

Нов Български Университет

Бакалавърски факултет

Департамент "Информатика"

Курсов проект

Курс:

CITB579 Практика по програмиране и интернет технологии

Тема:

Проект E-Mall.

Ръководство за администратора и програмиста.

Изготвил:

Тихомир Младенов

Специалност:

Бизнес Информатика

Курс: Четвърти

Фак. Н: F-61480

Проверили:

гл. ас. д-р Филип
Андонов

ас. д-р Делян
Керемедчиев

НБУ София, август 2019 г.

Съдържание

1. Въведение	стр. 3
2. Описание на проекта	стр. 4
2.1. Публична част на апликацията	стр. 4
2.2.Административна част на апликацията	стр. 5
3. Структура на апликацията	стр. 6
4. Използвана литература	стр. 9

РЪКОВОДСТВО ЗА АДМИНИСТРАТОРА

1. Въведение

Целта на проекта е да се създаде уеб базирана информационна система, която да агрегира обяви за стоки, които се продават в онлайн магазини. За целта трябва да се използват магазини, които предоставят достъп до своите бази данни, за да може информацията за стоките да се събира.

В планиращата фаза на този проект открих, че в следствие на наскоро въведените строги мерки за контрол на достъпа до лични данни, като GDPR в ЕС, много сайтове, включително и магазини, са затегнали достъпа до техните ресурси (т.н. API) за предоставяне на информация, може би защото представляват евентуален вектор на атака, при който от не добре защитено API може да се достигне до лична информация на потребителите. Това от своя страна ще доведе до огромни глоби за собственика на сайта, защото не е защитил личните данни на потребителите си.

Например в началото на юли месец 2019 година, Brittish Airways бяха глобени 183 000 000 паунда заради теч на потребителска информация.

Поради това след много търсене открих единствено Etsy, електронен магазин за арт стоки, все още да предоставя достатъчен обществен достъп до своите ресурси чрез своето RESTful API, което да ми позволи да изградя логически смислен проект. Amazon нямат конкретен API за теглене на информация от техния Marketplace уебсайт, а за достъп до Ebay се чака 2 седмици за преглед на кандидатурата за достъп до RESTful API ресурсите им за предоставяне на информация за активните в продажба предмети. Срок, който не можех да чакам.

Затова разработих апликацията само за логиката на Etsy API, но лесно може да се добави логика за събиране на информация и от друго API, като на Ebay например.

2. Описание на проекта

Изградената уеб система се дели на две основни части. Публична, за достъп от потребителите, и администраторска, за достъп от администраторите на сайта. Тъй като в изискванията не бе описано, не е изградена никаква форма за оторизиране на достъпа до администраторската част.

2.1. Публична част на апликацията

Намира се на www.example.com/ или www.example.com/index.html.

Потребителят може да инициира търсене с или без критерии в полето "Search here". Може и да поясни откъде иска да търси съответствия: Name – по името на предмет; Description – по описанието на предмет.

Ако бъдат намерени резултати те ще бъдат заредени, ако няма резултати, ще бъде изписано съобщение за липса на такива.

Заредените резултати първоначално ще бъдат 50 за търсене с или без параметър, след това ще се добавят по 30 автоматично като се стигне края на страницата. Трябва да се отбележи, че при търсене с термин / дума, при опит за зареждане на още 30 предмети, не винаги ще се заредят още 30, защото може и да няма други резултати със зададените критерии за търсене, а и търсачката не е особено сложна, за да търси сложни фрази. Най-добре се справя с една дума или изрази с по две думи.

След като се заредят в прозореца на клиента, резултатите допълнително могат да се филтрират по категория (зареждат се програмно с търсенето), дали са промоционални (Etsy API няма такава опция, затова нито един не е промоционален) и по ценови диапазон.

За всеки предмет се зарежда име, снимка, цена, валута на цената, описание на предмета, линк към сайта на продавача.

2.2. Административна част на апликацията

Намира се на www.example.com/admin.html .

Състои се от две части, “Edit categories” и “Browse the visitor’s data”.

Първата категория дава възможност на администратора да прегледа всички предмети, имената им и техните категории в базата данни. Списъкът се зарежда автоматично при отваряне на страницата. При избор на даден предмет, чиято категория може да се промени, избореният ред се визуализира и потребителя може да въведе новата категория за него и получава потвърждение дали е успешна промяната или не. При успешно обновяване на категория се презарежда страницата автоматично.

Втората категория дава възможност на администратора да наблюдава данните от посещенията по продукти и категории, магазини, ценови диапазон, и уникални потребители. Може да използва която и да е от критериите за търсене по негово желание.

- Products, categories and visits for each category:

Зарежда името на продукта, категорията му, и общия брой посещения за дадената категория.

Има филтър за махане на продукти без посещения.

- Shop name and total shop visits:

Зарежда магазините и броя посещения към всеки магазин.

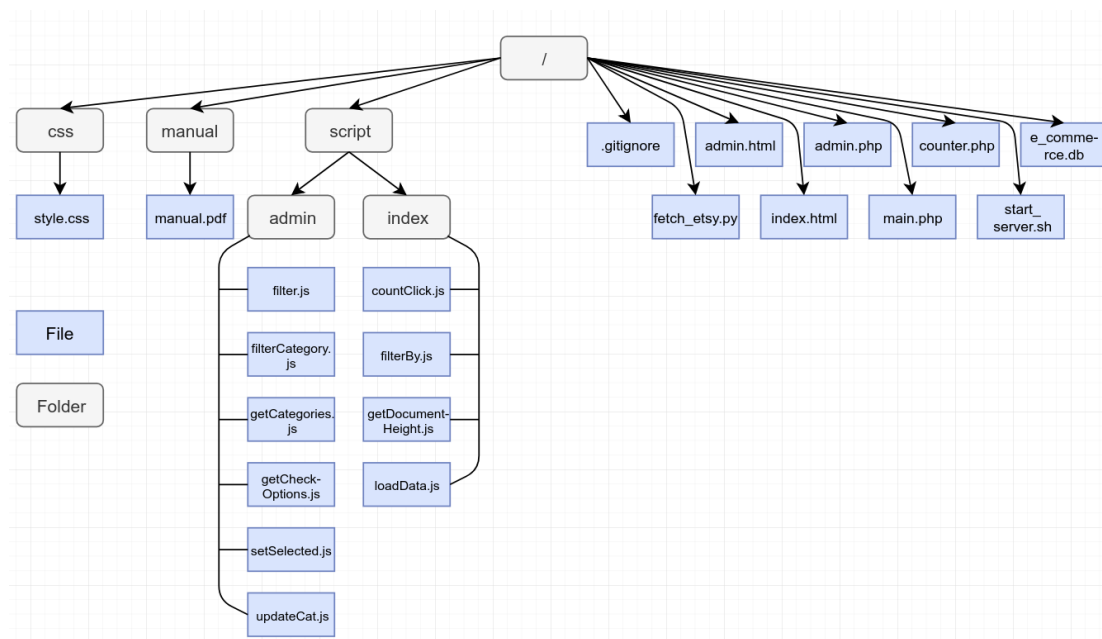
- Price range and product visits: (не уникални)

Зарежда името, цената, валута, ценови диапазон на продукта, както и общия брой посещения за него. Визуализира сумарния брой посещения за всеки ценови диапазон. Редовете без посещения могат да бъдат филтрирани.

- Visits by unique users:

Зарежда броя уникални посещения за всеки продукт.

3. Структура на апликацията



Файл / Папка	Описание
/css/style.css	Основни CSS стилове за някои от елементите в index.html.
/manual/manual.pdf	Ръководство
/script/admin/	Папка с JavaScript функции за /admin.html страницата на сайта. files.html съдържа документираните основни JS функции.
-filter.js	Съдържа функцията filter() , която филтрира и крие продуктите, които нямат посещения.
-filterCategory.js	Съдържа функцията filterCategory() , която филтрира и крие продуктите и категории, които нямат посещения.
-getCategories.js	Съдържа функцията getCategories() , която изпраща заявка към back-end-a да й се предадат всички продукти с техните категории. Получава отговор в JSON формат, информацията от който зарежда в

	admin.html
-getCheckOptions.js	Съдържа функцията <code>getCheckOptions()</code> , която създава и изпраща заявка според зададените ѝ параметри като информация за посещения на продукти, категории, уникални посещения и др. Получената информация се предава на <code>parseResponse()</code> изтегляне на информацията, която вика <code>setPriceRangeVisitsTotal()</code> за пресмятането на общ брой посещения за ценова група, след като данните се заредят в страницата.
-setSelected.js	Съдържа функцията <code>setSelected()</code> , която изобразява избраната в момента категория за промяна.
-updateCat.js	Съдържа <code>updateCat()</code> , която изпраща заявка към Back-End-а за промяна на категорията на даден продукт.
/script/index/	Папка с JavaScript функции за /index.html страницата на сайта. <code>files.html</code> съдържа документираните основните JS функции.
-countClick.js	Съдържа <code>countClick()</code> ф-я, която изпраща събитие към back-end-а при натискане на връзката към магазина за даден продукт, за да се следи броя на посещенията.
-filterBy.js	Съдържа ф-я <code>filterBy()</code> , която филтрира заредените резултати с предмети по следните критерии: категория; е в промоция; ценови диапазон.
-loadData.js	Съдържа ф-я <code>sendData()</code> , която изпраща заявка към back-end-а за зареждане на информация по зададен критерии за търсене. Използва ф-я <code>parseJson()</code> за

	преобразуване на върнатия JSON форматиран низ в JSON обект. Извиква ф-я <code>sortObjects()</code> като ѝ подава JSON обекта за да го сортира във възходящ ред по ID на продукта. Извиква <code>setCategories()</code> , за програмно зареждане зареждане на категориите на резултатите от търсенето.
<code>.gitignore</code>	Файл за блокиране на някои файлове да бъдат качвани във version control.
<code>admin.html</code>	<code>.html</code> страницата за администриране на апликацията.
<code>admin.php</code>	<code>.php</code> скрипт, който обработва заявките към DB от <code>admin.html</code> страницата.
<code>counter.php</code>	<code>.php</code> скрипт, който обработва заявките към DB от <code>index.html</code> относно водене статистика за посещения на предметите.
<code>e_commerce.db</code>	Базата данни, в която се зареждат продукти от електронните магазини, и от която се четат.
<code>fetch_etsy.py</code>	Python3 скрипт, който тегли информация от RESTful API на Etsy , преструктурира и зарежда информацията в базата данни.
<code>index.html</code>	Потребителската страница на апликацията.
<code>main.php</code>	<code>.php</code> скрипт, който е главен за обработването на заявки към базата данни от <code>index.html</code> страницата.
<code>start_server.sh</code>	Скрипт за стартиране на локален php тестови сървър.

4. Използвана литература

https://www.etsy.com/developers/documentation/getting_started/resources	Документация на Etsy API.
https://www.w3schools.com/js/ https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference	JS examples and documentation
https://www.w3schools.com/jquery/ https://api.jquery.com/	JQUERY examples and documentation
https://www.w3schools.com/php/ https://www.php.net/docs.php	PHP examples and documentation
https://getbootstrap.com/docs/4.0/getting-started/introduction/ https://www.w3schools.com/js/js_ajax_intro.asp https://www.sqlite.org/docs.html	Bootstrap documentation and examples. AJAX examples and documentation. SQLite3 documentation and examples.
https://make.wordpress.org/core/handbook/best-practices/inline-documentation-standards/javascript/ https://riptutorial.com/php/example/6140/adding-metadata-to-functions	JS code documentation standard. PHP code documentation standard.