МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ

ХАРЬКОВСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ РАДИОЭЛЕКТРОНИКИ

Кафедра ЭВМ

**ОТЧЕТ**

по лабораторной работе №2 на тему:

**«НЕРЕЛЯЦИОННАЯ СУБД MONGODB И ХРАНЕНИЕ ДАННЫХ НА СТОРОНЕ КЛИЕНТА»**

по курсу «Internet-технологии»

Выполнил: Проверила:

ст. гр. КИУКИу–18–2 Бологова Н.Н.

Кравчик К.В.

Харьков 2020

**2.1 Цель работы**

Изучение методов для работы с нереляционной СУБД MongoDB в языке PHP.

Знакомство с хранением данных на стороне клиента, используя WebStorage.

**2.2 Задание на лабораторную работу**

**Вариант 3.** Создать и заполнить БД для хранения информации о дежурствах в поликлинике. Дежурство описывается сменой (первая, вторая, третья), датой, задействованными медсестрами (не менее одной), названием отделения, номерами используемых на этом дежурстве палат.

Предоставить пользователю возможность получения следующей информации:

* перечень палат, в которых дежурит выбранная медсестра;
* медсестры, дежурившие в указанном отделении;
* всех дежурствах (в любых палатах) в указанную смену в указанном отделении.

**Исходный код PHP-скриптов:**

**index.php:**

<?php

require 'connection.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 2. 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>

<style>tr,td,th{padding: 10px;}</style>

</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">

<input type="text" id="nurseName" name="nurseName"><br>

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

<input type="button" value="Данные с LocalStorage" onclick="wardsByNurseLocal(this)">

</form>

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

<div id="result1local" style="display: none">

<span>Local storage:</span>

<div id="result1localinternal"></div>

</div>

<!--Второе задание-->

<h5>Медсестры, дежурившие в указанном отделении</h5>

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

<input type="text" id="dept" name="dept"><br>

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

<input type="button" value="Данные с LocalStorage" onclick="nurseByDeptLocal(this)">

</form>

<table id="result2">

<thead>

<tr>

<th>Имя</th>

<th>ID</th>

</tr>

</thead>

<tbody></tbody>

</table>

<div id="result2local" style="display: none">

<span>Local storage:</span>

<table id="result2localinternal">

<thead>

<tr>

<th>Имя</th>

<th>ID</th>

</tr>

</thead>

<tbody></tbody>

</table>

</div>

<!--Третье задание-->

<h5>Всех дежурствах (в любых палатах) в указанную смену в указанном отделении</h5>

<form action="queries/duty-by-ward-and-shift.php" class="form-group">

<input type="text" id="dept" name="dept"><br>

<input type="text" id="shift" name="shift"><br>

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

<input type="button" value="Данные с LocalStorage" onclick="dutyByWardAndShiftLocal(this)">

</form>

<table id="result3">

<thead>

<tr>

<th>Смена</th>

<th>Дата</th>

<th>Медсестры</th>

<th>Отделение</th>

<th>Палаты</th>

</tr>

</thead>

<tbody></tbody>

</table>

<div id="result3local" style="display: none">

<span>Local storage:</span>

<table id="result3localinternal">

<thead>

<tr>

<th>Смена</th>

<th>Дата</th>

<th>Медсестры</th>

<th>Отделение</th>

<th>Палаты</th>

</tr>

</thead>

<tbody></tbody>

</table>

</div>

</section>

</body>

</html>

**connection.php:**

<?php

require\_once \_\_DIR\_\_ .'/vendor/autoload.php'; // подключаем автоподгрузчик классов Composer

$db = (new MongoDB\Client)->dbforlab;

**queries/duty-by-ward-and-shift.php:**

<?php

require "../connection.php";

$cursor = $db->duties->find(["departmentName" => $\_GET['dept'], "shift" => $\_GET['shift']], [ 'projection' => ['\_id' => 0] ]);

echo json\_encode(iterator\_to\_array($cursor));

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

<?php

require "../connection.php";

$cursor = $db->duties->find(["departmentName" => $\_GET['dept']], [ 'projection' => ['\_id' => 0, "nurses" => 1] ]);

echo json\_encode(iterator\_to\_array($cursor));

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

<?php

require "../connection.php";

$nurseName = $\_GET['nurseName'];

$query = ["nurses.name" => ['$all' => [ $nurseName ]]];

$cursor = $db->duties->find($query, [ 'projection' => ['\_id' => 0, "wards" => 1] ]);

echo json\_encode(iterator\_to\_array($cursor));

**ajax/ajax.js:**

// 1

function wardsByNurse(e){

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

$.ajax({

type: "GET",

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

data: currentForm.serialize(),

dataType: "json",

success: function(result){

const nurse = document.getElementById("nurseName").value;

let finalResult = [];

result.forEach(x => finalResult = [...finalResult, ...x.wards]);

finalResult = finalResult.filter((item, pos) => finalResult.indexOf(item) === pos);

localStorage.setItem('first' + nurse, finalResult.join(','));

$('#result1').html(finalResult.join(','));

}

});

}

// 2

function nurseByDept(e){

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

$.ajax({

type: "GET",

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

data: currentForm.serialize(),

dataType: "json",

success: function(result){

let finalResult = [];

const dept = document.getElementById("dept").value;

result.forEach(x => finalResult = [...finalResult, ...x.nurses]);

finalResult = finalResult.filter((item, pos) => finalResult.indexOf(item) === pos);

var output = "";

finalResult.forEach(item => {

output += '<tr>'+

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

'<td>' + item.\_id + '</td>' +

'</tr>';

});

localStorage.setItem('second'+dept, output);

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

}

});

}

// 3

function dutyByWardAndShift(e){

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

$.ajax({

type: "GET",

url: "queries/duty-by-ward-and-shift.php",

data: currentForm.serialize(),

dataType: "json",

success: function(result){

var output = "";

const dept = document.getElementById("dept").value;

const shift = document.getElementById("shift").value;

result.forEach(item => {

output += '<tr>'+

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

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

'<td>' + JSON.stringify(item.nurses) + '</td>' +

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

'<td>' + JSON.stringify(item.wards) + '</td>' +

'</tr>';

});

localStorage.setItem(`third-${dept}-${shift}`, output);

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

}

});

}

// LOCAL STORAGE

// 1

function wardsByNurseLocal(e){

const nurse = document.getElementById("nurseName").value;

const result = localStorage.getItem('first' + nurse);

if (result) {

$('#result1local').css('display', 'block');

$('#result1localinternal').html(result);

}

else {

$('#result1local').css('display', 'block');

$('#result1localinternal').html("Local storage is empty");

}

}

// 2

function nurseByDeptLocal(e){

const dept = document.getElementById("dept").value;

const result = localStorage.getItem('second'+dept);

if (result) {

$('#result2local').css('display', 'block');

$('#result2localinternal').html(result);

}

else {

$('#result2local').css('display', 'block');

$('#result2localinternal').html("Local storage is empty");

}

}

// 3

function dutyByWardAndShiftLocal(e){

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

const dept = document.getElementById("dept").value;

const shift = document.getElementById("shift").value;

const result = localStorage.getItem(`third-${dept}-${shift}`);

if (result) {

$('#result3local').css('display', 'block');

$('#result3localinternal').html(result);

}

else {

$('#result3local').css('display', 'block');

$('#result3localinternal').html("Local storage is empty");

}

}

**Результаты выполнения запросов в виде снимков экрана браузера**:

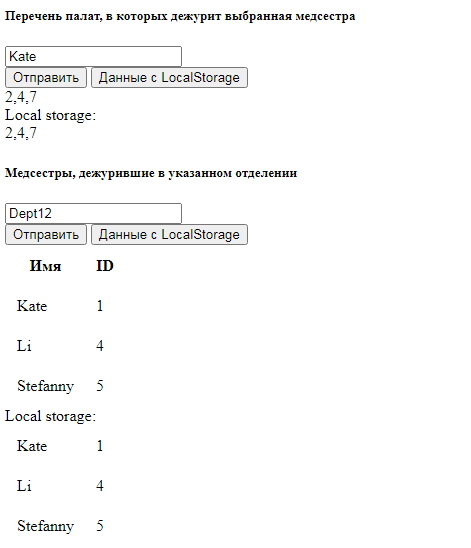


Рисунок 1.1 – Результат выполнения первых двух запросов

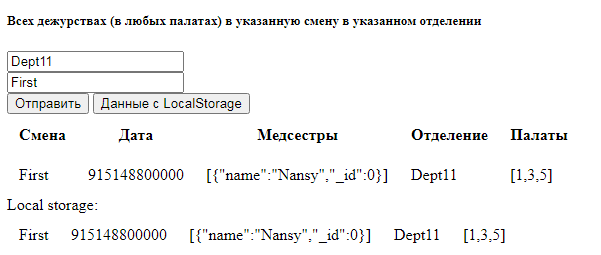


Рисунок 1.2 – Результат выполнения третьего запроса

**Вывод**: в ходе лабораторной работы были изучены методы работы и получен практический опыт работы с базой данных MongoDB