum, amely a termék nevét (nev) és képét (image) tárolja.
- ujTermekNev: Az új termék neve, amit a felhasználó szeretne hozzáadni a kívánságlistához.
- ujTermekKep: Az új termék képének URL-je.

Metódusok:

- hozzaadTermek(): Hozzáad egy új terméket a kívánságlistához, ha mindkét mező (név és kép) kitöltésre került. A hozzáadás után a ujTermekNev és ujTermekKep változókat kiüríti.
- torolTermek(index: number): Eltávolít egy terméket a kívánságlistáról a megadott index alapján.

Márkákhoz tartozó komponensek:

Ezek a komponensek az egyes termékmárkák egyedi oldalainak megjelenítésére szolgálnak (ugyan azok az osztályváltozók és metódusok mindegyik komponensnél csak a nevük változik):

ArizonaComponent

Osztályváltozók:

- arizona: Arizona[] – Terméklista az Arizona márka italairól.
- showNotification: boolean – Értesítés megjelenítéséhez.

Metódusok:

- kosarbaRak(i: number) – Hozzáadja az adott terméket a kosárhoz és megjelenít egy értesítést.
- sortProducts(order: string) – Rendezési opciók (ár szerint növekvő/csökkenő sorrend).
- scrollToTop() – Görgetés az oldal tetejére.

BazookaComponent

CheetosComponent

FantaComponent

GhostComponent

NerdsComponent

PrimeComponent

PringlesComponent

SkittlesComponent

SourComponent

TakisComponent

WarheadsComponent

Frontend szervizek:

AuthService - A felhasználói hitelesítés kezelése.

Osztályváltozók:

- registeredUser: A regisztrált felhasználó adatainak tárolására használt változó, amely null értékre van állítva kezdetben.

Metódusok:

- constructor(): Az osztály konstruktora, amely az AuthService példányosításához szükséges.

- register(userData: { firstName: string; lastName: string; email: string; password: string; confirmPassword: string; phone: string }): A felhasználó regisztrációját végzi el. Az átadott userData objektumot másolja be a registeredUser változóba, majd a konzolra kiírja a regisztrált felhasználó adatait.
- login(email: string, password: string): boolean: A felhasználó bejelentkezését végzi el az email és jelszó ellenőrzése alapján. Ha a bejelentkezés sikeres (a registeredUser létezik, és az email és jelszó megegyezik), akkor true-t ad vissza, különben false-t.
- isRegistered(): boolean: Ellenőrzi, hogy van-e regisztrált felhasználó (vagyis, hogy a registeredUser nem null).
- getRegisteredUser(): Visszaadja a regisztrált felhasználó adatokat (a registeredUser változót).
- logout(): void: Kijelentkezteti a felhasználót, és törli a registeredUser adatokat. A konzolra kiírja a kijelentkezés sikerességét.

CartService - A bevásárlókosár műveleteinek kezelése (pl. termék hozzáadása, eltávolítása, összeg kiszámítása).

Osztályváltozók:

- cart: A vásárlókosár termékeit tároló tömb. Kezdetben üres.
- ordersRef: Az AngularFireList típusú referencia a Firebase Realtime Database-ben található "orders" listához.

Metódusok:

- constructor(private db: AngularFireDatabase): Az osztály konstruktora, amely inicializálja az ordersRef-et a Firebase adatbázis "orders" referenciájával, és betölti a kosarat (loadCart()).
- addToCart(candy: any): A terméket hozzáadja a kosárhoz. Ha a termék már benne van a kosárban, akkor frissíti annak mennyiségét. Ha új termékről van szó, akkor hozzáadja a kosárhoz. A változásokat elmenti a saveCart() metódussal.
- getCartItems(): Visszaadja a kosárban lévő termékeket (cart tömb).
- getTotalPrice(): number: Kiszámítja és visszaadja a kosárban lévő termékek összesített árát (mennyiség és ár alapján).
- removeFromCart(productKey: string): Eltávolítja a terméket a kosárból a termék kulcsa alapján, majd elmenti a frissített kosarat.
- private saveCart(): Elmenti a kosarat a localStorage-ba JSON formátumban.
- private loadCart(): Betölti a kosarat a localStorage-ból, ha ott van mentett adat.
- addOrder(order: any): Rendelést ad hozzá a Firebase adatbázishoz. Az order objektumot kiegészíti egy Uid-val és a kosár elemeivel, majd az ordersRef segítségével elmenti a rendelést. A kosár kiürül a rendelés leadása után.

- getOrders(): Lekéri az összes rendelést a Firebase adatbázisból és visszaadja azokat egy átalakított formátumban, amely tartalmazza a kulcsot (key) is.

ProductListService - A terméklisták kezelése és a termékek adatainak lekérése.

Osztályváltozók:

- produtctsSub: Egy BehaviorSubject, amely tárolja a termékek adatait. Kezdetben null értékre van állítva, és a termékek adatainak frissítése során figyelhetjük.
- productsRef: Az AngularFireList<any> típusú változó, amely az adatbázisban található "products" lista referencia.

Metódusok:

- getProducts(): Lekéri az összes terméket az adatbázisból. A snapshotChanges() metódust használja, hogy figyelje az adatbázisban történt változásokat, és visszaadja azokat.
- pushProduct(candy: any): Új terméket ad hozzá az adatbázishoz a push() metódus segítségével.
- deleteProduct(candy: any): Eltávolítja a megadott terméket az adatbázisból a remove() metódussal.
- updateProduct(candy: any): Frissíti a megadott terméket az adatbázisban a update() metódussal. A termék key-ét eltávolítja a frissítés előtt, hogy ne szerepeljen az adatban.

SearchService - A termékkeresés biztosítása az oldalon.

Osztályváltozók:

- searchSub: Egy BehaviorSubject<string> típusú változó, amely tárolja a keresési szót vagy kifejezést. Kezdetben üres stringre van állítva ("").

Metódusok:

- getSearch(): Visszaadja a searchSub BehaviorSubject-et, hogy más komponensek figyelni tudják a keresési szót.
- setSearch(text: string): Beállítja a keresési szót a searchSub-ban a next() metódus segítségével. Ezzel frissíti a keresési kifejezést és értesíti az összes előfizetőt a változásról.

SortPipe - A termékek szűrésére és rendezésére szolgáló pipe.

Metódusok:

- transform(candies: any[], order: string): any:
- A transform metódus felelős a szűrési logika megvalósításáért, amely rendezi a candies tömböt a megadott order paraméter alapján.
- Ha a candies tömb nincs megadva, null-t ad vissza.

- Ha az order értéke nem adható meg, a tömb változatlanul marad.
- A két lehetséges rendezettségi sorrend:
- 'high-to-low': a termékek árát csökkenő sorrendben rendezi.
- 'low-to-high': a termékek árát növekvő sorrendben rendezi.

3.4. Adatbázis szerkezete, táblák neve, mezőnevek

A Firestore adatbázis az alábbi kollekciókból áll:

users (felhasználói adatok)

products (termékinformációk)

orders (rendelések)

products tábla szerkezete:

id (string) - Egyedi azonosító.

name (string) - Termék neve.

price (number) - Ár.

description (string) - Termék leírása.

imageUrl (string) - Kép elérési útja.

3.5. Végpontok megnevezése, leírása

3.6. Továbbfejlesztési lehetőségek

Mobilalkalmazás fejlesztése - A webshop natív mobilappként való elérhetővé tétele Android és iOS platformokon.

Felhasználói profil oldal - Személyes adatok és rendelések megtekintése.

Termékértékelések - Felhasználói vélemények és csillagozás.

Push értesítések - Új termékekről vagy akciókról értesítés küldése a felhasználóknak.

5. Felhasználói dokumentáció

5.1. A webshop bemutatása

A Noel Candy Shop egy online webáruház, amely amerikai és japán édességek széles választékát kínálja. A webshop célja, hogy a vásárlók kényelmesen, online is elérhessék a termékeket, anélkül, hogy személyesen kellene ellátogatniuk a boltba.

A webshopot számítógépen, laptopon, tableten és okostelefonon egyaránt lehet használni, internetkapcsolattal rendelkező böngészőprogram segítségével.

5.2. A használathoz szükséges eszközök és szoftverek

Hardver követelmények:

- Asztali számítógép / laptop / okostelefon / tablet
- Internetkapcsolat (Wi-Fi vagy mobilinternet)

Szoftver követelmények:

- Frissített web böngésző (ajánlott: Google Chrome, Mozilla Firefox, Microsoft Edge, Safari)
- Internetkapcsolat szükséges a webshop használatához
- Regisztrációhoz és rendeléshez érvényes e-mail cím szükséges

5.3. A webshop elérése

A Noel Candy Shop webshop böngészőből érhető el a következő címen:

Webshop link: www.noelcandyshop.hu (kitalált domain, nincs domain címe a webshopnak!)

5.4. Regisztráció és bejelentkezés

Regisztráció menete:

1. Nyisd meg a webshopot a böngészőben
2. Kattints a "Fiókom" gombra a navigációs sávban
3. Válaszd ki a "Regisztráció" gombot és nyomj rá
4. Add meg a következő adatokat:
    a. E-mail cím
    b. Jelszó

   c. Jelszó megerősítése
5. Kattints a "Regisztrálok" gombra
6. Az oldal kiírja, hogy "A regisztráció sikeres!"
7. A rendszer e-mailben visszaigazolást küld
8. Az e-mailben található linkre kattintva aktiváld a fiókodat

Bejelentkezés menete:

1. Kattints a "Fiókom" gombra a navigációs sávban
2. Kattints a "Bejelentkezés" gombra
3. Írd be az e-mail címedet és jelszavadat
4. Kattints a "Bejelentkezés" gombra
5. Az oldal kiírja az e-mail címed, ha sikeres a bejelentkezés
6. Továbbá felajánlja, hogy "Profil szerkesztése" vagy "Kijelentkezés"

5.5. Termékek böngészése és keresés

A főoldalon a brandek listája jelenik meg. Egy-egy brand nevére kattintva megnézheted az adott brand termékeit a webshop-ban és kedved szerint válogathatsz belőlük megkönnyítve ezzel a keresést.

A navigációs sávban a termékek nevű menü alatt kategóriák szerint megtekintheted milyen termékeket árul a webshop (pl. csípős chipsek, sós chipsek, savanyú üdítők, édes üdítők, savanyú gumicukrok, édes gumicukrok).

Minden egyes oldalon, ahol termékek vannak könnyíti a keresést a jobb felső sarokban lévő "Legolcsóbb/Legdrágább" nevű gomb ami ár szerint sorba rendezi a termékeket.

Ha van egy kimondott termék, amit keresel akkor a jobb felső sarokban a kosár melletti keresősávba beírhatod a kívánt termék nevét és a weboldal odanavigál a keresett termékre.

Végül pedig a navigációs sávban lévő "Kívánságlistám" menüpontban megadhatsz olyan termékeket, az internetről kikeresett termék neve és URL-je (képcím) segítségével, amiket szívesen látnál a webshop-ban.

5.6. Vásárlás és rendelés leadása

Kosár használata:

1. Válassz ki egy terméket, majd nyomd meg a "Kosárba rakom" gombot
2. A termék bekerül a kosárba, amit az oldal is jelez "A termék kosárba rakva!" címszóval
3. Ha végeztél a vásárlással és szeretnél fizetni akkor a jobb felső sarokban találod a kosarat

4. Kattints rá és megmutatja milyen termékeket raktál a kosaradba, illetve mi lenne a végösszege a vásárlásodnak
5. Ha meggondoltad magad és valamit mégse szeretnél megvenni akkor el tudod távolítani a kosaradból az "Eltávolítás" gombbal
6. Ha végeztél, nyomj a jobb felső sarokban lévő "Pénztárhoz" gombra, ami a "Rendelés leadása" oldalra navigál

Rendelés leadása:

1. A "Rendelés leadása" oldalon add meg a szállítási adataid (vezetéknév, keresztnév, irányítószám, város, utca, házszám, e-mail cím, telefonszám)
2. Kattints a "Rendelés leadása" gombra
3. Az oldal kiírja, hogy "Rendelés sikeresen leadva!", illetve e-mailben visszaigazolást kapsz a rendelésed részleteiről

5.7. Rendelések kezelése

Minden sikeres rendelés után a megadott e-mail címre visszaigazoló üzenetet küldünk.

Az e-mail tartalmazza a rendelés részleteit:

- Megrendelt termékek listája
- Összeg
- Szállítási adatok
- Rendelési azonosító

A rendelés állapotáról további értesítéseket is kaphatsz e-mailben (pl. csomag feladása).

Ha a rendelés után módosítani szeretnél valamit, ügyfélszolgálaton keresztül léphetsz kapcsolatba velünk.

5.8. Ügyfélszolgálat és kapcsolat

Ha bármilyen kérdésed van a "Kapcsolat" fülre kattintva érhetsz el minket:

Telefon: +36 30 123 4567

E-mail: ugyfelszolgalat.edessegvilag@gmail.com

Instagram: https://www.instagram.com/edesseg.vilag/

Tiktok: https://www.tiktok.com/@edesseg.vilag

Cím: 2000 Szentendre, Kucsera Ferenc utca 15.