МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

ITMO University

ФАКУЛЬТЕТ ИНФОКОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ

Отчет по лабораторной работе №3

по дисциплине “Web-программирование”

Выполнил: студент

группы K3323  
 Влазнев Д. В.

Преподаватель:

Марченко Е. В.

г. Санкт-Петербург,

2024

[Введение 3](#_Toc182069154)

[1. Настройка системы автоматизации сборки 4](#_Toc182069155)

[2. Создание формы обратной связи 5](#_Toc182069156)

[3. Настройка WordPress 7](#_Toc182069157)

[Вывод 10](#_Toc182069158)

# Введение

**Цель работы:** изучить основы автоматизации сборки и настройки сайта.

**Задачи:**

1. Настроить систему автоматизации сборки с помощью Gulp.
2. Создать форму обратной связи с пользователем.
3. Настроить локальный сервер и CMS WordPress,

# 1. Настройка системы автоматизации сборки

Выполним инициализацию папки с помощью команды npm init --yes. Далее установим необходимые пакеты (См. Рисунок 1).

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 1 – Необходимые пакеты

В корне проекта создадим и настроим gulpfile.js (См. Рисунок 2–3).



Рисунок 2 – Gulpfile.js

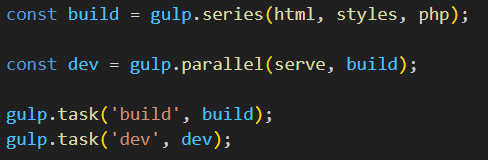


Рисунок 3 – Gulpfile.js

Теперь запустим таски и посмотрим на результат (См. Рисунок 4–5).

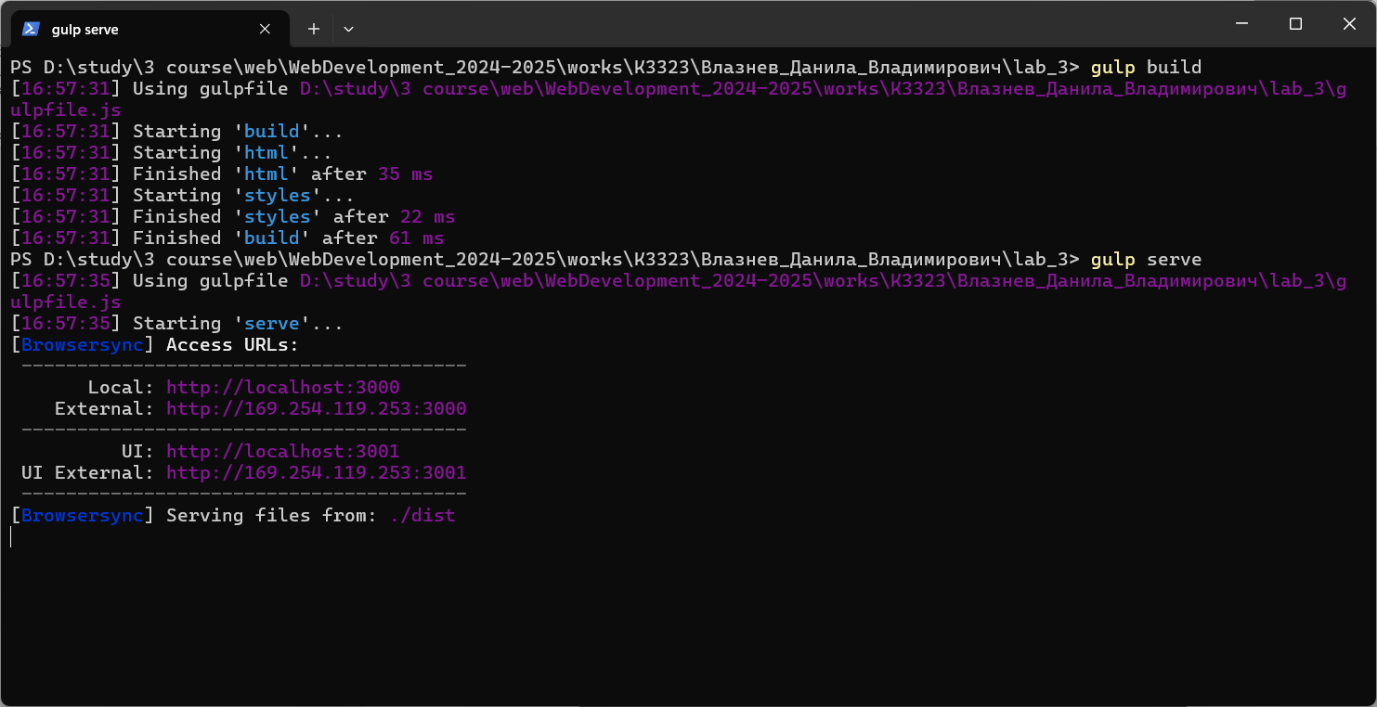


Рисунок 4 – Результат

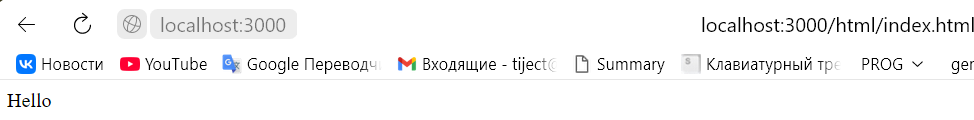


Рисунок 5 – Результат

# 2. Создание формы обратной связи

Была создана форма обратной связи в соответствии с заданием (См. Рисунок 6–9).



Рисунок 6 – Код формы

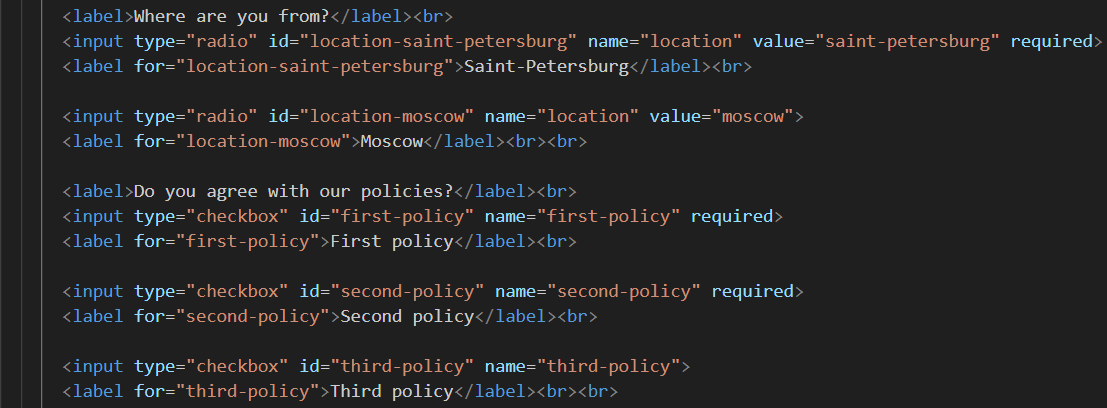


Рисунок 7 – Код формы

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Рисунок 8 – Код формы

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 9 – Форма

Также был создан простой php скрипт, который выводит полученную из формы информацию на страницу (См. Рисунок 10–11).

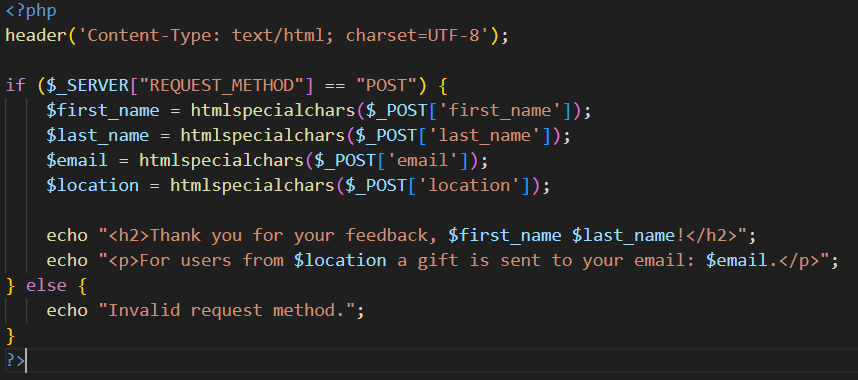


Рисунок 10 – PHP код

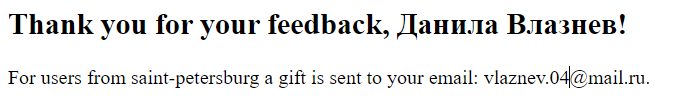


Рисунок 11 – Ответ формы

Существует два основных способа отправки формы: с помощью метода GET И POST. Метод GET конвертирует поля формы в URL параметры, так что их можно увидеть в адресной строке, что не всегда является безопасным решением. Метод POST отправляет данные в теле запроса, что не позволяет их увидеть в URL.

# 3. Настройка WordPress

Для начала необходимо установить локальный сервер. Я использовал XAMPP. (Cross platform, Apache, MariaDB, PHP).

С помощью инструмента PhpMyAdmin была создана база данных для WordPress (См. Рисунок 12).

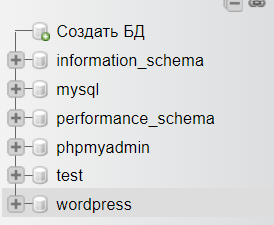


Рисунок 12 – База данных

Был скачан архив WordPress и настроен файл конфигурации, а именно подключение к базе данных (См. Рисунок 13).

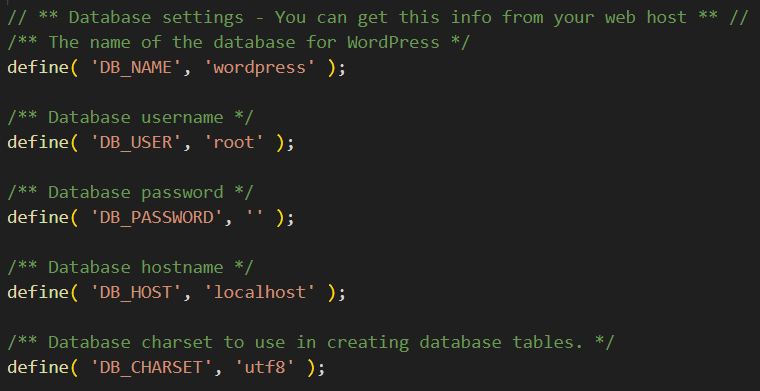


Рисунок 13– Конфигурация

Далее запустив сервер и перейдя в нужную папку, был настроен и запущен WordPress (См. Рисунок 14–15).

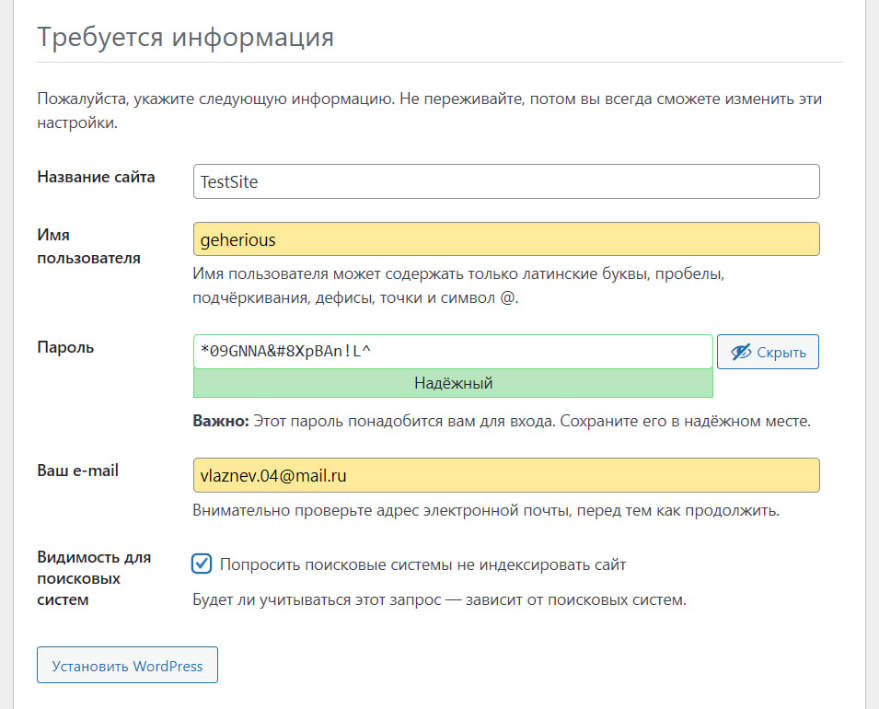


Рисунок 14– Настройка

Изображение выглядит как текст, снимок экрана, Веб-сайт, дизайн

Автоматически созданное описание

Рисунок 15 – WordPress

Чтобы портал отображался по адресу test.site, необходимо задать виртуальный хост в конфигурации Apache, изменить файл hosts на компьютере, а также изменить настройки WordPress (См. Рисунок 16–18).

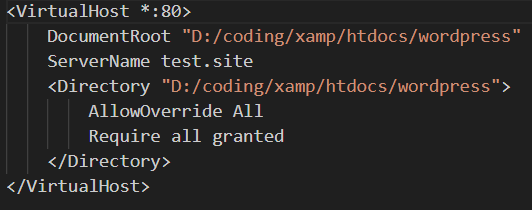


Рисунок 16– Настройка



Рисунок 17 – WordPress

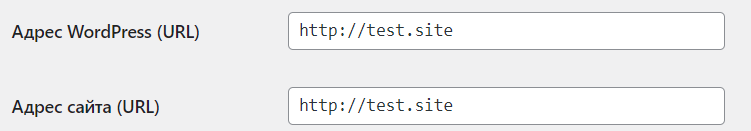


Рисунок 18 – WordPress

Тогда сайт отобразится на нужном домене (См. Рисунок 19).

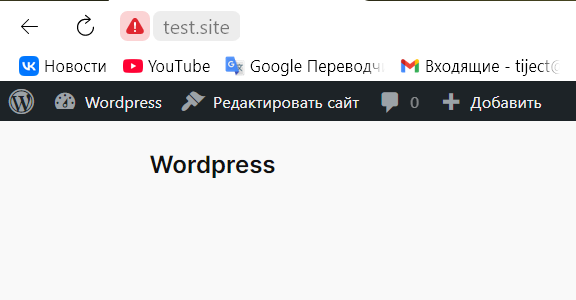


Рисунок 19 – Сайт

# Вывод

Я научился создавать последовательные и параллельные задания для gulp, что позволило автоматически обновлять разрабатываемый сайт при изменении кода. Также я создал базовый скрипт на PHP и настроил портал на WordPress.