КУРСОВ ПРОЕКТ

ПО

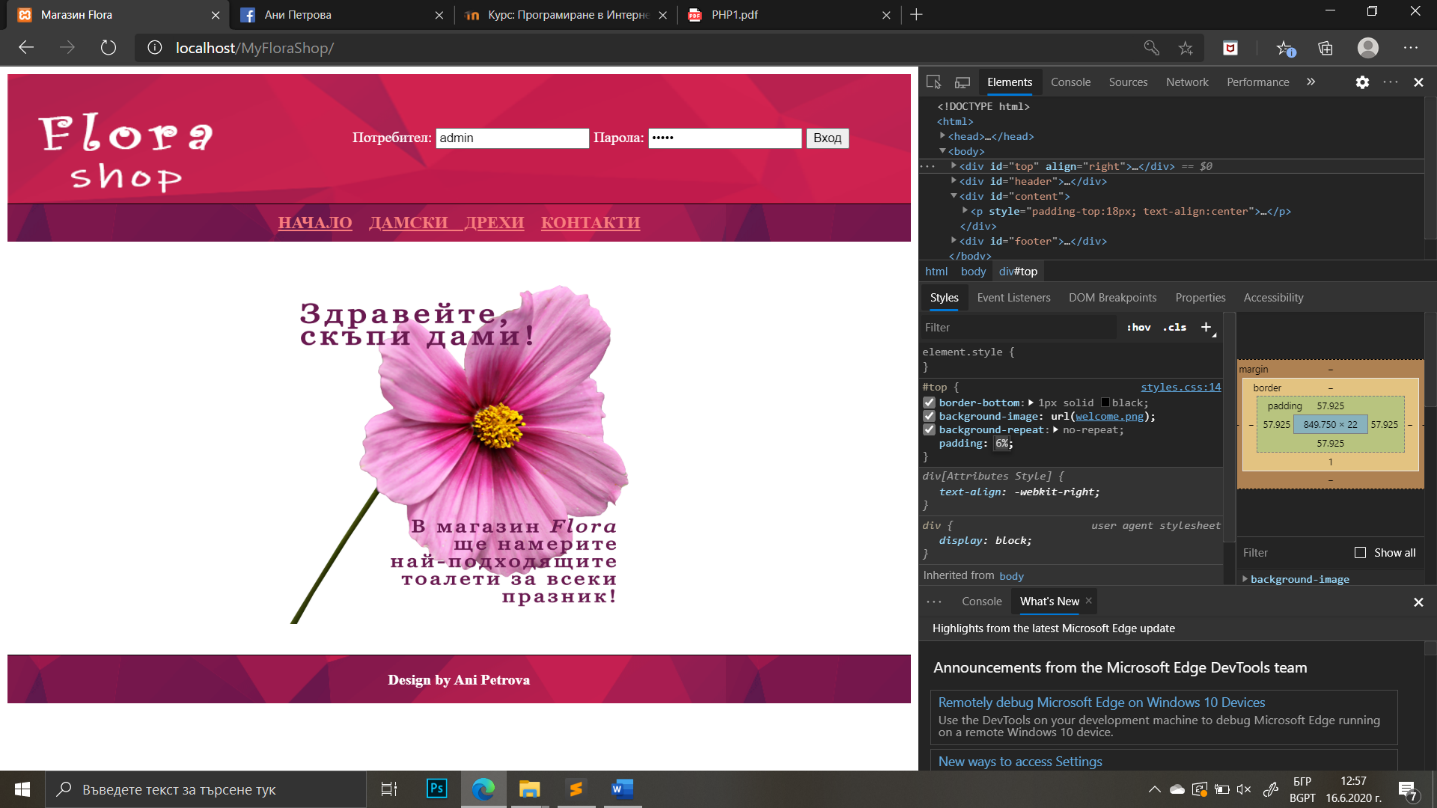
ПРОГРАМИРАНЕ В ИНТЕРНЕТ С PHP И MYSQL

на тема онлайн магазин „Flora“

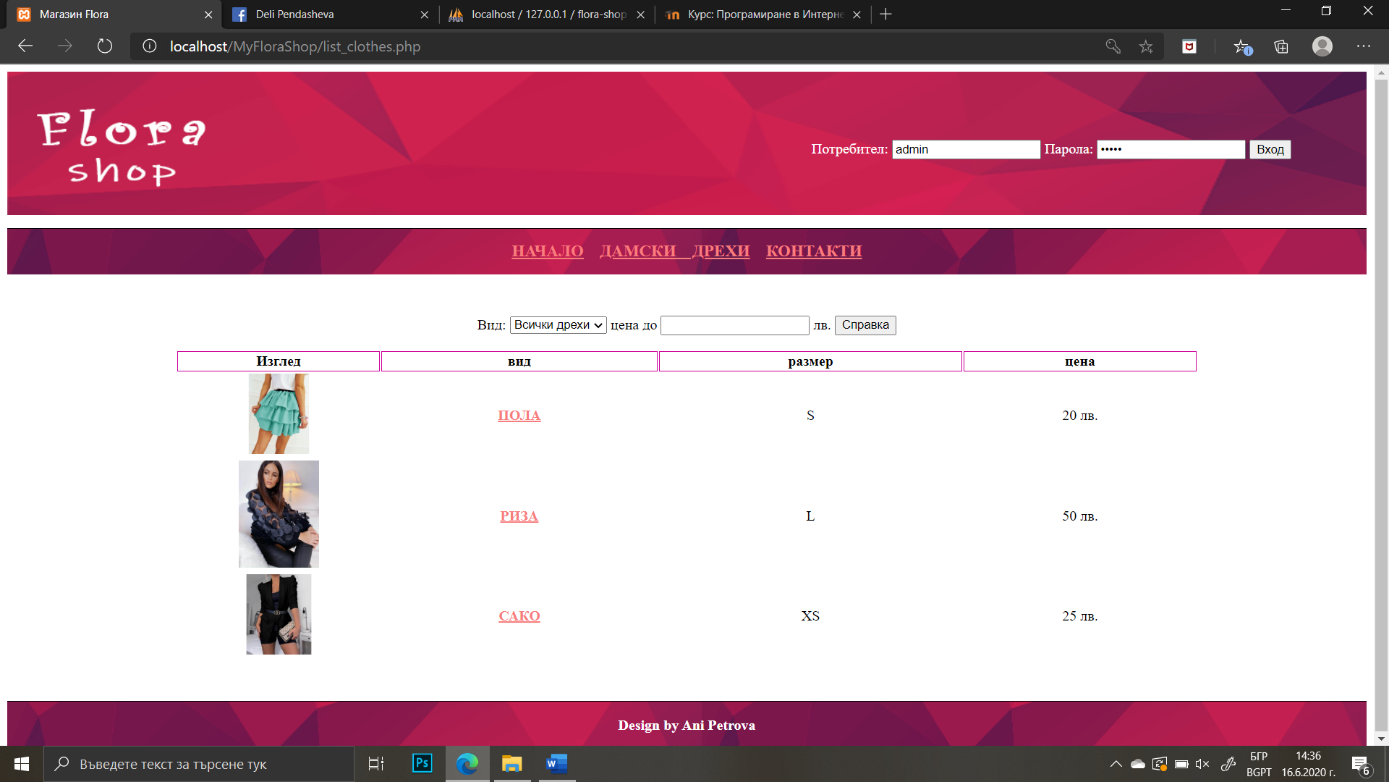
на Анелия Петрова   
СТД 2 курс   
фак. № 1801681036

Дата: 17.06.2020г

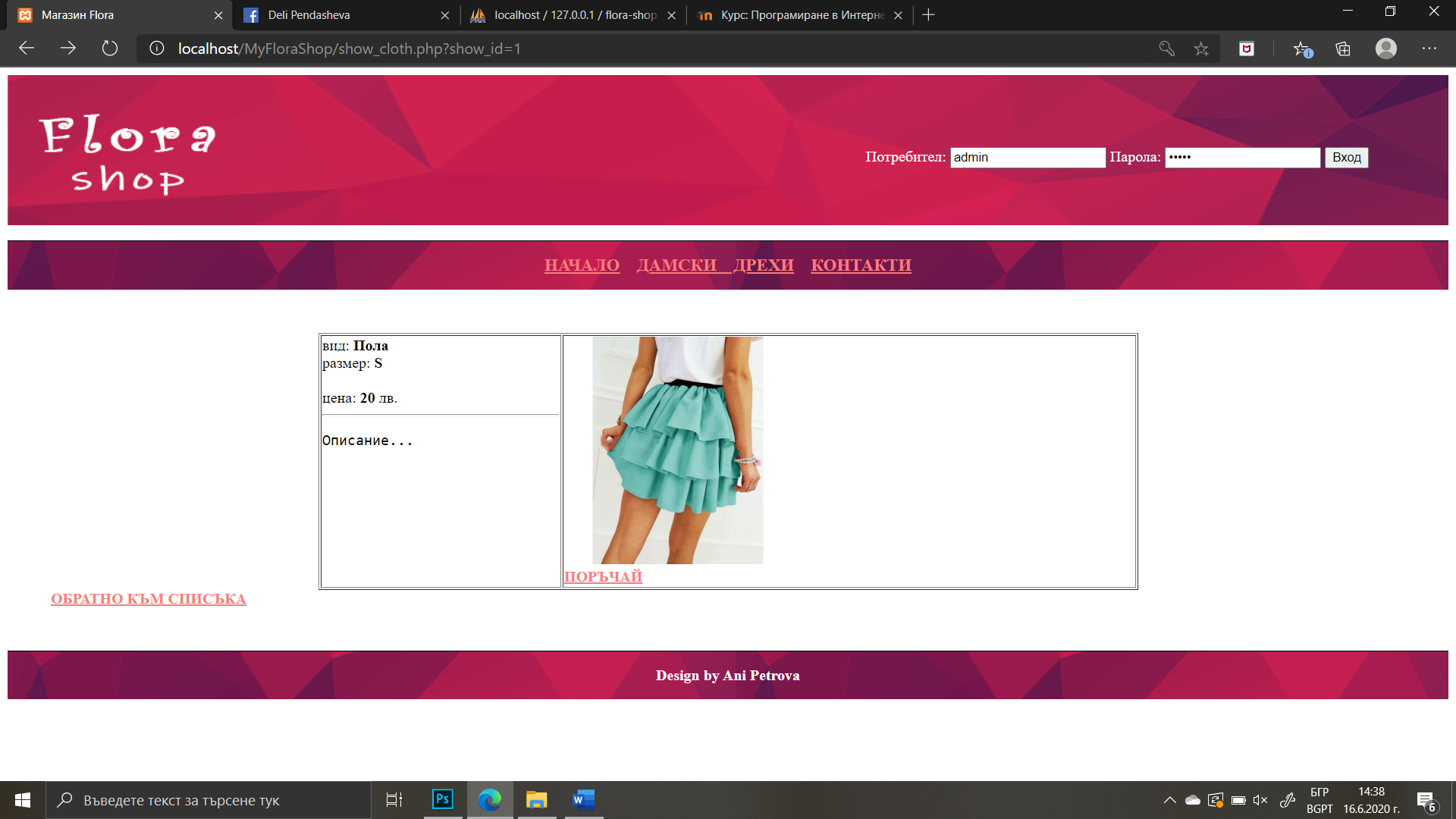
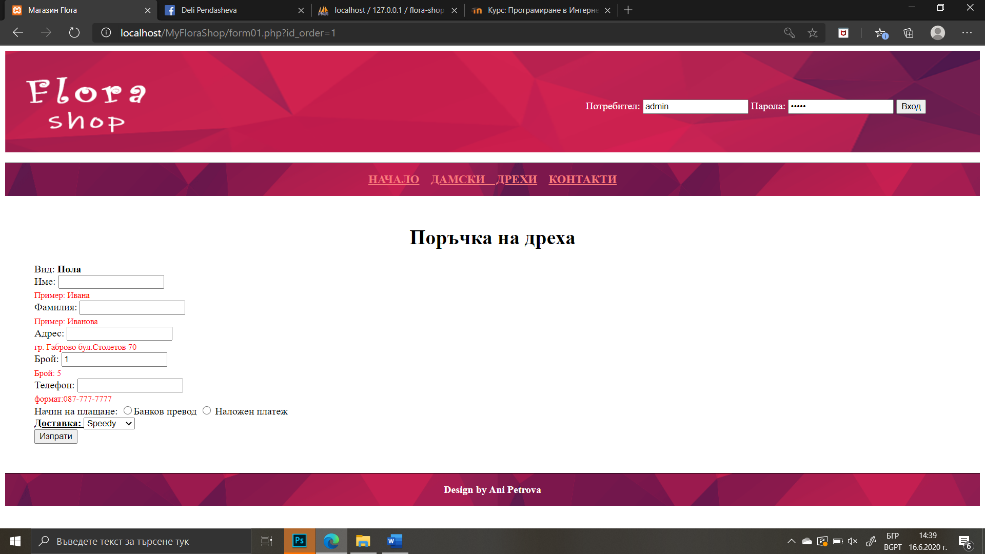
**1. Кратко описание на предназначението и възможностите на сайта**

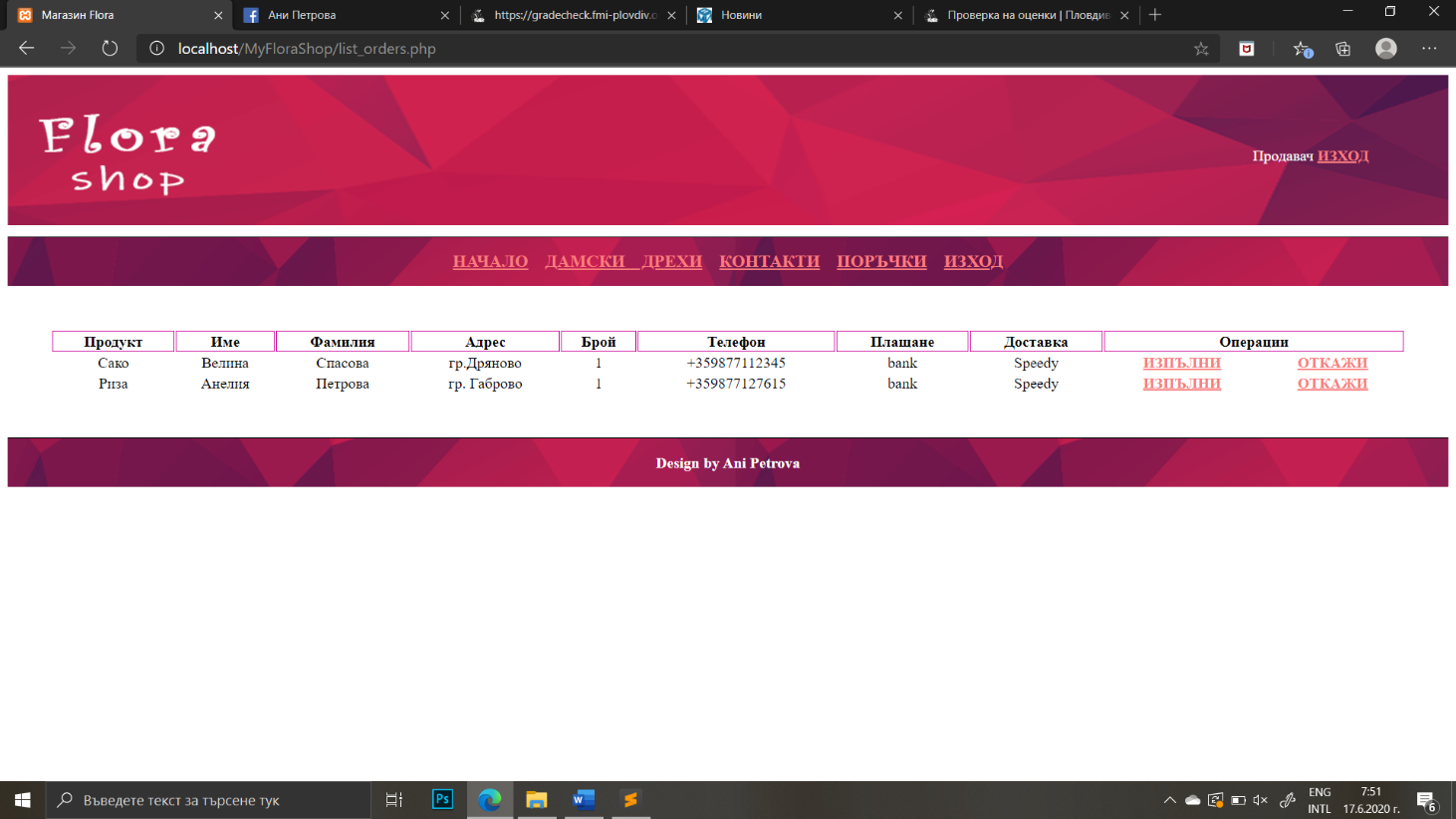
Проектът ми включва онлайн магазин за дамски дрехи. Първата ми страница е „Начало“, на нея има снимка с поздрав към всички потребители(дамите).

Всички потребители имат право да разглеждат стоката, която се намира в страница „Дамски дрехи“.

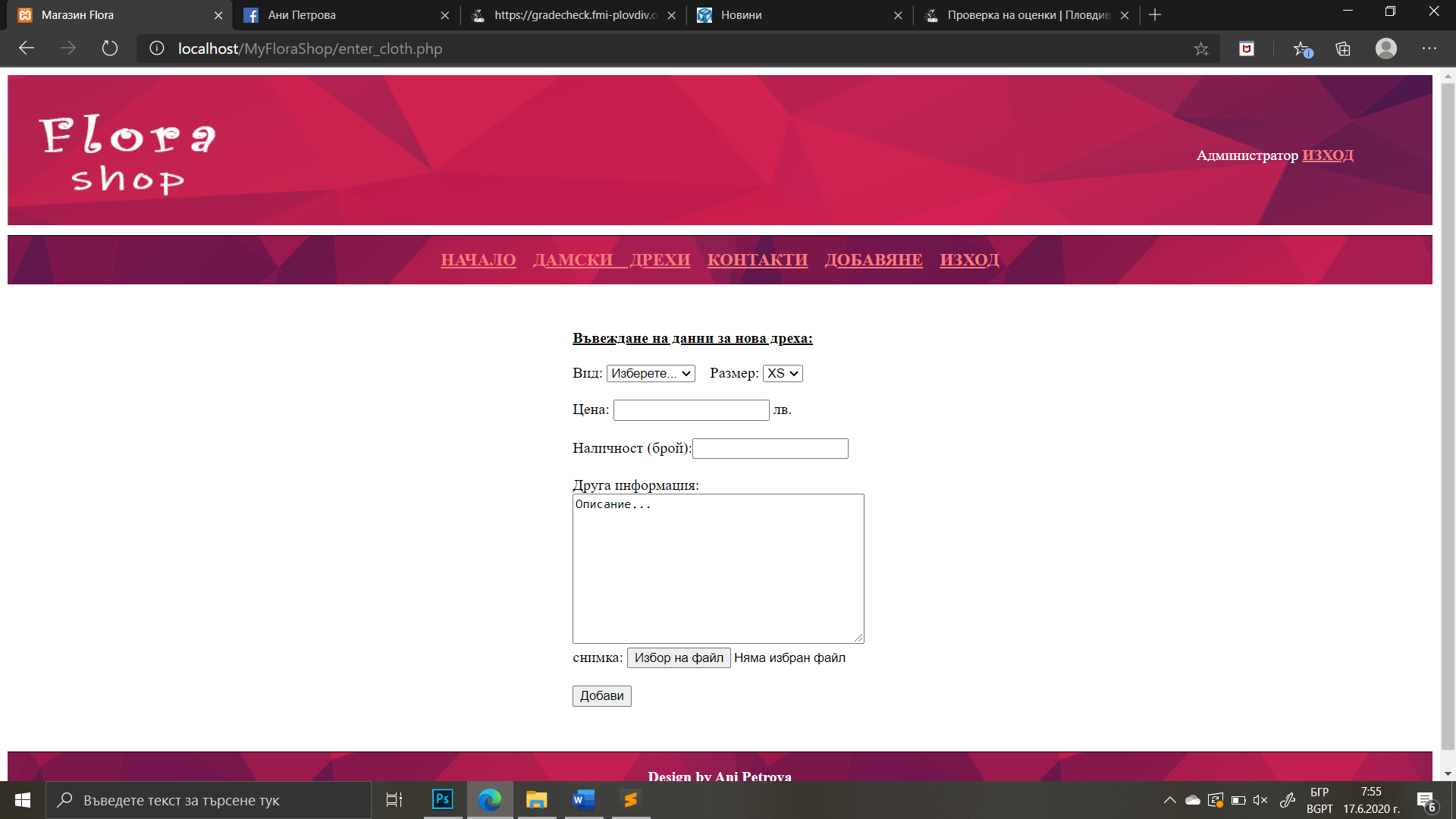


Право на вход в системата имат само администратор и продавач. Потребителят може да поръча дреха като попълни правилно формуляр.

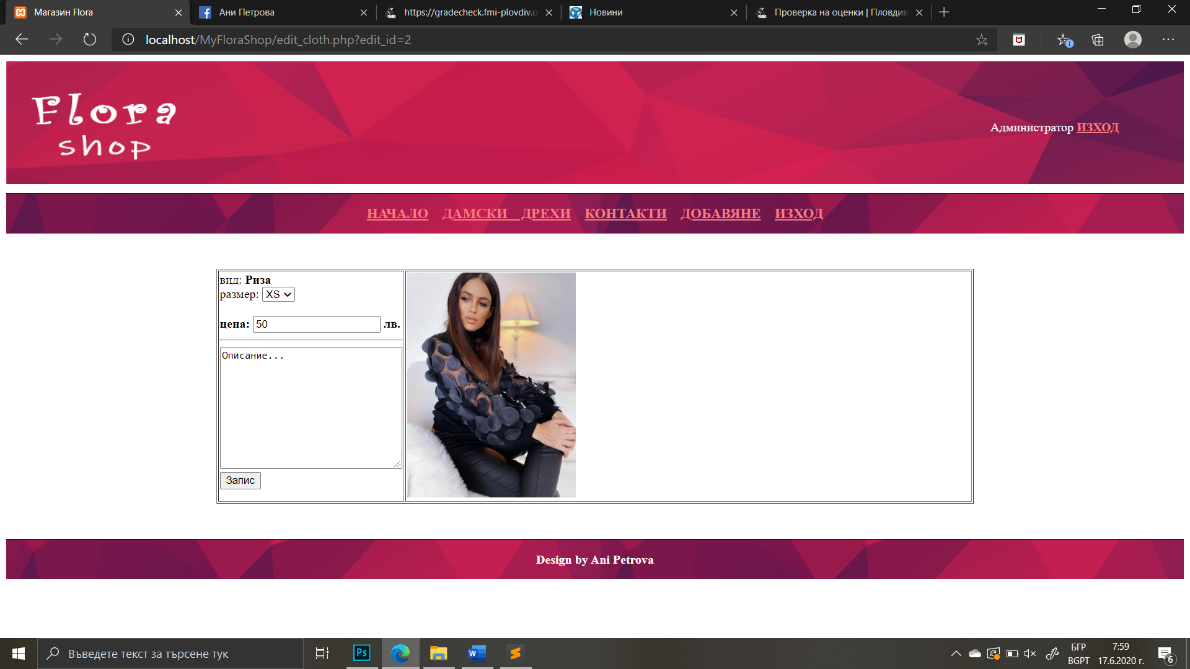


Той се изпраща до „Продавача“ в таблица „Поръчки“.

Администраторът след "Вход" в системата има право да извършва някой от следните операции: добавяне на нова дреха, редактиране и изтриване. Добавяне на нова дреха се извършва в страница „Добавяне на дреха“.



В нея трябва да се изберат вид и размер, да се въведат цена, брой на наличността, описание и да се добави снимка на артикула. Ако някой от тези критерии (с изключение на добавяне на снимка) не бъде изпълнен правилно, дрехата няма да бъде добавена.

В операция „редактиране“ могат да се извършат промени само върху размера, цената и описанието на дрехата.

Операция „изтриване“ изтрива записа от листа с Дамски дрехи и от базата данни. Продавачът има право да извършва операция „продажби“ и да изпълнява и отказва поръчки. Операциите „продажби“ и „изпълнение на поръчка“ имат една и съща функционалност- намаляване на наличността на бройките. Могат да се въведат по-малко бройки отколкото са били до момента. Ако се въведе по-голямо число отколкото е било предното, ще даде „Некоректна стойност“.

**2. Структура на БД− описание на таблиците и връзките (ако има) между тях.**

Проектиране на БД- **flora-shop**

Базата данни съдържа четири таблици: tbl\_clothes, tbl\_users, tbl\_makes и tbl\_orders

* Структура на **tbl\_clothes**
  + Таблицата съдържа седем на брой полета – id\_cloth, id\_make, размер(size), цена(price), брой наличност(number), описание(moreinfo) и снимкa(picture) и са изградени по същият начин както в tbl\_clothes
  + CREATE TABLE `tbl\_clothes` (

`id\_cloth` int(10) NOT NULL,

`id\_make` int(11) NOT NULL,

`size` varchar(4) NOT NULL,

`price` int(5) UNSIGNED NOT NULL,

`number` int(3) UNSIGNED NOT NULL,

`moreinfo` text NOT NULL,

`picture` varchar(15) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Dumping data for table `tbl\_clothes`

* + INSERT INTO `tbl\_clothes` (`id\_cloth`, `id\_make`, `size`, `price`, `number`, `moreinfo`, `picture`) VALUES

(1, 2, 'S', 20, 4, 'Описание...', 'Pic4.jpg'),

(2, 3, 'L', 50, 19, 'Описание...', 'Pic2.jpg'),

(8, 1, 'XS', 25, 12, 'Описание...', 'Pic8.jpg');

* + Полето id\_cloth e първичен ключ. Това гарантира уникалността на стойностите в полето.
  + Полето id\_make e външен ключ, тъй като е първичен ключ в tbl\_makes. Той може да приеме множество повтарящи се стойности. . Когато създаваме външен ключ, изолзвайки каскадно актуализиране, референтните редове се актуализират в дъщерната таблица, когато реферираният ред се актуализира в родителската таблица, която има първичен ключ.
  + ALTER TABLE `tbl\_clothes`

ADD PRIMARY KEY (`id\_cloth`),

ADD KEY `id\_make` (`id\_make`);

* + ALTER TABLE `tbl\_clothes`

ADD CONSTRAINT `tbl\_cloth\_ibfk\_1` FOREIGN KEY (`id\_make`) REFERENCES `tbl\_makes` (`id\_make`) ON UPDATE CASCADE;

COMMIT;

* + Автоматично създаване последователно id при създаване на нов запис

ALTER TABLE `tbl\_clothes`

MODIFY `id\_cloth` int(10) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=9;

* Структура на **tbl\_orders**
  + Таблицата съдържа девет на брой полета – id\_order, id\_make, първо име(firstName), фамилия(lastName), адрес(address), брой(count), телефон(phone), плащане(payment) и доставка(delivery)
  + CREATE TABLE `tbl\_orders` (

`id\_order` int(10) NOT NULL,

`id\_make` int(10) NOT NULL,

`firstName` varchar(20) NOT NULL,

`lastName` varchar(20) NOT NULL,

`address` varchar(15) NOT NULL,

`count` int(10) NOT NULL,

`phone` int(10) NOT NULL,

`payment` varchar(30) NOT NULL,

`delivery` varchar(10) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

* + INSERT INTO `tbl\_orders` (`id\_order`, `id\_make`, `firstName`, `lastName`, `address`, `count`, `phone`, `payment`, `delivery`) VALUES

(6, 1, 'Велина', 'Спасова', 'гр.Дряново', 1, 877112345, 'bank', 'Speedy'),

(7, 3, 'Анелия', 'Петрова', 'гр. Габрово', 1, 877127615, 'bank', 'Speedy'),

(11, 1, 'Ивана', 'Иванова', 'гр.Хасково', 1, 881231234, 'bank', 'Econt');

-- Indexes for table `tbl\_orders`

--

ALTER TABLE `tbl\_orders`

ADD PRIMARY KEY (`id\_order`),

ADD KEY `id\_make` (`id\_make`);

-- AUTO\_INCREMENT for table `tbl\_orders`

--

ALTER TABLE `tbl\_orders`

MODIFY `id\_order` int(10) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=13;

* Структура на **tbl\_makes**
  + Таблицата съдържа две на брой полета – id\_make и make
  + CREATE TABLE `tbl\_makes` (

`id\_make` int(10) NOT NULL,

`make` varchar(15) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

* + INSERT INTO `tbl\_makes` (`id\_make`, `make`) VALUES

(2, 'Пола'),

(3, 'Риза'),

(1, 'Сако');

* + Полето id\_make e първичен ключ. Това гарантира уникалността на стойностите в полето.
  + Полето make e уникален ключ. Той може да приема само една нулева стойност и не може да има дублирани стойности
  + ALTER TABLE `tbl\_makes`

ADD PRIMARY KEY (`id\_make`),

ADD UNIQUE KEY `make` (`make`);

* + ALTER TABLE `tbl\_makes`

MODIFY `id\_make` int(10) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=4;

* Структура на **tbl\_users**
  + Таблицата съдържа пет на брой полета – id\_user, username, passwd, usertype и personname  
    Има два вида – Администратор и Продавач
  + CREATE TABLE `tbl\_users` (

`id\_user` int(10) NOT NULL,

`username` varchar(15) NOT NULL,

`passwd` varchar(15) NOT NULL,

`usertype` tinyint(4) NOT NULL,

`personname` varchar(20) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

* + INSERT INTO `tbl\_users` (`id\_user`, `username`, `passwd`, `usertype`, `personname`) VALUES

(1, 'admin', 'admin', 1, 'Администратор'),

(2, 'seller', 'seller', 2, 'Продавач');

* + ALTER TABLE `tbl\_users`

ADD PRIMARY KEY (`id\_user`),

ADD UNIQUE KEY `username` (`username`);

**3. Структура на сайта− файлове/страници и връзки между тях**

Файл **login.php – Форма за вход**Тук потребителят въвежда своето име и парола. Когато потребителя изпрати данните, те се проверяват в базата данни дали съвпадат. Ако потребителското име и паролата съвпадат, потребителя има достъп до сайта. В противен случай искането за вход се отхвърля.

<?php include "inc-files/before\_content.code"; ?> //връзка с html файловете и стиловете

<div id="content">

<?php

$mysql=new mysqli("localhost", "root", "", "flora-shop");  
 $mysql->set\_charset("utf8"); //свързване с базата данни  
 $result = $mysql->query(

"select \* from tbl\_users where username='". addslashes($\_POST["username"]) . "' and passwd='" . addslashes($\_POST["passwd"]) . "'"); //селектират се данни от таблицата $mysql->close();  
 if ($row = $result->fetch\_assoc()) // проверява се дали съвпадат с попълнениете във формата  
 {  
 $\_SESSION["username"]=$row["username"];

$\_SESSION["usertype"]=$row["usertype"];

$\_SESSION["personname"]=htmlspecialchars($row["personname"]);

header("Location: ."); //ако съвпадат се изписва в header ролята на потребителя

exit;

}

else echo "<span class='errMsg'>Невалидно потребителско име или грешна парола!</span>"; //ако не съвпадат искането за вход се отхвърля

?>

</div>

<?php include "inc-files/after\_content.code"; ?> //връзка с html файловете и стиловете

Файлът **enter\_cloth.php** визуализира формуляр за въвеждане на данни за нова дреха. При натискане на бутона Добави, въведените данни се изпращат към файла **add\_cloth.php**. Тъй като това има право да го прави само администраторът, преди всичко се проверява дали, той е влязъл в системата.

<?php

if ((!isset($\_SESSION["username"])) || ($\_SESSION["usertype"]!=1))

{

echo "<span class='errMsg'>Нямате права...!</span>";

}

if ($\_POST["id\_make"]==0) $errMsg .="Не е избран вид!<br>"; //проверка дали е избран вид

if (empty($\_POST["price"])) //проверка дали е въведена цена

$errMsg .="Не е въведена цена!<br>";

else

if (!is\_numeric($\_POST["price"])) $errMsg .="Некоректно въведена цена!<br>";//проверка дали не е въведен текст, вместо число

$str\_query="insert into tbl\_clothes(id\_make, size, price, number, moreinfo, picture) values ('".$\_POST["id\_make"]."','".$\_POST["size"]."','".$\_POST["price"]."','".$\_POST["number"]."','".addslashes($\_POST["moreinfo"])."', '')"; // данните се добавят чрез метод POST в базата данни и таблица tbl\_clothes

Файлът **list\_cloth.php** прави връзка с MySQL, осъществява изпълнение на SQL−заявка за прочитане на всички данни, съдържащи се в таблицата **tbl\_clothes** от БД, и запомня получения резултат в променливата $result. Посредством метод fetch\_assoc, включен в цикъл while, данните от $result се вземат ред по ред и се визуализират в табличен вид в страницата.

echo "<table align='center' width='80%'>";

if (isset($\_SESSION["username"]) && $\_SESSION["usertype"]==1) //различните потребители имат различна визуализация, поради различните права за работа със системата

{

echo "<tr><th>Вид</th><th>Размер</th><th>Цена</th><th>Наличност (брой)</th><th colspan='2'>операции</th></tr>";

while($row = $result->fetch\_assoc())

{

echo "<tr align='center'>";

echo "<td>

<a href='show\_cloth.php?show\_id=".$row['id\_cloth']."' title='Подробна информация'>" . htmlspecialchars($row['make']) . " </a></td>

<td>" .$row['size']. "</td>

<td>" .$row['price'] . " лв.</td>

<td>".$row["number"]."</td>

<td><a href='edit\_cloth.php?edit\_id=".$row['id\_cloth']."' title='Промяна на цената и на допълнителната информация'>редактиране</a></td>

<td><a href='javascript:removeCloth(".$row['id\_cloth'].")' title='Изтриване на данните'>изтриване</a></td>";

echo "</tr>";

}

Във файла **list\_clothes.php** е добавена възможност за справка по вид и цена. За целта в скрипта е добавена SQL-заявка към БД за прочитане без повторение на наличните видове. Прочетените от БД видове се визуализират в падащ списък (html-елемент select) и поле в което може да се зададе желаната цена. При натискане на submit-бутона избраният от списъка вид и цена се изпраща по метод get към същия файл, т.е. файлът **list\_clothes.php** се извиква (презарежда) с параметър избраният вид и цена. Прочетените данни се визуализират в табличен вид.

$idMake=0; $prc=""; $insertText=""; $addWord=" where "; $addWordEnd="make";

if (isset($\_GET["id\_make"]))

{

$idMake=$\_GET['id\_make']; $prc=$\_GET['price'];

if ($\_GET['id\_make']!=0)

{$insertText .=$addWord."tbl\_clothes.id\_make=".$\_GET["id\_make"]; $addWord=" and "; $addWordEnd="price";}

if (!empty($\_GET['price']))

if (is\_numeric($\_GET['price']))

{$insertText .=$addWord."price<=".$\_GET['price']; $addWordEnd="price";}

else $prc=$\_GET['price']="";

}

$mysql=new mysqli("localhost", "root", "", "flora-shop");

$mysql->set\_charset("utf8");

$result = $mysql->query("select \* from tbl\_makes order by make");

echo "<form action='".$\_SERVER['PHP\_SELF']."' method='get'>";

echo "<p align='center'>";

echo "Вид: <select name='id\_make'>";

echo "<option value='0'>Всички дрехи</option>";

while($row = $result->fetch\_assoc())

{

echo "<option value='" . $row['id\_make'] . "'".(($row['id\_make']==$idMake)?' selected':'').">" . htmlspecialchars($row['make']) . "</option>";

}

echo "</select>";

echo " цена до <input type='number' name='price' value='".$prc."'> лв.";

echo " <input type='submit' value='Справка'>";

echo "</p>";

echo "</form>";

В показания списък (таблица) с данните за дрехите към всеки ред са добавени опции с линкове „редактиране“ „продажби“ и „изтриване“, извикващи съответно файловете **edit\_cloth.php, sales.php** и **delete\_cloth.php** с параметър id-то на дрехата за извършване на избраната операция. Файлът **edit\_cloth.php** визуализира формуляр, елементие на който получават като стойности по подразбиране старите данни на избраната дреха. След натискане на бутона запис, редактираните данни се изпращат по метод post към файла **update\_cloth.php**, който посредством съответната SQL-заявка актуализира данните в БД.

$errMsg="";  
if ($errMsg)

{  
 echo "<span class='errMsg'>".$errMsg."</span><br>";

echo "<a href='edit\_cloth.php?edit\_id=".$\_POST["id\_cloth"]."'> Корекция на данните</a>";

}

else

{

$mysql=new mysqli("localhost", "root", "", "flora-shop");

$mysql->set\_charset("utf8");

$str\_query="update tbl\_clothes set size=' ".addslashes($\_POST["size"])." ', price=".$\_POST["price"].", moreinfo='".addslashes($\_POST["moreinfo"])."' where id\_cloth=".$\_POST["id\_cloth"];

if ($mysql->query($str\_query))

echo "Данните са обновени...<br>";

$mysql->close();

}

Същото се получава и при файл **sales.php.** Визуализира се поле, в което може да се въведат само останалите налични бройки на дрехата. След натискане на бутона Запис, редактираните данни се изпращат по метод post към файла **update\_number.php**, който посредством съответната SQL-заявка актуализира данните в БД.

$number=$\_POST["number"];

$updatenumber\_id=$\_POST["updatenumber\_id"];

if (empty($number) )

{

echo "<span class='errMsg'>Непълни данни!</span><br>";

echo "<a href='javascript:history.back()' title='Връщане към предходната страница'>Обратно към списъка</a>";

}

else

{

$mysql=new mysqli("localhost", "root", "", "flora-shop");

$mysql->set\_charset("utf8");

$result = $mysql->query("select \* from tbl\_clothes where id\_cloth=".$\_REQUEST['updatenumber\_id']);

$row = $result->fetch\_assoc();

if ($row=$result->fetch\_assoc())

{

if ($result) $oldnumber=$row["number"];

else die("Грешка в БД!");

}

$oldnumber=$row["number"];

if ($oldnumber > $number)

{

echo "<span class='errMsg'>Некоректни данни!</span><br>";

echo "<a href='javascript:history.back()' title='Връщане към предходната страница'>Обратно към списъка</a>";

}

else

{

$mysql=new mysqli("localhost", "root", "", "flora-shop");

$mysql->set\_charset("utf8");

$result= $mysql->query("update tbl\_clothes set number=$number where id\_cloth=".$\_REQUEST['updatenumber\_id']);

if ($result)

echo "Данните за дрехата са обновени...<br>";

$mysql->close();

}

}

Файлът **delete\_cloth.php** изтрива записа(дрехата) от базата данните посредством съответната SQL-заявка.

$result = $mysql->query("select \* from tbl\_clothes where id\_cloth=".$\_REQUEST['del\_id']);

$row = $result->fetch\_assoc();

if ($row['picture']) unlink("pictures/".$row['picture']);

if ($mysql->query("delete from tbl\_clothes where id\_cloth=".$\_REQUEST['del\_id']))

if ($result) echo "Данните за дрехата са изтрити.";

<script type="text/javascript">

function removeCloth(num)

{

if (confirm("Сигурни ли сте, че искате да изтриете данните за дрехата?"))

self.location.href="delete\_cloth.php?del\_id="+num;

}

</script>