

**Министерство науки и высшего образования Российской
Федерации ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ИТМО
ITMO University**

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3

По дисциплине Web-программирование

Обучающийся Шестаков Максим Олегович

Факультет Инфокоммуникационных технологий

Группа K3321

**Направление подготовки 11.03.02 Инфокоммуникационные
технологии и системы связи**

**Образовательная программа
Программирование в инфокоммуникационных
системах**

Обучающийся	<u>24.11.2024</u> (дата)	<u> </u> (подпись)	<u>Шестаков М.О.</u> (Ф.И.О.)
Руководитель	<u> </u> (дата)	<u> </u> (подпись)	<u>Марченко Е.В.</u> (Ф.И.О.)

Санкт-
Петербург
2024 г.

СОДЕРЖАНИЕ

Стр.

ВВЕДЕНИЕ	3
1 Ход работы	4
1.1 Задание №1	4
1.2 Задание №2	4
1.3 Задание №3	6
ЗАКЛЮЧЕНИЕ	7

ВВЕДЕНИЕ

Цель работы: Изучение PHP-скриптов обработки форм отправки web-документов

1 Ход работы

1.1 Задание №1

В рамках первого упражнения были созданы две задачи, которые выполнялись сначала параллельно, а потом последовательно. По логам данный порядок можно отследить, но на времени выполнения это не отразилось, поскольку Node.js - однопоточная среда выполнения.

Рисунок 1.1 — Сравнение разных сценариев выполнения

Далее в упражнении требовалось создать задачу, которая бы отслеживала изменения в целевых файлах.

Рисунок 1.2 — Задача на перезагрузку страницы при изменении

1.2 Задание №2

Во втором задании требовалось создать форму для отправки информации на обработку в php-скрипт. Для этого предварительно была создана html-страница:

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <link rel="stylesheet" href="css/second.css">
7   <title>Second task</title>
8 </head>
9 <body>
10  <form action="script.php" method="post">
11    <h2>Форма регистрации</h2>
12    Введите имя: <input type="text" id="name" name="name"> </br>
13    Введите фамилию: <input type="text" id="surname" name="surname"> </br>
14    Почта: <input type="text" id="email" name="email"></br>
15    Род деятельности: </br>
16    <input type="radio" id="option1" name="radio-options" value="student">Студент </br>
17    <input type="radio" id="option2" name="radio-options" value="employee">Сотрудник</br>
18    <input type="radio" id="option3" name="radio-options" value="other">Другое</br>
19    Увлечения:</br>
20    <input type="checkbox" id="option4" name="check-options[]" value="sport">Спорт</br>
21    <input type="checkbox" id="option5" name="check-options[]" value="music">Музыка</br>
22    <input type="checkbox" id="option6" name="check-options[]" value="other">Путешествия</br>
23    <input type="checkbox" id="option7" name="check-options[]" value="other">Другое</br>
24    <textarea name="feedback" id="feedback">Дополнительная информация: </textarea></br>
25    <button type="submit" id="button">Отправить</button></br>
26  </form>
27 </body>
28 </html>

```

Рисунок 1.3 — HTML-страница

Далее был написан php-скрипт, через который отдавался результат обработки - выводились введенные пользователем данные по категориям.

```

1 <?php
2 if (!empty($_POST['name']) &&
3     !empty($_POST['surname']) &&
4     !empty($_POST['email']) &&
5     !empty($_POST['feedback']) &&
6     !empty($_POST['radio-options']) &&
7     !empty($_POST['check-options'])) {
8     $name = $_POST['name'];
9     $surname = $_POST['surname'];
10    $email = $_POST['email'];
11    $feedback = $_POST['feedback'];
12    $activity = $_POST['radio-options'];
13    $hobbies = $_POST['check-options'] ?? [];
14
15    echo "Имя: $name<br>";
16    echo "Фамилия: $surname<br>";
17    echo "Электронная почта: $email<br>";
18    echo "Дополнительная информация: $feedback<br>";
19    echo "Выбранный род деятельности: $activity<br>";
20    echo "Выбранные хобби: ";
21    echo implode(", ", $hobbies);
22  } else {
23    echo "Форма не была заполнена";
24  }
25 ?>

```

Рисунок 1.4 — PHP-скрипт

В задании также было необходимо привести информацию о запросах GET и POST.

1. GET-запросы используются для получения данных с веб-сервера. Допускаются параметры, используемые как часть URL, а также параметры в строке запроса в виде пар ключ-значение.
2. POST-запросы используются для отправки данных на сервер. Допускается использование тела запроса, а также параметров в строке запроса. Возможен возврат тела ответа в качестве результата работы.

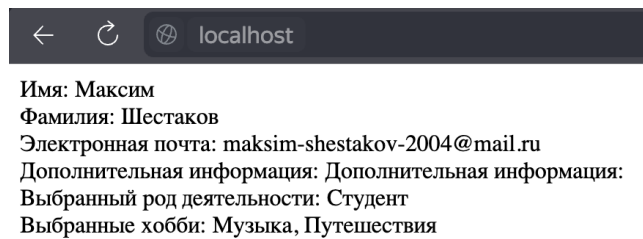


Рисунок 1.5 — Результат исполнения скрипта

1.3 Задание №3

В третьем задании необходимо было установить инструментарий для отладки проектов. Так как в заданиях использовался PHP-скрипт, был выбран инструментарий ХАМРР, в котором есть поддержка сервера Apache. После включения сервера и добавления скрипта с html-документом в папку htdocs получилось добиться результата выполнения скрипта.

ЗАКЛЮЧЕНИЕ

В результате выполнения лабораторной работы были изучены методы параллельного и последовательного запуска тасок в `gulp`, создан первый РНР-скрипт с соответствующей формой отправки, а также установлен и опробован инструмент `XAMPP`.