

КАТЕДРА: КОМПЮТЪРНИ СИСТЕМИ И ТЕХНОЛОГИИ  
ДИСЦИПЛИНА: ИНФОРМАЦИОННИ СИСТЕМИ

**ЛАБОРАТОРНО УПРАЖНЕНИЕ № 11**

***ТЕМА: Разработка на модул за обработка на потребителски заявки***

**ЦЕЛ:**

Целта на упражнението е студентите да получат практически знания и умения за разработване на модул за обработка на потребителски заявки чрез HTML и PHP скрипт.

**!ВАЖНО:** За да можете да тествате упражнението ви е необходим следния софтуер. **Web server, Php интерпретатор, MYSQL база данни.** Всеки един от тези софтуери ги има в пакетите **WAMP** или **XAMP**. Настоящото упражнение е тествано върху **WAMP Version 3.2.0**.

**I. ТЕОРЕТИЧНА ЧАСТ**

В предходните упражнения вече се запознахме с модул за вход и модул за въвеждане на информация. Друг модул от който най-често се състои една информационна система е модул чрез който потребителите ще могат да подават своите потребителски заявки към базата данни.

Дизайна на модулите за обработка на потребителска информация би могъл да е много разнообразен. Той зависи от: поставената задача, от програмиста, от клиента поставил задачата на програмиста и от доста други фактори. В настоящото упражнение е разработена чрез елементарна структура за дизайн, така че да не затруднява разбирането на кода.

За създаването на тези модули ще се използват знанията и опита на предходните упражнения.

**II. ПРАКТИЧЕСКА ЧАСТ**

За целите на упражнението ще използваме създадената база данни от предходното упражнение „Магазин“. Към нея добавяме 4ри нови таблици: Клиенти, Продукти, Поръчки, Поръчки\_Продукти. Таблиците са подобни на тези от Лабораторно упражнение №6. За улеснение може направо да копираме таблиците в нашата база данни.

**ЗАДАЧА:** *Да се разработи модул който позволява на потребите да извършва следните заявки към базата данни:*

- *Да филтрира поръчките на точно определен Клиент от базата данни;*
- *Да филтрира всички поръчки от дата, до дата;*

### Код за решение на задача:

```
//-----file forma.php-----
<html>
<head>
<meta http-equiv="Content-type" content="text/html; charset=UTF-8">
<title>Форма за справки</title>
</head>
<body>
    <?php
        require_once("database.php");
        $queryCust="SELECT DISTINCT Surname FROM customers";
        $dbCustRecords=mysqli_query($con,$queryCust);
        if (!$dbCustRecords) die('Неуспешно запитване. Получи следната
грешка'.mysqli_error($con));
    ?>
    <div style="width:1000px">
        <div style="text-align:center">
            <h1>Изберете желаната справка</h1><br>
        </div>
        <form method="post" action="script.php">
            <div style="height:130px;width:400px;float:left;">
            </div>
            <div style="height:130px; width:600px; float:left;">
                <label for="Name"> Клиент:</label>
                <select id="Name">
                    <option value="noquery">Name</option>
                    <?php
                        While ($CustRecords=mysqli_fetch_array($dbCustRecords))
                        {
                            ?>
                            <option value="<?php echo $CustRecords['Surname'];
?>"><?php echo $CustRecords['Surname'];?></option>
                            <?php
                                }
                            ?>
                        </select>
                        <br><br>
                        <label for="date1"> От дата:</label>
                        <input type="date" name="date" id="date1">
                        <br><br>
                        <label for="date2"> До дата:</label>
                        <input type="date" name="date" id="date2">
                        <br>
                    </div>
                    <div style="text-align:center">
                        <span> <input type="submit" name="submit" value="GO">
                        </span>
                    </div>
                </form>
```

```
</div>
</body> </html>
```

```
//-----file script.php-----<?php
    require_once("database.php");
if (isset($_POST['submit']))
{   If($_POST['name']!="Name")   $strSurname=$_POST['name'];           if
(isset($_POST['date1']))         $ot_data=$_POST['date1'];           if
(isset($_POST['date2']))         $do_data=$_POST['date2'];
    }
    $queryCust="SELECT * FROM customers WHERE Surname='$strSurname' ";
    $dbCustRecords=mysqli_query($con,$queryCust);
    if (!$dbCustRecords) die('Неуспешно запитване. Получи следната
грешка' .mysqli_error($con));
    While ($CustRecords=mysqli_fetch_array($dbCustRecords))
    {
.....
.....
.....
?>
```

### Пояснение за задача 1

Кода във файл-а „forma.php“ реализира форма показана на фиг.1. Този модул предоставя възможност на потребителя да избере клиента за който да реализира справка както и за поръчките направени от дата до дата за избрания клиент. В случая избора на клиента се осъществява чрез падащ списък който е реализиран чрез тагове <select> и <options>. Той е пълен със клиенти които вече са съхранени в базата данни. Затова в html таговете е вграден на няколко пъти php скрипт чрез който се прочитат имената на клиентите от базата. Така <input> със атрибут type=”date” създава поле което е специализирано за посочване на дата. Като се кликне на това поле се появява календар който подпомага потребителя с избирането ѝ. На фигурата се вижда как се осъществява избора на датата. При натискане на бутон “GO” се задейства скрипта във файла “script.php”.

Чрез „script.php“ се изпраща информацията към базата данни като са направени проверки дали информацията в полетата е попълнена. Изпращането на информацията се извърша чрез функцията mysqli\_query в която се подава необходимата заявка. В тази част няма нови стъпки и затова не е даден целия код. Той наподобява кода от практическата част на лабораторно упражнение №6, с разликата че в заявката се подава информация и за датата.

# Изберете желната справка

Клиент:

От дата:

До дата:

април 2020 г.

пон	вт	ср	четв	пет	съб	нед
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3

Фиг.1. Модул за обработка на потребителска заявка

## III. Задача за самостоятелна работа.

1. Опитайте се да довършите кода във файл script.php така че да се покажат всички покупки на потребителя за избраната дата.
2. Създайте модул за обработка на информация чрез който потребителя да може да генерира всички стоки за които наличността им е под определен брой. Потребителя да посочва броя.
3. Създайте модул за обработка на информация чрез който потребителя да може да генерира всички стоки за които цената им е под или над някаква стойност. Потребителя да избира цената.
4. \*(задача с повишена трудност)Разработете модул който да позволява на потребителя да избира кои стоки от даден вид да се показват. Също така нека да може да ги сортира по негова преценка по низходящ или възходящ ред.

Указание: За улеснение може да използвате бутон който да сортира елементите по избраната преценка на потребителя.