МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ

ХАРЬКОВСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ РАДИОЭЛЕКТРОНИКИ

Кафедра ЭВМ

**ОТЧЕТ**

по лабораторной работе №3 на тему:

**«АСИНХРОННЫЙ ОБМЕН ДАННЫМИ С СЕРВЕРОМ НА ОСНОВЕ ТЕХНОЛОГИИ AJAX»**

по курсу «Internet-технологии»

Выполнил: Проверила:

ст. гр. КИУКИу–18–2 Бологова Н.Н.

Кравчик К. В.

Харьков 2020

**3.1 Цель работы**

Изучить программные механизмы технологии AJAX для динамической отправки или загрузки данных с сервера.

**3.2 Задание на лабораторную работу**

Вариант 5

1. Используется исходный код с лабораторной работы, посвященной работе с данными с помощью интерфейса PDO или MySQLi. Проверить его работу.
2. Изменить код для обработки данных результатов запросов (для параметризированных запросов, реализуемых в прошлой лабораторной по вариантам) таким образом, чтобы вывод результата пользователю не приводил к перезагрузке страницы - т.е. использовать технологию Ajax. Использовать следующие форматы ответа от сервера:

* в формате простого текста (непосредственно получать сгенерированный фрагмент HTML-кода и выводить его в заданном элементе стартовой страницы);
* в формате XML (результат запроса на выборку помещается в генерируемом сервером XML-документе, который клиент считывает через свойство responseXml и формирует вывод пользователю);
* в формате JSON (результат запроса на выборку помещается в массив, который затем преобразуется в JSON-строку с помощью метода json\_encode и отправляется клиенту, который получает данные выборки из строки с помощью метода JSON.parse и формирует вывод пользователю).

Обратите внимание, необходимо показать применение всех трех форматов ответа от сервера - по одному формату на каждый из трех запросов на выборку.

Исходный код PHP-скриптов:

**index.php**:

<?php

require "connection.php";

require "tables/nurses.php";

require "tables/wards.php";

?>

<!doctype html>

<html lang="ru">

<head>

<meta charset="UTF-8">

<meta name="viewport"

content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>Лабораторная 3. Кравчик К. В.</title>

<script src="https://code.jquery.com/jquery-3.5.1.min.js" integrity="sha256-9/aliU8dGd2tb6OSsuzixeV4y/faTqgFtohetphbbj0=" crossorigin="anonymous"></script>

<script src="ajax/ajax.js"></script>

</head>

<body>

<section class="container">

<!--Первое задание-->

<h5>Перечень палат, в которых дежурит выбранная медсестра</h5>

<form action="queries/wards-by-nurse.php" class="form-group">

<select name="nurse\_id">

<?php

foreach ($nurses as $n) {

echo "<option value=\"". $n['id'] ."\">". $n['name'] ."</option>";

}

?>

</select>

<input type="button" value="Отправить" onclick="wardsByNurse(this)">

</form>

<div id="result1"></div>

<!--Второе задание-->

<h5>Медсестры выбранного отделения</h5>

<form action="queries/nurses-by-department.php" class="form-group">

<select name="depart">

<?php

foreach ($departments as $d) {

echo "<option value=\"". $d ."\">". $d ."</option>";

}

?>

</select>

<input type="button" value="Отправить" onclick="nursesByDepartment(this)">

</form>

<table id="result2">

<thead>

<tr>

<th>ID</th>

<th>Имя</th>

<th>Дата</th>

<th>Отделение</th>

<th>Смена</th>

</tr>

</thead>

<tbody></tbody>

</table>

<!--Третье задание-->

<h5>Дежурства (в любых палатах) в указанную смену</h5>

<form action="queries/wards-by-shift.php" class="form-group">

<select name="shift">

<?php

foreach ($shifts as $d) {

echo "<option value=\"". $d ."\">". $d ."</option>";

}

?>

</select>

<input type="button" value="Отправить" onclick="wardsByShift(this)">

</form>

<table id="result3">

<thead>

<tr>

<th>ID</th>

<th>Имя</th>

<th>Дата</th>

<th>Отделение</th>

<th>Смена</th>

<th>ID Палаты</th>

<th>Палата</th>

</tr>

</thead>

<tbody></tbody>

</table>

</section>

</body>

</html>

**connection.php**:

<?php

$dsn = "mysql:host=localhost; dbname=iteh2lb1var4";

$user = "root";

$pass = "";

try {

$dbh = new PDO($dsn, $user, $pass);

$dbh->exec('SET NAMES utf8');

} catch (PDOException $e) {

echo "ERROR!! "."$e->getMessage()";

}

**tables/nurses.php**:

<?php

require "./connection.php";

$sqlSelect = "SELECT \* FROM `nurse`";

$nurses = array();

$departments = array();

$shifts = array();

foreach ($dbh->query($sqlSelect) as $n) {

$nurses[] = array(

'id' => $n['id\_nurse'],

'name' => $n['name'],

'date' => $n['date'],

'department' => $n['department'],

'shift' => $n['shift'],

);

}

foreach ($nurses as $item) {

$departments[] = $item['department'];

}

foreach ($nurses as $item) {

$shifts[] = $item['shift'];

}

$shifts = array\_unique($shifts);

$departments = array\_unique($departments);

**tables/wards.php**:

<?php

require "./connection.php";

$sqlSelect = "SELECT \* FROM `ward`";

$wards = array();

foreach ($dbh->query($sqlSelect) as $n) {

$wards[] = array(

'id' => $n['id\_ward'],

'name' => $n['name'],

);

}

**queries/nurses-by-department.php**:

<?php

require "../connection.php";

$sqlSelect = "SELECT \* FROM `nurse` `n` WHERE `n`.department = :depart";

$sth = $dbh->prepare($sqlSelect, array(PDO::ATTR\_CURSOR => PDO::CURSOR\_FWDONLY));

$sth->execute(array(':depart' => $\_GET['depart']));

$items = array();

foreach ($sth as $index => $row) {

$items[] = array(

'id\_nurse' => $row['id\_nurse'],

'name' => $row['name'],

'date' => $row['date'],

'department' => $row['department'],

'shift' => $row['shift'],

);

}

echo '<?xml version="1.0" encoding="utf8" ?>';

echo "<root>";

foreach ($items as $value) :

echo "

<nurse>

<id\_nurse>".$value['id\_nurse']."</id\_nurse>

<name>".$value['name']."</name>

<date>".$value['date']."</date>

<department>".$value['department']."</department>

<shift>".$value['shift']."</shift>

</nurse>";

endforeach;

echo "</root>";

**queries/wards-by-nurse.php**:

<?php

require "../connection.php";

$sqlSelect = "SELECT DISTINCT

`w`.id\_ward,

`w`.name

FROM `nurse\_ward` `nw`

JOIN `ward` `w` on `nw`.fid\_ward = `w`.`id\_ward`

WHERE `nw`.fid\_nurse = :nurse\_id";

$sth = $dbh->prepare($sqlSelect, array(PDO::ATTR\_CURSOR => PDO::CURSOR\_FWDONLY));

$sth->execute(array(':nurse\_id' => $\_GET['nurse\_id']));

$items = array();

foreach ($sth as $index => $row) {

$items[] = array(

'id\_ward' => $row['id\_ward'],

'name' => $row['name'],

);

}

?>

<table>

<thead>

<tr>

<th>ID</th>

<th>Имя</th>

</tr>

</thead>

<tbody>

<?php

foreach ($items as $value) : ?>

<tr>

<td><?php echo $value['id\_ward']; ?></td>

<td><?php echo $value['name']; ?></td>

</tr>

<?php endforeach; ?>

</tbody>

</table>

**queries/wards-by-shift.php**:

<?php

require "../connection.php";

$sqlSelect = "SELECT

`n`.id\_nurse,

`n`.name,

`n`.date,

`n`.department,

`n`.shift,

`w`.id\_ward,

`w`.name as 'ward\_name'

FROM `nurse\_ward` `nw`

JOIN `nurse` `n` on `n`.id\_nurse = `nw`.fid\_nurse

JOIN `ward` `w` on `w`.`id\_ward` = `nw`.fid\_ward

WHERE `n`.shift = :shift

";

$sth = $dbh->prepare($sqlSelect, array(PDO::ATTR\_CURSOR => PDO::CURSOR\_FWDONLY));

$sth->execute(array(':shift' => $\_GET['shift']));

$items = array();

foreach ($sth as $index => $row) {

$items[] = array(

'id\_nurse' => $row['id\_nurse'],

'name' => $row['name'],

'date' => $row['date'],

'department' => $row['department'],

'shift' => $row['shift'],

'id\_ward' => $row['id\_ward'],

'ward\_name' => $row['ward\_name'],

);

}

echo json\_encode($items);

**ajax/ajax.js**:

// HTML

function wardsByNurse(e){

var currentForm = $(e).parents('form');

$.ajax({

type: "GET",

url: "queries/wards-by-nurse.php",

data: currentForm.serialize(),

success: function(result){

$('#result1').html(result);

}

});

}

// XML

function nursesByDepartment(e){

var currentForm = $(e).parents('form');

$.ajax({

type: "GET",

url: "queries/nurses-by-department.php",

data: currentForm.serialize(),

dataType: "xml",

success: function(result){

var output = "";

$(result).find('nurse').each(function(){

output += '<tr>'+

'<td>' + $(this).find('id\_nurse').text() + '</td>' +

'<td>' + $(this).find('name').text() + '</td>' +

'<td>' + $(this).find('date').text() + '</td>' +

'<td>' + $(this).find('department').text() + '</td>' +

'<td>' + $(this).find('shift').text() + '</td>' +

'</tr>';

});

$('#result2 tbody').html(output);

},

error: function(data){

console.log(data);

}

});

}

// JSON

function wardsByShift(e){

var currentForm = $(e).parents('form');

$.ajax({

type: "GET",

url: "queries/wards-by-shift.php",

data: currentForm.serialize(),

dataType: "json",

success: function(result){

var output = "";

// var items = JSON.parse(result);

result.forEach(item => {

output += '<tr>'+

'<td>' + item.id\_nurse + '</td>' +

'<td>' + item.name + '</td>' +

'<td>' + item.date + '</td>' +

'<td>' + item.department + '</td>' +

'<td>' + item.shift + '</td>' +

'<td>' + item.id\_ward + '</td>' +

'<td>' + item.ward\_name + '</td>' +

'</tr>';

});

$('#result3 tbody').html(output);

}

});

}

### Результаты выполнения асинхронных запросов в виде снимков экрана браузера

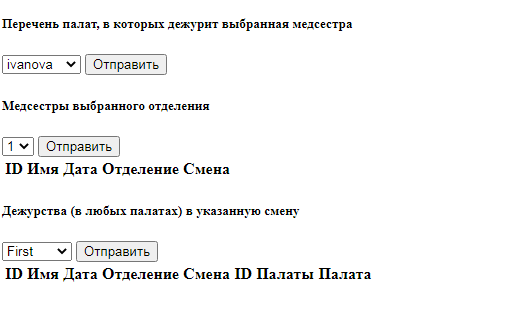


Рисунок 3.1 – Главная страница

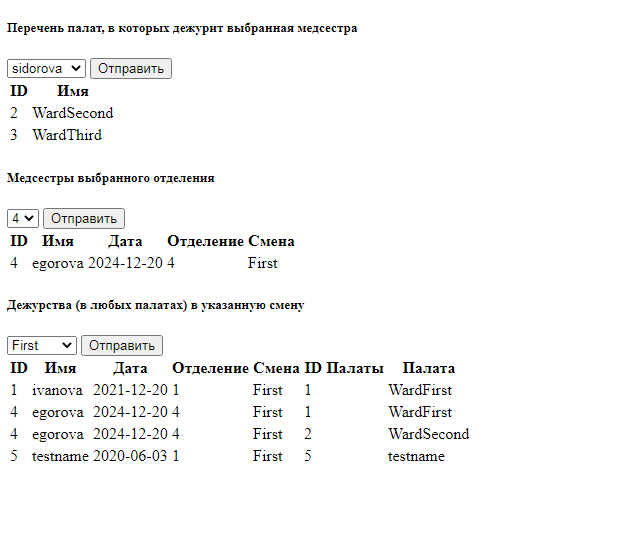


Рисунок 3.2 – Результаты выполнения запросов

**Выводы**: в ходе лабораторной работы были изучены методы и получены практические навыки работы технологии AJAX для обновления данных без перезагрузки страницы.