Санкт-Петербургский Национальный Исследовательский Университет

Информационных Технологий, Механики и Оптики

Факультет Инфокоммуникационных Технологий

**Web-программирование**

**Лабораторная работа 3**

Выполнил

Затикян С. А.

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 с использованием методов GET и POST
* Установить движок Wordpress и запустить сайт

**Задание 1**

Для выполнения данного задания был создан gulpfile с двумя тасками: copyHTML для копирования HTML файлов в папку dist и minifyCSS для минификации CSS, потом были созданы таски для выполнения их в последовательном (build) и параллельном режимах (parallel). Потом был создан таск serve для запуска сервера с отслеживанием изменений.

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 1 – gulpfile.js

Далее были написаны простые index.html и style.css.

Результаты запуска тасок:

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рисунок 2 – запуск тасок

Также был запущен локальный сервер с нашей страничкой:

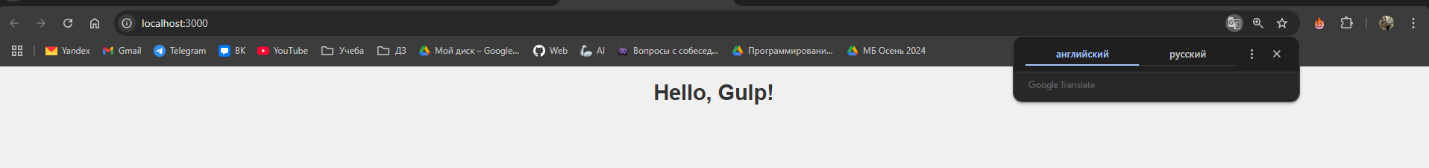


Рисунок 4 – сервер, запущенный с помощью gulp

При внесении изменений в index.html или style.css страница автоматически обновляется

**Изображение выглядит как текст, Шрифт, снимок экрана, линия

Автоматически созданное описание**

Рисунок 5 – обновление страницы с помощью browsersync

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

Автоматически созданное описание

Рисунок 6 – обновленная страница

**Задание 2**

Был написан feedback.html – форма, в которой можно отправлять и GET, и POST запросы:

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

Автоматически созданное описание

Рисунок 7 – код feedback.html

Потом был написан feedback.php для обработки этих запросов:

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

Автоматически созданное описание

Рисунок 8 – код feedback.php

Запущен php сервер:

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рисунок 9 – запуск php сервера

При переходе на страницу мы видим нашу форму, заполним её:

Изображение выглядит как текст, снимок экрана, число

Автоматически созданное описание

Рисунок 10 – форма обратной связи

Результат отправки POST:

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 11 – feedback.php при отправке POST запроса

Результат отправки GET:

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 12 – feedback.php при отправке GET запроса

Разница между методами GET и POST:

GET использует URL для передачи данных в виде строки запроса (см. рисунок 12).

Подходит для запросов, которые не изменяют данные на сервере (например, поиск информации).

Данные видны в адресной строке и ограничены по длине (~2048 символов).

POST передаёт данные в теле запроса (невидимо для пользователя).

Используется для безопасной передачи данных, которые должны быть скрыты (например, логины, пароли, личная информация).

Нет ограничений по длине данных.

**Задание 3**

Для установки WordPress был выбран XAMPP.

Был скачан XAMPP и архив wordpress.

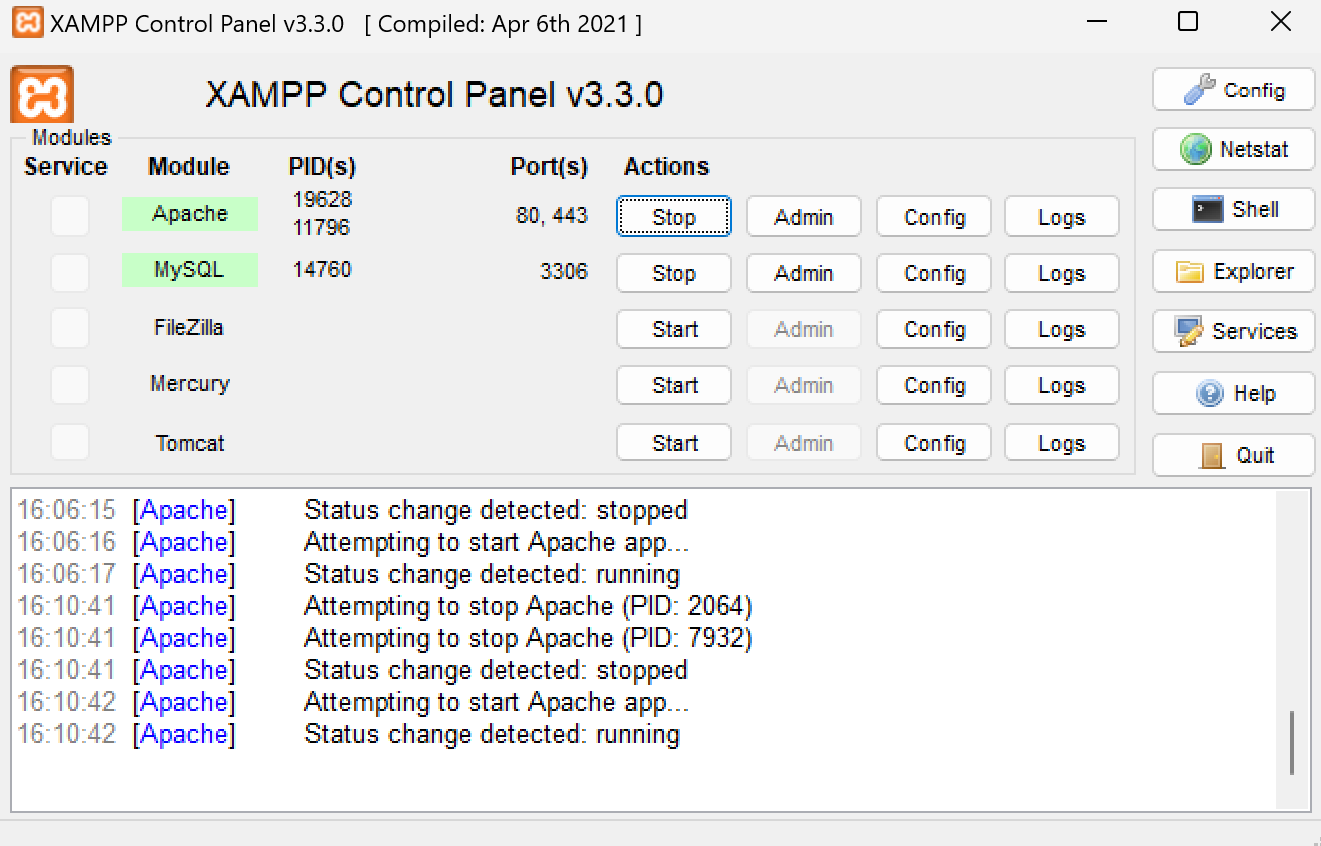


Рисунок 13 – панель управления XAMPP

Затем архив с wordpress был распакован в папку test.site в директории C:\xampp\htdocs. В файл hosts по адресу C:\Windows\System32\drivers\etc\hosts была добавлена строка “127.0.0.1 test.site”.

В файл с хостами XAMPP была добавлена информация о новом виртуальном хосте:

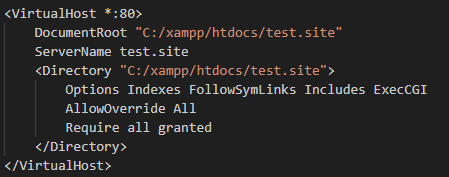


Рисунок 14 – добавление виртуального хоста

После перезапуска Apache в панели управления XAMPP был выполнен переход на test.site:

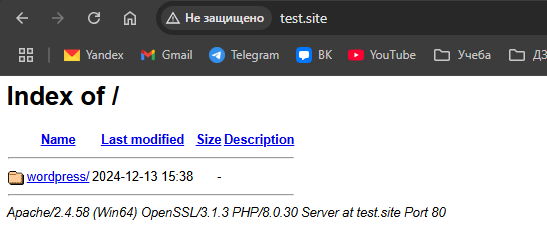


Рисунок 15 – test.site до настройки wordpress

После этого с помощью phpMyAdmin была создана новая база данных с названием wordpress:

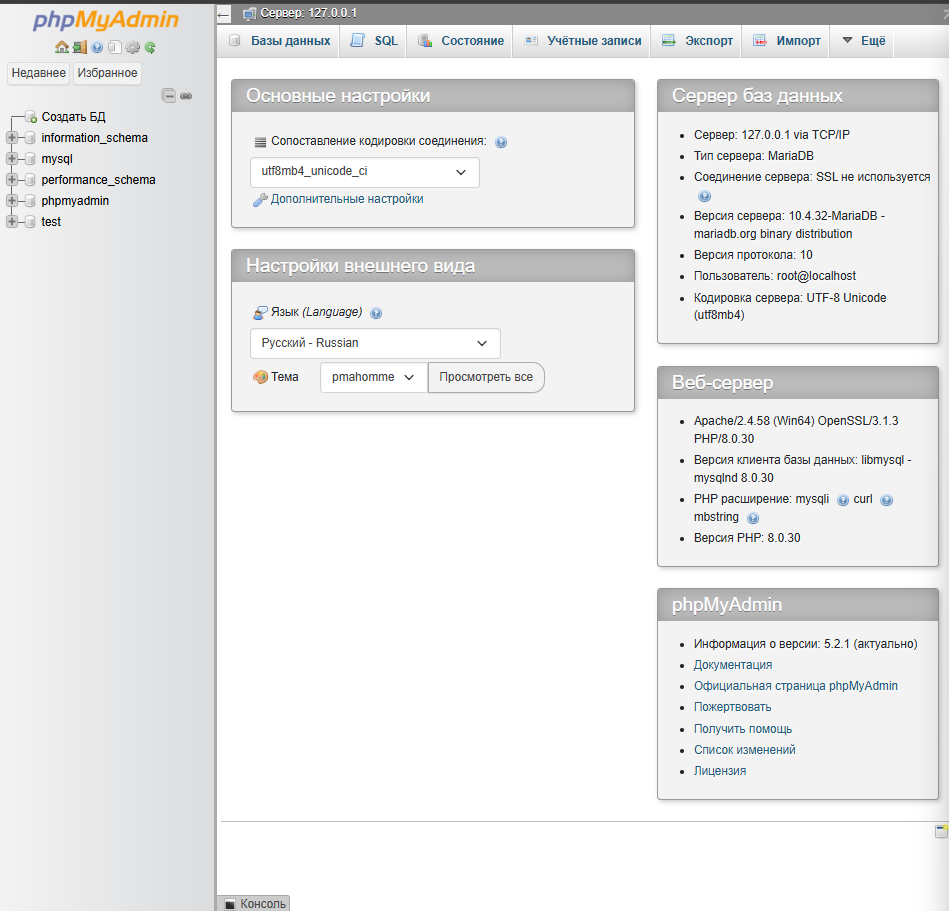


Рисунок 16 – создание БД в phpMyAdmin

Потом настройка wordpress продолжилась:

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

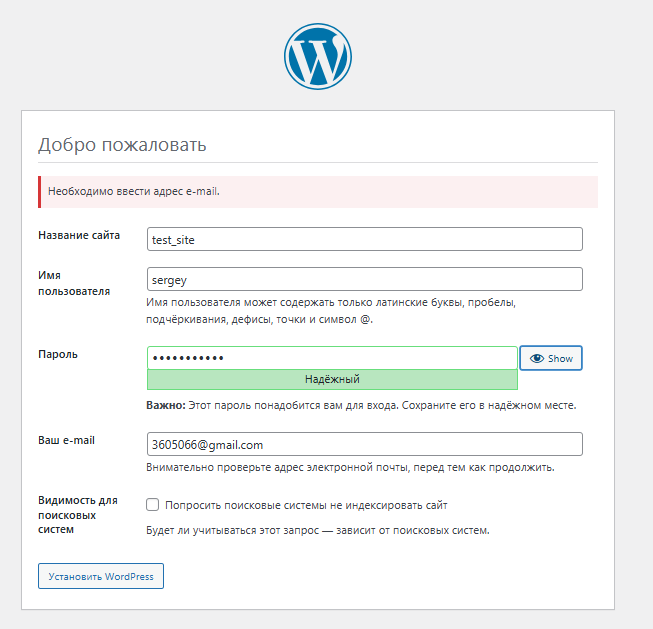


Рисунок 17 – настройка wordpress

После настройки был получен доступ к панели администратора:

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

Автоматически созданное описание

Рисунок 18 – панель управления wordpress

После установки темы сайт выглядит так:

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

Автоматически созданное описание

Рисунок 20 – test.site

**Вывод:** был изучен gulp, с его помощью созданы 2 таски и сервер с автоматическим обновлением, была создана форма обратной связи с использованием методов GET и POST и был создан сайт с использованием Wordpress.