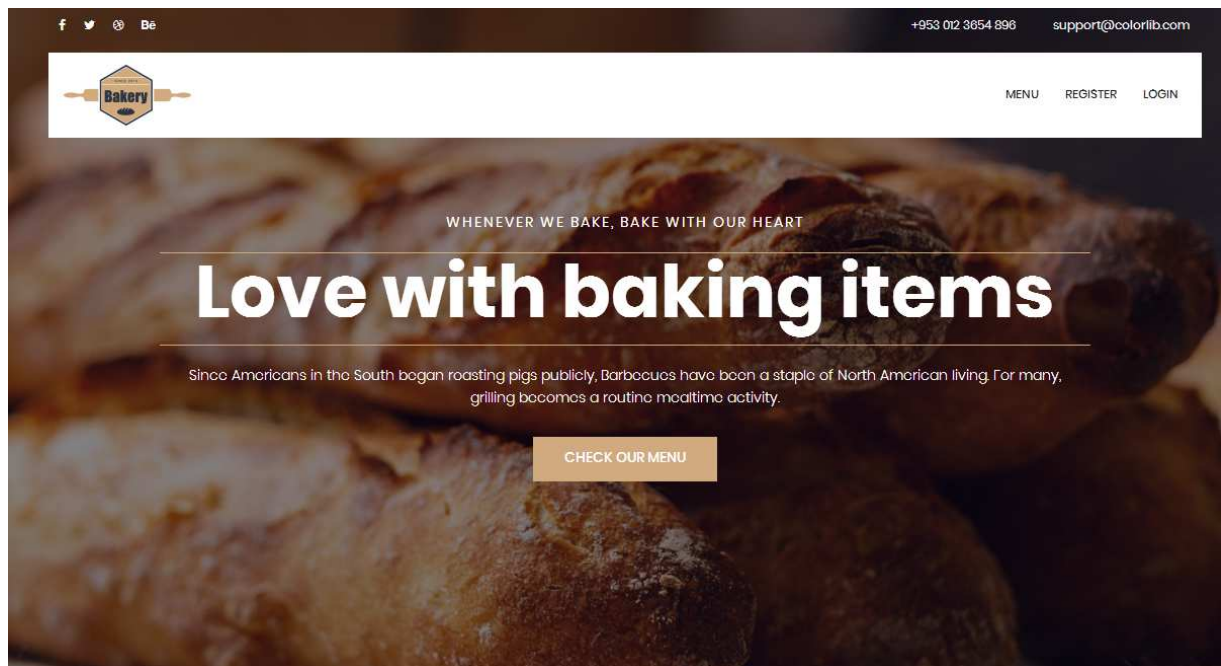


LER O ENUNCIADO ATÉ AO FIM ANTES DE COMEÇAR!

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;
2. “**Menu**” é a página principal do site;
3. “**Register**” para registo de utilizadores;
4. “**Login**” para o login de utilizadores;
5. “**Orders**” é página com o registo das encomendas do utilizador

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 10.10.23.184¹

```
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/EXTRAORDINARIO2
```

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

```
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/EXTRAORDINARIO/public/index.php/bakery
```

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

² a pasta EXTRAORDINARIO já existe na sua área pessoal

Deverá receber uma página web de boas-vindas

Hello BakeryController! ✓

This friendly message is coming from:

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

F. OPCIONAL: se preferir³ 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/EXTRAORDINARIO.git
user@meuPC:~/EXTRAORDINARIO$ 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/EXTRAORDINARIO/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 "EXTRAORDINARIO" no servidor web do departamento disponibilizado para o efeito: <http://daw.deei.fct.ualg.pt>
- O controlador principal tem que estar no ficheiro "BakeryController.php"

³ 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 “Bakery_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/EXTRAORDINARIO/public/index.php/bakery>

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

```
* @Route("/bakery name="bakery")
* @Route("/menu/{cat_id?}", name="menu")
* @Route("/register", name="register")
* @Route("/login", name="login")
* @Route("/logout", name="logout")
* @Route("/order/{prod_id}", name="order")
* @Route("/orders", name="orders")
```

Pode ver um demo do site no URL

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

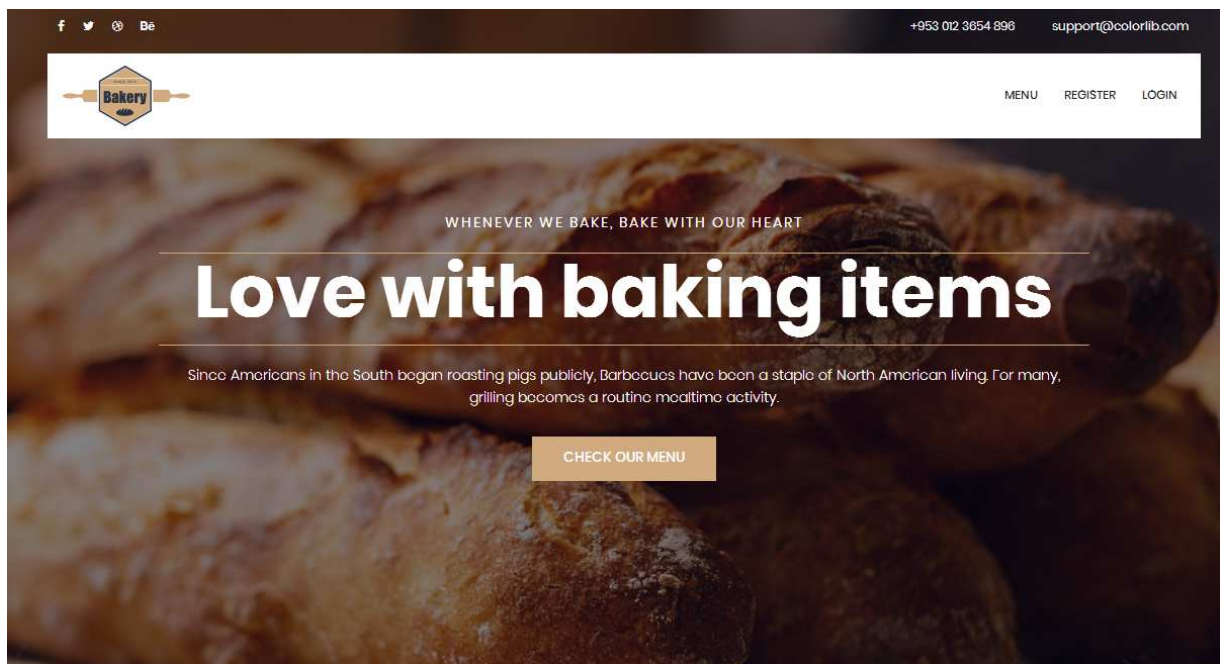
FUNCIONALIDADE “HOME”

A funcionalidade “Home” é página de rosto do site

1. [2 valores]

Construa o template Twig para esta página.

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



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

O botão “CHECK OUR MENU” é um hyperlink (“/menu”) que permite mostrar a página com os produtos disponíveis para encomenda

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.

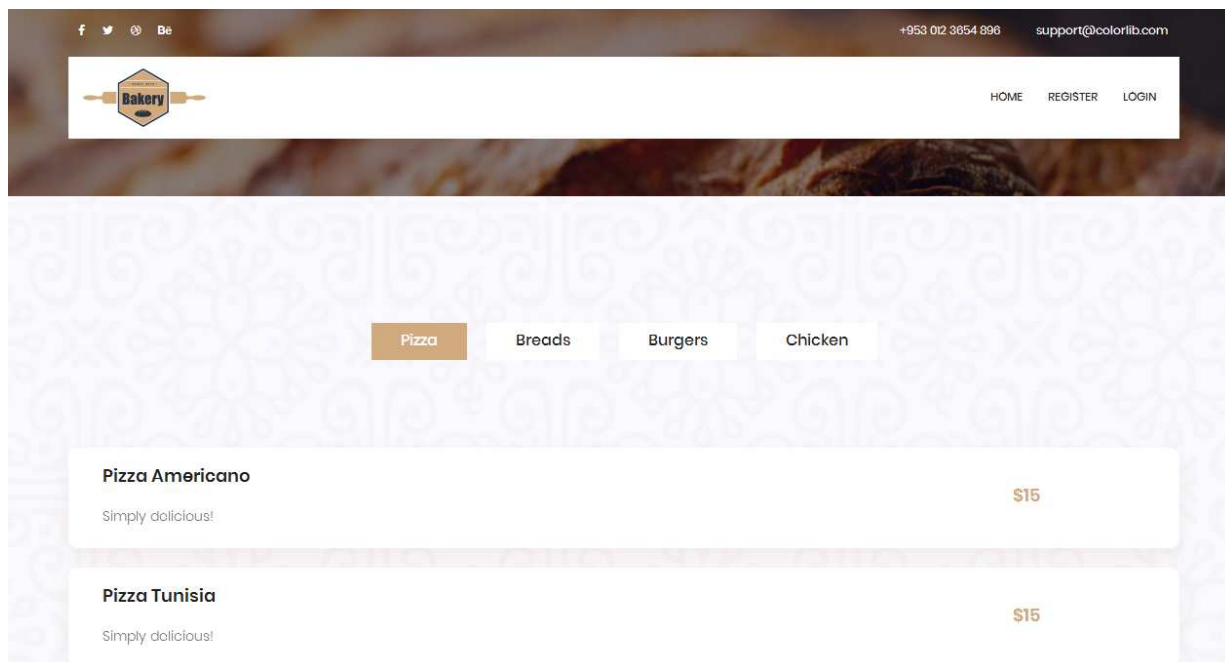
FUNCIONALIDADE “MENU”

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

2. [3 valores]

Construa o template Twig para esta página.

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



Construa o método no controlador `BakeryController.php` e a query correspondente no controlador `Bakery_modelController.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 todos os produtos
- Preencha a lista de produtos com os campos “name”, “description”, “price” existentes na tabela “products”
- Ignore os campos vazios, quer na tabela “categories”, quer na tabela “products”

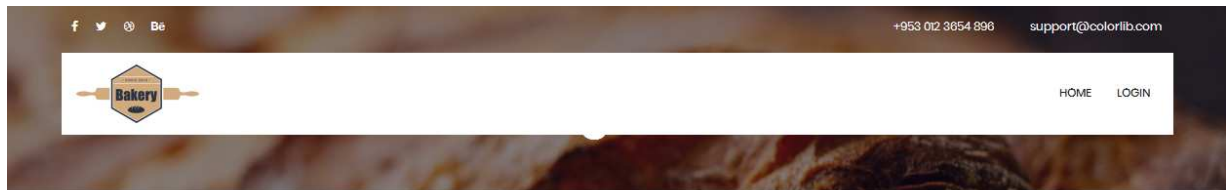
FUNCIONALIDADE “REGISTER”

A funcionalidade “Register” permite registar um utilizador.

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 na pasta “templates”.



Construa o método no controlador `BakeryController.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`.

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. A “semente” é gerada de forma pseudoaleatória utilizando os últimos 4 algarismos da função `time()`. Sugere-se a seguinte estratégia

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


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 "customers"

FUNCIONALIDADE “LOGIN” & “LOGOUT”

A funcionalidade “Login” permite autenticar um utilizador.

4. [3 valores]

Construa o template Twig correspondente.

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

Construa o método necessário no controlador `BakeryController.php`

- Se todos os campos obrigatórios não estão preenchidos, redirecciona novamente para o formulário com uma mensagem de erro
- 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 a página “menu” do 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 páginas “index” e “menu”
 - À direita de cada produto na página “menu” é visível um hyperlink “Order” com o id do produto embutido (`/order/1`, `/order/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 que encerra a sessão no controlador `BakeryController.php`

FUNCIONALIDADE “ORDER”

A funcionalidade “Order” permite a um utilizador registado encomendar um produto.

5. [2 valores]

Construa o método no controlador `BakeryController.php` e a query correspondente no controlador `Bakery_modelController.php` responsável por introduzir o produto encomendados na tabela “orders” da base de dados

FUNCIONALIDADE “MY ORDERS”

A funcionalidade “My Orders” permite a um utilizador registado ver uma listagem dos produtos que encomendou.

6. [3 valores]

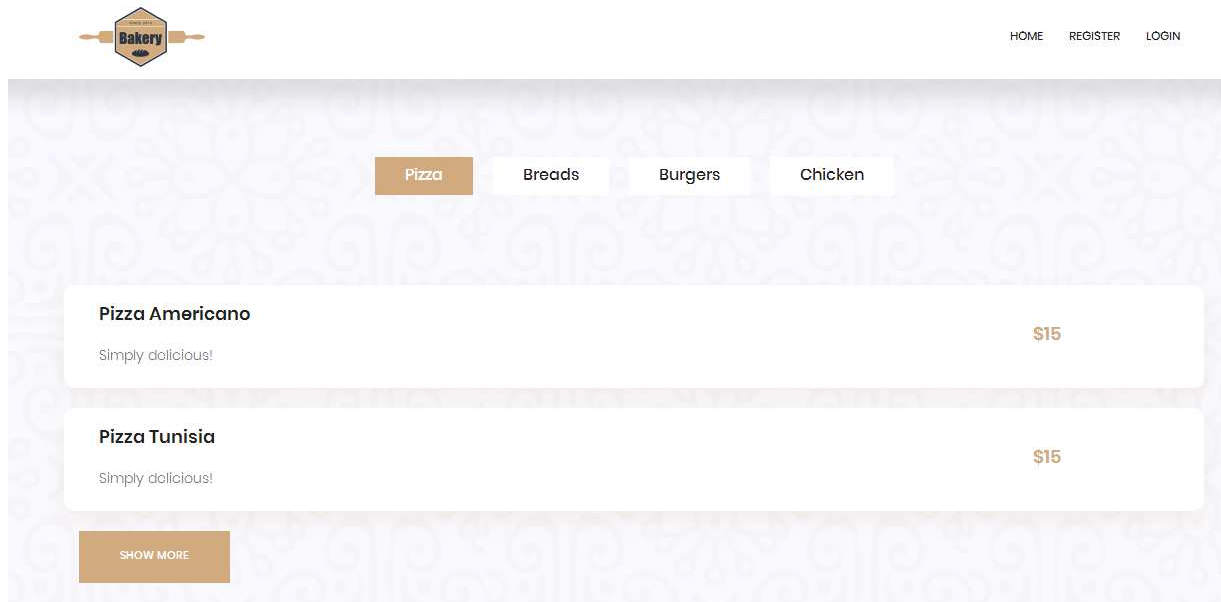
Construa o template Twig correspondente.

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

Construa o método no controlador `BakeryController.php` e a query correspondente no controlador `Bakery_modelController.php` responsável por fazer a listagem dos produto encomendados pelo utilizador.

FUNCIONALIDADE “SHOW MORE”

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



5. [5 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 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 no controlador `BakeryController.php`, e/ou a query correspondente em `Bakery_modelController.php`, para a página HTML inicial apenas conter os dois primeiros produtos de uma determinada categoria que constam da tabela “products”
- Construa o método no controlador `BakeryController.php` (e a query correspondente em `Bakery_modelController.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 “templates”.

NOTAS:

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

```
/users/a12345/public_html/EXTRAORDINARIO
```

(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 10.10.23.184 db_a12345
```

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

- <http://daw.deei.fct.ualg.pt/phpMyAdmin>
- <http://10.10.23.184/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);

--
-- Table structure for table `orders`
--

CREATE TABLE IF NOT EXISTS `orders` (
  `id` int(11) NOT NULL auto_increment,
  `user_id` int(11) NOT NULL,
  `prod_id` int(11) NOT NULL,
  `created_at` datetime NOT NULL,
  PRIMARY KEY (`id`),
  CONSTRAINT FOREIGN KEY (`user_id`) REFERENCES `customers` (`id`),
  CONSTRAINT FOREIGN KEY (`prod_id`) REFERENCES `products` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```