Санкт-Петербургский Национальный Исследовательский Университет

Информационных Технологий, Механики и Оптики

Факультет Инфокоммуникационных Технологий

**Web-программирование**

**Лабораторная работа 3**

Выполнил

Бордюг В. Ю.

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: styles для компиляции scss в css и scripts для объединения js файлов в один main.js, потом были созданы таски для выполнения их в последовательном (build-sequential) и параллельном режимах (build-parallel). Потом был создан task watch для отслеживания изменений и перезагрузки браузера.

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

Автоматически созданное описание

Рисунок 1 – gulpfile.js

Изображение выглядит как текст, снимок экрана, линия, Шрифт

Автоматически созданное описание

Рисунок 2 – gulpfile.js

Далее были написаны простые index.html, style.scss и file.js:

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

Автоматически созданное описание

Рисунок 3 – index.html

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

Автоматически созданное описание

Рисунок 4 – style.scss

Изображение выглядит как текст, программное обеспечение, снимок экрана, Значок на компьютере

Автоматически созданное описание

Рисунок 5 – file.js

Результат параллельного запуска:

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 6 – параллельный запуск

Результат последовательного запуска:

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 7 – последовательный запуск

В файлы проекта успешно добавились style.css и main.js:

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

Автоматически созданное описание

Рисунок 8 – папка проекта

Потом был запущен browser-sync:

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 9 – запуск gulp с browser-sync

**Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание**

Рисунок 10 – результат запуска gulp с browser-sync

Как видно, при внесении изменений страница автоматически перезагрузилась и изменения отобразились (текст «Gulp!»):

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок 11 – изменения в index.html

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рисунок 12 – 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, Операционная система

Автоматически созданное описание

Рисунок 13 – код contact\_form.html

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

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

Автоматически созданное описание

Рисунок 14 – process\_form.php

Потом был запущен сервер:

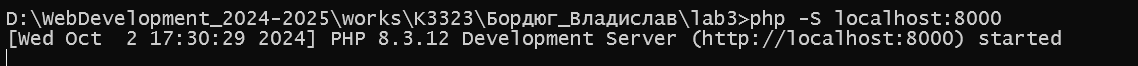


Рисунок 15 – запуск сервера

При переходе открывается форма:

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

Автоматически созданное описание

Рисунок 16 – contact\_form.html

Заполнение формы:

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок 17 – contact\_form.html

Результат отправки POST:

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

Автоматически созданное описание

Рисунок 17 – process\_form.php

Результат отправки GET:

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

Автоматически созданное описание

Рисунок 18 – process\_form.php

Метод GET передает данные в URL, подходит, когда данные не конфиденциальны, в данном случае такой URL получается:

****

Рисунок 19 – URL

****

Рисунок 20 – URL

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

****

Рисунок 21 – URL

**Задание 3**

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

Потом была добавлена test\_db:

Изображение выглядит как программное обеспечение, текст, Значок на компьютере, веб-страница

Автоматически созданное описание

Рисунок 22 – test\_db

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

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рисунок 23 – wordpress

Чтобы был доступ через <http://test.site> сначала была добавлена такая запись в файл hosts:

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок 24 – файл hosts

Потом был добавлен такой виртуальный хост в Apache:

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

Автоматически созданное описание

Рисунок 25 – файл httpd-vhosts.conf

Потом в настройках wordpress адреса были изменены на <http://test.site> :

Изображение выглядит как текст, программное обеспечение, Значок на компьютере, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 26 – изменение ссылок в настройках wordpress

В итоге при вводе <http://test.site> отображается сайт:

Изображение выглядит как снимок экрана, текст, программное обеспечение, веб-страница

Автоматически созданное описание

Рисунок 27 – переход на <http://test.site>

Была установлена и активирована тема Neve:

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

Автоматически созданное описание

Рисунок 28 – установка темы Neve

Получился такой внешний вид:

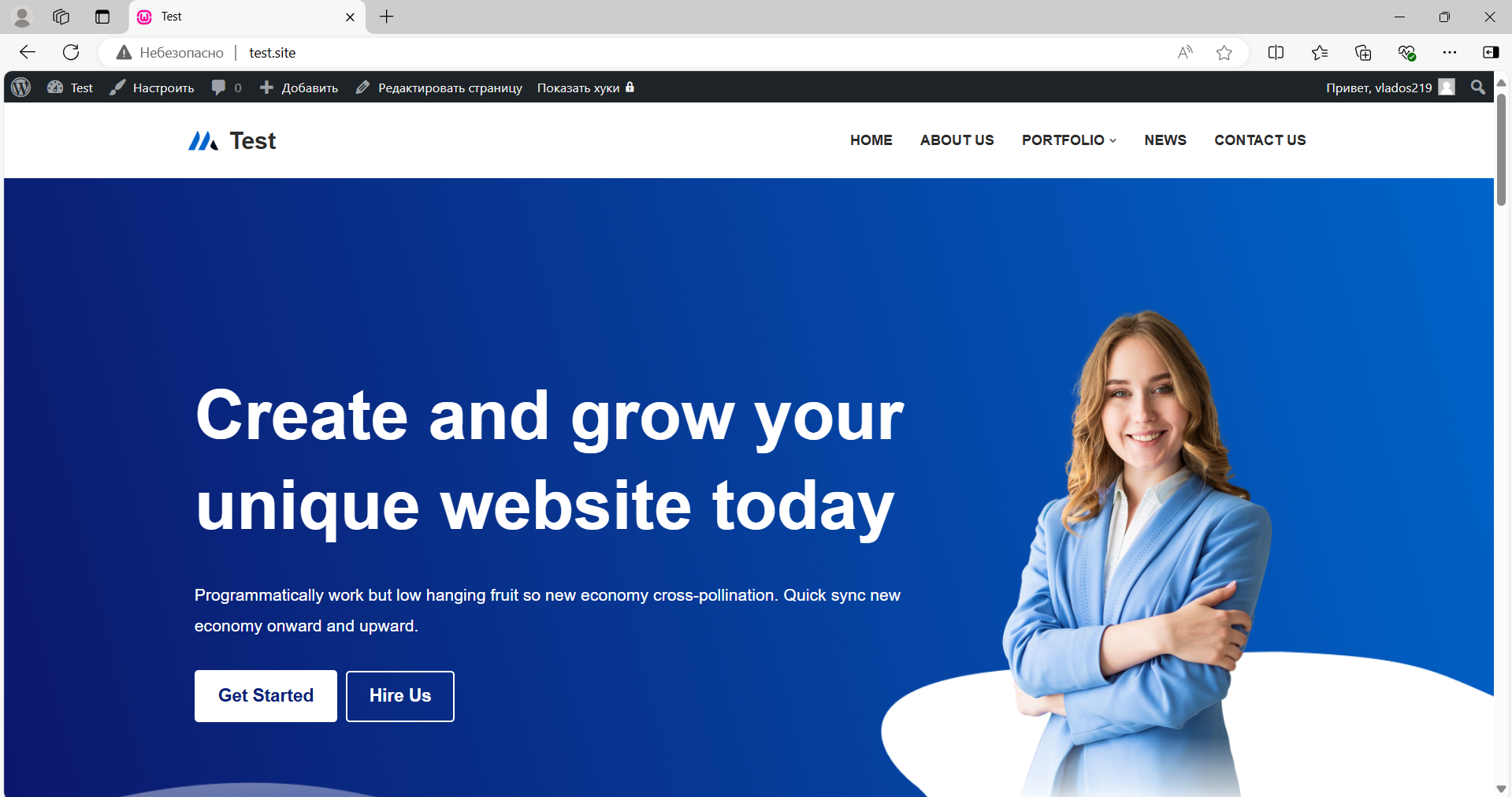


Рисунок 29 – <http://test.site>

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