

Санкт-Петербургский Национальный Исследовательский

Университет ИТМО

Факультет «Информационных технологий и программирования»

Направление подготовки «Инфокоммуникационные технологии и системы
связи»

Практическая работа №3

Выполнила:

Бакланова А.Г.

Группа: К3322

Проверил:

Марченко Е.В.

Санкт-Петербург,

2024

Содержание

Цель работы:	3
Ход работы.....	4
1. Задание 1 – Настроить GULP	4
2. Задание 2 – Форма для отправки информации	9
3. Задание 3 – Инструментарий для отладки проектов.....	12
Заключение	15

Цель работы:

Применяя навыки web-программирования необходимо настроить gulp, создать web-сайт, на котором будет отображаться форма для отправки информации об обратной связи, создать php скрипт, который будет отправлять полученные данные из формы в базу данных и настроить портал в Wordpress.

Ход работы

1. Задание 1 – Настроить GULP

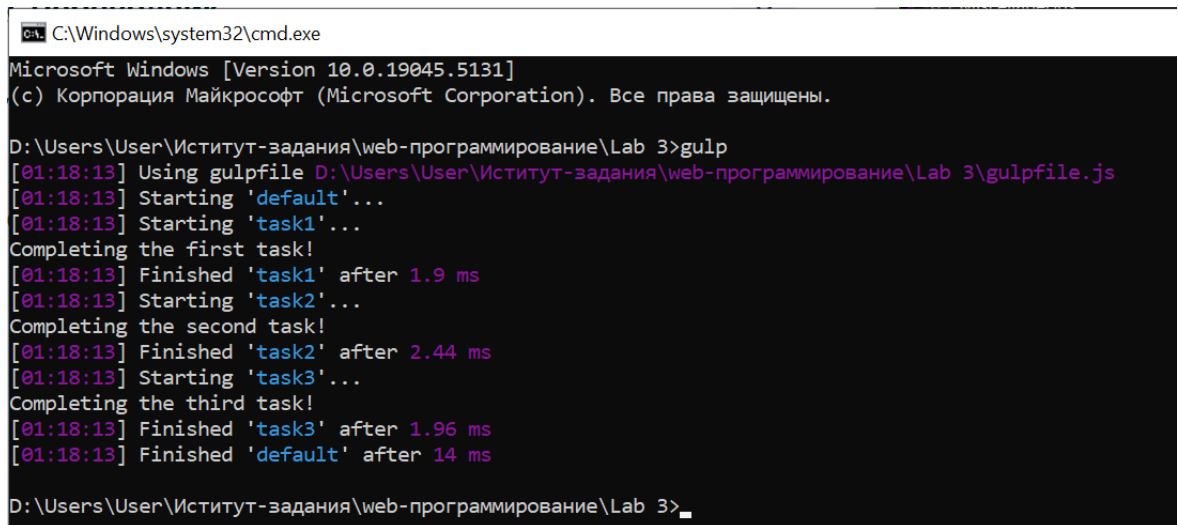
Первое задание включает в себя следующие пункты: настроить gulp: а) создать два таска – настроить на последовательное и параллельное выполнение; б) настроить отображение файлов проекта в браузере и автоматическую перезагрузку при изменении одного из контролируемых файлов проекта.

Для настройки последовательного выполнения тасков была использована функция `series()`. Код с последовательным выполнением представлен на рисунке 1.

```
1  const { series } = require('gulp');
2
3
4  1 reference
   function task1(cb) {
5     console.log('Completing the first task!');
6     cb();
7   }
8
9  1 reference
   function task2(cb) {
10    console.log('Completing the second task!');
11    cb();
12  }
13
14  0 references
   function task3(cb) {
15    console.log('Completing the third task!');
16    cb();
17  }
18
19
20  const sequentialTask = series(task1, task2, task3);
21
22
23  exports.default = sequentialTask;
24
```

Рисунок 1 – Код с последовательным выполнением тасков

На рисунке 2 представлен результат кода при запуске gulpfile.js через КОНСОЛЬ.



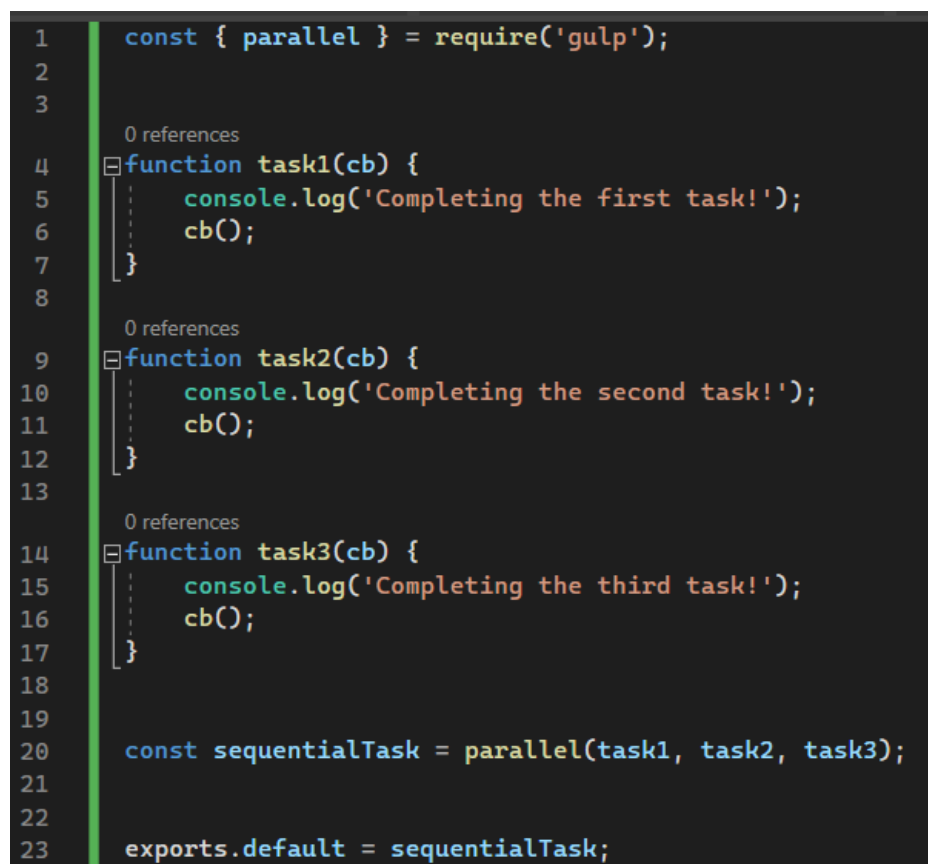
```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.19045.5131]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

D:\Users\User\Иститут-задания\web-программирование\Lab 3>gulp
[01:18:13] Using gulpfile D:\Users\User\Иститут-задания\web-программирование\Lab 3\gulpfile.js
[01:18:13] Starting 'default'...
[01:18:13] Starting 'task1'...
Completing the first task!
[01:18:13] Finished 'task1' after 1.9 ms
[01:18:13] Starting 'task2'...
Completing the second task!
[01:18:13] Finished 'task2' after 2.44 ms
[01:18:13] Starting 'task3'...
Completing the third task!
[01:18:13] Finished 'task3' after 1.96 ms
[01:18:13] Finished 'default' after 14 ms

D:\Users\User\Иститут-задания\web-программирование\Lab 3>_
```

Рисунок 2 – Результат последовательного выполнения задач

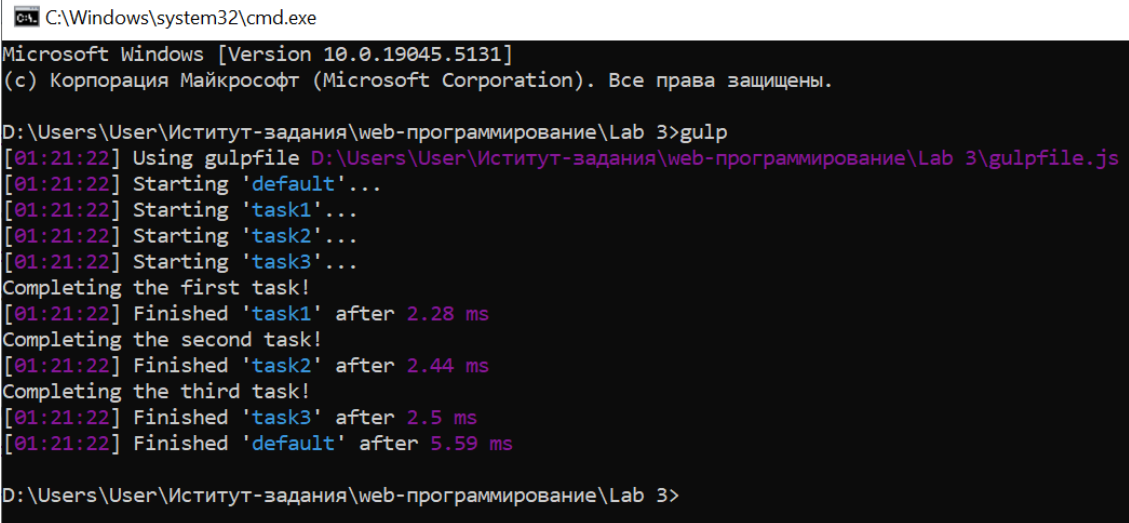
Для настройки параллельного выполнения задач была использована функция `parallel()`. Код с параллельным выполнением представлен на рисунке 3.



```
1  const { parallel } = require('gulp');
2
3
4  function task1(cb) {
5    console.log('Completing the first task!');
6    cb();
7  }
8
9  function task2(cb) {
10   console.log('Completing the second task!');
11   cb();
12 }
13
14 function task3(cb) {
15   console.log('Completing the third task!');
16   cb();
17 }
18
19
20 const sequentialTask = parallel(task1, task2, task3);
21
22
23 exports.default = sequentialTask;
```

Рисунок 3 – Код с параллельным выполнением задач

На рисунке 4 представлен результат кода при запуске gulpfile.js через КОНСОЛЬ.



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.19045.5131]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

D:\Users\User\Иститут-задания\web-программирование\Lab 3>gulp
[01:21:22] Using gulpfile D:\Users\User\Иститут-задания\web-программирование\Lab 3\gulpfile.js
[01:21:22] Starting 'default'...
[01:21:22] Starting 'task1'...
[01:21:22] Starting 'task2'...
[01:21:22] Starting 'task3'...
Completing the first task!
[01:21:22] Finished 'task1' after 2.28 ms
Completing the second task!
[01:21:22] Finished 'task2' after 2.44 ms
Completing the third task!
[01:21:22] Finished 'task3' after 2.5 ms
[01:21:22] Finished 'default' after 5.59 ms

D:\Users\User\Иститут-задания\web-программирование\Lab 3>
```

Рисунок 4 – Результат параллельного выполнения задач

Для следующего пункта был установлен инструмент автоматизации browsersync, который позволяет отслеживать изменения в файлах и перезагружать автоматически страницу в реальном времени. Установлен он был с помощью команды `npm install gulp browser-sync`.

При выполнении команды `gulp` в консоли выдается подобная информация по загрузке информации. Для перезагрузки браузера после внесения изменений в файл `.html` или `.css` была использована функция `browserSync.reload()`. Результат кода представлен на рисунке 5.

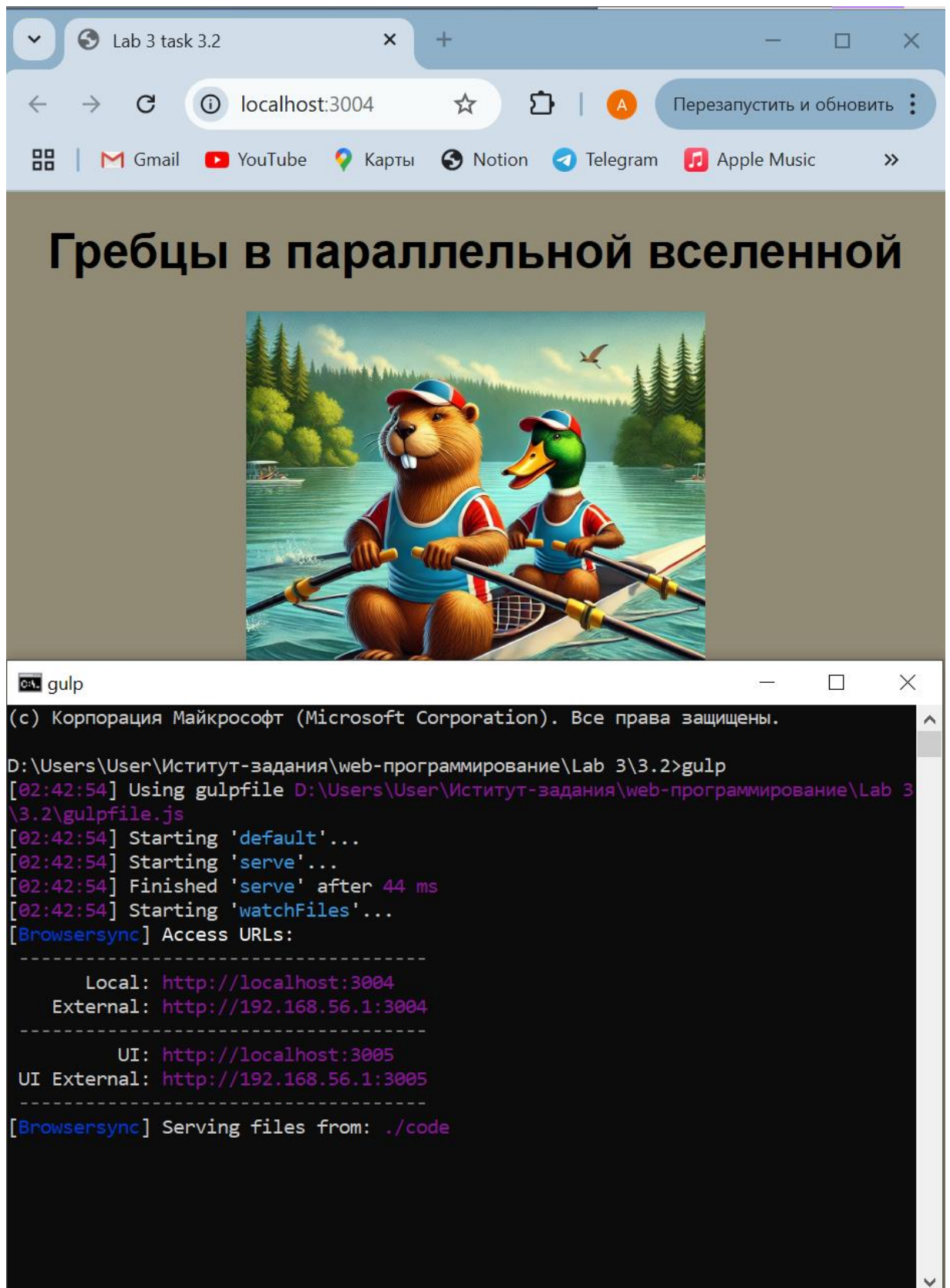


Рисунок 5 – Запуск gulpfile

Были внесены изменения в файл `style_for_file1.css`. На рисунке 6 представлено отображение наличия внесенных изменений и автоматическая перезагрузка страницы.

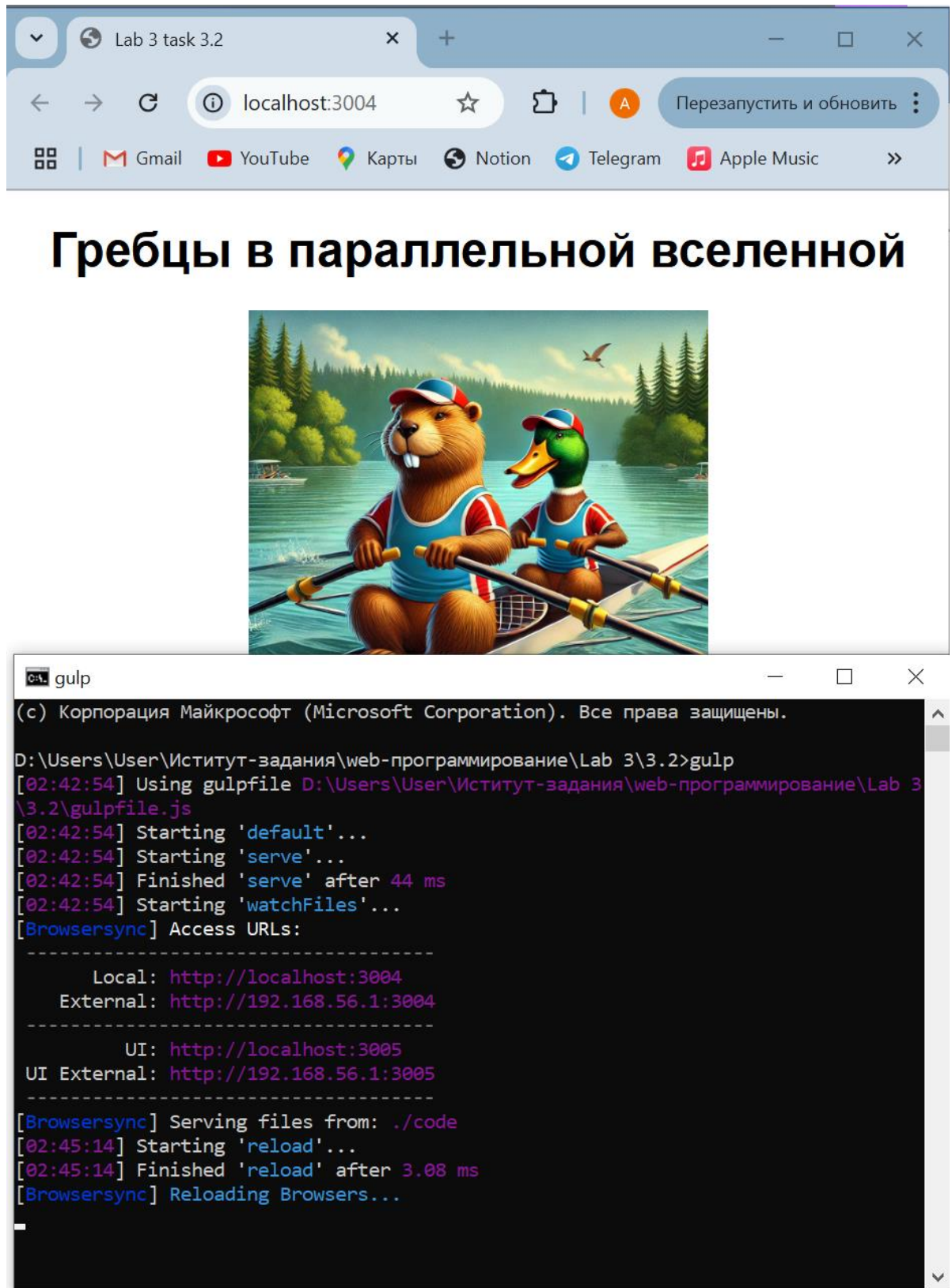
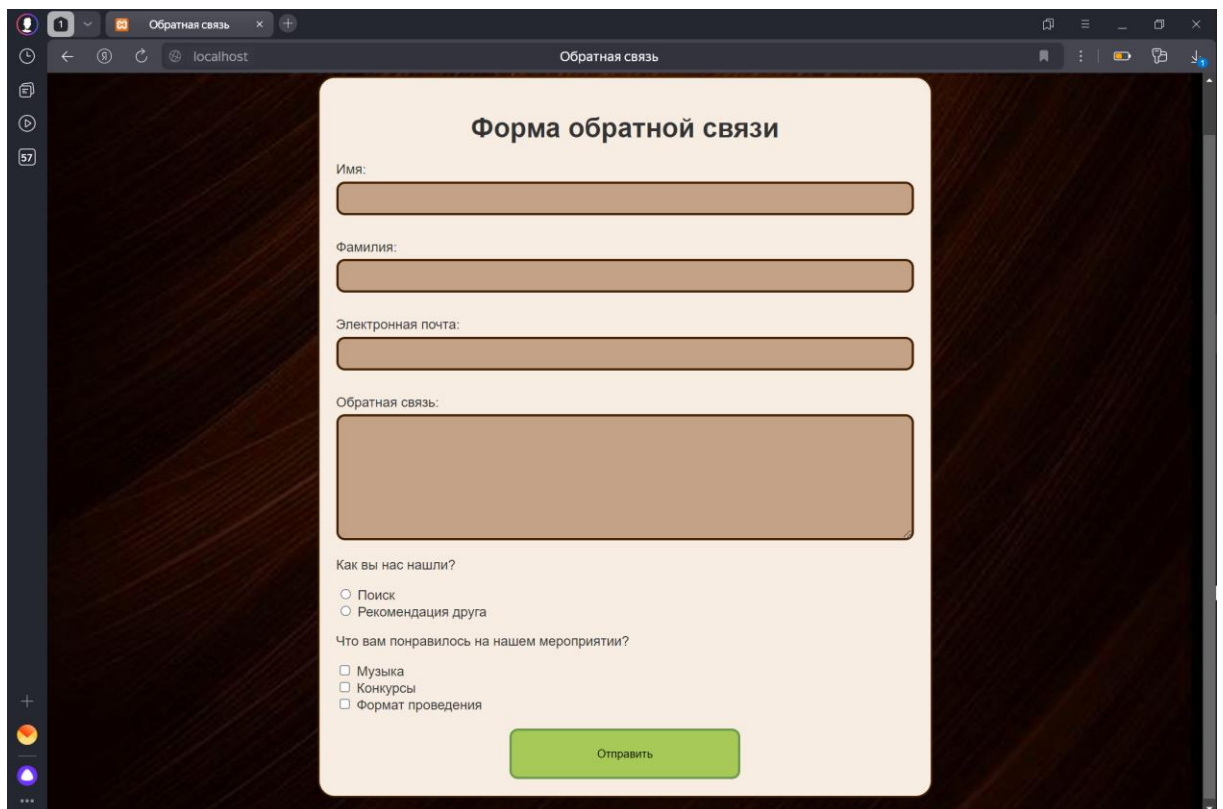


Рисунок 6 – Отображение внесенных изменений

2. Задание 2 – Форма для отправки информации

В этом задании необходимо создать форму для отправки информации по обратной связи от пользователя сайта – передает информацию о себе: имя, фамилия, электронная почта, поле с обратной связью, должны быть радиокнопки (по меньшей мере 2 шт.) и должны быть чекбоксы (не менее трех).

Реализованная страница с формой обратной связи представлена на рисунке 7.



The screenshot shows a web browser window with the title 'Обратная связь' and the address bar displaying 'localhost'. The main content is a feedback form titled 'Форма обратной связи'. The form contains the following fields and controls:

- Имя: (text input)
- Фамилия: (text input)
- Электронная почта: (text input)
- Обратная связь: (large text area)
- Как вы нас нашли? (radio buttons):
 - ☐ Поиск
 - ☐ Рекомендация друга
- Что вам понравилось на нашем мероприятии? (checkboxes):
 - ☐ Музыка
 - ☐ Конкурсы
 - ☐ Формат проведения
- Отправить (green button)

Рисунок 7 – web-страница с формой обратной связи

На рисунке 8 представлена часть кода из файла `feedback_form.html`, описывающую форму.

```

<h1>Форма обратной связи</h1>
<form action="feedback_handler.php" method="POST">

    <label for="first_name" id="label_name">Имя:</label><br>
    <input type="text" id="first_name" name="first_name" required><br><br>

    <label for="last_name" id="label_name">Фамилия:</label><br>
    <input type="text" id="last_name" name="last_name" required><br><br>

    <label for="email" id="label_name">Электронная почта:</label><br>
    <input type="email" id="email" name="email" required><br><br>

    <label for="feedback">Обратная связь:</label><br>
    <textarea id="feedback" name="feedback" rows="4" required></textarea><br>
    <p>Как вы нас нашли?</p>
    <div class="find_us">
        <div>
            <input type="radio" id="search" name="source" value="search" required>
            <label for="search">Поиск</label><br>
        </div>
        <div>
            <input type="radio" id="friend" name="source" value="friend" required>
            <label for="friend">Рекомендация друга</label><br>
        </div>
    </div>

    <p>Что вам понравилось на нашем мероприятии?</p>
    <div class="like">
        <div>
            <input type="checkbox" id="news" name="interests[]" value="music">
            <label for="music">Музыка</label><br>
        </div>
        <div>
            <input type="checkbox" id="updates" name="interests[]" value="competitions">
            <label for="competitions">Конкурсы</label><br>
        </div>
        <div>
            <input type="checkbox" id="offers" name="interests[]" value="format">
            <label for="format">Формат проведения</label><br><br>
        </div>
    </div>

    <button type="submit" id="b_send">Отправить</button>

```

Рисунок 8 – Код, описывающий форму

Для реализации работы с локальным сервером и базами данных был установлен XAMPP. XAMPP — бесплатный кроссплатформенный дистрибутив для сборки локального веб-сервера. Содержит Apache, MariaDB, язык программирования Perl, интерпретатор скриптов PHP и дополнительные библиотеки. Разработчики могут использовать сборку для тестирования работы веб-страниц и сайтов до внесения изменения в основной проект.

Был написан PHP скрипт, который позволяет переносить внесенные данные в форму в таблицу базы данных MySQL Server. После настройки и

синхронизации MySQL Server и XAMPP на странице в phpMyAdmin была создана БД «new_data_07_12» с таблицей «feedbacks», структура которой представлена на рисунке 9.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
<input type="checkbox"/> 1	id	int(11)			Нет	<i>Нет</i>		AUTO_INCREMENT
<input type="checkbox"/> 2	first_name	varchar(255)	utf8_general_ci		Нет	<i>Нет</i>		
<input type="checkbox"/> 3	last_name	varchar(255)	utf8_general_ci		Нет	<i>Нет</i>		
<input type="checkbox"/> 4	email	varchar(255)	utf8_general_ci		Нет	<i>Нет</i>		
<input type="checkbox"/> 5	feedback	text	utf8_general_ci		Да	<i>NULL</i>		
<input type="checkbox"/> 6	source	varchar(255)	utf8_general_ci		Да	<i>NULL</i>		
<input type="checkbox"/> 7	interests	text	utf8_general_ci		Да	<i>NULL</i>		
<input type="checkbox"/> 8	created_at	timestamp			Нет	<i>current_timestamp()</i>		

Рисунок 9 – Структура таблицы в БД

Методы GET и POST используются для передачи данных между клиентом и сервером в веб-приложениях. Метод GET передает данные через URL, добавляя их в строку запроса после символа ?, что ограничивает объем передаваемой информации и делает данные видимыми в адресной строке. Он подходит для выполнения запросов, которые не изменяют состояние сервера, например, при поиске или фильтрации. В PHP данные GET-запроса доступны через массив \$_GET, что позволяет легко их обрабатывать, проверяя наличие параметров и выводя результаты. Метод POST передает данные в теле HTTP-запроса, что обеспечивает большую безопасность и отсутствие ограничений на объем передаваемой информации. Он используется для операций, связанных с изменением данных на сервере, например, отправки форм регистрации или обновления данных. В PHP данные POST-запроса доступны через массив \$_POST. GET чаще применяется для запросов, требующих кэширования или видимости данных в URL, тогда как POST предпочтителен для передачи больших объемов или конфиденциальной информации. Для обработки запросов на сервере используется проверка метода через \$_SERVER['REQUEST_METHOD'], что позволяет создавать скрипты, поддерживающие оба метода.

На рисунке 10 представлена часть кода php скрипта, на котором представлено получение данных из формы.

```
if ($_SERVER["REQUEST_METHOD"] === "POST") {
    // Подключение к базе данных MySQL
    $servername = "localhost:3307";
    $username = "root";
    $password = "";
    $dbname = "new_data_07_12";

    // Создаем соединение
    $conn = new mysqli($servername, $username, $password, $dbname);

    // Проверка соединения
    if ($conn->connect_error) {
        die("Ошибка подключения: " . $conn->connect_error);
    }

    // Получение данных из формы
    $firstName = htmlspecialchars($_POST['first_name']);
    $lastName = htmlspecialchars($_POST['last_name']);
    $email = htmlspecialchars($_POST['email']);
    $feedback = htmlspecialchars($_POST['feedback']);
    $source = htmlspecialchars($_POST['source']);
    $interests = isset($_POST['interests']) ? implode(", ", $_POST['interests']) : '';

    // Проверяем наличие таблицы и создаем её, если необходимо
}
```

Рисунок 10 – Получение данных из формы

После отправки формы на сайте phpMyAdmin можем посмотреть все отправленные формы (рисунок 11).

	id	first_name	last_name	email	feedback	source	interests	created_at
ить	1	Анастасия	Бакланова	nastya.baklanova004@gmail.com	hi	search	music, format	2024-12-07 17:22:52
ить	2	Анастасия	Бакланова	nastya.baklanova004@gmail.com	Мне очень понравилось мероприятие!	friend	music, competitions, format	2024-12-07 17:29:58

Рисунок 11 – Все данные БД

3. Задание 3 – Инструментарий для отладки проектов

В третьем задании необходимо установить инструментарий для отладки проектов, а также с портала wordpress установить движок и настроить портал <http://test.site>, при вводе которого будет отвечать наш портал. Инструментарий для отладки проектов (XAMPP) был установлен на этапе выполнения второго задания.

В файл etc/hosts была добавлена строка для настройки доменного имени (рисунок 12).

```
0.0.0.1 mssplus.mcafee.com
127.0.0.1      test.siteS
```

Рисунок 12 – настройка доменного имени

Далее был настроен веб-сервер, внося в файл httpd-vhosts.conf данные виртуального хоста для test.site (Рисунок 13)

```
<VirtualHost *:80>
    ServerName test.site
    DocumentRoot "D:\xampp\htdocs\wordpress"
    <Directory "D:\xampp\htdocs\wordpress">
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

Рисунок 13 – Настройка веб-сервера

Также настроен WordPress (Рисунок 14).

Адрес WordPress (URL)
<input type="text" value="http://test.site"/>
Адрес сайта (URL)
<input type="text" value="http://test.site"/>

Рисунок 14 – Настройка WordPress

Теперь при вводе <http://test/site> отображается страница тестового сайта (Рисунок 15).

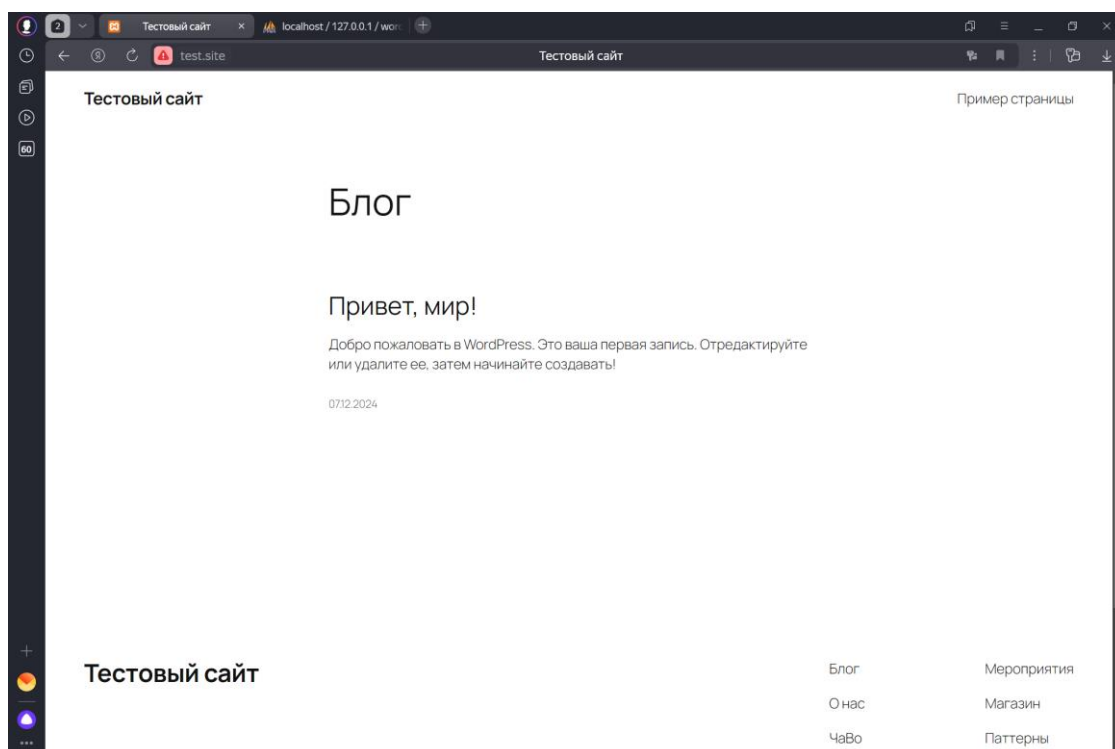


Рисунок 15 – Просмотр страницы по новому адресу

Заключение

В третьей практической работе был настроен gulp, создана web-сайт, на котором отображается форма для отправки информации об обратной связи, написан php скрипт, который отправляет полученные данные из формы в базу данных и настроен портал в Wordpress.