

Министерство образования и науки  
федеральное государственное автономное образовательное учреждение  
высшего образования

«Национальный исследовательский университет ИТМО»

Факультет инфокоммуникационных технологий

Отчет по дисциплине: «Web-программирование»

### **Практическая работа 3**

Выполнила: Полтавец  
Елена Андреевна

Группа: К3322

Проверила: Марченко  
Елена Вадимовна

Санкт-Петербург

2024

**Цель:** применив навыки разработки веб-страниц, настроить gulp и создать форму для отправки информации.

**Задачи:**

- Настроить gulp и создать задачи согласно заданию;
- Создать форму для отправки информации по обратной связи от пользователя сайта;
- Настроить портал в Wordpress.

### **Ход работы**

#### **Задание 1**

В данном задании было необходимо настроить gulp: а) создать две задачи – настроить на последовательное и параллельное выполнение; б) настроить отображение файлов проекта в браузере и автоматическую перезагрузку при изменении одного из контролируемых файлов проекта.

Документация: <https://webdesign-master.ru/blog/tools/gulp-beginners.html>

Установка включала в себя следующие этапы (подробнее информация есть в отчете по практической работе 2):

1. Для работы с Gulp был установлен Node.js: `brew install node`;
2. После был установлен Gulp (Рисунок 1): `npm i gulp -g` (запускается менеджер пакетов npm (Node Package Manager), который командой `install` устанавливает Gulp в систему. Ключ `-g` говорит о том, что пакет установится в систему глобально, то-есть в систему, а не в папку проекта. Без ключа `-g` пакет устанавливаются в ту папку, в которой выполняются текущие команды).

```
[drachioaelena@MacBook-Air-Elena-2 ~ % npm i gulp -g
added 143 packages in 17s

14 packages are looking for funding
  run `npm fund` for details
npm notice
npm notice New patch version of npm available! 10.8.2 -> 10.8.3
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.8.3
npm notice To update run: npm install -g npm@10.8.3
npm notice
```

Рисунок 1 – Установка Gulp

Далее были написаны таски для последовательного и параллельного выполнения задач. Последовательное выполнения реализуется функцией `series()`. Смысл этой функции в том, что пока не завершилась работа одной функции не начнётся выполнение другой (Рисунок 2).

```
[drachioaelena@MacBook-Air-Elena-2 task 3.1 % gulp
[09:04:57] Using gulpfile ~/Desktop/итмо/5 сем/БЕБ/task 3.1/gulpfile.js
[09:04:57] Starting 'default'...
[09:04:57] Starting 'task1'...
Выполнение задачи 1
[09:04:57] Finished 'task1' after 560 μs
[09:04:57] Starting 'task2'...
Выполнение задачи 2
[09:04:57] Finished 'task2' after 381 μs
[09:04:57] Finished 'default' after 2.42 ms
```

Рисунок 2 – Последовательное выполнение

Для параллельного выполнения была использована функция `parallel()`, которая выполняет функции параллельно/одновременно (Рисунок 3). В примере функции не стоят в какой-то очереди или ожидают окончания друг друга, а начинают работать вместе.

```
[drachioaelena@MacBook-Air-Elena-2 task 3.1 % gulp
[09:05:39] Using gulpfile ~/Desktop/итмо/5 сем/БЕБ/task 3.1/gulpfile.js
[09:05:39] Starting 'default'...
[09:05:39] Starting 'task1'...
[09:05:39] Starting 'task2'...
Выполнение задачи 1
[09:05:39] Finished 'task1' after 670 μs
Выполнение задачи 2
[09:05:39] Finished 'task2' after 1.11 ms
[09:05:39] Finished 'default' after 2.33 ms
```

Рисунок 3 – Параллельное выполнение

Далее был написан task для автоматического обновления страниц с использованием Bbrowser Sync.

Browser Sync – это отличное решение для LiveReload страниц при сохранении файлов. При чем релоад происходит не только в одном браузере, но и во всех браузерах сети, будь это мобильные устройства или другие компьютеры в одной Wi-Fi сети.

При выполнении в терминале команды `gulp` в консоли появляется подробная информация, а также добавляется файл `css`, стили которого применяются на страницу `html` (Рисунок 4).

```
[drachiovaelena@MacBook-Air-Elena-2 task 3.1 % gulp
[08:45:20] Using gulpfile ~/Desktop/итмо/5 сем/БЕБ/task 3.1/gulpfile.js
[08:45:20] Starting 'default'...
[08:45:20] Starting 'sass'...
[08:45:20] Starting 'browser-sync'...
[08:45:20] Starting 'watch'...
[08:45:20] Finished 'sass' after 32 ms
[08:45:20] Finished 'browser-sync' after 33 ms
[08:45:20] Finished 'watch' after 32 ms
[08:45:20] Finished 'default' after 34 ms
Deprecation Warning: The legacy JS API is deprecated and will be removed in Dart Sass 2.0.0.

[Browsersync] Access URLs:
-----
    Local: http://localhost:3000
  External: http://192.168.0.103:3000
-----
    UI: http://localhost:3001
  UI External: http://192.168.0.103:3001
-----
[Browsersync] Serving files from: app
```

Рисунок 4 – Запуск командой

Для того, чтобы страница обновлялась при сохранении `html`, был добавлен дополнительный код, здесь была использована функция `browserSync.reload`, которую предоставил пакет `Browser Sync`. При выполнении страница сама открылась с установленными настройками (Рисунок 5).

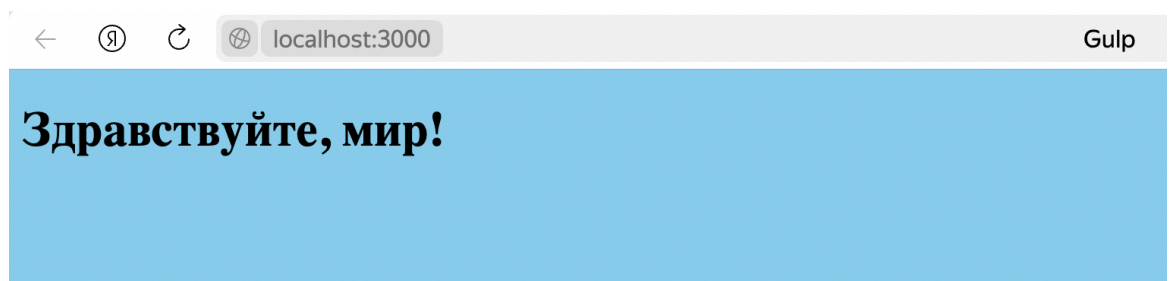
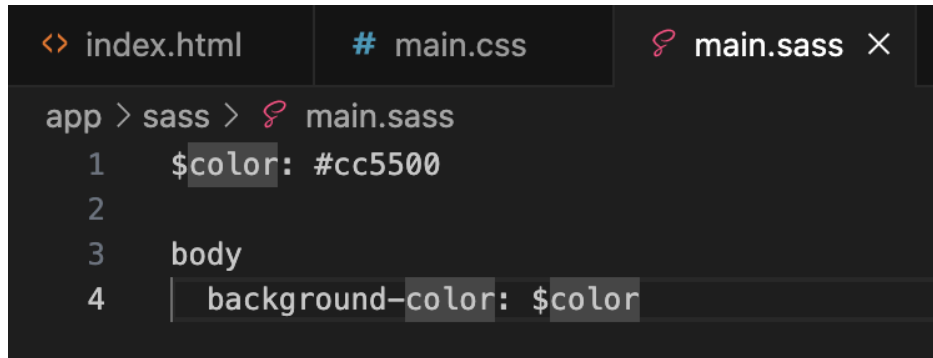


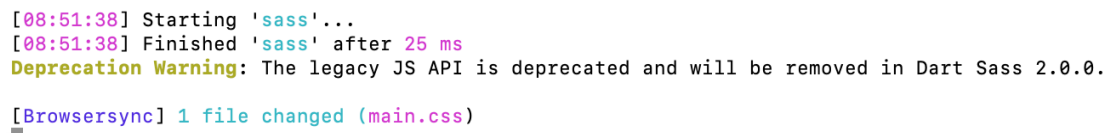
Рисунок 5 – Сайт при запуске `gulp`

Далее в файле были произведены изменения (Рисунок 6). В консоли и на сайте произошли изменения (Рисунки 7 - 8).



```
<> index.html # main.css main.sass X
app > sass > main.sass
1  $color: #cc5500
2
3  body
4  background-color: $color
```

Рисунок 6 – Внесение изменений



```
[08:51:38] Starting 'sass'...
[08:51:38] Finished 'sass' after 25 ms
Deprecation Warning: The legacy JS API is deprecated and will be removed in Dart Sass 2.0.0.
[Browsersync] 1 file changed (main.css)
```

Рисунок 7 – Вывод в консоли



Рисунок 8 – Изменение цвета сайта после сохранения файла

## Задание 2

В данном задании была создана форма для отправки информации по обратной связи от пользователя сайта (Рисунок 9) – 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). Разработан файл с формой и php скрипт по образцу (Рисунок 10).

**Форма обратной связи**

ФИО:

Телефон:

Email:

Выберите оказанную услугу:

☐ Услуга 1

☐ Услуга 2

Отметьте, если описание соответствует оказанной услуге:

☐ Описание 1

☐ Описание 2

☐ Описание 3

Комментарий:

Отправить форму

Рисунок 9 – Форма обратной связи

```

<input type="text" id="lastname" name="lastname" required><br>

<label for="phone">Телефон:</label><br>
<input type="tel" id="phone" name="phone" required><br>

<label for="email">Email:</label><br>
<input type="email" id="email" name="email" required><br>

<label>Выберите оказанную услугу:</label><br>
<input type="radio" id="category1" name="category" value="category1">
<label for="category1">Услуга 1</label><br>
<input type="radio" id="category2" name="category" value="category2">
<label for="category2">Услуга 2</label><br>

<label>Отметьте, если описание соответствует оказанной услуге:</label><br>
<input type="checkbox" id="descr1" name="descr1" value="descr1">
<label for="descr1">Описание 1</label><br>
<input type="checkbox" id="descr2" name="descr2" value="descr2">
<label for="descr2">Описание 2</label><br>
<input type="checkbox" id="descr3" name="descr3" value="descr3">
<label for="descr3">Описание 3</label><br>

<label for="comments">Комментарий:</label><br>
<textarea id="comments" name="comments"></textarea><br>

<button type="submit">Отправить форму</button>
</form>
<?php

```

Рисунок 10 – Часть кода, описывающая форму

Был установлен MAMP (Macintosh, Apache, MySQL, PHP) – это программное обеспечение, которое создано для установки серверной среды на компьютерах Mac, позволяющее локально разрабатывать и тестировать веб-приложения, используя комбинацию Apache в качестве веб-сервера, MySQL в

качестве системы управления базами данных и PHP в качестве языка программирования.

Данные, введенные в форму, обрабатываются PHP скриптом и заносятся в таблицу БД MySQL Server. Разработана и создана структура таблицы в PhpMyAdmin (Рисунок 11).


	#	Имя	Тип	Сравнение	Атрибут
<input type="checkbox"/>	1	<b>ID</b> 	int		
<input type="checkbox"/>	2	<b>fullname</b>	varchar(100)	utf8mb4_0900_ai_ci	
<input type="checkbox"/>	3	<b>phone</b>	varchar(20)	utf8mb4_0900_ai_ci	
<input type="checkbox"/>	4	<b>email</b>	varchar(100)	utf8mb4_0900_ai_ci	
<input type="checkbox"/>	5	<b>category</b>	varchar(100)	utf8mb4_0900_ai_ci	
<input type="checkbox"/>	6	<b>description</b>	varchar(100)	utf8mb4_0900_ai_ci	
<input type="checkbox"/>	7	<b>comments</b>	text	utf8mb4_0900_ai_ci	

Рисунок 11 – Структура таблицы для хранения данных из формы

В качестве обработчика POST-запроса при отправке формы используется PHP-скрипт, который добавляет данные из формы (Рисунок 12). Методы POST и GET являются часто используемыми веб-протоколами для отправки запросов на сервер и получения данных. Метод GET используется для получения данных с сервера. Он отправляет параметры запроса в URL строке после вопросительного знака. Этот метод удобен для получения данных, но не рекомендуется для отправки конфиденциальной информации, так как параметры запроса видны в URL. Метод POST, с другой стороны, используется для отправки данных на сервер в теле запроса, что делает его более безопасным для отправки конфиденциальной информации, такой как пароли или персональные данные.

```

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST"){
    $conn = new mysqli("localhost", "main_user", "1234", "lab_web");

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

    $fullname = $_POST["fullname"];
    $phone = $_POST["phone"];
    $email = $_POST["email"];
    $category = $_POST["category"];
    $descr = $_POST["descr"];
    $comments = $_POST["comments"];

    $sql = "INSERT INTO persons (fullname, phone, email, category, description, comments)
    VALUES ('" . $fullname . "', '" . $phone . "', '" . $email . "', '" . $category . "', '" . $descr . "', '" . $comments . "')";

    if($conn->query($sql)){
        header("Location: success.php");
    } else{
        header("Location: error.php");
    }

    $conn->close();
}
?>
</div>

```

Рисунок 12 – Часть кода, добавляющая данные в таблицу баз данных

Если все прошло успешно, то пользователь видит сообщение на новой странице, а данные появляются в базе данных (Рисунок 13).







			ID	fullname	phone	email	category	description	comments
<input type="checkbox"/>	 Изменить	 Копировать	 Удалить	1	Таня Волкова	+79218348521	nayyy@yandex.ru	category1	descr1, descr2
<input type="checkbox"/>	 Изменить	 Копировать	 Удалить	2	Дмитрий Кучевой	+79118312521	dm__y@yandex.ru	category2	descr1, descr2, descr3 Все чудесно!

Рисунок 13 – База данных с обратной связью

### Задание 3

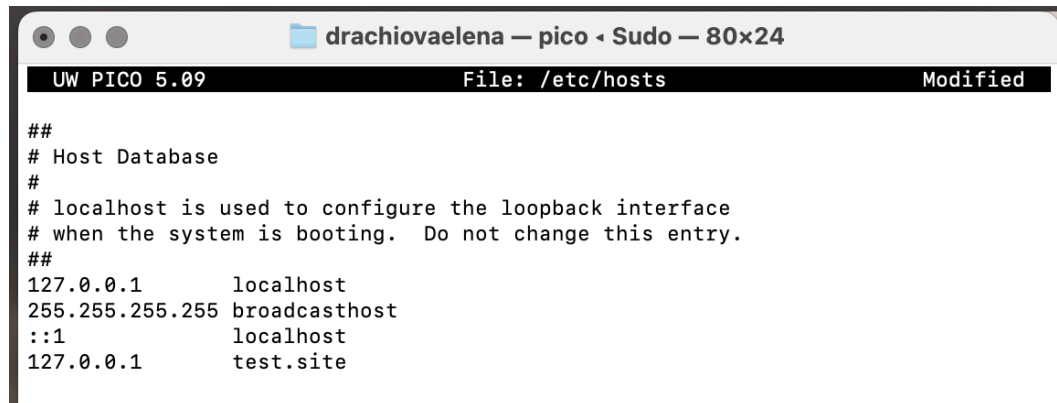
В данном задании необходимо установить инструментальный для отладки проектов, а также движок WordPress. После этого необходимо настроить сервер так, чтобы при запросе по адресу <http://test.site> открывался портал WordPress.

WordPress был скачан с официального сайта, после чего архив был распакован в папку htdocs в MAMP. С помощью PHPMyAdmin была создана база данных wordpress.

В файл `/etc/hosts` была добавлена строка (Рисунок 14). Файл `hosts` — текстовый документ, который содержит в себе информацию о домене и IP-



адресе, который ему соответствует. Данная строка назначает домену test.site IP- адрес 127.0.0.1, то есть адрес localhost.

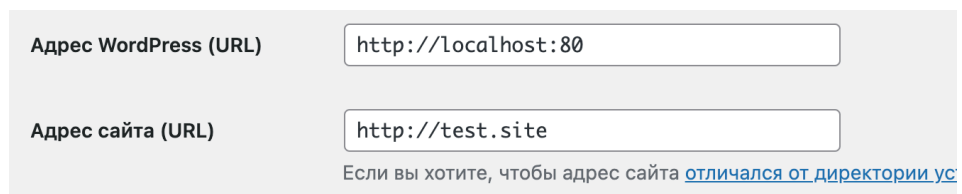


```
drachioaelena — pico — Sudo — 80x24
UW PICO 5.09 File: /etc/hosts Modified

##
# 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 14 – Добавленная строка

Далее было необходимо настроить WordPress (Рисунок 15).



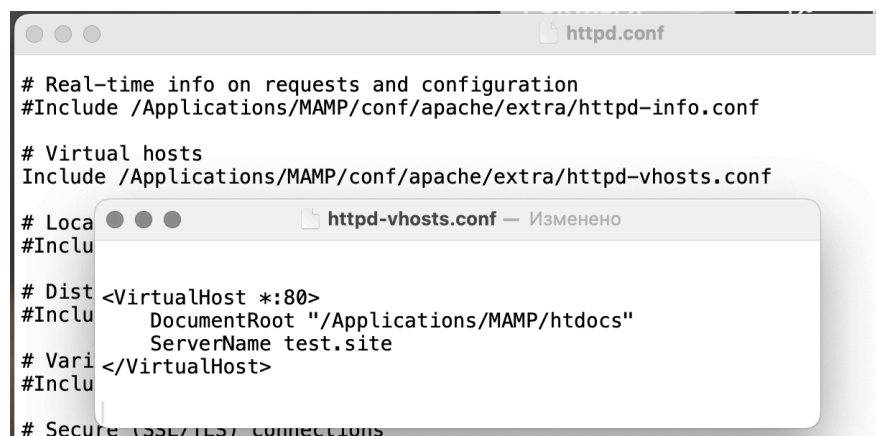
Адрес WordPress (URL)

Адрес сайта (URL)

Если вы хотите, чтобы адрес сайта [отличался от директории УС](#)

Рисунок 15 – Настройки в wordpress

Далее были отредактированы конфигурационные файлы Apache (Рисунок 16).



```
httpd.conf
# Real-time info on requests and configuration
#Include /Applications/MAMP/conf/apache/extra/httpd-info.conf

# Virtual hosts
Include /Applications/MAMP/conf/apache/extra/httpd-vhosts.conf

# Local
#Include

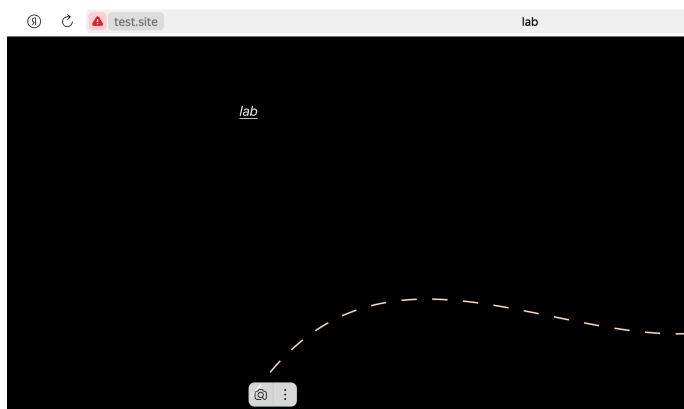
# Distribution
<VirtualHost *:80>
#Include
    DocumentRoot "/Applications/MAMP/htdocs"
    ServerName test.site
# Vari
#Include
</VirtualHost>

# Secure (SSL/TLS) connections
```

```
httpd-vhosts.conf — Изменено
# VirtualHost *:80>
    DocumentRoot "/Applications/MAMP/htdocs"
    ServerName test.site
</VirtualHost>
```

Рисунок 16 – Изменение конфигурационных файлов Apache

При вводе <http://test.site> отображается сайт (Рисунок 16).



Привет, мир!

Добро пожаловать в WordPress. Это ваша первая запись. Отредактируйте  
или удалите ее, затем начинайте создавать!

Рисунок 16 – Отображаемый сайт по нужному адресу

**Вывод:** применив навыки разработки веб-страниц, был настроен gulp и созданы задачи, была создана форма для отправки информации и написан скрипт php, а также был настроен портал.