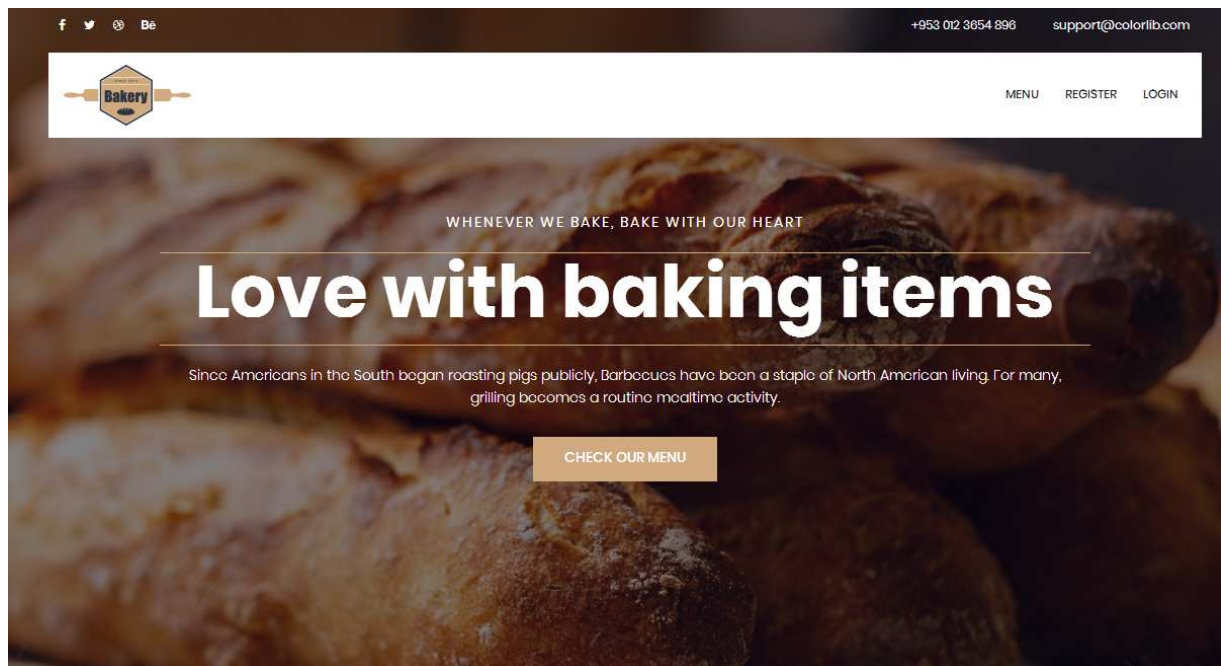


LER O ENUNCIADO ATÉ AO FIM ANTES DE COMEÇAR!

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 padaria que permite fazer encomendar online pizzas e outros produtos a utilizadores registados. O site tem as seguintes páginas:

1. “**Home**” é a página de rosto do site (já construída);
2. “**Menu**” é a página principal do site;
3. “**Register**” para registo de utilizadores;
4. “**Login**” para o login de utilizadores;

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 *CodeIgniter* e *Smarty*, apenas das páginas “Menu”, “Register” e “Login” 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.** Faça download para a pasta “ESPECIAL”\*, do código do site web

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

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

```
wget http://10.10.23.13/DAW/especial30042019.tgz
```

```
a12345@daw:~/public_html$ tar xzf especial30042019.tgz
```

**C.** Ponha as permissões de escrita na pasta `template_c`:

```
chmod -R g+w ~/public_html/ESPECIAL/application/views/templates_c
```

**D.** Actualize a base URL in `config.php`

```
a12345@daw:~$ nano ~/public_html/ESPECIAL/application/config/config.php
```

```
$config['base_url'] = '~/a12345/ESPECIAL';
```

### NOTAS:

- **substitua “12345” pelo seu número de aluno!**
- O seu site deve ser portátil; utilize sempre as funções `base_url()` e `site_url()` para gerar hyperlinks!

**E.** Crie a sua cópia da base de dados

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

```
mysql -u a12345 -p db_a12345 < ~/public_html/ESPECIAL/database.SQL
```

e actualize as credenciais em `database.php`

```
a12345@daw:~$ nano ~/public_html/ESPECIAL/application/config/database.php
```

```
$db['default'] = array(  
    'dsn' => '',  
    'hostname' => '10.10.23.183',
```

---

\* a pasta “ESPECIAL” é criada automaticamente pelo comando `git clone`

```
'username' => 'a12345',  
'password' => '*****',  
'database' => 'db_a12345',  
'dbdriver' => 'mysqli',
```

#### 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

#### F. Verifique que o site é visível no URL

```
http://10.10.23.183/~a12345/ESPECIAL/index.php/bakery
```

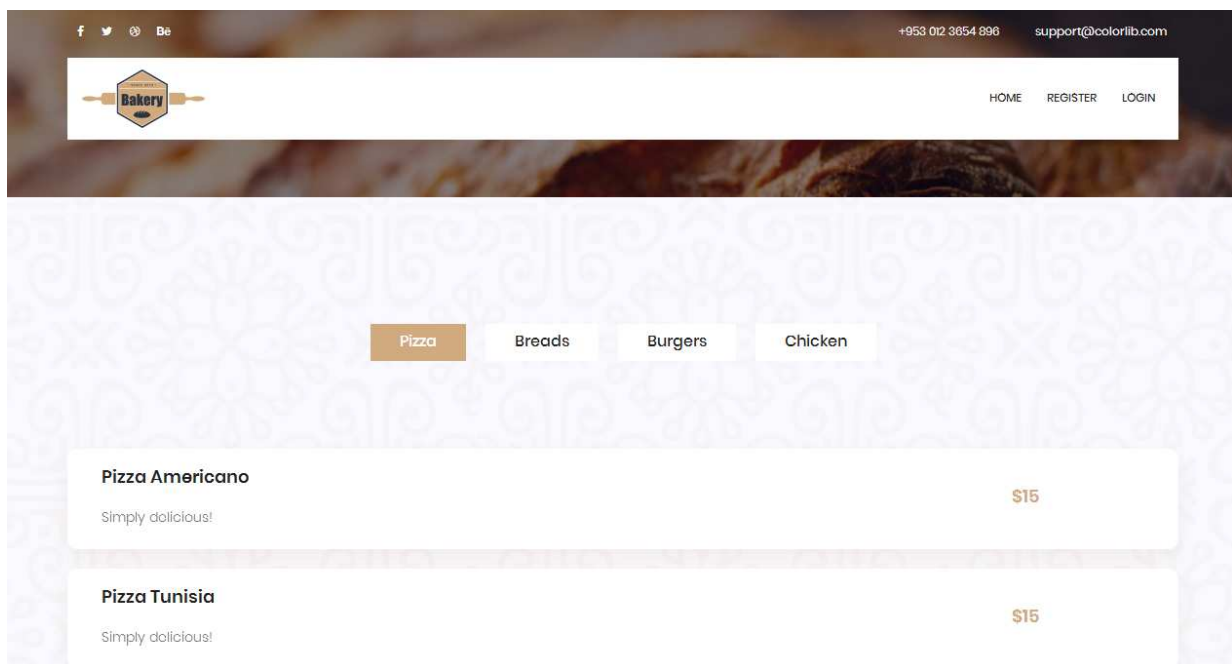
## FUNCIONALIDADE “MENU”

A funcionalidade “Menu” é página principal do site e contém uma lista dos produtos disponíveis na padaria.

### 1. [5 valores]

Construa o template Smarty “menu.tpl”.

SUGESTÃO: Adapte o template “menu.html” fornecido pela empresa de web design. Encontra este template na pasta “views/templates”



Construa o método `menu()` no controlador `Bakery.php` e a query correspondente no controlador `Bakery_model.php` responsável por fazer uma lista dos produtos existentes na tabela “products” da base de dados

- As categorias de produtos (“Pizza”, “Breads”, ...) encontram-se na tabela “categories” da base de dados;
- Na página “menu”, as categorias de produtos (“Pizza”, “Breads”, ...) são hyperlinks com dados embutidos (“/menu/1”, “/menu/2” etc) que permitem mostrar uma lista de produtos da categoria seleccionada

- A página de entrada no site, por omissão, mostra os produtos da categoria "Pizza"
- Preencha a lista com os campos "name", "description", "price" existentes na tabela "products"
- Ignore os campos vazios, quer na tabela "categories", quer na tabela "products"

## FUNCIONALIDADE “REGISTER”

A funcionalidade “Register” permite registar um utilizador.

### 2. [2.5 valores]

Construa o template Smarty “register.tpl”.

SUGESTÃO: Adapte o template “register.html” fornecido pela empresa de web design. Encontra este template na pasta “views/templates”. Note que este formulário apenas contém as tags e atributos HTML necessários para o layout, pelo que se encontra incompleto!

Construa o método `register()` no controlador `Bakery.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 “customers” utilizando a query correspondente no controlador `Bakery_model.php`.

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

`substr(md5($_POST['pass1_utilizador']), 0, 32)`. Pode utilizar a função MySQL `CURDATE()` ou a função PHP `date("Y-m-d H:i:s")` para actualizar os campos `created_at`, `updated_at` da tabela “customers”

## **FUNCIONALIDADE “LOGIN” & “LOGOUT”**

A funcionalidade “Login” permite autenticar um utilizador.

### **3. [2.5 valores]**

Construa o template Smarty “login.tpl”.

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

Construa o métodos `login()` e `login_action()` no controlador `Bakery.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, o método `login_action()` e a query correspondente no controlador `Bakery_model.php`,
- em caso de sucesso:
  - regista no array da sessão `$_SESSION` os dados “id” e “name” do utilizador retirados da base de dados
  - re-direcciona para o portal de entrada no site
  - 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) nas paginas “index” e “menu”
  - À direita de cada produto é visível um hyperlink “Order” com o id do produto embutido (`/order/1`, `/order/2` etc)\*
- em caso de insucesso
  - re-direcciona novamente para a página `login`.
  - Envia uma mensagem de erro “Login failed”

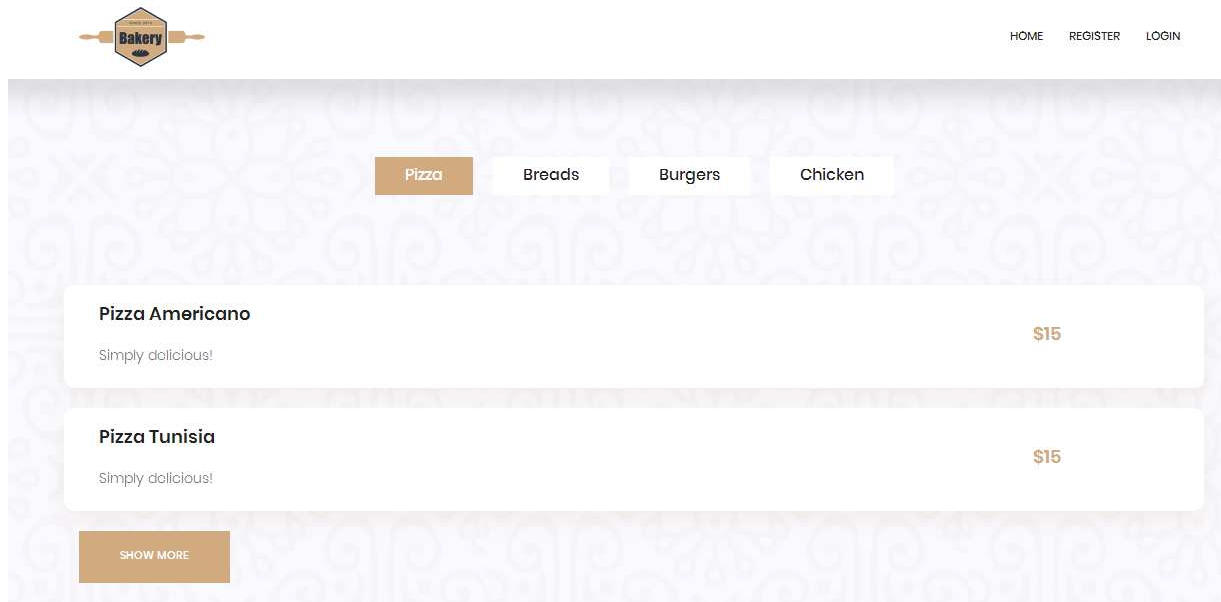
Construa o método `logout()` que encerra a sessão no controlador `Bakery.php`

---

\* A construção do método `order()` NÃO faz parte deste exame

## FUNCIONALIDADE “SHOW MORE”

A funcionalidade “Show More” permite mostrar os produtos da padaria em incrementos de dois.



### 4. [10 valores]

Quando a página “Menu” é carregada é feito o *download* de uma página HTML com apenas os primeiros dois produtos de uma determinada categoria .

- Altere o template `menu.tpl` e adicione um hyperlink “SHOW MORE” para uma função em AJAX que, sempre que se clica no hyperlink, mostra mais dois produtos.
- A informação deve ser transferida do servidor em JSON
- Altere o método `menu()` no controlador `Bakery.php`, e/ou a query correspondente em `Bakery_model.php`, para a página HTML inicial apenas conter os dois primeiros produtos de uma determinada categoria que constam da tabela “products”
- Construa o método `get2more()` no controlador `Bakery.php` (e a query correspondente em `Bakery_model.php`) que é o interlocutor no servidor da função AJAX

SUGESTÃO: Tem à sua disposição elementos de layout (botões etc) em “`elements.html`” fornecido pela empresa de web design. Encontra este template na pasta “`views/templates`”.



## NOTAS:

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

```
/users/a12345/public_html/ESPECIAL
```

(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-- ---). **Verifique que o site fica operacional.**

- 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
```

---

\* NÃO faça o upload de pastas! Se precisar de criar uma pasta faça-o com o comando  
a12345@daw:~\$ mkdir nome\_da\_pasta

## ANEXO 1 Acesso à base de dados MySQL

- O acesso à base de dados MySQL pode ser feita utilizando um cliente gráfico à sua escolha (por exemplo <http://www.heidisql.com/>),

ou 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://all.deei.fct.ualg.pt/phpMyAdmin>
- <http://10.10.23.183/phpMyAdmin>

## ANEXO 2 : estrutura da base de dados

```
--
-- Table structure for table `customers`
--

CREATE TABLE IF NOT EXISTS `customers` (
  `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 `categories`
--

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

INSERT INTO `categories` VALUES
(1, 'Pizza', NULL, NULL), (2, 'Breads', NULL, NULL), (3, 'Burgers', NULL, NULL), (4, 'C
hicken', NULL, NULL);

--
-- Table structure for table `products`
--

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

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

```
--
-- Dumping data for table `products`
--
```

```
INSERT INTO `products` VALUES (1,1,'Pizza Americano','Simply
delicious',15,NULL),(2,1,'Pizza Tunisia','Simply
wonderful!',20,NULL),(3,1,'Beef lovers Pizza','For meat
lovers!',20,NULL),(4,1,'Chicken lovers Pizza','Chicken
Chicken!',10,NULL),(5,1,'Meatball Pizza','No teeth
required!',12,NULL),(6,1,'Bakery special Pizza','Our
special!',25,NULL),(7,1,'Pizza Alfredo','Typical
italian!',13,NULL),(8,1,'Cheese lovers Pizza','For
vegetarians!',9,NULL),(9,2,'White bread','Simply
delicious!',2,NULL),(10,2,'Whole grain bread','So
healthy!',4,NULL),(11,3,'Bakery burger','Our
special!',10,NULL),(12,3,'Cheese Burger','So
tasty!',12,NULL),(13,3,'Vegetarian burger','So
healty!',8,NULL),(14,3,'Spicy Burger!','Hot hot
hot!',9,NULL),(15,4,'Chicken wings','So tasty!',10,NULL),(16,4,'Frango da
Guia','Algarve special!',20,NULL);
```