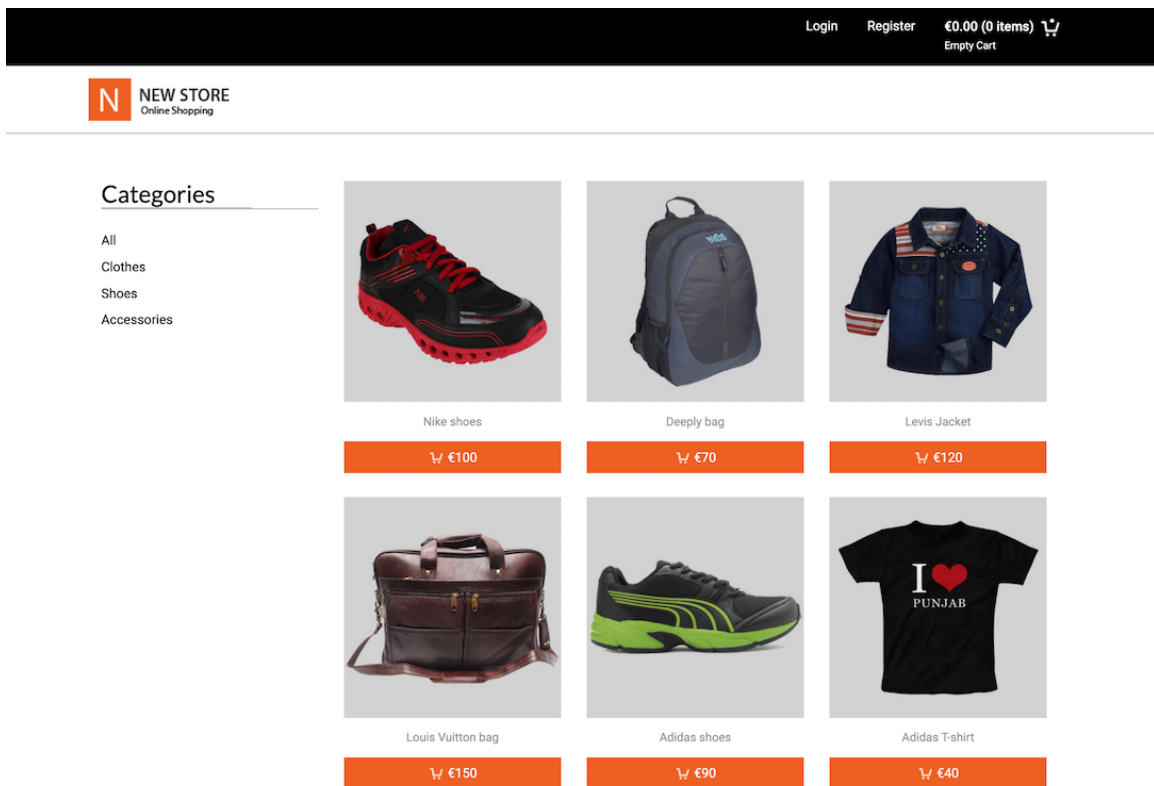


LAB 11 – Construção de uma E-shop em VueJS

O objectivo do trabalho é fazer uma loja electrónica ("E-shop") que vende produtos ou oferece serviços.



REQUISITOS

- A aplicação deve ser realizada **obrigatoriamente e exclusivamente** em programação front-end com o **framework VueJS** ou num framework derivado de VueJS (quasar, bootstrap-vue, etc).
- A aplicação é suportada por uma base de dados; a **estrutura da base de dados é dada** (ver apêndice)
- **O código “backend” da aplicação (APIs) é dado. A “store” de VueJS é dada.**
- É permitido alterar a estrutura da base de dados, mas terá de alterar as APIs!
- Os produtos na base de dados devem ser **diferentes para todos os alunos**, e a quantidade de produtos deve ser maior que seis
- Os produtos/serviços podem pertencer a categorias, mas é OPCIONAL
- Os *templates* podem ser livremente descarregados da net, mas o aluno **não pode utilizar** os mesmos *templates* que outro aluno
- É permitido livremente o uso de código JavaScript (Bootstrap, JQuery...) inclusive de código JavaScript *open source* descarregado da Internet, mas recomenda-se apenas utilizar livrarias compatíveis com o framework utilizado
- A aplicação funciona **obrigatoriamente** com um carrinho de compras:
 1. É obrigatório realizar o carrinho de compras em VueJS.
 2. APENAS utilizadores autenticados podem transformar o carrinho de compras numa encomenda através do mecanismo de “checkout”.
- O site deve implementar regras de usabilidade básicas, por exemplo
 1. utilizadores anónimos não devem ver no menu funcionalidades a que não têm acesso
 2. formulários com erros ou incompletos devem informar o utilizador do problema
- A estrutura do site é descrita na secção 1.

- A **aplicação deve correr obrigatoriamente** na área pessoal do aluno na pasta “**LAB11**” no servidor web do departamento disponibilizado para o efeito:

<http://daw.deei.fct.ualg.pt/>

- A **página de entrada no site tem de ser**

<http://daw.deei.fct.ualg.pt/~a12345/LAB11/dist/>

- **Exemplos** de sites que têm as funcionalidades desejadas encontram-se em

<http://daw.deei.fct.ualg.pt/~a555551/vue-app.eshop/dist/> ¹

em

<http://all.deei.fct.ualg.pt/~a999994/store> ²

e em

<http://daw.deei.fct.ualg.pt/~a555552/eshop/public/products> ³

NOTA: Os sites exemplo nos links referidos contêm funcionalidades (página individual do produto, *invoice*, geração de pdfs, ...) que são OPCIONAIS.

¹ site realizado com o framework VueJS

² site realizado em symfony; apenas referido aqui para demonstração das funcionalidades

³ site realizado parcialmente com o framework VueJS; apenas referido aqui para demonstração das funcionalidades

0. PRELIMINARES

Este guião realiza inteiramente o desenvolvimento do laboratório no seu PC/portátil.

- **VUEJS**

Faça o download de uma instalação base do VueJS adaptada para este lab em⁴

<https://github.com/jmatbastos/LAB11/archive/refs/heads/master.zip>

e expanda o ficheiro `LAB11-master.zip` para dentro pasta

`C:\XAMPP\htdocs`

- Se necessário mude o nome da pasta de `LAB11-master` para `LAB11`

- **VUE STORE**

Na pasta `C:\XAMPP\htdocs\LAB11\src\store` encontra o código de todas as acções necessárias para acesso AJAX à base de dados.

Nos ficheiros `"comments.js"`, `"microposts.js"`, `"users.js"` **actualize os URLs**

`http://daw.deei.fct.ualg.pt/~a12345/LAB11/api/...`

e substitua `"12345"` pelo seu nº de aluno

- **API E CREDENCIAIS DE ACESSO À BASE DE DADOS**

Na pasta `C:\XAMPP\htdocs\LAB11\api` encontra o código de todas as APIs necessárias para acesso à base de dados, bem como encontra no ficheiro `index.html` uma descrição de todas as APIs

Faça login com o PuTTY no servidor daw2 (IP 10.10.23.184)

⁴ em alternativa se tem a aplicação git instalada no seu PC/portátil:

`C:\XAMPP\htdocs> git clone https://github.com/jmatbastos/LAB11.git`

```
a12345@daw2:~$ cd public_html
a12345@daw2:~/public_html$
git clone https://github.com/jmatbastos/LAB11.git LAB11
```

Abra o ficheiro LAB11/api/db.php

```
a12345@daw2:~/public_html$ cd LAB11/api
a12345@daw2:~/public_html/LAB11/api$ nano db.php
```

e actualize com as suas credenciais de acesso à base de dados

```
$hostname = "localhost";
$db_name = "db_a12345";
$db_user = "a12345";
$db_passwd = "PASS";
```

(substitua '12345' pelo seu nº de aluno e 'PASS' pela sua password de acesso à base de dados...)

- **ATUALIZAÇÃO DA BASE DE DADOS**

Atualize a base de dados no servidor de produção com o comando

```
a12345@daw2:~/public_html/LAB11/api$
mysql -ua12345 -pPASS db_a12345 < ~/public_html/LAB11/api/db.SQL
```

Encontra no **APÊNDICE 1** a descrição da estrutura da base de dados, bem como um método alternativo de criação das tabelas.

TESTE A INSTALAÇÃO DO FRAMEWORK

- Execute o comando

```
C:\XAMPP\htdocs\LAB11> npm install5
```

- Execute o comando

```
C:\XAMPP\htdocs\LAB11> npm run serve
```

A partir do seu browser preferido vá ao seguinte URL⁶

`http://localhost:8080/LAB11/dist/`

Deverá receber uma página web de boas-vindas



Welcome to Your LAB11 Vue App

⁵ tem de ter instalado node.js versão 16.18.0 ou 16.18.1

⁶ se tem o serviço web Apache instalado pode também correr o comando

```
C:\XAMPP\htdocs\LAB11> npm run build
```

e ir com o seu browser ao URL **`http://localhost/LAB11/dist/`**

1. ESTRUTURA DO SITE

O site tem as seguintes páginas/vistas (Views):

1. **"Home"** é a página de entrada do site
2. **"Products"** é a página de produtos do site
3. **"Register"** para registo de utilizadores;
4. **"Login"** para o login de utilizadores;
5. **"Basket"** mostra o carrinho de compras;
6. **"Orders"** mostra uma lista de encomendas finalizadas;
7. **"Message"** página utilizada para o envio de mensagens;

O site pode ter mais (ou menos) páginas, desde que a funcionalidade mínima seja realizada!

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

```
const routes = [  
  {  
    path: '/',  
    component: Home  
  },  
  {  
    path: '/products',  
    component: Products  
  },  
  {  
    path: '/basket',  
    component: Basket  
  },  
  {  
    path: '/register',  
    component: Register  
  },  
  {  
    path: '/login',
```

```
    component: Login
  },
  {
    path: '/orders',
    component: Orders
  },
  {
    path: '/message',
    component: Message
  },
]
```

No ficheiro "zip" que é distribuído encontra um esboço das vistas requeridas.

Considere o laboratório concluído quando obtiver as funcionalidades realizadas em **VueJS** que se mostram no exemplo em

<http://daw.deei.fct.ualg.pt/~a555551/vue-app.eshop/dist/>

2. UPLOAD

Atualize o “publicPath” no ficheiro

C:\XAMPP\htdocs\LAB11\vue.config.js

com o conteúdo

```
// vue.config.js
module.exports = {
  publicPath: '/~a12345/LAB11/dist/'
}
```

(substitua '12345' pelo seu nº de aluno!)

Execute o comando

C:\XAMPP\htdocs\LAB11> npm run build

Faça o upload com scp/WinSCP/FileZilla das pastas

- "public"
- "src"
- "dist"

para a pasta “LAB11” no seu site web pessoal no servidor de produção

Teste o funcionamento do site no URL

<http://daw.deei.fct.ualg.pt/~a12345/LAB11/dist>

(substitua '12345' pelo seu nº de aluno...)

**NÃO espere pelo último minuto para fazer o upload!!
Faça o upload e teste o seu site no servidor de
produção a intervalos regulares!!**

NÃO faça upload da pasta "node_modules"!!

REFERÊNCIAS

- <https://vuejs.org/>
- <http://intranet.deei.fct.ualg.pt/IPM/labVueJS>
- <http://daw.deei.fct.ualg.pt/~a555551/LAB11/api/>
- <http://intranet.deei.fct.ualg.pt/IPM/docs/VueJS.pdf>
- <http://intranet.deei.fct.ualg.pt/IPM/docs/Vuex.pdf>
- <http://intranet.deei.fct.ualg.pt/IPM/docs/VueRouter.pdf>

ANEXO 1. Estrutura da base de dados

A estrutura da base de dados pode ser criada com

- na shell do servidor de produção (IP 10.10.23.184) com o comando

```
a12345s@daw2:~$  
mysql -ua12345 -pPASS db_a12345 < ~/public_html/LAB11/api/db.SQL
```

- com o "tab" "IMPORT" em

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

onde db.SQL é o ficheiro que contem a estrutura da base de dados:

```
CREATE TABLE `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;
```

```
CREATE TABLE `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;
```

```
CREATE TABLE `status` (  
  `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;
```

```

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

```

```

CREATE TABLE `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;

```

```

CREATE TABLE `orders` (
  `id` int(11) NOT NULL auto_increment,
  `user_id` int(11) NOT NULL,
  `created_at` datetime NOT NULL,
  `status` tinyint(1) default NULL,
  `total` int(5) default NULL,
  PRIMARY KEY (`id`),
  CONSTRAINT FOREIGN KEY (`user_id`) REFERENCES `users`
(`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

CREATE TABLE `order_items` (
  `id` int(11) NOT NULL auto_increment,
  `order_id` int(11) NOT NULL,
  `product_id` int(11) NOT NULL,
  `quantity` int(5) default NULL,
  PRIMARY KEY (`id`),
  CONSTRAINT FOREIGN KEY (`order_id`) REFERENCES `orders`
(`id`),
  CONSTRAINT FOREIGN KEY (`product_id`) REFERENCES
`products` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8

```

ANEXO 2. Descrição dos métodos disponíveis na API de acesso à base de dados

Todos os dados recebidos da API ou enviados à API estão no formato JSON. Pode testar os métodos disponíveis com o comando "curl", por exemplo

```
curl -X GET 'http://daw.deei.fct.ualg.pt/~a12345/LAB11/api/users.php?email=jbastos@ualg.pt'  
curl -X POST 'http://daw.deei.fct.ualg.pt/~a12345/LAB11/api/users.php' -d '{"name":"Mary Stevens", "email":"stevens@gmail.com", "password":"sWd356"}'
```

ou utilizando o site <https://postman.com>

Tabela users

- **Verifica se um utilizador já se encontra registado na base de dados:**
GET `http://daw.deei.fct.ualg.pt/~a12345/LAB11/api/users.php?email=jbastos@ualg.pt`
retorna: JSON string `{"id":"1","name":"Jose Bastos","email":"jbastos@ualg.pt"}` ou null
- **Registo de um utilizador:**
POST `http://daw.deei.fct.ualg.pt/~a12345/LAB11/api/users.php`
Body: JSON string `{"name":"Jose Bastos","email":"jbastos@ualg.pt","password":"segredo"}`
retorna: JSON string `{"id":"1","name":"Jose Bastos","email":"jbastos@ualg.pt"}` ou null
- **Login de um utilizador:**
GET `http://daw.deei.fct.ualg.pt/~a12345/LAB11/api/users.php?email=jbastos@ualg.pt&password="segredo"`
retorna: JSON string `{"id":"1","name":"Jose Bastos","email":"jbastos@ualg.pt","session_id":"9pgcsrpgjrj3lm7sr6glju679v"}` ou null

- **Logout de um utilizador:**

GET http://daw.deei.fct.ualg.pt/~a12345/LAB11/api/users.php?session_id=9pgcsrpgjrj3lm7sr6glju679v
 retorna: JSON string {"message":"Session destroyed"} ou null

Tabela products

- **Todos os produtos:**

GET <http://daw.deei.fct.ualg.pt/~a12345/LAB11/api/products.php>
 retorna: JSON string [{"cat_id":"SUSHI","id":"79","name":"Salmon Roll","description":"Simply delicious!", ...}, ...]
 ou null

Tabela categories

- **Todas as categorias:**

GET <http://daw.deei.fct.ualg.pt/~a12345/LAB11/api/categories.php>
 retorna: retorna: JSON string [{"id":"","name":"SUSHI","description":""," ...}, ...]

Tabela orders

- **Todos as ordens do utilizador:**

GET http://daw.deei.fct.ualg.pt/~a12345/LAB11/api/orders.php?session_id=9pgcsrpgjrj3lm7sr6glju679v
 retorna: JSON string [{"id":"45", "user_id":"19", "created_at":"2021-12-03 18:20:31", "total":"102", "status_id":"1", "items":[{"id":"1", "name":"Salmon Roll", "price":"18", "quantity":1}, {"id":"2", "name":"Spicy Fish Set", "price":"28", "quantity":2}], ...] ou null

- **Registo de uma nova ordem do utilizador:**

POST http://daw.deei.fct.ualg.pt/~a12345/LAB11/api/orders.php?session_id=9pgcsrpgjrj3lm7sr6glju679v
 Body: JSON string
 {"user_id":"19", "totalAmount":"102", "status_id":"1", "items":[{"id":"1", "quantity":1}, {"id":"2", "quantity":2}]}
 retorna: JSON string {"id":"45", "user_id":"19", "totalAmount":"102", "status_id":"1"}