

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:

Королева А.С.

Группа:

К3322

Проверила:

Марченко Е.В.

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**

Последовательное выполнение (series): Был создан таск, который выполняет последовательные операции. Это удобно, когда задачи зависят друг от друга, и необходимо обеспечить строгое выполнение по порядку. Пример структуры кода:

```
1  const { series } = require('gulp');
2
3  function taskOne(done) {
4    console.log('Task One Completed');
5    done();
6  }
7
8  function taskTwo(done) {
9    console.log('Task Two Completed');
10   done();
11 }
12
13 exports.default = series(taskOne, taskTwo);
```

Рисунок 1 - Последовательное выполнение

```
● nastya@Air-Anastasia-8 lab3 % gulp --gulpfile gulpfile_1.js
[20:26:57] Using gulpfile ~/WebstormProjects/WebDevelopment_2024-2025/works/K3322/Королева_Анастасия/lab3/gulpfile_1.js
[20:26:57] Starting 'default'...
[20:26:57] Starting 'taskOne'...
Task One Completed
[20:26:57] Finished 'taskOne' after 470 μs
[20:26:57] Starting 'taskTwo'...
Task Two Completed
[20:26:57] Finished 'taskTwo' after 581 μs
[20:26:57] Finished 'default' after 3.43 ms
● nastya@Air-Anastasia-8 lab3 %
```

Рисунок 2 - Последовательное выполнение (терминал)

Параллельное выполнение (parallel): Параллельное выполнение задач позволяет ускорить процесс, когда задачи независимы друг от друга. Пример структуры кода:

```
works > K3322 > Королева_Анастасия > lab3 > js gulpfile_2.js > ...
1  const { parallel } = require('gulp');
2
3  function taskA(done) {
4    console.log('Task A Completed');
5    done();
6  }
7
8  function taskB(done) {
9    console.log('Task B Completed');
10   done();
11 }
12
13 exports.default = parallel(taskA, taskB);
14
```

Рисунок 3 - Параллельное выполнение

```
● nastya@Air-Anastasia-8 lab3 % gulp --gulpfile gulpfile_2.js
[20:31:20] Using gulpfile ~/WebstormProjects/WebDevelopment_2024-2025/works/K3322/Королева_Анастасия/lab3/gulpfile_2.js
[20:31:20] Starting 'default'...
[20:31:20] Starting 'taskA'...
[20:31:20] Starting 'taskB'...
Task A Completed
[20:31:20] Finished 'taskA' after 789 μs
Task B Completed
[20:31:20] Finished 'taskB' after 1.26 ms
[20:31:20] Finished 'default' after 3.29 ms
```

Рисунок 4 - Параллельное выполнение (терминал)

В коде Gulp создан таск, который следит за изменениями файлов и обновляет браузер.

```

1  const gulp = require('gulp');
2  const browserSync = require('browser-sync').create();
3
4  function browserSyncTask() {
5    browserSync.init({
6      server: {
7        baseDir: './'
8      }
9    });
10
11    gulp.watch('index.html').on('change', browserSync.reload);
12    gulp.watch('styles.css').on('change', browserSync.reload);
13    gulp.watch('script.js').on('change', browserSync.reload);
14  }
15
16  exports.default = browserSyncTask;
17

```

Рисунок 5 - Перезагрузка при изменении файлов

При изменении HTML, CSS или JS файлов браузер автоматически перезагружался, отображая актуальное состояние проекта.

## Задание 2

Установим PHP на Mac с помощью Homebrew (уже был установлен):  
`brew install php`

Запустим PHP-сервер: `brew services start php`

Инструкция: <https://htmlacademy.ru/blog/php/php-on-macos>

Для запуска PHP-сервера можно воспользоваться расширением PHP Server в Visual Studio Code. Но так как в проекте уже подключен gulp и browser-sync, можно немного исправить файл `gulpfile_3.js` из первого задания.

Скачаем `gulp-connect-php`: `npm install --save-dev gulp-connect-php`

Код переписанного `gulpfile_3.js`:

```

1  const gulp = require('gulp');
2  const connect = require('gulp-connect-php');
3  const browserSync = require('browser-sync').create();
4
5  function browserSyncTask() {
6    connect.server({}, function () {
7      browserSync.init({
8        proxy: '127.0.0.1:8000'
9      });
10   });
11
12   gulp.watch('**/*.php').on('change', browserSync.reload);
13   gulp.watch('styles.css').on('change', browserSync.reload);
14   gulp.watch('script.js').on('change', browserSync.reload);
15   gulp.watch('**/*.html').on('change', browserSync.reload);
16 }
17
18 exports.default = browserSyncTask;
19

```

Рисунок 6 - Изменение файла для работы с php

Создан HTML-документ с формой:

```

5  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6  <title>Page Viewer</title>
7  <link rel="stylesheet" href="styles.css" />
8  </head>
9  <body>
10   <form action="feedback.php" method="POST">
11     <label for="first_name">Имя:</label>
12     <input type="text" id="first_name" name="first_name" required><br><br>
13
14     <label for="last_name">Фамилия:</label>
15     <input type="text" id="last_name" name="last_name" 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">Обратная связь:</label><br>
21     <textarea id="feedback" name="feedback" rows="4" required></textarea><br><br>
22
23     <label>Выберите категорию:</label><br>
24     <input type="radio" id="category1" name="category" value="вопрос" required>
25     <label for="category1">Вопрос</label><br>
26     <input type="radio" id="category2" name="category" value="предложение" required>
27     <label for="category2">Предложение</label><br><br>
28
29     <label>Какие услуги вас интересуют?</label><br>
30     <input type="checkbox" id="service1" name="services[]" value="услуга1">
31     <label for="service1">Услуга 1</label><br>
32     <input type="checkbox" id="service2" name="services[]" value="услуга2">
33     <label for="service2">Услуга 2</label><br>
34     <input type="checkbox" id="service3" name="services[]" value="услуга3">
35     <label for="service3">Услуга 3</label><br><br>
36
37     <input type="submit" value="Отправить">
38   </form>
39 </body>
40 </html>

```

Рисунок 7 - Форма

Создан файл `feedback.php`, который принимает данные, отправленные с формы, и выводит их на экран. Данные обрабатываются с использованием метода POST для повышения безопасности и конфиденциальности.

```
1 <!DOCTYPE html>
2 <html lang="ru">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Обратная связь</title>
7     <link rel="stylesheet" href="styles.css">
8 </head>
9 <body>
10     <h1>Данные обратной связи</h1>
11
12     <?php
13     if ($_SERVER["REQUEST_METHOD"] == "POST") {
14
15         $first_name = htmlspecialchars($_POST['first_name']);
16         $last_name = htmlspecialchars($_POST['last_name']);
17         $email = htmlspecialchars($_POST['email']);
18         $feedback = htmlspecialchars($_POST['feedback']);
19         $category = htmlspecialchars($_POST['category']);
20         $services = isset($_POST['services']) ? $_POST['services'] : [];
21
22         echo "<h2>Ваши данные:</h2>";
23         echo "<p><strong>Имя:</strong> " . $first_name . "</p>";
24         echo "<p><strong>Фамилия:</strong> " . $last_name . "</p>";
25         echo "<p><strong>Электронная почта:</strong> " . $email . "</p>";
26         echo "<p><strong>Категория:</strong> " . $category . "</p>";
27         echo "<p><strong>Обратная связь:</strong> " . $feedback . "</p>";
28         echo "<p><strong>Выбранные услуги:</strong> " . implode(", ", $services) . "</p>";
29     } else {
30         echo "<p>Форма не была отправлена корректно.</p>";
31     }
32     ?>
33
34     <br>
35     <a href="index.html">Вернуться к форме</a>
36 </body>
37 </html>
```

Рисунок 8 - Обработка формы

Функция `htmlspecialchars()` используется для защиты от XSS-атак (межсайтового скриптинга). `htmlspecialchars()` используется для форматирования данных, которые выводятся в контексте HTML. При этом управляющие символы HTML заменяются на их эквиваленты, чтобы какая-нибудь кавычка лишняя не испортила всю разметку.

В коде используется проверка функции `isset()` для определения, был ли отправлен массив с чекбоксами. Если ничего не было выбрано, переменной

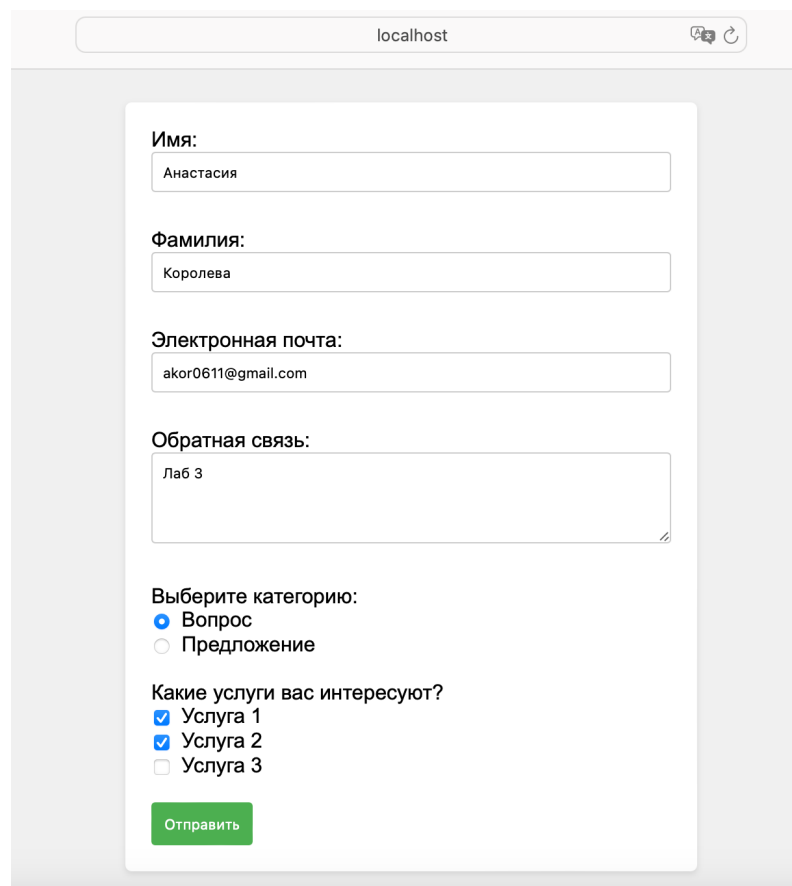
присваивается пустой массив, что предотвращает ошибки доступа к несуществующему значению.

PHP предоставляет два основных метода для передачи данных между браузером и сервером: GET и POST.

Метод GET используется для получения ресурса или страницы из веб-сервера. Он передает данные в URL-адресе запроса, и они видимы для любого пользователя. Поэтому этот метод используется для отправки малых данных.

Метод POST используется для отправки данных на сервер. Он передает данные в теле запроса, и они не видны для пользователя. Этот метод используется для отправки больших данных.

Результат работы представлен на рисунках.



The image shows a web browser window with the address bar displaying 'localhost'. The main content area contains a contact form with the following elements:

- Имя:** Text input field containing 'Анастасия'.
- Фамилия:** Text input field containing 'Королева'.
- Электронная почта:** Text input field containing 'akor0611@gmail.com'.
- Обратная связь:** Text area containing 'Лаб 3'.
- Выберите категорию:** Radio button group with 'Вопрос' selected and 'Предложение' unselected.
- Какие услуги вас интересуют?** Checkable list with 'Услуга 1' and 'Услуга 2' checked, and 'Услуга 3' unchecked.
- Отправить:** Green button at the bottom of the form.

Рисунок 9 - форма в браузере

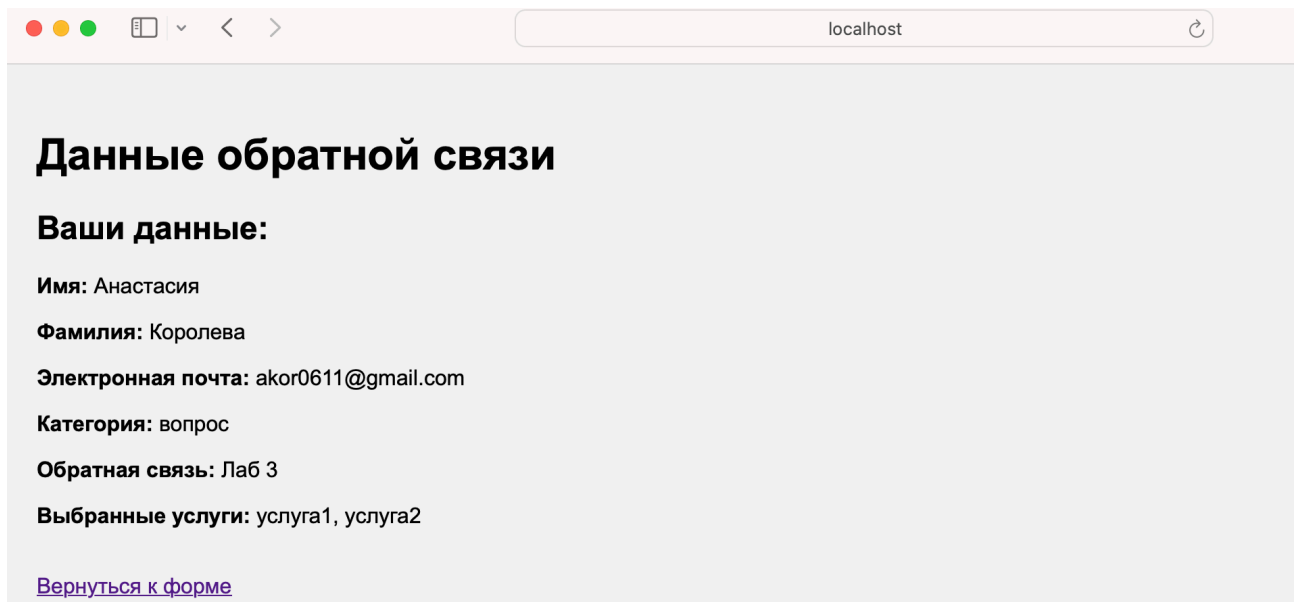


Рисунок 10 - Результат отправки формы

### Задание 3

Для установки локального сервера на Mac был использован пакет MAMP (Macintosh, Apache, MySQL, PHP). Это программное обеспечение предоставляет все необходимые компоненты для работы WordPress.

MAMP был скачан с официального сайта:  
<https://www.mamp.info/en/downloads/>

Затем скачан WordPress, распакованный архив загружен в папку MAMP\htdocs.

WordPress был скачан с официального сайта:  
<https://ru.wordpress.org/download/>

Была создана база данных wordpress и произведена установка WordPress в MAMP.

Настройка и установка WordPress в MAMP производилась по данной инструкции: <https://documentation.mamp.info/en/MAMP-Mac/FAQ/How-do-I-install-WordPress/>



По итогу открылся сайт на localhost.

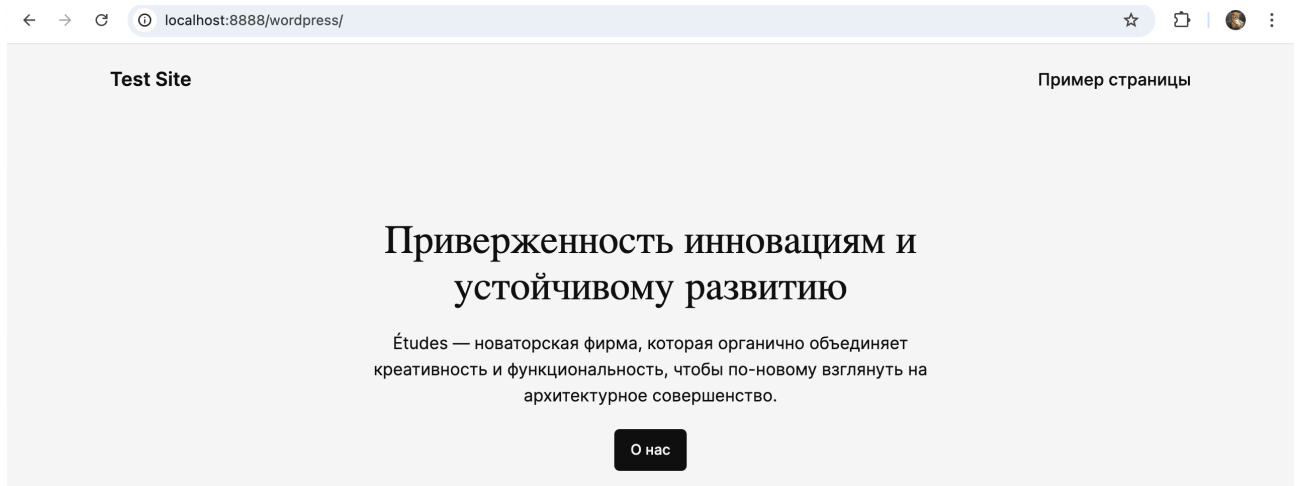


Рисунок 11 - wordpress на localhost

В файл etc/hosts была добавлена строка. С помощью файла hosts можно настроить соответствие между доменом и конкретным IP-адресом.

Выполнена команда:

```
sudo nano /etc/hosts
```

Добавлена следующая строка:

```
127.0.0.1 test.site
```

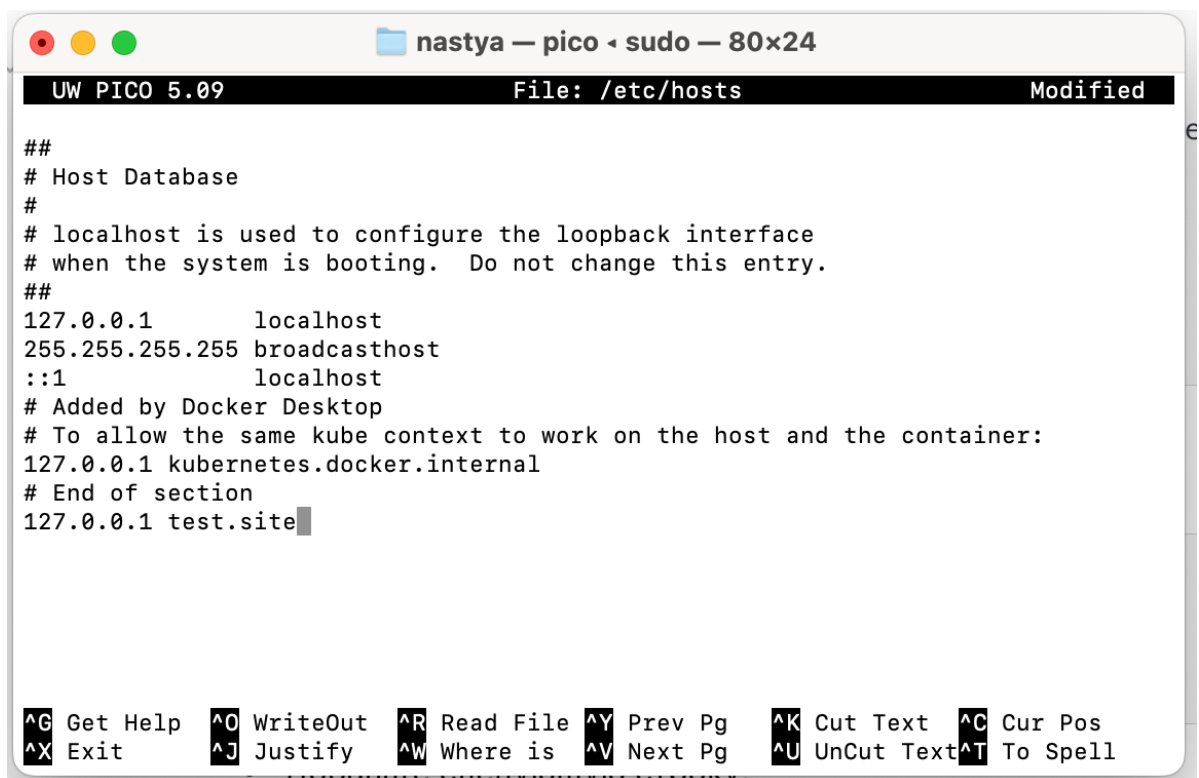
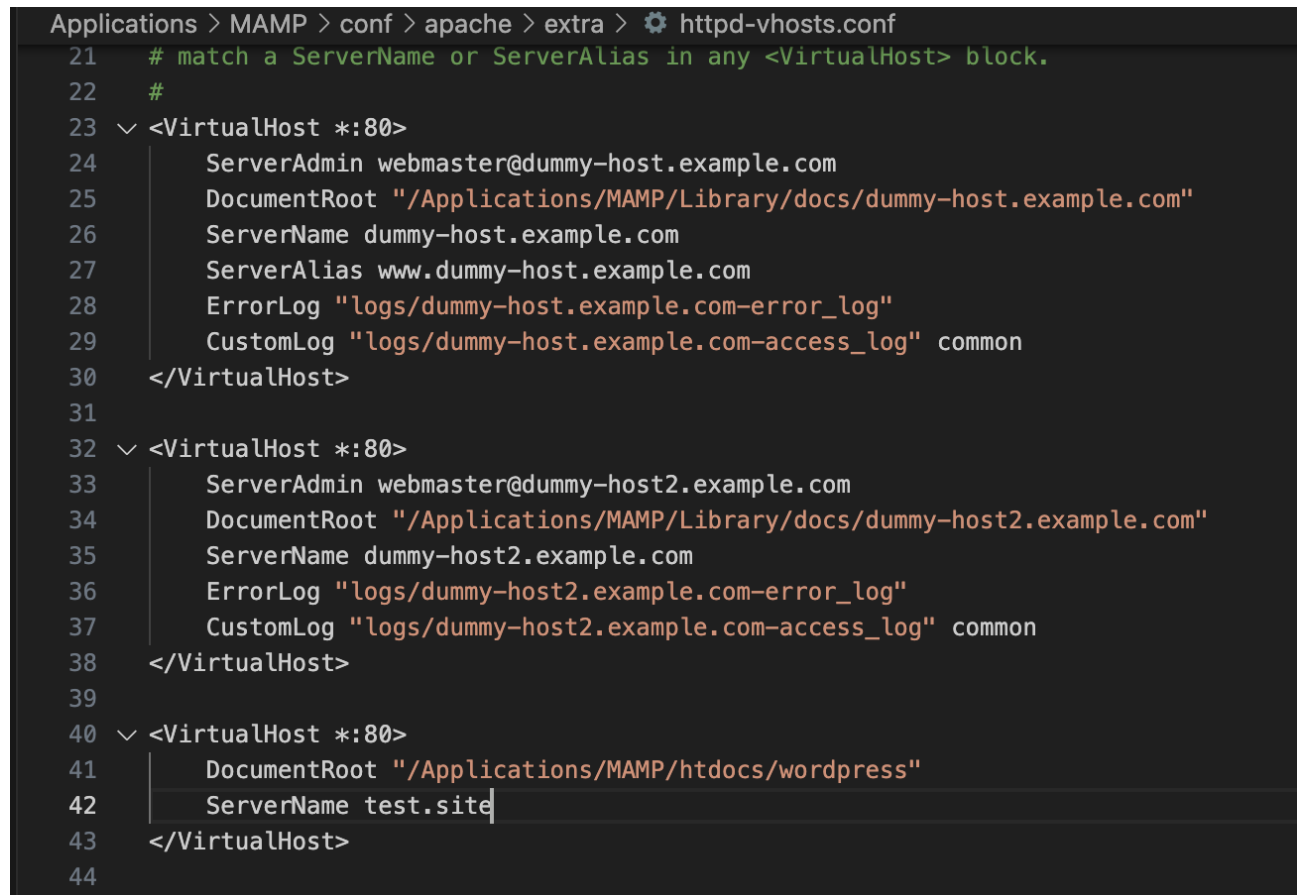


Рисунок 12 - настройка host

Для доступа к сайту по адресу `http://test.site` необходимо настроить виртуальный хост.

В VS Code открыт файл `/Applications/MAMP/conf/apache/extra/httpd-vhosts.conf`

Добавлен следующий код в конец файла (строки 40-43):



```
Applications > MAMP > conf > apache > extra > ⚙ httpd-vhosts.conf
21 # match a ServerName or ServerAlias in any <VirtualHost> block.
22 #
23 <VirtualHost *:80>
24     ServerAdmin webmaster@dummy-host.example.com
25     DocumentRoot "/Applications/MAMP/Library/docs/dummy-host.example.com"
26     ServerName dummy-host.example.com
27     ServerAlias www.dummy-host.example.com
28     ErrorLog "logs/dummy-host.example.com-error_log"
29     CustomLog "logs/dummy-host.example.com-access_log" common
30 </VirtualHost>
31
32 <VirtualHost *:80>
33     ServerAdmin webmaster@dummy-host2.example.com
34     DocumentRoot "/Applications/MAMP/Library/docs/dummy-host2.example.com"
35     ServerName dummy-host2.example.com
36     ErrorLog "logs/dummy-host2.example.com-error_log"
37     CustomLog "logs/dummy-host2.example.com-access_log" common
38 </VirtualHost>
39
40 <VirtualHost *:80>
41     DocumentRoot "/Applications/MAMP/htdocs/wordpress"
42     ServerName test.site
43 </VirtualHost>
44
```

Рисунок 13 - добавление виртуального хоста

В VS Code открыт файл `/Applications/MAMP/conf/apache/httpd.conf`.

Директива `Listen` указывает серверу принимать входящие запросы только на указанный порт(порты) или комбинации адреса и порта. Чтобы сервер принимал соединения как через порт 80, так и через порт 8888 на всех интерфейсах, была добавлена строка 56.

```
Applications > MAMP > conf > apache > ⚙ httpd.conf
44  #
45  #Mutex default
46
47  #
48  # Listen: Allows you to bind Apache to specific IP addresses
49  # ports, instead of the default. See also the <VirtualHost>
50  # directive.
51  #
52  # Change this to Listen on specific IP addresses as shown b
53  # prevent Apache from glomming onto all bound IP addresses.
54  #
55  #Listen 12.34.56.78:80
56  Listen 80
57  Listen 8888
58
```

Рисунок 14 - Добавление порта

Кроме того, в этом же файле была раскомментирована строка 668, которая указывает Apache загружать дополнительный файл конфигурации, который используется для настройки виртуальных хостов.

```
663
664  # Real-time info on requests and configuration
665  #Include /Applications/MAMP/conf/apache/extra/httpd-info.conf
666
667  # Virtual hosts
668  Include /Applications/MAMP/conf/apache/extra/httpd-vhosts.conf
669
670  # Local access to the Apache HTTP Server Manual
671  #Include /Applications/MAMP/conf/apache/extra/httpd-manual.conf
672
```

Рисунок 15 - добавление конфигурационного файла

В настройках WordPress был изменен адрес сайта.

Адрес WordPress (URL)	<input type="text" value="http://test.site"/>
Адрес сайта (URL)	<input type="text" value="http://test.site"/>
<small>Если вы хотите, чтобы адрес сайта <a href="#">отличался от директории установки WordPress</a>, введите адрес здесь.</small>	

Рисунок 16 - Изменение адреса сайта

Теперь при вводе http://test.site отображается сайт.

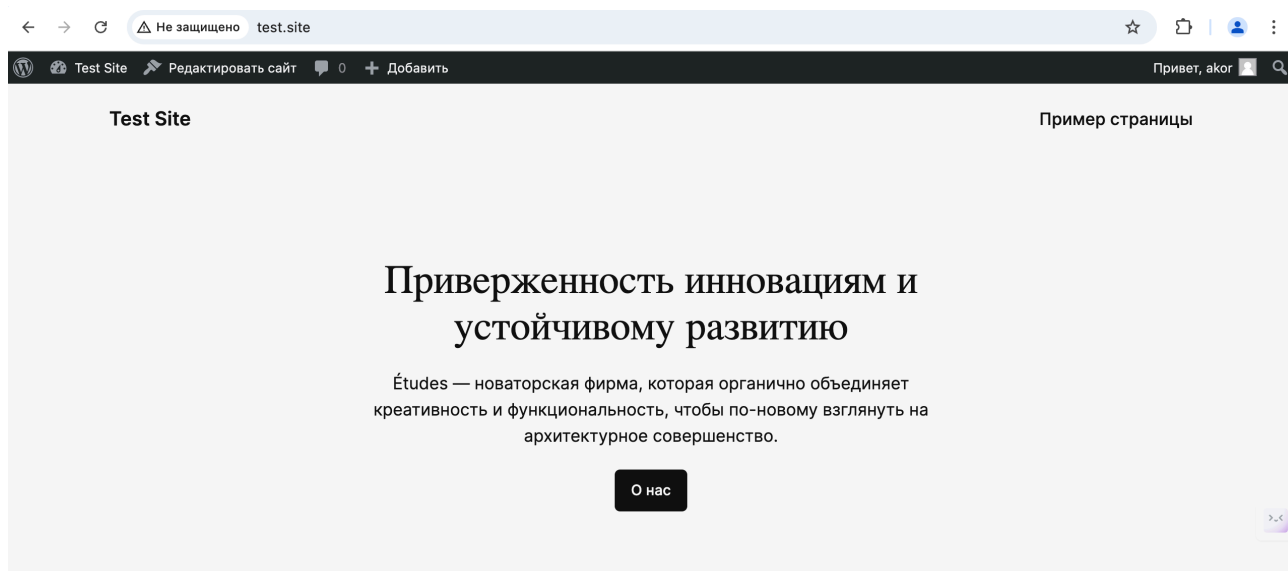


Рисунок 17 - Результат работы

**Вывод:** в данной практической работе получены навыки по работе с Gulp, а именно были созданы таски для последовательного и параллельного выполнения и настроена автоматическая перезагрузка при изменении одного из контролируемых файлов проекта; создана форма для отправки информации по обратной связи с дальнейшей обработкой данных php скриптом, настроен портал <http://test.site>.