

 [laravel_beadando_2023_24_2.md](#)

Szerveroldali webprogramozás 2023/24/2 - Laravel beadandó

- [Követelmények](#)
 - [Minimumkövetelmények](#)
 - [Általános elvárások](#)
 - [Értékelési szempontok](#)
- [Feladat \(50 pont\)](#)
 - [Checklist](#)
 - [Adatbázis \(3 pont\)](#)
 - [Seeder \(3 pont\)](#)
 - [Főoldal \(2 pont\)](#)
 - [Karakterek listázása oldal \(2 pont\)](#)
 - [Karakter részletes adatai tartalmazó oldal \(2 pont\)](#)
 - [Új karakter létrehozása \(6 pont\)](#)
 - [Karakter módosítása \(4 pont\)](#)
 - [Karakter törlése \(2 pont\)](#)
 - [Új mérkőzés létrehozása \(3 pont\)](#)
 - [Mérkőzés oldal \(10 pont\)](#)
 - [Helyszínek listázása oldal \(2 pont\)](#)
 - [Új helyszín létrehozása \(2 pont\)](#)
 - [Helyszín módosítása \(3 pont\)](#)
 - [Helyszín törlése \(2 pont\)](#)
 - [Védésre szerezhető pontszám \(4 pont\)](#)

Követelmények

Figyelmesen olvass el minden követelményt, általános elvárást és értékelési szempontot, mielőtt megkezded a munkát!

Ha a feladattal vagy a követelményekkel kapcsolatban bármilyen kérdésed van, akkor nyugodtan kérj segítséget a gyakorlatvezetődtől, de írhatasz a *Teams* csoport *Általános* csatornájára is.

Minimumkövetelmények

Amennyiben a beadandó feladat nem felel meg **BÁRMELYIK** minimumkövetelménynek, úgy nem áll módunkban elfogadni, és nem teljesítheted a tárgyat a félévben!

▼ Minimumkövetelmények megtekintése

- Határidő betartása: **2024. április 28. 23:59!**
 - Késés alapvetően **NEM** megengedett; kivéve egyéni megegyezésre a gyakorlatvezető felé **időben jelzett, igazolható, méltányossági** alapon (pl. haláleset, tartós betegség). *(A határidő köteleddel érkező "el kellett utaznom", "zh-t írtam tegnap" és "nem működik a(z) gépem/internet" szintű megkereséseket **mérlegelés nélkül** el fogjuk utasítani, hiszen circa **1,5 hónap** állt rendelkezésre ezek áthidalására.)*
- Az alkalmazást **Laravel 11** keretrendszerben, **SQLite** adatbázis használatával kell megvalósítani!
 - Ha az ORM-et szabályosan használod, akkor egyébként nem számít, hogy milyen adatbázis fut a háttérben a fejlesztés során, de mi a teszteléshez és értékeléshez **SQLite** adatbázist fogunk használni.
- Az elkészült munkát a tárgy **Canvas-felületére** kell feltölteni az alábbiak szerint:
 - egyetlen **.zip** archívumot kell beadni
 - az archívum **NEM tartalmazhatja** a csomagkezelők által karbantartott **vendor** és **node_modules** mappákat, SEM a **.git**, **.idea**, **.vscode** mappákat
 - beadott **.zip**-ből az alkalmazás a következő inicializációs fájlokkal (és nem többel) **beüzemeltető** legyen:
 - [init.bat](#) (Windows-on)
 - [init.sh](#) (Linux-on / Mac-en)
 - A hallgató saját **felelőssége**, hogy a feltöltést időben megkezdje; valamint meggyőződjön róla, hogy a végleges kidolgozást tartalmazó helyes állományt adja be a fenti követelményeknek megfelelően! **Ezzel kapcsolatban utólag semmilyen kifogást nem fogadunk el, nincs lehetőség a határidő után a beadást javítani vagy pótolni!**
- Minden megvalósított funkció **értelmes módon** elérhető kell legyen a frontendről is!
- Az alkalmazás nem dobhat **kritikus hibát** a beüzemelés és alapvető használat során! Ha a beadás nem inicializálható a fenti scriptekkel vagy az átlag felhasználó kritikus hibát tud okozni, akkor a beadandó feladat nem értékelhető!
- A **.zip** gyökérmappájába a **<NÉV>** és **<NEPTUN>** mezők kitöltése után mellékelni kell az alábbi nyilatkozatot **STATEMENT.md** néven:

Nyilatkozat

Hallgató neve: <NÉV>

Hallgató Neptun kódja: <NEPTUN>

Ezt a megoldást a fent nevezett hallgató küldte be és készítette a Szerveroldali webprogramozás tárgy Laravel beadandó számonkérésére.

Kijelentem, hogy ez a megoldás a saját munkám. Nem másoltam vagy használtam harmadik féltől származó megoldásokat. Nem továbbítottam megoldást hallgatótársaimnak, és nem is tettem közzé. Nem használtam mesterséges intelligencia által generált kódot, kódrészletet. Az ELTE HKR 377/A. § értelmében, ha nem megengedett segédeszközt veszek igénybe, vagy más hallgatónak nem megengedett segítséget nyújtok, a tantárgyat nem teljesíthetem.

ELTE Hallgatói Követelményrendszer, IK kari különös rész, 377/A. §: "Az a hallgató, aki olyan tanulmányi teljesítménymérés (vizsga, zárthelyi, beadandó feladat) során, amelynek keretében számítógépes program vagy programmodul elkészítése a feladat, az oktató által meghatározottakon kívül más segédeszközt vesz igénybe, illetve más hallgatónak meg nem engedett segítséget nyújt, tanulmányi szabálytalanságot követ el, ezért az adott félévben a tantárgyat nem teljesítheti és a tantárgy kreditjét nem szerezheti meg."



Általános elvárások

▼ Általános elvárások megtekintése

Az egyes részfeladatok korrekt megvalósításához a következők is hozzátartoznak akkor is, ha az adott feladatnál nem részleteztük:

- Minden esetben a Laravel által biztosított **validációs** lehetőségeket **kötelező** alkalmazni! A kliensoldali validációért **pontlevonás** jár, hiszen nem erre vagyunk kíváncsiak ezen a tárgyon!
- Az űrlapoknak **állapottartóknak** kell lenniük, azaz ha hiba történik, akkor a felhasználó által megadott adatokat vissza kell tölteni az űrlapba, illetve a **pontos hibaüzeneteket** kell jeleníteni.
- Szerkesztés esetén az űrlapot ki kell tölteni a **meglévő** adatokkal.
- Ha valamilyen **hiba történik**, akkor azt minden esetben érthetően kell jelezni a felhasználó számára. Ha a kódod "elszáll" vagy lefagy, az **pontlevonást** jelenthet - súlyos esetben a beadandó feladat pontozás nélkül elutasítható!
- A **jogosultságkezelésnél** nem elegendő csak a frontendhez kötődő végpontokat levédeni, hanem a műveletet végző végpontnál is kell ellenőrzés.
- A frontendhez használt technológia szabadon választható, de nem érdemes túlbonyolítani, mivel pontot nem ér. Ajánlott az órán tanultakat alkalmazni, de nem

kötelező. A frontendnek nem kell szépnek lennie, csak használhatónak.

Értékelési szempontok

▼ Értékelési szempontok megtekintése

- Összesen **50 pont** érhető el, amelyből **legalább 40%-ot**, tehát 20 pontot el kell érni a beadandó feladat sikeres teljesítéséhez. Lehet részpontokat is szerezni az egyes feladatok részleges megoldására.
- **Nem értékelünk** olyan beadásokat, amik nem felelnek meg hiánytalanul **MINDEN minimumkövetelménynek!**
- A kikommentelt vagy használatlan kódrészletekre **nem jár pont** akkor sem, ha egyébként helyes lenne.
- Az adatbázis kivételével az alkalmazást elsősorban (de nem kizárólag) **funkcionálisan** értékeljük, ezért minden funkció elérhető kell legyen a frontendről is **értelmes** helyről!
- A beadandók többlépcsős, automatizált **plágiumellenőrzésen** esnek át, ezért kérünk mindenkit, hogy a saját munkáját adja le, mivel a másolásokat észre fogjuk venni! Egyezés esetén **MINDEN** érintett hallgató automatikusan elégtelent kap! **Emiatt különösen érdemes arra ügyelni, hogy időközben ne szivároгjon ki a munka publikus tárhelyre vagy repozitóriumba, mert ilyen esetben nem fogunk nyomozásokat folytatni, hogy ki a forrás!**

Feladat (50 pont)

A beadandó feladat során egy egyszerűsített egy játékos módú, körökre osztott, arcade típusú harcolós játékot kell elkészítened Laravel keretrendszer használatával.

Szeretnénk, ha a feladatot alapvetően kellő **alkotói szabadsággal** fognátok meg, nem pedig kőbe vésett dologként. Lényegében minden (tisztá módon keletkezett) megoldás elfogadható, amíg az alább részletezett követelményeket teljesíti; tehát abban, ami nincs a továbbiakban specifikálva, **teljesen szabadon** mozoghattok. Érdemes jól tanulmányozni az elvárásokat, ugyanis **csak az ér pontot, amit expliciten leírtunk a pontozásban**; a kurzus teljesítése szempontjából tehát felesleges egy túlgondolt/bonyolultabb feladatot megoldani, persze mindig örülünk, ha extra szorgalmasak vagytok. :)

A feladathoz **kötelező kiinduló csomag nincs**, javasolt azonban a **Laravel Breeze** használata, amely a frontend beüzemelésén felül a hitelesítés alapját is biztosítja.

Checklist

A gyorsabb és egyszerűbb ellenőrzés érdekében jelöld be egy "X"-el, hogy mely feladatokat teljesítetted! Ezt másold be a beadandód `README.md` fájljába!

- [] - Adatbázis és modellek 3 pont
- [] - Seeder 3 pont
- [] - Főoldal 2 pont
- [] - Karakterek listázása oldal 2 pont
- [] - Karakter részletes adatai tartalmazó oldal 2 pont
- [] - Új karakter létrehozása 6 pont
- [] - Karakter módosítása 4 pont
- [] - Karakter törlése 2 pont
- [] - Új mérkőzés létrehozása 3 pont
- [] - Mérkőzés oldal 10 pont
- [] - Helyszínek listázása oldal 2 pont
- [] - Új helyszín létrehozása 2 pont
- [] - Helyszín módosítása 3 pont
- [] - Helyszín törlése 2 pont

Adatbázis (3 pont)

Készítsd el a megfelelő adatbázis táblákat és modelleket az alábbiak szerint *(az alapvető mezők (id, created_at, updated_at) minden táblában szerepeljenek):*

Modellek:

- User
 - ez a Laravel alapértelmezett táblája, egy mezőt kell hozzáadni:
 - admin [boolean] - adminisztrátor-e (alapértelmezett: false)
- Character - a felhasználók karakterei és az ellenséges, nem játékos karakterek
 - name [string] - a karakter neve
 - enemy [boolean] - nem játszható ellenséges karakter-e (alapértelmezett: false)
 - defence [integer] - védekező képesség (minimum: 0 , maximum: 3)
 - strength [integer] - támadó képesség (minimum: 0 , maximum: 20)
 - accuracy [integer] - pontosság képesség (minimum: 0 , maximum: 20)
 - magic [integer] - mágikus képesség (minimum: 0 , maximum: 20)
 - *a képességpontok (defence, strength, accuracy, magic) összege nem haladhatja meg a 20 -at*
- Place - a játéktér helyszínei
 - name [string] - a helyszín neve
 - kép a helyszínről
- Contest - a felhasználók mérkőzései
 - win [boolean] - a felhasználó nyert-e a mérkőzésen (nullable)
 - history - a mérkőzés története (a megvalósítást szabadon te választhatod meg; pl.: JSON, string, stb.)

Kapcsolatok:

- User 1 : N Character
 - egy felhasználó több karaktert is létrehozhat, de egy karakter csak egy felhasználóhoz tartozhat
- User 1 : N Contest
 - egy felhasználónak több mérkőzésen is részt vehet egy karaktere, de egy mérkőzésben csak egy felhasználó vehet részt (egyszemélyes a játék)
- Place 1 : N Contest
 - egy mérkőzésnek csak egy helyszíne lehet, de egy helyszínen több mérkőzés is zajlhat
- Character N : M Contest
 - egy karakter több mérkőzésen is részt vehet és egy mérkőzésen mindig két karakter vesz részt
 - a kapcsolótábla tartalmazza a következő mezőket az N : M kapcsolathoz szükséges idegen kulcsokon kívül:
 - hero_hp [float] - a felhasználó karakterének életerejének (minimum: 0 , maximum: 20 , alapértelmezett: 20)
 - enemy_hp [float] - az ellenséges karakter életerejének (minimum: 0 , maximum: 20 , alapértelmezett: 20)

Seeder (3 pont)

Készíts egy seedert, ami feltölti az adatbázist!

Ügyelj arra, hogy a seeder:

- minden eshetőségre fel legyen készítve, tehát kezelje a kapcsolatokat, illetve
- konzisztens adatokat generáljon, tehát pl. a karakterek képességpontszámainak összege ne haladja meg a 20-at; győzelem vagy vereség esetén a vesztes karakter életerejének legyen 0 ; ellenséges karakter csak admin felhasználó alatt legyen stb.
- a history mezőnek nem kell konzisztensnek lennie az életerők változását illetően

Fontos továbbá, hogy:

- ne beégetett adatokat használj, hanem generáld őket pl. *faker* használatával
- **nem kell rengeteg adatot generálni** - csak néhányat, amennyi elég a teszteléshez

Főoldal (2 pont)

A főoldalon jelenjen meg egy rövid ismertető szöveg a játékról, valamint jelenítsd meg a következő statisztikákat:

- a játékban létrehozott karakterek száma

- az eddigi mérkőzések száma Csak ez az oldal érhető el nem bejelentkezett felhasználók számára is.

Karakterek listázása oldal (2 pont)

Bejelentkezést követően automatikusan irányítsd át a felhasználót a saját karaktereinek listázására. Itt jelenjen meg egy táblázat, amiben a karakterek neve, védekező, támadó, pontosság és mágikus képességpontszámai szerepelnek. A táblázatban legyen lehetőség átnavigálni a karakter részletes adatait tartalmazó oldalra.

Karakter részletes adatai tartalmazó oldal (2 pont)

Jelenítsd meg a karakter minden adatát (név, védekező, támadó, pontosság és mágikus képességpontszámok). Továbbá egy listában jelenítsd meg a karakter mérkőzéseit is (helyszín és ellenfél neve). A mérkőzésre kattintva navigáld át a felhasználót a mérkőzés oldalra. Az oldalon legyen lehetőség a karakter szerkesztésére és törlésére. Az oldalon szerepeljen egy gomb, amivel új mérkőzést indíthat a felhasználó. Az oldal csak a karaktert létrehozó felhasználó számára legyen elérhető.

Új karakter létrehozása (6 pont)

Jeleníts meg egy űrlapot, ahol a felhasználó egy új karaktert tud létrehozni. A küldést követően végezd el a megfelelő validációkat a **Modellek** részben leírtak szerint, illetve a tulajdonságpontok (defence, strength, accuracy és magic) összegének 20-nak kell lennie. Az `admin` felhasználók is képesek saját karaktereket létrehozni, de nekik jelenjen meg egy mező, ahol ellenséges karakterként tudják létrehozni a karaktert (azaz a `enemy` mező értékét `true`-ra tudják állítani).

Karakter módosítása (4 pont)

Jeleníts meg egy űrlapot, ahol a felhasználó módosíthatja a karakter adatait. A küldést követően végezd el a megfelelő validációkat a **Modellek** részben leírtak szerint. A felhasználó csak a saját karakterét tudja módosítani. Az `admin` felhasználók a saját karaktereiken túl képesek az ellenséges karaktereket is módosítani (amelyeknél az `enemy` mező `true`).

Karakter törlése (2 pont)

Legyen lehetőség a karakter törlésére. A felhasználó csak a saját karakterét tudja törölni. Az `admin` felhasználók a saját karaktereiken túl képesek az ellenséges karaktereket is törölni (amelyeknél az `enemy` mező `true`).

Új mérkőzés létrehozása (3 pont)

Ha a felhasználó az egyik karakter részletező oldalán kiválasztja az új mérkőzés indítását, akkor készüljön egy új mérkőzés (Contest). A helyszín és az ellenfél véletlenszerűen legyenek kiválasztva. A létrehozást követően a felhasználó automatikusan kerüljön átirányításra a mérkőzés oldalára. Két karakter között több mérkőzés is lehetséges, akár szimultán is :)

Mérkőzés oldal (10 pont)

A mérkőzés oldalon jelenjen meg a helyszín neve és az oldal háttérképe legyen a helyszínhez feltöltött kép. Az oldalon jelenjenek meg a karakterek statjai (defence, strength, accuracy, magic), valamint az életerejük (hero_hp / enemy_hp). A felhasználó karakterének és az ellenfél karakterének cselekedetei is jelenjenek meg (history). Az oldalon legyen három gomb, melyek segítségével a felhasználó kiválaszthatja a következő támadását: melee, ranged vagy special (magic). Amennyiben a mérkőzés már véget ért (win mező nem null), akkor jelenítsd meg a végeredményt (győzelem vagy vereség) és ebben az esetben ne jelenjenek meg a támadó gombok.

Egy forduló menete:

- A mérkőzés kezdetén mindkét karakternek az életerejé legyen 20 .
- A felhasználó a három támadó gomb segítségével kiválasztja a következő cselekedetét.
- Ha a mérkőzés már véget ért, akkor ne történjen semmi.
- A kiválasztott támadás hatására számítsd ki a sérülés pontértékét a **Sérülés pontszámának kiszámítása** részben leírtak szerint.
- Frissítsd az adatbázisban az ellenséges karakter életerejét (ha negatív lenne, akkor legyen 0) és a karakterek cselekedeteihez (history) add hozzá a támadást.
 - pl.: CHARACTER_NAME: melee attack - 5 damage
- Ellenőrizd, hogy a mérkőzés véget ért-e (az ellenség karakterének életerejé (enemy_hp) 0).
 - Ha igen, állítsd a mérkőzés win mezőjét true -ra és irányítsd vissza a felhasználót a mérkőzés oldalra.
 - Ha nem, akkor az ellenség karaktere következik.
- Az ellenség karaktere véletlenszerűen válasszon támadást. Innentől kezdve minden ugyanúgy történik, mint fentebb a felhasználó karakterének támadásánál (sérülési pontszám kiszámítása, karakter életerejének és a history frissítése).
- Ellenőrizd, hogy a mérkőzés véget ért-e (a felhasználó karakterének életerejé (hero_hp) 0).
 - Ha igen, állítsd a mérkőzés win mezőjét false -ra.
- Irányítsd a felhasználót a mérkőzés oldalra.

Sérülés pontszámának kiszámítása:

Készíts egy metódust, ami kiszámítja a sérülés pontszámát a következők szerint:

- Paraméterek:
 - a támadás típusa (melee, ranged, special)
 - a támadó karakter (ATT) adatai (hp, defence, strength, accuracy, magic)
 - a védekező karakter (DEF) adatai (hp, defence, strength, accuracy, magic)
- Kimenet: a sérülés pontszáma (float)
- Sérülés kiszámítása:
 - Melee: $(DEF.HP) - ((ATT.STRENGTH * 0.7 + ATT.ACCURACY * 0.1 + ATT.MAGIC * 0.1) - DEF.DEFENCE)$
 - Ranged: $(DEF.HP) - ((ATT.STRENGTH * 0.1 + ATT.ACCURACY * 0.7 + ATT.MAGIC * 0.1) - DEF.DEFENCE)$
 - Special (magic): $(DEF.HP) - ((ATT.STRENGTH * 0.1 + ATT.ACCURACY * 0.1 + ATT.MAGIC * 0.7) - DEF.DEFENCE)$
 - Magyarul: a védekező karakter életerejéből vond ki a védekező karakter védekező képességével csökkentett támadó karakter támadó képességének súlyozott összegét.
 - **Ha a sérülés pontszám negatív lenne (nagyobb a védekező karakter védekező pontszáma (defence), mint a támadó karakter támadásának ereje), akkor 0-át adj vissza!**

Tipp a megvalósításhoz:

► Tipp megtekintése – a végleges megoldás ettől eltérhet, ha megakadtál, kattints ide egy lehetséges megvalósításért

Helyszínek listázása oldal (2 pont)

Csak admin felhasználók érhetik el! Jelenjen meg egy táblázat, amiben a helyszínek nevei és képei szerepelnek, továbbá legyen lehetőség a helyszínek szerkesztésére és törlésére.

Új helyszín létrehozása (2 pont)

Csak admin felhasználók érhetik el! Jeleníts meg egy űrlapot, ahol az admin felhasználó új helyszínt tud létrehozni. Kötelező képet feltölteni. A küldést követően végezd el a megfelelő validációkat a **Modellek** részben leírtak szerint.

Helyszín módosítása (3 pont)

Csak admin felhasználók érhetik el! Jeleníts meg egy űrlapot, ahol az admin felhasználó módosíthatja a helyszín adatait. Ekkor már nem kötelező képet feltölteni. Ha nem került új kép feltöltésre, akkor a régi kép maradjon meg. Ha került új kép feltöltésre, akkor a régi cserélődjön le az újra, a régit töröld ebben az esetben.

Helyszín törlése (2 pont)

Csak admin felhasználók érhetik el! Legyen lehetőség a helyszín törlésére, ne felejtse el a hozzá tartozó képet is törölni!

Védésre szerezhető pontszám (4 pont)

- További 4 pont szerezhető a védés során mutatott **általános jártasságra** a témában.
- Nem várjuk el senkitől, hogy másfél-két hónap alatt mesterévé váljon a Laravel lelki világának; viszont azt igen, hogy a **saját projektjét** alapvetően tudja navigálni és a **gyakorlaton lefedett** ismeretekkel kapcsolatos kiegészítő kérdésekre tudjon értékelhetően válaszolni. Normál esetben a védés 10-15 percnél tovább nem tart.
- **Amennyiben a hallgató a szóbeli védés során teljes tájékoztatlanságot mutat, az egész beadandó feladat visszautasítható csalás gyanújával!**
- Ha a gyakorlatvezető valamilyen okból nem tud védést szervezni, akkor az eddig elért pontszámmal arányosan meg kell adni a védésre járó pontokra.

anonymus1928 commented yesterday

Author

CHANGELOG 2024. 04. 12.

- MATCHES védett kulcsszó SQL-ben, ezért használható helyette alternatív elnevezés (pl.: "contest"- "contests")
- Pivot táblában az enemy_hp és hero_hp integer helyett float.