

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

Факультет инфокоммуникационных технологий

**Лабораторная работа №3**

Выполнил: Долгов К.М.

Проверила: Марченко Е.В.

Санкт-Петербург,

2024

## Оглавление

Введение .....	3
Ход работы .....	4
1. Работа с Gulp.....	4
2. Знакомство с PHP.....	8
3. Работа с WordPress .....	12
Заключение.....	15

## **Введение**

**Цель работы:** изучить основы работы с Gulp, написать сайт с формой обратной связи, познакомиться с PHP и WordPress.

## Ход работы

### 1. Работа с Gulp.

В данном задании необходимо создать два таска. (Рисунок 1).

```
function task1(done) {  
  console.log('Первый таск');  
  done();  
}  
function task2(done) {  
  console.log('Второй таск');  
  done();  
}
```

Рисунок 1 – Создание тасков

Было создано два простых таска с выводом в консоль. Теперь, с помощью функции `series()` и `parallel()` было настроено выполнение тасков последовательно и параллельно соответственно (Рисунок 2).

```
// Последовательное выполнение задач  
const series = gulp.series(task1, task2);  
  
// Параллельное выполнение задач  
const parallel = gulp.parallel(task1, task2);  
  
exports.series = series;  
exports.parallel = parallel;
```

Рисунок 2 – Настройка выполнения тасков

Далее задачи были запущены через консоль (Рисунок 3).

```
[konstantindolgov@MacBook-Pro-Konstantin-4 task_1 % gulp series
[17:54:09] Using gulpfile ~/Web/lab_3/task_1/gulpfile.js
[17:54:09] Starting 'series'...
[17:54:09] Starting 'task1'...
Первый task
[17:54:09] Finished 'task1' after 406 µs
[17:54:09] Starting 'task2'...
Второй task
[17:54:09] Finished 'task2' after 344 µs
[17:54:09] Finished 'series' after 1.75 ms
[konstantindolgov@MacBook-Pro-Konstantin-4 task_1 % gulp parallel
[17:54:16] Using gulpfile ~/Web/lab_3/task_1/gulpfile.js
[17:54:16] Starting 'parallel'...
[17:54:16] Starting 'task1'...
[17:54:16] Starting 'task2'...
Первый task
[17:54:16] Finished 'task1' after 664 µs
Второй task
[17:54:16] Finished 'task2' after 715 µs
[17:54:16] Finished 'parallel' after 1.64 ms
konstantindolgov@MacBook-Pro-Konstantin-4 task_1 %
```

Рисунок 3 – Выполнение задач

Теперь для подпункта б был добавлен еще один task, который запускает сервер и перезагружает его при изменении файлов html и css, что облегчает разработку (Рисунок 4).

```
function watchFiles() {
  browserSync.init({
    server: {
      baseDir: "./src"
    }
  });

  gulp.watch("./src/*.html").on('change', browserSync.reload);
  gulp.watch("./src/*.css").on('change', browserSync.reload);
}
```

Рисунок 4 – Задача для подпункта б

После запуска таска через консоль (Рисунок 5) была открыта страница (Рисунок 6)

```
[konstantindolgov@MacBook-Pro-Konstantin-4 task_1 % gulp watch  
[18:17:23] Using gulpfile ~/Web/lab_3/task_1/gulpfile.js  
[18:17:23] Starting 'watch'...  
[Browsersync] Access URLs:  
-----  
Local: http://localhost:3000  
External: http://192.168.3.152:3000  
-----  
UI: http://localhost:3001  
UI External: http://192.168.3.152:3001  
-----  
[Browsersync] Serving files from: ./src
```

Рисунок 5 – Запуск таска

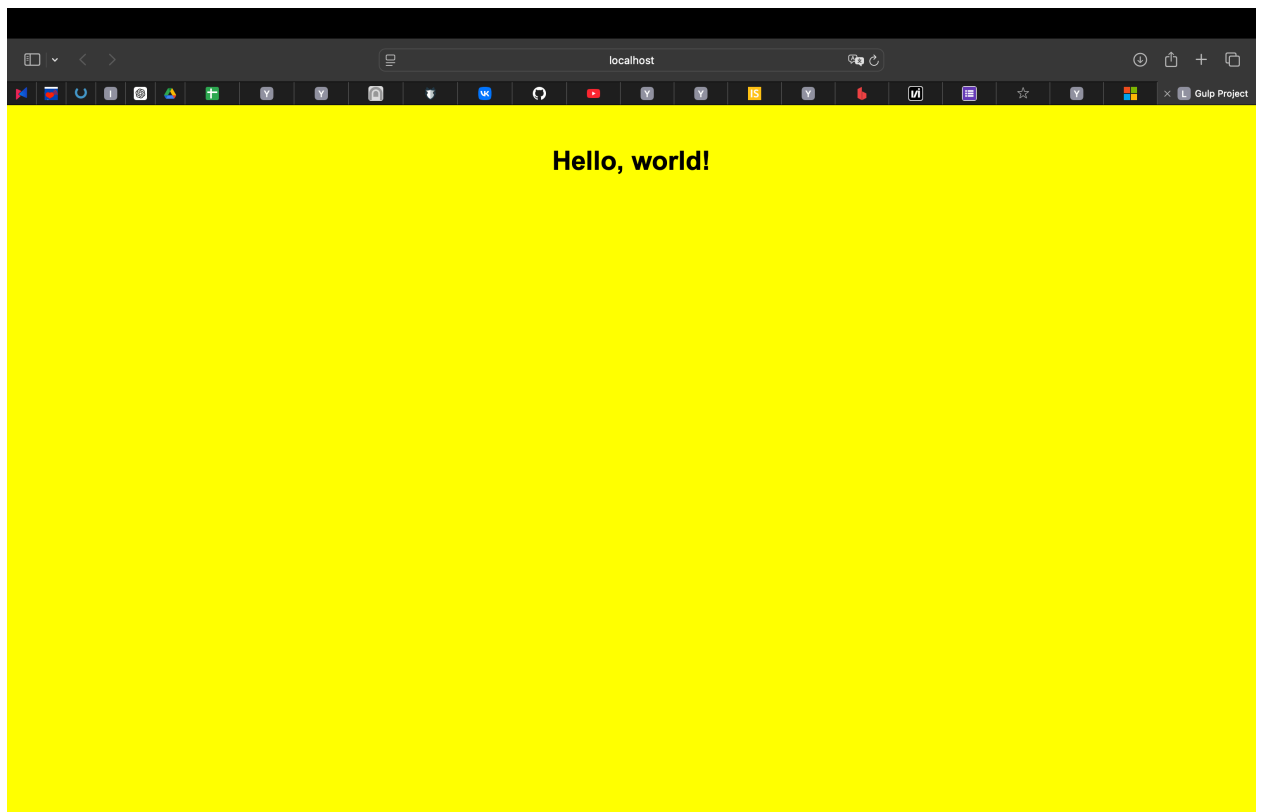


Рисунок 6 – Страница

Теперь, если поменять файлы проекта, то страница автоматически перезагрузится (Рисунок 7-8).

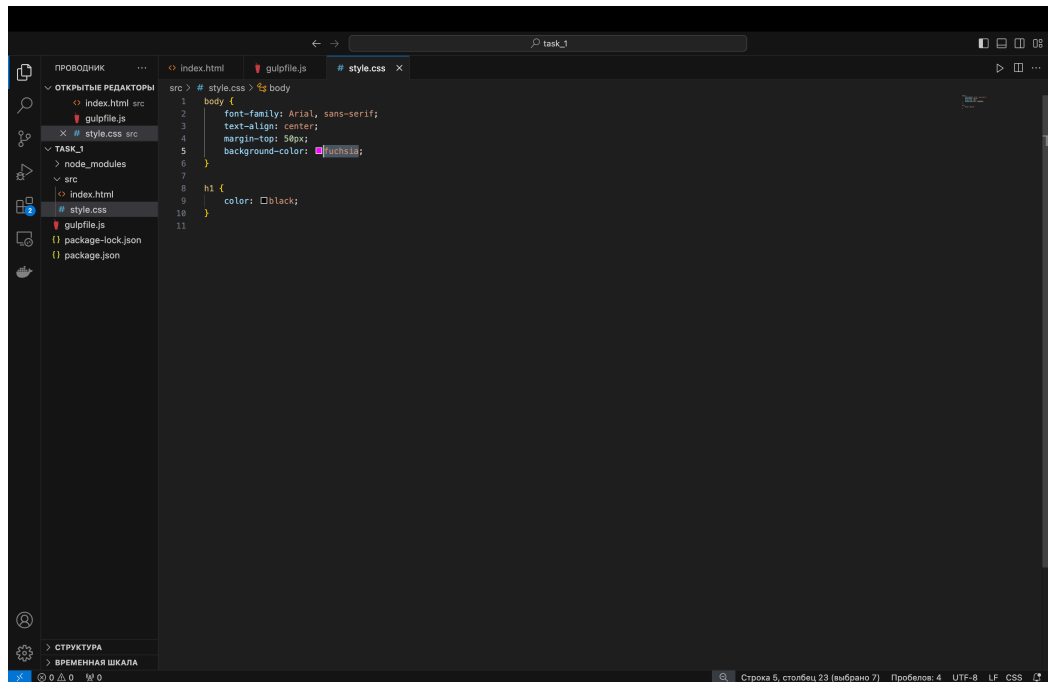


Рисунок 7 – Изменения файла CSS

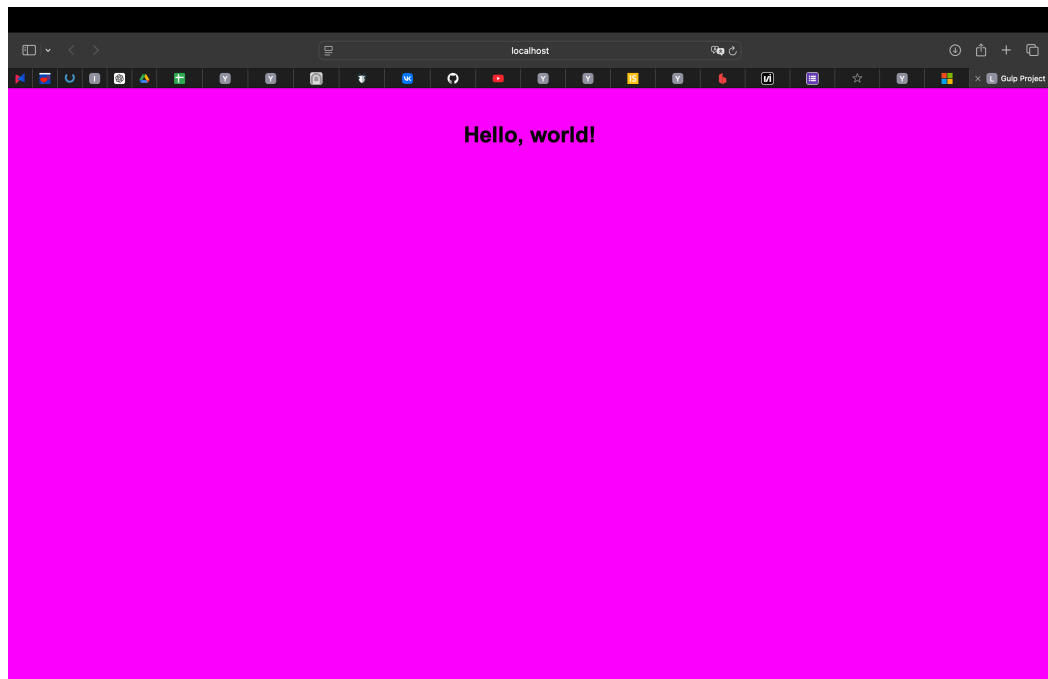


Рисунок 8 – Измененная страница

Изменения файлов отображаются в консоли (Рисунок 9).

```
[Browsersync] File event [change] : src/style.css
```

Рисунок 9 – Изменение файла в консоли

## 2. Знакомство с PHP.

В данном задании необходимо создать сайт с формой обратной связи. Был написан код формы на html (Рисунок 10).

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Обратная связь по соревнованиям</title>
</head>
<body>
  <h1>Форма обратной связи по соревнованиям</h1>
  <form action="process.php" method="post">
    <label for="first_name">Имя:</label>
    <input type="text" id="first_name" name="first_name" required><br><br>

    <label for="last_name">Фамилия:</label>
    <input type="text" id="last_name" name="last_name" 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="4" cols="50" required></textarea><br><br>

    <p>Выберите категорию:</p>
    <input type="radio" id="sport" name="category" value="Спортсмены" required>
    <label for="sport">Спортсмены</label><br>
    <input type="radio" id="new" name="category" value="Новички">
    <label for="new">Новички</label><br><br>

    <p>Выберите дистанции, в которых участвовали (минимум одну):</p>
    <input type="checkbox" id="topic1" name="topics[]" value="500 м">
    <label for="topic1">500 м</label><br>
    <input type="checkbox" id="topic2" name="topics[]" value="1000 м">
    <label for="topic2">1000 м</label><br>
    <input type="checkbox" id="topic3" name="topics[]" value="2000 м">
    <label for="topic3">2000 м</label><br><br>

    <button type="submit">Отправить</button>
  </form>
</body>
</html>
```

Рисунок 10 – Код формы

Далее был написан код на PHP, который считывает POST-запрос, заносит их в переменные и выводит на экран сообщение о «Спасибо за ваш отзыв!» и данные из формы (Рисунок 11).



```

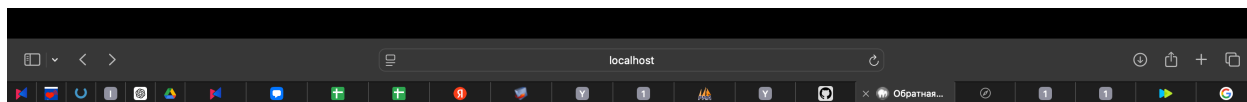
<?php
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    // Получение данных из формы
    $first_name = htmlspecialchars($_POST['first_name']);
    $last_name = htmlspecialchars($_POST['last_name']);
    $email = htmlspecialchars($_POST['email']);
    $feedback = htmlspecialchars($_POST['feedback']);
    $category = htmlspecialchars($_POST['category']);
    $topics = isset($_POST['topics']) ? $_POST['topics'] : [];

    // Вывод обработанной информации
    echo "<h1>Спасибо за ваш отзыв!</h1>";
    echo "<p><strong>Имя:</strong> $first_name</p>";
    echo "<p><strong>Фамилия:</strong> $last_name</p>";
    echo "<p><strong>Email:</strong> $email</p>";
    echo "<p><strong>Категория:</strong> $category</p>";
    echo "<p><strong>Дистанция:</strong> " . implode(", ", $topics) . "</p>";
    echo "<p><strong>Сообщение:</strong> $feedback</p>";
} else {
    echo "<h1>Ошибка: Данные не отправлены.</h1>";
}
?>

```

Рисунок 11 – PHP код.

Для запуска сайта и работы PHP используется MAMP (Macintosh, Apache, MySQL, PHP) который создает локальный сервер. Файлы проекта были перемещены в папку htdocs в папке feedback, после чего открыты в браузере по адресу localhost/feedback (Рисунок 11-12).



### Форма обратной связи по соревнованиям

Имя:  
Константин

Фамилия:  
Долгов

Электронная почта:  
konstantindolgov2302@mai.ru

Обратная связь:  
Тестовый отзыв

Выберите категорию:

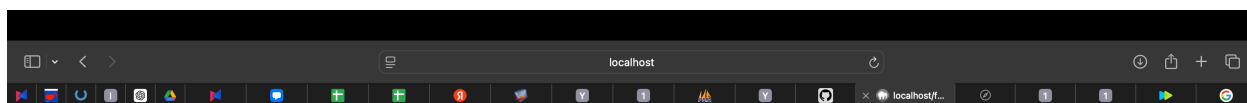
☒ Спортсмены  
☐ Новички

Выберите дистанции, в которых участвовали (минимум одну):

☒ 500 м  
☐ 1000 м  
☒ 2000 м

Отправить

Рисунок 12 – Форма



### Спасибо за ваш отзыв!

Имя: Константин  
Фамилия: Долгов  
Email: konstantindolgov2302@mai.ru  
Категория: Спортсмены  
Дистанция: 500 м, 2000 м  
Сообщение: Тестовый отзыв

Рисунок 13 – Ответ PHP

GET и POST — это два наиболее распространённых метода HTTP-запросов, которые используются для взаимодействия клиента (обычно веб-браузера) с сервером.

GET используется для запросов, которые не изменяют состояние сервера (например, получение информации, отображение страниц). Параметры передаются через URL (в строке запроса после ?), видны в адресной строке и могут быть закэшированы. Также есть ограничение на длину URL (в зависимости от браузера/сервера).

POST используется для отправки данных на сервер (например, при отправке формы или загрузке файла), операций, которые могут изменять состояние сервера (например, добавление записи в базу данных). Данные передаются в теле запроса, а не в URL, что подходит для передачи больших объёмов данных или чувствительной информации. Также POST-запросы не кэшируются браузером (по умолчанию).

### 3. Работа с WordPress

Для реализации третьего задания был скачан пакет wordpress и распакован в htdocs. В MySQL была создана БД для сайта (Рисунок 14).



Рисунок 14 – Создание БД в MySQL

Далее запускаем localhost/wordpress и заполняем формы данными о пользователе и сайте (Рисунок 15).

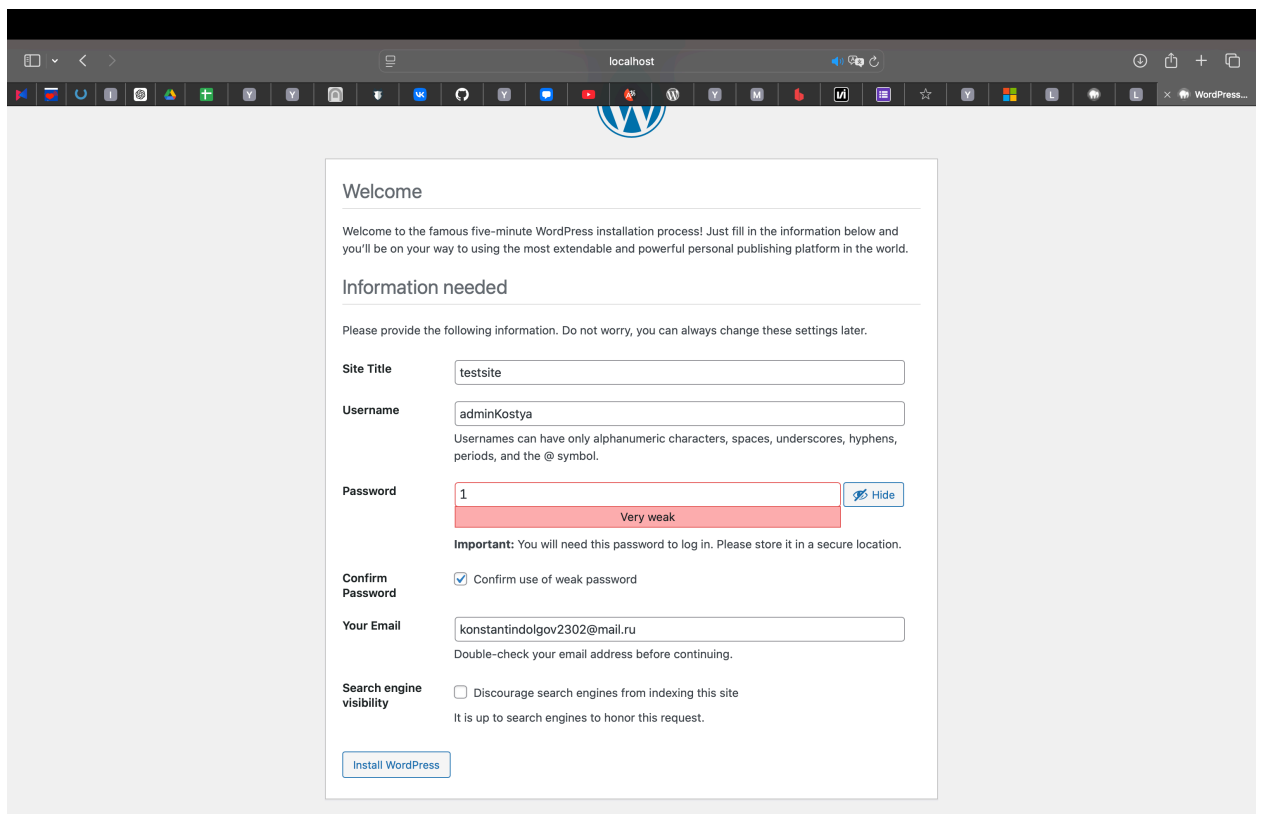
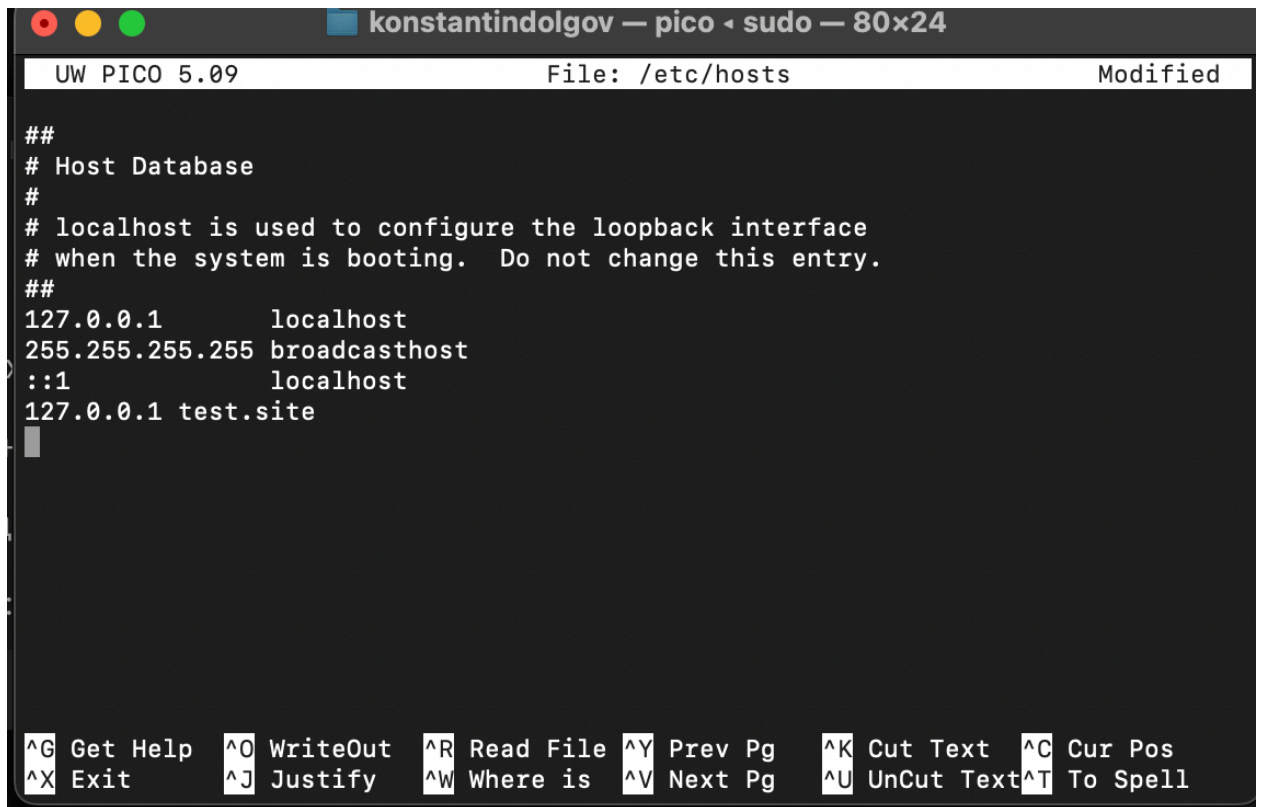


Рисунок 15 – Заполнение данных о сайте

Теперь необходимо сделать так, чтобы сайт открывался по адресу test.site. Открывает файл host и добавляем в него адрес (Рисунок 16).



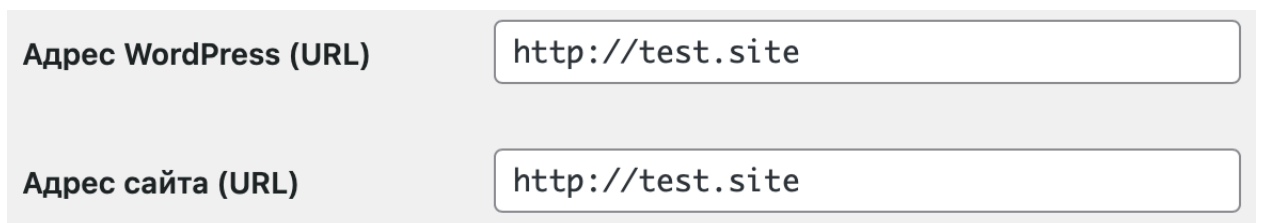
The screenshot shows a terminal window with the title bar "konstantindolgov — pico — sudo — 80x24". The editor is in "UW PICO 5.09" mode, editing the file "/etc/hosts". The content of the file is as follows:

```
##
# 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
```

The bottom status bar of the editor shows various keyboard shortcuts: ^G Get Help, ^O WriteOut, ^R Read File, ^Y Prev Pg, ^K Cut Text, ^C Cur Pos, ^X Exit, ^J Justify, ^W Where is, ^V Next Pg, ^U UnCut Text, and ^T To Spell.

Рисунок 16 – Файл host

В настройках wordpress также указываем адрес (Рисунок 17).



The screenshot shows the WordPress settings interface. There are two input fields, both containing the text "http://test.site".

Адрес WordPress (URL)	http://test.site
Адрес сайта (URL)	http://test.site

Рисунок 17 – Настройки wordpress

В файле /Applications/MAMP/conf/apache/extra/httpd-vhosts.conf настраиваем виртуальные хосты (Рисунок 18).

```

<VirtualHost *:80>
    ServerName test.site
    DocumentRoot "/Applications/MAMP/htdocs/wordpress"
</VirtualHost>

<VirtualHost *:80>
    ServerName localhost
    DocumentRoot "/Applications/MAMP/htdocs"
</VirtualHost>

```

Рисунок 18 – Настройка виртуальных хостов

В файле конфигурации /Applications/MAMP/conf/apache/httpd.conf включаем этот файл конфигурации (Рисунок 19).

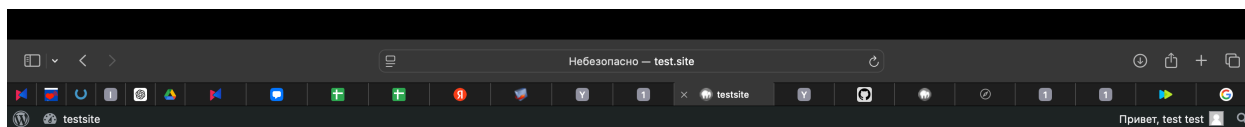
```

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

```

Рисунок 19 – Настройка MAMP

Теперь проверяем, открывается ли сайт по адресу (Рисунок 20).



testsite

Пример страницы

## Блог

Привет, мир!

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

09.12.2024

testsite

Блог

Мероприятия

О нас

Магазин

Рисунок 20 – Проверка работоспособности

## **Заключение**

В ходе выполнения данной лабораторной работы были изучены основы работы с Gulp, написан сайт с формой обратной связи. Были написан файл PHP для обработки формы и настроен WordPress. Также были получены навыки настройки локального сервера.