**Záródolgozat feladatkiírás**

Tanuló(k) neve[[1]](#footnote-1): Dezamics Bence, Joó Barnabás, Nyikos Kata

Képzés: nappali munkarend

Szak: 5 0613 12 03 Szoftverfejlesztő és tesztelő technikus

**A záródolgozat címe:**

**„VetCare Connect” állatorvosi időpontfoglaló rendszer**

Konzulens: Bólya Gábor

Beadási határidő: 2024. 04. 15.

Győr, 2024. 04. 15.

**Módos Gábor**igazgató

**Konzultációs lap[[2]](#footnote-2)**

|  |  |  |  |
| --- | --- | --- | --- |
|  | A konzultáció | | Konzulens aláírása |
| ideje | témája |
| 1. | 2023.10.15. | Témaválasztás és specifikáció |  |
| 2. | 2024.03.01. | Záródolgozat készültségi fokának értékelése |  |
| 3. | 2024.04.05. | Dokumentáció véglegesítése |  |

**Tulajdonosi nyilatkozat**

Ez a dolgozat a saját munkánk eredménye. Dolgozatunk azon részeit, melyeket más szerzők munkájából vettünk át, egyértelműen megjelöltük.

Ha kiderülne, hogy ez a nyilatkozat valótlan, tudomásul vesszük, hogy a szakmai vizsgabizottság a szakmai vizsgáról kizár minket és szakmai vizsgát csak új záródolgozat készítése után tehetünk.

Győr, 2024. április 15.

Dezamics Bence

Joó Barnabás

Nyikos Kata

A képen Betűtípus, Grafika, Grafikus tervezés, szöveg látható

Automatikusan generált leírás

Készítette:

Joó Baranbás, Dezamics Bence, Nyikos Kata

Tartalomjegyzék

[2. Bevezetés 5](#_Toc161056390)

[2.1 Probléma és megoldás 5](#_Toc161056391)

[2.2 Kezdet 5](#_Toc161056392)

[2.3 Jövőbeni tervek 5](#_Toc161056393)

[3. Csapatmunka megvalósítása 5](#_Toc161056394)

[3.1 Feladatok felosztása 5](#_Toc161056395)

[3.2 Git és GitHub 5](#_Toc161056396)

[4. Adatbázis 7](#_Toc161056397)

[4.1 Felhasználó típusú táblák 7](#_Toc161056398)

[4.1.1 Vet tábla 7](#_Toc161056399)

[4.1.2 Owner tábla 8](#_Toc161056400)

[4.1.3 Admin tábla 8](#_Toc161056401)

[4.2 Nyitvatartás táblák 8](#_Toc161056402)

[4.2.1 Opening tábla 8](#_Toc161056403)

[4.3 Kezelésekhez kapcsolódó táblák 8](#_Toc161056404)

[4.3.1 Cure tábla 9](#_Toc161056405)

[4.3.2 Pet tábla 9](#_Toc161056406)

[4.3.3 Gyik tábla 9](#_Toc161056407)

[5. Frontend 10](#_Toc161056408)

[5.1 Nyilvános oldalak 10](#_Toc161056409)

[5.1.1 Kezdőlap 10](#_Toc161056410)

[5.1.2 GYIK 10](#_Toc161056411)

[5.1.3 Állatorvosok 10](#_Toc161056412)

[5.1.4 Regisztráció 10](#_Toc161056413)

# Bevezetés

A VetCare Connect egy állatorvosok és gazdák közötti rendszer.

Fő funkciói az időpontfoglalás és a felhasználók állatainak nyilvántartása, a kezelések időpontjainak számontartása, email-es emlékeztetők küldése.

## Probléma és megoldás

Jelenleg Magyarországon nem létezik olyan oldal, amelyen keresztül könnyen válogathatunk az állatorvosok között, és online tudunk hozzájuk időpontot foglalni. A VetCare Connect ebben is nyújt segítséget a gazdáknak. Emellett az orvosok számára is hasznos funkciókat tartogat. Amellett, hogy a gazdák könnyebben rátalálnak egy orvosra (tehát az oldal egy „reklámfelületként” is szolgál), az orvosok időpontjainak nyilvántartásában is segít a rendszer. Egy helyen, gyorsan és egyszerűen tudja megtekinteni, hogy melyik nap, milyen kezelésekre foglaltak időpontot hozzá.

## Kezdet

Akkor fogalmazódott meg bennünk az ötlet, hogy egy ilyen alkalmazást készítsünk, amikor egyikünknek állatorvoshoz kellett vinnie a kutyáját. A bonyolult időpontkérés, hosszas várakozás váltotta ki az ötletet. Az alapkoncepció után mindenkinek rengeteg elképzelése volt, hogy hogyan, milyen funkciókkal színesíthetnénk az oldalunkat.

## Jövőbeni tervek

# Csapatmunka megvalósítása

## Feladatok felosztása

A csapatmunkát a feladatok felosztásával kezdtük. Alapelvünk az volt, hogy minimális szinten mindenki szeretne minden munkafolyamatba belelátni.

A frontendet Joó Barnabás és Nyikos Kata, míg a backendet Dezamics Bence valósította meg. Ennek ellenére a frontendesek is írtak API hívásokat, sőt Barnabás az email rendszer kialakításában vett részt aktívan. Kata főleg frontenden tevékenykedett, de kisebb API hívásokban és az adatbázis megtervezésében segített. Bence backendes feladatai mellett kisebb-nagyobb munkálatokat végzett a bejelentkezés és az állatok oldal frontendes megjelenítésével is.

## Git és GitHub

A közös munkához a GitHub webes felületét alkalmaztuk. A GitHub egy nyílt forráskódú szoftver, amely arra szolgál, hogy egy programozó nyomon követhesse a változásokat egy elosztott verziókezelő rendszerben. Git segítségével szoftverfejlesztési verziókövetés-szolgáltatást nyújt.

A Git a változásokat egy elosztott verziókezelő rendszeren keresztül követi. Követni tudja a projekt különböző verzióinak állapotát, amíg fejlesztjük őket. Elosztott, mert hozzáférhetünk a kód fájljaihoz egy másik számítógépről - és ugyanígy más fejlesztők is hozzáférhetnek.

<https://hu.wikipedia.org/wiki/GitHub>

<https://hub.hellowp.io/docs/tudasbazis/oktatoanyagok/github/github-kezdoknek/>

Választásunk azért esett ezekre, mivel nagyon elterjedt, így nem is volt kérdés, hogy ezt használjuk-e. Elterjedtsége mellett rendkívül praktikus, hiszen iskolai körülmények miatt sok tanteremből, sok gépről kellett dolgoznunk projektünkön. Könnyen elérhettük a programot, gyorsan frissíthettük a legújabb verzióra. Amikor gond adódott, egyszerűen tudtuk a kód előző működő verzióit használni a commitoknak köszönhetően.

A GitHub ezek mellett segítséget nyújtott a feladatok nyomon követésében is. Mindenki törekedett tömör, kifejező commit leírásokat adni, így amikor külön dolgoztunk is megtudtuk nézni, hogy a többiek mivel foglalkoztak.

A projektünk GitHub repository-ja elérhető a <https://github.com/dezbence/14AA-C-VetCareConnect> linken.

# Adatbázis

Adatbázis kezelő rendszernek MySQL-t választottunk. A MySQL egy SQL alapú többfelhasználós, relációs adatbázis kezelő rendszer. Az adatbázis szerkesztéséhez a phpMyAdmint és a DB Forge Studiot hívtuk segítségül.

<https://hu.wikipedia.org/wiki/MySQL>

Azért döntöttünk a MySQL használata mellett, mivel a technológiában már mindannyian jártasak vagyunk és nem valószínű, hogy egyhamar kimegy a divatból, így programunk korszerű tud maradni.

A képen szöveg, szoftver, Betűtípus, Weblap látható

Automatikusan generált leírás

## Felhasználó típusú táblák

Az adatbázisunkban összesen 3 féle felhasználó típus létezik. Ezek tárolása külön-külön táblában történik. Ennek oka az, hogy más adatokat tárolunk el egy-egy felhasználó típusnál.

Az admin táblában csak egy egyszerű belépési adatokat tárolunk.

A vet és owner táblákban a belépési adatokon kívül különböző kapcsolattartási adatok is megtalálhatók.

### Vet tábla

Ebben a táblában az állatorvosok adatai találhatók.

* id: egyedi azonosító, elsődleges kulcs
* name: az állatorvos neve
* email: email cím
* password: hashelt jelszó
* postal code: irányítószám
* address: cím
* phone: telefonszám
* stamp number: orvosi pecsétszám
* email\_verified\_at: email megerősítésének időpontja

### Owner tábla

Ebben a táblában a gazdák adatait tároljuk.

* id: egyedi azonosító, elsődleges kulcs
* name: gazda neve
* email: email cím
* password: hashelt jelszó
* postal\_code: irányítószám
* phone: telefonszám
* email\_verified\_at: email megerősítésének időpontja

### Admin tábla

Ebben a táblában az adminok bejelentkezési adatai találhatók

* id: egyedi azonosító, elsődleges kulcs
* username:
* password: hashelt jelszó

## Nyitvatartás táblák

Kétféle nyitvatartás létezik az adatbázisunkban. A special\_opening táblában adott dátumra vonatkozóan tárolunk nyitvatartásokat, míg az opening táblában egy általános nyitvatartást mentünk, amely hetente ismétlődik.

### Opening tábla

* id: egyedi azonosító, elsődelges kulcs
* working\_hours: a nyitvatartás órái
* day: nap
* vet\_id: az orvos egyedi azonosítója, akihez a nyitvatartás tartozik, idegenkulcs
* Special\_opening tábla
* id: egyedi azonosító, elsődelges kulcs
* working\_hours: a nyitvatartás órái
* date: dátum, amelyre a nyitvatartás vonatkozik
* vet\_id: az orvos egyedi azonosítója, akihez a nyitvatartás tartozik, idegenkulcs

## Kezelésekhez kapcsolódó táblák

A cure táblában tároljuk az egyes kezeléseket, időpontokat. A cure\_types táblában pedig a lehetséges időpont típusok találhatók.

### Cure tábla

* id: egyedi azonosító, elsődleges kulcs
* date: a kezelés dátuma és időpontja (pl: 2024-05-05 10:00)
* pet\_id: a kezelendő állat egyedi azonosítója, idegenkulcs
* cure\_type\_id: a kezelés típusának egyedi azonosítója, idegenkulcs
* vet\_id: a kezelő állatorvos egyedi azonosítója, idegenkulcs
* Cure\_type tábla
* id: egyedi azonosító, elsődleges kulcs
* type: kezelés típusa
* period: ezt am ki kéne szedni

### Pet tábla

A tárolt állatok adatai találhatók benne.

* id: egyedi azonosító, elsődleges kulcs
* name: az állat neve
* species: az állat fajtája
* gender: az állat ivara
* weight: az állat súlya
* born\_date: az állat születési időpontja
* comment: megjegyzések (opcionális mező, pl: állat allergiái, ismertetőjegyei)
* owner\_id: az állat gazdájának egyedi azonosítója
* chip\_number: chip szám
* pedigree\_number: törzskönyv szám

### Gyik tábla

A gyakori kérdések adatait innen hívjuk le.

* id: egyedi azonosító, elsődleges kulcs
* question: kérdés
* answer: válasz

# Frontend

## Nyilvános oldalak

### Kezdőlap

A kezdőlap bemutatja azt, hogy mi is az a VetCare Connect, valamint elérhető rajta egy-egy útmutató gazdák, és orvosok számára is. Ezt igyekeztünk látványosan megoldani, animációk és design elemek segítségével. Az útmutatók a típusnak megfelelő gombok segítségével érhetőek el.

Az útmutatókban szerepelnek képek, annak érdekében, hogy könnyen értelmezhető legyen a mondanivaló. Itt ismertetjük az oldal funkcióit a felhasználókkal.

### GYIK

A gyakori kérdések menüpont a /gyik route-ra vezeti a felhasználót. Itt egy primevue komponens segítségével oldottuk meg a kérdések és a válaszok megjelenítését. A kérdéseket adatbázisból hívjuk le, majd jelenítjük meg az oldalon.

### Állatorvosok

Az Állatorvosok menüpont egy állatorvos keresőhöz vezet a /allatorvosok route-on keresztül. Itt található egy szűrő, melynek segítségével név, irányítószám és település alapján kereshetünk az állatorvosok között. Az állatorvosok neve mellett található „Időpontot foglalok” gombra kattintva, egyből az adott orvoshoz foglalhatunk időpontot. Ez a funkció csak akkor érhető el, ha a felhasználó be van jelentkezve.

### Regisztráció

A főoldalon a jobb felső sarokba kattintva a „Regisztráció” gombra kattintva tud regisztrálni mindkét típusú felhasználó. Ezzel a /regisztracio route-ra irányul az oldal. Itt először ki kell választani a felhasználói fiók típusát.

A tovább gombra kattintva érhető el a regisztráció. Itt meg kell adnia a felhasználónak az adatait. Minden mező validálva van:

* A névben nem szerepelhetnek speciális karakterek (pl. @ vagy !, stb.)
* A telefonszám és az irányítószám mezők a primevue segítségével készültek. Input mask-ot használunk, így ezen mezőkbe csak számok kerülhetnek, a megfelelő formátumban
* Az email cím mezőnél ellenőrizzük, hogy megfelelő formátumban adja-e meg a felhasználó az email címét (pl.: bodri@gmail.com)
* A jelszónak meg kell felelnie az alábbi feltételeknek: 8-64 karakter, minimum 1 nagybetű, minimum 1 kisbetű és minimum 1 szám
* A jelszót meg kell erősíteni, annak érdekében, hogy ne történhessen elírás abban
* Valamint el kell fogadni a felhasználói feltételeket

Amennyiben a felhasználó nem tölt ki minden mezőt, vagy nem fogadja el a felhasználói feltételeket, akkor egy primevue-s komponens, a toast segítségével jelenítünk meg hibaüzenetet.

A regisztráció gomb megnyomása után a felhasználó kap egy emailt, melyben meg kell erősítenie a regisztrációját. Csak ezek után tud bejelentkezni az oldalra.

REGISZTRÁCIÓ UTÁN BEJELENTKEZÉSRE IRÁNYULJON, előtte az a komponens menjen, amit kiír, és legyen egy gomb, hogy bejelentkezés, sikeres megerősítésnek normális kinézet kell

### Bejelentkezés

A főoldalon a jobb felső sarokban a Bejelentkezés gombra kattintva van lehetősége a felhasználóknak bejelentkezni. Ekkor az oldal a /bejelentkezes végpontot tölti be.

Bejelentkezéskor a felhasználó email címét és a jelszavát kell megadnia. A felhasználó megtekintheti a beírt jelszavát a szem ikonra kattintva, majd le is rejtheti azt.

Amennyiben sikeresen adta meg a bejelentkezési adatait, akkor a Bejelentkezés gombra kattintva az oldal a főoldalra irányítja a felhasználót. Ekkor az oldalon megjelennek különböző menüpontok, amelyeket csak bejelentkezés után lehet használni.

Ha a felhasználó még nem erősítette meg az email címét, akkor egy komponens jelenik meg, emlékezteti a felhasználót, hogy erősítse meg az email fiókját. Amikor megerősíti azt, akkor bekerül az adatbázis „email\_verified\_at” mezőjébe a megerősítés dátuma.

### Admin bejelentkezés

## Gazda felhasználói oldalak

Adatbázis – táblák- mezők – mit miért, miért van külön vet és owner

Felhasználói kézikönyv – orvos és gazda is

Features

Csapatmunka – github, commitok, feladat felosztása

publikálás – backend hol, frontend hol, domain vásárlás

Technológiák – mit – miért

Autentikáció 2 user fajtával – technológia, megvalósítás backend és frontend oldalon is

Jelszótitkosítás

Jogosultságok – azok kezelése roleok

Backend: technológia, miért, mit hogyan, controllerek – bennük milyen apik vannak mit tudnak – a postmanes képekkel kiegészítve

Frontend: : technológia, fejlesztői környezet, miért, mit hogyan, komponensek bemutatása, design terv, reszponzivitás, oldalak

Tesztek

Fejlesztői futtatási kézikönyv

Email rendszer működése

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírás

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírás

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírás

1. Szakmajegyzékes záródolgozat esetében több szerzője is lehet a dokumentumnak, OKJ-s záródolgozatnál egyetlen személy ad le záródolgozatot. [↑](#footnote-ref-1)
2. Szakmajegyzékes, csoportos konzultációs lap [↑](#footnote-ref-2)