

Отчет по лабораторной работе № 3

«Простая и сложная реакция на движущийся объект»

Выполнила группа «думаем»

Студенты группы

Белоусова Анна Константиновна Р3125

Гафурова Фарангиз Фуркатовна Р3120

Гольцман Глеб Михайлович Р3125

Федоров Данил Максимович Р3125

Садовников Олег Юрьевич Р3125

Заборщиков Артем Тарасович Р3124

Павловская Анна Алексеевна Р3125

Михайлов Степан Сергеевич Р3124

г. Санкт-Петербург

2024

Цель лабораторной работы:

Научиться разрабатывать системы в проектной деятельности, разработать систему оценки простых и сложных сенсомоторных реакций человека, как первый элемент батареи тестов.

Задачи:

1. Используя html, css, как минимум, разработать систему оценки простых и сложных сенсомоторных реакций человека (класс тестов в проекте – «Реакция на движущийся объект»)
2. Оценка простой реакции человека на движущийся объект (варианты):
 - Выбор времени выполнения (от 120 секунд до 45 минут)
 - С отражением времени выполнения/или без;
 - Отражением результата за минуту и в целом за тест/ без отражения;
 - Отображение прогресса выполнения теста;
 - Ускорение движения объекта – на сколько и через какой период времени и как часто;
3. Оценка сложной реакции человека на движущиеся объекты:
 - Выбор времени выполнения (от 120 секунд до 45 минут)
 - С отражением времени выполнения/или без;
 - Отражением результата за минуту и в целом за тест/ без отражения;
 - Отображение прогресса выполнения теста;
 - Ускорение движения объектов – на сколько и через какой период времени и как часто;
4. Система должна позволять:
 - Отображать прогресс выполнения теста;
 - Хранить общий список респондентов;
 - Хранить результаты тестирования;
 - Смотреть динамику результатов одного респондента одного вида теста, который выполнялся несколько раз;
 - Нормировать результаты с учетом пола, возраста ± 5 лет по текущей выборке;
 - Система отображения результатов тестирования для эксперта:
 - Результаты отдельного теста
 - Результаты динамики одного вида теста
 - Результаты всех тестов
 - Система отображения результатов тестирования для респондента (подумать зачем нужен этот пункт и в выводах отразить ответ):
 - Результаты отдельного теста
 - Результаты динамики одного вида теста
 - Результаты всех тестов
 - Возможность эксперту произвольно выбирать, какие тесты необходимо пройти респонденту и в каком порядке;
 - Различные варианты регистрации пользователей/респондентов/экспертов – предложить и обосновать;
 - Возможность проходить тестирование по ссылке-приглашению

Описание лабораторной работы:

Папки проекта:

- [Корневая директория](#) – тут лежат страницы веб-приложения.
- [css](#) – файлы стилей.
- [img](#) – картинки.
- [js](#) – js скрипты.
- [vendor](#) – js и css для внешнего вида взятой за основу админки "SB Admin 2".
- [scripts](#) – служебные бэкенд скрипты для работы с БД и сессиями.
- [tests](#) – [test.php](#) – родительский файл страницы с тестами, все остальные страницы ([colors.php](#), [light.php](#) и т. д.) — это шаблоны, вставляющиеся в родительский файл в зависимости от номера теста, который проходит пользователь.

База данных:

- tests - список тестов с названиями и ссылкой на файл (например, [light.php](#)).
- users - список юзеров.
- user_tests - таблица, связывающая юзеров и тесты (поля test_id и user_id вместе образуют первичный ключ, поэтому не могут дублироваться).
- user_test_results таблица, содержащая результаты прохождения теста юзером (поля test_id и user_id могут дублироваться, как угодно, так как возможно повторное прохождение теста).

Как работает регистрация/авторизация:

Страницы [login.html](#), [register.html](#) – интерфейсы входа и регистрации.

Скрипты [login.php](#), [reg.php](#), [logout.php](#) реализуют авторизацию/ регистрацию/выход.

Файл [gates.php](#) используется почти всеми скриптами и страницами приложения, т. к. в нем прописан функционал проверки авторизации и прав доступа. Например, с помощью функции `check_auth()`.

```
function check_auth(){
    if(!getStatus()){
        header("Location: ../login.html");
        exit();
    }
}
```

Так же с используя функцию из этого файла можно получить текущего авторизованного пользователя:

```

function getCurrentUser(){
    $pdo = getPDO();
    $user = null;
    if(isset($_COOKIE['auth_email'])) {
        $email = $_COOKIE['auth_email'];
        $result = $pdo->query("SELECT * FROM users WHERE email = '$email'");
        $user = $result->fetch(PDO::FETCH_ASSOC);
    }
    return $user;
}

```

Управление юзерами

На странице [users.php](#) выводятся все пользователи и функции (кнопки) для работы с ними (редактировать, изменить набор тестов, посмотреть результаты). Эти функции доступны только экспертам и админам (при чем редактировать статус юзера может только админ).

Все эти кнопки ведут на другие страницы, функционал которых в целом несложный.

Стоит обратить внимание на то, как выводятся результаты тестирования юзера (код идентичен в файле [main.php](#), и в [test_results.php](#).

```

<?php
$first_res = 0;
$last_res = 0;
$max = max(array_column($test['results'], 'result'));
if (isset($test['results'][0])) {
    $first_res = $test['results'][0]['result'];
    $last_res = $test['results'][sizeof($test['results']) - 1]['result'];
}
$dynamic = $last_res - $first_res;
$dynamic_percent = round($last_res / $max * 100 - $first_res / $max * 100, 2);
foreach ($test['results'] as $result):
    ?>
    <h4 class="small font-weight-bold mt-3"><?= $result['date'] ?><span
        class="float-right"><?= $result['result'] ?> мс</span></h4>
    <div class="progress mb-4">
        <div class="progress-bar bg-success" role="progressbar" style="width: <?= $result['r
            aria-valuenow="20" aria-valuemin="0" aria-valuemax="100"></div>
    </div>
<?php endforeach; ?>

<h5 class="font-weight-bold mt-4">Динамика: <?= (($dynamic >= 0) ? "+" : "-") . abs($dynamic)
    (<?= (($dynamic >= 0) ? "ухудшилась" : "улучшилась") ?> на <?= abs($dynamic_percent) ?>%

```

Приглашение для прохождения теста

На странице [generate_invitation_link.php](#) эксперт или админ может сгенерировать ссылку для отправки респонденту. При этом в БД “регистрируется” пользователь без пароля, который может войти на сайт только по этой ссылке. Этот пользователь появится в общем списке пользователей, где ему нужно будет назначить тесты. Иначе, перейдя по ссылке, он увидит сообщение:

```
<?php if(!$test){ ?>
    <div class="d-sm-flex align-items-center justify-content-between mb-4">
        <h1 class="h3 mb-0 text-gray-800">К сожалению, тесты для Вас еще не подготовили, сообщите об этом своему эксперту и вернитесь немного позже</h1>
    </div>
<?php } else { ?>
```

Тесты

В папке [scripts](#) лежит очень полезный файл [get_user_tests.php](#). С помощью функций, описанных в нем, мы можем получить список тестов для определенного юзера. Получив этот список в файле [test.php](#), мы начинаем по очереди “подсовывать” пользователю тесты, подключая соответствующий файл

```
<?php include $test['link']; ?>
```

В этом файле есть форма, которая отправляется по завершении теста (отправляется в скрипт [scripts/save_test_results.php](#)), где результаты сохраняются в БД, а также вычисляется id следующего теста

```
$next_test_id = getNextTestForCurrentUser($test_id)['test_id'];
```

```
if($result){
    echo "ok";
    header("Location: ../tests/test.php?id=$next_test_id");
}
```

Пользователь снова попадает на страницу [test.php](#), где ему “подсовывается” уже следующий тест. Когда тесты кончаются, происходит редирект на страницу [main.php](#).

Код в [simple_rdo.php](#) представляет собой интерактивный тест, в которой пользователю предлагается нажимать на клавишу «Пробел» в момент, когда шарик пересекает самую

верхнюю точку окружности. Время теста составляет 1 минуту. Так же, функция `reaction(key)` определяет отклонение реакции пользователя от заданного момента, а функция `endTest()` завершает тест, подсчитывает и отображает результаты тестирования.

Код в [hard_rdo.php](#) – интерактивный тест для проверки реакции пользователя на сигналы. На странице есть инструкция, объясняющая, как проходить тест, и форма с круглыми секциями разных цветов, где пользователь должен нажимать на клавиши в соответствии с цветом мяча, пересекающего верхнюю точку секции. JavaScript отвечает за анимацию движения мячей по окружности, установка времени теста, проверка правильности реакции пользователя и отправка результатов на сервер. Весь код организован таким образом, чтобы создать виртуальное тестирование скорости реагирования.

Как устроены файлы самих тестов ([sound.php](#), [count_visual.php](#) и т.д.)

Общий шаблон:

```
<form action="../../scripts/save_test_results.php" method="post">
//Тут вся верстка теста
</form>

<script>
//Самое главное - js сценарий теста
/*
Он состоит из
глобальных переменных (var test_start = false; var total_score = 0;)
таймера (function countdown()),
обработчика нажатий клавиш ( $('body').keydown(function (e)... )
функций, отражающих жизненный цикл теста (testStart(),light(),reaction(),endTest())

*/
</script>

<style>
//Стили теста
</style>
```

Для включения сигнала взята функция из интернета (<https://ru.stackoverflow.com/questions/1374087/Как-установить-на-событие-Звуковой-сигнал>)

Для проговаривания математического выражения использовался api данного сервиса (<https://responsivevoice.org/>). Поэтому для работы этого теста нужен интернет (можно было положить несколько mp3 файлов с аудиозаписью названий цифр, но мы посчитали что этот вариант интереснее) Тем более можно заставить его говорить что угодно, просто изменив первый аргумент вот здесь:

```
responsiveVoice.speak(num + " + " + num2, "Russian Female");
```

Вывод:

В ходе выполнения третьей лабораторной работы была разработана и реализована система оценки простых и сложных реакций человека на движущийся объект, включающая использование языков разметки HTML и CSS, а также языков программирования PHP, MySQL и JavaScript.

Для оценки простой и сложной реакций на движущиеся объекты предусмотрены различные опции, включая выбор времени выполнения теста, отображение результатов, прогресса выполнения теста, настройку скорости и ускорения движения объектов.

В системе реализована возможность отображения прогресса выполнения теста, хранения списка респондентов и результатов тестирования, нормирования результатов с учетом различных критериев, а также просмотра результатов для экспертов и респондентов.

Экспертам доступен просмотр результатов тестирования, анализ динамики результатов и выбор тестов для респондентов, а респондентам предоставлена возможность просмотра и анализа своих результатов.