1. **Csoport tagjai: Boros Dániel, Dékány Csaba**
2. **Ki melyik feladatrészt dolgozza ki?**

Regisztrációs és bejelentkezési felület (frontend). Boros

Backend része és adatbázis: Dékány

Elfelejtett jelszó (frontend, backend, adatbázis): Dékány

Saját profil módosítása (frontend, backend, adatbázis): Boros

Email-ek és super user generálása: Dékány

Email template-ek: Boros, Dékány

Felhasználók kezelése (frontend): Dékány

Felhasználók kezelése (backend, adatbázis): Boros

Állatfajok és típusok kezelése (frontend): Boros

Állatfajok és típusok kezelése (backend, adatbázis): Dékány

Egység, integrációs end to end tesztek: Boros, Dékány

Autentikáció (backend): Dékány

Betegek kezelése (backend): Boros

Betegek kezelése (frontend): Dékány

Kezelések kezelése (frontend): Dékány

Kezelések kezelése (backend): Boros

Folyamatban lévő aktivitások:

Prezentáció készítés: Dékány, Boros

Végső finomhangolások

Dokumentáció végső simításai

1. **Vizsgaremek címe:**

**Vet(erinary)CMS**

1. **Vizsgaremek témája:**

Állatorvosi rendelő adatainak kezelése

1. **Témaválasztás indoklása. (okok leírása)**

#### ****1. Digitalizáció hiánya az állatorvosi szektorban****

* Sok állatorvosi rendelő még mindig papíralapú vagy elavult szoftvereket használ.
* A digitális átállás jelentősen növelheti a hatékonyságot, csökkentheti az adminisztrációs terheket és javíthatja az ügyfélkezelést.
* Egy modern, könnyen használható CMS megoldás segítheti az állatorvosokat a napi munkájuk során.

#### ****2. Egyedi és kevésbé telített piac****

* Bár léteznek állatorvosi menedzsmentrendszerek, ezek sokszor drágák, nehezen testreszabhatók vagy nem felelnek meg minden rendelő igényeinek.
* Egy saját fejlesztésű rendszerrel lehetőség van **moduláris és rugalmas megoldást** létrehozni, amelyet a felhasználók egyedi igényeik szerint alakíthatnak.

#### ****3. Valós piaci igényekre reflektál****

* Az állatorvosok gyakran küzdenek az időpontkezeléssel, betegnyilvántartással és készletgazdálkodással.
* A rendszer célja, hogy **automatizált értesítésekkel, könnyen kezelhető betegprofilokkal és egyértelmű dokumentációval** segítse őket.
* A gazdik számára is előnyös, hiszen könnyebben nyomon követhetik háziállataik egészségügyi előzményeit.

#### ****4. Fejlesztési és technológiai kihívások****

* A projekt egy komplex fejlesztési folyamatot igényel, amely lehetőséget biztosít a különböző technológiák és fejlesztési módszerek alkalmazására.
* A Blazor WASM frontend, a .NET backend és az AWS alapú adatbázis **modern, skálázható és hosszú távon fenntartható** megoldást biztosít.

#### ****5. Társadalmi és etikai szerep****

* Az állatorvosi ellátás fejlesztése hozzájárulhat az állatok jobb egészségügyi ellátásához.
* Egy hatékonyabb rendszer csökkentheti a rendelők leterheltségét, ezáltal javíthatja az állatorvosok és a páciensek közötti kommunikációt.

A **Veterinary CMS** témakörének választása egy logikus és releváns döntés volt, amely valós problémákra nyújt megoldást, kihasználja a modern technológiák adta lehetőségeket, és egy olyan területen hoz innovációt, amely még nem teljesen digitalizált.

1. **Célközönség:**

A **Veterinary CMS** célközönsége elsősorban azok az állatorvosi rendelők, klinikák és szakemberek, akik hatékonyabb, digitalizált megoldást keresnek a praxisuk működtetésére. Az alábbi csoportok számára különösen előnyös:

#### ****1. Állatorvosi rendelők és klinikák****

* Kisállat- és nagyállat-rendelők egyaránt használhatják.

#### ****2. Független állatorvosok és mobil praxisok****

* Azoknak az állatorvosoknak, akik házhoz mennek, szükségük lehet egy **felhőalapú rendszerre**, amelyet bárhonnan elérhetnek.

#### ****3. Állatkórházak és sürgősségi központok****

* Könnyen integrálható laboratóriumi rendszerekkel és diagnosztikai eszközökkel.

#### ****4. Állatmenhelyek és rehabilitációs központok****

#### ****5. Oktatási és kutatóintézetek****

* Állatorvosi egyetemek és kutatólaborok számára hasznos lehet a betegadatok és esettanulmányok rögzítésére.
* Oktatási célra demonstrálhatók benne a kezelések és diagnosztikai folyamatok.

#### ****6. Állatorvosi gyógyszertárak és rendelők készletkezelése****

1. **Funkciók leírása:**

<https://github.com/Vizsgaremek-borosd-dekanycs/dev-docs/tree/main/Writerside/topics>

1. **A projekt fejlesztő környezete**
   1. Hardvere:

Fejlesztéshez:

1. **Processzor**:
   * Minimális: Kétmagos processzor
   * Ajánlott: Négymagos processzor vagy jobb
2. **Memória**:
   * Minimális: 8 GB RAM
   * Ajánlott: 16 GB RAM vagy több
3. **Tárhely**:
   * Minimális: 20 GB szabad lemezterület
   * Ajánlott: SSD, 50 GB szabad lemezterülettel
4. **Operációs rendszer**:
   * Windows 10 1909-es verzió vagy újabb, macOS 10.15 vagy újabb, vagy egy friss Linux disztribúció
5. **IDE**:
   * Visual Studio 2022 (17.0 vagy újabb verzió), Visual Studio Code, vagy JetBrains Rider
6. **.NET SDK**:
   * .NET 6.0 SDK
   1. Fejlesztés során használt szoftverek (AB, frontend, backend):

Microsoft Visual Studio (.NET C#, Blazor), SQL lite (később amazon webszerver), Dokumentáció: WriterSide

* 1. Ezek használatának indoklása (Miért hogy miért azt a programnyelvet, és azt a fejlesztőkörnyezetet választottam, amit):

A projektben a **C# programnyelvet** és a **Visual Studio** fejlesztőkörnyezetet választottuk, mert:

* **C#** egy modern, objektumorientált nyelv, amely jól skálázható és könnyen karbantartható kódot tesz lehetővé.
* **Visual Studio** egy teljeskörű fejlesztőkörnyezet, amely beépített támogatást nyújt a C#-hoz, gyors hibakeresési lehetőségeket kínál, és számos hasznos fejlesztői eszközt biztosít.
* A .NET keretrendszernek köszönhetően könnyedén tudunk backend API-kat készíteni és más Microsoft-technológiákkal integrálni a rendszert.

Az adatbáziskezelésnél egyelőre **SQLite**-ot használunk, mert:

* Egyszerűen kezelhető és nincs szükség külön szerverre az üzemeltetéséhez.
* Kisebb adatmennyiségek és lokális fejlesztési fázisok során gyors és hatékony megoldást nyújt.
* Könnyen migrálható később nagyobb rendszerekre.

A későbbiekben **Amazon Web Services (AWS)** adatbázisra térünk át, mert:

* AWS RDS (Relational Database Service) skálázhatóbb és nagyobb adatmennyiségeket is könnyen kezel.
* Felhőalapú megoldásként biztosítja az automatikus biztonsági mentéseket, magas rendelkezésre állást és teljesítmény-optimalizálást.
* Könnyebb integrációt biztosít más AWS szolgáltatásokkal, ha a projekt később bővül.

1. **Miért volt szükség a program elkészítésére, és miben más, mint más hasonló létező program**

A **veterinary CMS (Content Management System)** létrehozására azért volt szükség, mert a meglévő állatorvosi szoftverek gyakran nem felelnek meg teljes mértékben a modern klinikák igényeinek. Sok rendszer elavult, bonyolult kezelőfelülettel rendelkezik, vagy nem biztosít elegendő testreszabhatóságot a különböző rendelők számára.

### Miért volt szükség egy új rendszerre?

1. **Digitalizáció és hatékonyság**
   * Az állatorvosi klinikák egyre inkább digitális megoldásokra támaszkodnak az adminisztráció és az ügyfélkezelés terén.
   * Egy modern CMS segít az időpontfoglalások, kezelési előzmények és számlázások egyszerűbb kezelésében.
2. **Testreszabható és rugalmas működés**
   * A meglévő szoftverek sokszor túl általánosak, míg egy egyedi rendszer pontosan az adott klinika igényeire szabható.
   * Moduláris felépítésével lehetőség van specifikus funkciók hozzáadására, például gyógyszerkészlet-kezelés, állatprofilok vagy speciális jelentések generálása.
3. **Felhasználóbarát felület**
   * Az állatorvosok és asszisztensek számára egy gyorsan tanulható, intuitív felület elengedhetetlen a hatékony munkavégzéshez.
   * Az egyszerűbb kezelhetőség csökkenti az adminisztratív terheket, így több idő marad a betegek ellátására.

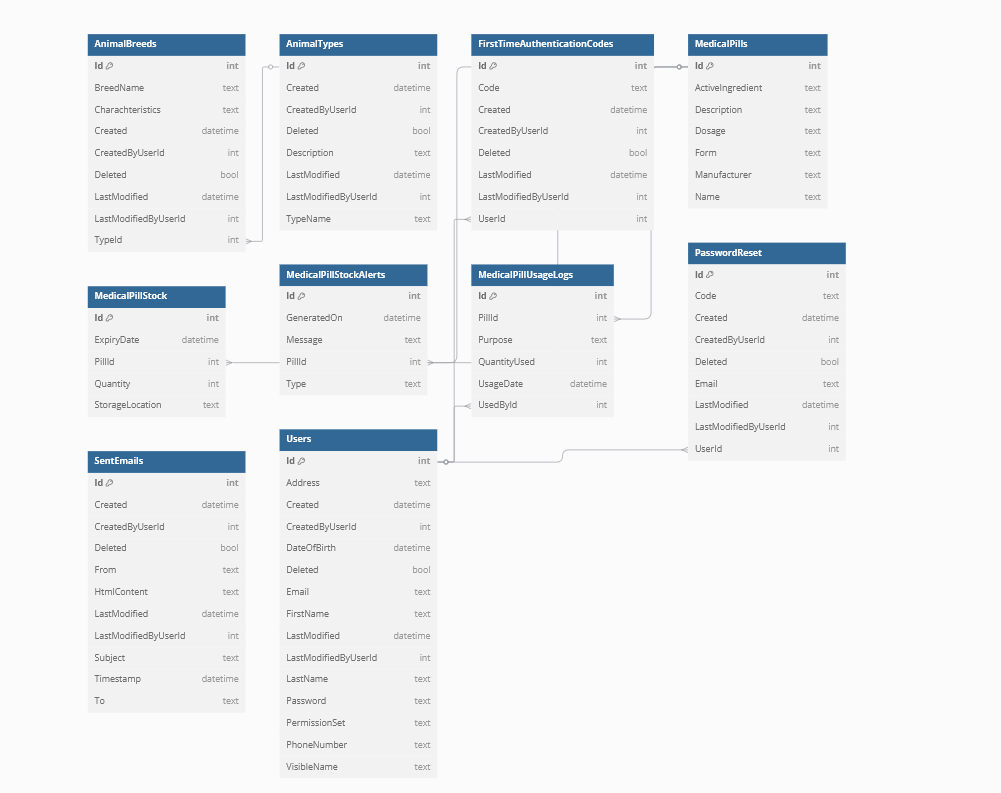
### Miben más ez a program a többihez képest?

* **Felhőalapú megoldás**: A rendszer bármilyen eszközről elérhető, így az állatorvosok akár távolról is hozzáférhetnek a páciensek adataihoz.
* **Integrációk és automatizáció**: Beépített SMS/email értesítések az ügyfeleknek, gyógyszerek nyilvántartása és automatikus időpont-emlékeztetők.
* **Biztonság és adatvédelem**: Modern titkosítási és adatvédelmi eljárások biztosítják a páciensek és tulajdonosaik adatainak védelmét.
* **Skálázhatóság**: Kisebb rendelők és nagyobb klinikák egyaránt használhatják, az igényeikhez igazítva.

1. **A program használatához milyen hardverre és szoftverre van szükség**

### Klienst gép (Blazor WASM alkalmazások futtatásához):

1. **Processzor**:
   * Minimális: Kétmagos processzor
   * Ajánlott: Négymagos processzor vagy jobb
2. **Memória**:
   * Minimális: 4 GB RAM
   * Ajánlott: 8 GB RAM vagy több
3. **Böngésző**:
   * A legújabb verziójú modern böngészők, mint a Microsoft Edge, Google Chrome, Mozilla Firefox vagy Safari
4. **Mutasd be projekthez tervezett adatbázis tábláit, tulajdonságait és kapcsolatait. Írd le a felépítést.**

**Későbbiekben a további funkciók miatt bővülni fog a diagram **

1. **Hogyan lehet letölteni, telepíteni, elindítani a programot.**

### **Hogyan lehet letölteni, telepíteni és elindítani a programot?**

#### ****1. Letöltés****

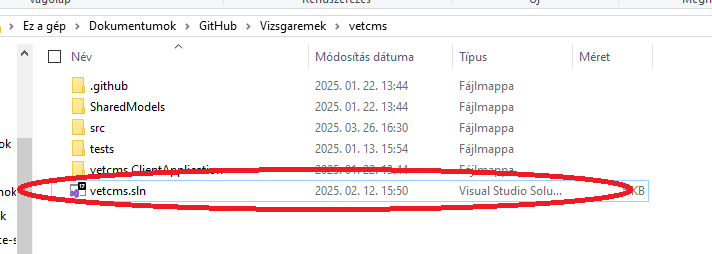
A programot a következő módokon lehet letölteni:

* **GitHub/Tárolóból**: Ha a projekt elérhető egy GitHub vagy más verziókezelő rendszerben, akkor a következő parancs segítségével klónozható:
* git clone https://github.com/felhasznalonev/projekt-neve.git
* **Telepítőfájlból**: Ha rendelkezésre áll egy telepítő (pl. .exe vagy .msi Windowsra, .pkg macOS-re), akkor a fejlesztői dokumentációban megadott forrásból lehet letölteni.

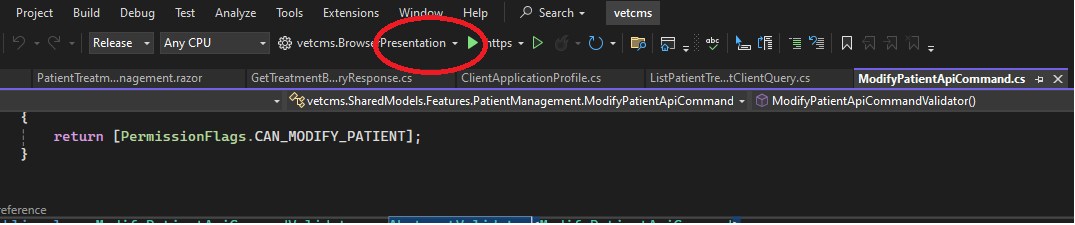
#### ****2. Telepítés****

**Fejlesztői gépre (ha a forráskódból futtatjuk)**

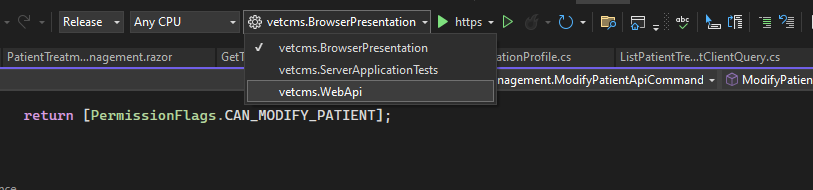
1. Győződjünk meg róla, hogy a következő szoftverek telepítve vannak:
   * **Visual Studio 2022 (vagy VS Code/Rider)**
   * **.NET 6.0 SDK**
   * **Git** (ha verziókezelőből töltjük le)
   * **SQLite vagy egyéb adatbáziskezelő** (ha szükséges)
2. Nyissuk meg a projektet a Visual Studio 2022 fejlesztői környezetben.
3. Nyissa meg az alábbi kép alapján



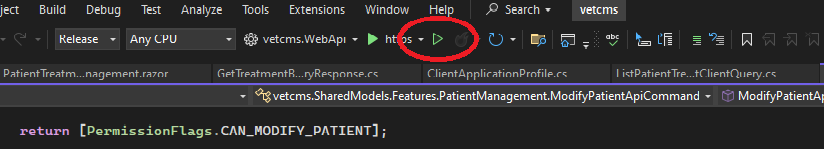
1. A fogaskerék mellei nyílra (a zöld gombtól balra lévő) kattintson, hogy lenyíljon a fül



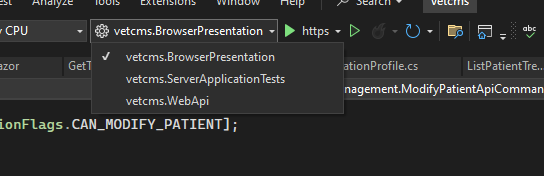
1. Válassza ki a „vetcms.WebApi” lehetőséget



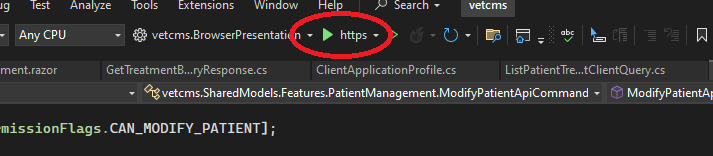
1. Kiválasztás után a halvány zöld gombot nyomja meg

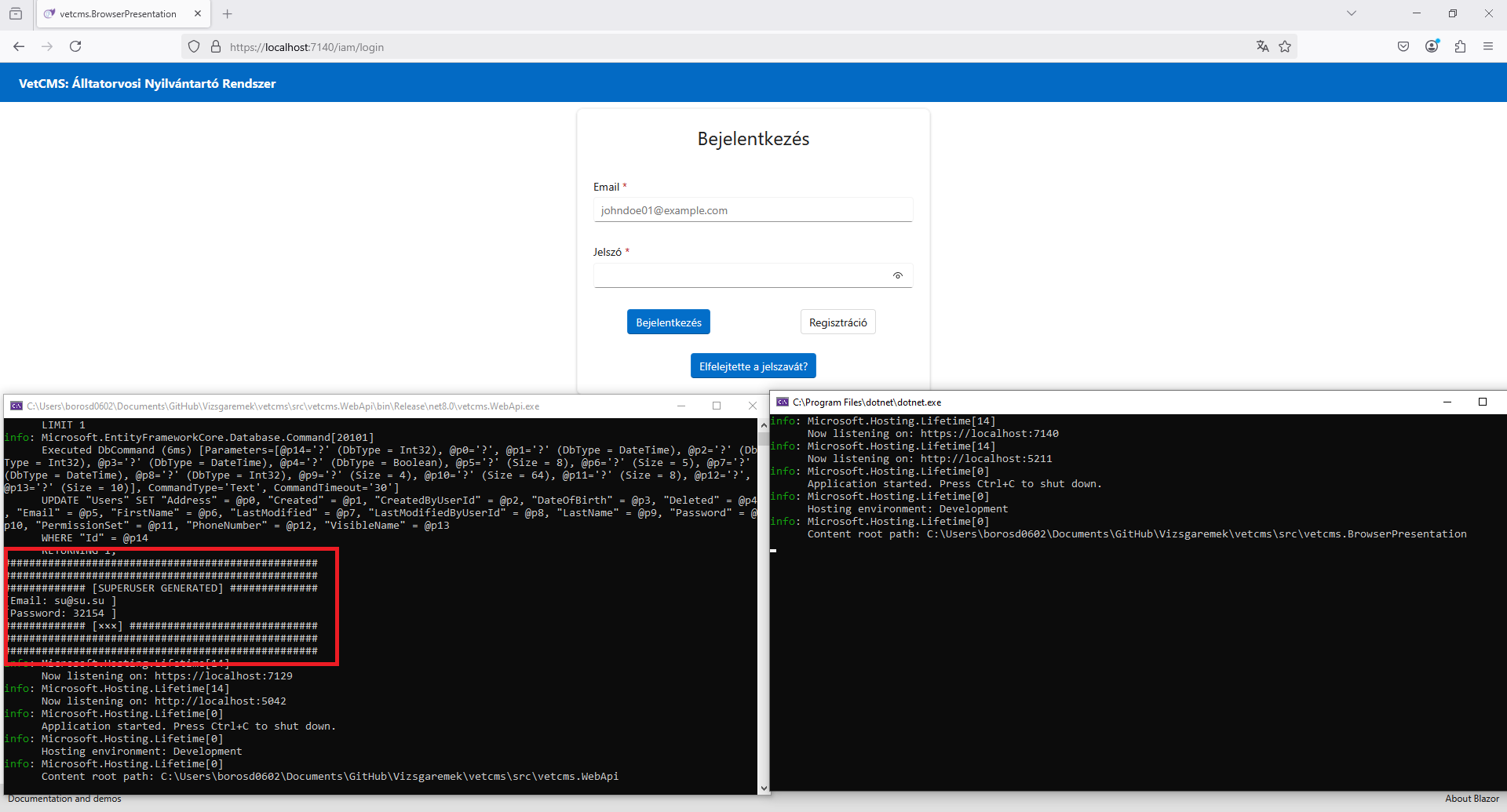


1. Miután megnyílt a CMD ablak és a keresőbe a „Swagger” az előző metódus alapján válassza ki a „vetcms.BrowserPresentation” lehetőséget



1. Ebben az esetbe a teli zöld gombbal futassa a programot



1. Futtatás után ezt kell lássuk. Látható, hogy a program automatikusan generál egy felhasználót, amivel minden funkció elérhető.
2. 

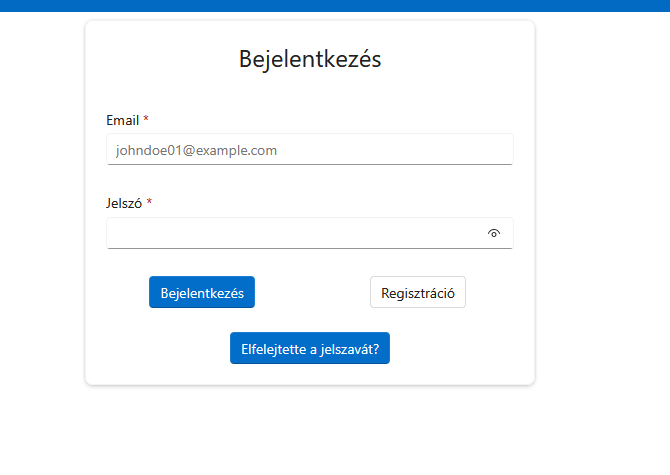
**Felhasználói (klienseknél történő) „telepítés”**

1. Ha az alkalmazás egy **Blazor WASM kliensalkalmazás**, akkor egyszerűen egy modern böngészőben kell megnyitni a megadott webcímet.**3. Program indítása**

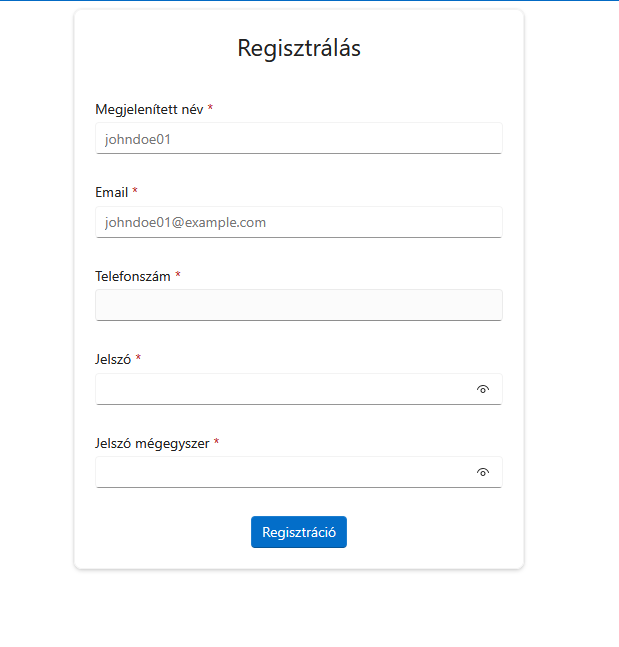
* **Fejlesztői környezetből**:
  + Visual Studio esetén a **"Start"** gombra kattintva vagy a dotnet run parancs kiadásával.
* **Felhasználói kliensnél**:
  + Böngészőben megnyitva a webalkalmazás URL-jét.

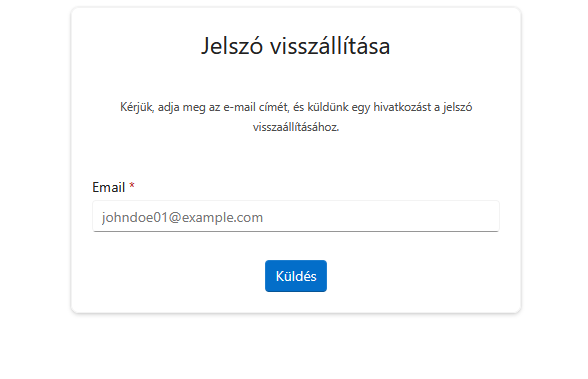
1. **Ismertesd a program használatát, szemléltesd képernyőképekkel**

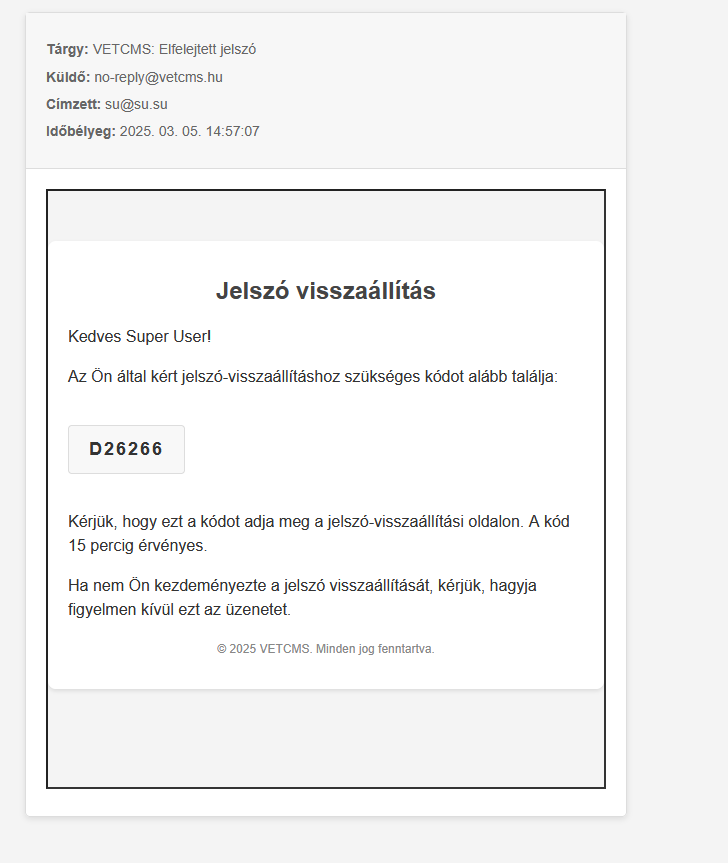
Bejelentkezési oldal. Ez a kezdőlap fogadja a felhasználót ha rákeres az oldalra. Ha van profilja a felhasználónak be tud itt lépni. Látható a regisztráció és elfelejtett jelszó.

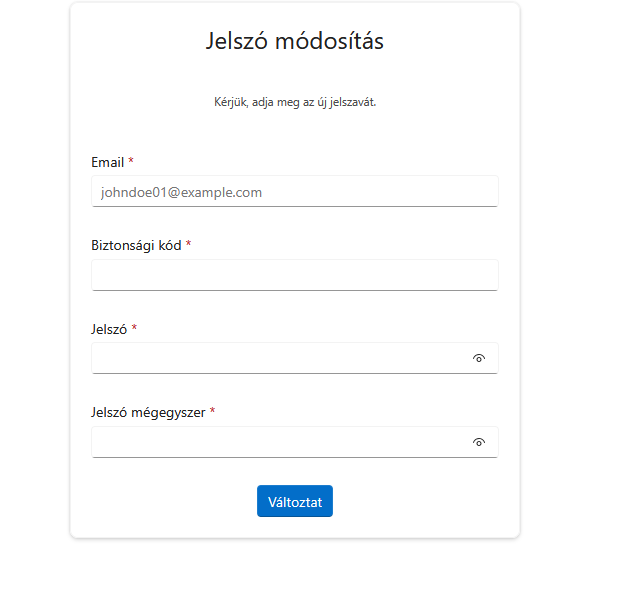


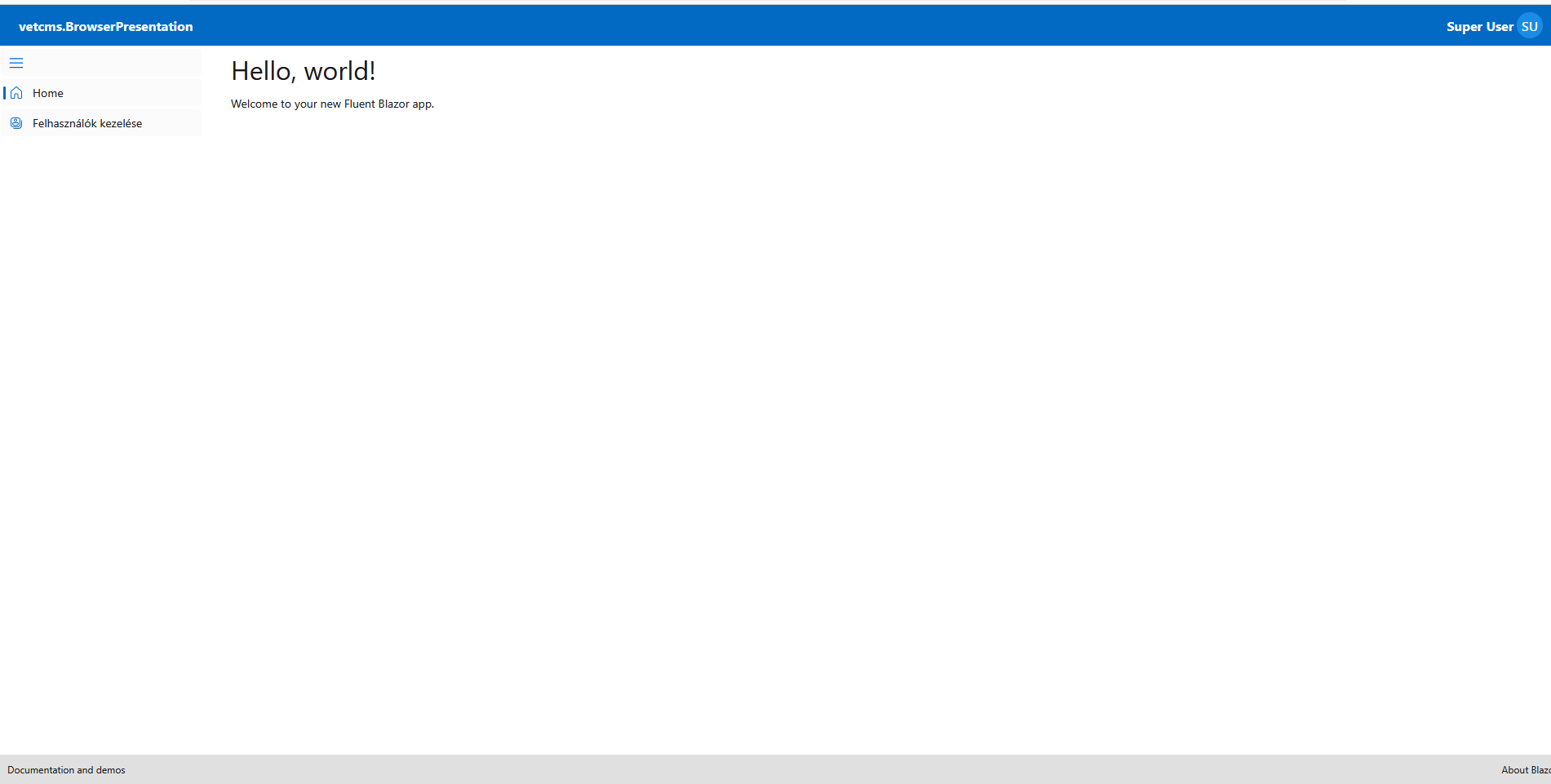
Regisztrációs felület. Itt lehet megadni az alapvető adatokat.

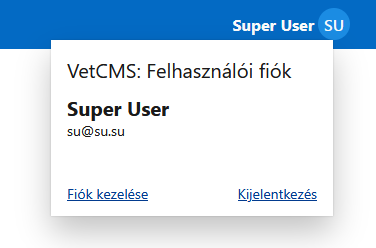


Ha elfelejtette a felhasználó a jelszavát, akkor az email címe magadásával egy újat meg tud adni. Mivel az email elküldése technikai okok miatt nem még lehetséges, ezért egy template-et kap a felhasználó, hogyan nézne ki az email, amit kapna. Ezután átirányításra kerül a felhasználó az oldalra, ahol meg tudja adni az új jelszavát.

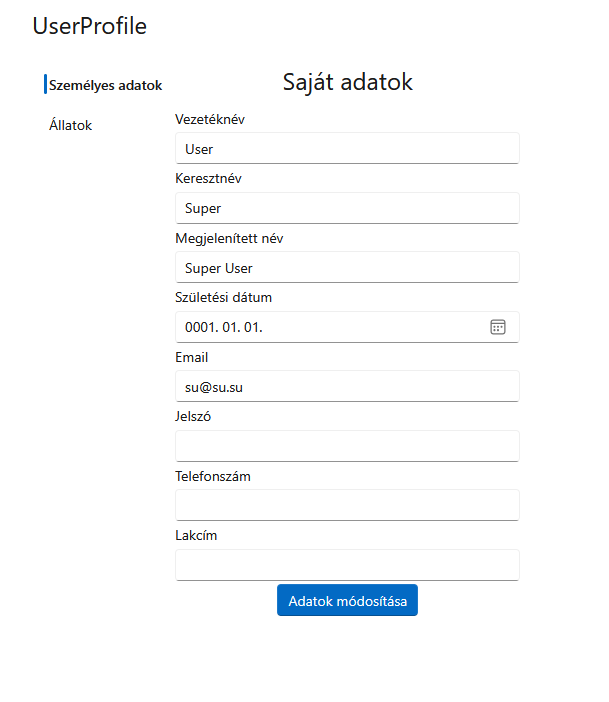
Email kinézete.

Új jelszó megadása

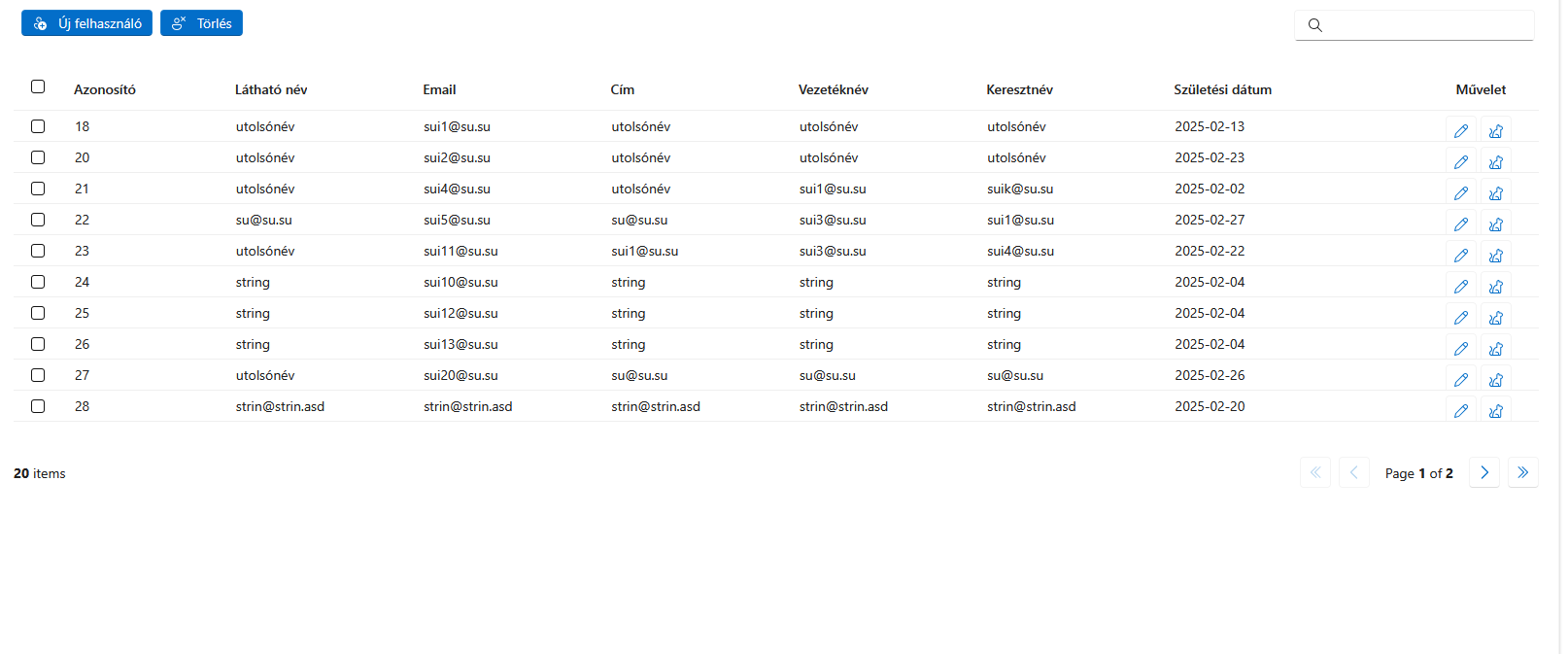
Egyenlőre ez az alap képernyő, ha belép a felhasználó. Baloldalt látható a navigációs sáv. A felhasználó azokat az oldalakat látja, amikhez van hozzáférési joga. (Ezekről és a adatok validációjáról leírás a fejlesztői dokumentációban Autentikáció és autorizáció fülben található)

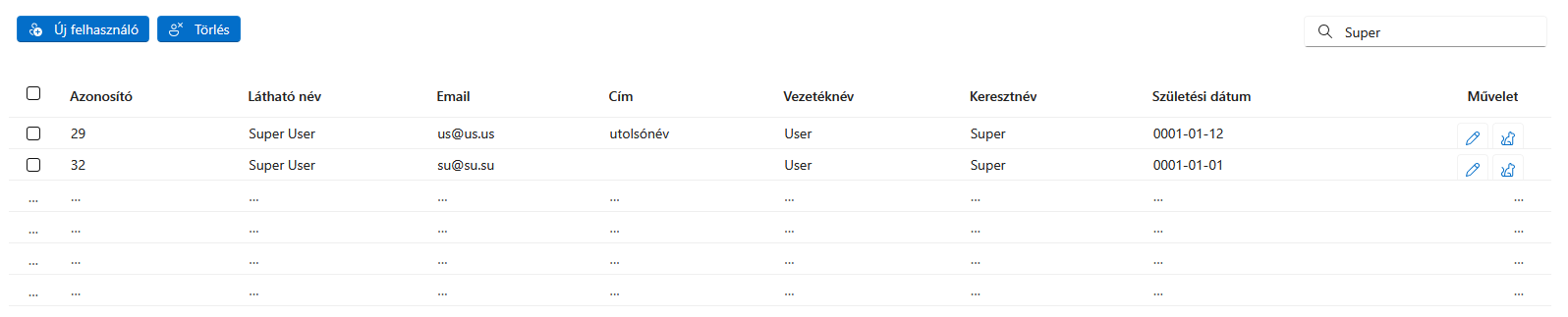


Jobb felső sarokban található a felhasználó. Kijelentkezni lehet itt és a saját profilt lehet itt módosítani.

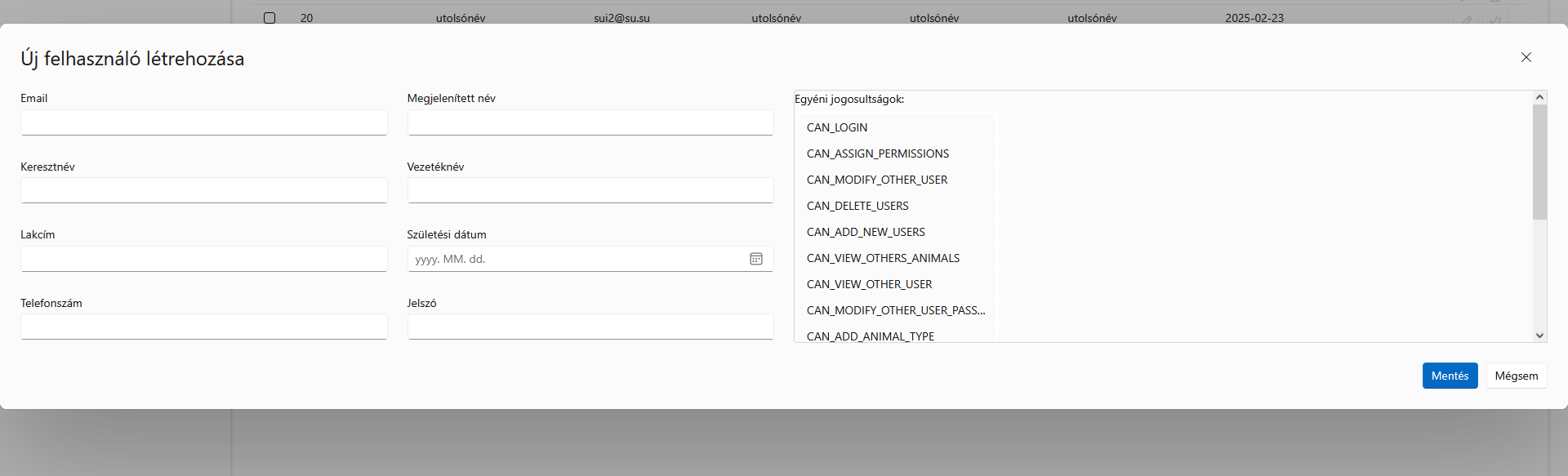


Itt találhatóak a felhasználói adatok. Mivel ez az admin profilja, ezért hiányos, viszont alap esetben tejlesen ki lenne töltve. Látható baloldalt két fül, az „Állatok”-ban lesznek később láthatóak a felhasználó állatainak lapjai és korábbi kezeléseik.

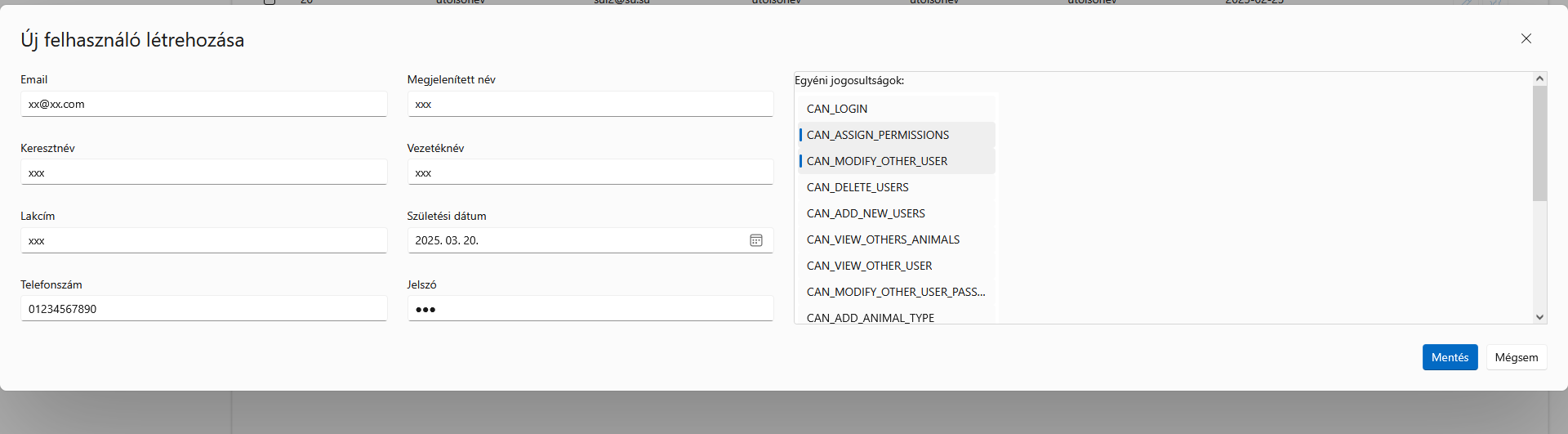
A fő navigációs menün rányomva a „Felhasználók kezelése” fülre ezt az oldalt kapjuk. Ha van megfelelő hozzáférése a felhasználónak tud „új felhasználó”-t létrehozni, törölni és módosítani. Lehetséges keresni a felhasználók között. Későbbiekben hozzáférés is lehet arra, hogy a felhasználó állatait megnézhesse a felhasználó és új kezelést hozzáadjon, felvegyen állatot vagy módosítsa azt.

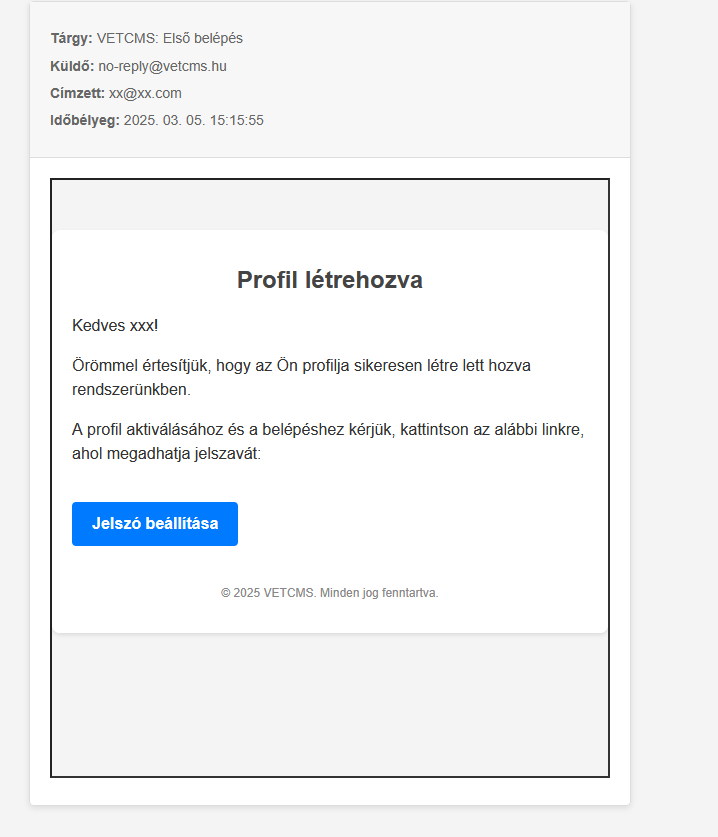


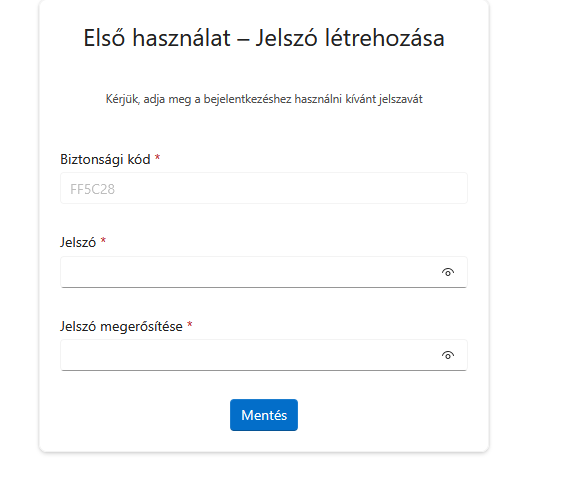
Keresésre példa.



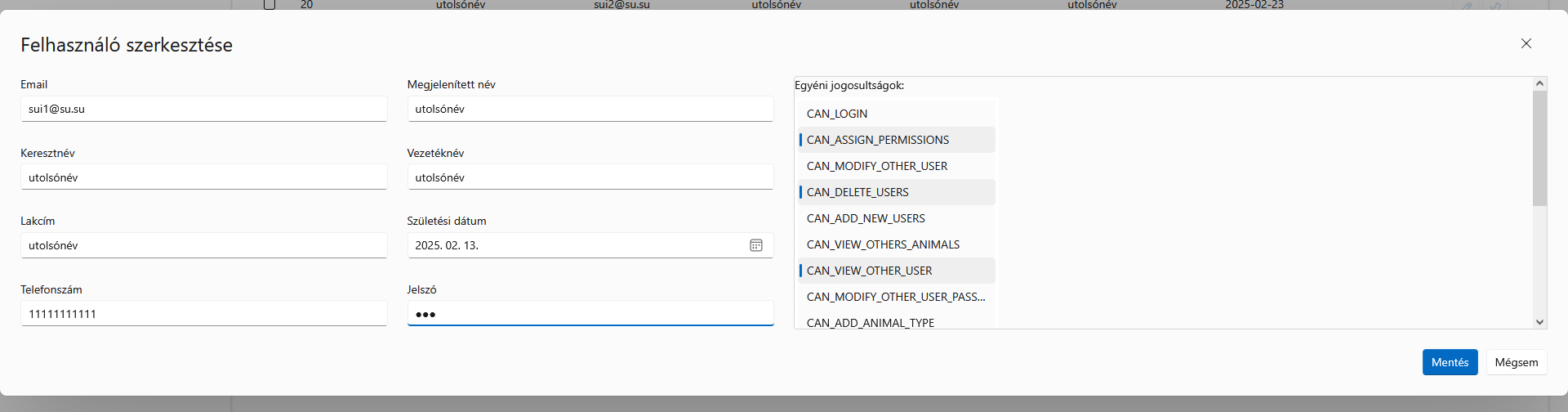
Új felhasználó létrehozása. Meg kell adni az új felhasználó adatait megfelelő jogosultságokat hozzáadni. Miután létre lett hozva a felhasználó kap egy email-t és azon keresztül be tud lépni jelszóváltoztatás után.



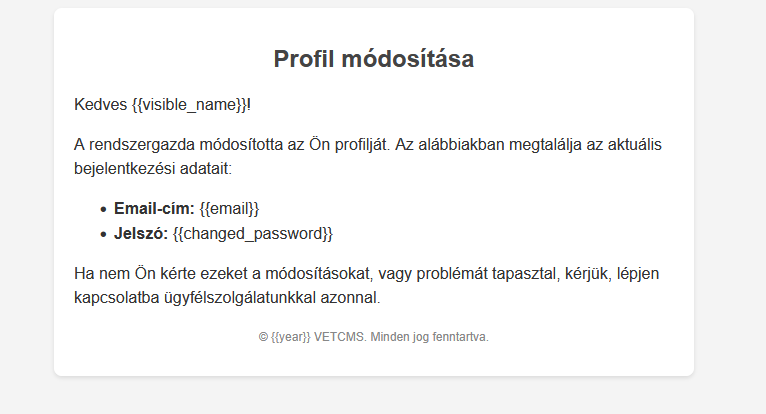
Az új felhasználó email-je.



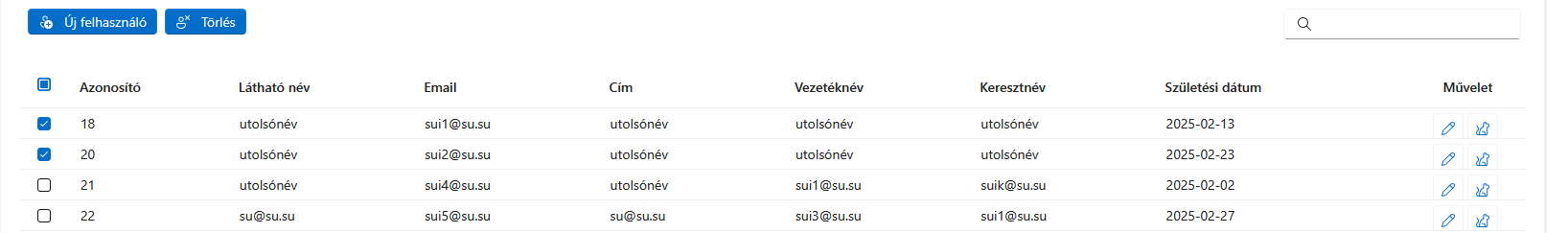
Felhasznlót beazonosító kód látható felül. Biztonság miatt új jelszót kell megadni a bejelentkezéshez. A megadott jogosultságoknak megfelelően láthatóak lesznek az a megfelelő funkcionalitások.



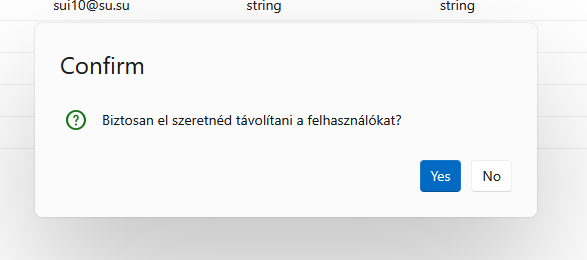
Felhasználó módosítása. Változtatásokról email-t kap a felhasználó. Ha változtak a bejelentkezési adatok azt is megkapja a felhasználó. Utána jelszót kell változtatni az előbb bemutatott funkció segítségével.

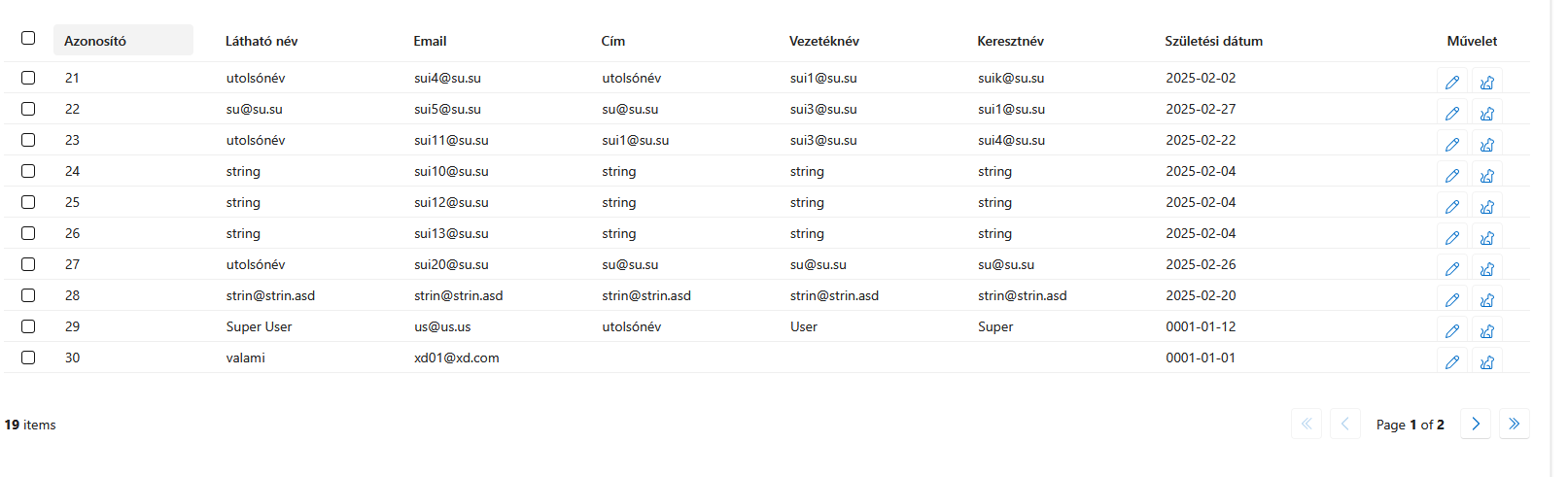


Profil módosításáról kapott értesítés.



Törléshez való kijelölés.



  
El is tűntek a felhasználók. Az elemszámból látszódik is.

Ami látszódik jobb oldalt állat piktogram, későbbiekben lesz a gazdához rendelt állatok adatlapja, ahol föl lehet venni új kezelést is és visszanézni a korábbiakat.