**Санкт-Петербургский Национальный Исследовательский Университет Информационных технологий, механики и оптики**

**Отчет**

**Дисциплина: Web-программирование**

**Лабораторная работа 3.**

**Выполнил: Сарычев С.И.**

**Группа № K3321**

**Проверила: Марченко Е.В.**

**Санкт-Петербург**

**2024**

**Оглавление**

[Введение 3](#_Toc184782633)

[Ход работы 4](#_Toc184782634)

[1. Работа с Gulp. 4](#_Toc184782635)

[2. Создание формы. 7](#_Toc184782636)

[3. Установка движка. 14](#_Toc184782637)

[Вывод 18](#_Toc184782638)

# **Введение**

**Цель работы**: изучить инструменты gulp, wordpress, познакомиться с php, MAMP.

# **Ход работы**

1. **Работа с Gulp.**

В прошлой лабораторной работе уже было изучено как устанавливать gulp и инициализировать проект, поэтому просто повторим все это, но также установим и browser-sync, он поможет нам отслеживать изменения в файлах и автоматически отображать новые файлы в браузере (рисунок 1)

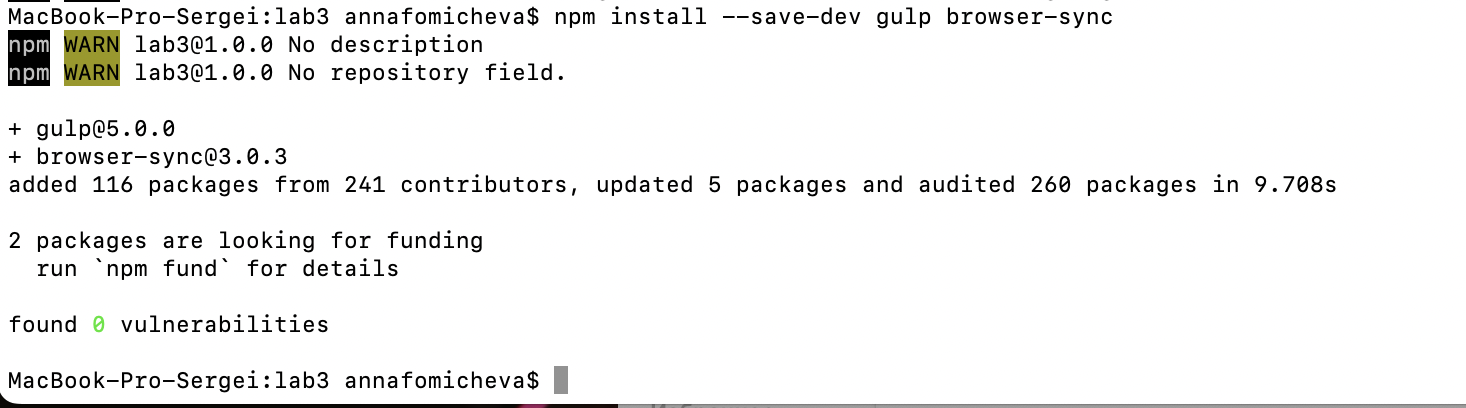


Рисунок 1 – Установка browser-sync

Далее добавим таски в Gulpfile, для работы с CSS, HTML, JS (рисунок 2).



Рисунок 2 – Добавление тасков

После этого протестируем последовательное и параллельное выполнение тасков, для последовательного выполнения используем gulp.series, а для параллельного gulp.parallel, разницу в выполнении можно отследить в консоли (при выполнении таска он выводит сообщение) результат представлен на рисунках 3 и 4.

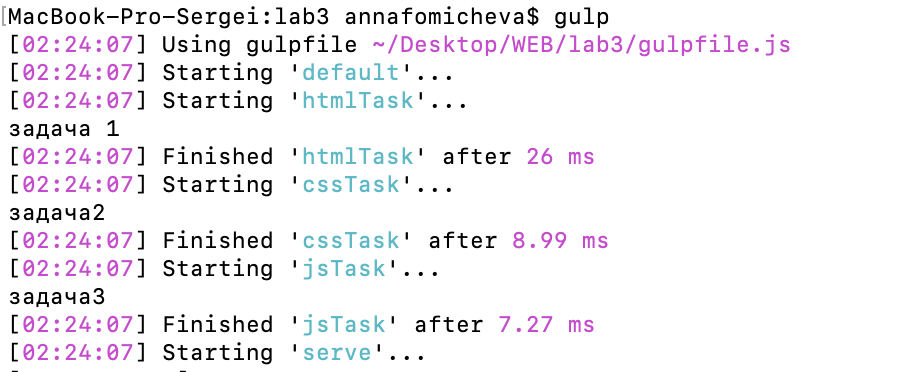


Рисунок 3 – Последовательное выполнение



Рисунок 4 – Параллельное выполнение

Далее по заданию реализуем таск для автоматической перезарузки при изменении одного из контролируемых файлов проекта, код таска представлен на рисунке 5.

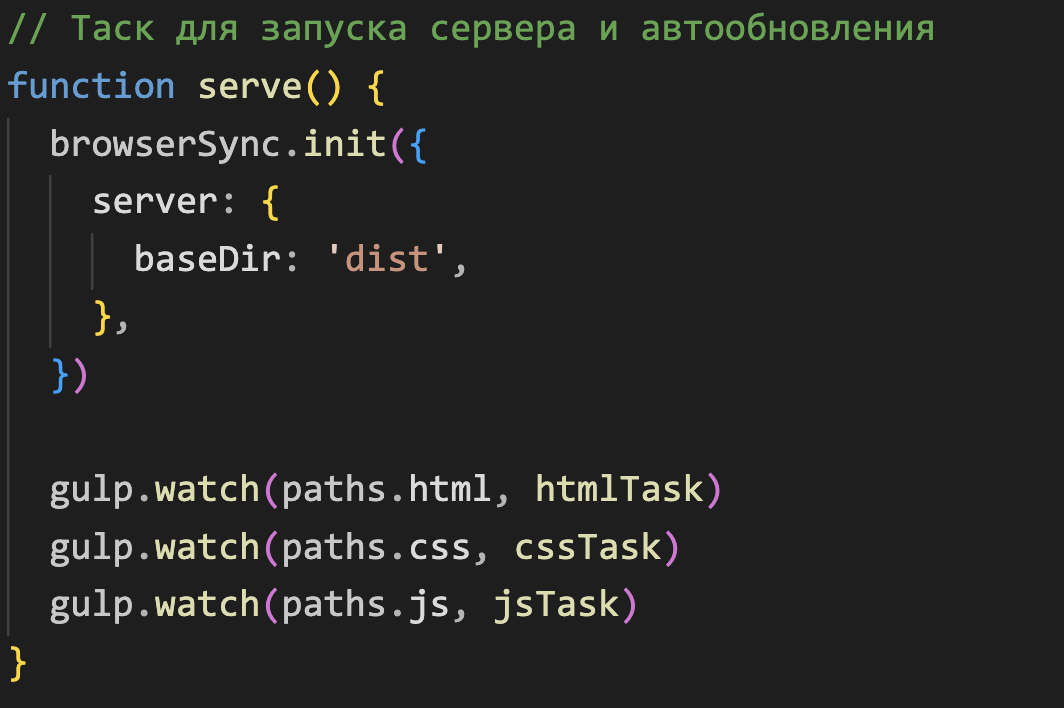


Рисунок 5 – Таск для автоматической перезагрузки

После проверим что при изменении css файла, цвет шрифта автоматически меняется на веб странице (рисунок 6)

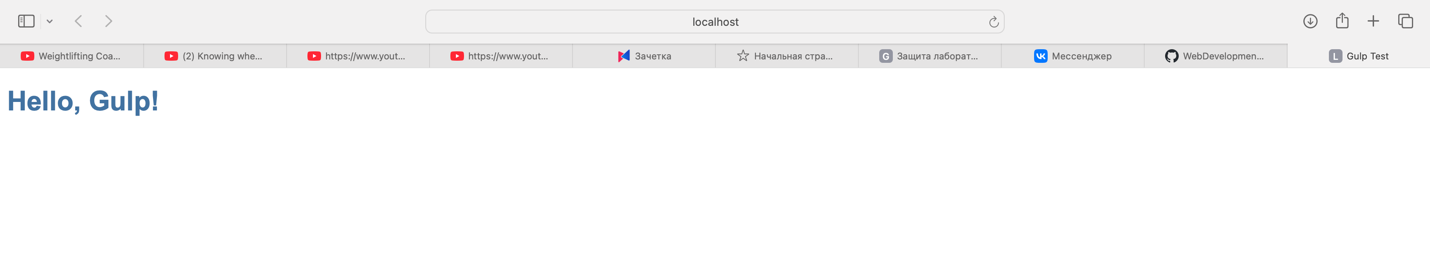


Рисунок 6 – Проверка автоматической перезагрузки

1. **Создание формы.**

В начале установим с помощью homebrew php (рисунок 7)

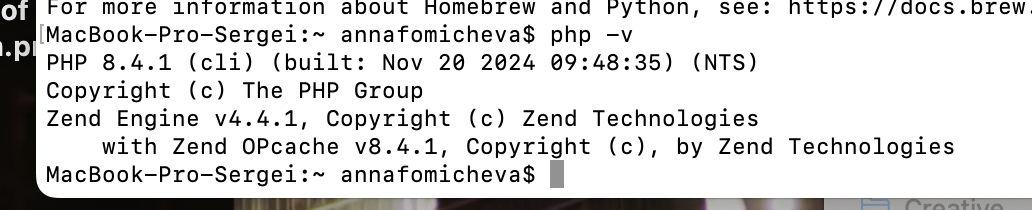
****

Рисунок 7 – Проверка версии php

Далее создаем файл index.html и заполняем его как на рисунке 8.

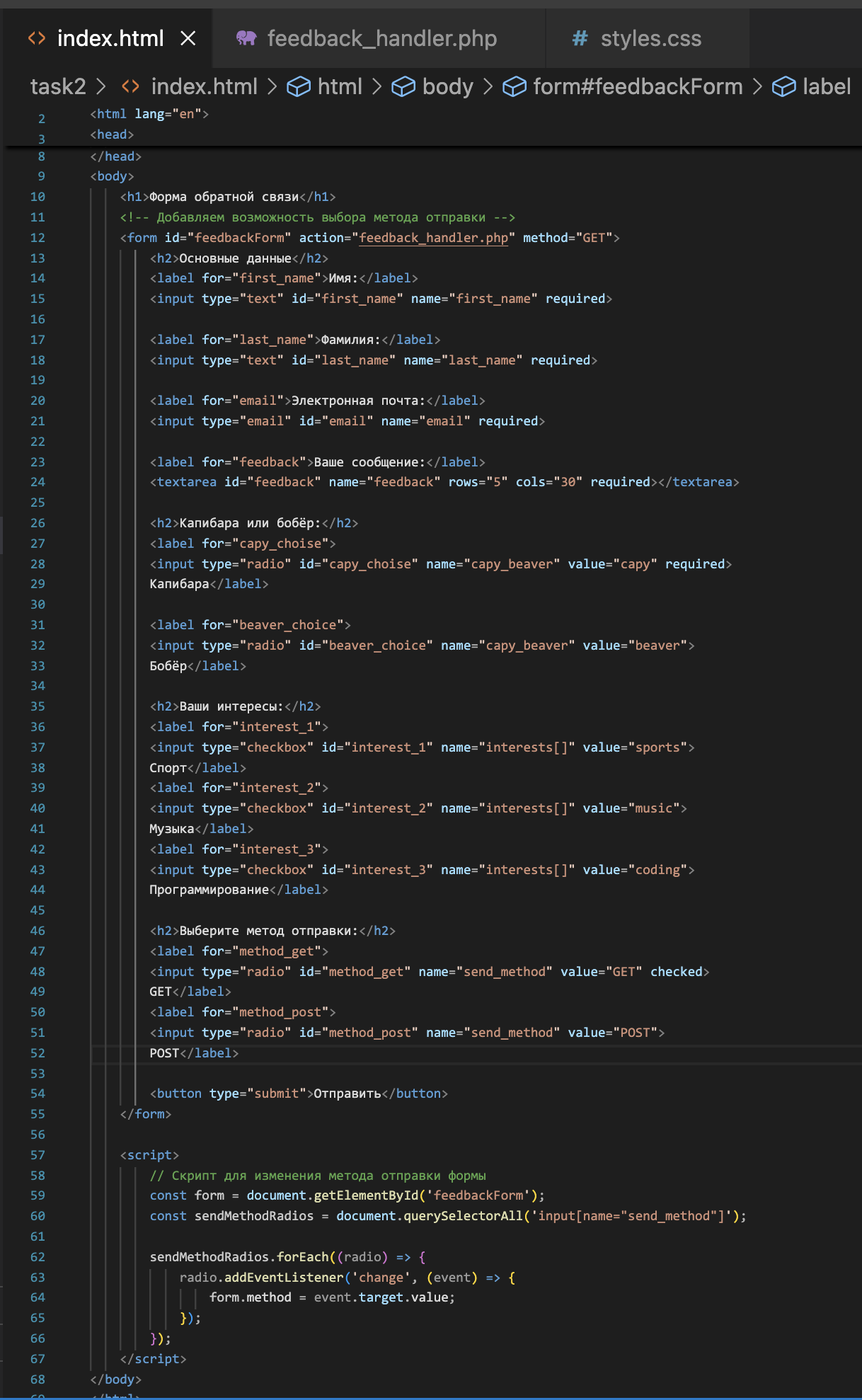
****

Рисунок 8 – index.html

После создадим styles.css (рисунок 9)



Рисунок 9 – styles.css

И наконец создадим feedback\_handler.php (рисунок 10)

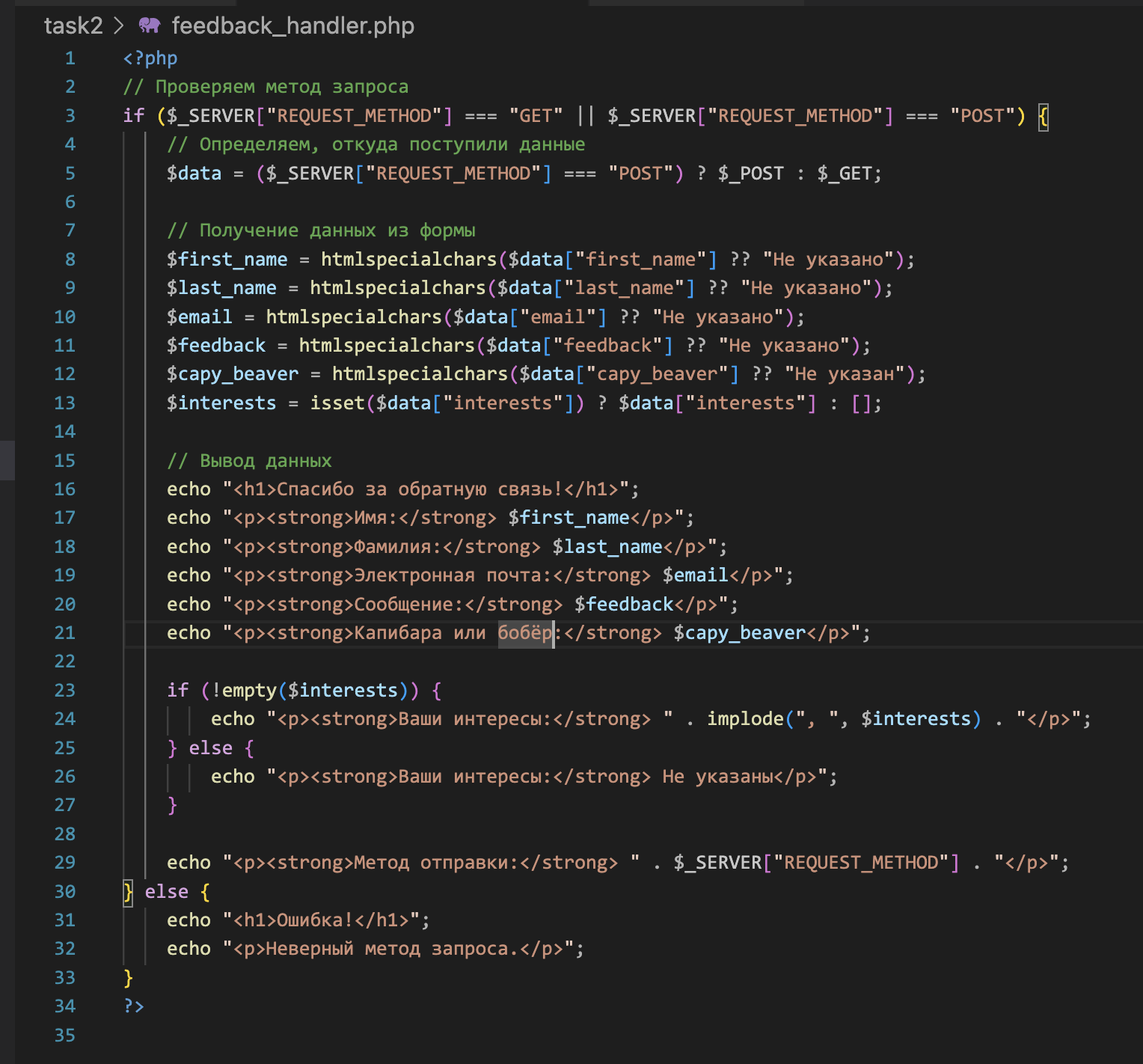


Рисунок 10 - feedback\_handler.php

Проверим работу веб страницы, создадим локальный сервер с помощью php -S localhost:8000, и перейдем по советующему адресу в браузере (рисунок 11)

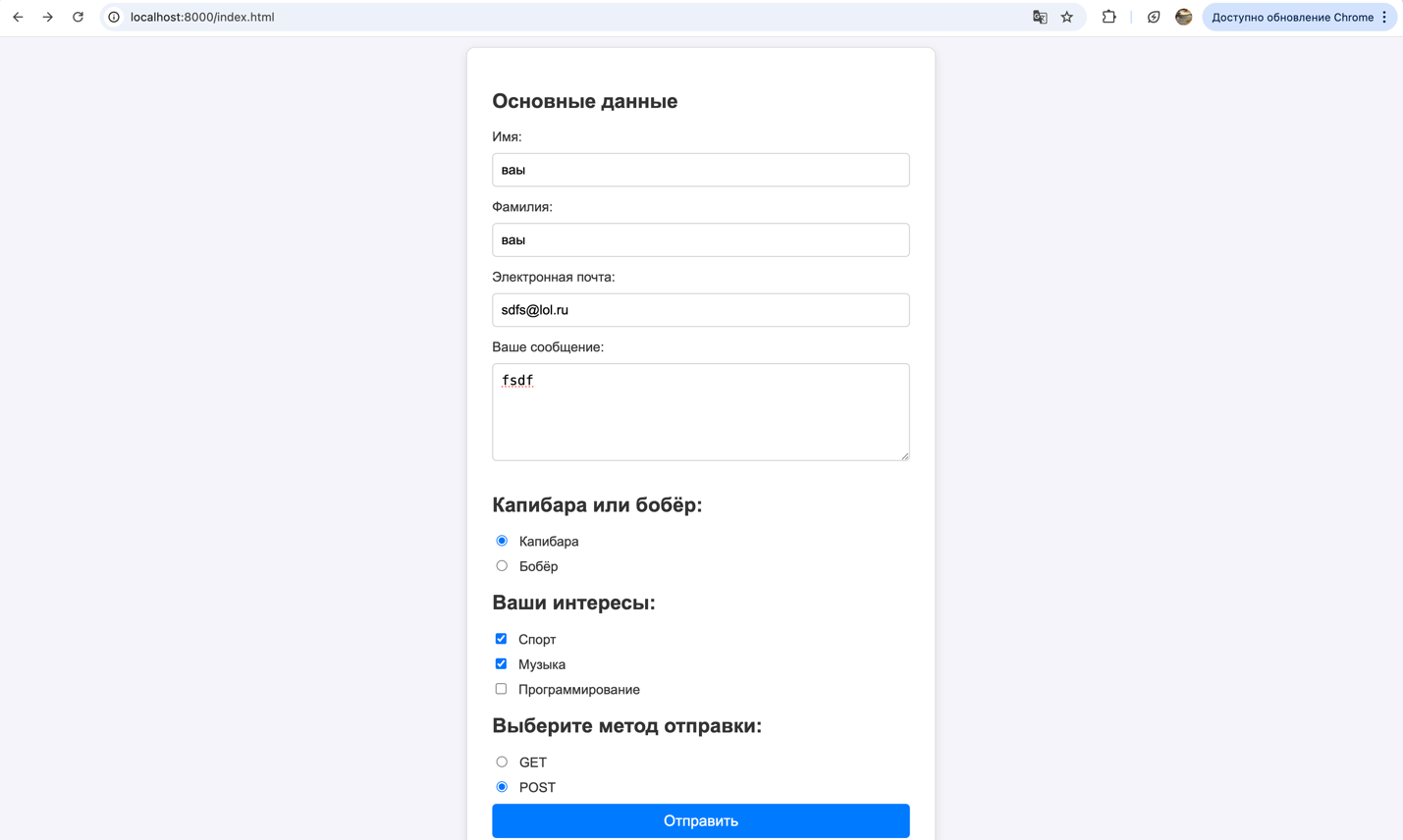


Рисунок 11 – Веб страница

Протестируем методы GET и POST, на рисунке 12 мы используем метод GET, а на рисунке 13 метод POST, это также можно увидеть в инструментах разработчика.

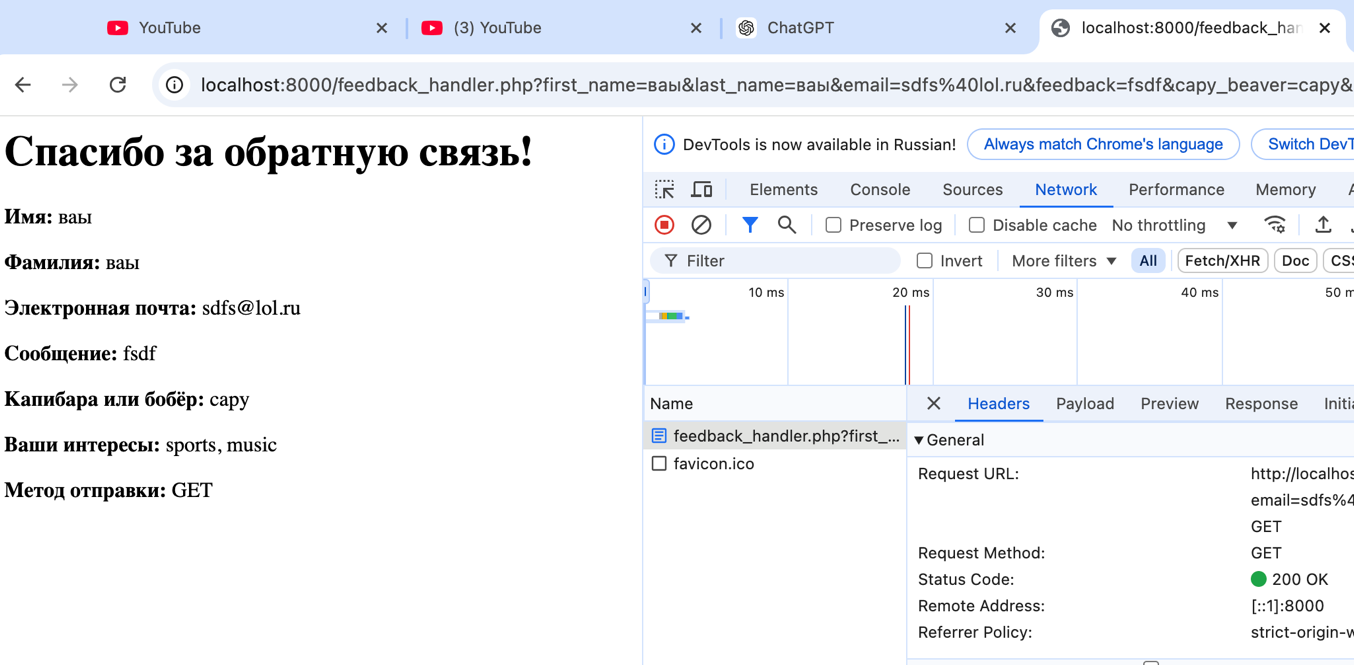


Рисунок 12 – Метод GET

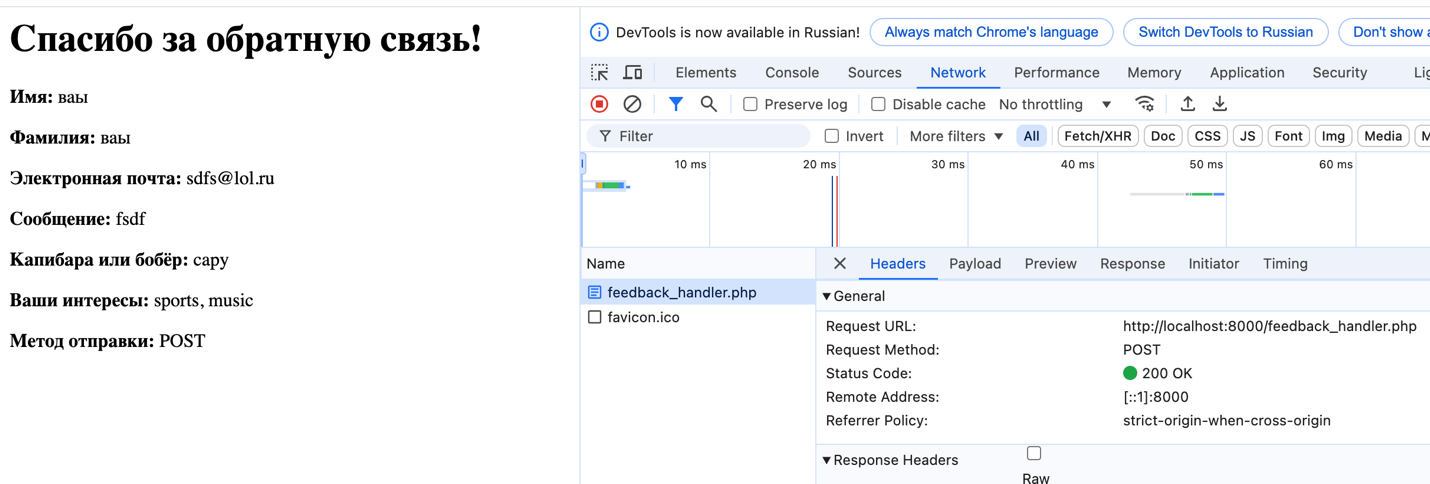


Рисунок 13 – Метод POST

Методы GET и POST используются в HTTP-протоколе для отправки данных с клиента на сервер. Оба метода являются основными средствами для взаимодействия между веб-страницами и сервером при отправке данных.

Метод GET отправляет данные в URL-строке запроса (в адресной строке браузера). Этот метод используется для получения информации с сервера, например, для запросов на поиск или при переходе по страницам.

**Особенности метода GET:**

* Данные в URL: Все данные передаются как часть URL в строке запроса (например: https://example.com?name=John&email=john@example.com).
* Ограничения на размер данных: Поскольку данные передаются через URL, существует ограничение на размер данных (обычно около 2000 символов, но это зависит от браузера и сервера).
* Безопасность: Данные, передаваемые через GET, видны в адресной строке и могут быть перехвачены, что делает метод менее безопасным для передачи конфиденциальной информации.
* Кэширование: Запросы, использующие GET, могут быть кэшированы браузером или прокси-серверами, что позволяет ускорить повторные запросы.
* Идемпотентность: GET-запросы считаются идемпотентными, что означает, что повторный запрос с теми же параметрами не должен изменять состояние на сервере.

Метод POST отправляет данные в теле запроса, что позволяет передавать более сложные и объемные данные. Этот метод чаще всего используется для отправки данных, например, при заполнении формы на сайте.

**Особенности метода POST:**

* Данные в теле запроса: Данные не отображаются в адресной строке браузера, что делает метод более безопасным для передачи конфиденциальной информации.
* Отсутствие ограничений по размеру данных: В отличие от GET, в POST-запросе нет таких строгих ограничений на размер передаваемых данных.
* Безопасность: Данные, передаваемые через POST, не видны в URL и передаются в теле запроса, что делает метод более безопасным для отправки чувствительной информации (например, паролей).
* Не кэшируется: Запросы POST не кэшируются браузером.
* Не идемпотентность: Запросы POST могут изменять состояние на сервере, например, добавлять новые записи в базу данных, поэтому их повторное выполнение может привести к различным результатам.

1. **Установка движка.**

Для начала необходимо скачать MAMP, так как лабораторная выполняется на MacOS, после этого необходимо также скачать wordpress с официального сайта и распаковать в папку MAMP/htdocs, далее запускаем MAMP и создаем локальный сервер (рисунок 14).

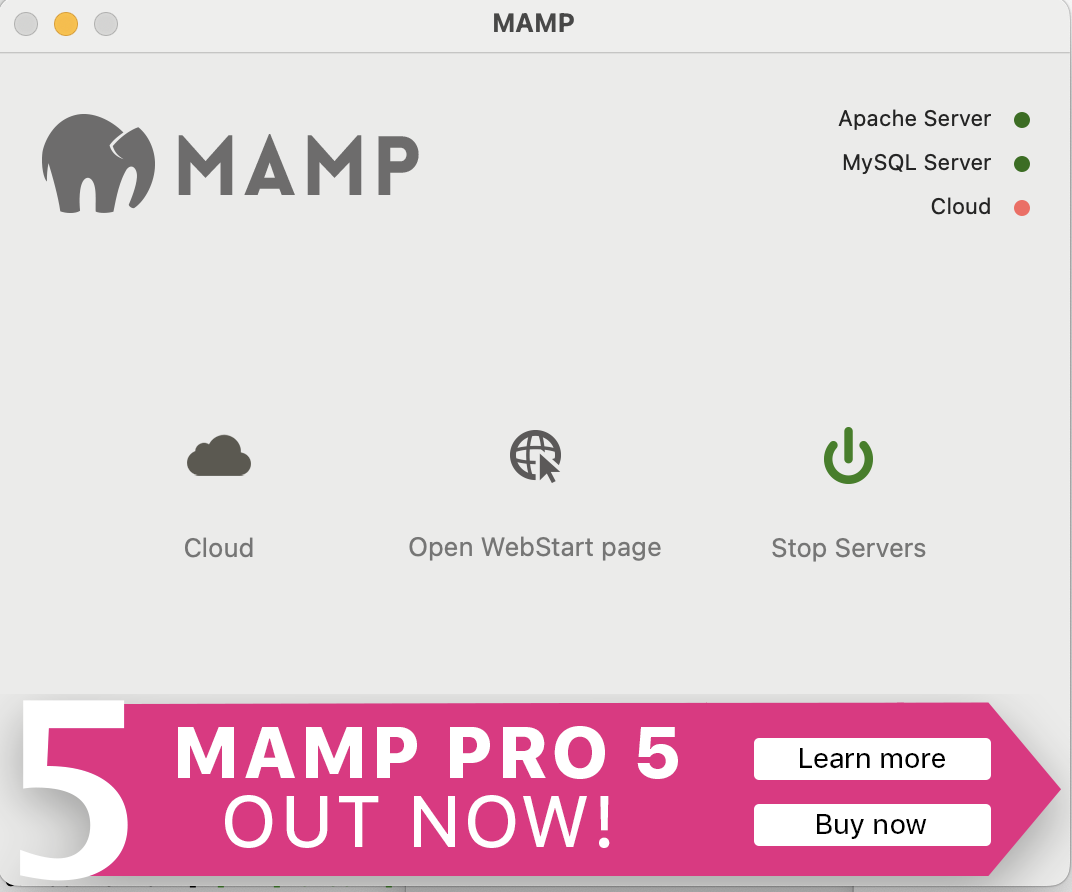


Рисунок 14 – MAMP

После этого создадим базу данных с помощью phpMyAdmin (рисунок 15).

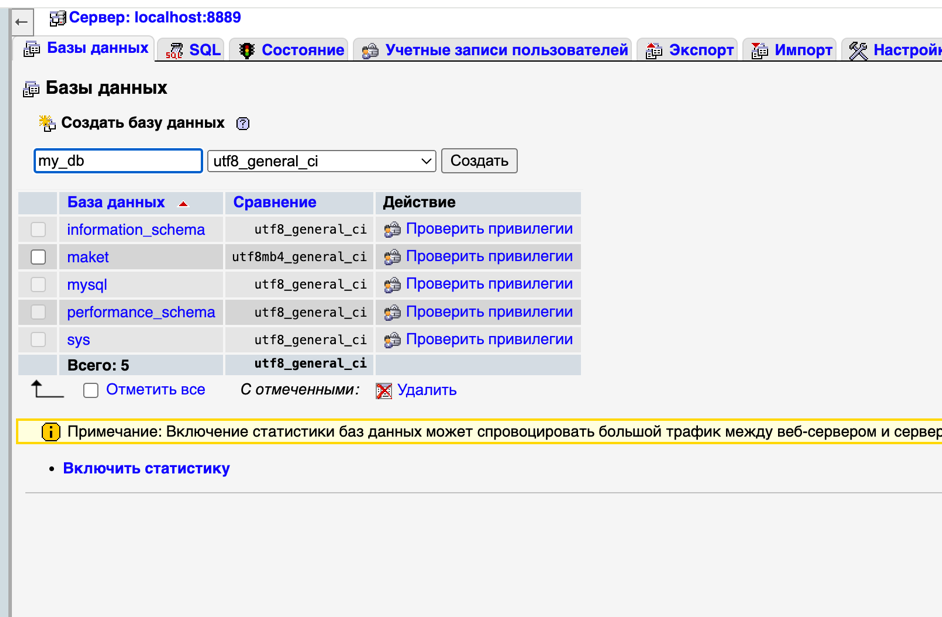


Рисунок 15 – Создание базы данных

Далее переходим по адресу <http://localhost:8888/wordpress>, и начинаем процесс установки WordPress (рисунок 16).

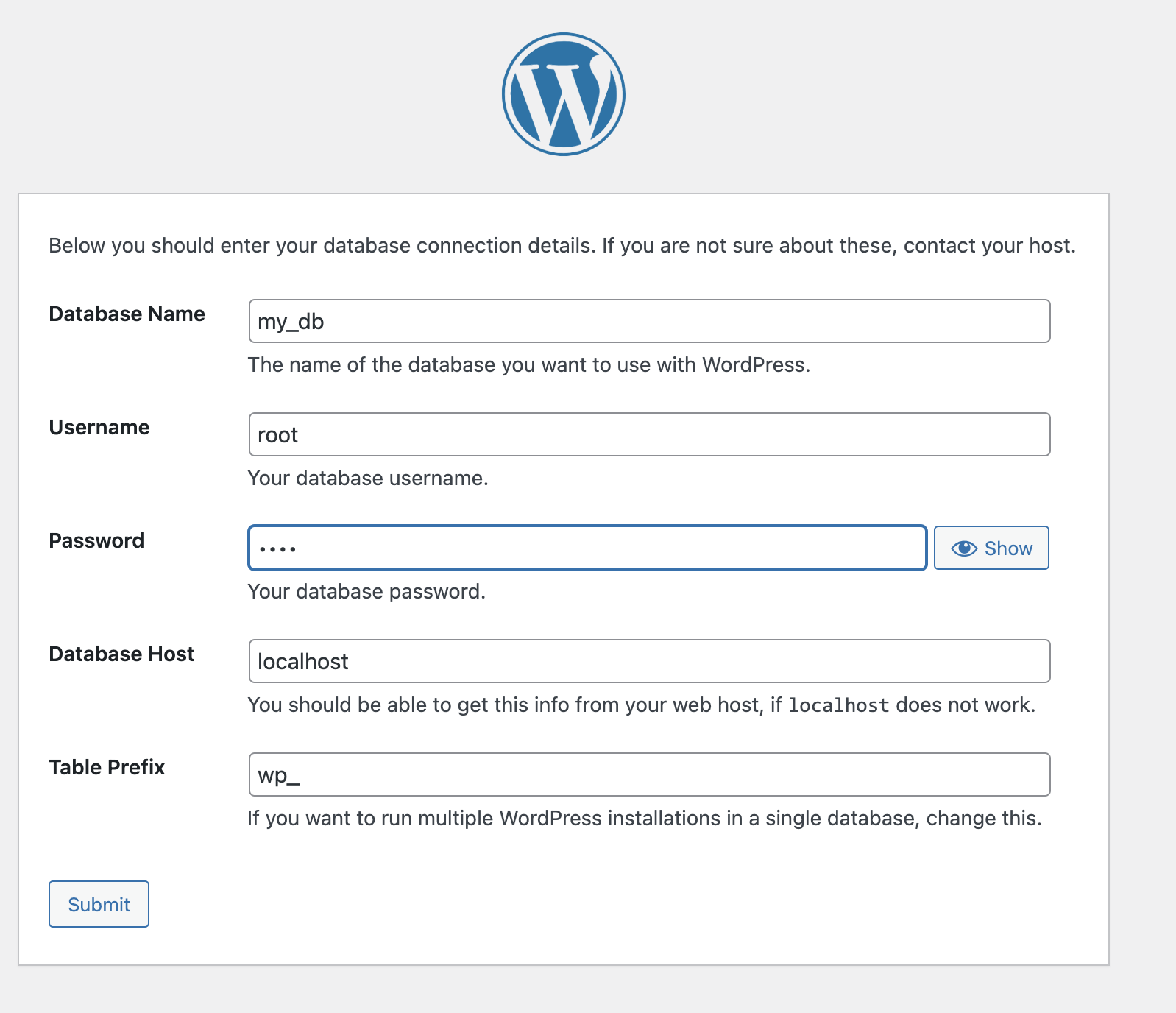


Рисунок 16 – Установка WordPress

После заполняем еще одну форму и попадаем в панель управления (рисунок 17)

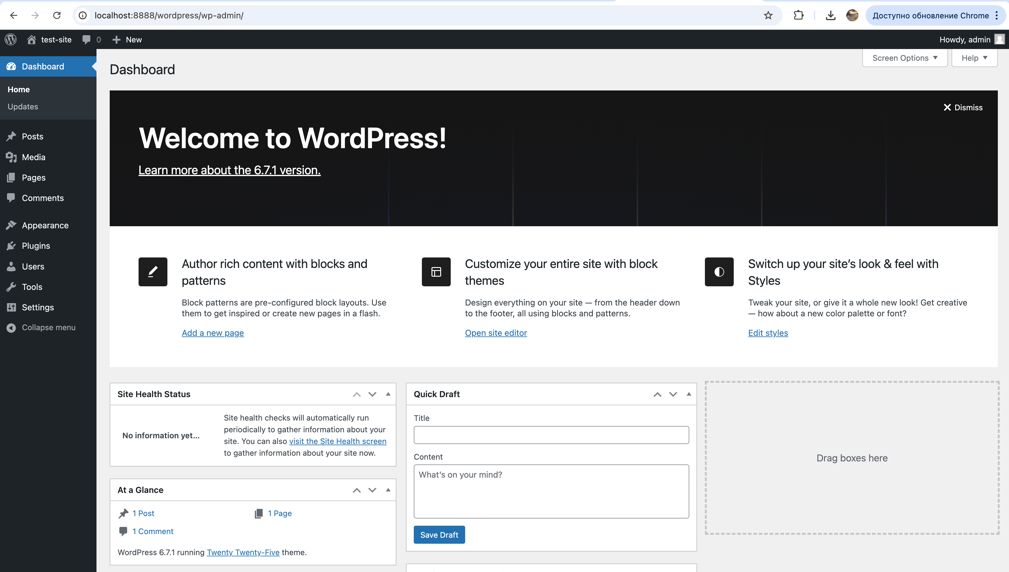
****

Рисунок 17 – Админ панель WordPress

Для того чтобы адрес http://test.site указывал на локальный сервер, необходимо отредактировать файл hosts на macOS. Этот файл связывает доменные имена с IP-адресами (рисунок 18).

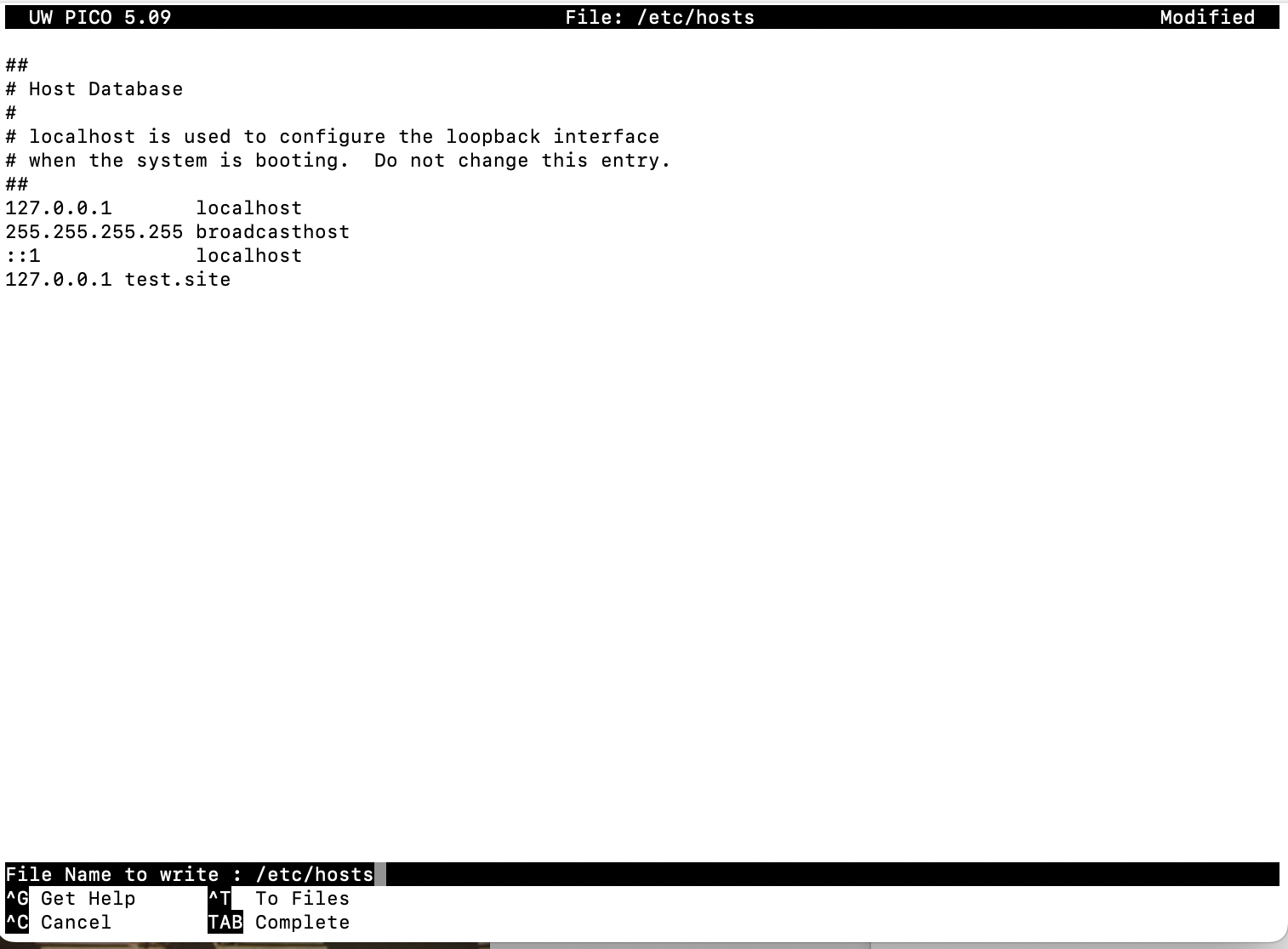


Рисунок 18 – Изменение файла hosts

Далее переходим по адресу <http://test.site>, и видим следующую страницу (рисунок 19).

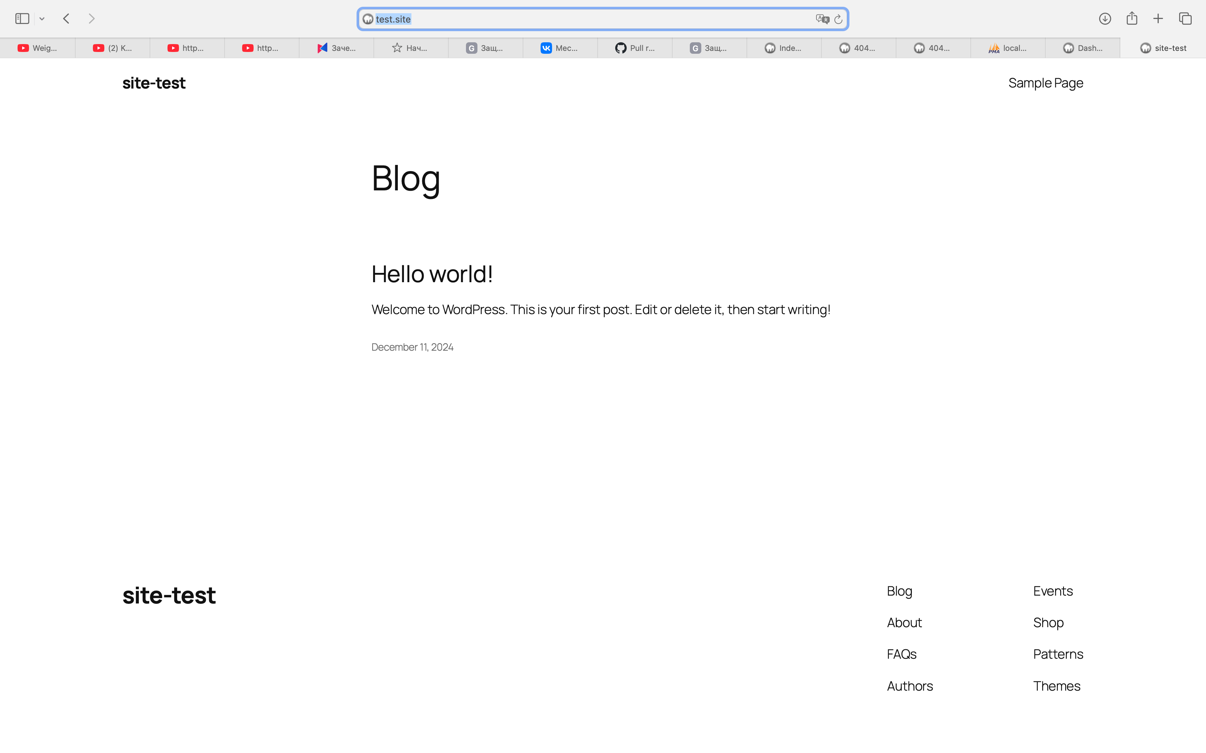
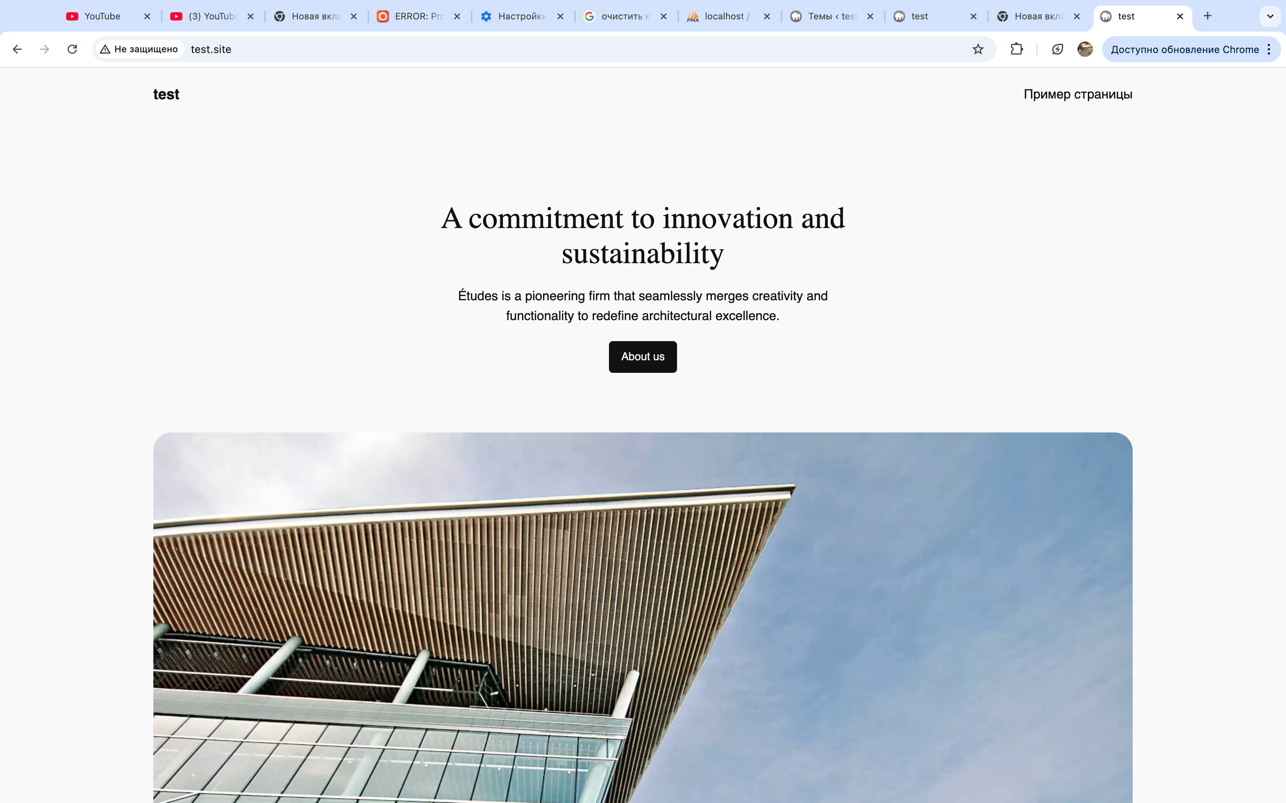


Рисунок 19 - http://test.site

После устанавливаем новую тему в настройках WordPress (рисунок 20).

  
Рисунок 20 – Новая тема

# **Вывод**

В рамках данной работы были изучен инструмент gulp, а также создан простейший сайт с использованием инструментов MAMP и Wordpress