Санкт-Петербургский Национальный Исследовательский Университет ИТМО

Факультет инфокоммуникационных технологий

Лабораторная работа №3

Выполнила: Митилева Людмила Игоревна

Группа: K3320

Проверила: Марченко Елена Вадимовна

Санкт-Петербург

2024

**Цель работы:** настроить gulp: последовательное и параллельное выполнение заданий, отображение файлов проекта в браузере, автоматическая перезарузка. Создать форму для отправки обратной связи от пользователя сайта. Установить инструментарий для отладки проектов.

**Ход работы:**

Задание 1. Настройка Gulp

Для начала был инициализирован проект с помощью команды npm init. После этого в папке проекта появился файл package.json, который содержит информацию о проекте (рис.1).

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 1 – Содержимое файла package.json

Далее был создан файл gulpfile.js, в нем создаются два таска, функция gulp.series указывает на то, что они должны быть выполнены последовательно (рис.2).

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание

Рисунок 2 – Код для последовательного запуска заданий

С помощью команды gulp запускаем задания и видим, что они были выполнены последовательно (рис.3).

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 3 – Результат выполнения команды gulp

Для того, чтобы задания выполнялись параллельно, функция gulp.series была заменена на gulp.parallel (рис.4)

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 4 – Код для параллельного запуска заданий

При запуске команды gulp видно, что задания выполняются параллельно (рис.5).

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 5 – Результат выполнения команды gulp

Далее был создан таск browserSync, который отображает файлы проекта в браузере, отслеживает изменения в HTML-файлах и обновляет страницу браузера при сохранении изменений (рис.6).

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

Рисунок 6 – Создание таска browserSync

При выполнении команды gulp запускается сервер, который отслеживает изменения в HTML-файлах (рис.7).

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 7 – Результат выполнения команды gulp

Был создан и запущен файл index.html (рис.8).

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 8 – Страница браузера до изменения файла

После сохранения изменений в файле index.html страница браузера была автоматически перезагружена (рис.9).

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Рисунок 9 – Страница браузера после сохранения изменений

Задание 2. Создание формы обратной связи

Метод GET — это способ отправки данных из формы на сервер через URL. Все данные, которые передаются методом GET, включаются непосредственно в строку URL, которая отображается в адресной строке браузера. Этот метод используется для запроса информации с сервера, и данные передаются в виде параметров строки запроса.

Метод POST — это способ отправки данных из формы на сервер через тело HTTP-запроса. В отличие от метода GET, данные, отправленные с помощью POST, не видны в URL и передаются в теле запроса.

Была создана форма обратной связи о музыке с использованием радиокнопок и чекбоксов.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок10 – Форма обратной связи

Изображение выглядит как текст, электроника, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 11 – Код для формы

При нажатии на кнопку отзыв записывается в текстовый файл с помощью файла process\_feedback.php.

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Автоматически созданное описание

Рисунок 12 – Код файла process\_feedback.php

Если данные успешно записаны в файл, пользователя перекидывает на страницу с благодарностью за заполнение формы.

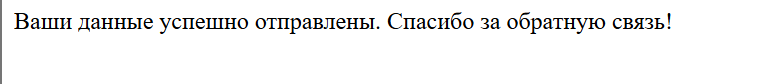


Рисунок 13 – Страница после заполнения формы

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

Рисунок 14 – Данные, записанный в текстовый файл

Задание 3. Настройка портала http://test.site

Для отладки проекта был выбран XAMPP, после установки были включены Apache и MySQL:

**Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание**

Рисунок 15 – XAMPP

Была настроена база данных и добавлен пользователь со всеми правами

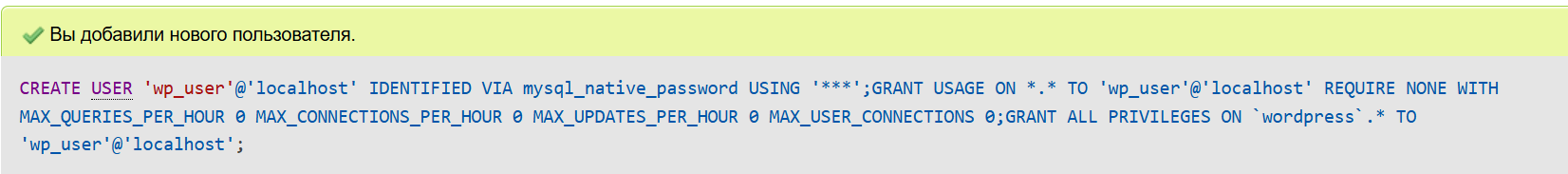


Рисунок 16 - phpMyAdmin

Были скачен Wordpress и настроена конфигурация

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 17 – Измененная конфигурация WordPress

Далее были настроен виртуальный хост Apache и файл hosts

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 18 – Изменение конфигурации для сайта

Изображение выглядит как текст, Шрифт, снимок экрана, Графика

Автоматически созданное описание

Рисунок 19 – Изменение файла hosts

После этого по URL-адресу: <http://test.site/> получилось увидеть следующую веб-страницу:

Изображение выглядит как текст, снимок экрана, Веб-сайт, веб-страница

Автоматически созданное описание

Рисунок 21 – Созданный портал с темой

**Вывод**: В ходе работы получилось поработать с GULP, создать форму для отправки информации по обратной связи от пользователя сайта и установить инструментарий для отладки проектов.