

**TANTÁRGYAK****TANANYAGOK****PROJEKTEK**

# **SZERVEROLDALI WEBPROGRAMOZÁS BEADANDÓ**

## **BEADANDÓ KÉT FELVONÁSBAN**

A feladatod az alábbi beadandó feladat megvalósítása. Az elkészült beadandókat a Canvas rendszeren keresztül kell majd beadni.

## **MODELLEK**

- Az alábbi modelleket kell létrehozni a megadott mezőkkel és tulajdonságokkal.
  - Category
    - id
    - name (string)
    - timestamps
  - Item
    - name (string)
    - description (text)
    - price (double)
    - image\_url (string, nullable)
    - timestamps
    - soft delete beállítása (ráér a 2. felvonásban)
  - Order
    - id
    - user\_id
    - address (string, nullable)
    - comment (string, nullable)
    - payment\_method (enum: CASH, CARD) (default: CASH)
    - status (enum: CART, RECEIVED, REJECTED, ACCEPTED)
    - received\_on (datetime, nullable)
    - processed\_on (datetime, nullable)
    - timestamps
  - OrderedItem
    - id
    - item\_id
    - order\_id
    - quantity (integer)
    - timestamps
  - User
    - id
    - ...
    - is\_admin (boolean)
    - ...

## **RELÁCIÓK**

- A modellek közötti relációk a következőképpen alakulnak:
  - Category N - N Item
  - User 1 - N Order
  - Order 1 - N OrderedItem
  - Item 1 - N OrderedItem

## SZEREPKÖRÖK

- A szerepkörök a következők:
  - Vendég
  - Felhasználó
  - Admin

## 1. FELVONÁS

**Határidő: 2020. 11. 15. 23:59**

- Vendég:
  - Regisztráció (**/register**):
    - A vendég tudjon regisztrálni az alkalmazásba. A regisztráció eredményeként egyszerű felhasználó jöjjön létre, az admint "rendszergazdai" eszközökkel lehessen csak hozzáadni, tehát pl. ha futtatjuk a seeder-t, akkor létrejön alpból egy admin fiók.
  - Bejelentkezés (**/login**):
    - A vendégnek legyen lehetősége bejelentkezni. Az elfelejtett jelszóra vonatkozó hivatkozás legyen eltávolítva a login oldalról!
  - Névjegy (**/about**):
    - Írja ki az alkalmazás nevét, alatta pedig a készítő adatait (név, neptun kód, email cím)
  - Főoldal (**/**):
    - Írja ki a következő adatok számát: felhasználók, kategóriák, item-ek
  - Menü (**/menu**):
    - Listázza ki valamilyen módon az item-eket (pl. card-ként)!
    - Minden item-hez tartozzon egy form, ami a következőképpen épül fel:
      - Egy számbeviteli mező, aminek a minimális értéke 1, az alapértelmezett értéke 1, és a maximális értéke 10.
      - Egy "Kosárba" nevű gomb.
      - Vendégfelhasználó esetén ennek a formnak minden eleme legyen disabled, és a hozzá tartozó route-ok is legyenek hitelesítéshez kötve!
- Felhasználó:
  - Menü (**/menu**):
    - Felül listázd ki valamilyen módon a kategóriákat (lehet badge, select, stb)
      - Ha a user kiválaszt egy kategóriát, akkor csak ahhoz a kategóriához adja be az item-eket (**GET: /menu/category/{id}**)
      - Ha egy adott kategória van megnyitva, akkor is ugyanúgy jelenjenek meg felül a kategóriák, alatta pedig írja, hogy melyik kategóriához vannak éppen listázva az item-ek
    - Listázza ki valamilyen módon az item-eket (pl. card-ként)!
    - Az item-ek kategóriák szerint legyenek listázva, pl. kiírod a kategória nevét, és alatta listázod az oda tartozó item-eket.
    - Az item-hez tartozzon egy form, ami a következőképpen épül fel:
      - Egy számbeviteli mező, aminek a minimális értéke 1, az alapértelmezett értéke 1, és a maximális értéke 10.
      - Egy "Kosárba" nevű gomb, amire rákattintva adja hozzá az item-et a megadott mennyiségben és irányítson is át a kosárra, ami már úgy jelenjen meg, hogy az item hozzá van adva! (**POST: /cart/add, az item id és a mennyiség request body-ban menjen át**)
        - Teljesen opcionális, de lehet akár úgy is csinálni, hogy az item-nek legyen külön oldala, és onnan lehessen kosárba rakni, ne a card-ok közül (**GET: /menu/item/{id}**)
        - Ha valamilyen validációs hiba történik, akkor az a Kosár oldalon jelenjen meg
        - Segítség, ötlet:

- Ha a kiválasztott item már bent van a kosárban, akkor csak a mennyiség növekedjen.
- Az Order-nek van CART állapota, és az Order egyértelműen hozzáköthető egy felhasználóhoz.
- **Kosár (/cart):**
  - Listázza ki a kosárban levő item-eket, hozzájuk a mennyiségüket és az árukat
    - Ha még nincs item a kosárban, akkor jelenjen meg erről egy üzenet vagy valamilyen értelmes visszajelzés a felhasználónak
  - Legyen az itemek mellett egy törlés gomb, arra kattintva pedig törölje a kosárból az adott item-et (**POST: /cart/remove/{itemId}**)
  - Alul írja ki a teljes árat, tehát adja össze az itemek árát.
  - Legyen egy cím mező, az egyszerűség kedvéért ez csak egy szövegbeviteli mező, ahol megadja a user a szállítási címét. Ugyan a cím nullable, viszont a form-nál elvárjuk, hogy meg legyen adva, és csak így jöhet létre rendelés.
  - Legyen egy komment textarea, ahol megjegyzést lehet fűzni a rendeléshez, ez nem kötelezően kitöltendő mező.
  - Valamilyen módon ki lehessen választani a fizetés módját (készpénz vagy bankkártya).
  - A legvégén pedig legyen egy "Megrendelem", "Rendelés leadása", stb. nevű gomb (**POST: /cart/send**)
    - A kosár a beküldéskor legyen validálva:
      - A szállítási cím és a fizetés módjának megadása legyen kötelező
      - Üres kosarat ne lehessen elküldeni
    - A rendelés állapota a sikeres leadást követően RECEIVED legyen, a received\_on mező pedig kapja meg az aktuális időt értékként
- **Profil (/profile):**
  - Írja ki az aktuálisan bejelentkezett user adatait (nevét, email címét, és szerepkörét (sima felhasználó / admin))
- **Rendeléseim (/orders):**
  - Listázza ki a userhez kapcsolódó rendeléseket
  - A rendelésnél látszódjon a rendelt dolgok listája, a teljes ár, valamint a rendelés állapota (feldolgozás alatt, elfogadva, elutasítva)

## 2. FELVONÁS

**Határidő: 2020. 11. 29. 23:59**

A második felvonáshoz az első felvonásban létrehozott alkalmazást és annak funkcióit kell továbbfejleszteni, kiegészíteni.

- **Felhasználó:**
  - **Menü (/menu):**
    - Listázza ki valamilyen módon az item-eket (pl. card-ként)!
    - **Az item-hez jelenjen meg egy kép is, ha pedig nincs hozzá feltöltve kép, akkor egy default kép legyen ahelyett placeholder-ként**
  - **Rendeléseim (/orders):**
    - A rendelésnél látszódjon a rendelt dolgok listája, a teljes ár, valamint a rendelés állapota (feldolgozás alatt, elfogadva, elutasítva)
    - **Ha az item törölve van, akkor pirossal jelenjen meg a törölt item neve**
- **Admin:**
  - Tudjon kategóriát létrehozni, szerkeszteni és törölni
    - Kategória létrehozó form: **/admin/category/new**
    - A kategória létrehozó form ide küldje el az adatokat: **/admin/category/store**
    - Kategória szerkesztő form: **/admin/category/{id}/edit**
    - A kategória szerkesztő form ide küldje el az adatokat: **/admin/category/{id}/update**
    - Kategória törlése: **POST /admin/category/delete**
  - Tudjon itemet létrehozni, szerkeszteni, törölni és visszaállítani
    - A logika ugyanaz, mint a kategóriánál, törölt itemet pedig ne lehessen szerkeszteni
      - Ha törölt item-et próbál szerkeszteni a felhasználó, akkor elég egy hibaüzenet, ami jelzi, hogy az item törölve van, de aki akarja, csinálhat egy például egy "Visszaállítás" gombot is, és az admin közvetlenül a hibaüzenetből vissza is tudja állítani az item-et
    - A visszaállítás route-ja ez legyen: **POST /admin/item/{id}/restore**

- Fontos, hogy ha az item a soft delete funkcionalitást használja, akkor "nem törlődik ki ténylegesen", csak kap egy deleted\_at mezőt
- Rendelések kezelése (/admin/manage)
  - Az admin tudja kezelni a rendeléseket, ez azt jelenti, hogy amikor bejön egy rendelés, annak a status mezője RECEIVED lesz, és az admin dönti el, hogy elfogadja (ACCEPTED) vagy elutasítja (REJECTED) a rendelést (POST /admin/manage/accept/{orderId}) és (POST /admin/manage/reject/{orderId}). Ezt jelenti a feldolgozás, és ekkor kap értéket a 'processed\_on' mező is (now).
  - Alapvetően a rendelések feldolgozása két oldalra bomlik:
    - Folyamatban levő rendelések (/admin/manage/received):
      - Azok a rendelések jelennek meg, amik még nem lettek feldolgozva (tehát elfogadva vagy elutasítva)
      - A rendelés mutasson egy hivatkozásra, amivel betölthető a rendelés "adatlapja":
        - Melyik user adta le (név, email)
        - Mikor
        - Tételesen miből, mennyit rendelt
        - Mi a megjegyzés, cím
      - Ezen az adatlapon legyen lehetőség egy rendelést elfogadni vagy elutasítani
    - Feldolgozott, már lezárt rendelések (/admin/manage/processed):
      - A már lezárt rendelések vannak itt.
      - Itt ugyanaz a logika, mint a folyamatban levő rendeléseknél, tehát megjelenik ugyanaz az adatlap oldal a rendelésre kattintva, annyi különbséggel, hogy alul nem 2 gomb van, hanem csak annyi, hogy a rendelés már fel lett dolgozva
- Nem probléma, ha az admin is tud rendelést leadni, de megcsinálhatod úgy is, hogy ne tudjon

## TOVÁBBI KÖVETELMÉNYEK

- **Ezek a követelmények mindkét felvonásra egyaránt vonatkoznak!**
- Az alkalmazást Laravel 8 keretrendszerben kell megvalósítani.
- Az alkalmazást úgy kell elkészíteni, hogy a zip-ből kicsomagolva az alábbi parancsok kiadásával elinduljon:

```
composer install
npm install
npm run dev
php artisan migrate:fresh
php artisan db:seed
php artisan serve
```

- A felhasználói felület igényesen nézzen ki. Erre tökéletes valamilyen CSS framework és komponenskönyvtár, mi a Twitter Bootstrap használatát ajánljuk, amit a Laravel UI alapból biztosít, az órákon is azt használtuk.
- Az időzóna legyen magyarra állítva ( config/app.php ).
- A beküldött form-okat minden esetben validálni kell.
- Az alkalmazást úgy kell beállítani, hogy az SQLite adatbázist használjon, amit a database/database.db fájl kell, hogy biztosítson.
- Az alkalmazás legyen feltöltve adatokkal (seedelés), **a felhasználói fiókok pedig az alábbiak legyenek, hogy ne kelljen minden alkalmazáshoz külön kikeresni, ki mit talált ki:**
  - Admin: admin@szerveroldali.hu - password
  - Felhasználók:
    - user1@szerveroldali.hu - password
    - user2@szerveroldali.hu - password
    - ...
- **Kötelező** mellékelni a munka tisztaságáról szóló STATEMENT.md nevű nyilatkozatot, a részleteket lásd lentebb.

- **Tilos** mellékelni a **vendor** és a **node\_modules** mappákat! Ne terheld a rendszert azzal, hogy feleslegesen 30-40 MB méretű beadandót adsz be! Mikor kijelölöd a becsomagolni kívánt állományokat, **hagyd ki** ezeket a mappákat!
- Canvas-ben az 1. és a 2. fázishoz is fel kell tölteni a beadandót, mivel úgy lesz értékelve!

## SEGÍTSÉGGKÉRÉS, KONZULTÁCIÓS ALKALMAK

Ha bárkinek kérdése van a beadandóval kapcsolatban, akkor keresse az oktatókat Teams üzenetben (Horváth Győző, Tóta Dávid).

Ezen felül az őszi szünetben, valamint az 1. és a 2. felvonás határidejének lejárta előtt is lesz egy-egy konzultációs lehetőség, amelyeknek a konkrét időpontját majd egyeztetjük a Teams csoportban. A konzultációk alkalmával szintén lehet kérdéseket feltenni, de próbálunk általánosan is segíteni.

## HATÁRIDŐK, KÉSÉS

A beadandót igyekeztünk úgy kialakítani és olyan beadási határidőkkel ellátni, hogy azt mindenkinek legyen elegendő ideje rendesen, nyugodt körülmények között kidolgozni. Oszd be az idődet, ne az utolsó pillanatokra hagyd a feladatot!

A határidők lejárta után még van 2 hét türelmi idő, pontlevonásért cserébe. 1 hét késés -6 pont, 2 hét késés pedig -12 pont levonást jelent az adott felvonásból. Érdemes szem előtt tartani, hogy a 40% elérése kötelező a tárgy követelményei szerint.

## A MUNKA TISZTASÁGA

Kérjük, hogy a saját érdekében csak olyan munkát adj be, amit Te készítettél! A hasonló megoldások eredményes kiszűrése érdekében a beadott munkák szigorú, több lépcsőből álló plágiumellenőrzési folyamaton mennek keresztül. Ezen felül bizonyos gyanús esetben felkérhetünk arra, hogy véd meg szóban a munkád. Ezzel a módszerrel fel tudjuk mérni, hogy mennyire ismered, érted, valamint mennyire látod át azt a programot, amit beadtál.

A beadott feladatnak tartalmaznia kell egy STATEMENT.md nevű fájlt, a következő tartalommal (értelemszerűen az adataidat behelyettesítve):

## # Nyilatkozat

Én, <NÉV> (Neptun kód: <NEPTUN KÓD>) kijelentem, hogy ezt a megoldást én küldtem be a Szerveroldali webprogramozás Laravel beadandó feladatához.

A feladat beadásával elismerem, hogy tudomásul vettem a nyilatkozatban foglaltakat.

- Kijelentem, hogy ez a megoldás a saját munkám.
- Kijelentem, hogy nem másoltam vagy használtam harmadik féltől származó megoldásokat.
- Kijelentem, hogy nem továbbítottam megoldást hallgatótársaimnak, és nem is tettem azt közzé.
- Tudomásul vettem, hogy az Eötvös Loránd Tudományegyetem Hallgatói Követelményrendszere (ELTE szervezeti és működési szabályzata, II. Kötet, 74/C. §) kimondja, hogy mindaddig, amíg egy hallgató egy másik hallgató munkáját - vagy legalábbis annak jelentős részét - saját munkájaként mutatja be, az fegyelmi vétségnek számít.
- Tudomásul vettem, hogy a fegyelmi vétség legsúlyosabb következménye a hallgató elbocsátása az egyetemről.

A beadandót nem értékeljük, amíg a STATEMENT.md nincs mellékelve, vagy az egyedi azonosítók nincsenek benne átírva (név, neptun kód).

## HASZNOS HIVATKOZÁSOK

Az alábbiakban adunk néhány hasznos hivatkozást, amiket érdemes szemügyre venni a beadandó elkészítésekor.

- **Példaalkalmazás 1**
- **Példaalkalmazás 2**
- **Laravel nyelvi csomag - magyarosításhoz**
- Tantárgyi Laravel jegyzetek
  - **Kimenet generálása**
  - **Bemeneti adatok, úrlapfeldolgozás**
  - **Adattárolás, egyszerű modellek**
  - **Relációk a modellek között**
  - **Hitelesítés és jogosultságkezelés**
- Dokumentációk
  - **Laravel dokumentáció**
  - **Laravel API dokumentáció**
  - **PHP dokumentáció**
  - **Bootstrap dokumentáció**
- Programok, fejlesztői eszközök
  - **Automatikus PHP és Composer telepítő**
  - **Visual Studio Code**
    - **Live Share**
    - **Laravel Extension Pack**
  - **DB Browser for SQLite**
- További CSS framework tippek
  - **Fontawesome ikonkészlet**
  - **Material Bootstrap**
  - **Materialize**
  - **Bulma**

## CHANGELOG

- október. 31.
  - [Pótolva] A feladat elején az Order-ből lemaradt a processed\_on mező.
  - [Új] Az Orderhez hozzá kell adni ezen felül egy received\_on mezőt is, mivel ha CART statusba kerül, akár 1 héttel utána is le lehet adni rendelésként, ezért kell egy időpont, hogy konkrétan mikor lett leadva a rendelés.
  - [Javítva] Az Item és az OrderedItem között 1-N kapcsolat kell
  - [Pótolva] Amíg az Order CART-ként működik, a rendelés leadásához szükséges mezők lehetnek null/default értékűek (pl. address)
  - [Új] A Menü oldal elején listázni kell a kategóriákat, és biztosítani kell, hogy meg lehessen tekinteni az egyes kategóriákhoz tartozó item-eket.
  - [Új] A Menü oldalon az item-eket kategóriák szerint rendezve kell megjeleníteni.
  - [Új] Opcionálisan készíthető az item-nek saját oldal és onnan is berakható kosárba.
  - [Javítva] Az item kosárba rakásának útvonala megváltozott, hogy könnyebb legyen átadni a mennyiséget.
  - [Pótolva] Az admin route-oknál lemaradt a delete útvonal.
  - [Új] Az admin ne tudjon törölt item-et szerkeszteni.
  - [Pótolva] Ki lett emelve, hogy az admin route-oknál a restore POST.
  - [Új] A hivatkozások közé bekerült egy 2. példaprogram is.

© ELTE IK 2019.  
Powered by **Gumdrop** and **MDSS**

Horváth Győző