Санкт-Петербургский Национальный Исследовательский Университет

Информационных Технологий, Механики и Оптики

Факультет Инфокоммуникационных Технологий

**Web-программирование**

**Лабораторная работа 3**

Выполнил

Федоров В.Д.

Проверила

Марченко Е. В.

Санкт-Петербург, 2024

**Цель:** научиться работать с gulp, создать форму для получения информации от клиента и научиться работать с Wordpress.

**Задачи:**

* Научиться работать с gulp – создать таски, настроить параллельно и последовательно выполнение, а также настроить автоматическую перезагрузку при изменении файлов проекта
* Создать форму для приема обратной связи от клиента
* Установить движок Wordpress и запустить сайт

**Задание 1**

Для выполнения данного задания был создан gulpfile с двумя тасками: taskOne, который показывает, что выполняется первый таск и takTwo, который показывает, что выполняется второй таск. Потом были добавлены функции html и css, которые автоматически перезагружаются при имениях в этих файлах

Изображение выглядит как текст, снимок экрана, Шрифт, меню

Автоматически созданное описание

Рисунок 1 – gulpfile.js

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 2 – gulpfile.js

Далее были написаны простые index.html, styles.css:

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 3 – index.html

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 4 – styles.css

Результат параллельного запуска:

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 5 – параллельный запуск

Результат последовательного запуска:

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 6– последовательный запуск

Потом был запущен browser-sync:

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 7 – запуск gulp с browser-sync

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

Автоматически созданное описание**

Рисунок 8 – результат запуска gulp с browser-sync

Как видно, при внесении изменений страница автоматически перезагрузилась и изменения отобразились:

\ Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Рисунок 9 – изменение файлов

Изображение выглядит как снимок экрана, текст, программное обеспечение, Операционная система

Автоматически созданное описание

Рисунок 10 – автоматическое обновление

**Задание 2**

Сначала была написана простая форма, где можно отправлять и GET, и POST запросы:

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 11 – код feedback.html

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 12 – код feedback.html

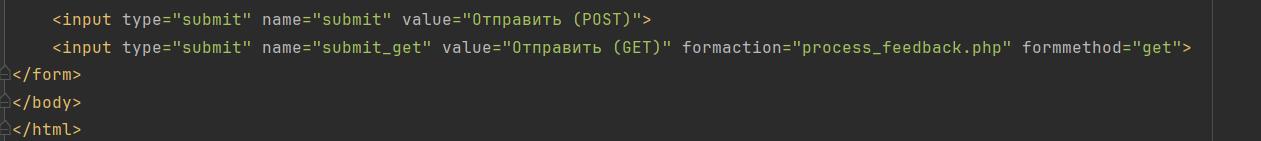


Рисунок 13 – код feedback.html

Потом был написан php файл для обработки и POST, и GET запросов:

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 14 – код process\_feedback

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 15 – код process\_feedback

Потом был запущен сервер:

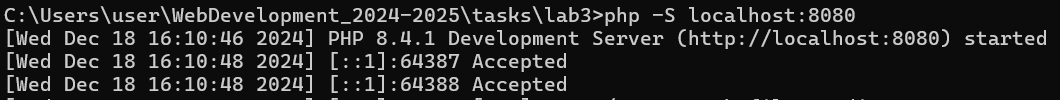


Рисунок 16 – запуск сервера

Заполнение формы:

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

Автоматически созданное описание

Рисунок 17 – feedback.html

Результат отправки POST:

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

Автоматически созданное описание

Рисунок 17 – process\_feedback.php

Результат отправки GET:

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

Автоматически созданное описание

Рисунок 18 – process\_feedback.php

Метод GET передает данные в URL, подходит, когда данные не конфиденциальны:

****

Рисунок 19 – URL

Метод POST передает данные в теле запроса, они не видны в URL, что позволяет передавать конфиденциальную информацию:

****

Рисунок 20 – URL

**Задание 3**

Для установки WordPress был использован XAMPP. Сначала были скачены движок WordPress и XAMPP.

Потом была добавлена база данных wordpress:

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 21 – test\_db

Потом был выполнен переход на <http://localhost/wordpress/> и была выполнена настройка WordPress:

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Рисунок 22 – wordpress

Чтобы был доступ через <http://test.site> необходимо добавить следующую запись в файл hosts:



Рисунок 23 – файл hosts

Потом был добавлен такой виртуальный хост в Apache:

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 24 – файл httpd-vhosts.conf

Сайт до изменения домена.

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 25 – wordpress

В итоге получаем:

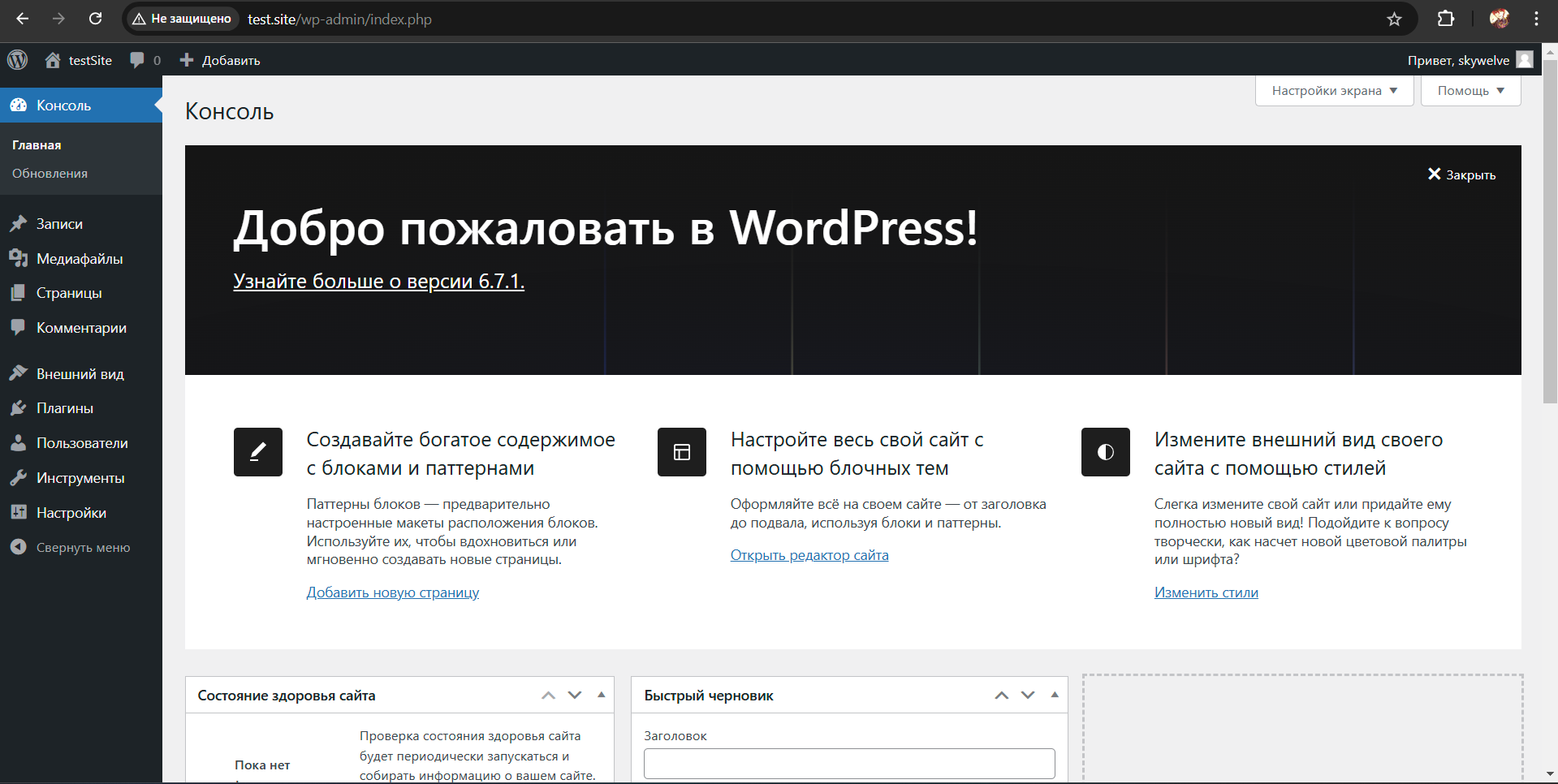


Рисунок 26 – test.site

**Вывод:** был изучен gulp, была создана форма обратной связи и был создан сайт на Wordpress.