

**Baranya Megyei Szakképzési Centrum
Radnóti Miklós Közgazdasági Technikum iskola**

Szakma megnevezése: Szoftverfejlesztő és –tesztelő
A szakma azonosító száma: 5 0613 12 03

Vizsgaremek

Háziállat kereső oldal

Készítették: Borsos Bence és Csősz Dániel

Osztály: 2/14-2

Pécs

2023

Tartalom jegyzék

3. oldal: Projekt leírása:

Munka felosztás

A program általános specifikációja

Adatbázis és szükséges oldalak

4-6. oldal: Weboldal bemutatása

7. oldal: Rendszer követelmények

Témaválasztás indoklása

Fejlesztői eszközök

Tervezés

8. oldal: Adattmodell

9-10. oldal: A program fontos függvényeinek leírása:

11. oldal: Tesztelés

12. oldal: Tovább fejlesztési lehetőségek

A projekt működése és célja:

A projekt célja, hogy egy olyan alkalmazást hozzunk létre, amely segít az állatbarátoknak megtalálni az eltűnt háziállataikat, illetve megtalált állatok tulajdonosát és segítséget nyújtson a keresésben. Az oldal a felhasználók által létrehozott hirdetések alapján működik. A felhasználók regisztráció után hozzáférhetnek a felületen található hirdetésfeladó űrlaphoz, amely segítségével könnyedén tudnak hirdetést feladni az eltűnt vagy megtalált háziállatokról. A hirdetések létrehozásakor többféle adatot is meg lehetne adni (pl: Melyik településen tűnt el/találták meg az adott kedvencet vagy az eltűnés időpontja), amelyek alapján lehetne szűrni a hirdetések között. Az így létrehozott hirdetések alapján a felhasználók könnyen kereshetnek a talált vagy eltűnt állatok között, és így növelhetik az esélyét annak, hogy megtalálják a keresett háziállatot. A felhasználó, ha szeretne, csatolhat képet a hirdetéséhez, így könnyebbé válhat az állat azonosítása. A felhasználók email segítségével vehetik fel egymással a kapcsolatot. Az oldal egyszerű, felhasználóbarát felületet kínál, amely segítségével a felhasználók könnyen és gyorsan megtalálhatják a keresett állatot.

Munka felosztása:

Feladat:	Eszköz	Készítő:
Adatbázis	Msql	Csősz Dániel
Front-end	Angular	Borsos Bence
Back-end	Java	Csősz Dániel
Verzió kezelő	Github	Mindketten

Az adatbázis eljárásainak teszteléséhez postmant használunk.

A program általános specifikációja:

A felhasználó, ha regisztrált és bejelentkezett az oldalra, akkor a kezdőlap fogadja a már meglévő talált vagy elveszett állatokról szóló hirdetésekkel. Ha a navigációs sávon megnyitja a saját hirdetések oldalt, akkor ott hozhat létre új hirdetést. A hirdetés mindenki számára megjelenik a kezdőlapon. Ha netán meglelt az állat vagy az állat gazdája vagy csak ki akarja törölni a hirdetést, azt is a saját hirdetésekénél tudja.

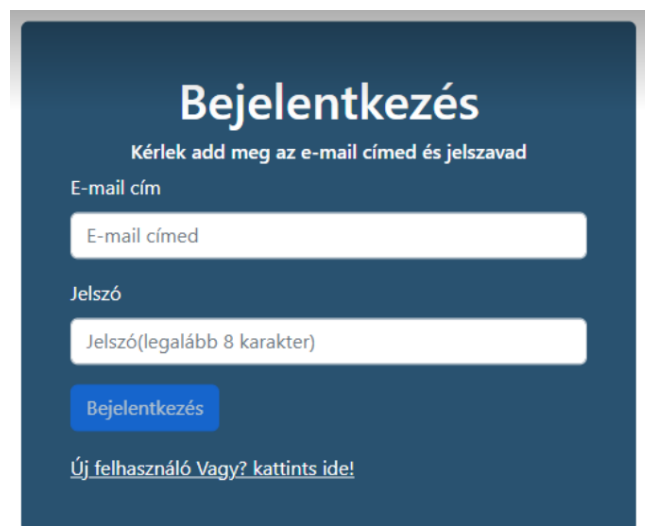
Adatbázis és szükséges oldalak:

- Felhasználóhoz tartozó tábla
- Hirdetésekhez tartozó tábla
- Címekhez tartozó tábla

- Képekhez tartozó tábla
- Regisztrációs és bejelentkezés oldal
- Kezdőlap a hirdetésekkel
- Saját profil adatai oldal
- Saját hirdetéseim oldal

A weboldal használatának leírása:

A felhasználót az oldal megnyitásakor egy login oldal fogadja, amin bejelentkezhet, vagy ha még nincs fiókja, akkor egy gombra kattintva regisztrálhat egy új fiókot. Ha hiba nélkül megadja a kért adatokat a regisztrációhoz, akkor létrejön az új felhasználó, amivel bejelentkezhet.



Bejelentkezés

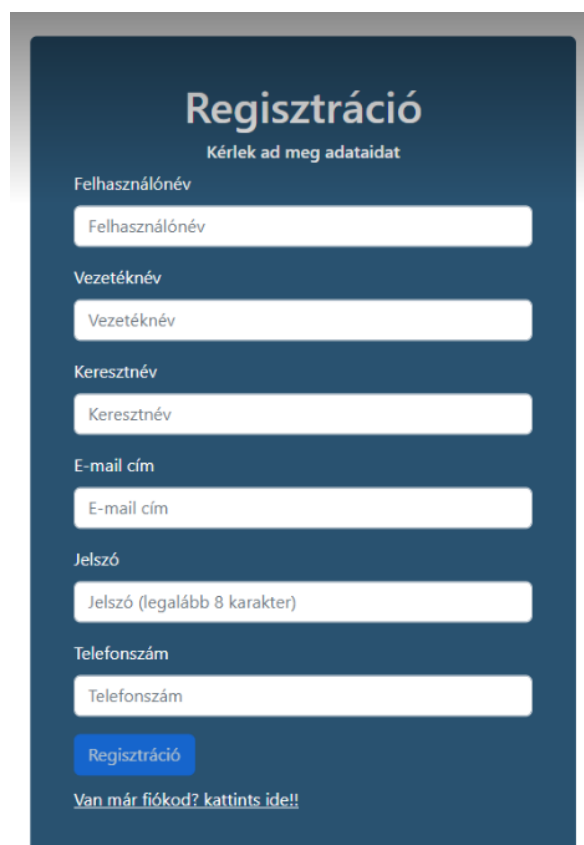
Kérlek add meg az e-mail címed és jelszavad

E-mail cím

Jelszó

Bejelentkezés

[Új felhasználó Vagy? kattints ide!](#)



Regisztráció

Kérlek ad meg adataidat

Felhasználónév

Vezetéknév

Keresztnév

E-mail cím

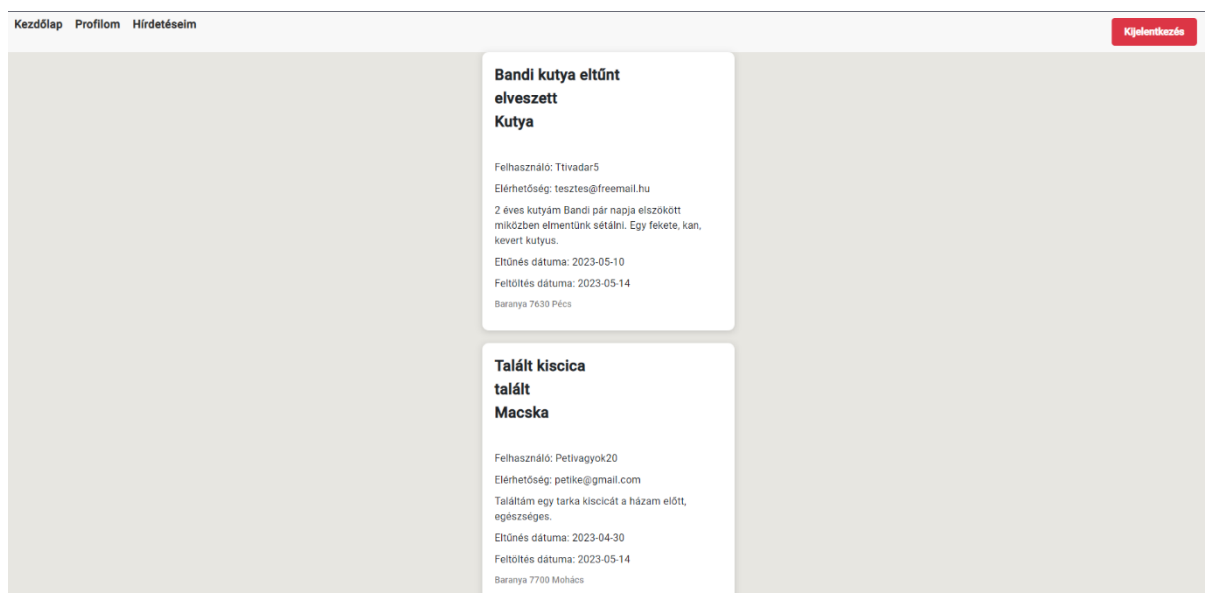
Jelszó

Telefonszám

Regisztráció

[Van már fiókod? kattints ide!!](#)

A bejelentkezés után a felhasználót a kezdőlap fogadja, ahol az összes aktív hirdetés megjelenik egy listában. A hirdetések a megadott adatokkal egy-egy kártyán jelennek meg. A weboldal tetején megtalálható egy navigációs sáv, amelyen a kezdőlap, a "profil" és a "hirdetésem" oldalhoz tartozó linkek és egy kijelentkezés gomb van.



A "profil" oldalon tekintheti meg az adatait a felhasználó és változtathatja meg őket.

The screenshot shows a form titled 'Személyes adatok' (Personal data). It contains several input fields: 'Felhasználónév' (Username) with the value 'Tivadar5', 'E-mail cím' (Email address) with the value 'tesztes@freemail.hu', 'Jelszó' (Password) with a masked value '.....', and 'Telefonszám' (Phone number) with the value '+36 70 111 0100'. At the bottom of the form, there are two links: 'Módosítások mentése' (Save changes) and 'Fiók törlése' (Delete account).

A felhasználó a "saját hirdetések" oldalon hozhat létre, törölhet, csatolhat képet és nézheti meg az általa létrehozott hirdetéseket. Az "új hirdetés létrehozás" gombra kattintva felugrik egy ablak, ahol a megadott adatok kitöltésével lehet publikálni hirdetést, amely minden aktív felhasználónak megjelenik a kezdőlapon.

Saját hirdetéseim

Hirdetés feladás

Bandi kutya eltűnt elveszett Kutya

Felhasználó: Ttivadar5

Elérhetőség: tesztes@freemail.hu

2 éves kutyám Bandi pár napja elszökött
miközben elmentünk sétálni. Egy fekete,
kan, kevert kutyus.

Eltűnés dátuma: 2023-05-10

Feltöltés dátuma: 2023-05-14

Baranya 7630 Pécs

Delete

Hirdetés Feladás

Állat Faj*

Cím*

Leírás*

Eltűnés dátuma*



MM/DD/YYYY

☐ Elveszett ☒ Talált

Cím megadása

Megye

pl: Baranya

Irányítószám

pl: 7232

Város

pl: Pécs

Cancel

Hirdetés feladása

Rendszer követelmények:

A weboldal működéséhez WildFly-szerver, Xampp és internet kapcsolat szükséges.

A témaválasztás indoklása:

Az állatok eltűnése és megtalálása sok állatbarát embert és gazdát érint. Az elkészítendő weboldal nem csak az állatok gazdáinak nyújt segítséget, hanem az állatvédők és a helyi közösség számára is. Az állatvédőknek lehetőséget nyújt arra, hogy figyelemmel kísérjék a helyi állatokat és segítséget nyújtsanak az eltűnt állatok megtalálásában. A helyi közösség számára pedig lehetőséget nyújt arra, hogy értesüljenek az eltűnt vagy megtalált állatokról, és segítséget nyújtsanak azoknak az állatoknak, amelyeknek szükségük van rá.

Fejlesztői eszközök:

Az adatbázist Msql-el, Xampp segítségével készítettük el.

A backendhez Java-t használtunk NetBeans keretrendszerben.

A frontendhez Angular-t használtunk Visual Studio Code keretrendszerben.

Az eljárások működését Postmanban ellenőriztük.

A verzió kezeléshez Githubot használtunk.

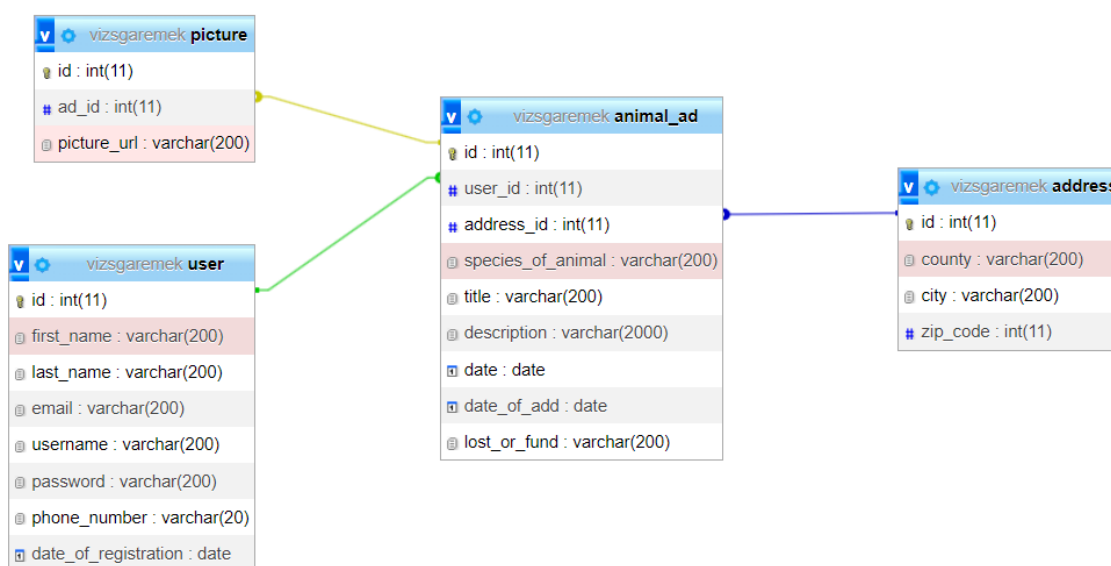
A dokumentációt wordben írtuk meg.

Tervezés:

Azt hogy weboldalt szeretnénk készíteni vizsgaremeknek hamar eldöntöttük. Utána több ötlet közül kiválasztottuk a háziállat keresőt. Ötleteltünk, hogy milyen funkció legyen az oldalon és kiválasztottuk azokat, amelyek megakartunk valósítani. A döntés után megterveztük az adatbázist a dbdiagram.io oldalon. Weboldal kinézetét Paint-be rajzolva terveztük meg, majd később változtattunk rajta, amikor már létre hoztuk a projektet.

Adatmodell:

Az adatbázis egy user, egy animal_ad, egy address, egy picture táblából és a hozzájuk tartozó eljárásokból áll. Mindegyik táblában egy id oszlop az elsődleges kulcs. Az animal_ad táblában megtalálható egy user_id, illetve egy address_id mező, mely alapján vannak össze kötve a táblák. A picture táblában egy animal_ad_id mezővel van összekötve a két tábla. A user táblában lévő password mezőt SHA1-es titkosítással titkosítottunk.



	id	first_name	last_name	email	username	password	phone_number	date_of_registration
<input type="checkbox"/> Módosítás <input type="checkbox"/> Másolás <input type="checkbox"/> Törés	1	Tivadar	Teszt	tesztes@freemail.hu	Tivadar5	71931ce13d6cc32d0c82e16ab8c19eb196475801	+36 70 111 0100	2023-05-15
<input type="checkbox"/> Módosítás <input type="checkbox"/> Másolás <input type="checkbox"/> Törés	15	Péter	Próba	petike@gmail.com	Petivagyok20	0dbe5ab58ca87a4bcbfe7be2918ecd1b22836b9	+36 30 111 0100	2023-05-15
<input type="checkbox"/> Módosítás <input type="checkbox"/> Másolás <input type="checkbox"/> Törés	16	Sándor	Kiss	sanyika70@gmail.com	Smintsándor	aca0365afa91b0d4ce66434f863ee8045f8c2789	+36 30 254 0100	2023-05-15
<input type="checkbox"/> Módosítás <input type="checkbox"/> Másolás <input type="checkbox"/> Törés	17	László	Kovács	kovacsclaci05@gmail.com	lacika	68b7a10b7500b4c41b5a35716daaa89a10c301e7	+36 20 254 0100	2023-05-15
<input type="checkbox"/> Módosítás <input type="checkbox"/> Másolás <input type="checkbox"/> Törés	18	Judit	Nagy	jucuska@citromail.hu	jutka_néni	4a75229d915f8c2c8f4d689d8efec5d1f13276c	+36 20 254 3480	2023-05-15
<input type="checkbox"/> Módosítás <input type="checkbox"/> Másolás <input type="checkbox"/> Törés	19	Elek	Vicc	vicces@gmail.com	megviccellek	edb8411b326bfe8f3fdc894db3ec4062b937b3ef	+36 30 554 3420	2023-05-15
<input type="checkbox"/> Módosítás <input type="checkbox"/> Másolás <input type="checkbox"/> Törés	20	Pál	Bekre	palivagyok@gmail.com	pali8	695db044a416f8b1308701f3db8cd7015b605568	+36 70 954 3420	2023-05-15
<input type="checkbox"/> Módosítás <input type="checkbox"/> Másolás <input type="checkbox"/> Törés	21	Márta	Szabó	mártika@gmail.com	mártuska	355c9248fb6e18bd382f96e23ae076a24f7dc0c6	+36 20 254 3720	2023-05-15

↑ ☐ Összes bejelölése A kijelöltekkel végzendő művelet: ☐ Módosítás ☐ Másolás ☐ Törés ☐ Exportálás

☐ Összes megjelenítése | Sorok száma: 25 | Sorok szűrése: Keresés a táblában | Sort by key: Nincs

#	Név	Típus	Illesztés	Tulajdonságok	Nulla	Alapértelmezett	Megjegyzések	Extra	Művelet
<input type="checkbox"/>	1 id	int(11)			Nem	Nincs		AUTO_INCREMENT	Módosítás Eldobás Több
<input type="checkbox"/>	2 first_name	varchar(200)	utf8_hungarian_ci		Nem	Nincs			Módosítás Eldobás Több
<input type="checkbox"/>	3 last_name	varchar(200)	utf8_hungarian_ci		Nem	Nincs			Módosítás Eldobás Több
<input type="checkbox"/>	4 email	varchar(200)	utf8_hungarian_ci		Nem	Nincs			Módosítás Eldobás Több
<input type="checkbox"/>	5 username	varchar(200)	utf8_hungarian_ci		Nem	Nincs			Módosítás Eldobás Több
<input type="checkbox"/>	6 password	varchar(200)	utf8_hungarian_ci		Nem	Nincs			Módosítás Eldobás Több
<input type="checkbox"/>	7 phone_number	varchar(20)	utf8_hungarian_ci		Nem	Nincs			Módosítás Eldobás Több
<input type="checkbox"/>	8 date_of_registration	date			Nem	Nincs			Módosítás Eldobás Több

```

    spq.setParameter( name: "email_in", value: u.getEmail() );
    spq.setParameter( name: "username_in", value: u.getUsername() );
    spq.setParameter( name: "password_in", value: u.getPassword() );
    spq.setParameter( name: "phone_number_in", value: u.getPhoneNumber() );

    spq.execute();
    return "Sikeres regisztráció!";
}
catch(Exception ex){
    //Handle database exceptions
    if(ex.getMessage().equals( anObject: "org.hibernate.exception.ConstraintViolationException: Error calling CallableStatement.getMoreResults()" ) ){
        return "Some unique value is duplicate!";
    }
    return "Hiba";
}

```

A program fontos függvényeinek leírása:

A regisztráció függvény egy új user-t hoz létre és rakja bele az adatbázisba. A bemenő paramétere egy User típusú elem. Megkell adni az új user-nek a keresztnévét, a vezetéknévét, az emailcímét, a felhasználónevét, a jelszavát és a telefonszámát. Ha minden adatot helyesen adott meg, akkor a függvény vissza adja, hogy „Sikeres regisztráció!”, viszont ha valami nem jó, akkor „Hibát” ír ki. Az adatok a frontenden vannak lekezelve, mint például az email-t formailag helyesen kell megadni. A függvényt a UserController-ben hívjuk meg.

```

@POST
@Path("/register")
@Consumes({ "application/json" })
public Response register(User user) {
    String response = "";
    try {
        response = userController.register(user);
    } catch (Exception e) {
        response = "Hiba";
    }
    return Response.status(200).entity(response).build();
}

```

```

<form [formGroup]="loginForm" (ngSubmit)="login(loginForm.value)">
  <div class="mb-3">
    <label for="exampleInputEmail1" class="form-label">E-mail cím</label>
    <input formControlName="email" type="email" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp" placeholder="E-mail címéd">
    <div *ngIf="loginForm.get('email').hasError('pattern') && loginForm.get('email').touched" class="text-danger">Helytelen e-mail cím formátum</div>
    <div *ngIf="loginForm.get('email').hasError('required') && loginForm.get('email').touched" class="text-danger">Az e-mail cím megadása kötelező</div>
  </div>
  <div class="mb-3">
    <label for="exampleInputPassword1" class="form-label">Jelszó</label>
    <input formControlName="password" type="password" class="form-control" id="exampleInputPassword1" placeholder="Jelszó(legalább 8 karakter)" minlength="8">
    <div *ngIf="loginForm.get('password').hasError('required') && loginForm.get('password').touched" class="text-danger">A jelszó megadása kötelező</div>
    <div *ngIf="loginForm.get('password').hasError('minlength') && loginForm.get('password').touched" class="text-danger">A jelszó legalább 8 karakter hosszú kell legyen</div>
  </div>
  <button type="submit" class="btn btn-primary">Regisztráció</button>
</form>

```

```

public static String create_new_ad(AnimalAd a){
EntityManagerFactory emf = Persistence.createEntityManagerFactory( persistenceUnitName:Database.getPuName());
EntityManager em = emf.createEntityManager();

try{
    //Create SPQ and run it
    StoredProcedureQuery spq = em.createStoredProcedureQuery( procedureName:"create_new_ad");

    spq.registerStoredProcedureParameter( parameterName:"user_id_in", type:Integer.class, mode:ParameterMode.IN);
    spq.registerStoredProcedureParameter( parameterName:"species_of_animal_in", type:String.class, mode:ParameterMode.IN);
    spq.registerStoredProcedureParameter( parameterName:"title_in", type:String.class, mode:ParameterMode.IN);
    spq.registerStoredProcedureParameter( parameterName:"description_in", type:String.class, mode:ParameterMode.IN);
    spq.registerStoredProcedureParameter( parameterName:"date_in", type:Date.class, mode:ParameterMode.IN);
    spq.registerStoredProcedureParameter( parameterName:"lost_or_fund_in", type:String.class, mode:ParameterMode.IN);
    spq.registerStoredProcedureParameter( parameterName:"county_in", type:String.class, mode:ParameterMode.IN);
    spq.registerStoredProcedureParameter( parameterName:"city_in", type:String.class, mode:ParameterMode.IN);
    spq.registerStoredProcedureParameter( parameterName:"zip_code_in", type:Integer.class, mode:ParameterMode.IN);

    spq.setParameter( name:"user_id_in", value:a.getUserId().getId());
    spq.setParameter( name:"species_of_animal_in", value:a.getSpeciesOfAnimal());
    spq.setParameter( name:"title_in", value:a.getTitle());
    spq.setParameter( name:"description_in", value:a.getDescription());
    spq.setParameter( name:"date_in", value:a.getDate());
    spq.setParameter( name:"lost_or_fund_in", value:a.getLostOrFund());
    spq.setParameter( name:"county_in", value:a.address.getCounty());
    spq.setParameter( name:"city_in", value:a.address.getCity());
    spq.setParameter( name:"zip_code_in", value:a.address.getZipCode());

    spq.execute();
    return "Új hirdetés sikeresen létre lett hozva!";
}
catch(Exception ex){
    //Handle database exceptions
    if(ex.getMessage().equals( anObject:"org.hibernate.exception.ConstraintViolationException: Error calling CallableStatement.getMoreResults")){
        return "Some unique value is duplicate!";
    }
    return "Hiba";
}

```

Az `create_new_ad` függvénnyel tudunk létrehozni új hirdetést. A bemenő paramétere egy `AnimalAd` típusú elem. Megkell adni egy már létező user-nek az id-ját, az állat fajtát, egy címet a hirdetésnek, a hirdetés leírását, az adott eset dátumát, azt hogy elveszett vagy talált állat, a megyét, a település nevét és a település irányító számát. Ha minden adatot helyesen adott meg, akkor a függvény vissza adja, hogy „Új hirdetés sikeresen létre lett hozva!”, viszont ha valami nem jó, akkor „Hibát” ír ki. A függvény az új hirdetés mellet új címet is hoz létre az `address` táblában. Azt hogy talált vagy eltűnt állatról van szó, frontenden radio-buttonnal oldottuk meg, így nem kell a felhasználónak kézzel beírnia.

```

<mat-radio-group aria-label="Select an option" FormControlName="lostOrFund">
    <mat-radio-button value="elveszett">Elveszett</mat-radio-button>
    <mat-radio-button value="talalt">Talált</mat-radio-button>
</mat-radio-group>

```

Hírdetés Feladás

Állat Faj*

Állat Faj

Cím*

Leírás*

Eltűnés dátuma*

MM/DD/YYYY

☐ Elveszett ☒ Talált

Tesztelés:

Az elkészített eljárások működését Postmanban ellenőriztük

The screenshot shows the Postman REST client interface. On the left, a list of requests is visible, including GET, POST, and DELETE requests to the 'vizsgare...' endpoint. The main panel displays a POST request to 'http://127.0.0.1:8080/vizsgaremek-1.0-SNAPSHOT/webresources/User/Registration'. The request body is in JSON format, containing user details: firstName: 'Anna', lastName: 'Példa', email: 'anna10@gmail.com', username: 'annapelda1', password: '1232', and phoneNumber: '+36 70 111 01111'. The response status is 200 OK, and the response body shows '1 Sikeres regisztráció!'.

Továbbfejlesztési lehetőségek:

Chat:

Egy chat vagy más üzenő felület hasznos lehet az oldalon mert lehetővé tenné a felhasználók közötti könnyed és gyors kommunikációt és a közösség építésben is segíthet. Ezen kívül a készítőkkel is kapcsolatba léphetnének a felhasználók.

Örökbe fogadás:

Nem csak talált és eltűnt állatok, hanem örökbe fogadható állatok hirdetés. Ha valaki régóta nem találja eltűnt kedvencét, akkor az örökbefogadható állatok között talán megtalálhatja az új társát. Az örökbefogadók számára is előnyös lehet az oldalon való keresgélés és könnyen tudnának választani a gazdátlan állatok közül.

Jutalmazási rendszer:

Az oldalon lehetőség van jutalmakat adni azoknak a felhasználóknak, akik segítenek megtalálni az eltűnt állatokat vagy akik örökbe fogadnak egy állatot. Ez ösztönözheti a felhasználókat, hogy aktívabban részt vegyenek az oldal használatában.