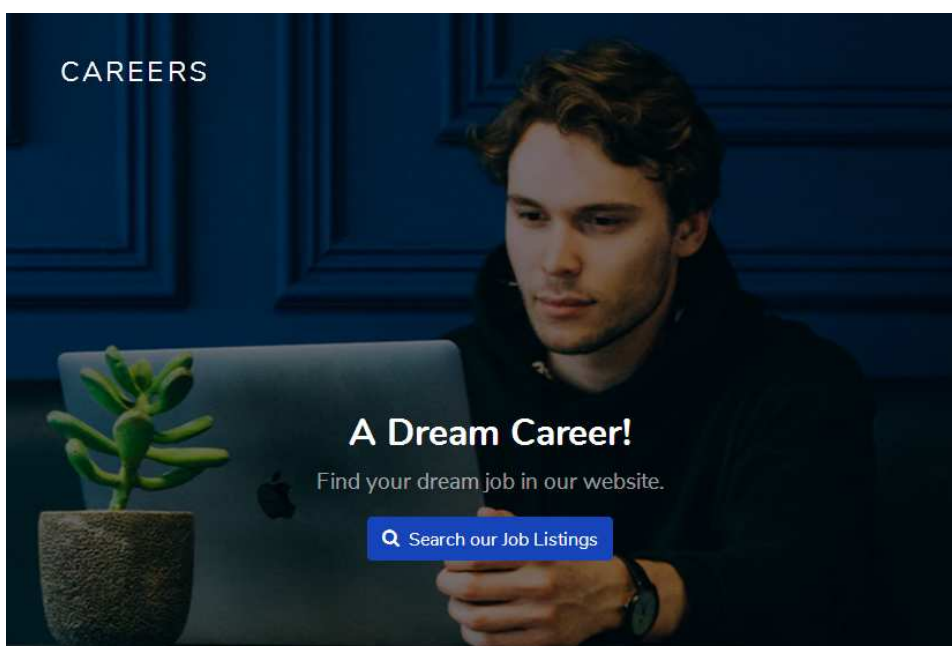


LER O ENUNCIADO ATÉ AO FIM ANTES DE COMEÇAR!

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 empresa de empregos que permite a utilizadores registados candidatar-se a empregos. O site tem as seguintes páginas:

1. “**Home**” é a página de rosto do site;
2. “**Job Listings**” é a página principal do site;
3. “**Register**” para registo de utilizadores;
4. “**Login**” para o login de utilizadores;
5. “**Apply**” permite ao utilizador registado candidatar-se a um emprego
6. “**My jobs**” permite ao utilizador registado ver uma listagem dos empregos a que se candidatou;
7. “**Message**” permite enviar mensagens;

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 *Laravel versão 5.4*, apenas das páginas acima descritas.

PRELIMINARES

A. Faça login por ssh (com o PuTTY, por exemplo) no servidor com o IP 10.10.23.183

```
a12345@daw: ~$
```

B. Mude (*change directory*) para a pasta `public_html`:

```
a12345@daw: ~$ cd public_html
```

C. Faça download para a pasta “MELHORIA”*, do código do site web

```
a12345@daw: ~/public_html$
```

```
git clone https://github.com/jmatbastos/MELHORIA.git MELHORIA
```

D. Dê permissões de escrita em pastas seleccionadas

```
a12345@daw: ~/public_html/$ chmod -R g+w MELHORIA/storage
```

```
a12345@daw: ~/public_html/$ chmod -R g+w MELHORIA/bootstrap/cache
```

```
a12345@daw: ~/public_html/$ mkdir MELHORIA/uploads
```

```
a12345@daw: ~/public_html/$ chmod -R g+w MELHORIA/uploads
```

E. Crie a sua cópia da base de dados

```
a12345@daw: ~/public_html$
```

```
mysql -u a12345 -p db_a12345 < MELHORIA/database.SQL
```

NOTAS:

- **substitua “12345” pelo seu número de aluno!**
- Se **não** se recorda da password da sua base de dados, recupere-a com o comando

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

- Se tiver **dificuldade** em criar a base de dados em linha de commando pode utilizar o acesso web **phpMyAdmin** e os comandos SQL descritos no APÊNDICE 2

* NOTA: a pasta “MELHORIA” é criada automaticamente pelo comando `git clone`

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

```
a12345@daw:~/public_html$ cd MELHORIA
a12345@daw:~/public_html/MELHORIA $ nano .env
```

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=db_a12345
DB_USERNAME=a12345
DB_PASSWORD=*****
```

G. Teste a instalação do framework

A partir do seu browser preferido vá ao seguinte URL

<http://10.10.23.183/~a12345/MELHORIA>

Deverá receber uma página web de boas-vindas

The image shows the Laravel logo, which consists of the word "Laravel" in a light gray, sans-serif font.

[DOCUMENTATION](#)

[LARACASTS](#)

[NEWS](#)

[FORGE](#)

[GITHUB](#)

1. [2 valores] FUNCIONALIDADE “HOME”

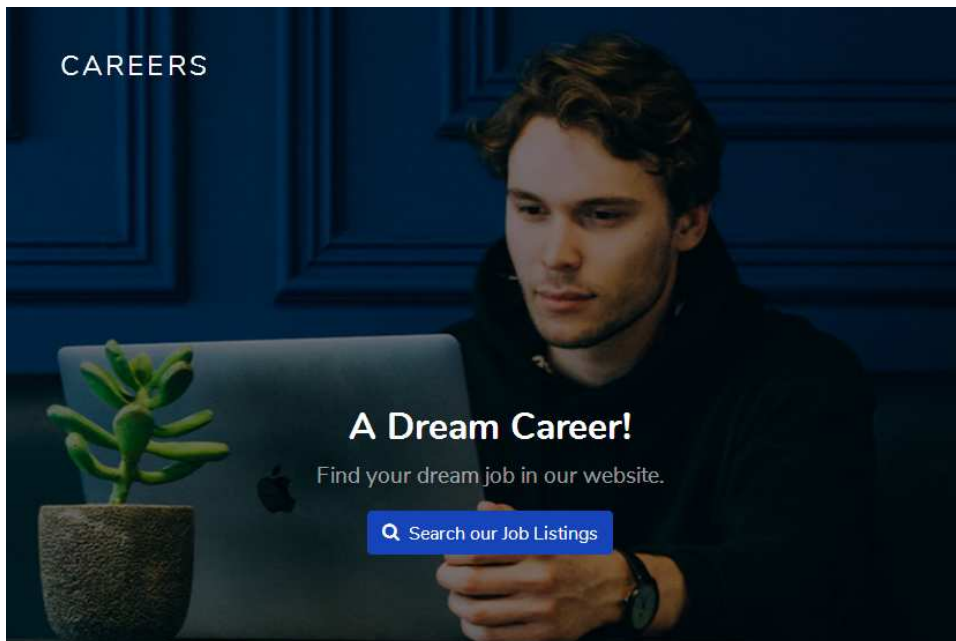
A funcionalidade “Home” é página de rosto do site

Construa o template Blade para esta página.

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

Pode ver um demo do site no URL

<http://10.10.23.183/~a12345/MELHORIA/assets/index.html>



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

O botão “Search our Job Listings” é um hyperlink (“/job-listings”) que permite mostrar a página com os empregos publicados

NOTA:

- O seu site TEM que ser portátil; utilize sempre as funções `action()`, `url()` e `asset()` para gerar todos os hyperlinks internos!

- É OBRIGATÓRIO o único controlador do seu site ser `Careers.php` e TODOS os métodos necessários para gerar este site encontrarem-se neste controlador,
- EXCEPTO todas as queries à base de dados, que têm que se encontrar no ficheiro `Careers_model.php` (caso use a classe Eloquent esta restrição é levantada e pode utilizar um modelo para cada tabela da base de dados)
- É OBRIGATÓRIO a página de entrada no site ser

`http://all.deei.fct.ualg.pt/~a12345/MELHORIA/index.php/home`

(ou `http://all.deei.fct.ualg.pt/~a12345/MELHORIA/home`)

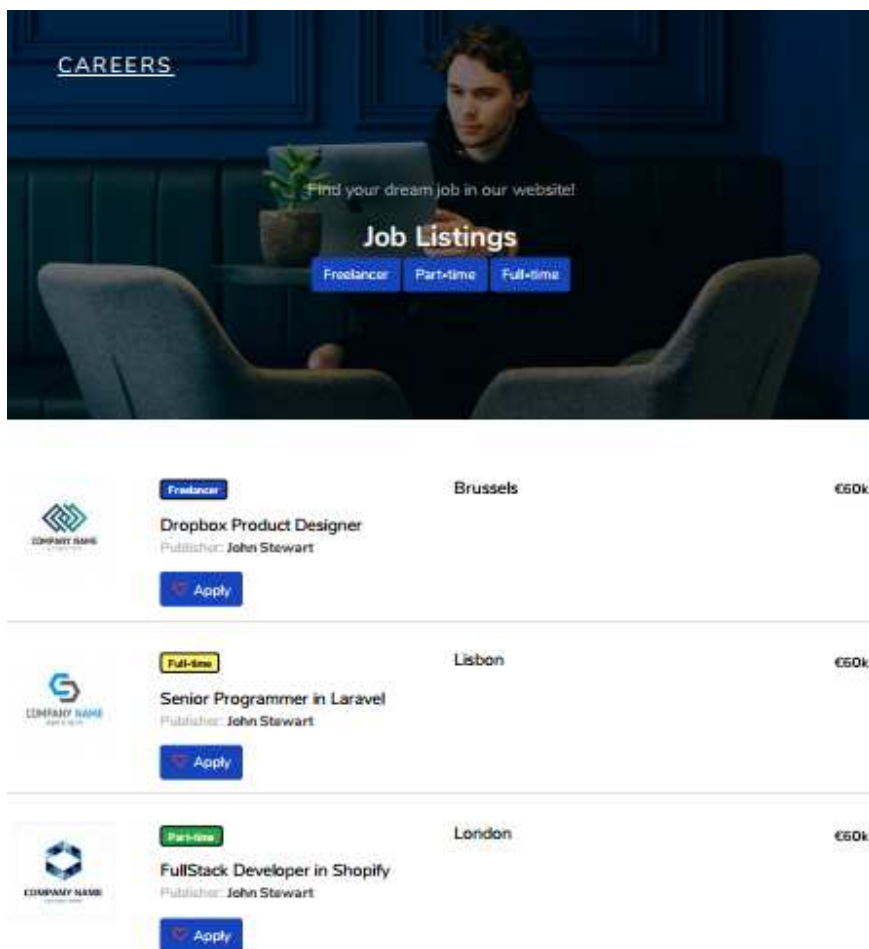
NOTA: Se vai utilizar o seu PC ou portátil como sistema de desenvolvimento é **imperativo ter instalada a versão Laravel 5.4**, caso contrário poderá ser obrigado a modificar o seu código para ele correr no servidor web do departamento!

2. [5 valores] FUNCIONALIDADE “JOB LISTINGS”

A funcionalidade “Job Listings” é página principal do site e contem uma lista dos empregos publicados.

Construa o template Blade para esta página.

SUGESTÃO: Adapte o template “job-listings.html” fornecido pela empresa de web design. Encontra este template na pasta “assets”



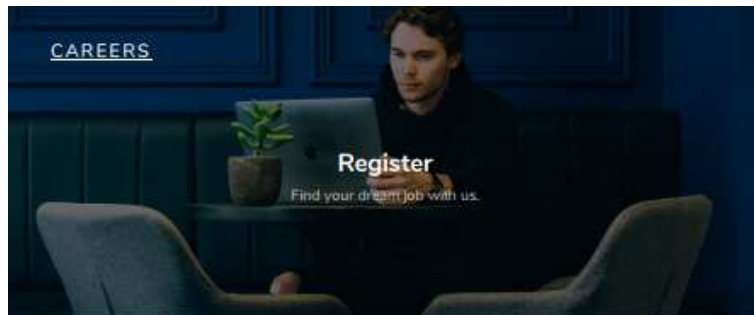
Construa o método no controlador `Careers.php` e a query correspondente no controlador `Careers_model.php` responsável por fazer uma lista de todas as ofertas de emprego existentes na tabela “jobs” da base de dados

- A página principal do site (“/job-listings”), mostra todas as ofertas de emprego
- A página principal do site tem um menu de “job categories” (Full-time, Part-time etc) criado dinamicamente a partir da tabela “jobcategories” existente na base de dados. Para cada “job category” deverá haver um hyperlink (com URL “/job-listings/1”, “/job-listings/2” etc).
- Para cada emprego mostre a imagem, a categoria, o título, o “publisher”, a localização e o salario existentes na tabela “jobs” da base de dados
- Para cada emprego deverá ser mostrado um hyperlink “Apply” (com URL “/apply/1”, “/apply/2” etc). **Nota:** a funcionalidade “Apply” só está visível para utilizadores que fizeram login.

3. [1.5 valores] FUNCIONALIDADE “REGISTER”

A funcionalidade “Register” permite registar um utilizador.

Construa o template Blade para esta página. SUGESTÃO: Adapte o template “register.html” fornecido pela empresa de web design. Encontra este template pasta “assets”.



Name

Email

Password

Password

Construa o método no controlador `Careers.php`

- Se todos os campos obrigatórios não estão preenchidos, redirecciona novamente para o formulário com uma mensagem de erro
- Se todos os campos obrigatórios estão correctamente preenchidos, e o email ainda não existe na base de dados, regista o utilizador na tabela “users” utilizando a query correspondente no controlador `Careers_model.php`. Redirecciona para a página `message` com a mensagem “Welcome to our site”. SUGESTÃO: Adapte o template “message.html” fornecido pela empresa de web design. Encontra este template pasta “assets”.

A `password_digest` é a hash utilizando o algoritmo MD5 da password:

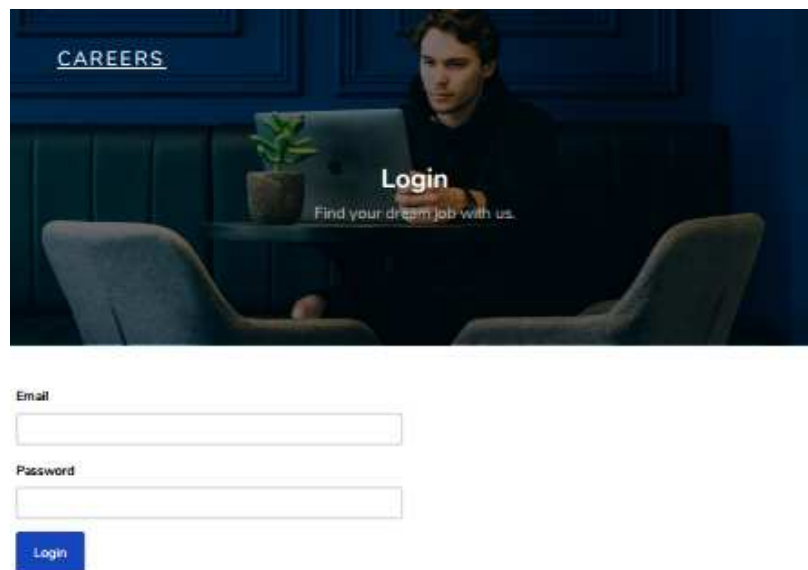
`$password_digest = substr(md5($password), 0, 32)`. Pode utilizar a função MySQL `NOW()` ou a função PHP `date("Y-m-d H:i:s")` para actualizar os campos `created_at`, `updated_at` da tabela “users”

4. [1.5 valores] FUNCIONALIDADE “LOGIN” & “LOGOUT”

A funcionalidade “Login” permite autenticar um utilizador. A funcionalidade “Logout” permite fazer reset à sessão.

Construa o template Blade correspondente.

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



Construa os métodos necessários no controlador `Careers.php` e a query correspondente no controlador `Careers_model.php`,

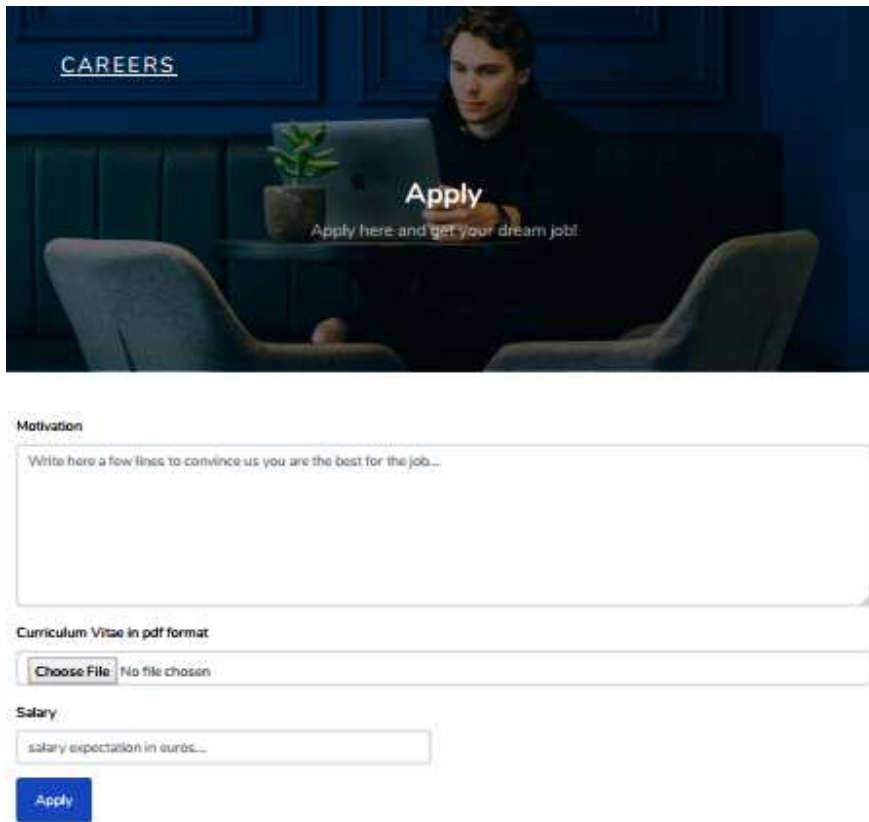
- em caso de sucesso no login:
 - regista no array da sessão os dados “id” e “name” do utilizador retirados da base de dados
 - Na página “Job Listings” o hyperlink “Login” transforma-se no hyperlink “Logout” e o hyperlink “Register” transforma-se no texto “Welcome user” (onde “user” é o nome do utilizador registado)
 - Na página “Job Listings” aparece um novo hyperlink “Publish”
 - re-direcciona para a página “Job Listings” ou (à sua escolha) para a página “message” com a mensagem “Welcome back!”
- em caso de insucesso no login
 - re-direcciona novamente para a página `login`.
 - Envia a mensagem de erro “Login failed”

5. [3 valores] FUNCIONALIDADE “APPLY”

A funcionalidade “Apply” **permite ao utilizador registado** candidatar-se a um anuncio de emprego

Construa o template Blade para esta página.

SUGESTÃO: Adapte o template “`apply.html`” fornecido pela empresa de web design. Encontra este template na pasta “assets”.



The image shows a user interface for a job application. At the top, there is a navigation menu with the word "CAREERS" highlighted. Below it, a man is sitting at a desk with a laptop, looking at the screen. The word "Apply" is prominently displayed in the center, with the tagline "Apply here and get your dream job!" underneath. Below the image, there is a form with three main sections: "Motivation" with a text area for writing a message; "Curriculum Vitae in pdf format" with a file upload button labeled "Choose File" and the text "No file chosen"; and "Salary" with a text input field for "salary expectation in euros...". At the bottom of the form is a blue "Apply" button.

Construa o método no controlador `Careers.php` e a query correspondente no controlador `Careers_model.php` responsável por introduzir a candidatura na tabela “applications” da base de dados, com os dados constantes do formulario (motivação, curriculum vitae , salario desejado)

- Para o campo motivação invente umas linhas de texto...
- No campo “Salary” ponha o salario bruto mensal desejado (em euros)...
- Pode utilizar a função MySQL `NOW()` ou a função PHP `date("Y-m-d H:i:s")` para actualizar o campo `applied_at` da tabela “jobs”

- O ficheiro seleccionado com o CV é um ficheiro em pdf.
 - O controlador **apenas** deve aceitar ficheiros com a terminação “pdf” e com um tamanho máximo de 100kbytes!
 - O nome do ficheiro deve ficar registado na base de dados no campo “cv”, **mas deve ser feito o upload do ficheiro com o CV para a pasta “uploads”** já existente com permissões de escrita.

- em caso de insucesso no registo
 - re-direcciona novamente para a página `apply` indicando os erros. Todos os campos do formulário correctamente preenchidos devem ser devolvidos

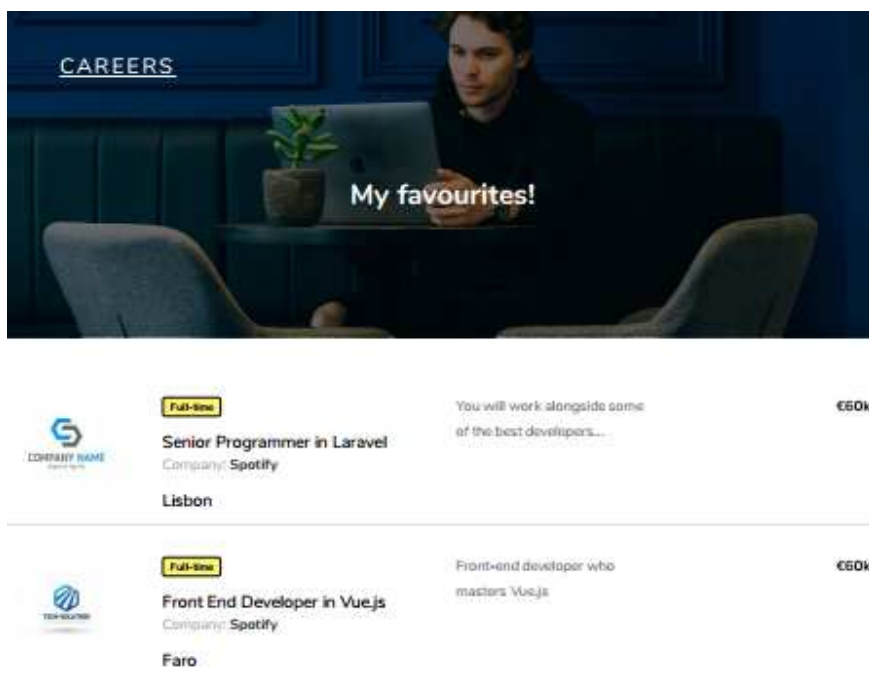
- em caso de sucesso no registo
 - redirecciona para a página “message” com a mensagem “Job application successfully submitted!”

5. [2 valores] FUNCIONALIDADE “MY JOBS”

A funcionalidade “My Jobs” **permite ao utilizador registado** ver uma listagem dos empregos a que concorreu

Construa o template Blade para esta página.

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



- Para cada emprego mostre o logotipo, a categoria, o título, a companhia, a localização, a descrição e o salario existentes na tabela “jobs” da base de dados

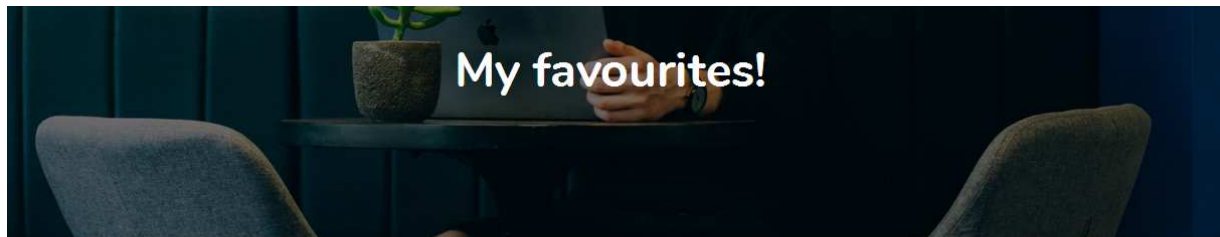
Construa o método no controlador `Careers.php` e a query correspondente no controlador `Careers_model.php` responsável por fazer a listagem

6. [5 valores] FUNCIONALIDADE “ ON MOUSE OVER”

Adicione ao template “myjobs.blade.php” uma funcionalidade em Javascript ou JQuery de forma que se o “rato” passar “algures” em cima de uma oferta de emprego todos os campos são substituídos por um novo conteúdo

- O campo logotipo é substituído pela data de submissão da candidatura
- Os campos categoria, titulo, companhia, localização, e descrição são substituídos pelo campo “motivação”
- O campo salário é substituído por um hyperlink que permite fazer o download do “CV” submetido

Se o “rato” sair de cima dessa oferta de emprego deve aparecer novamente o conteúdo dos campos originais



2020-01-18 23:25:53	I will be your best programmer!	cv2.pdf
	Full-time Front End Developer in Vue.js Company: Spotify Faro	Front-end developer who masters Vue.js €60k

No exemplo mostra-se um screenshot com o “rato” em cima da oferta de emprego Spotify em Lisboa.

NOTAS:

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

```
/users/a12345/public_html/MELHORIA
```

(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

```
a12345@daw:~$ mkdir nome_da_pasta
```

- 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 -h 10.10.23.183 db_a12345
```

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

<http://10.10.23.183/phpMyAdmin>

ANEXO 2 : estrutura da base de dados

```
--
-- Table structure for table `publishers`
--

CREATE TABLE IF NOT EXISTS `publishers` (
  `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_publishers_on_email` (`email`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Dumping data for table `publishers`
--

INSERT INTO `publishers` VALUES (1,'John
Stewart','john.stewart@gmail.com','2019-12-18 23:14:58','2019-12-18
23:14:58','42CFD85A880905FA8B680D0B53FDF0C1',NULL,NULL,NULL,NULL,NULL,
NULL),(2,'Peter Smith','peter.smith@spotify.com','2019-12-18
23:14:58','2019-12-18
23:14:58','2646F109C7263629DC7EC9FFF3662FFD',NULL,NULL,NULL,NULL,NULL,
NULL),(3,'Neil Diamond','neil.diamond@facebook.com','2019-12-18
23:16:57','2019-12-18
23:16:57','D2B58B63DDD255DC8C1041D47E01F57D',NULL,NULL,NULL,NULL,NULL,
NULL),(4,'Marc Andressen','marc.andressen@yahoo.com','2019-12-18
23:16:57','2019-12-18
23:16:57','FA0B17CEBB9EADA7AD58BF824474B99F',NULL,NULL,NULL,NULL,NULL,
NULL),(5,'Luc Bresson','luc.bresson@ebay.com','2019-12-18
23:19:37','2019-12-18
23:19:37','0BE083C2E2C9D04239483381EFA2471C',NULL,NULL,NULL,NULL,NULL,
NULL);

--
-- Table structure for table `jobcategories`
--

CREATE TABLE IF NOT EXISTS `jobcategories` (
  `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 `jobcategories`
--

INSERT INTO `jobcategories` VALUES (1,'Full-time',NULL,NULL),(2,'Part-
time',NULL,NULL),(3,'Freelancer',NULL,NULL),(4,'Internship',NULL,NULL),
(5,'Temporary',NULL,NULL);

--
-- Table structure for table `jobs`
--

CREATE TABLE IF NOT EXISTS `jobs` (
  `id` int(11) NOT NULL auto_increment,
  `cat_id` int(11) NOT NULL,
  `publisher_id` int(11) NOT NULL,
  `title` varchar(255) default NULL,
  `company` varchar(255) default NULL,
  `created_at` datetime NOT NULL,
  `description` varchar(255) default NULL,
  `location` varchar(255) default NULL,
  `salary` int(11) default NULL,
  `image` varchar(255) default NULL,
  PRIMARY KEY (`id`),
  CONSTRAINT FOREIGN KEY (`cat_id`) REFERENCES `jobcategories` (`id`),
  CONSTRAINT FOREIGN KEY (`publisher_id`) REFERENCES `publishers` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Dumping data for table `jobs`
--

INSERT INTO `jobs` VALUES (1,3,1,'Dropbox Product Designer','Software
startup','2019-12-18 23:25:53','Developing cutting-edge digital
solutions','Barcelona',50,'1.jpg'),(2,2,2,'Senior Programmer in
Laravel','Spotify ','2019-12-18 23:25:53','You will work alongside some of
the best developers','Faro',70,'2.jpg'),(3,1,3,'FullStack Developer in
Shopify','Facebook','2019-12-18 23:30:51','Do you look beyond the code and
back-end solution?','Brussels',80,'3.jpg'),(4,4,4,'Dropbox Product
Designer','Yahoo','2019-12-18 23:30:51','Back-end developer with expertise
in Java','Dublin',60,'4.jpg'),(5,5,5,'Front End Developer in
Vue.js','Ebay','2019-12-18 23:33:15','Front-end developer who masters
JavaScript, Angular, React, Vue','London',50,'5.jpg');

--
-- Table structure for table `users`
--

CREATE TABLE IF NOT EXISTS `users` (
  `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 `applications`
--

CREATE TABLE IF NOT EXISTS `applications` (
  `id` int(11) NOT NULL auto_increment,
  `job_id` int(11) NOT NULL,
  `user_id` int(11) NOT NULL,
  `motivation` varchar(255) default NULL,
  `salary` int(11) default NULL,
  `cv` varchar(255) default NULL,
  `applied_at` datetime NOT NULL,
  PRIMARY KEY (`id`),
  CONSTRAINT FOREIGN KEY (`job_id`) REFERENCES `jobs` (`id`),
  CONSTRAINT FOREIGN KEY (`user_id`) REFERENCES `users` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```