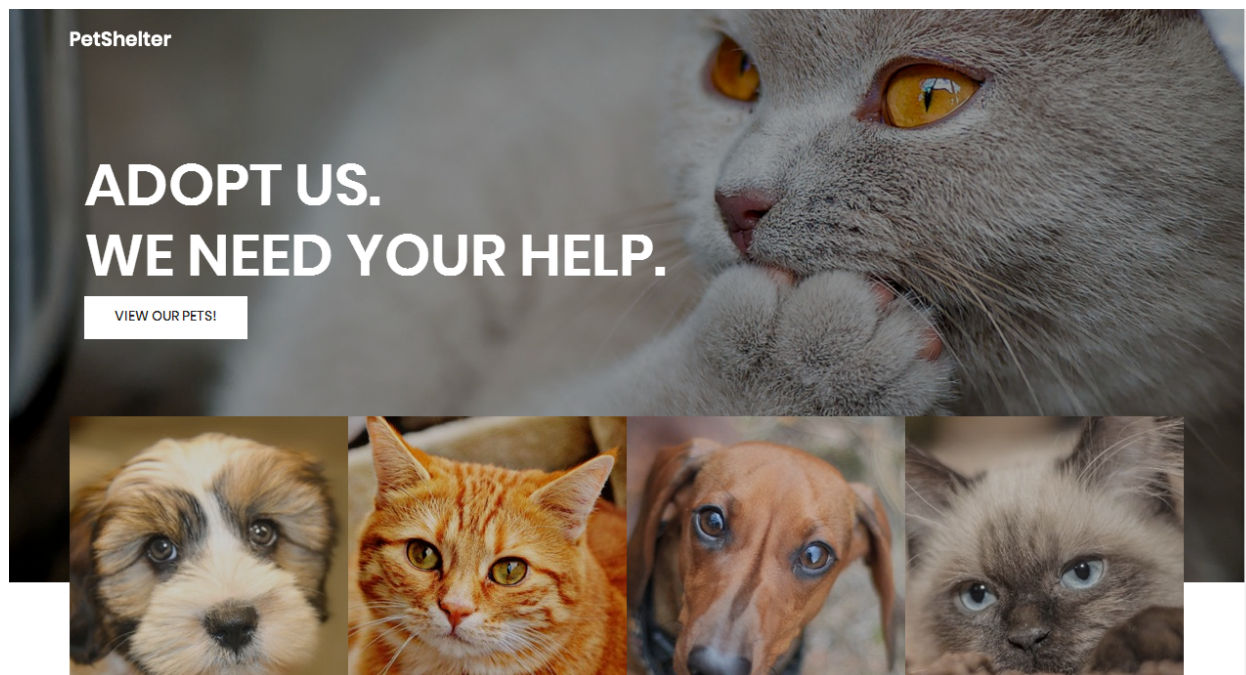


LER O ENUNCIADO ATÉ AO FIM ANTES DE COMEÇAR!

O objectivo do exame é construir um site web suportado por uma base de dados relacional. Informação sobre a base de dados encontra-se em ANEXO.



O site web consiste no portal de uma instituição protectora dos animais que permite a utilizadores registados adoptar cães e gatos. O site tem as seguintes páginas:

1. “**Home**” é a página de rosto do site;
2. “**Pets**” é a página principal do site;
3. “**Register**” para registo de utilizadores;
4. “**Login**” para o login de utilizadores;
5. “**My Pets**” permite ao utilizador registado ver os animais que adoptou ;

É dado acesso a um conjunto de templates HTML construídos em *Bootstrap* fornecidos pela empresa de web design colorlib.com.

Solicita-se ao aluno a realização do site web em *Symfony* e *Twig*, apenas das páginas acima descritas.

## PRELIMINARES

A. Faça login por ssh (com o PuTTY, por exemplo) no servidor com o IP 10.10.23.184<sup>1</sup>

```
a12345@daw2:~$
```

B. Por razões de segurança gere uma nova password de acesso à base de dados

```
a12345@daw2:~$ /usr/local/bin/mysql-db
```

C. Actualize o ficheiro .env

Altere o ficheiro “.env” para utilizar as credenciais da sua base de dados.

```
a12345@daw2:~$ cd public_html/RECURSO2
```

```
a12345@daw2:~/public_html/RECURSO$ nano .env
```

```
DATABASE_URL=mysql://a12345:*****@10.10.23.184:3306/db_a12345?serverVersion=15.1
```

D. Crie a sua cópia da base de dados que é fornecida

```
a12345@daw2:~/public_html/RECURSO$
```

```
mysql -u a12345 --password=***** db_a12345 < database.SQL
```

Substitua “a12345” pelo seu login e “\*\*\*\*\*” pela password de acesso à sua base de dados

E. Teste a instalação do framework

A partir do seu browser preferido vá ao seguinte URL

```
http://daw.deei.fct.ualg.pt/~a12345/RECURSO/public/index.php/shelter
```

---

<sup>1</sup> se não se encontra dentro da rede UALG terá que fazer primeiro login por ssh para ssh.deei.fct.ualg.pt

<sup>2</sup> a pasta RECURSO já existe na sua área pessoal

Deverá receber uma página web de boas-vindas

## Hello ShelterController!

This friendly message is coming from:

- Your controller at [src/Controller/ShelterController.php](#)
- Your template at [templates/shelter/home.html.twig](#)

**F. OPCIONAL:** se preferes<sup>3</sup> desenvolver o site no seu PC/portátil faça o download do repositório do exame com a aplicação git que tem instalada no seu PC/portátil:

```
user@meuPC:~$ git clone git://github.com/jmatbastos/RECURSO.git
user@meuPC:~/RECURSO$ composer install
```

Em alternativa, pode fazer o download do repositório num ficheiro "zip" apontando o browser no seu PC/portátil para o URL

<https://github.com/jmatbastos/RECURSO/archive/master.zip>

### NOTAS:

- Se tiver dificuldade em criar a base de dados em linha de comando pode utilizar o acesso web *phpMyAdmin* e os comandos SQL descritos no APÊNDICE
- No caso de haver algum conflito com uma tabela já existente na sua base de dados, mude o nome à tabela existente *ou* apague a tabela existente
- A aplicação deve correr obrigatoriamente na área pessoal do aluno na pasta "RECURSO" no servidor web do departamento disponibilizado para o efeito: <http://daw.deei.fct.ualg.pt>
- O controlador principal tem que estar no ficheiro "ShelterController.php"

---

<sup>3</sup> Recomenda-se que desenvolva o site no seu PC/portátil para evitar a sobrecarga do servidor web

- A classe com as funções de acesso à base de dados tem que estar no ficheiro “Shelter\_modelController.php”
- **Opcional!** Caso use o componente **Doctrine ORM** a restrição acima é levantada e pode utilizar um ficheiro para cada modelo que representa uma tabela da base de dados
- **Opcional!** Pode utilizar **bin/console** para “scaffolding” das funcionalidades de registo e autenticação de utilizadores, bem como para a criação de formulários.
- **Opcional!** Pode utilizar **Doctrine SQL Query Builder** para o acesso à base de dados.

A página de entrada no site tem que ser

**<http://daw.deei.fct.ualg.pt/~a12345/RECURSO/public/index.php/shelter>**

Deve ser considerado uma **SUGESTÃO** o seguinte mapeamento entre URLs e controladores:

```
* @Route("/shelter name="shelter")
* @Route("/pets/{cat_id?}", name="pets")
* @Route("/register", name="register")
* @Route("/login", name="login")
* @Route("/logout", name="logout")
* @Route("/adopt/{pet_id}", name="adopt")
* @Route("/myPets", name="myPets")
```

**Pode ver um demo do site no URL**

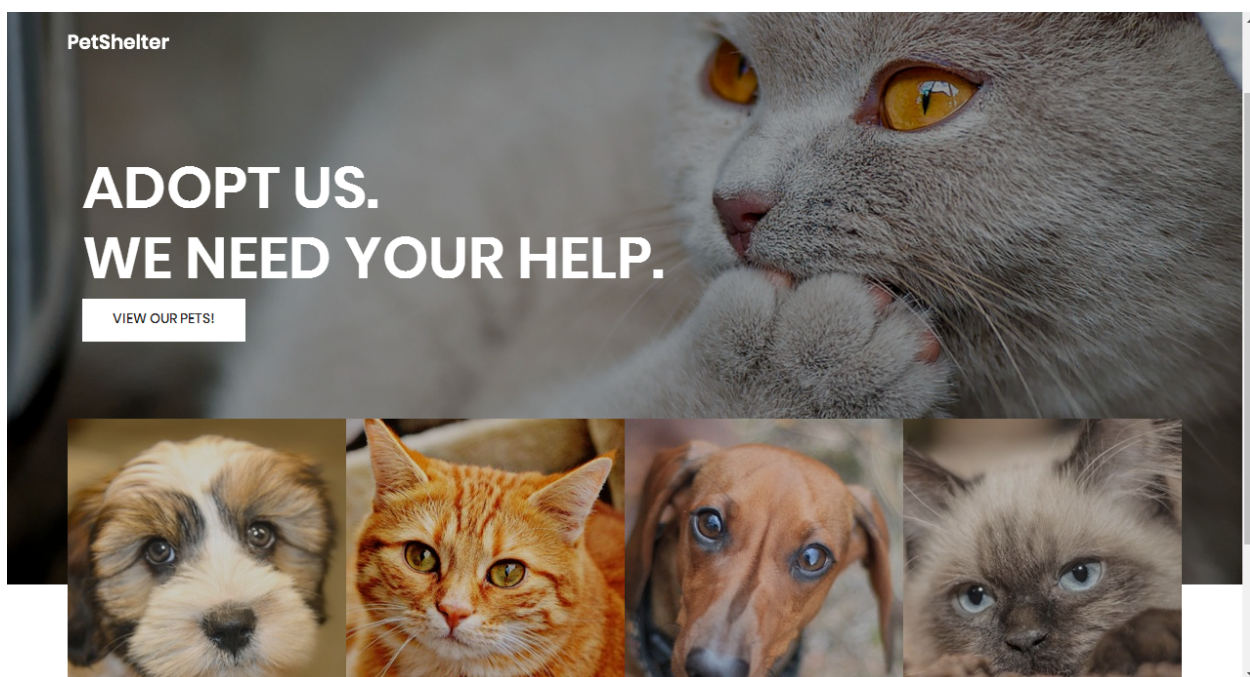
**<http://daw.deei.fct.ualg.pt/~a12345/RECURSO/demo>**

# 1. [2 valores] FUNCIONALIDADE “HOME”

A funcionalidade “Home” é página de rosto do site

Construa o template Twig para esta página.

SUGESTÃO: Adapte o template “index.html” fornecido pela empresa de web design. Encontra este template na pasta “demo” (~public\_html/RECURSO/demo), bem como os componentes imagens, css, e javascript utilizados.



Construa o método no controlador `ShelterController.php` responsável por gerar a página de rosto do site

O botão “VIEW OUR PETS” é um hyperlink (“/pets”) que permite mostrar a página com os animais disponíveis para adoção

## NOTA:

- É OBRIGATÓRIO o seu site ser portátil: os hyperlinks para recursos locais não podem ser absolutos! Recomenda-se que utilize as funções `path()` e `asset()` para gerar todos os hyperlinks locais.

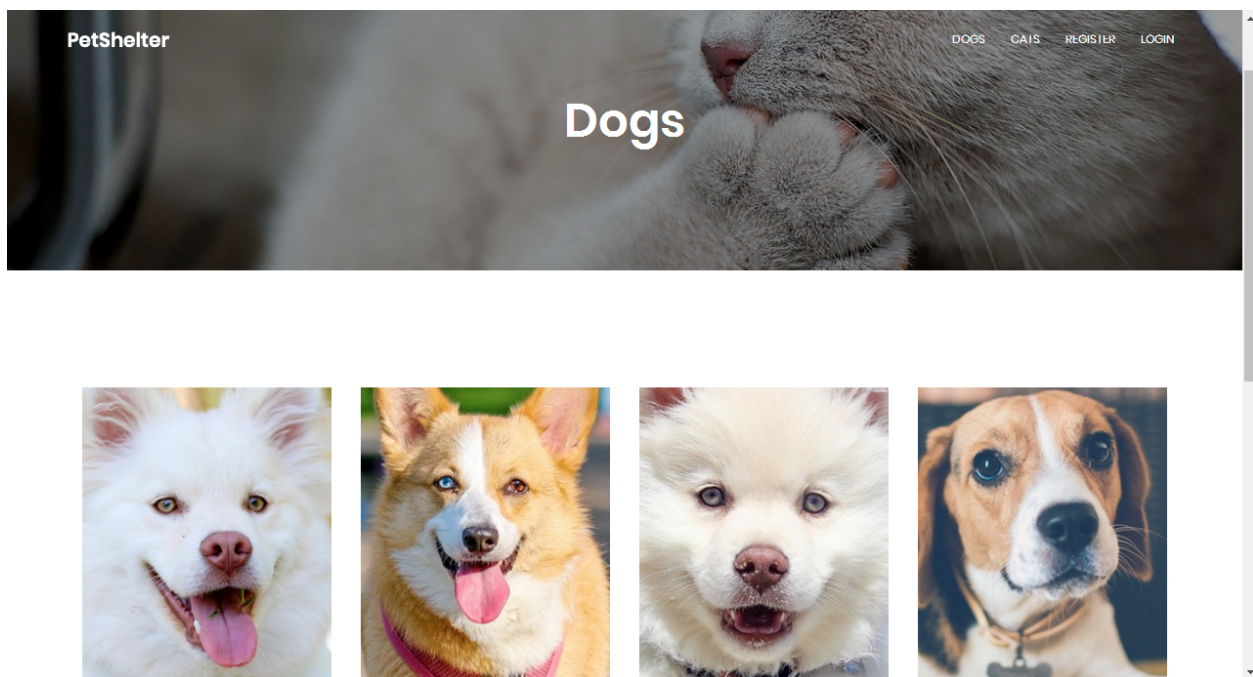


## 2. [3 valores] FUNCIONALIDADE “PETS”

A funcionalidade “Pets” é página principal do site e contem uma lista dos animais disponíveis para adoção.

Construa o template Twig para esta página.

SUGESTÃO: Adapte o template “pets.html” fornecido pela empresa de web design. Encontra este template na pasta “demo”



Construa o método no controlador `ShelterController.php` e a query correspondente no controlador `Shelter_model.php` responsável por fazer uma lista dos animais existentes na tabela “pets” da base de dados

- A página principal do site (“/pets”), mostra todos os animais *que ainda não foram adoptados*

- Na página “pets”, as categorias de animais (“Dogs”, “Cats”, ...) são hyperlinks com dados embutidos (“/pets/1”, “/pets/2” etc) que permitem mostrar uma lista de animais apenas da categoria seleccionada, e *que ainda não foram adoptados* (“status” igual a “NULL” ou “0”)
- Para cada animal mostre a imagem e o nome (“name”, “image”, existentes na tabela “pets”)
- Caso o utilizador tenha feito login, na foto de cada animal deverá haver um hyperlink com dados na forma de segmentos (“/adopt/1”, “/adopt/2” etc) que permite ao utilizador adoptar o animal
- “REGISTER”, “LOGIN” são hyperlinks (“/register”, “/login”) activos caso o utilizador seja anonimo. Caso o utilizador tenha feito login, transformam-se em “LOGOUT”, “MY PETS” (“/logout”, “/myPets”), bem como aparece o texto “Welcome user” onde *user* é o login do utilizador

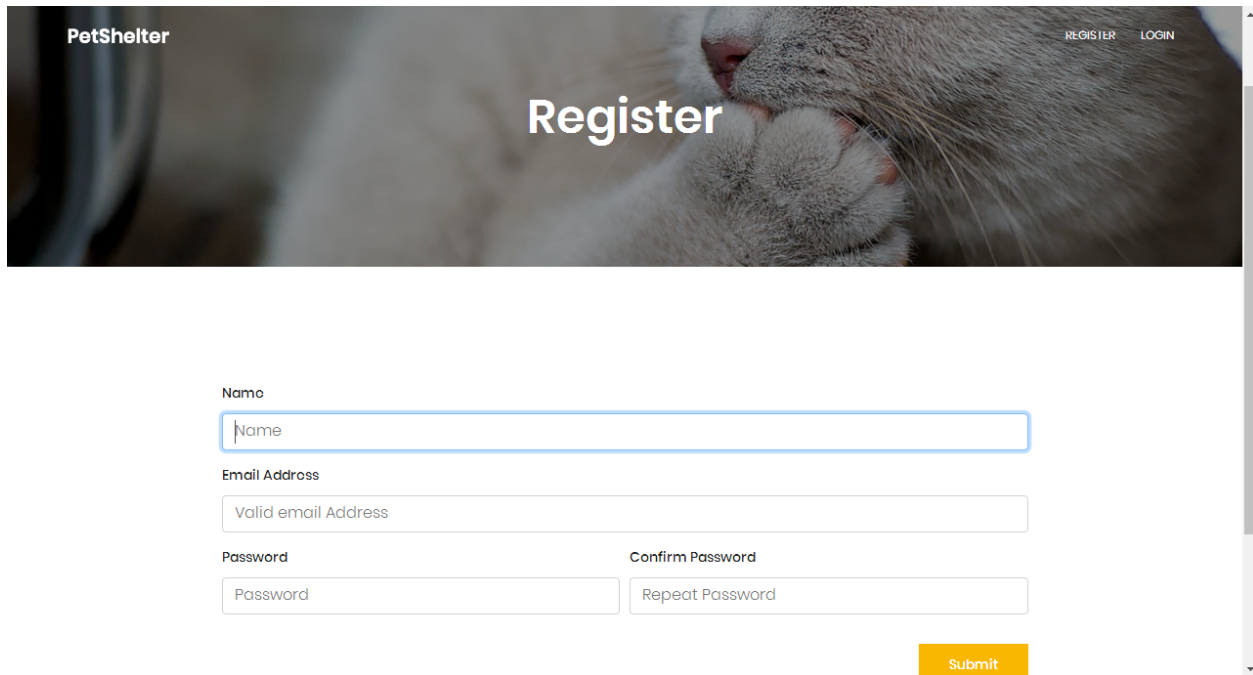
## FUNCIONALIDADE “REGISTER”

A funcionalidade “Register” permite registar um utilizador.

### 3. [2 valores]

Construa o template Twig para esta página.

SUGESTÃO: Adapte o template “register.html” fornecido pela empresa de web design. Encontra este template pasta “demo”.



The screenshot shows a web registration form for 'PetShelter'. The page has a dark header with the 'PetShelter' logo on the left and 'REGISTER' and 'LOGIN' links on the right. The main content area features a large, light-colored image of a cat's face with the word 'Register' overlaid in white. Below the image is a registration form with the following fields: 'Name' (text input), 'Email Address' (text input with placeholder 'Valid email Address'), 'Password' (text input), and 'Confirm Password' (text input with placeholder 'Repeat Password'). A yellow 'Submit' button is located at the bottom right of the form.

Construa o método no controlador `ShelterController.php`

- Se todos os campos não estão preenchidos, redirecciona novamente para o formulário com uma mensagem de erro. Os campos correctamente preenchidos devem ser retornados ao formulário.
- Se todos os campos obrigatórios estão correctamente preenchidos, e o email ainda não existe na base de dados, regista o utilizador na tabela “petlovers” utilizando a query correspondente no controlador `Shelter_model.php`.

Para evitar que haja *hashes* iguais para passwords iguais na base de dados, o que é uma falha de segurança, vai-se encriptar a password usando uma “semente” (*seed*) para todas as passwords. Sugere-se a seguinte estratégia

```
$seed = substr(time(),-4);  
$password_digest = $seed . md5($seed . $password);
```



A "semente" é gerada de forma pseudoaleatória utilizando os últimos 4 algarismos da função `time()`.

A password é a *hash* utilizando o algoritmo MD5 da concatenação da "semente" e da password. A "semente" fica guardada nos primeiros 4 caracteres da `password_digest`.

Pode utilizar a função MySQL `NOW()` ou a função PHP `date("Y-m-d H:i:s")` para actualizar os campos `created_at`, `updated_at` da tabela "petlovers"

## FUNCIONALIDADE “LOGIN” & “LOGOUT”

A funcionalidade “Login” permite autenticar um utilizador.

### 4. [5 valores]

Construa o template Twig correspondente.

SUGESTÃO: Adapte o template “register.html” fornecido pela empresa de web design. Encontra este template na pasta “demo”

Construa os métodos necessários no controlador `ShelterController.php` e a query correspondente no controlador `Shelter_model.php`,

- em caso de sucesso no login:
  - regista no array da sessão os dados “id” e “name” do utilizador retirados da base de dados
  - re-direcciona para o portal principal do site “Pets”
  - O hyperlink “Login” transforma-se no hyperlink “Logout” e o hyperlink “Register” transforma-se no texto “Welcome user!” (onde “user” é o nome do utilizador registado) na pagina “Pets”
  - Na página “Pets” aparece um novo hyperlink “My Pets”
  - Na imagem de cada animal é visível um hyperlink “adopt” com o id do animal embutido (`/adopt/1`, `/adopt/2` etc)
- em caso de insucesso no login
  - re-direcciona novamente para a página `login`.
  - Envia uma mensagem de erro “Login failed”

Construa o método “logout” que encerra a sessão no controlador `ShelterController.php`

## 5. [2 valores] FUNCIONALIDADE “ADOPT”

A funcionalidade “Adopt” permite ao utilizador registado adoptar um animal

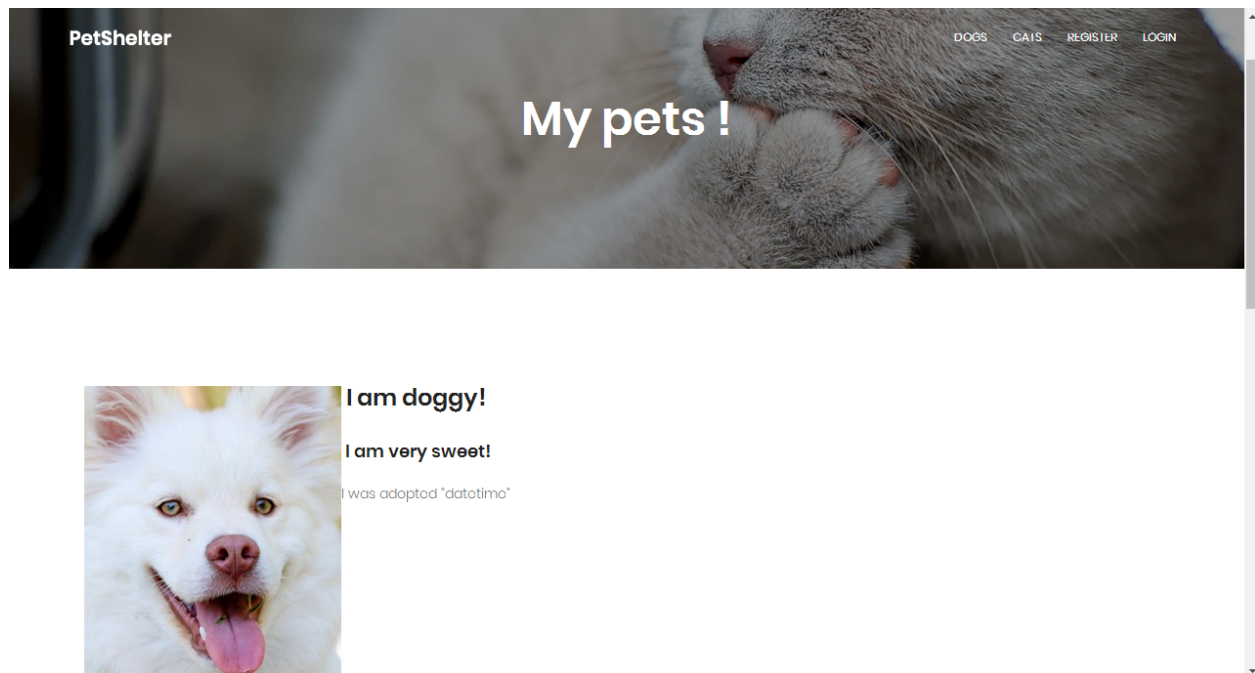
Construa o método no controlador `ShelterController.php` e as queries correspondentes no controlador `Shelter_model.php` responsáveis por fazer

- a actualização do “status=1” do animal na tabela “pets”,
- o registo da adopção do animal na tabela “adoptions”. Pode utilizar a função MySQL `NOW()` ou a função PHP `date("Y-m-d H:i:s")` para actualizar o campo `created_at`
- Re-direccione o utilizador para a página principal do site “Pets” (a mensagem de sucesso da operação é opcional)

(A utilização de uma *transaction* para garantir a atomicidade da operação na base de dados é opcional)

## 6. [3 valores] FUNCIONALIDADE “MY PETS”

A funcionalidade “My Pets” permite ao utilizador registado mostrar os animais que adoptou



Construa o template Twig para esta página.

SUGESTÃO: Adapte o template “mypets.html” fornecido pela empresa de web design. Encontra este template na pasta “demo”.

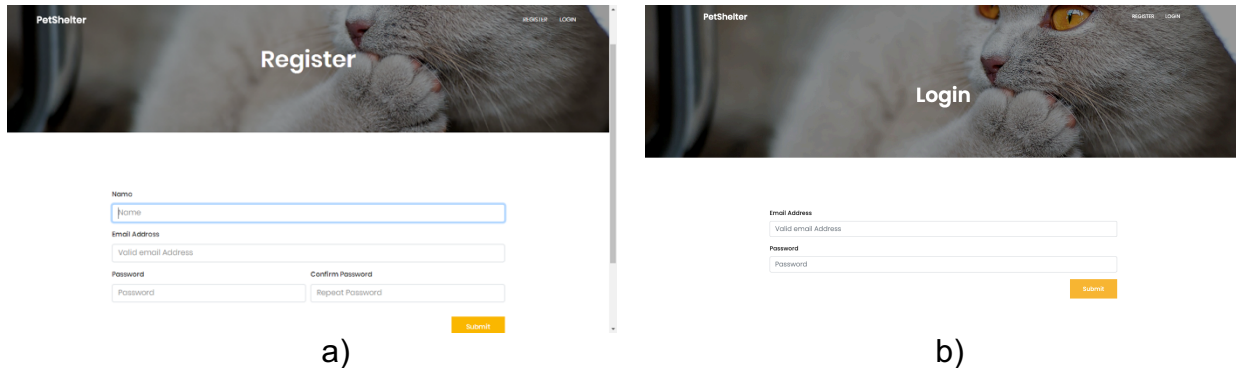
Construa o método no controlador `ShelterController.php` e a query correspondente no controlador `Shelter_modelController.php` responsável por fazer uma lista dos animais existentes na tabela “adoptions” da base de dados, cujo “id” da pessoa que adoptou corresponde à do utilizador que fez login.

Para além da foto, a lista deve mostrar o nome, a descrição e a data de adopção do animal.

Na página existem ainda os hyperlinks “CATS” “DOGS” e “LOGOUT”, bem como o texto “Welcome user”. “Petshelter” é em todas as páginas do site um hyperlink para a página de rosto do site (“/shelter”).

## 7. [3 valores] FUNCIONALIDADE "AUTHENTICATE"

A funcionalidade "Authenticate" junta num único template os templates register e login:



- a) Se o utilizador carregar no link REGISTER deve aparecer o formulário "register"
- b) Se o utilizador carregar no link LOGIN deve aparecer o formulário "login"

Ambos os formulários já existem no template; apenas um está visível, o outro está "escondido". (Por omissão está visível o formulário REGISTER.)

Sugere-se **não** mexer nos templates login.html.twig e register.html.twig que criou anteriormente. Programe no novo template "access.html.twig" em Javascript/JQuery/Vue.js (à sua escolha) esta funcionalidade.

Sugere-se **não** mexer nos controladores "login" e "register", mas sim criar um novo controlador "access"

```
* @Route("/access", name="access")
```

e direccionar os hyperlinks na página "Pets" para este novo controlador

# NOTAS:

- Caso tenha trabalhado no seu PC/portatil, **é obrigatório fazer o upload de todos os ficheiros** para a pasta “RECURSO” no seu site web pessoal

```
/users/a12345/public_html/RECURSO
```

(onde 12345 é o seu número de aluno). Utilize scp (Linux) ou WinSCP (Windows) ou FileZilla (Windows e MAC) para fazer a cópia. As permissões dos ficheiros devem ser octal 640 (rw- r-- ---).

- NÃO faça o upload de pastas! Se precisar de criar uma pasta faça-o no servidor com o comando

```
mkdir nome_da_pasta
```

- Se precisar de copiar uma pasta faça-o no servidor com o comando

```
cp -R nome_da_pasta_original nome_da_pasta_destino
```

- Caso tenha problemas como seu código, lembre-se que pode consultar o log do servidor web com o comando

```
a12345@daw:~$ tail -f /var/log/apache2/error.log
```

**Verifique que o site fica operacional.**



# ANEXO 1

Acesso à base de dados MySQL

- O acesso à base de dados MySQL pode ser feita em linha de comando (substitua "12345" pelo seu número de aluno)

```
a12345@daw:~$mysql -u a12345 -p db_a12345
```

ou ainda utilizando o software **phpMyAdmin** disponível no URL

- <http://daw.deei.fct.ualg.pt/phpMyAdmin>

## ANEXO 2 Estrutura da base de dados

```
--
-- Table structure for table `petlovers`
--

CREATE TABLE IF NOT EXISTS `petlovers` (
  `id` int(11) NOT NULL auto_increment,
  `name` varchar(255) default NULL,
  `email` varchar(255) default NULL,
  `created_at` datetime NOT NULL,
  `updated_at` datetime NOT NULL,
  `password_digest` varchar(255) default NULL,
  `remember_digest` varchar(255) default NULL,
  `admin` tinyint(1) default NULL,
  `activation_digest` varchar(255) default NULL,
  `activated` tinyint(1) default NULL,
  `activated_at` datetime default NULL,
  `reset_digest` varchar(255) default NULL,
  `reset_sent_at` datetime default NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `index_users_on_email` (`email`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Table structure for table `petcategories`
--

CREATE TABLE IF NOT EXISTS `petcategories` (
  `id` int(11) NOT NULL auto_increment,
  `name` varchar(255) default NULL,
  `description` varchar(255) default NULL,
  `image` varchar(255) default NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Dumping data for table `petcategories`
--

INSERT INTO `petcategories` VALUES
(1, 'Dogs', NULL, NULL), (2, 'Cats', NULL, NULL);

--
-- Table structure for table `pets`
--

CREATE TABLE IF NOT EXISTS `pets` (
  `id` int(11) NOT NULL auto_increment,
  `cat_id` int(11) NOT NULL,
  `name` varchar(255) default NULL,
  `description` varchar(255) default NULL,
  `status` int(5) default NULL,
  `image` varchar(255) default NULL,
  PRIMARY KEY (`id`),
```

```

    CONSTRAINT FOREIGN KEY (`cat_id`) REFERENCES `petcategories` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Dumping data for table `pets`
--

INSERT INTO `pets` VALUES (1,1,'Max','So full of
life!',0,'dg1.jpg'), (2,1,'Buddy','I love
company!',0,'dg2.jpg'), (3,1,'Charlie','I like
biscuits!',0,'dg3.jpg'), (4,1,'Rocky','I love my
boss!',0,'dg4.jpg'), (5,1,'Cooper','I am bad!',0,'dg5.jpg'), (6,1,'Bear','I
love going to the beach!',0,'dg6.jpg'), (7,1,'Doggy','So
sweet!',0,'dg7.jpg'), (8,1,'Duke','I am you best
friend!',0,'dg8.jpg'), (9,2,'Kitty','So shy!',0,'ct1.jpg'), (10,2,'Lucky','I
love to hunt mice!',0,'ct2.jpg'), (11,2,'Princess','I am very
sweet!',0,'ct3.jpg'), (12,2,'Misty','I love to
play!',0,'ct4.jpg'), (13,2,'Smokey','Miau!',0,'ct5.jpg'), (14,2,'Sassy','I
am very curious!',0,'ct6.jpg'), (15,2,'Simba','I am
bad!',0,'ct7.jpg'), (16,2,'Tareco','I love sunny days!',0,'ct8.jpg');

--
-- Table structure for table `adoptions`
--

CREATE TABLE IF NOT EXISTS `adoptions` (
  `id` int(11) NOT NULL auto_increment,
  `petlover_id` int(11) NOT NULL,
  `pet_id` int(11) NOT NULL,
  `created_at` datetime NOT NULL,
  PRIMARY KEY (`id`),
  CONSTRAINT FOREIGN KEY (`petlover_id`) REFERENCES `petlovers` (`id`),
  CONSTRAINT FOREIGN KEY (`pet_id`) REFERENCES `pets` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```