**EKVK**

**Csapattagok:**  
Konorót Kira Panka  
Boda Kristóf  
Busai Patrik Márk  
  
**1. Bevezetés**  
Ez az alkalmazás egy modern, internetes szövegszerkesztő, amely lehetővé teszi a felhasználók számára, hogy dokumentumokat hozzanak létre, szerkesszenek és mentsenek el a böngészőből. Az alkalmazás **csak regisztrált és bejelentkezett felhasználók számára érhető el**. Célunk, hogy egy gyors, megbízható és könnyen használható alternatívát nyújtsunk az online dokumentumkezelésre.

**2. Funkciók áttekintése**

* ✅ Felhasználói regisztráció és bejelentkezés
* ✅ Rich text szerkesztés (félkövér, dőlt, listák stb.)
* ✅ Dokumentumok tárolása és visszatöltése a felhasználói fiókhoz kötve
* ✅ Mobilbarát felület

**3. Technológiai háttér**

* **Frontend:** Tailwind CSS
* **Backend:** Node.js
* **Adatbázis:** MySQL
* **Fájlkezelés:** Szerveren vagy felhőben tárolt dokumentumok

**4. Telepítés (Fejlesztői környezethez)**

***Előfeltételek***

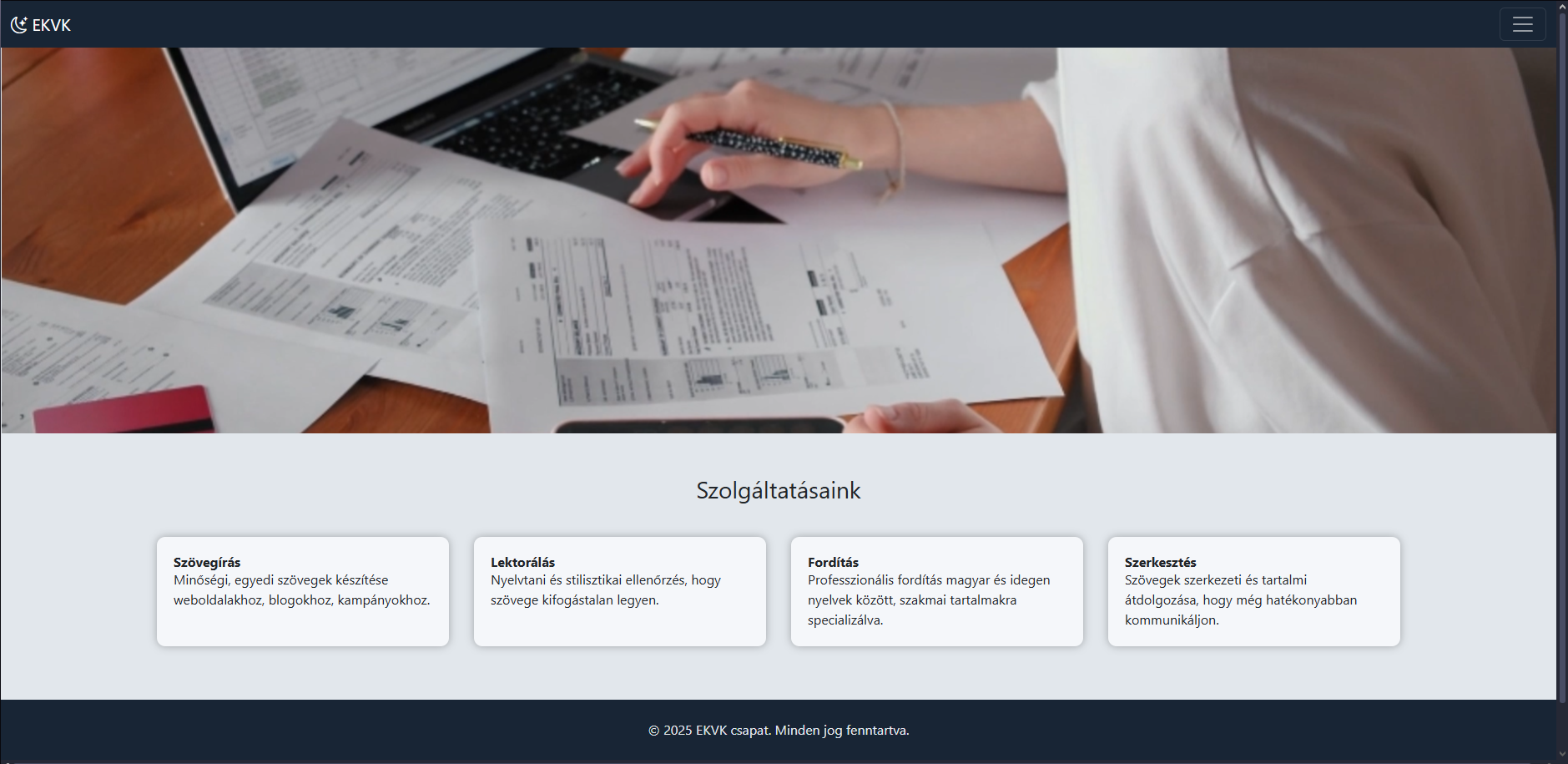
* Node.js (v18+)
* Git
* Xampp
* Composer
* PHP
* Laravel

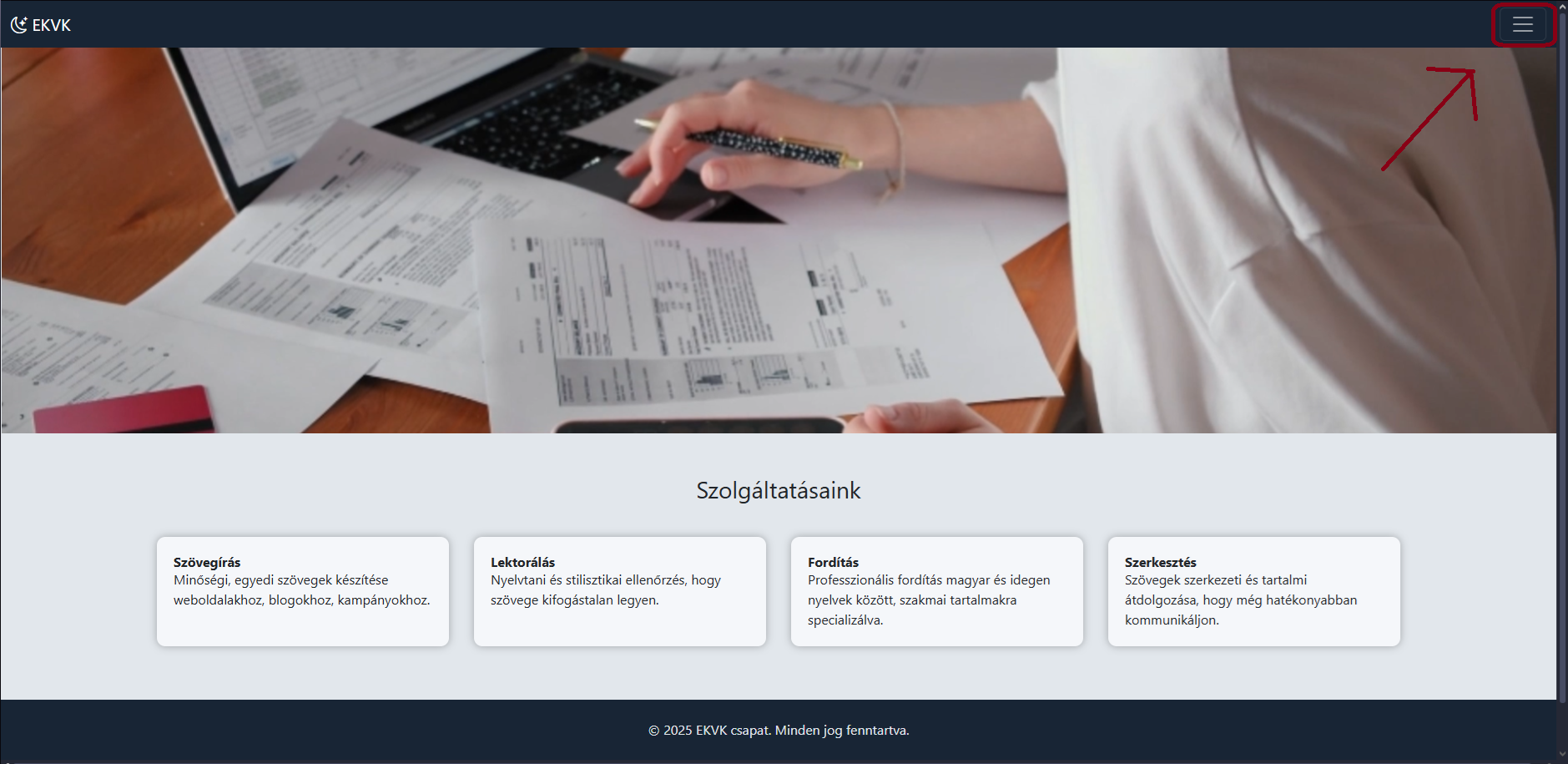
***Lépések***

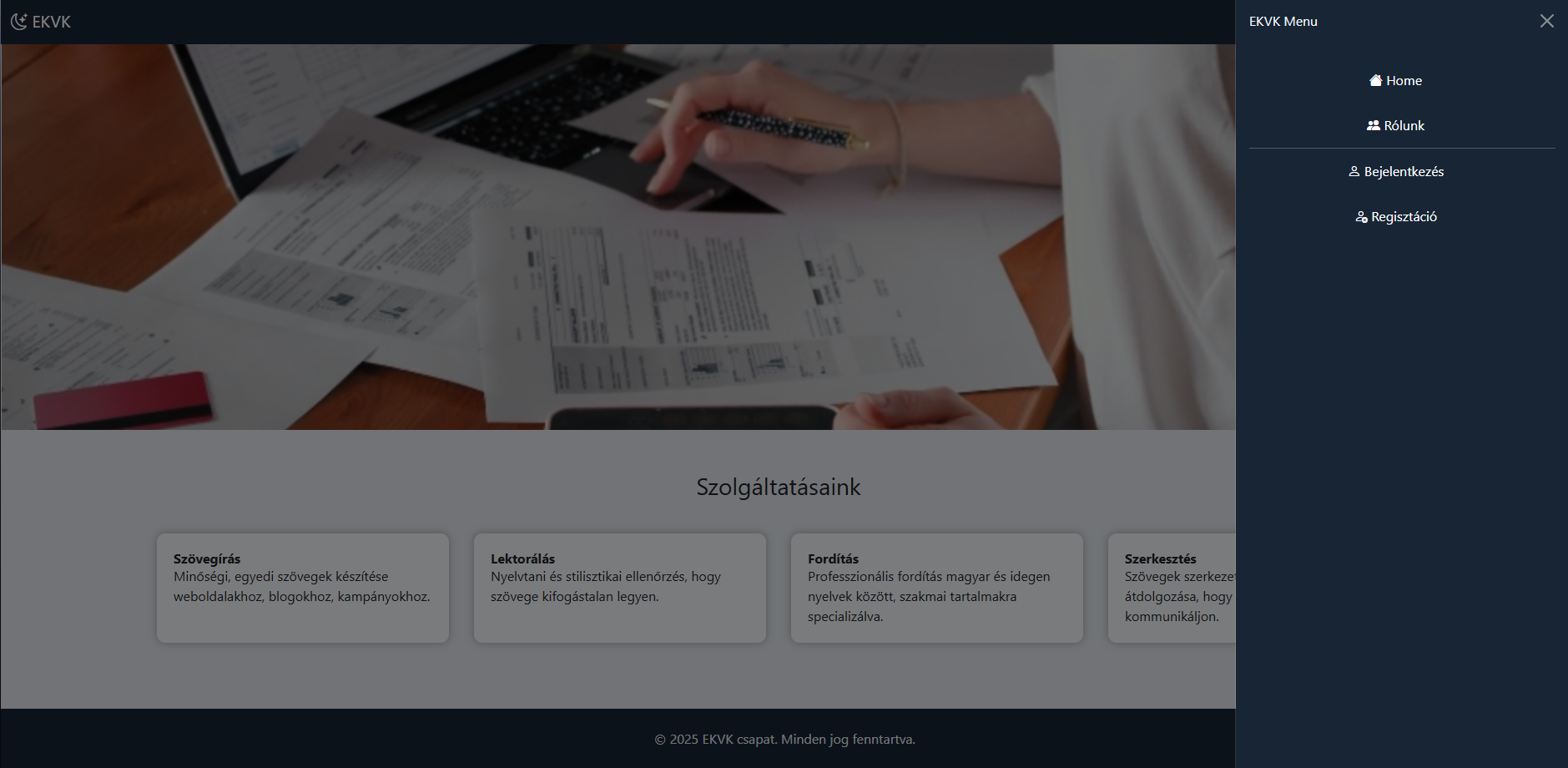
git clone <https://github.com/Raenon/EKVK_WebEditor>cd EKVK\_WebEditor  
npm install  
.env fájl létrehozása, a .env.example fájl átmásolása  
npm run dev  
php artisan serve

**5. Használati útmutató (Felhasználók számára)**

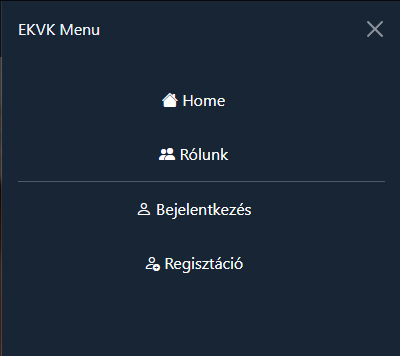
Lépésről lépésre  
  
**A Főoldal:**  
A főoldal a projektünk alapvető weboldala, az első dolog, amit meglát a felhasználó. Itt látható néhány alap információ a projektünkről és innen elérhető a navigációs menü.



**A Navigációs menü:**  


A Navigációs menü elhelyezkedése és csukott állapota  
  
  
A Navigációs menü nyitott állapota

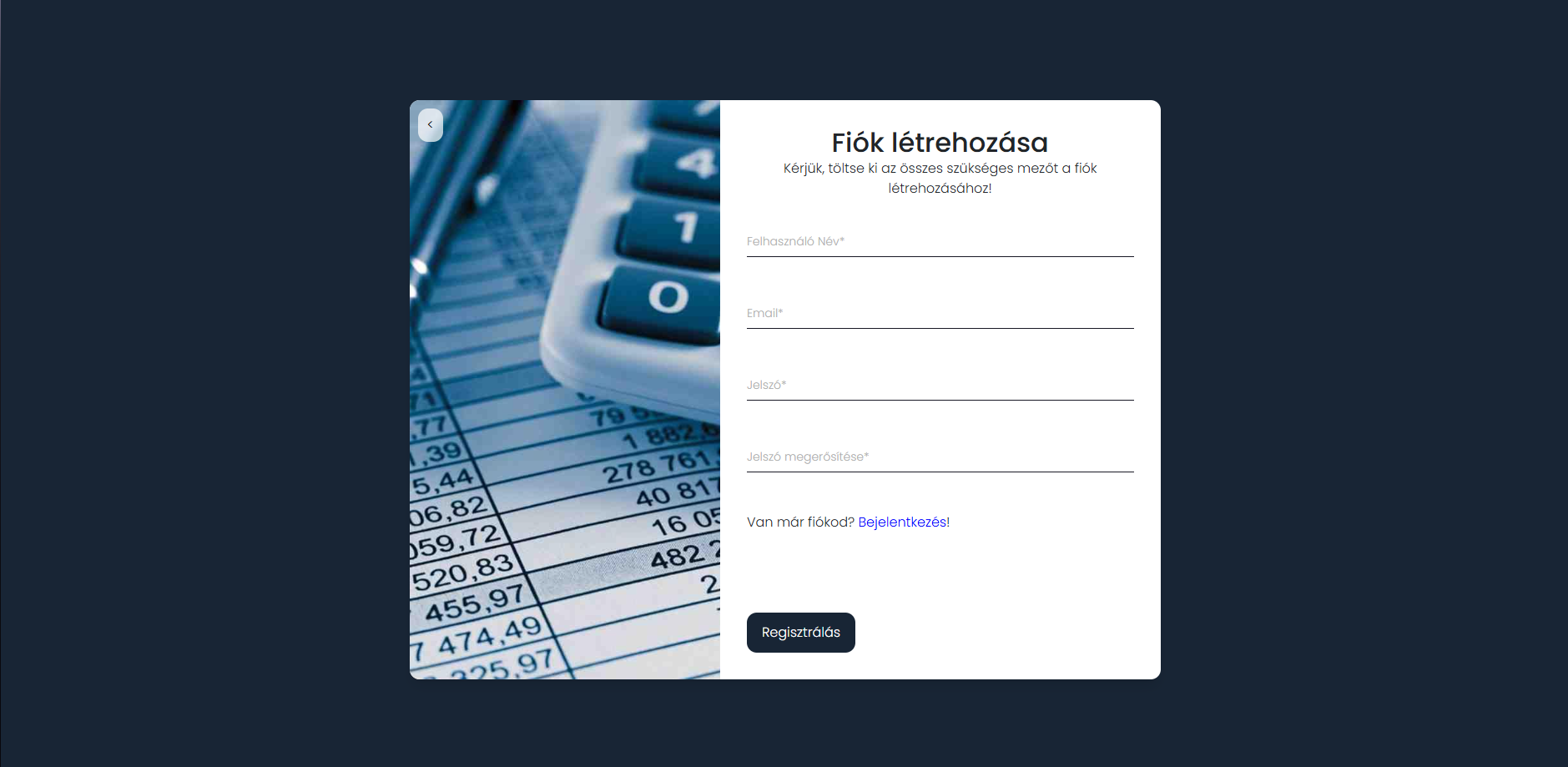
A navigációs menün találhatjuk a különböző menüpontokat, amik közül a felhasználók kiválaszthatják melyik aloldalakra lépnek tovább. A menü felhasználó típusonként változik.

  
A navigációs menü bejelentkezett felhasználó nélkül.

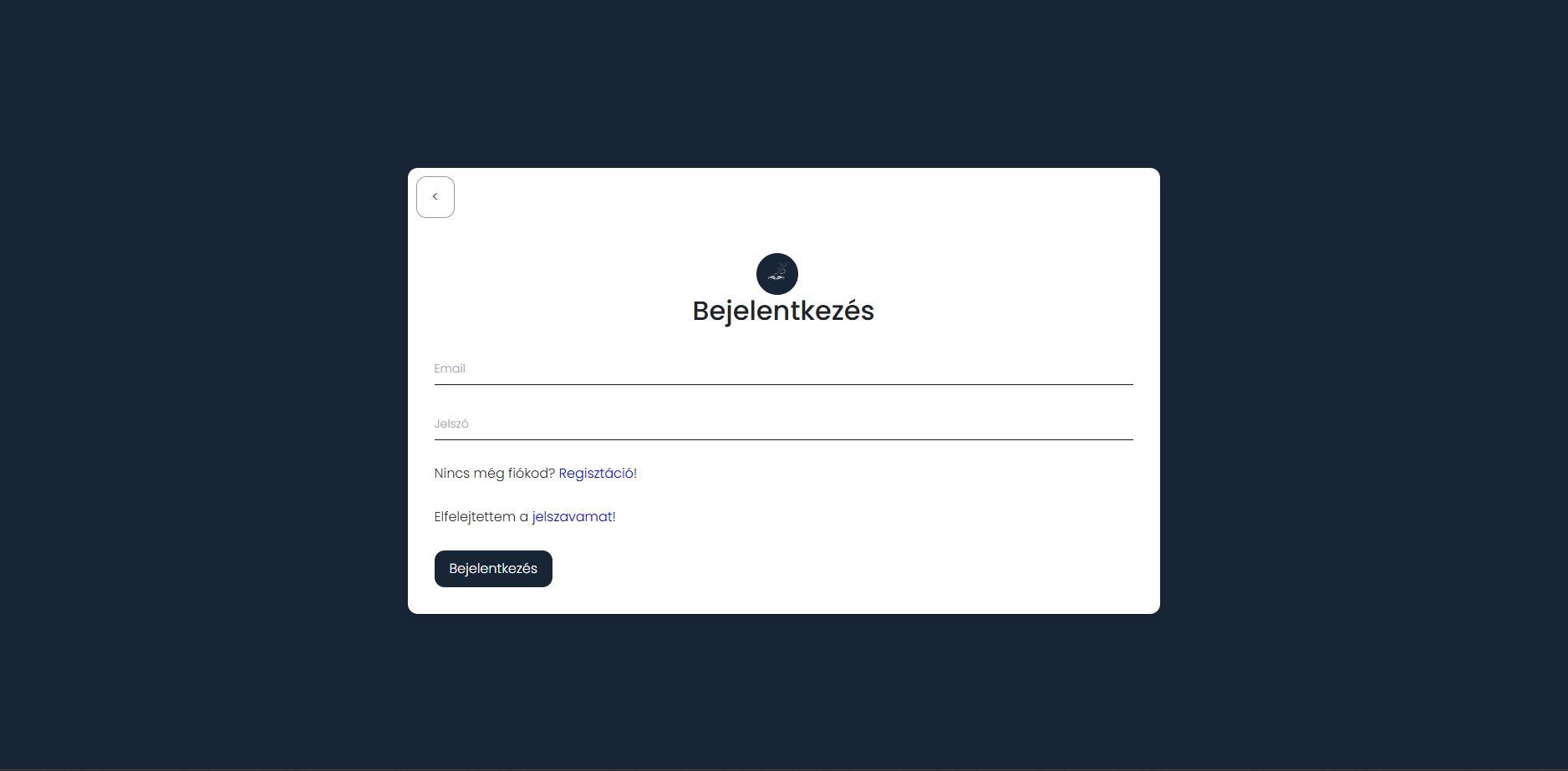
Mielőtt egy felhasználó bejelentkezne négy menüpont várja.

* ***Home,*** ami visszaviszi a felhasználót a főoldalra.
* ***Rólunk***, ami a projekt készítőiről néhány alap információt megjelenítő aloldalra irányítja a felhasználót.
* ***Bejelentkezés***, erre az opcióra kattintva tud a már regisztrációval rendelkező felhasználó bejelentkezni és az oldal további funkcióinak való hozzáférést szerezni.
* ***Regisztráció***, egy felhasználó, akinek még nincsen fiókja ezzel az opcióval tud eljutni az aloldalra, ahol tud regisztrálni.

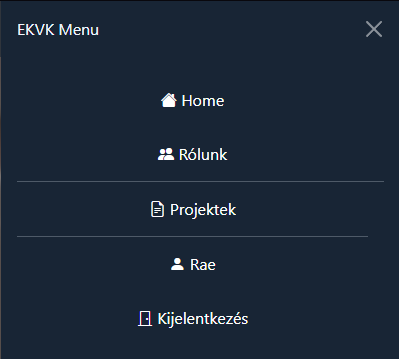
**Bejelentkezés és regisztráció:**

A regisztrációs oldal

A regisztrációs oldalon a felhasználónak kell megadnia egy felhasználó nevet, egy email cimet, és egy jelszót, amit egyszer meg kell erősítenie a sikeres regisztrációhoz. Miután a **Regisztráció** gombra kattint a felhasználó érvényes adatokkal sikeresen rögzítjük a felhasználót és visszakerül a főoldalra, ahonnan bejelentkezhet új fiókjába.

A bejelentkezési oldal

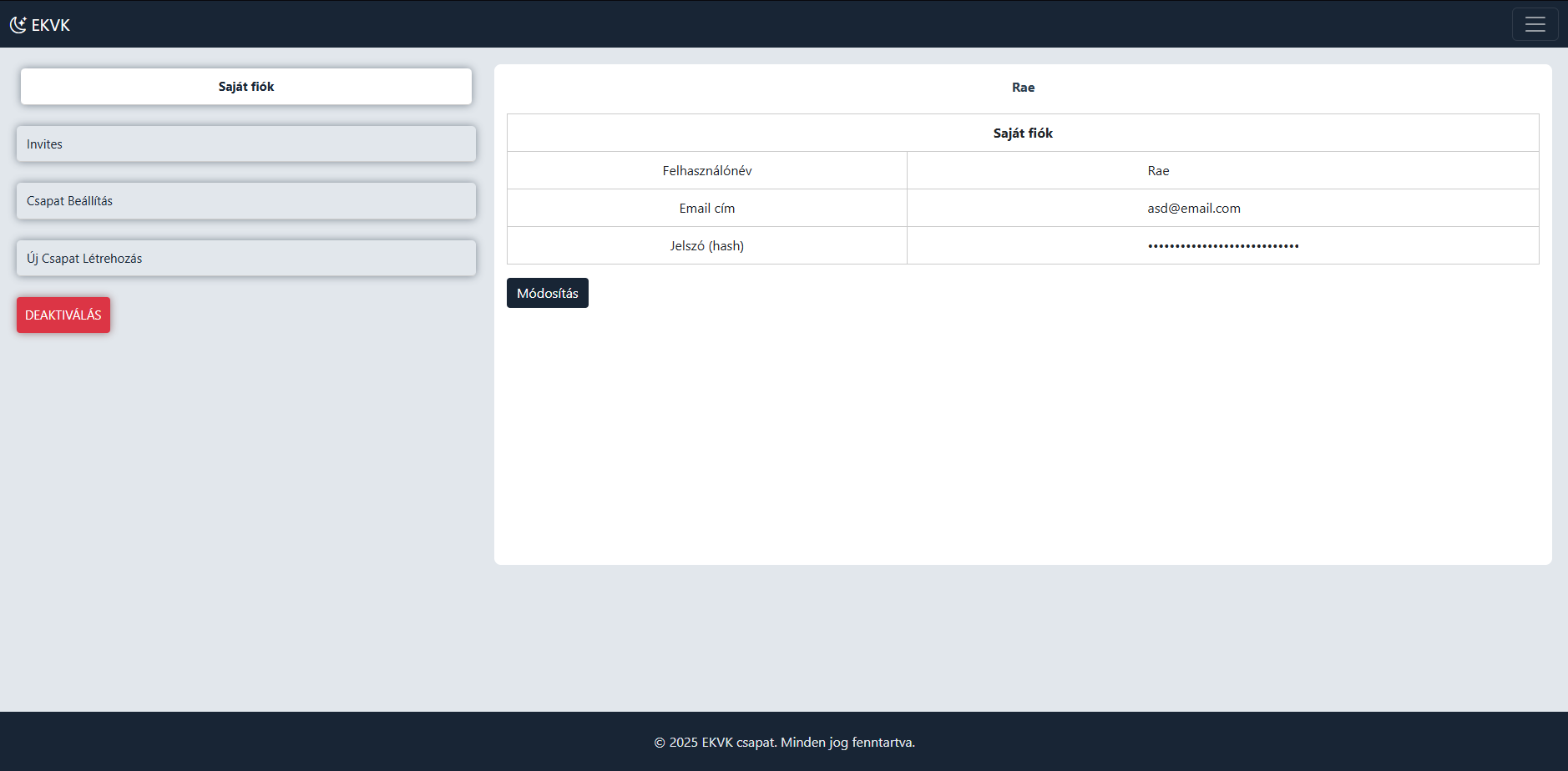
Ezen az oldalon tud a már regisztrációval rendelkező belépni fiókjába a rögzített email címével és jelszavával. Ha a felhasználó esetleg elfelejti a jelszavát van lehetősége email címmel megváltoztatni azt. Amint a felhasználó a **Bejelentkezés** gombra kattint érvényes bejelentkezési adatokkal, visszakerül a főoldalra, ahonnan kijelentkezésig hozzá fér a fiókja beállításaihoz, projektjeihez, és ki tud jelentkezni.

  
A navigációs menü bejelentkezett felhasználóval.

**A felhasználói oldal:**

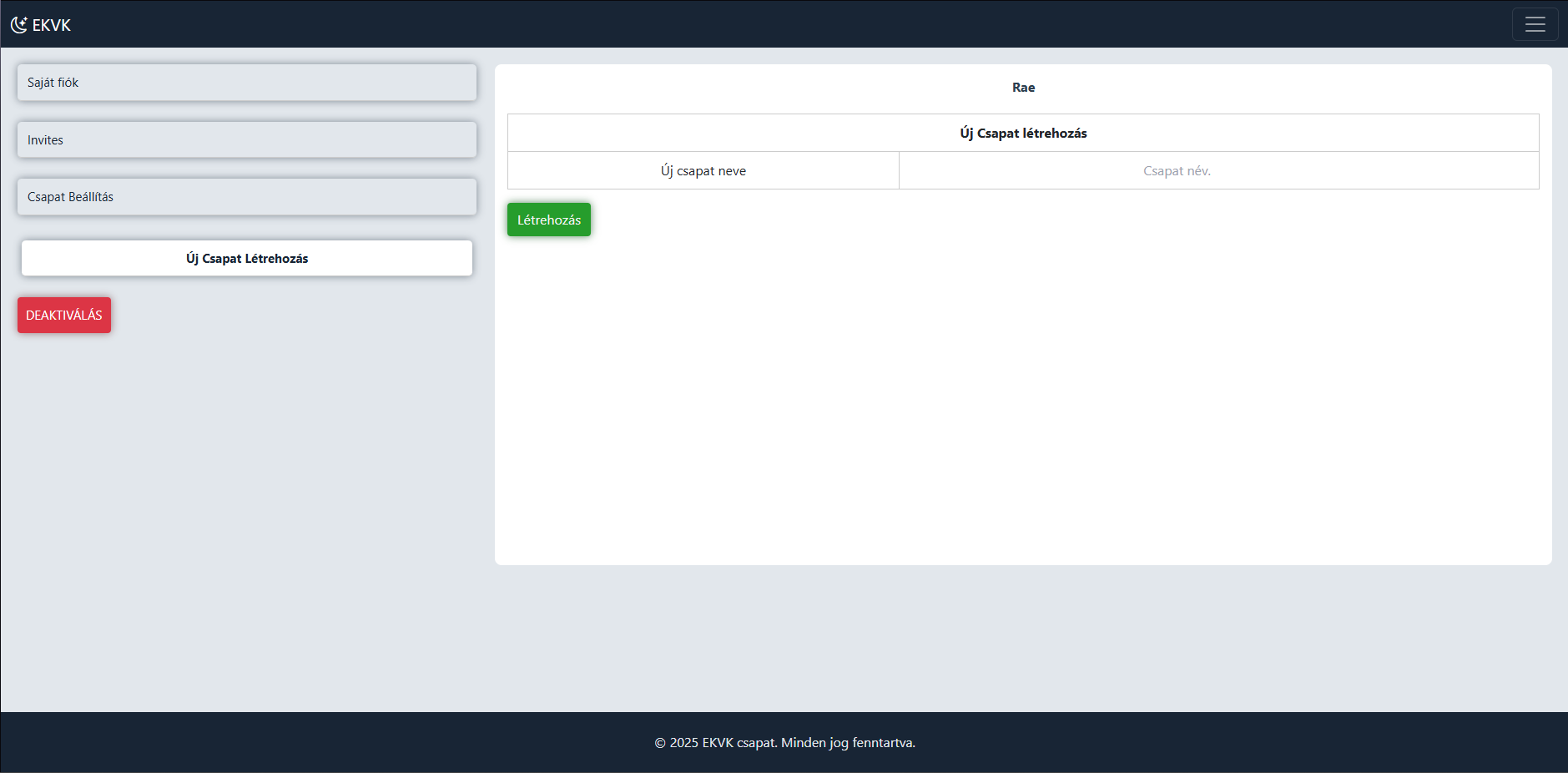
Ha a bejelentkezett felhasználó rákattint a gombra, amin a felhasználó neve látható átkerül a fiók oldalra, ahol,

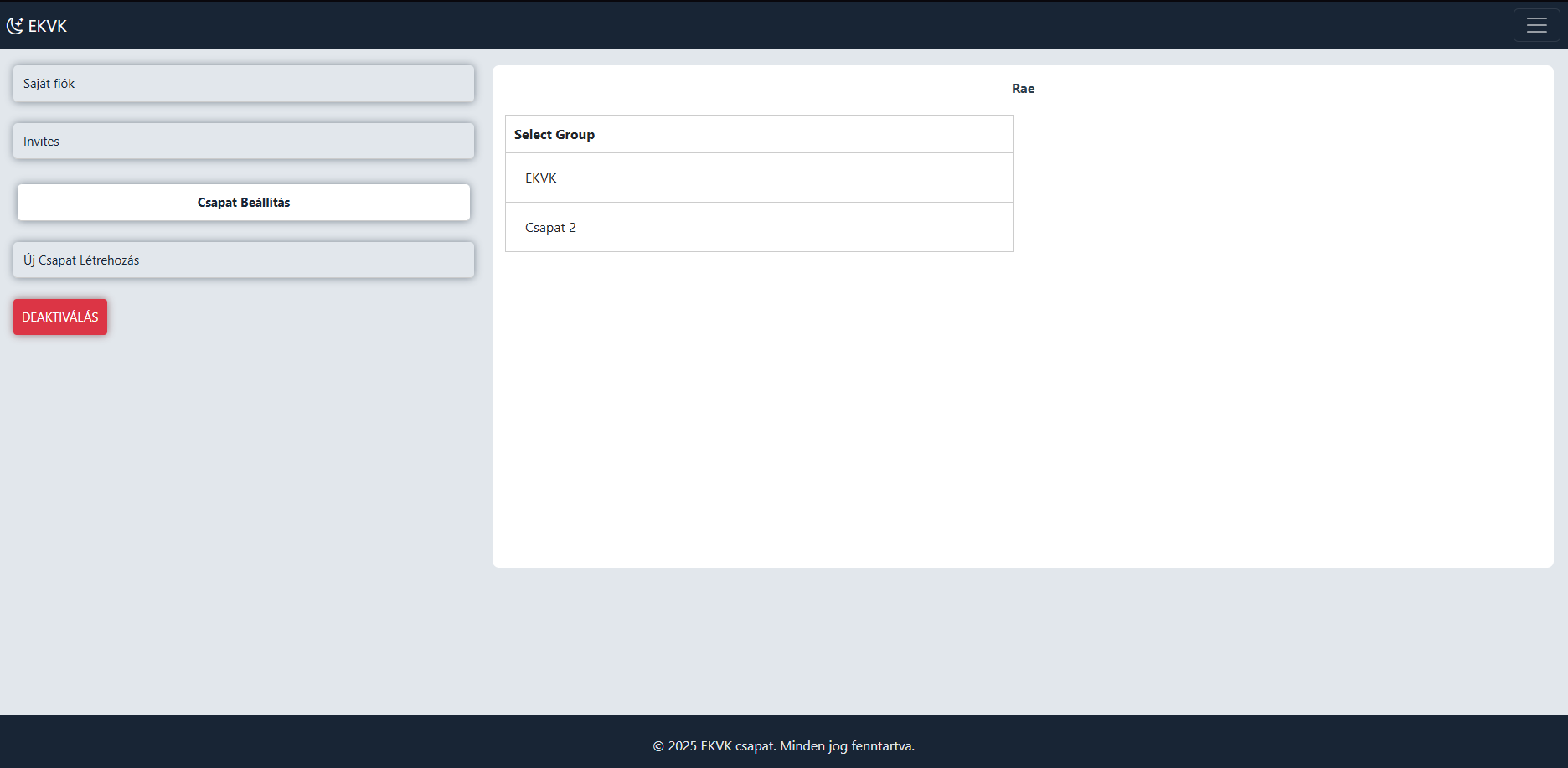
* Láthatja és Módosíthatja fiókja adatait
* Csapat kérelmeket fogadhat vagy utasíthat el
* Csapatot hozhat létre,
* Csapata beállítás oldalára kerülhet át
* Deaktiválhatja fiókját

A fiók oldal

**Csapatok:**

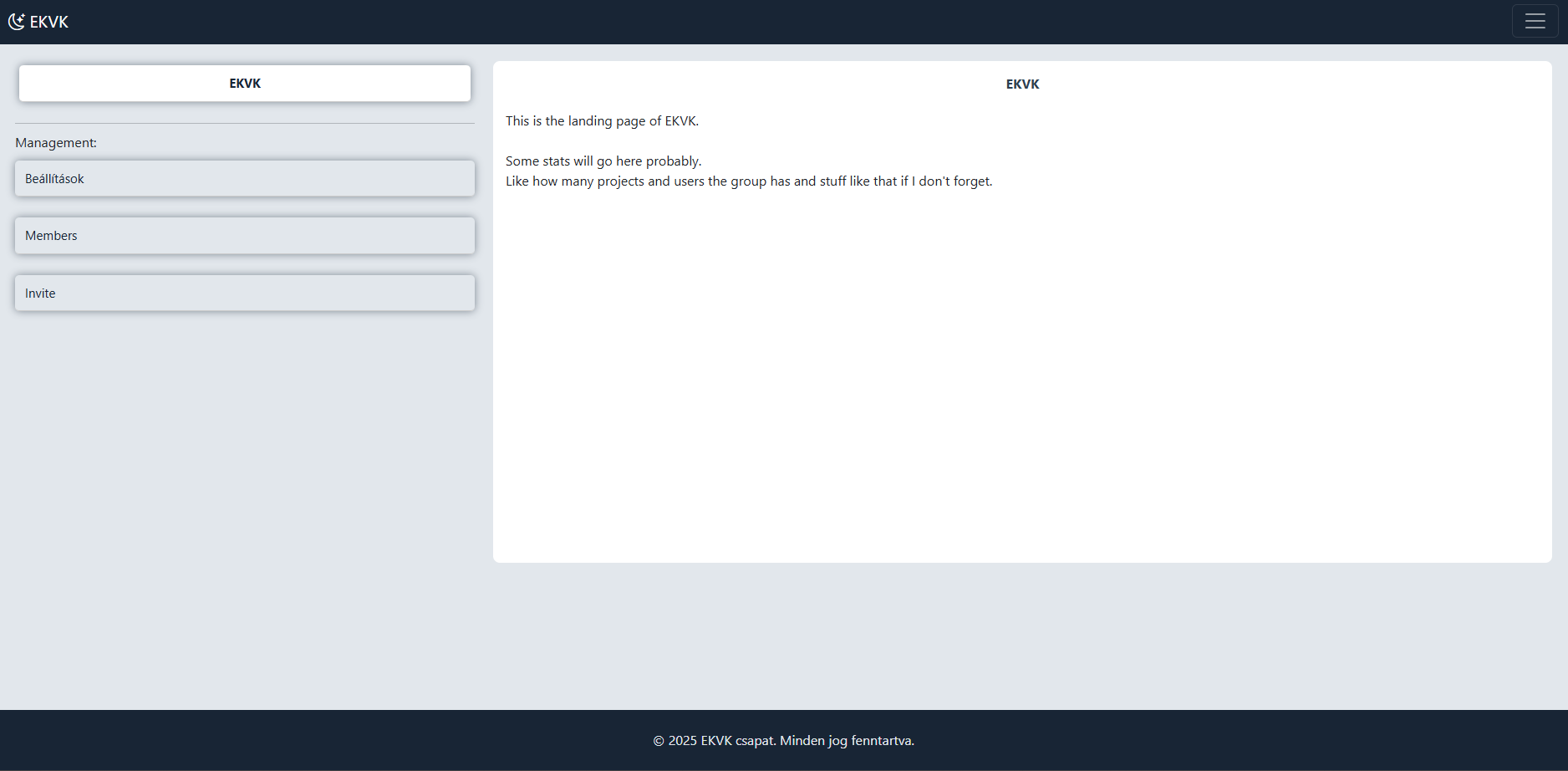
A felhasználók csapatokat tudnak létrehozni a közös projekt vezetés és szöveg szerkesztés érdekében. Új csapat létrehozásához az “Új Csapat Létrehozása” gombra kattintva, megkell egy csapat nevet adni és a “Létrehozás” gombra kattintani. Ha a felhasználó létrehozott egy Csapatot vagy már benne van egyben a “Csapat Beállítások” gombra kattintva tudja kiválasztani melyik csapat oldalára tud továbblépni.

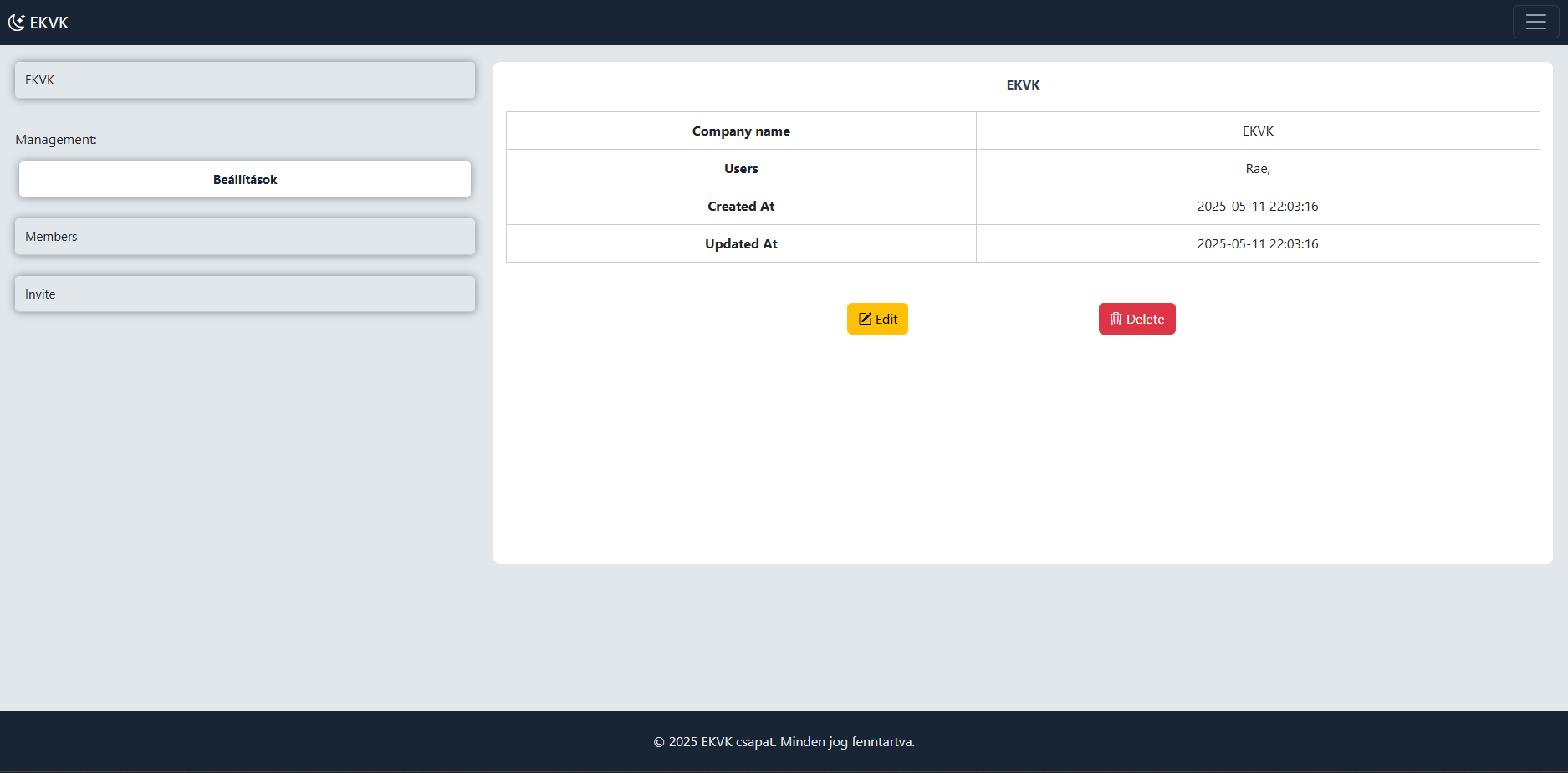
Új Csapat Létrehozása menüpont a fiók oldalon

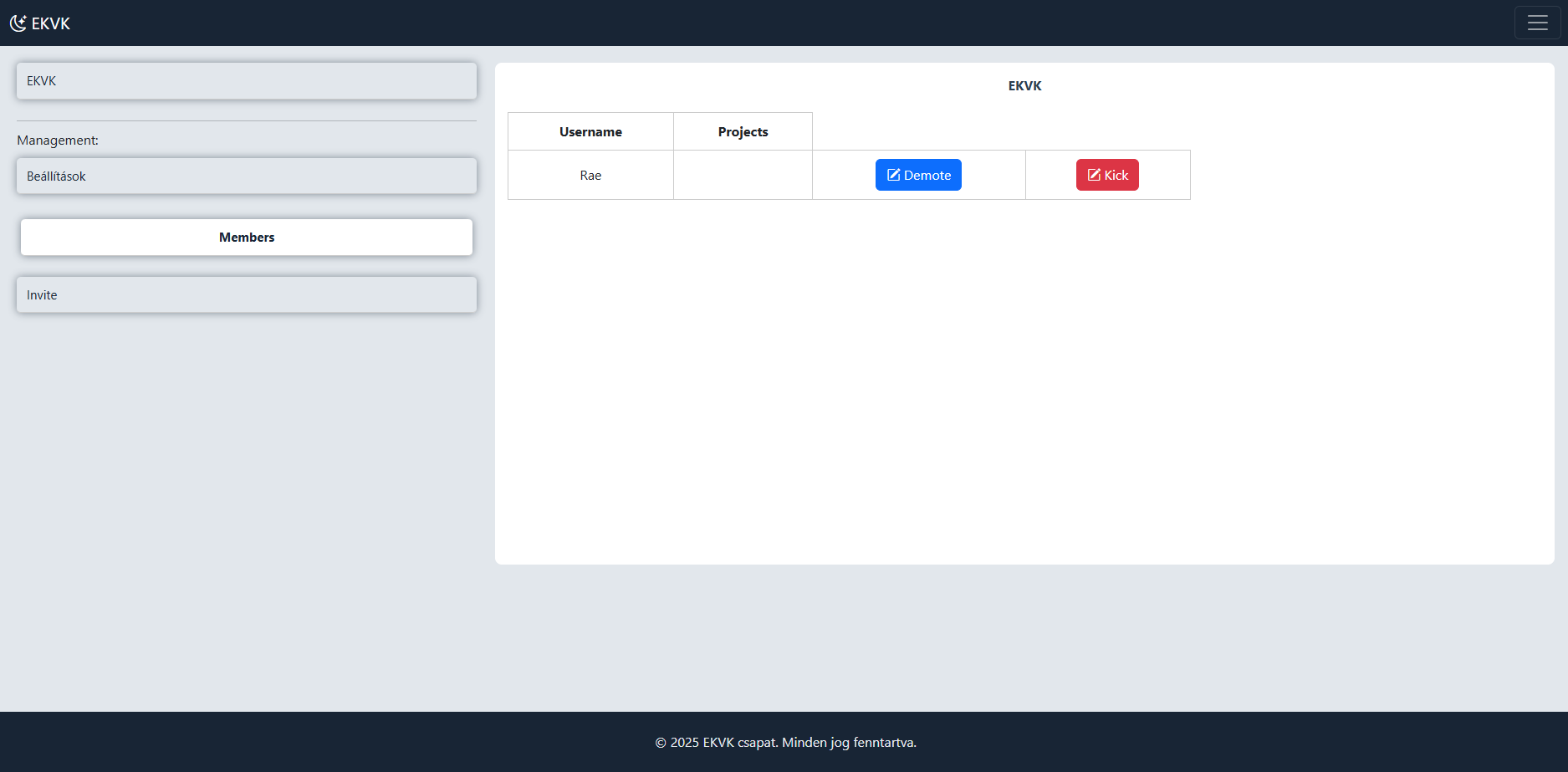
A Csapat Beállítások menüpont a fiók oldalon

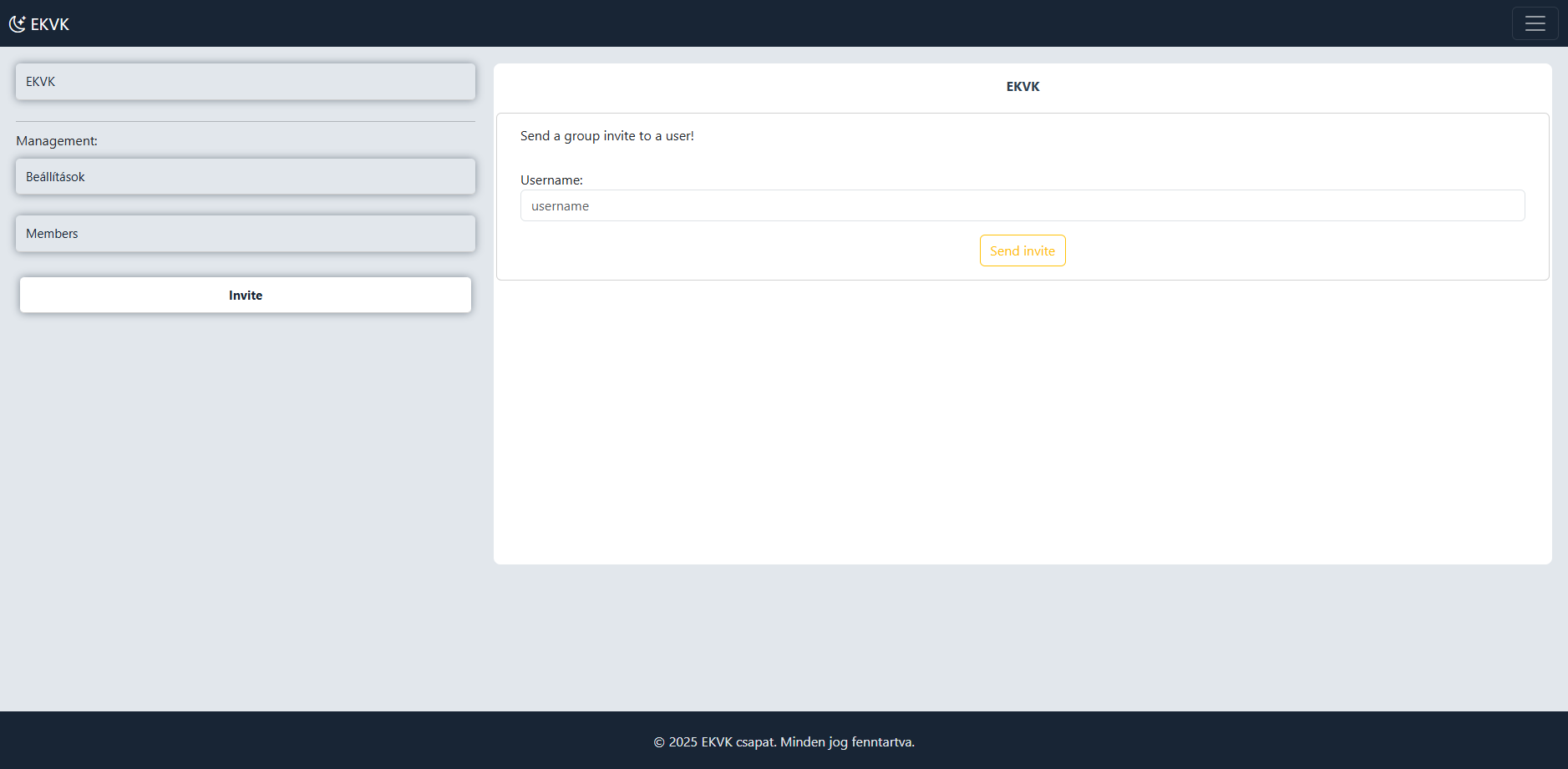
A Csapat oldalon tudják a felhasználók megnézni a csapat projektjeit és módosítani őket. (Ez a jelen verzióban még sajnos nincsen benne.) A csapat “Adminok”, akik a csapaton belül jogosultsággal rendelkeznek hozzáférnek a “Management” részhez, ahol

* A csapat beállításait (Egyelőre csak a csapat nevét) tudják módosítani a.
* A tagoknak tudnak jogosultságot adni vagy elvenni.
* Tagokat eltávolítani a csapatból.
* Tagokat meghívni a csapatba.

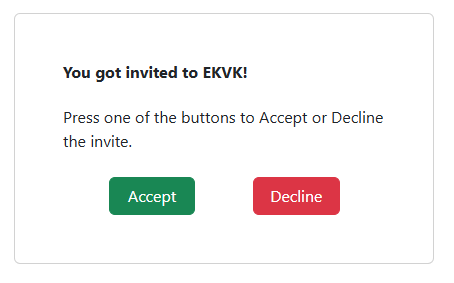
A csapat oldal főoldala

A csapat beállítások oldala

A csapattag beállítások oldala

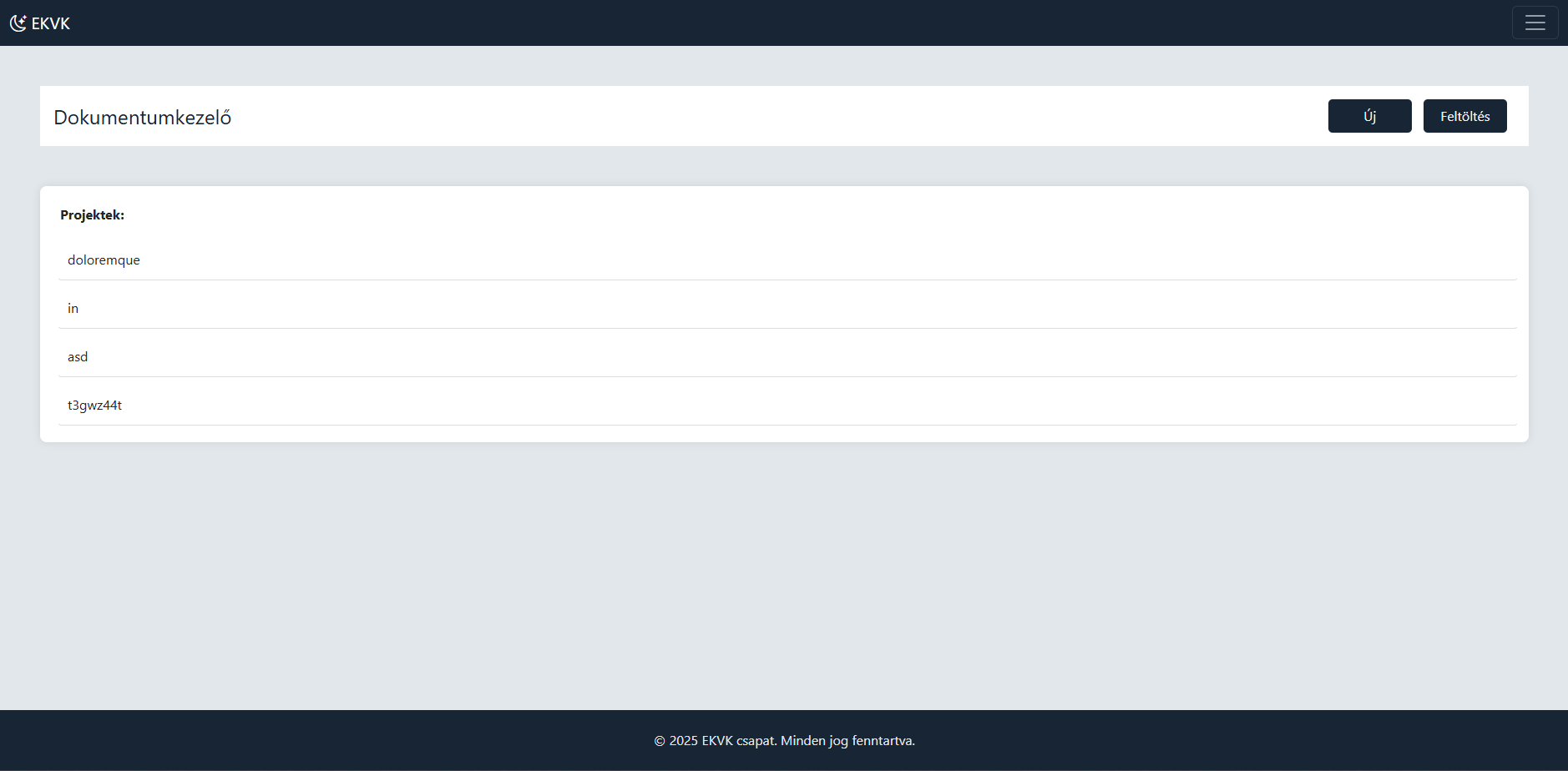
A csapat megivó oldala

Ha egy felhasználó meghívást kap egy csapatba, egy invite megjelenik a fiók oldalán meghívások részen, amit elfogadhat és csatlakozhat a csapathoz, amelyik meghívta, vagy elutasítja és törlődik a meghívó csatlakozás nélkül.

  
Az meghívás, amit a felhasználó kap.

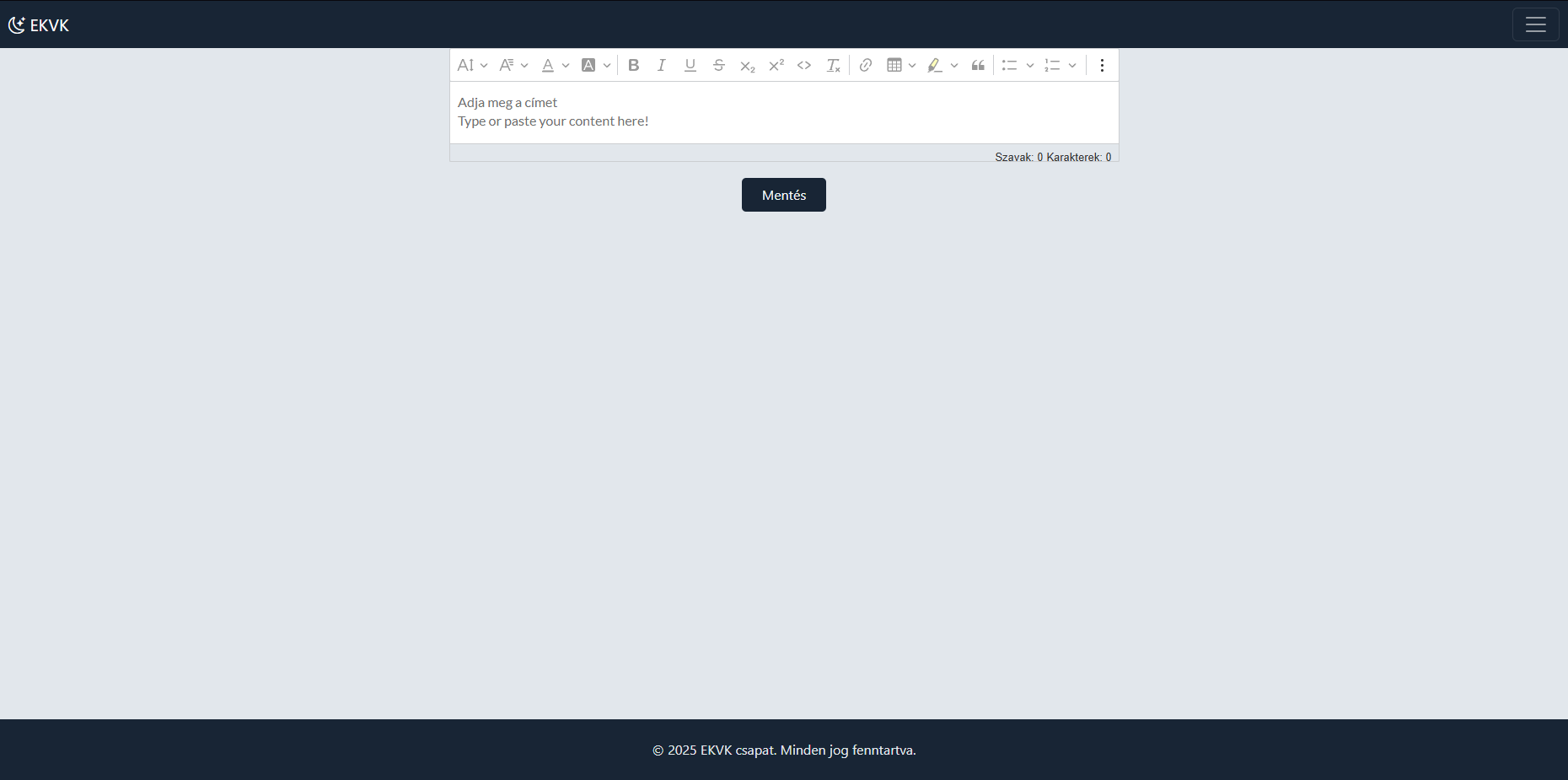
**Projektek:**  
Az oldalunk fő lényege egy böngésző alapú szöveg szerkesztő környezet, ahol felhasználók egyénileg, vagy csapatban tudnak dolgozni, ezeket a dokumentumokat hívjuk Projekteknek.   
Ezekhez a projektekhez a felhasználó akármikor hozzáférhet a navigációs menün a “Projektek” gombra kattintva. A projektek oldalon létrehozhat új projektet és folytathatja meglévő projektjeit a szövegszerkesztőben.

Projektek gomb a navigációs menün.

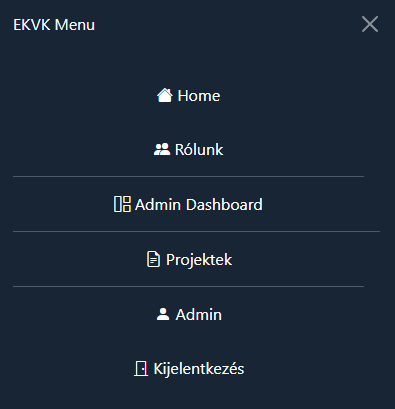
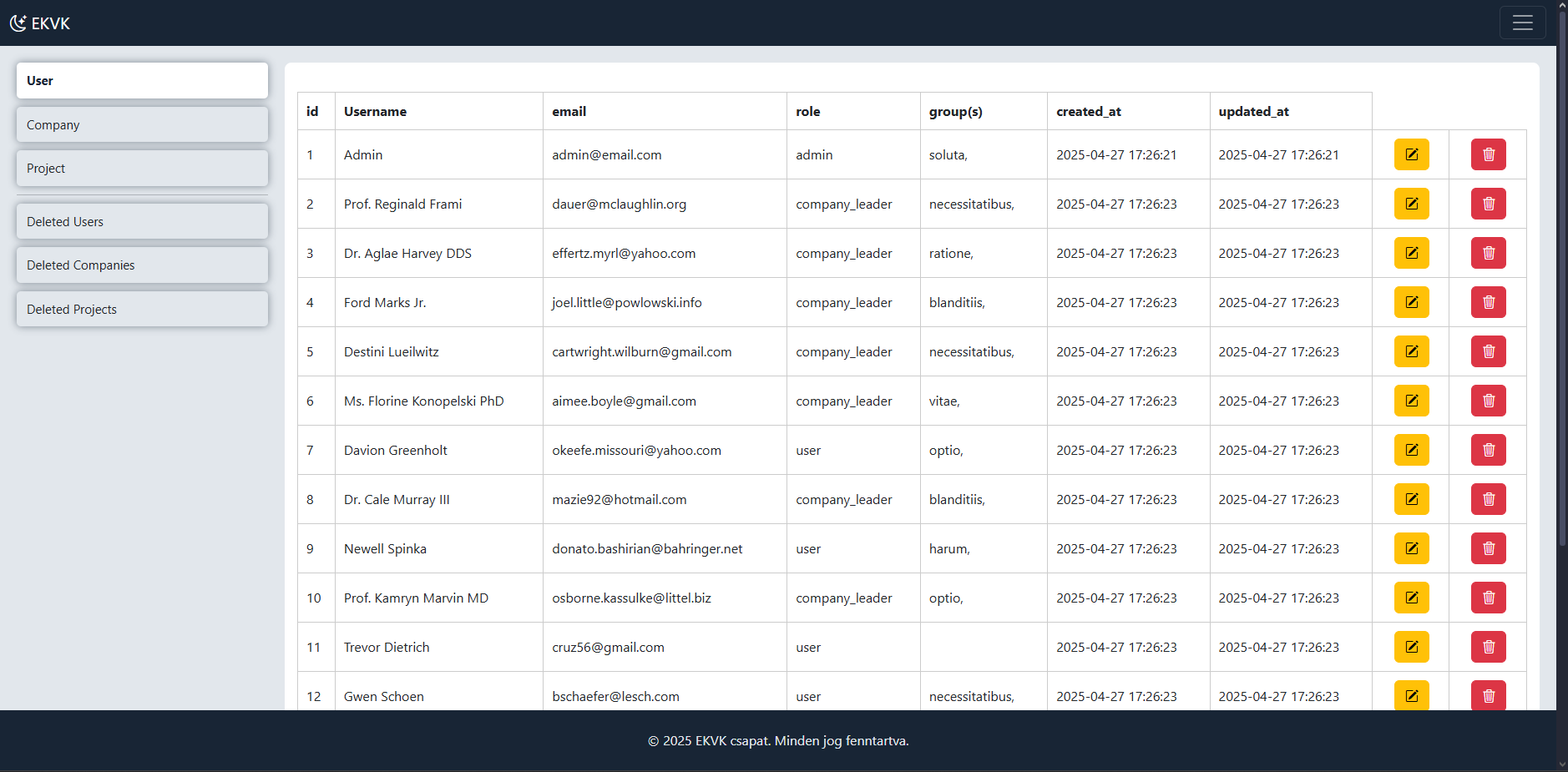
Projektek oldal.

**A szövegszerkesztő:**

A szövegszerkesztő 5 alapvető részből áll. Az első részben egy menüsort láthatunk, ahol ki tudjuk választani, hogy hogyan szeretnénk formázni a szövegünket. balról jobbra haladva tudunk állítani betűméretet, betűtípust, betűszínt, betű háttérszínt, félkövért, dőlt, aláhúzott, áthúzott, alsó index, felső index, forráskód hozzáadása, formázás eltávolítása, linkelés, táblázat beszúrása, kiemelést, idézet, pontozott lista, számozott lista és a további elemek gomb, ahol még tudunk tennivaló listát hozzáadni vagy a behúzást csökkenteni vagy növelni lehet. A második részben tudjuk megadni a szövegünk címét. Ezt nem lehet formázni. A harmadik részben tudjuk megadni a szövegünket, amit már formázni is lehet. A negyedik részben a szövegszerkesztő alatt láthatjuk, hogy összesen hány szót és hány karaktert írtunk eddig. Kifejezetten hasznos, ha egy esszét kell írnunk, ahol meg van adva, hogy hány szót vagy hány karaktert lehet maximum írni. Az ötödik rész pedig a mentés gomb. Ennek segítségével tudjuk eddigi munkánkat elmenteni. A gomb megnyomásával miután elmentett mindent visszakerülünk az előző oldalra és elkezdhetünk egy másikat, ha szeretnénk. Abban az esetben, ha egy meglévő dokumentumot nyitunk meg, amit így mentettünk el, akkor a rendszer ezt felismeri és feltölti a szövegmezőket úgy ahogyan abbahagytuk.

A szövegszerkesztő felület

**Admin felület:**   
Az oldalunkon a weboldal adminoknak van egy külön felülete, ahol tudják kezelni a felhasználókat, csapatokat, projekteket szükség esetén. Ezen az oldalon táblázatokba tölti az oldal az adatbázisból az adatokat, és az admin itt tud szerkeszteni, törölni és visszaállítani adatokat.

  
A navigációs menü admin jogosultságú felhasználóval.  
  
Admin oldal teszt adatokkal feltöltve.

**6. Program leírása (Fejlesztők számára)**  
  
A weboldal adabázisához több modelt is használunk:

* Companies – Groupokért felelős model.
* Invite - Meghivásokért felelős model.
* Projects - Projektekért felelős model.
* Role – Weboldal jogosultságokért felelős model.
* UserCompany – Userek és Companyk közötti összekötés.
* UserRole – Userek és Role-ok közötti összekötés.
* Users - Felhasználók közötti összekötés.

A weboldelünk controllerei:

**AccountController**  
  
Az AccountController a felhasználói fiókok kezeléséért felelős komponens a rendszerben. Elsődleges célja, hogy a bejelentkezett felhasználók elérjék és módosíthassák saját adataikat, kezelhessék a céges meghívóikat, illetve létrehozhassanak új cégeket a rendszerben. A controller kizárólag hitelesített felhasználók számára érhető el, így minden funkciója ellenőrzi a bejelentkezési állapotot.

A vezérlő egyik fő metódusa az index, amely betölti a bejelentkezett felhasználó adatait, valamint a hozzá tartozó meghívókat és az elérhető cégeket, majd ezeket továbbítja a megfelelő nézethez, ahol a felhasználó áttekintheti az aktuális állapotát. A meghívók kezelését az inviteHandler metódus végzi, amely lehetővé teszi a felhasználó számára, hogy egy meghívót elfogadjon vagy elutasítson. Elfogadás esetén a rendszer a felhasználót hozzárendeli a céghez, míg elutasítás esetén törli a meghívót.

A felhasználó saját adatainak módosítását az update metódus végzi, amely validált bemenet alapján frissíti a kapcsolódó adatokat. Emellett a storeCompany metódus segítségével a felhasználó új céget hozhat létre, amelyhez adminisztrátorként kerül hozzárendelésre. Amennyiben nem rendelkezik rendszerszintű admin jogosultsággal, szerepköre automatikusan módosul a megfelelő szintre.

Végül, a deactivate metódus lehetővé teszi, hogy a felhasználó deaktiválja saját fiókját. Ez a művelet kijelentkezteti a felhasználót, majd soft delete-et alkalmaz a fiókjára, így az a rendszerben nem jelenik meg többé, de az adatai megőrződnek.

**AdminController:**

Az AdminController a rendszer adminisztrációs felületéért felelős controller. Ez a vezérlő kizárólag olyan felhasználók számára érhető el, akik rendelkeznek adminisztrátori jogosultságokkal (azaz role\_id = 1). A controller egyetlen metódust tartalmaz, az index() függvényt, amely ellenőrzi, hogy a felhasználó be van-e jelentkezve, majd azt is, hogy admin jogosultságokkal rendelkezik-e.

Amennyiben mindkét feltétel teljesül, a metódus lekéri az összes felhasználót, céget és projektet az adatbázisból, beleértve a már törölt (soft deleted) rekordokat is. Ezeket a withTrashed() metódussal éri el, majd az adathalmazokat ID szerint rendezi. A lekért adatokat továbbadja az admin.index nézetnek, ahol az adminisztrátor áttekintheti és kezelheti ezeket az entitásokat.

Ha a bejelentkezett felhasználó nem rendelkezik admin jogosultsággal, a rendszer visszairányítja a főoldalra egy figyelmeztető üzenettel. Hasonlóan, ha a felhasználó nincs bejelentkezve, a rendszer nem engedi be az admin felületre, és szintén a kezdőlapra irányítja vissza egy megfelelő üzenettel.

**AuthController:**  
Az AuthController a rendszer felhasználóinak hitelesítéséért és regisztrációjáért felelős vezérlő. Három fő funkciót lát el: regisztráció, bejelentkezés és kijelentkezés. Célja, hogy biztonságos és validált módon kezelje a felhasználók belépését és a rendszerhez való csatlakozásukat.

A regisztrációs folyamatot a register metódus valósítja meg. Ez először validálja a beküldött űrlapadatokat, beleértve a felhasználónév, e-mail cím és jelszó mezőket. A jelszóra vonatkozó követelmények között szerepel minimum 8 karakter, kis- és nagybetűk használata, valamint számjegyek megléte. A validálás után létrejön egy új Users objektum, amelynek jelszava titkosítva (hashelve) kerül mentésre. A felhasználóhoz automatikusan egy alapértelmezett szerepkört (role\_id = 2) is hozzárendel a rendszer a UserRole modellen keresztül. A sikeres regisztráció után a felhasználó visszairányításra kerül a főoldalra egy megerősítő üzenettel.

A bejelentkezésért a login metódus felel, amely hasonló módon validálja az e-mail címet és a jelszót. Ezt követően megkeresi az e-mailhez tartozó felhasználót, és ellenőrzi, hogy a megadott jelszó megegyezik-e az adatbázisban tárolt hashelt jelszóval. Amennyiben az adatok érvényesek, a Auth::login függvénnyel belépteti a felhasználót, majd a főoldalra irányítja vissza. Hibás adat esetén visszatér az űrlaphoz hibaüzenettel.

A kijelentkezés a logout metóduson keresztül történik. Ez egyszerűen kijelentkezteti a felhasználót a rendszerből az Auth::logout függvény segítségével, majd visszairányítja a kezdőlapra.

**CompaniesController** (Admin oldal segéd controller):  
A CompaniesController egy controller, amely a csapatok erőforrásainak kezelésére szolgál a rendszerben. Ez a controller főként RESTful módszerek szerint épül fel, és lehetővé teszi csapatok létrehozását, szerkesztését, frissítését, törlését és helyreállítását.

A vezérlő az index, create és show metódusokat is tartalmazza, de ezek jelenleg nem implementáltak, csak helykitöltőként szerepelnek. A tényleges működés az alábbi metódusokban történik:

A store metódus új csapat létrehozását végzi el. Ez a művelet validált adatok alapján történik, amelyeket a StoreCompaniesRequest osztály ellenőriz. A mentés során a rendszer elmenti az új csapat nevét az adatbázisba.

Az edit metódus egy meglévő csapat szerkesztésére szolgáló nézetet ad vissza, amely tartalmazza a kiválasztott csapat adatait. Ezt követően az update metódus frissíti az adott csapat adatait a felhasználó által megadott új értékekkel, amelyek szintén validálva vannak az UpdateCompaniesRequest osztályon keresztül. Sikeres frissítés után a felhasználó visszairányításra kerül az admin oldalra.

A destroy metódus a kiválasztott csapat „soft delete” típusú törlését végzi el, azaz az adatbázisban a csapat inaktív státuszba kerül, de nem véglegesen törlődik. A felhasználó visszajelzést kap a törlés sikerességéről.

A restore metódus lehetővé teszi egy korábban törölt csapat helyreállítását. Ez a withTrashed() metódussal éri el az archivált csapatokat, és a restore() hívással újra aktívvá teszi az adott csapatot. A visszairányítás során egy üzenetet is megjelenít, amely megerősíti a helyreállítás sikerességét.

**CompanyPageController :**  
A CompanyPageController osztály kezeli a csapatokkal kapcsolatos különböző műveleteket a felhasználó oldalon. Az index funkció megjeleníti a csapatot és a hozzá tartozó felhasználókat. Az invite lehetővé teszi, hogy egy felhasználót meghívjunk egy csapathoz, ha az nem saját magunk és a felhasználó létezik. Az edit és update funkciók a csapat nevét szerkesztik és frissítik. A destroy törli a csapatot. A promote és demote lehetővé teszik, hogy felhasználókat adminisztrátorrá emeljünk vagy visszaminősítsünk. Végül a kick eltávolít egy felhasználót a csapatból, ha az nem önmaga. Minden művelet után visszairányítja a felhasználót, és megfelelő üzeneteket küld a művelet eredményéről.

**ProjectPageController:**  
 Projektoldal megjelenítéséért felelős controller.

**ProjectsController**  (Admin oldal segéd controller) –   
Projektek admin oldali kezeléséért felelős controller.

**UsersController (**Admin oldal segéd controller)   
Felhasználók admin oldali kezeléséért felelős controller.

Az editor oldalhoz segítségül hívtuk a CKeditor-t. Ez egy olyan program/oldal, ahol egy regisztrálás után könnyebben kezelhető szövegszerkesztőt tudunk csinálni, mint akár a word. Ezt 3 részre tudjuk bontani. Az első maga a webes oldal. Miután beillesztettük a saját elemeinket, a content részen belül létre kell hozni a div-eket. Ezeknek olyan class nevet kell adni, amiket kiválasztottunk a CKeditor oldalon, hogy melyik funkcióit szeretnénk használni. Látható maga a main-container, benne lévő classokkal megadjuk miket szeretnénk használni, utánna különszedjük 2 részre. Az első rész maga az editor. Ide kerülnek a gombok, amikkel a szöveget tudjuk formázni és maga a 2 szövegtároló (h1 és p) is ide kerülnek a js álltal. A második részbe pedig megy a szöveg és a karakter számláló div. Ezután látható egy sima container ami a gombot tárolja, hogy el tudjuk menteni a munkánkat. Itt található gomb egy js-hez van hozzákötve, ami arra szólgál, hogy megkeresi a h1 és p tagokat, ezekből kiszed innerHTML-lel mindent, (erre azért van szügség, mert így a p-n belüli módosításokat is megkapjuk pl.: strong vagy esetleg css módosítások), majd ezt összerakjuk egy JSON file-ban és a JSON-t elküldjük a form-on belüli láthatatlan inputnak, ami ezt eltárolja. Ezek után az EditorController.php segítségével feltőltjük az adatbázisban lévő "project\_data" részt. Amikor belépünk az editorba, az input autómatikusan megkapja a project id-ját és a "project\_data"-t így mindig onnan tudjuk folytatni, ahol abbahagytuk. Ezt úgy értük el, hogy egy window.onload funkióval megvárjuk, míg betőlt az oldal, lekérjük az inputban lévő value-t és feltőltjük az a megfelelő helyére.

A következő rész az editor.js file-hoz kell mennünk. Itt kerjük le a Ckeditor funkcióit egy felhőn keresztűl. Először el kell őket nevezni, majd egy License key kell az alap működéshez. Majd ismét megadjuk a funciókat egy JSON file-ba, a plugin-okat, a betűtípusokat és betűméreteket, képek lehetőségét, az alap adatokat (ilyenek a nyelv, a license key, hogyan szeretnénk használni, hogy letőltjük vagy esetleg felővel dolgozunk), megadjuk a listák tulajdonságát, megadjuk az alap szöveget és a táblázat formázási lehetőségeit. Ezek után lekérjük magát az editor a felhőről és azzal visszatérünk a main-container-be.

A harmadik része pedig a css, ahol megadjuk, hogy hogyan nézzen ki az alap. Legyen kerete, alap betűtípus, színek.