Továbbá megfigyelve több modernebb kialakítású weboldalt, azt tapasztaltuk, hogy az a dizájn melyet a mobileszközökön való adatmegjelenítésnél használnak teljes képernyős nézetben is egyre nagyobb teret hódít, ez a megközelítés tetszett nekünk így hasonló dizájn fejlesztésébe kezdtünk.

Ami a menüpontokat illeti az eredeti elképzelés szerint a weboldal dizájn teljes mértékben dinamikus lett volna és minden adatot adatbázisból kértünk volna le. De végül ebből az elképzelésből engedtünk az adatforgalom csökkentése érdekében.

Footer komponens:

A footer manapság egy standard komponensnek számít minden weboldalon, a megléte gyakorlatilag már-már elvárt. Ami a funkcióját illeti elsősorban két célunk volt a létrehozásával. Először is bizonyos információk megjelenítése, többek között a felhasznált technológiák logóinak feltűntetése, copyright jogok feltűntetése és a weboldal logója.

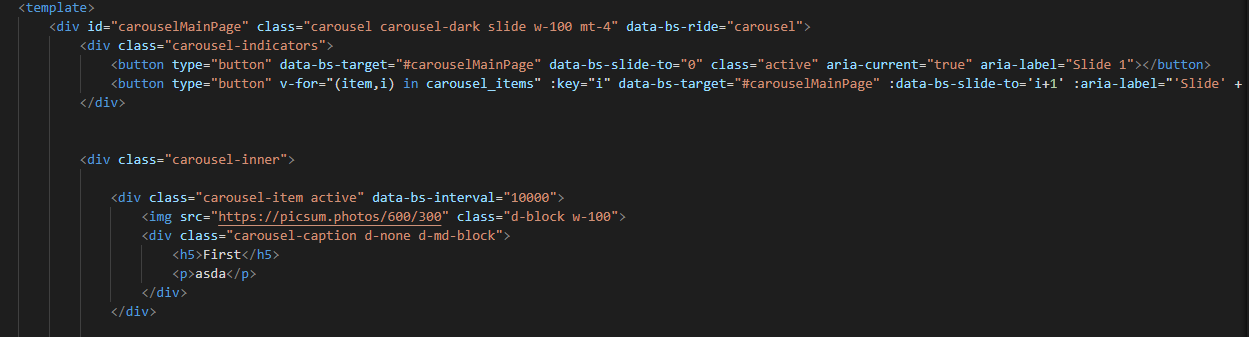
Másodszor pedig némely navigációs link ismételt megjelenítése. Erre azért van szükség, mert a látogatónak a figyelmét nagyobb valószínűséggel tudjuk megragadni, hogyha egyes menüpontok a weboldal tetején és a footer segítségével a weboldal alján is megjelennek, hiszen így a további szolgáltatásainkra például a fő oldal áttekintése után ismét felhívjuk a figyelmet.

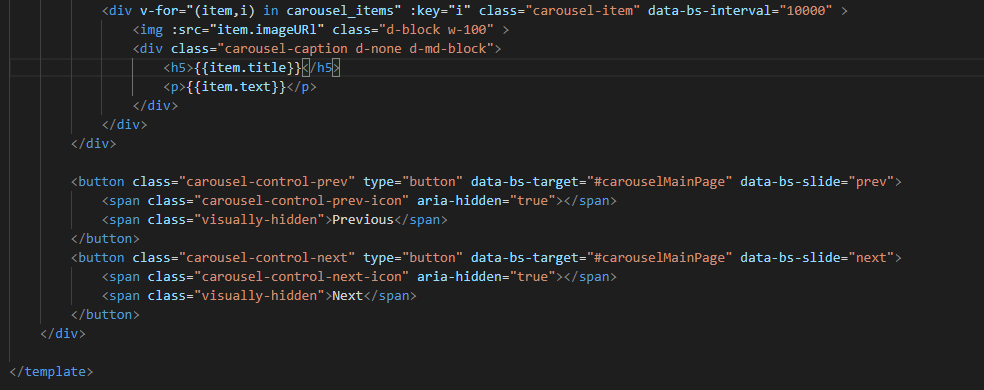
Home page:

A fejlesztés során több elképzelés is született a fő oldal megvalósításával kapcsolatban. Az alapelv az volt, hogy a fő oldalnak figyelemfelkeltőnek kell lennie, de nem is szabad túl sok információval elárasztania a felhasználót. Az imént említett navigációs panel kialakításának módját figyelembe véve és azt, hogy alapértelmezetten a navigation komponens nem jelenik meg, így, több adatot tudunk megjeleníteni úgy, hogy az még ne legyen zavaró.

A fő oldal elsődleges célja a látogató figyelmének a felkeltése és a további böngészésre való ösztönzés, ezért a fontosabb szolgáltatásaink linkjei a főoldalon akár háromszor is megjelenhetnek különböző formában.

További szempont volt a megfelelő vizuális megjelenítés is. Tehát az információ ízléses tálalása. Az egyik ilyen módszer volt a carousel komponens felhasználása, mely periodikusan egy képlistán megy végig magyarázó szöveg kíséretében. Ez a fajta megjelenítés napjainkban nagyon elterjedt és jól mutat mind típusú kijelzőn, hiszen a komponens reszponzívitás könnyen megoldható. A weboldalunkon a bootstrap carousel külső komponenst használtuk fel.





A forráskód átírásával a komponens dinamikusan a megadott adatok alapján generálódik le.

About us:

Az about us oldal egy egyszerű oldal ahol magunkról, a projekt céljáról és olyan dolgokról melyek, a látogatónak szükségesek lehetnek, osztunk meg alap információkat. Talán ez a legegyszerűbb oldal a projektben ugyanis, egyszerű kiíratásról van itt szó listákba a pragrafusokba rendezve.

Products (user):

Az oldal elsődleges célja hogy az elérhető termékek közül a látogató tudjon böngészni és egyéb műveleteket végre hajtani. A termékek úgynevezett cardok segítségével vannak feltűntetve. Mind a bootstrapnek és a primevue-nak vannak card moduljai, de ezek nemteljesen feleltek meg az elvárásainknak.

A cardokkal kapcsolatban a következő elvárásaink voltak:

1. A cardnak tartalmazni kell egy lekicsinyített képet a termékről.
2. A termék nevét és árát felkell tudnunk rajta tűntetni.
3. Ha a felhasználó vagy egy gombra kattintva vagy mouseover esemény következtében a cardnak a méretét megkell növelnünk annyira, hogy az utóbbi információk mellett még egy részletesebb leírást is megtudjunk jeleníteni.

A products oldalon a felhasználó több műveletet és képes végrehajtani, névlegesen a terméket kosárba tudja helyezni, a választott termékről bővebb információhoz juthat, ehez a művelethez a product-details oldalra való átlépés szükséges, és a lehetőség nyílik a termék kívánságlistára, való felvételére is.

Ezekhez a műveletekhez backend api-hívásokra lesz szükség.

A termék kosárba való felvételéhez, a felhasználónak nemszükséges regisztrálnia és bejelentkeznie, hiszen a kosár tartalmát nem adatbázisban tároljuk, azonban a kívánságlistára való felvételhez alap követelmény hogy a felhasználó be legyen jelentkezve, az api hívás csak így lehetséges, a hívás előtt egy session api hívásával, illetve a visszakapott boolean változó értékével validáció történik. Igaz, már az oldal renderelése előtt lefut egy hasonló validáció, hogy olyan gombokat, menüket és információkat ne osszunk meg melyekre a felhasználónak nincs jogosultsága. Ez azért fontos, mert így eltudjuk kerülni, azt hogy összezavarjuk a felhasználót.

Signup és Login:

A signup oldal teszi lehetővé, hogy a felhasználó regisztrálni tudja magát az oldalra, és hogy aztán élvezhesse annak előnyeit.

A regisztrációs folyamathoz egy form kitöltése szükséges, melyen több hiba lehetőséget is figyelembe kellet vennünk.

1. A megadott adatokat validálni kell, hogy véletlenül se lehessen hibás adatokat megadni.
2. A felhasználónak további segítséget kell nyújtani az által hogy:
   1. Segítünk megfelelő erősségű jelszót választani a Primevue: password komponens segítségével.
   2. Redukáljuk az elírás lehetőségét azáltal, hogy validációt készítünk a jelszó kétszeres lekérése után.
   3. A form kitültése után validáljuk a megadott adatokat.

Továbbá az kitöltött mezőket adatkötésekkel tároljuk.

A validációk sikeressége után, egy api hívás segítségével folytatódik a regisztrációs folyamat. Egy felugró ablak jelzi, hogy a link mellyel az Account aktiválható a megadott emailre lesz elküldve.

Az emailben elküldött link által való fiók aktiválását követően, a fiók adatbázisban való tárolása megtörténik. Ez úgy valósul meg, hogy az emailben elküldött link, betöltésekor a frontend az URL-ből kinyeri az aktiváláshoz szükséges tokent, melyet felhasználva egy API hívást követően az adatbázisban az account aktiválásra kerül.

Ha ez sikerült egy új felugró ablak tájékoztatja a felhasználót a művelet sikerességéről vagy az esetleges hibákról, majd átirányítja a főoldalra.

A login oldalon hasonló képpen, a validációs műveletek után egy login api hívást követően a frontend átnavigálja a felhasználót a főoldalra, ami egy ismételt session api hívással megállapítja, hogy a kliensoldalon való session valid, így a bejelentkezett userek számára elérhető tartalmak renderelésre kerülnek.

Ilyen tartalmak többek között, a navbar tartalma, és hozzáférés egyes pagek-hez.

Wishlist:

A wishlist a weboldal azon eszközei közé tartozik, melyeknek előnyeit csak regisztrált felhasználók élvezhetik. Ugyanis ez egy adatbázisban tárolt lista készítésének a lehetőségével ruházza fel a felhasználót. Így az ezen a listán feltűntetett termékek árában történő változásokat, potenciális leértékeléseket sokkal könnyebben nyomon lehet követni.

A lista szabadon szerkeszthető, a tulajdonosa szabad hozzáférést kap minden ezzel kapcsolatos tevékenységre.

Orders és a cart:

A felhasználó a vásárlás során az „add to cart” gombra kattintva tudja a kiválasztott termékeket a kosárba helyezni. Az hogy egy termék a kosárban van, önmagában nemjelent semmit, a rendelés csak akkor lesz véglegesítve, ha a felhasználó tovább folytatja a vásárlást a megfelelő gombra kattintva. Ha ezt megteszi, akkor a megfelelő adatok rögzítése után (regisztrált felhasználóknak ez a lépés kihagyható, elmentett fizetési adatok esetén) a cart azaz kosár tartalma átkerül az orders azaz rendelések listába. A megrendelések listában lévő termékek már elvannak különítve a vásárlónak és a termékek már nincsenek számon tartva, mint elérhető termékek, úgyis mondhatjuk, hogy ami a többi látogatót illeti, számukra a rendelések listában lévő termék már eladott így raktáron nem elérhető termékként van kezelve.

Az orders listából a megrendelt terméket el lehet távolítani, így azok ismét nem eladott, eladásra szánt terméknek tekinthetők. A sikeres kiszállítás és kifizetés után, az orders lista tartalma törlődik.

Admin products:

Az admin products célja különbözik a mindenki számára publikus products oldaltól. Ehhez az oldalhoz a hozzáférés csakis validált adminok számára lehetséges.

Az ezen az oldalon használt szerkesztett és teljes mértékben testre szabott primevue modul segítségével az admin hatékonyon tud API hívásokkal hozzáadni, szerkeszteni és törölni termékeket, tehát úgy is mondhatjuk, hogy egy kész CRUD product interface áll a rendelkezésükre a könnyű és hatékony adatbázis kezelés érdekében, mindezt egy oldalon.

A CRUD műveletekkel való munkának minden változatát további felugró panelek könnyítik meg, lehetővé téve programozási előismerettel nem rendelkező kollégák számára is az admin munkakör gyors, kielégítő szintű ellátását.

A felugró ablakokkal, üzenetekkel és interfacekkel az admin folyamatos visszajelzést kap a munkája állapotáról illetve annak sikerességéről. Illetve a kritikus műveletek, mint például a törlés előtt felugró megerősítő ablakok minimalizálják a véletlenül elkövetett törlések és változtatások előfordulását.

Fontos megemlíteni, hogy az admin products interface használata elsősorban számítógépről való munkavégzéshez lett tervezve. Természetesen ez nem azt jelenti, hogy itt a reszponszivitást mint szempontot mellőztük, de egy ekkora táblázatot ésszerűtlen lenne, mobilnézetben megjeleníteni úgy, hogy mellőznénk az X irányú csúszka használatát. Természetesen a csúszkától eltekintve, mint minden más az oldalon ez az oldal is a reszponszivitás alapelvei szerint lett tervezve.

Ez az oldal, mint minden más admin oldal nem elérhető csak adminok számára. Annak érdekében, hogy ez a kritérium megvalósulhasson. Az oldal betöltésekor minden alkalommal a háttérben lefut egy autentikációs algoritmus, ami két dolgot ellenőriz: elsősorban hogy a felhasználó be van e jelentkezve, másodszor pedig hogy a felhasználó rendelkezik e admin jogosultsággal. Ha ezek közül a kritériumok közül bármelyik nem teljesül, akkor az oldal egy kényszerített átirányítással visszalép a főoldalra.

Change password:

A jelszó változtatás egy két lépcsős művelet, elsőnek például a login oldalon a jelszó változtatást kérvényezni kell. Ez a megfelelő gombra való kattintással lehetséges, ez után egy felugró ablakban a felhasználónak megkell adni az emailcímét. Fontos hogy azt az email címet adja, meg amivel már előzőleg regisztrált az oldalra.

Hibás adat esetén vagy bármely egyéb probléma esetén az oldal ennek megfelelően informálja a felhasználót. Ha egy érvényes email cím megadása megtörtént, akkor a backend, a felhasználót emailben fogja értesíteni a művelet sikerességéről és egy token és link emailben történő elküldésével a jelszó változtatás következő lépése kezdetét veheti.

A linkre kattintva, a backend ellenőrzi a linkben lévő tokent és a token hitelességének megfelelően, lehetőség nyílik egy új jelszó megadásához. A megfelelő gomb lenyomásával az API amely a jelszóváltoztatást elvégzi meghívásra kerül, és miután a változtatás megtörtént, az oldal átnavigálja a usert-t a login oldalra.