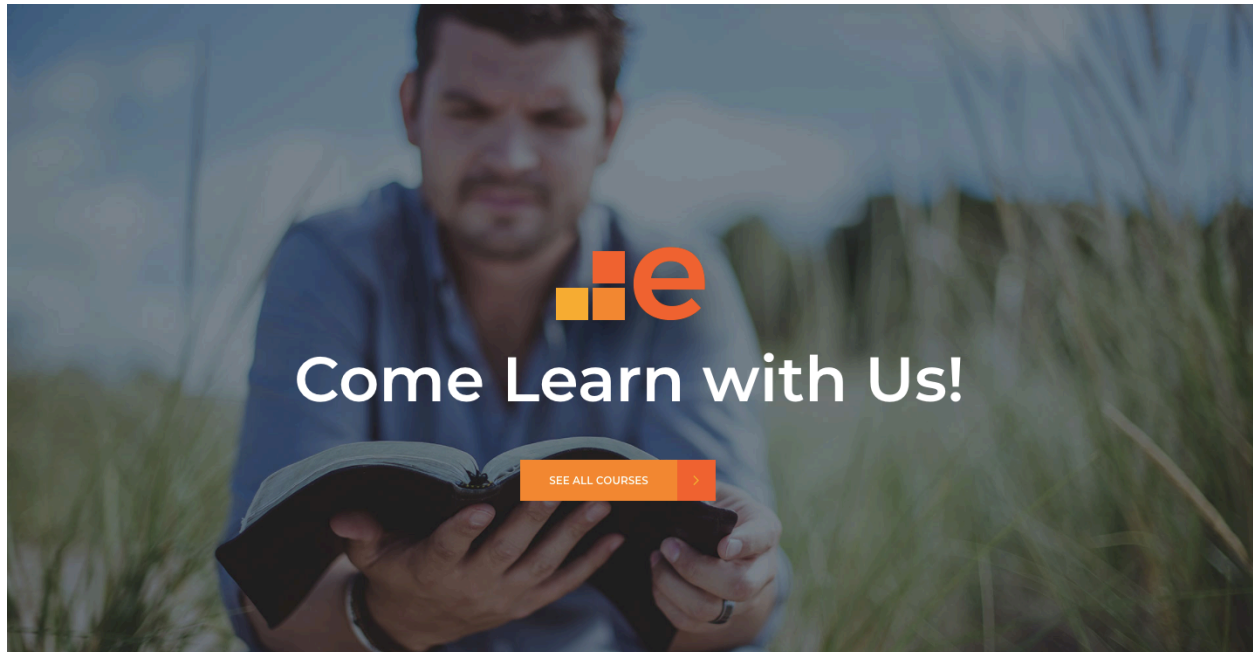


LER O ENUNCIADO ATÉ AO FIM ANTES DE COMEÇAR!

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 num portal que permite a um utilizador registado inscrever-se em cursos online. O site tem as seguintes páginas:

1. “**Home**” é a página de rosto do site;
2. “**Courses**” é a página principal do site;
3. “**Register**” para registo de utilizadores;
4. “**Login**” para o login de utilizadores;
5. “**myCourses**” mostra uma lista dos cursos em que o utilizador se inscreveu;

É dado acesso a um conjunto de templates HTML construídos em *Bootstrap* pela empresa de web design colorlib.com.  
Solicita-se 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. Faça download para a pasta “EXAME”<sup>2</sup>, do código do site web

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

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

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

```
git remote add origin git://github.com/jmatbastos/EXAME.git
```

```
a12345@daw2:~/public_html/EXAME$ git pull origin master
```

C. Complete a instalação

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

D. Actualize o ficheiro .env

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

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

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

E. Crie a sua cópia da base de dados

```
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

### NOTAS:

- Se **não** se recorda da password da sua base de dados, recupere-a com o comando

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

---

<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 “EXAME” já existe na sua área pessoal

- 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
- 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 “EXAME” no servidor web do departamento disponibilizado para o efeito: <http://daw.deei.fct.ualg.pt>
- O controlador principal tem que estar no ficheiro “ElearnController.php”
- A classe com as funções de acesso à base de dados tem que estar no ficheiro “Elearn\_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/elearn>

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

```
* @Route("/elearn name="elearn")
* @Route("/courses", name="courses")
* @Route("/register", name="register")
* @Route("/login", name="login")
* @Route("/logout", name="logout")
* @Route("/enroll/{course_id}", name="enroll")
* @Route("/myCourses", name="myCourses")
```

### Pode ver um demo do site no URL

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

## FUNCIONALIDADE “HOME”

A funcionalidade “Home” é página de rosto do site.

### 1. [3 valores]

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”.



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

#### NOTA:

- seu site deve ser portátil; utilize as funções `asset()` e `path()` para gerar os hyperlinks locais!

## FUNCIONALIDADE “COURSES”

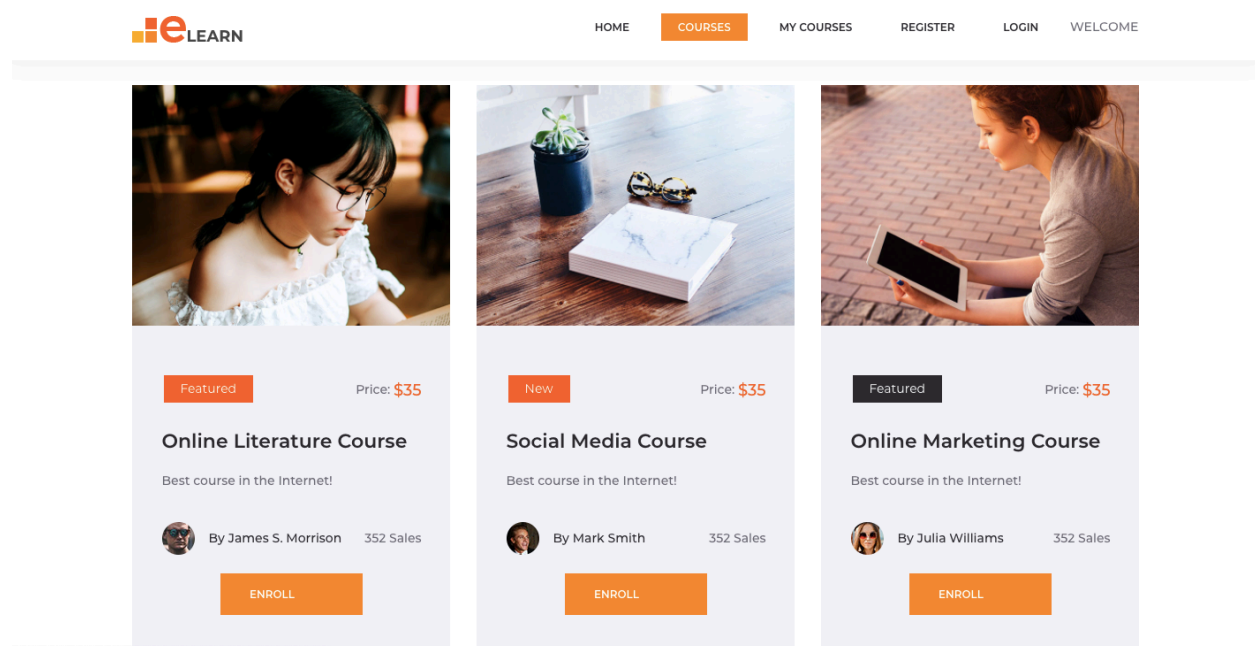
A funcionalidade “Course” é página principal do site e contem uma lista dos cursos disponíveis no site.

### 2. [5 valores]

Construa o template Twig para esta página.

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

Encontra este template na pasta “demo”



Construa o método `courses()` no controlador `ElearnController.php` e a query correspondente no controlador `Elearn_modelController.php` responsável por fazer uma lista dos cursos existentes na tabela “courses” da base de dados

- Preencha a lista com todos os campos (“name”, “description”, “price”, “category”, “teacher”, “sales”, “image”) existentes na tabela “courses”

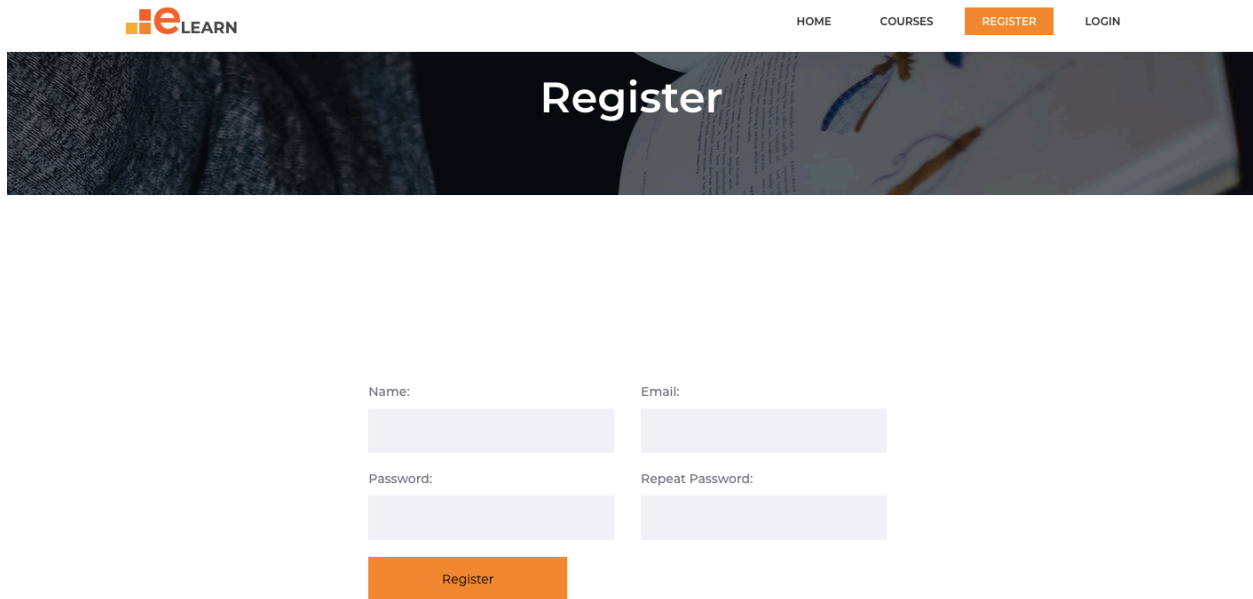
## FUNCIONALIDADE “REGISTER”

A funcionalidade “Register” permite registar um utilizador.

### 3. [2.5 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”.



Construa os métodos `register()` e `register_action()` no controlador `ElearnController.php`<sup>3</sup>

- Se todos os campos obrigatórios não estão preenchidos, redirecciona novamente para o formulário com uma mensagem de erro. A validação dos dados deve ser feita utilizando o componente Symfony "Validator".
- 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 `Elearn_modelController.php`.

A `password_digest` é a hash utilizando o algoritmo MD5 da password:  
`substr(md5($_POST['pass1_utilizador']), 0, 32)`. Pode utilizar a função

---

<sup>3</sup> Pode combinar os dois métodos `register()` e `register_action()` num só método

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

## FUNCIONALIDADE “LOGIN” & “LOGOUT”

A funcionalidade “Login” permite autenticar um utilizador.

### 4. [2.5 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 `login()` e `login_action()` no controlador `ElearnController.php`<sup>4</sup> e a query correspondente no controlador `Elearn_modelController.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 de entrada no site
  - na barra de navegação 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 “menu”
  - na barra de navegação da página “courses” aparece um novo hyperlink “My Courses”
  - na página “course” fica visível para cada curso um hyperlink “ENROLL” com o id do curso embutido (`/enroll/1`, `/enroll/2` etc)
- em caso de insucesso
  - re-direcciona novamente para a página `login`.
  - Envia uma mensagem de erro “Wrong email or password”

No controlador `ElearnController.php` construa o método `logout()` que encerra a sessão

---

<sup>4</sup> Se desejar pode combinar os dois métodos `login()` e `login_action()` num só método



## FUNCIONALIDADE “ENROLL”

A funcionalidade “Enroll” permite ao utilizador inscrever-se num curso.

### 5. [2 valores]

Construa o método `enroll(course_id)` no controlador `ElearnController.php` e a query correspondente no controlador `Elearn_modelController.php`, responsáveis pelo registo do curso seleccionado pelo utilizador na base de dados

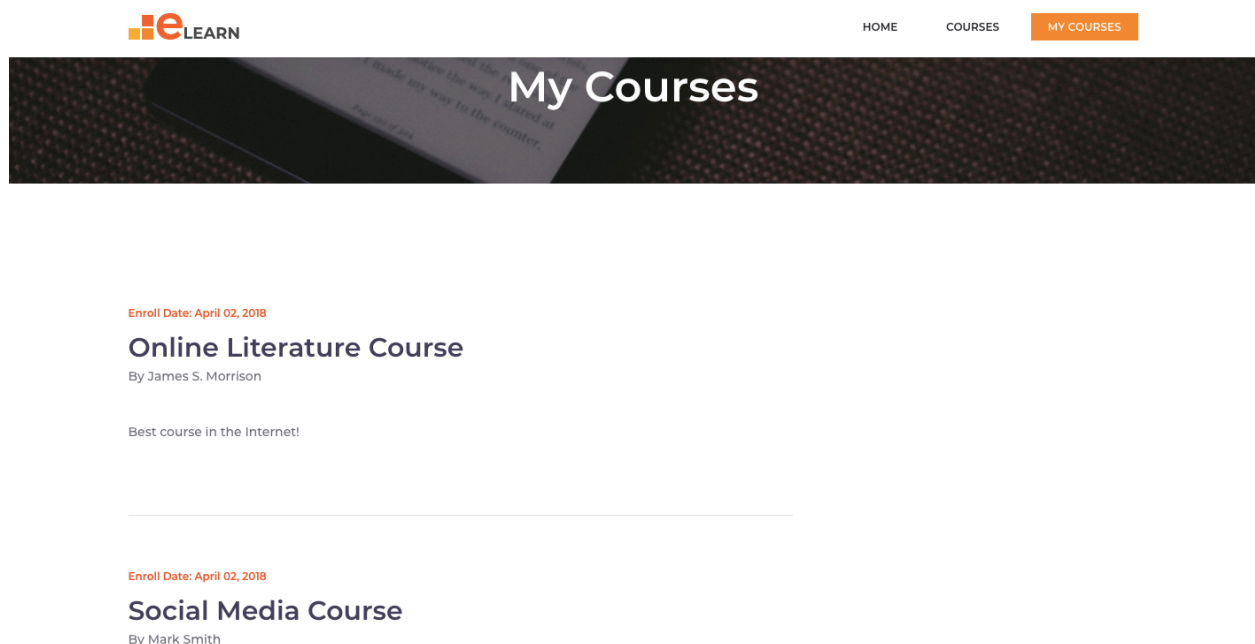
## FUNCIONALIDADE “MY COURSES”

A funcionalidade “My Courses” permite ao utilizador registado mostrar uma lista com os cursos em que se inscreveu.

### 6. [3 valores]

Construa o template Twig correspondente.

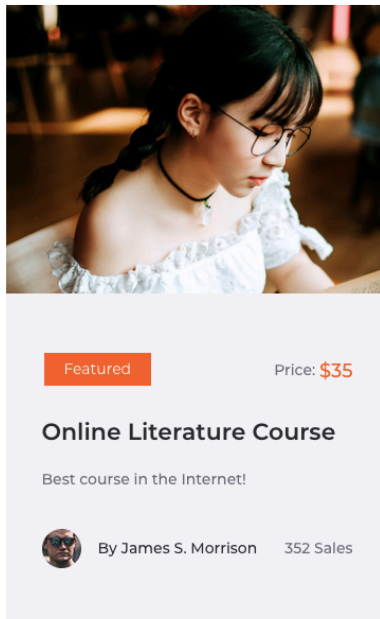
SUGESTÃO: Adapte o template “`myCourses.html`” fornecido pela empresa de web design. Encontra este template na pasta “demo”



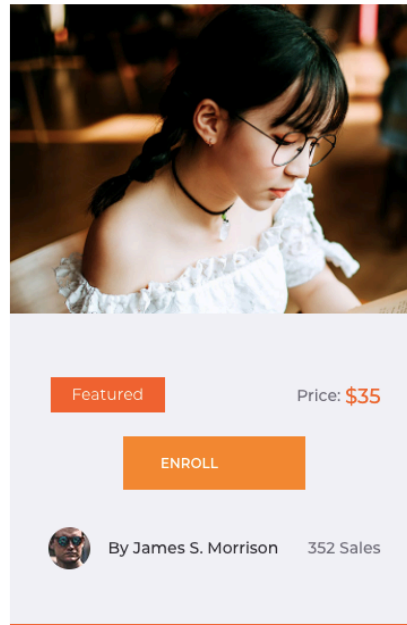
Construa o método `myCourses()` no controlador `ElearnController.php` e a query correspondente no controlador `Elearn_modelController.php`

## FUNCIONALIDADE “ON MOUSE OVER”

Na página "Courses", a funcionalidade “On Mouse Over” permite converter o nome do curso num hyperlink “Enroll” com o id do curso embutido (/enroll/1, /enroll/2 etc), como se mostra na figura



a) o "rato" não está em cima de "Online Literature Course"



b) o "rato" está em cima de "Online Literature Course"

## 7. [2 valores]

Quando o “rato” está em cima da área do curso aparece o botão “ENROLL” substituindo o texto do nome do curso, como se vê na figura. Programe no template "courses.html.twig" em Javascript/JQuery/Vue.js (à sua escolha) esta funcionalidade. Esta funcionalidade é válida para todos os utilizadores (anonimos e "logados")

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

## NOTAS:

- Caso tenha trabalhado no seu 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-- ---).

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

```
a12345@daw2:~$ mkdir nome_da_pasta
```

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

```
a12345@daw2:~$ cp -R nome_da_pasta_original nome_da_pasta_destino
```

### 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@daw2:~$ tail -f /var/log/apache2/error.log
```

## ANEXO 1 Acesso à base de dados MySQL

- O acesso à base de dados MySQL pode ser feito utilizando o software **phpMyAdmin** disponível no URL

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

ou em linha de comando (substitua “12345” pelo seu número de aluno)

```
a12345@daw2:~$ mysql -u a12345 -p db_a12345
```

## 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) NOT 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,
  `roles` longtext COMMENT '(DC2Type:json)',
  PRIMARY KEY (`id`),
  UNIQUE KEY (`email`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Table structure for table `teachers`
--

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

--
-- Dumping data for table `teachers`
--

INSERT INTO `teachers` VALUES (1,'James S.
Morrison','James.Morrison@gmail.com','teacher_1.jpg'),(2,'Patty
Smith','Patty.Smith@gmail.com','teacher_2.jpg'),(3,'Peter
Frampton','Peter.Frampton@gmail.com','teacher_3.jpg'),(4,'Mary
Kinget','Mary.Kinget@berkeley.edu','teacher_4.jpg'),(5,'Julia
Williams','Julia.Williams@mit.edu','teacher_5.jpg'),(6,'Paulo Nabo da
Horta','Paulo.Horta@ualg.pt','teacher_6.jpg');

--
-- Table structure for table `coursecategories`
--

CREATE TABLE IF NOT EXISTS `coursecategories` (
```

```

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

--
-- Dumping data for table `coursecategories`
--

INSERT INTO `coursecategories` VALUES
(1, 'Featured', NULL), (2, 'New', NULL), (3, 'Best
Seller', NULL), (4, 'Legacy', NULL), (5, 'Classic', NULL);

--
-- Table structure for table `courses`
--

CREATE TABLE IF NOT EXISTS `courses` (
  `id` int(11) NOT NULL auto_increment,
  `cat_id` int(11) NOT NULL,
  `teacher_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,
  `sales` int(5) default NULL,
  PRIMARY KEY (`id`),
  CONSTRAINT FOREIGN KEY (`cat_id`) REFERENCES `coursecategories` (`id`),
  CONSTRAINT FOREIGN KEY (`teacher_id`) REFERENCES `teachers` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Dumping data for table `courses`
--

INSERT INTO `courses` VALUES (1,1,2, 'Literature Course', 'The
best!', 40, 'course_1.jpg', 345), (2,2,5, 'Social Media Course', 'The best
Social in the Internet!', 35, 'course_2.jpg', 233), (3,3,4, 'Android
Course', 'Best Mobile app development
online!', 50, 'course_3.jpg', 124), (4,5,5, 'Literature Course', 'The best
present day literature course!', 30, 'course_4.jpg', 456), (5,4,5, 'Marketing
Course', 'The latest economy
theories!', 60, 'course_5.jpg', 896), (6,1,3, 'Javascript Course', 'The most
complete!', 55, 'course_6.jpg', 753), (7,3,5, 'Music Course', 'Let your
inspiration be free!', 20, 'course_7.jpg', 345), (8,1,2, 'Vue.JS
Course', 'Become an expert in frontend
programming!', 45, 'course_7.jpg', 245), (9,2,6, 'Advanced Symphony
Programming', 'Simply the best!', 60, 'course_8.jpg', 644);

--
-- Table structure for table `enrolls`

```

--

```
CREATE TABLE IF NOT EXISTS `enrolls` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `user_id` int(11) NOT NULL,  
  `course_id` int(11) NOT NULL,  
  `enroll_date` datetime NOT NULL,  
  PRIMARY KEY (`id`),  
  CONSTRAINT FOREIGN KEY (`user_id`) REFERENCES `users` (`id`),  
  CONSTRAINT FOREIGN KEY (`course_id`) REFERENCES `courses` (`id`)  
  ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```