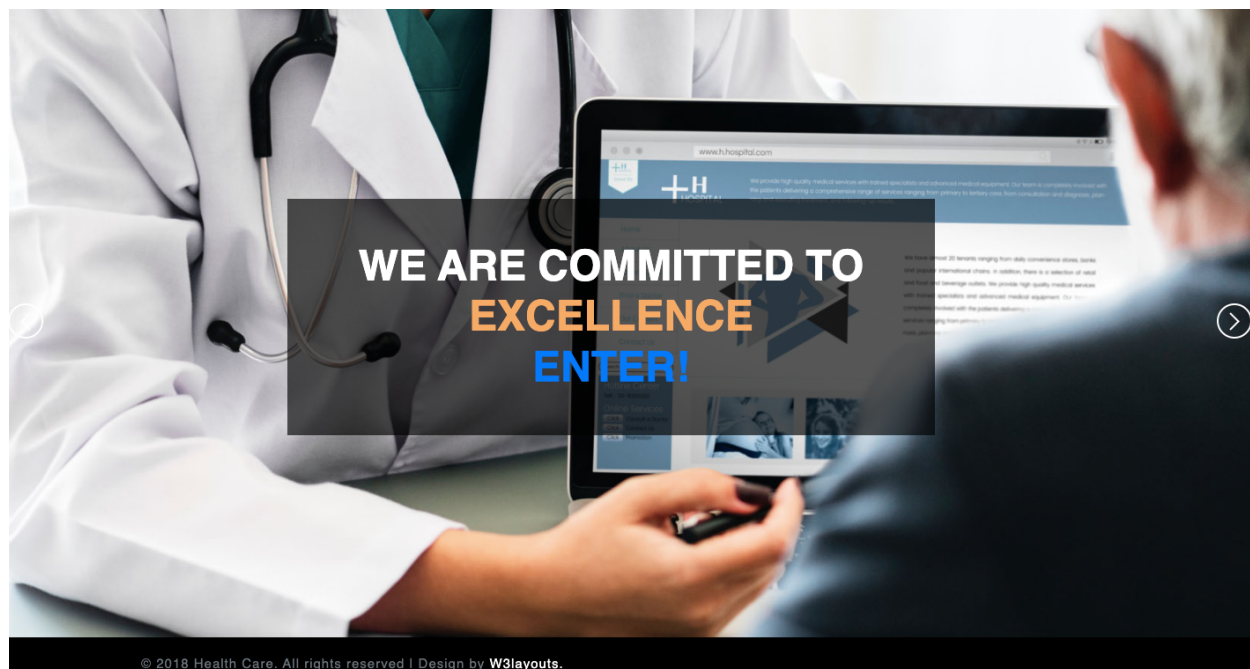


LER O ENUNCIADO ATÉ AO FIM ANTES DE COMEÇAR!

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 clinica médica que permite a utilizadores registados fazer reservas para consultas. O site tem as seguintes páginas:

1. “**Home**” é a página de rosto do site;
2. “**Departments**” é a página principal do site;
3. “**Register**” para registo de utilizadores;
4. “**Login**” para o login de utilizadores;
5. “**Book**” permite ao utilizador registado marcar uma consulta;
6. “**MyBookings**” permite ao utilizador registado ver as suas reservas;
7. “**Message**” permite enviar mensagens;

É dado acesso a um conjunto de templates HTML construídos em *Bootstrap* fornecidos pela empresa de web design **w3layouts.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. Actualize o ficheiro .env

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

```
a12345@daw2:~$ cd public_html/EXAME2
```

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

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

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

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

```
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/EXAME/public/index.php/clinic
```

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 EXAME já existe na sua área pessoal

Deverá receber uma página web de boas-vindas

Hello ClinicController!

This friendly message is coming from:

- Your controller at `src/Controller/ClinicController.php`
- Your template at `templates/clinic/index.html.twig`

F. OPCIONAL: se prefere desenvolver o site utilizando o PC/portátil como ambiente de desenvolvimento 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/EXAME.git
user@meuPC:~/EXAME$ 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/EXAME/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** (`DROP TABLE nome_da_tabela`)
- A aplicação deve correr obrigatoriamente na área pessoal do aluno na pasta "EXAME" no servidor web do departamento disponibilizado para o efeito: <http://daw.deei.fct.ualg.pt>

- O controlador principal tem que estar no ficheiro "ClinicController.php"
- A classe com as funções de acesso à base de dados tem que estar no ficheiro "Clinic_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/EXAME/public/index.php/clinic>

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

```
* @Route("/clinic name="clinic")
* @Route("/departments", name="departments")
* @Route("/register", name="register")
* @Route("/login", name="login")
* @Route("/logout", name="logout")
* @Route("/book", name="book")
* @Route("/mybookings", name="myBookings")
* @Route("/message", name="message") OPCIONAL
```

Pode ver um demo do site no URL

<http://daw.deei.fct.ualg.pt/~a12345/EXAME/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/EXAME/demo), bem como os componentes imagens, css, e javascript utilizados.



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

“JOIN US/COME IN/ENTER” é um hyperlink para a página (“/departments”) que permite mostrar as especialidades dos médicos da clínica

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.

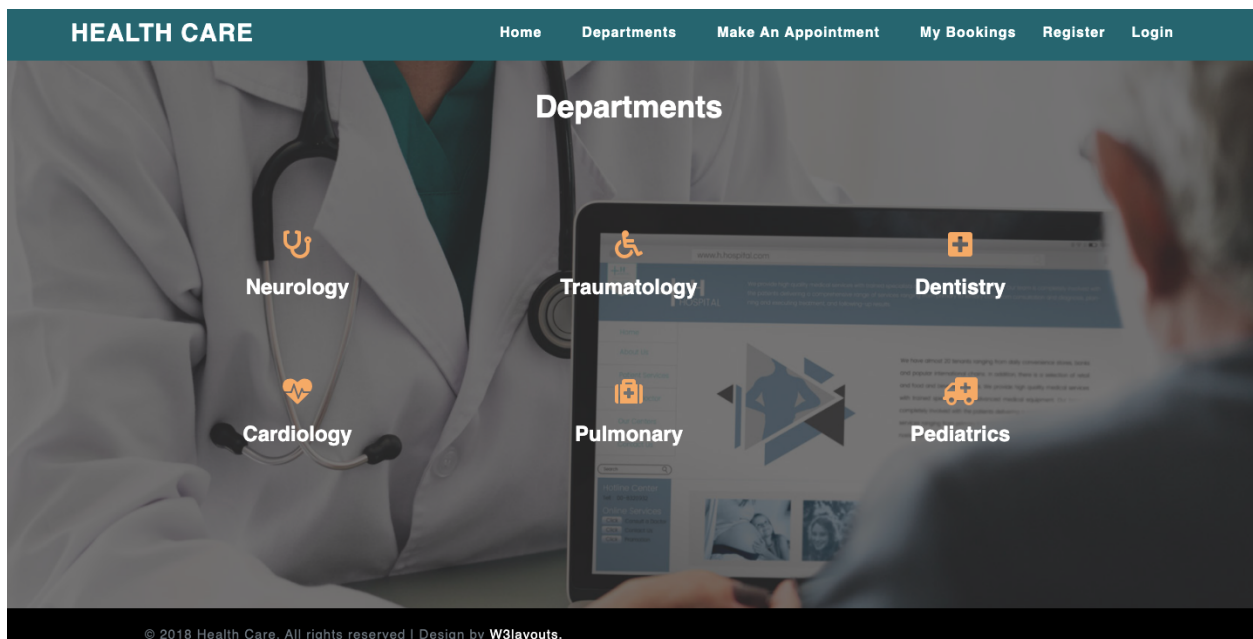
2. [4 valores] FUNCIONALIDADE “DEPARTMENTS”

A funcionalidade “Departments” é página principal do site e contem uma lista das especialidades médicas disponíveis para reserva.

Construa o template Twig para esta página.

SUGESTÃO: Adapte o template “departments.html” fornecido pela empresa de web design.

Encontra este template na pasta “demo”



Construa o método no controlador `ClinicController.php` e a query correspondente no controlador `ClinicController_model.php` responsável por fazer uma lista das especialidades médicas existentes na tabela “departments” da base de dados

- Para cada especialidade mostre o icon e o nome (“name”, “icon”, existentes na tabela “departments”)

- “REGISTER”, “LOGIN” são hyperlinks apenas visíveis caso o utilizador seja anonimo. Caso o utilizador tenha feito login, transformam-se em “LOGOUT”, “BOOK” (“/logout”, “/book”), fica visível um outro hyperlink “MY BOOKINGS” (“/mybookings”), bem como aparece o texto “Welcome user” onde *user* é o login do utilizador

FUNCIONALIDADE “REGISTER”

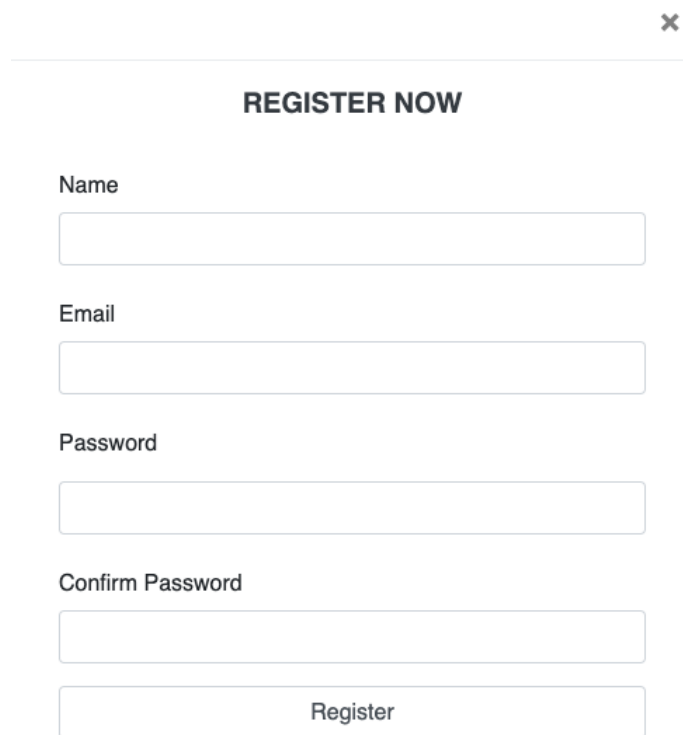
A funcionalidade “Register” permite registar um utilizador.

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 na pasta “demo”



The image shows a registration form titled "REGISTER NOW" with a close button (x) in the top right corner. The form contains four input fields: "Name", "Email", "Password", and "Confirm Password". Below the "Confirm Password" field is a "Register" button.

O formulário faz uma validação dos dados simples em javascript

Construa o método responsável pela acção deste formulário no controlador `ClinicController.php` e a query correspondente no controlador `ClinicController_model.php`

- Se o email ainda não existe na base de dados, regista o utilizador na tabela “users” utilizando a query correspondente no controlador `ClinicController_model.php`.

A password é a *hash* utilizando o algoritmo MD5 da password:

```
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"

Se o registo falhar deve enviar uma mensagem de erro para o utilizador e retornar o formulário com os campos corretos já previamente preenchidos.

FUNCIONALIDADE “LOGIN” & “LOGOUT”

A funcionalidade “Login” permite autenticar um utilizador.

4. [5 valores]

Construa o template Twig para esta página.

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

Construa os métodos necessários no controlador `ClinicController.php` e a query correspondente no controlador `ClinicController_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 “Departments”
 - No menu o hyperlink “Login” transforma-se no hyperlink “Logout” e
 - No menu o hyperlink “Register” desaparece
 - Aparece no menu o texto “Welcome user!” (onde “user” é o nome do utilizador registado)
 - No menu aparece um novo hyperlink “Book”
 - No menu aparece um novo hyperlink “My Bookings”
- em caso de insucesso no login
 - Envia uma mensagem de erro “Login failed: wrong credentials” e retorna o formulário ao utilizador

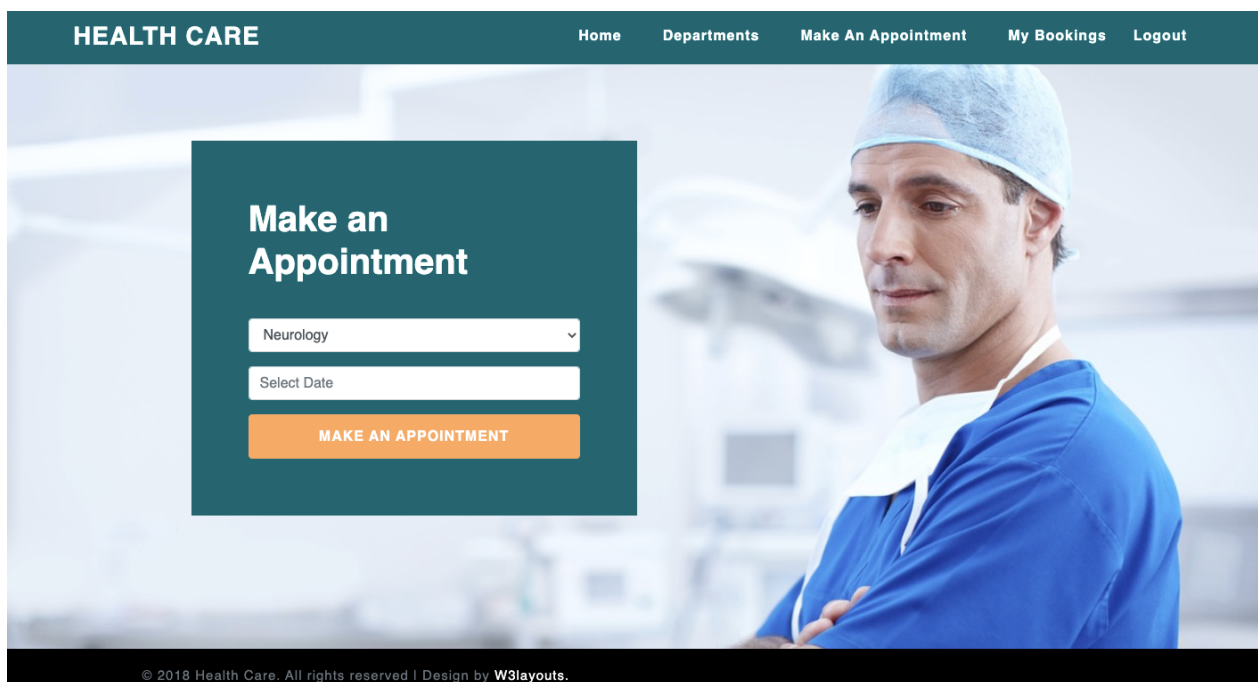
Construa o método “logout ()” que encerra a sessão no controlador `ClinicController.php`

5. [4 valores] FUNCIONALIDADE “Book”

A funcionalidade “Book” permite ao utilizador registado fazer uma reserva de uma consulta. **Nota: Overbooking não é permitido.** Fazer duas reservas para a mesma especialidade e para o mesmo dia é overbooking.

Construa o template Twig para esta página.

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

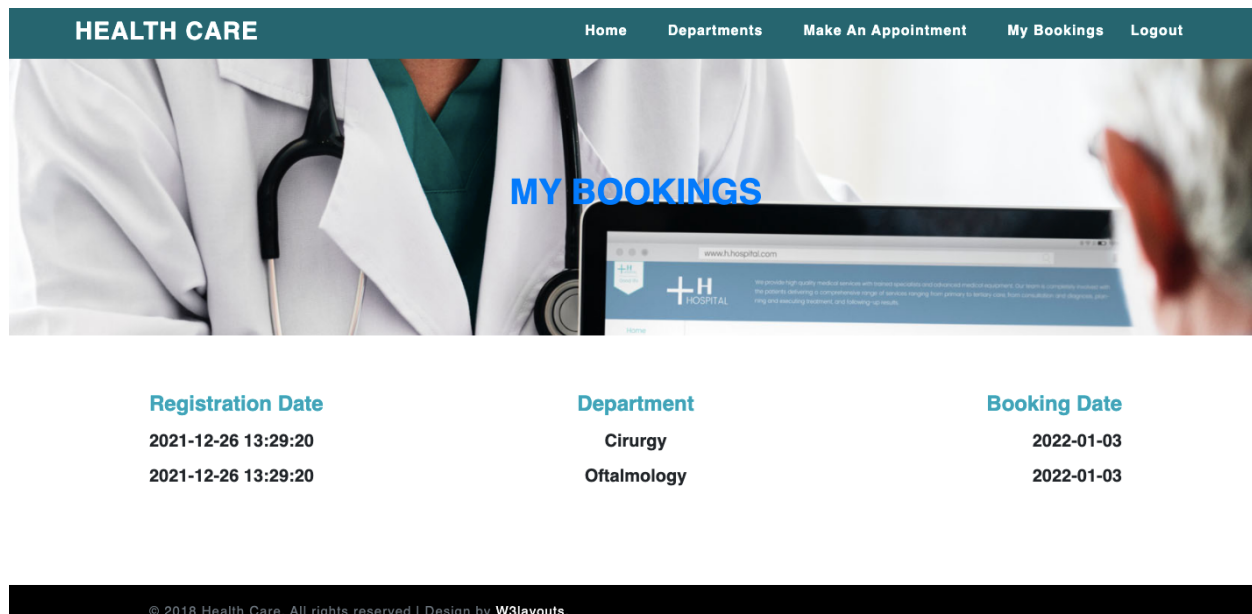


Construa os métodos no controlador `ClinicController.php` e as queries correspondente no controlador `ClinicController_model.php` responsáveis por fazer

- a geração das especialidades médicas disponíveis no *dropdown* do formulário a partir do conteúdo existente na tabela "departments" da base de dados (a query já foi realizada na pergunta 2)
- o registo da consulta na tabela "apointments". Pode utilizar a função MySQL `NOW()` ou a função PHP `date("Y-m-d H:i:s")` para actualizar o campo `created_at`
- Redireccione o utilizador para a página "My Bookings". OPCIONAL: Envie a mensagem "Success, your booking is registered"

6. [3 valores] FUNCIONALIDADE “MY Bookings”

A funcionalidade “My Bookings” permite ao utilizador registado consultar todos os pedidos de consulta que efectuou



(O template mostra duas reservas que **não** são overbooking: são duas reservas para especialidades diferentes no mesmo dia.)

Construa o template Twig para esta página.

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

Construa o método no controlador `ClinicController.php` e a query correspondente no controlador `Clinic_modelController.php` responsável por fazer uma lista das reservas existentes na tabela “appointments” da base de dados do utilizador que fez login.

NOTAS:

- Caso tenha trabalhado no seu PC/portátil, **é obrigatório fazer o upload de todos os ficheiros** para a pasta "EXAME" no seu site web pessoal

```
/users/a12345/public_html/EXAME
```

(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-- ---).

- 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
```

- Se receber uma mensagem de erro informando o acesso negado à pasta "**var/cache/dev**",

```
Warning:
file_put_contents(/users/a12345/public_html/EXAME/var/cache/dev/srcApp_KernelDevDebugContainerD
eprications.log): failed to open stream: Operation not permitted
```

terá que limpar o "cache" com o comando

```
a12345@daw2:~/public_html/EXAME$ rm -rf var/cache
```

OU (em alternativa) com o comando

```
a12345@daw2:~/public_html/EXAME$ php bin/console cache:clear
```

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 `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 (`email`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Table structure for table `departments`
--

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

--
-- Dumping data for table `departments`
--

INSERT INTO `departments` VALUES (1,'Neurology','fa fa-
stethoscope'), (2,'Traumatology','fa fa-wheelchair'), (3,'Dentistry','fa fa-
plus-square'), (4,'Cardiology','fa fa-heartbeat'), (5,'Oncology','fa fa-
medkit'), (6,'Pediatrics','fa fa-ambulance');

--
-- Table structure for table `appointments`
--

CREATE TABLE IF NOT EXISTS `appointments` (
  `id` int(11) NOT NULL auto_increment,
  `booking_date` date NOT NULL,
  `created_at` datetime NOT NULL,
  `user_id` int(11) NOT NULL,
```

```
`department_id` int(11) NOT NULL,  
PRIMARY KEY (`id`),  
UNIQUE KEY (`booking_date`,`user_id`,`department_id`)  
CONSTRAINT FOREIGN KEY (`user_id`) REFERENCES `users` (`id`),  
CONSTRAINT FOREIGN KEY (`department_id`) REFERENCES `departments` (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```