

**Министерство науки и высшего образования Российской Федерации**  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**  
**«Национальный исследовательский университет ИТМО»**  
**(Университет ИТМО)**

**Факультет инфокоммуникационных технологий**  
**Образовательная программа «Программирование в**  
**инфокоммуникационных системах»**

**ЛАБОРАТОРНАЯ РАБОТА 3**  
**Работа с gulp и wordpress**

Обучающийся: Куцак А.М.  
Преподаватель: Марченко Е. В.

Санкт-Петербург  
2024

## Содержание

<b>Содержание .....</b>	<b>2</b>
<b>Введение.....</b>	<b>3</b>
<b>Ход работы .....</b>	<b>3</b>
<b>Выбор репозитория.....</b>	<b>3</b>
<b>Анализ функциональности .....</b>	<b>3</b>
<b>Написание тестов.....</b>	<b>4</b>
<b>Заключение .....</b>	<b>6</b>

## Введение

Цель работы заключается в том, чтобы изучить основы работы с Gulp и Wordpress.

## Ход работы

### Задание 1.

В первом задании было необходимо настроить gulp – создать два таска и настроить последовательное и параллельное выполнение, а также настроить автоматическую перезагрузку страницы при внесении изменений.

Для начала мы убедились, что node.js и gulp установлены, далее создали проект и установили gulp в проект - `npm install --save-dev gulp`. Далее в корне проекта создали файл – `gulpfile.js`, где был создан простой таск, а также написаны функции для последовательного и параллельного выполнения - Рисунок 1.



```
gulpfile.js

const gulp = require('gulp');

function task1(done) {
  console.log('Task 1 выполнен!');
  done();
}

function task2(done) {
  console.log('Task 2 выполнен!');
  done();
}

// Последовательное выполнение
const sequence = gulp.series(task1, task2);

// Параллельное выполнение
const parallel = gulp.parallel(task1, task2);

exports.sequence = sequence;
exports.parallel = parallel;
```

Рисунок 1 – Таски для параллельного и последовательного выполнения.

С помощью команды - `gulp sequence` было запущено последовательное выполнение, результат работы команды показан на Рисунке 2.

```
anastasiakucak@MacBook-Air-Anastasia WEb % gulp sequence

[22:59:20] Using gulpfile ~/WEb/gulpfile.js
[22:59:20] Starting 'sequence'...
[22:59:20] Starting 'task1'...
Task 1 выполнен!
[22:59:20] Finished 'task1' after 780 µs
[22:59:20] Starting 'task2'...
Task 2 выполнен!
[22:59:20] Finished 'task2' after 500 µs
[22:59:20] Finished 'sequence' after 3.27 ms
```

Рисунок 2 – Последовательное выполнение.

С помощью команды - `gulp parallel` было запущено параллельное выполнение, результат работы команды показан на Рисунке 3.

```
anastasiakucak@MacBook-Air-Anastasia WEb % gulp parallel

[22:59:35] Using gulpfile ~/WEb/gulpfile.js
[22:59:35] Starting 'parallel'...
[22:59:35] Starting 'task1'...
[22:59:35] Starting 'task2'...
Task 1 выполнен!
[22:59:35] Finished 'task1' after 940 µs
Task 2 выполнен!
[22:59:35] Finished 'task2' after 1.03 ms
[22:59:35] Finished 'parallel' after 2.25 ms
```

Рисунок 3 – Параллельное выполнение.

Перейдем ко второй части задания, где надо было создать простую HTML страницу и настроить автоматическую перезагрузку. Установили пакет `browser-sync`: **`npm install --save-dev browser-sync`**. И создали другой `gulpfile.js` – Рисунок 4.

```

const gulp = require('gulp');
const browserSync = require('browser-sync').create();

function serve(done) {
  browserSync.init({
    server: {
      baseDir: '/Users/anastasiakucak/my-project'
    }
  });
  done();
}

function reload(done) {
  browserSync.reload();
  done();
}

function watchFiles() {
  gulp.watch('./*.html', reload);
  gulp.watch('./css/*.css', reload);
  gulp.watch('./js/*.js', reload);
}

const dev = gulp.series(serve, watchFiles);
exports.default = dev;

```

Рисунок 4 – gulpfile.js

Далее был создан простой HTML документ – Рисунок 5.

```

index.html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Форма обратной связи</title>
</head>
<body>
  <label for="name">Всем привет, это конец первого задания!</label>
</body>
</html>

```

Рисунок 5 – HTML -страница

С помощью команды `gulp` запустили файлы, и теперь при сохранении изменений в файле `index.html` страница в браузере автоматически перезагружается и обновляет данные. В консоли видим все изменения – Рисунок 6.

```
anastasiakucak@MacBook-Air-Anastasia my-project % gulp

[19:29:34] Using gulpfile ~/my-project/gulpfile.js
[19:29:34] Starting 'default'...
[19:29:34] Starting 'serve'...
[19:29:34] Finished 'serve' after 10 ms
[19:29:34] Starting 'watchFiles'...
[Browsersync] Access URLs:
  -----
    Local: http://localhost:3000
  External: http://192.168.31.73:3000
  -----
    UI: http://localhost:3001
  UI External: http://192.168.31.73:3001
  -----
[Browsersync] Serving files from: /Users/anastasiakucak/my-project
[19:29:41] Starting 'reload'...
[19:29:41] Finished 'reload' after 2.79 ms
[Browsersync] Reloading Browsers...
[19:29:52] Starting 'reload'...
[19:29:52] Finished 'reload' after 2.25 ms
[Browsersync] Reloading Browsers...
```

Рисунок 6 – Перезагрузка страницы.

## Задание 2.

Во втором задании надо было создать форму для отправки обратной связи, необходимо создать файл и формой и php файл для обработки полученных данных.

Для начала установили локальный сервер через Xampp и запустили Apache и MySQL.

Создали html файл – Рисунок 7.

```
feedback.html
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     <title>Feedback Form</title>
7 </head>
8 <body>
9     <h1>Обратная связь</h1>
10    <form action="feedback.php" method="post">
11        <label for="firstName">Имя:</label>
12        <input type="text" id="firstName" name="firstName" required><br><br>
13
14        <label for="lastName">Фамилия:</label>
15        <input type="text" id="lastName" name="lastName" required><br><br>
16
17        <label for="email">Электронная почта:</label>
18        <input type="email" id="email" name="email" required><br><br>
19
20        <label for="feedback">Ваш отзыв:</label><br>
21        <textarea id="feedback" name="feedback" rows="4" cols="50" required></
22        textarea><br><br>
23
24        <p>Выберите одну из опций:</p>
25        <label><input type="radio" name="option" value="K3320" required>Группа
26        K3320</label><br>
27        <label><input type="radio" name="option" value="K3321" required>Группа
28        K3321</label><br>
29        <label><input type="radio" name="option" value="K3322" required>Группа
30        K3322</label><br>
31        <label><input type="radio" name="option" value="K3323" required>Группа
32        K3323</label><br><br>
33
34        <p>Выберите опции (можно несколько):</p>
35        <label><input type="checkbox" name="checkbox1" value="Комфортная
36        атмосфера на занятии">Комфортная атмосфера на занятии</label><br>
37        <label><input type="checkbox" name="checkbox2" value="Четкие дедлайны">
38        Четкие дедлайны</label><br>
39        <label><input type="checkbox" name="checkbox3" value="Вовремя выставлены
40        баллы">Вовремя выставлены баллы</label><br><br>
41
42        <button type="submit">Отправить</button>
43    </form>
44 </body>
45 </html>
```

Рисунок 7 – html страница для отправки фидбека.

Php страница – для обработки данных с html страницы – Рисунок 8

```

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    echo "<h1>Данные, полученные через POST:</h1>";

    echo "Имя: " . htmlspecialchars($_POST['firstName']) . "<br>";
    echo "Фамилия: " . htmlspecialchars($_POST['lastName']) . "<br>";
    echo "Электронная почта: " . htmlspecialchars($_POST['email']) . "<br>";
    echo "Ваш отзыв: " . htmlspecialchars($_POST['feedback']) . "<br>";

    echo "Выбранная опция: " . htmlspecialchars($_POST['option']) . "<br>";

    echo "Выбранные чекбоксы:<br>";
    if (isset($_POST['checkbox1'])) echo "- " . htmlspecialchars($_POST['checkbox1']) . "<br>";
    if (isset($_POST['checkbox2'])) echo "- " . htmlspecialchars($_POST['checkbox2']) . "<br>";
    if (isset($_POST['checkbox3'])) echo "- " . htmlspecialchars($_POST['checkbox3']) . "<br>";
} else {
    echo "Метод GET не поддерживается для этой формы.";
}
?>

```

Рисунок 8 – Php страница.

По ссылке - <http://localhost/task3/feedback.html> был запущен файл, после введения данных Php скрипт их обрабатывает и выводит сообщение об удачной отправке данных. Ниже на рисунках 9 и 10 представлена работа программы.



# Обратная связь

Имя: Анастасия

Фамилия: Куцак

Электронная почта: amkutsak@gmail.com

Ваш отзыв:

Все круто!

Выберите одну из опций:

- ☐ Группа K3320
- ☐ Группа K3321
- ☐ Группа K3322
- ☒ Группа K3323

Выберите опции (можно несколько):

- ☒ Комфортная атмосфера на занятии
- ☒ Четкие дедлайны
- ☐ Вовремя выставлены баллы

Отправить

Рисунок 9 – Форма обратной связи.

# Данные, полученные через POST:

Имя: Анастасия  
Фамилия: Куцак  
Электронная почта: amkutsak@gmail.com  
Ваш отзыв: Все круто!  
Выбранная опция: K3323  
Выбранные чекбоксы:  
- Комфортная атмосфера на занятии  
- Четкие дедлайны

Рисунок 10 – Вывод данных после отправки.

В данной работе был использован метод Post, так как он используется для безопасной передачи данных, данные не видны в адресной строке. В форме используются личные данные пользователей, поэтому метод Get, который передает данные в адресной строке, не подходит для нашей задачи исходя из безопасности.

## Задание 3.

В последнем задании надо было настроить портал <http://test.site> При переходе по этой ссылке должен открываться наш портал.

Для этого задания уже был ранее скачан Xampp, к нему в папку htdocs дополнительно был установлен движок wordpress. Далее происходила настройка wordpress – Рисунок 11.



Введите здесь информацию о подключении к базе данных. Если вы в ней не уверены, свяжитесь с поддержкой вашего хостинга.

Имя базы данных

Имя базы данных, в которую вы хотите установить WordPress.

Имя пользователя

Имя пользователя базы данных.

Пароль

[Показать](#)

Пароль пользователя базы данных.

Сервер базы данных

Если localhost не работает, нужно узнать правильный адрес в службе поддержки хостинг-провайдера.

Префикс таблиц

Если вы хотите запустить несколько копий WordPress в одной базе, измените это значение.

Рисунок 11 – Настройка базы данных.

Далле настроили домен в файл – Рисунок 12.

```
anastasiakucak@Air-Anastasia ~ % sudo nano /etc/hosts
```

[Password:

UW PICO 5.09

File: /etc/hosts

```
##
# Host Database
#
# localhost is used to configure the loopback interface
# when the system is booting. Do not change this entry.
##
127.0.0.1        localhost
255.255.255.255 broadcasthost
::1             localhost
127.0.0.1 test.site
```

## Рисунок 12 – Настройка домена.

В файле httpd-vhosts.conf настроили виртуальный хост – Рисунок 13.

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot "/Applications/XAMPP/xamppfiles/htdocs"
    ServerName localhost
    ErrorLog "logs/localhost-error.log"
    CustomLog "logs/localhost-access.log" common
    <Directory "/Applications/XAMPP/xamppfiles/htdocs">
        Options Indexes FollowSymLinks Includes ExecCGI
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>

<VirtualHost *:80>
    ServerAdmin webmaster@test.site
    DocumentRoot "/Applications/XAMPP/xamppfiles/htdocs/wordpress"
    ServerName test.site
    ErrorLog "logs/test-site-error.log"
    CustomLog "logs/test-site-access.log" common
</VirtualHost>
```

## Рисунок 13 – Настройка виртуального хоста.

Далее надо было включить виртуальные хосты в конфигурационном файле Apache – Рисунок 14

```
# Virtual hosts
Include etc/extra/httpd-vhosts.conf
```

## Рисунок 14 – Включение виртуальных хостов

После этого перезапустили сервер Apache и все заработало, теперь можно открыть wordpress сайт через test.site – Рисунок 15.

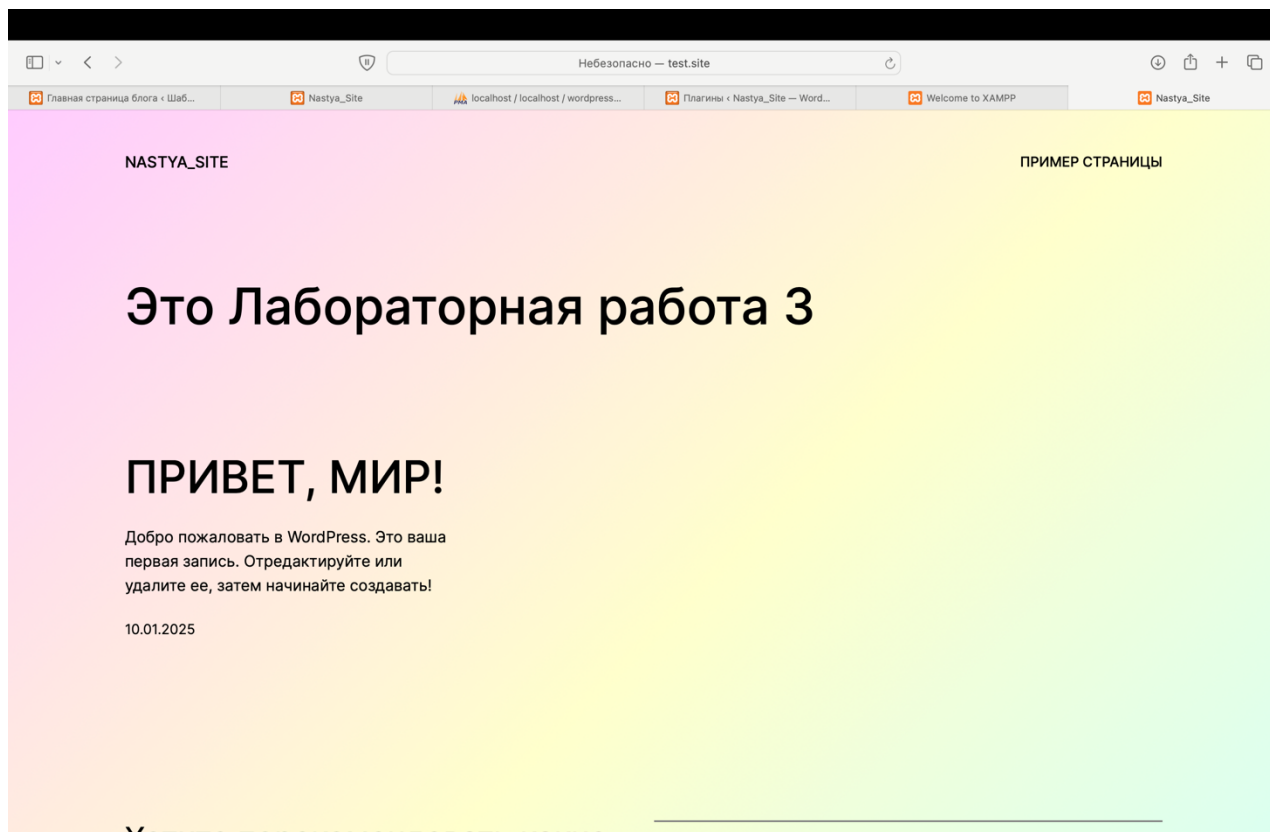


Рисунок 15 – Работа сайта через test.site.

**Вывод:** была проведена работа с gulp, базами данных phpMyAdmin и с движком wordpress.