

Programrendszerek fejlesztése munkanapló

Szolnoki Norbert Sándor

JFG74W

2022/tavaszi

Előszó

Első projektem ezekkel a webes technológiákkal, így az elején elég döcögősen indult a projekt. A végére nagyon megtetszett az a dinamika, amit ez a stack képvisel. Visszanézve sok a „kókány” megoldásom, ami pusztán a rutin hiányával tudok magyarázni. Mivel a beadandó project egy részét lefedi a gyakorlati videóanyag, így a videók haladásával, és lokális reprodukciójával együtt haladtam a beadandóval. A webshop témájául az Escape From Tarkov játékot vettem, mert nagyon szeretem, és volt lehetőségem a játékban található termékek adatának az exportálására.

2022.04.18

Ekkorra elkészült a first/second mintaoldal és a login/logout, de mivel a logout nem úgy működött nálam, ahogy az a videóban, már gitről másolt kóddal sem, így azt feladtam, és haladtam tovább.

2022.04.22

Elkészítettem az items modellt, és a hozzá tartozó backend get/post/put/delete funkciókat, ezeket postmannel teszteltem, működött, ahogy kell. A put hívást annyiban módosítottam, hogy nem kötelező minden adatot elküldeni, amikor változtatunk, elég csak az ID-t és a módosítani kívánt adatot. Küzdöttem azzal, hogy az elemeket milyen fancy módon jelenítsem meg, aztán eldobtam minden plusz libet, mert sok helyet ettek volna herokun, így az ngfor-al csak végigmegyek a listán, és egy pici animációval jelenítem meg az elemeket. Mivel a betöltött termékek ID-ja is odakerül a listába, így a rajta levő gomb megnyomására az ID-t paraméterül továbbbadjuk az URL-ben, így le tudjuk kérni az adott terméket az adatbázisból. Erre készítettem egy kérést, amely egy ID-t kér válaszul, és ha van ilyen ID-jú termék, akkor azt adja válaszul. Frontend-el nem sokat dolgozom, így ez a UI kialakítása igénybe vett 3-4 órát.

2022.04.23

Sessionben módosítottam az eltárolt adatot az órán látottakhoz képest, most az ID-t tárolja, amit a mongodb ad neki, így viszonylag nehezebb feltörni, persze brute-force-al bármi lehetséges, csak sokáig tart. Kibővítettem a user get metódusát, ha kap „uid” paramétert, akkor visszaadja az adott usert, erre azért volt szükség, hogy az app.component html felületén tudjam változtatni a megjelenő gombokat. Ha nincs bejelentkezve, akkor egyik sem látszik, ha be van jelentkezve, akkor a jogosultságtól függ, hogy mely gombok látszódnak. Kiegészítettem a regisztrációt a backenden, ha admin névvel regisztrál bárki, akkor 'admin'-t ad neki accesslevel-nek, minden egyéb esetben 'basic' lesz. Admin felhasználót nem engedi törölni a rendszer, 'Hiányos adatot ad válaszul. A user 'put' hívását is módosítottam, mert az órán elhangzottakkal meg tudjuk változtatni más adatait, ha kitaláljuk a usernevét. Közben a toolbart átrendeztem, és adtam neki egy másik style-t amit a w3 schools-ról szedtem, és link formájában hivatkoztam be a html fájlba (tudom, lehetett volna szebben is, de a UI tényleg nem az erősségem). Tettem egy csúszkát, amivel a vásárolni kívánt mennyiséget lehet állítani, és ez automatikusan számolja, hogy mennyibe kerül a kiválasztott mennyiségű termékek összege. Ezután elkezdtem a bevásárlókosárral dolgozni, amelyet a localStorage-ben tárolok. Tettem egy funkciót az item oldalba, ami ebbe a localStorage-be pakol. Ezután elkészítettem a 'My cart' oldalt, ami a localStorage-ből veszi ki a rendeléseket. Kikeresi őket az adatbázisból az itemService által adott getItem hívásnak, ez egy ID-t kér, és az adott termékkel tér vissza. Készítettem egy interface-t az itemeknek, és a getItemek minden eleméből csináltam egy item-et és azt az items tömbhöz adtam. Maga a cart html oldala gy ngfor-al végigmegy az items listán és egy

táblába tölti. A shopping cartban a névből linket akartam csinálni, amelyel visszamehetünk a termék oldalára, de a routerlink %2f entitássá konvertálja a stringben lévő előre mutató perjelet, így keresnem kellett megoldást, mely ez lett: „[routerLink]=[\", 'item', item.id]\"”. Készítettem egy '/order' végpontot a szerverre, ahol készíteni, listázni és törölni lehet rendeléseket, a készítés be is lett kötve a cart oldal 'ORDER' gombjára. Akartam egy visszajelző ablakot, hogy amikor a user berak egy elemet a kosárba, akkor lássa is, hogy megtörtént. Ez okozott egy nagyjából 2 órás debug kört, mert a felrakott „ng-angular-popup” duplikált egy css sort, amitől nem indult el az 'ng serve' parancsal. Sokadik oldalon megtaláltam, hogy végül a CSS duplikálódott, ezért megkerestem és töröltem. Majd ezután nem engedte megint indítani, mert nem a '—save' kapcsolóval szedtem le a pop-upot, erre mondhatni véletlen találtam rá, hogy miért lehet, de végül sikerült. A mai napon az alábbi funkciók érhetőek el:

- Belépés
- Kilépés
- Regisztráció
- Termékek listázása
- Termék kosárba tétele
- Kosár teljes kiürítése
- Egy elem kiürítése a kosárból
- Rendelés véglegesítése

2022.04.25

Egy darabig fennakadtam egy saját magam által generált problémán, ez pedig nem volt más mint a req-ben található user objektumból kiolvasható accessLevel és maga az autentikáció. Ha egy kérés a szerverfelé nem tartalmazta a withcredentials: true-t, akkor nem tudtam kiolvasni a jogosultságot, így nem tudtam levédeni a végpontokat. Miután kiegészítettem a hívásokat, könnyedén egy kis copy/paste-el már védve is voltak a végpontok, és elkülönítettem a login után elérhető hívásokat az admin jogosultsággal rendelkezőtől. Elkezdtem az admin felület módosításait, ez két gomb a tollbarra, ami pluszban van az adminnak. A felhasználók kezelése, és a termékek kezelése. A felhasználók kezelése nem más mint egy lista a felhasználókból, ami tartalmaz egy törlés és egy módosítás gombot. A törlés azonnal törli a szerverről, a módosítás megnyitja azt az oldalt, amit a My account gomb megnyomására hoz elő, DE a user ID-ját teszi oda az url-be, így az admin megnézheti a user adatait, és módosíthatja is azokat. Folytattam a termékek módosításával, itt jelenleg is hezitálok, mert 1990 elem van az adatbázisban, és ennek a betöltése nem oly gyors. Maga az termékek szerkesztése egy lista az termékekből szintén két gomb-al, egy törlés és módosítás. Tartalmaz egy 'Reset Database' gombot, amellyel az adatbázist a kiindulási pontba lehet tenni, ez töröl minden terméket, és a szerverre feltöltött klienskódban található json file-ból felpopulálja. A képeket nem törölöm, mert azt úgy raktam a kód mellé, és tesztelés során macerás lett volna mindig feltölteni 1990 képet. Az adatbázis visszaállítás kicsit sokáig tart, akartam berakni egy spinnert, de valamiért nem sikerült, hogy reszponzív legyen a user felé, ne gondolja, hogy lefagyott az oldal. Admin oldalak megnyitását az adminguard-al terveztem védeni, de mivel adatbázisból kérdezem a jogosultságot, így írtam egy funkciót, ami visszadob az items oldalra, mert sajnos nem lehet subscribe-olni a guard-ban. Vagyis lehet, de nem várja meg és a default beállított értéket adja vissza. Néhány tulajdonságát lehet módosítani a terméknek. A mai napon az alábbi funkciók érhetőek el:

- Belépés
- Kilépés
- Regisztráció
- Termékek listázása

- Termék kosárba tétele
- Kosár teljes kiürítése
- Egy elem kiürítése a kosárból
- Rendelés véglegesítése
- Felhasználókezelés
- Termék kezelés (törlés/módosítás) végponton létezik a termék hozzáadása
- Végpont védelem
- Admin oldal védelem

2022-04-26

Ezen a napon célul tűzttem ki, hogy heroku-ra is kirakom a szerveret és a frontend-et, itt kellett keresgéljek egy kicsit, hogy hogyan lehet ezt megtenni és miket kell módosítsak annak érdekében, hogy ott is fusson minden. A deploy sikeresen megtörtént, de az autentikáció ismét kifogott rajtam kicsit, mert az órán elhangzott anyaggal mi a memóriában tároltuk a session-t, amit herokunál nem lehet megcsinálni. Áttértem a session mongodb-ben való tárolására. Ehhez szükségem volt a mongo-connect csomagra. Sajnos ezen csomag használata sem oldotta meg a problémám, így a herokus működést el kell vessem idő szűkében.

Képek a használatról:

Regisztráció:

Register

Username József

Email j@gmail.com

Password *****

REGISTER

Login

Username

Password

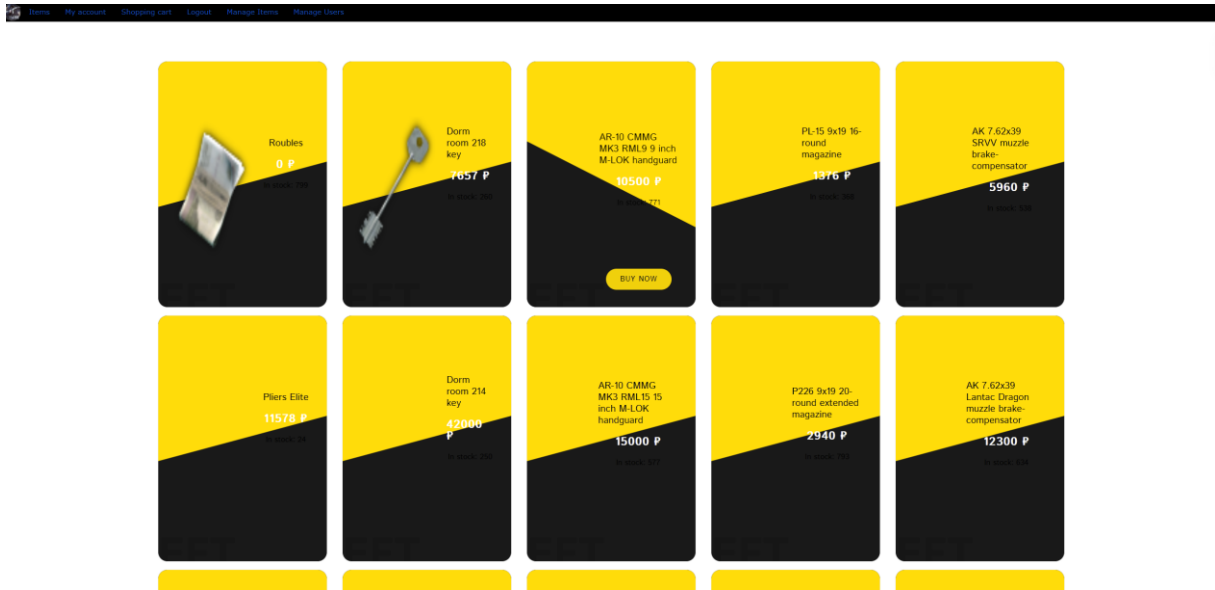
LOGIN

✓ Registration completed.
Your user has been created, now you can
log in.

Regisztráció gomb megnyomásával elkészül a user, és egy felugró ablak tájékoztat erről. H hibával tér vissza a kérés, akkor a felugró ablak tartalma az error tartalma.

Belépés:

Sikerese belépést követően az /items oldal jön be, ahol a termékek vannak listázva:



Hibás jelszó esetén, a felugró ablak tájékoztat:

Register

Username

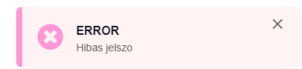
Email

Password

Login

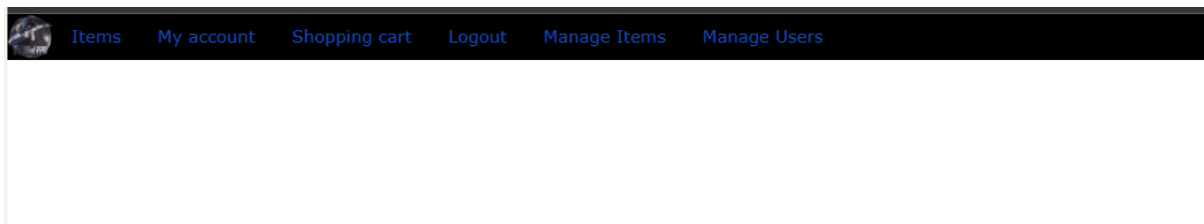
Username

Password

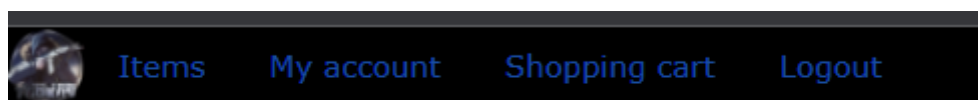


Layoutok:

Admin layout:

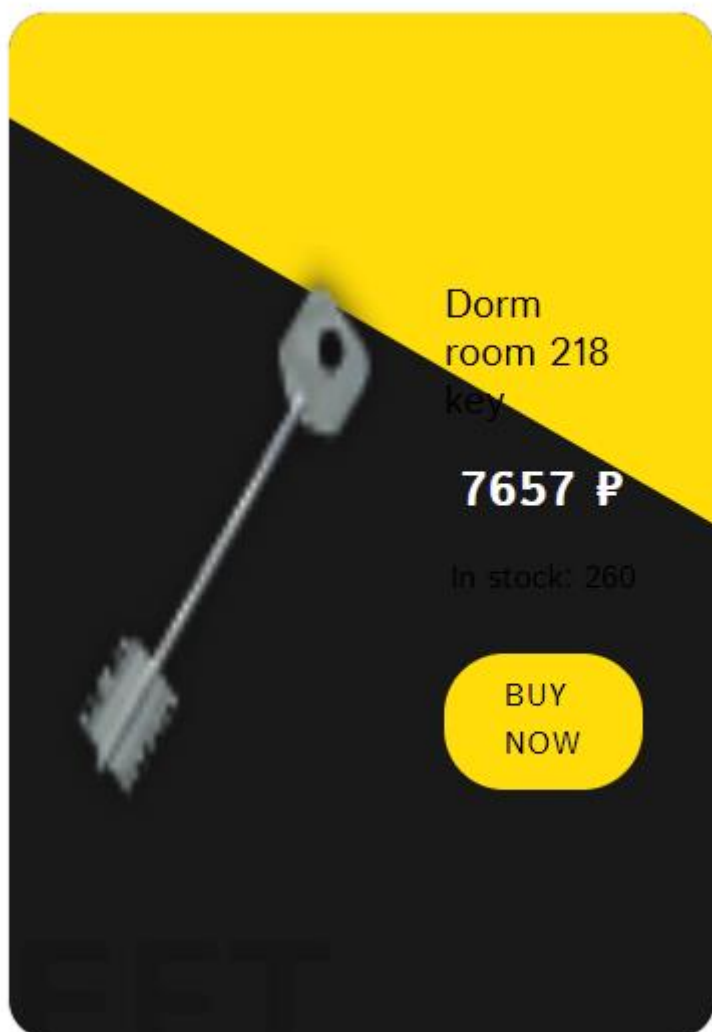


User layout:



Vásárlás menete:

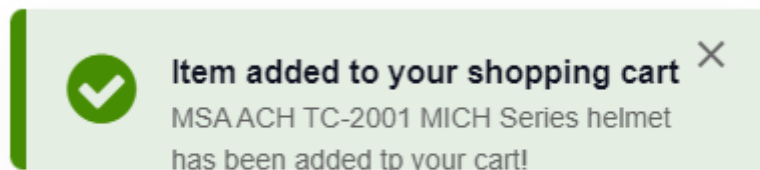
Egeret rá kell vinni a termékre, akkor megjelenik a Buy Now gomb:





Buy now gombra kattintva megjelenik a termék oldala:



A csúszkával állítani tudjuk a vásárolni kívánt mennyiséget, majd az 'Add to cart' gomb megnyomásával a kosarunkba tesszük, és erről egy felugró ablak tájékoztat.



Kosár:

My shopping cart					
	Image	Name	Count	Price per unit	Total price
REMOVE		Dorm room 218 key	91	7657 P	696787 P
REMOVE		MSA ACH TC-2001 MICH Series helmet	139	46200 P	6421800 P
TOTAL			230		7118587 P
			ORDER	CLEAR	

Törölni tudjuk a terméket a kosárból, illetve véglegesíteni a rendelésünket az ORDER gomb-al, az ezután átvizsgál minket a My account oldalunkra, valamint teljesen kiüríteni a kosarat a CLEAR gombal. Az order gomb megnyomásával mentjük az adatbázisba az ordert.

My account:

Edit user

Username	<input type="text" value="prf"/>
Email	<input type="text" value="prf@prf.com"/>
Password	<input type="password" value="....."/>
<input type="button" value="UPDATE"/>	

My orders

Date	Price
24.04.2022	298623 ₺
24.04.2022	260338 ₺
24.04.2022	635531 ₺
24.04.2022	122512 ₺
24.04.2022	803985 ₺
24.04.2022	796328 ₺
24.04.2022	819299 ₺
24.04.2022	819299 ₺
24.04.2022	375193 ₺
24.04.2022	673816 ₺
24.04.2022	803985 ₺
24.04.2022	773357 ₺
24.04.2022	796328 ₺
24.04.2022	1155000 ₺
25.04.2022	1990820 ₺
26.04.2022	7118587 ₺

Módosítani tudjuk a saját adatainkat, illetve láthatjuk, hogy mikor mennyiért rendeltünk. Az orderek objektumként jönnek vissza a szervertől, így akár meg lehetne a rendelések tartalmát is mutatni.

Manage Users:

Manage users

		Username	Email
DELETE	EDIT	prf99910	prf@prf.prf
DELETE	EDIT	prf3444444444444norb.szolnoki@gmail.com	norb.szolnoki@gmail.com
DELETE	EDIT	prf22	prf@prf.prf
DELETE	EDIT	prf222333	prf@prf.prf
DELETE	EDIT	prf2344	norb.szolnoki@gmail.com
DELETE	EDIT	prf2333	norb.szolnoki@gmail.com
DELETE	EDIT	prf33	norb.szolnoki@gmail.com
DELETE	EDIT	prf	prf@prf.com
DELETE	EDIT	prf23333	prf@prf.com
DELETE	EDIT	admin	admin@admin.com
DELETE	EDIT	asd	admin
DELETE	EDIT	józsef	J@gmail.com

A regisztrált felhasználókat tudjuk törölni, illetve módosítani. a Módosítás gombra kattintva behozza a My account oldalt, de az url-be annak a usernek az ID-ját teszi, akire kattintottunk, ezzel újrafelhasználva a my account oldalt.

Manage items:

Manage items

[Reset items to default](#)

Manage	Image	Name
DELETE EDIT		Roubles
DELETE EDIT		Pliers Elite
DELETE EDIT		WD-40 (100ml)
DELETE EDIT		Electric motor
DELETE EDIT		Car battery
DELETE EDIT		CPU Fan
DELETE EDIT		Corrugated hose
DELETE EDIT		Pack of screws
DELETE EDIT		Electric drill

Itt törölni, illetve módosítani tudjuk a termékeket, valamint visszaállni az adatbázis eredeti állapotára a termékek tekintetében.

Edit item:

Edit item

Name	<input type="text" value="Car battery"/>
Description	<div>Typical 12 V, 51 Ah lead-acid car batteries are in a big demand with traders, who resell them in the areas that have been left without power supply.</div>
CreditsPrice	<input type="text" value="1600000"/>
<input type="button" value="UPDATE"/>	