

Санкт-Петербургский Национальный Исследовательский
Университет Информационных Технологий, Механики и
Оптики

Факультет Инфокоммуникационных Технологий

Практическая работа №3
«GULP, PHP, WordPress»

Выполнила:
Соколова Дарья Максимовна
Группа: К3323
Проверила:
Марченко Елена Вадимовна

Санкт-Петербург,

2024

Цель работы

Получить практические навыки по работе с Gulp и WordPress, созданию веб-страниц с дальнейшей обработкой данных php скриптом.

Задачи

1. Настроить Gulp, добавить задачи для последовательного и параллельного выполнения, настроить отображение файлов проекта в браузере и автоматическую перезагрузку при изменении одного из контролируемых файлов проекта;
2. Создать форму для отправки информации по обратной связи от пользователя, включая имя, фамилию, электронную почту, поле с обратной связью, добавить радиокнопки и чекбоксы. Обработать данные php скриптом;
3. Установить инструментарий для отладки проектов и WordPress, настроить портал <http://test.site>.

Ход работы

Задание 1

В первом задании необходимо настроить gulp и создать два таска: для последовательного и параллельного выполнения; а также настроить отображение файлов проекта в браузере и автоматическую перезагрузку при изменении одного из контролируемых файлов проекта.

Процесс установки Gulp был описан во второй практической работе, в данном задании был инициализирован проект командой `npm init -yes`, затем установлен gulp и другие необходимые пакеты.

```

● (base) darya@Darias-MacBook-Air 3.1 % npm init --yes
Wrote to /Users/darya/WebDevelopment_2024-2025/works/K3323/Соколова_Дарья/lab3/3.1/package.json:

{
  "name": "3.1",
  "version": "1.0.0",
  "description": "",
  "main": "gulpfile.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}

● (base) darya@Darias-MacBook-Air 3.1 % npm i gulp --save-dev
added 144 packages, and audited 145 packages in 3s

14 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

```

Рисунок 1 – Инициализация проекта

Для выполнения первой части задания было создано два файла gulp с простыми задачами, реализующими копирование файлов из одной папки в другую.

В первом файле – gulpfile_A_parallel.js (рисунок 2) задачи выполняются параллельно, для этого используется gulp.parallel – функция, которая позволяет запускать несколько задач одновременно, в данном примере copyJs и copyCss выполняются параллельно (рисунок 3).

```

1  const gulp = require("gulp");
2
3  function copyJs() {
4    |   return gulp.src("src/*.pug").pipe(gulp.dest("dist"));
5  }
6
7  function copyCss() {
8    |   return gulp.src("src/*.scss").pipe(gulp.dest("dist"));
9  }
10
11 exports.default = gulp.parallel(copyJs, copyCss);

```

Рисунок 2 – Файл gulpfile_A_parallel.js

```

● (base) darya@Darias-MacBook-Air 3.1 % gulp --gulpfile gulpfile_A_parallel.js
[15:36:08] Using gulpfile ~/WebDevelopment_2024-2025/works/K3323/Соколова_Дарья/lab3/3.1/gulpfile_A_parallel.js
[15:36:08] Starting 'default'...
[15:36:08] Starting 'copyJs'...
[15:36:08] Starting 'copyCss'...
[15:36:08] Finished 'copyJs' after 10 ms
[15:36:08] Finished 'copyCss' after 11 ms
[15:36:08] Finished 'default' after 12 ms

```

Рисунок 3 – Параллельное выполнение задач

Во втором файле – gulpfile_A_series.js (рисунок 4) задачи выполняются последовательно, для этого используется gulp.series – функция, с которой

задачи выполняются один за другим, в данном примере сначала выполняется `copyJs`, а затем `copyCss` (рисунок 5).

```
1  const gulp = require("gulp");
2
3  function copyJs() {
4    |   return gulp.src("src/*.pug").pipe(gulp.dest("dist"));
5  }
6
7  function copyCss() {
8    |   return gulp.src("src/*.scss").pipe(gulp.dest("dist"));
9  }
10
11 exports.default = gulp.series(copyJs, copyCss);
```

Рисунок 4 – Файл `gulpfile_A_series.js`

```
● (base) darya@Darias-MacBook-Air 3.1 % gulp --gulpfile gulpfile_A_series.js
[15:40:08] Using gulpfile ~/WebDevelopment_2024-2025/works/K3323/Соколова_Дарья/lab3/3.1/gulpfile_A_series.js
[15:40:08] Starting 'default'...
[15:40:08] Starting 'copyJs'...
[15:40:08] Finished 'copyJs' after 9.45 ms
[15:40:08] Starting 'copyCss'...
[15:40:08] Finished 'copyCss' after 2.56 ms
[15:40:08] Finished 'default' after 13 ms
```

Рисунок 5 – Последовательное выполнение задач

Для выполнения второй части задания использовался `BrowserSync` – чтобы происходила автоматическая перезагрузка при изменении одного из контролируемых файлов. Пример выполнения созданного файла – на рисунке 6. Отображение страницы в браузере – на рисунке 7.

```
○ (base) darya@Darias-MacBook-Air 3.1 % gulp --gulpfile gulpfile_B.js
[15:17:34] Using gulpfile ~/WebDevelopment_2024-2025/works/K3323/Соколова_Дарья/lab3/3.1/gulpfile_B.js
[15:17:34] Starting 'default'...
[15:17:34] Starting 'pug'...
[15:17:34] Starting 'scss'...
Deprecation Warning: The legacy JS API is deprecated and will be removed in Dart Sass 2.0.0.
More info: https://sass-lang.com/d/legacy-js-api
[15:17:34] Finished 'scss' after 48 ms
[15:17:34] Finished 'pug' after 48 ms
[15:17:34] Starting 'browser-sync'...
[Browsersync] Access URLs:
    Local: http://localhost:3002
    External: http://192.168.31.24:3002
    UI: http://localhost:3003
    UI External: http://192.168.31.24:3003
[Browsersync] Serving files from: ./build
```

Рисунок 6 – Выполнение файла `gulpfile_B.js`

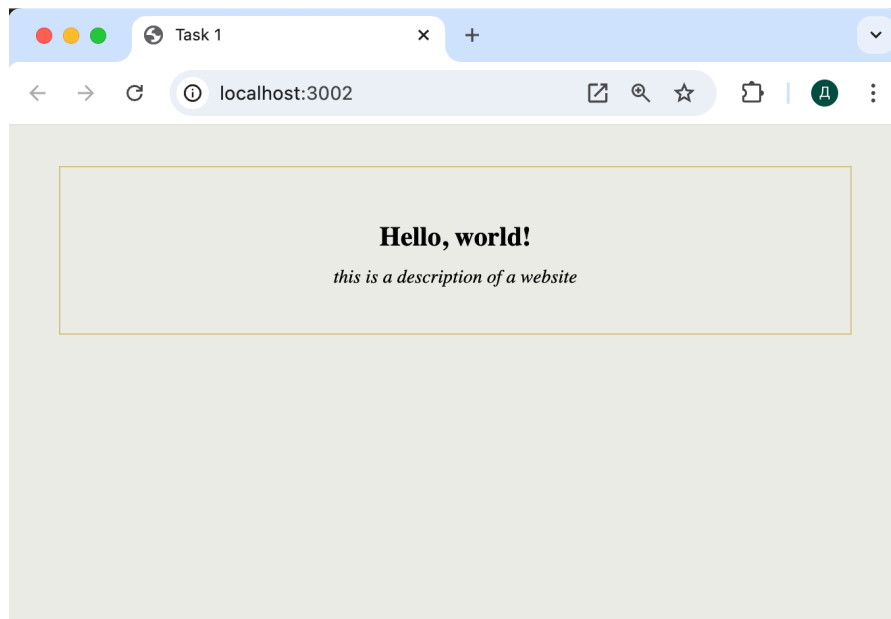


Рисунок 7 – Отображение страницы в браузере

Для демонстрации автоматической перезагрузки страницы при изменении файла были изменены некоторые стили в файле `styles.scss`, а затем текст в файле `index.pug`. Видим, что изменения применились и отображаются в браузере (рисунки 8-9).

```
[15:21:44] Starting 'scss'...  
Deprecation Warning: The legacy JS API is deprecated and will be removed in Dart Sass 2.0.0.  
More info: https://sass-lang.com/d/legacy-js-api  
[Browsersync] 1 file changed (styles.css)  
[15:21:44] Finished 'scss' after 17 ms  
[15:22:42] Starting 'pug'...  
[Browsersync] 1 file changed (index.html)  
[15:22:42] Finished 'pug' after 26 ms  
[Browsersync] Reloading Browsers...
```

Рисунок 8 – Внесение изменений в файлах

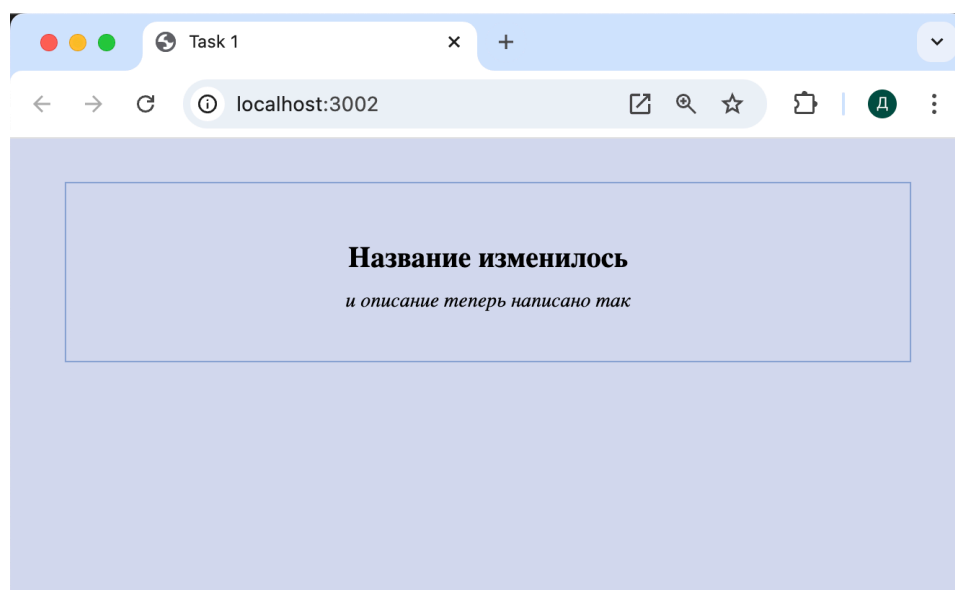


Рисунок 9 – Автоматическая перезагрузка страницы

Задание 2

Во втором задании нужно создать форму для отправки информации от пользователя с данными его имени, фамилии, электронной почты, поля с обратной связью, радиокнопками и чекбоксами.

Для выполнения задания был установлен XAMPP, позволяющий быстро настроить локальную среду для работы с PHP и MySQL, а также управлять базами данных MySQL через phpMyAdmin.

Сначала в MySQL через phpMyAdmin создана структура таблицы feedback (рисунок 10).


#	Name	Type	Collation	Attributes	Null	Default
<input type="checkbox"/> 1	id 	int(11)			No	None
<input type="checkbox"/> 2	first_name	varchar(50)	utf8mb4_general_ci		No	None
<input type="checkbox"/> 3	last_name	varchar(50)	utf8mb4_general_ci		No	None
<input type="checkbox"/> 4	email	varchar(100)	utf8mb4_general_ci		No	None
<input type="checkbox"/> 5	feedback	text	utf8mb4_general_ci		No	None
<input type="checkbox"/> 6	gender	varchar(10)	utf8mb4_general_ci		No	None
<input type="checkbox"/> 7	options	text	utf8mb4_general_ci		No	None
<input type="checkbox"/> 8	created_at	timestamp			No	current_timestamp()

Рисунок 10 – Структура таблицы feedback

В папке XAMPP\htdocs\ создана папка, в которой созданы файлы index.html с разметкой (рисунок 11), process_feedback.php с обработкой данных из формы и занесением их в таблицу БД MySQL (рисунок 12) и styles.css со стилями.

Методы GET и POST — это два наиболее распространённых метода HTTP, используемых для передачи данных между клиентом и сервером.

Метод GET используется для получения данных с сервера. Данные передаются через URL в виде параметров, что делает их уязвимыми для перехватов.

Метод POST используется для отправки данных на сервер. Данные передаются в теле запроса, что является более безопасным для передачи конфиденциальной информации.

```

<div class="container">
  <h1>Форма обратной связи</h1>
  <form action="process_feedback.php" method="post">
    <label for="firstName">Имя:</label>
    <input
      type="text"
      id="firstName"
      name="firstName"
      required
    /><br /><br />

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

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

    <label for="feedback">Обратная связь:</label><br />
    <textarea id="feedback" name="feedback" rows="5" cols="50" required></textarea>
    <br />

    <p>Пол:</p>
    <input type="radio" id="male" name="gender" value="Мужчина" required />
    <label for="male">Мужчина</label><br />
    <input type="radio" id="female" name="gender" value="Женщина" />
    <label for="female">Женщина</label><br />

    <p>Выберите опции:</p>
    <input type="checkbox" id="option1" name="options[]" value="Опция 1" />
    <label for="option1">Опция 1</label><br />
    <input type="checkbox" id="option2" name="options[]" value="Опция 2" />
    <label for="option2">Опция 2</label><br />
    <input type="checkbox" id="option3" name="options[]" value="Опция 3" />
    <label for="option3">Опция 3</label><br /><br />

    <button type="submit">Отправить</button>
  </form>
</div>

```

Рисунок 11 – Файл index.html

```

<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "feedback_db";

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

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

$firstName = $_POST['firstName'];
$lastName = $_POST['lastName'];
$email = $_POST['email'];
$feedback = $_POST['feedback'];
$gender = $_POST['gender'];
$options = implode(" ", $_POST['options']);

$sql = "INSERT INTO feedback (first_name, last_name, email, feedback, gender, options)
VALUES ('$firstName', '$lastName', '$email', '$feedback', '$gender', '$options')";

if ($conn->query($sql) === TRUE) {
    echo "Обратная связь принята!";
} else {
    echo "Ошибка: " . $sql . "<br>" . $conn->error;
}

$conn->close();
?>

```

Рисунок 12 – Файл process_feedback.php

Веб-страница и пример ввода данных представлены на рисунке 13, таблица БД после занесения данных о нескольких пользователях – на рисунке 14.

Форма обратной связи

Имя:
Иван

Фамилия:
Иванов

Электронная почта:
ivan.ivanov@mail.ru

Обратная связь:
мой комментарий

Пол:
☒ Мужчина
☐ Женщина

Выберите опции:
☒ Опция 1
☒ Опция 2
☐ Опция 3

Отправить

Рисунок 13 – Веб-страница

	id	first_name	last_name	email	feedback	gender	options	created_at
<input type="checkbox"/> Edit Copy Delete	1	Дарья	Соколова	dasha.sokolova2204@yandex.ru	без замечаний	Женщина	Опция 1, Опция 3	2024-10-09 16:34:03
<input type="checkbox"/> Edit Copy Delete	2	Иван	Иванов	ivan.ivanov@mail.ru	мой комментарий	Мужчина	Опция 1, Опция 2	2024-10-09 17:01:23

Рисунок 14 – Таблица БД после внесения данных

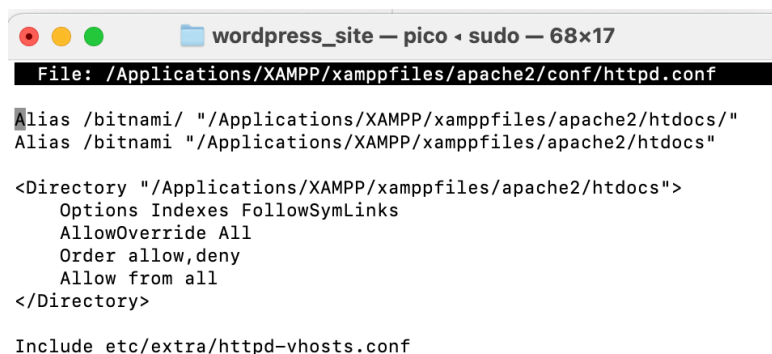
Задание 3

В третьем задании необходимо установить инструментарий для отладки проектов и WordPress, настроить портал <http://test.site>, чтобы при вводе данного адреса открывался WordPress.

Сначала был установлен XAMPP, в XAMPP Control Panel запущены модули Apache и MySQL. Затем скачан WordPress, распакованный архив загружен в папку XAMPP\htdocs\.

Создана база данных `wordpress_db` в MySQL и произведена установка WordPress.

В файле `httpd.conf` добавлена строка (рисунок 15), указывающая Apache загружать и использовать настройки, определенные в файле `httpd-vhosts.conf`.

A screenshot of a terminal window titled "wordpress_site — pico • sudo — 68x17". The terminal shows the contents of the file `/Applications/XAMPP/xamppfiles/apache2/conf/httpd.conf`. The visible code includes two `Alias` directives pointing to `/Applications/XAMPP/xamppfiles/apache2/htdocs/"`, a `<Directory>` block for `/Applications/XAMPP/xamppfiles/apache2/htdocs/"` with options `Indexes FollowSymLinks`, `AllowOverride All`, `Order allow,deny`, and `Allow from all`, and an `Include` directive for `etc/extra/httpd-vhosts.conf`.

```
File: /Applications/XAMPP/xamppfiles/apache2/conf/httpd.conf

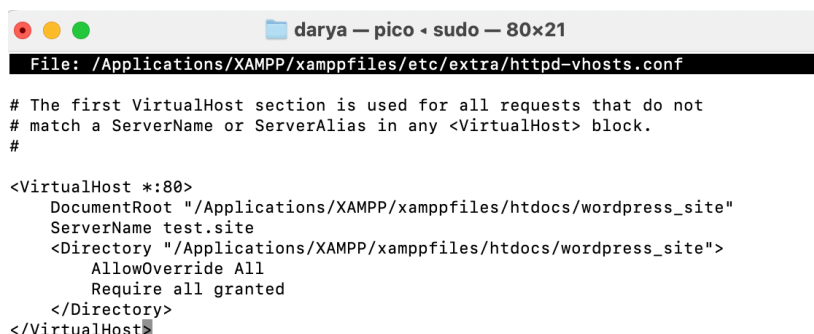
Alias /bitnami/ "/Applications/XAMPP/xamppfiles/apache2/htdocs/"
Alias /bitnami "/Applications/XAMPP/xamppfiles/apache2/htdocs"

<Directory "/Applications/XAMPP/xamppfiles/apache2/htdocs">
    Options Indexes FollowSymLinks
    AllowOverride All
    Order allow,deny
    Allow from all
</Directory>

Include etc/extra/httpd-vhosts.conf
```

Рисунок 15 – Изменения в файле `httpd.conf`

Затем в файле `httpd-vhosts.conf` добавлен код для настройки виртуального хоста (рисунок 16) с указанием директории (`wordpress_site`), из которой Apache будет обслуживать файлы для этого виртуального хоста.

A screenshot of a terminal window titled "darya — pico • sudo — 80x21". The terminal shows the contents of the file `/Applications/XAMPP/xamppfiles/etc/extra/httpd-vhosts.conf`. The visible code includes a comment block, a `<VirtualHost>` block for `*:80` with `DocumentRoot` set to `/Applications/XAMPP/xamppfiles/htdocs/wordpress_site`, `ServerName` set to `test.site`, and a `<Directory>` block for `/Applications/XAMPP/xamppfiles/htdocs/wordpress_site` with `AllowOverride All` and `Require all granted`.

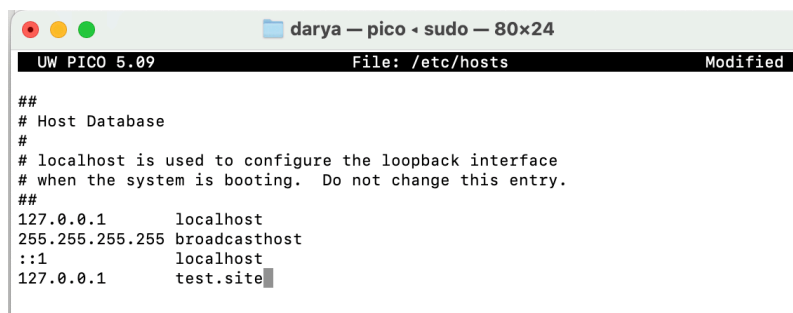
```
File: /Applications/XAMPP/xamppfiles/etc/extra/httpd-vhosts.conf

# The first VirtualHost section is used for all requests that do not
# match a ServerName or ServerAlias in any <VirtualHost> block.
#

<VirtualHost *:80>
    DocumentRoot "/Applications/XAMPP/xamppfiles/htdocs/wordpress_site"
    ServerName test.site
    <Directory "/Applications/XAMPP/xamppfiles/htdocs/wordpress_site">
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

Рисунок 16 – Изменения в файле `httpd-vhosts.conf`

В файле `hosts` добавлена строка `127.0.0.1 test.site` (рисунок 17), связывающая доменное имя `test.site` с локальным IP-адресом `127.0.0.1`.

A screenshot of a terminal window titled "darya — pico • sudo — 80x24". The terminal shows the contents of the file `/etc/hosts`. The visible code includes a comment block and a list of IP addresses mapped to hostnames, with `127.0.0.1` mapped to `localhost` and `test.site`.

```
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 17 – Изменения в файле `hosts`

В БД `wordpress_db` в таблице `wp_options` были изменены параметры `siteurl` и `home` (рисунок 18).

<input type="checkbox"/>	Edit	Copy	Delete	2 siteurl	http://test.site	on
<input type="checkbox"/>	Edit	Copy	Delete	3 home	http://test.site	on

Рисунок 18 – Изменения в таблице `wp_options`

Теперь при вводе `http://test.site` отображается сайт (рисунок 19).

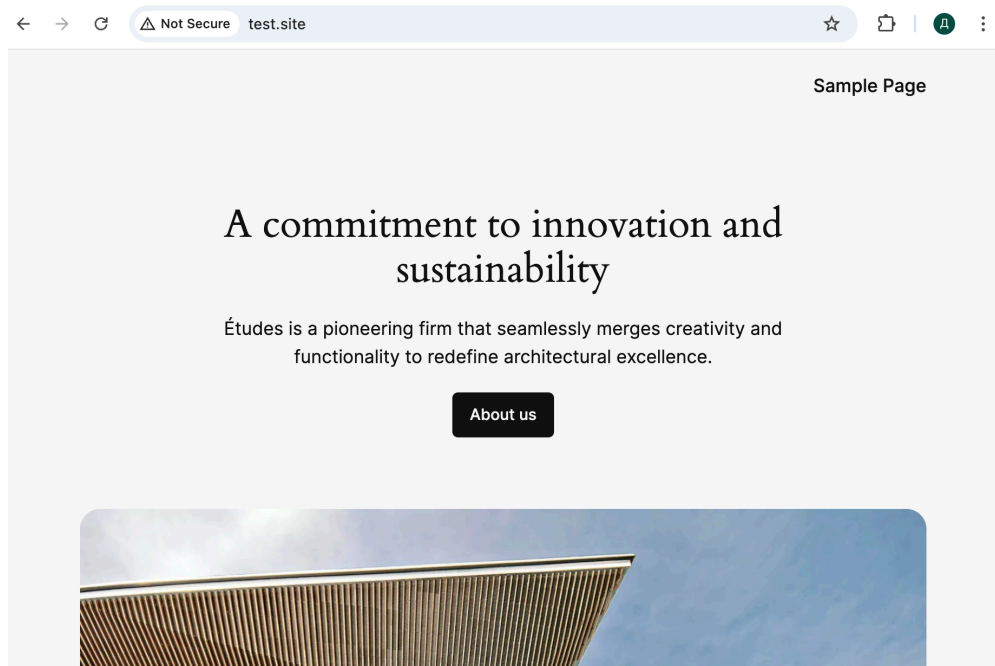


Рисунок 19 – Отображаемый сайт при вводе `http://test.site`

Вывод

В данной практической работе получены практические 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>.