

Санкт-Петербургский Национальный Исследовательский Университет  
Информационных Технологий, Механики и Оптики  
Факультет Инфокоммуникационных Технологий  
Направление подготовки Программирование в инфокоммуникационных  
системах

Лабораторная работа №3.

Выполнил  
Сидненко Д.Ю.  
Группа №К3320  
Проверила  
Марченко Е.В.

Санкт-Петербург, 2024

## **СОДЕРЖАНИЕ**

ЦЕЛЬ РАБОТЫ .....	3
ХОД РАБОТЫ.....	4
Задание 1.....	4
Задания 2 .....	6
Задание 3.....	10
ВЫВОД .....	14

## **ЦЕЛЬ РАБОТЫ**

1. Настроить gulp: а) создать два таска – настроить на последовательное и параллельное выполнение; б) настроить отображение файлов проекта в браузере и автоматическую перезагрузку при изменении одного из контролируемых файлов проекта.

2. Создать форму для отправки информации по обратной связи от пользователя сайта – передает информацию о себе: имя, фамилия, электронная почта, поле с обратной связью, должны быть радиокнопки (по меньшей мере 2 шт.) и должны быть чекбоксы (не менее трех). Разработать файл с формой и php скрипт по образцу.

3. Установить инструментарий для отладки проектов. Установить движок WordPress.

## ХОД РАБОТЫ

### Задание 1.

В начале выполнения работы был создан gulpfile с параллельным и последовательным выполнением двух простых задач и задачи с отображением изменений файлов проекта в браузере с автоматической перезагрузкой при изменении одного из файлов проекта.

```
const { src, watch, series, parallel } = require('gulp');
const browserSync = require('browser-sync').create();

// Задача: вывод текста в консоль
function logText(done) {
  console.log("Привет, это задача Gulp!");
  done(); //
}

// Задача: сложение чисел
function addNumbers(done) {
  const a = 5;
  const b = 7;
  console.log(`Сумма чисел ${a} и ${b} равна ${a + b}`);
  done();
}

// Задача: запуск локального сервера
function serve(done) {
  browserSync.init({
    server: {
      baseDir: './',
    },
    port: 3000,
  });
  done();
}

// Задача: перезагрузка браузера
function reload(done) {
  browserSync.reload();
  done();
}

// Задача: отслеживание изменений
function watchFiles() {
  watch('/*.html', reload); // Отслеживание HTML файлов
  watch('./css/*.css', reload); // Отслеживание CSS файлов
  watch('./js/*.js', reload); // Отслеживание JS файлов
}

// Последовательное выполнение задач
const runInSeries = series(logText, addNumbers, serve, watchFiles);

// Параллельное выполнение задач
const runInParallel = parallel(logText, addNumbers, series(serve, watchFiles));
```

Рисунок 1 – Код gulpfile часть 1

```
exports.logText = logText;
exports.addNumbers = addNumbers;
exports.serve = serve;
exports.reload = reload;
exports.watchFiles = watchFiles;
exports.runInSeries = runInSeries;
exports.runInParallel = runInParallel;

exports.default = runInSeries;
```

Рисунок 2 – Код gulpfile часть 2

После этого код был протестирован.

```
C:\Users\Dmitr\WebDevelopment_2024-2025\work\K3320\Сидненко Дмитрий Юрьевич\lab3>gulp runInParallel
[02:50:07] Using gulpfile ~\WebDevelopment_2024-2025\work\K3320\Сидненко Дмитрий Юрьевич\lab3\gulpfile.js
[02:50:07] Starting 'runInParallel'...
[02:50:07] Starting 'logText'...
[02:50:07] Starting 'addNumbers'...
Привет, это задача Gulp!
[02:50:07] Finished 'logText' after 1.7 ms
Сумма чисел 5 и 7 равна 12
[02:50:07] Finished 'addNumbers' after 1.91 ms
[02:50:07] Starting 'serve'...
[02:50:07] Finished 'serve' after 21 ms
[02:50:07] Starting 'watchFiles'...
[Browsersync] Access URLs:
    Local: http://localhost:3000
    External: http://10.102.37.150:3000
    UI: http://localhost:3001
    UI External: http://10.102.37.150:3001
[Browsersync] Serving files from: ./
[02:51:08] Starting 'reload'...
[02:51:08] Finished 'reload' after 1.25 ms
[Browsersync] Reloading Browsers...
```

Рисунок 3 – Выполнение gulpfile



Рисунок 4 – Автоматическое обновление страницы при изменении файла

## Задания 2

В этом задании была разработана форма по отправке обратной связи от пользователя сайта которая обрабатывается скриптом php.

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Обратная связь</title>
7 </head>
8 <body>
9   <h1>Форма обратной связи</h1>
10  <form action="submit.php" method="POST">
11    <!-- Имя -->
12    <label>Имя:</label>
13    <input type="text" name="name" required><br><br>
14
15    <!-- фамилия -->
16    <label>Фамилия:</label>
17    <input type="text" name="surname" required><br><br>
18
19    <!-- Электронная почта -->
20    <label>Электронная почта:</label>
21    <input type="email" name="email" required><br><br>
22
23    <!-- Отзыв -->
24    <label>Ваш отзыв:</label><br>
25    <textarea name="feedback" required></textarea><br><br>
26
27    <!-- Радиокнопки -->
28    <p>Как вы узнали о нас?</p>
29    <label><input type="radio" name="source" value="Интернет" required> Интернет</label><br>
30    <label><input type="radio" name="source" value="Друзья"> Друзья</label><br><br>
31
32    <!-- Чекбоксы -->
33    <p>Какие услуги вас интересуют?</p>
34    <label><input type="checkbox" name="services[]" value="Консультация"> Консультация</label><br>
35    <label><input type="checkbox" name="services[]" value="Обучение"> Обучение</label><br>
36    <label><input type="checkbox" name="services[]" value="Поддержка"> Поддержка</label><br><br>
37
38    <!-- Кнопка отправки -->
39    <button type="submit">Отправить</button>
40  </form>
41 </body>
42 </html>
43

```

Рисунок 5 – Код html

```

<?php
// Проверяем, какой метод используется (GET или POST)
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $method = $_POST;
} else {
    $method = $_GET;
}

// Получение данных из формы
$name = $method['name'] ?? '';
$surname = $method['surname'] ?? '';
$email = $method['email'] ?? '';
$feedback = $method['feedback'] ?? '';
$source = $method['source'] ?? '';
$services = $method['services'] ?? [];

// Проверяем обязательные поля
if (empty($name) || empty($surname) || empty($email) || empty($feedback) || empty($source)) {
    echo "<h1>Ошибка!</h1><p>Пожалуйста, заполните все обязательные поля.</p>";
    exit;
}

// Вывод данных
echo "<h1>Спасибо за обратную связь!</h1>";
echo "<p><b>Имя:</b> $name</p>";
echo "<p><b>Фамилия:</b> $surname</p>";
echo "<p><b>Электронная почта:</b> $email</p>";
echo "<p><b>Ваш отзыв:</b> $feedback</p>";
echo "<p><b>Как узнали:</b> $source</p>";

// Обрабатываем чекбоксы
if (!empty($services)) {
    echo "<p><b>Интересующие услуги:</b> " . implode(' ', $services) . "</p>";
} else {
    echo "<p><b>Интересующие услуги:</b> Не выбрано</p>";
}
?>

```

Рисунок 6 – Код php

## 1. Метод GET

Метод GET передаёт данные через URL (в строке запроса).

Особенности метода GET:

Данные видны в строке запроса, что делает метод менее безопасным для передачи конфиденциальной информации.

Имеет ограничения на объём данных, зависящие от длины URL (обычно до 2000 символов).

Может быть закеширован браузером.



Используется для запросов, которые не изменяют состояние сервера (например, получение данных или поиск).

Получение данных в PHP:

Используется суперглобальный массив `$_GET`.

## 2. Метод POST

Метод POST передаёт данные в теле HTTP-запроса, а не в URL.

Особенности метода POST:

Данные не отображаются в URL, что делает его более безопасным для передачи конфиденциальной информации, например, паролей.

Нет ограничения на объём данных, кроме ограничений на сервере (например, параметр `post_max_size` в конфигурации PHP).

Данные не кэшируются браузером.

Используется для операций, изменяющих состояние сервера (например, отправка формы или добавление данных).

Получение данных в PHP:

Используется суперглобальный массив `$_POST`.

← → ⓘ ↻ 🌐 localhost Обратная связь

### Форма обратной связи

Имя:

Фамилия:

Электронная почта:

Ваш отзыв:

Как вы узнали о нас?

☒ Интернет  
☐ Друзья

Какие услуги вас интересуют?

☐ Консультация  
☐ Обучение  
☒ Поддержка

Рисунок 7 – Вид страницы

← ⓘ ↻ 🌐 localhost localhost/feedback/submit.php

### Спасибо за обратную связь!

Имя: Дмитрий

Фамилия: Сидненко

Электронная почта: DmitriySidnenko@yandex.ru

Ваш отзыв: Круто

Как узнали: Интернет

Интересующие услуги: Поддержка

Рисунок 8 – Обработка скриптом php

### Задание 3.

В этом задании был установлен инструментальный ХАМРР а также движок WordPress и test.site был настроен на отображение собственного сайта. Для этого создаем базу данных на сайте phpMyAdmin, после устанавливаем WordPress по инструкции, настраиваем файл hosts и vhost-httpd. Выполнение задания представлено на рисунках 9 - 13.

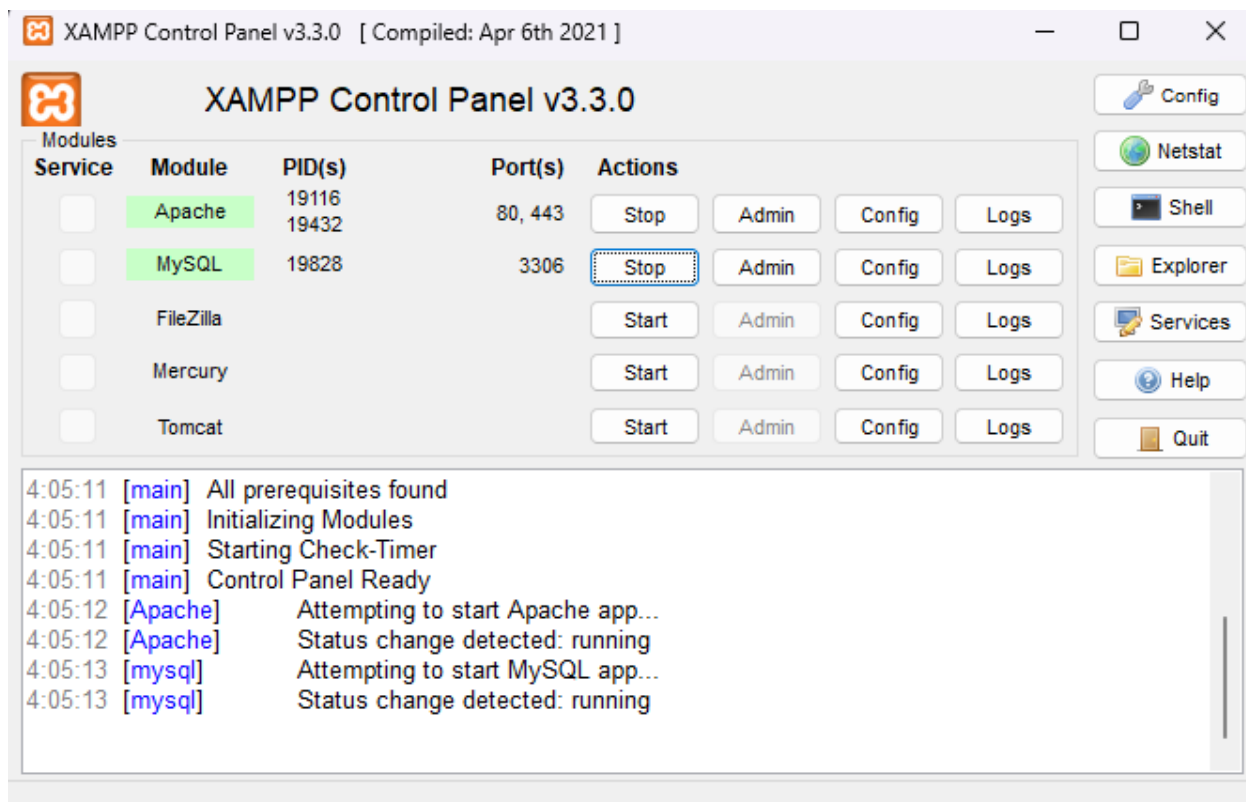


Рисунок 9 – Настройка XAMPP

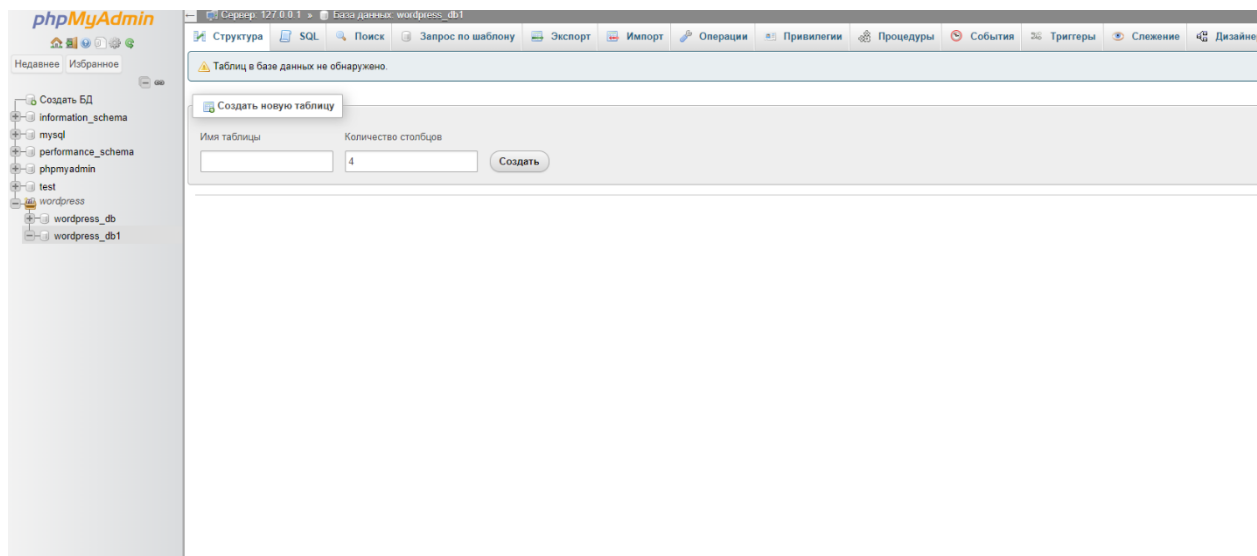
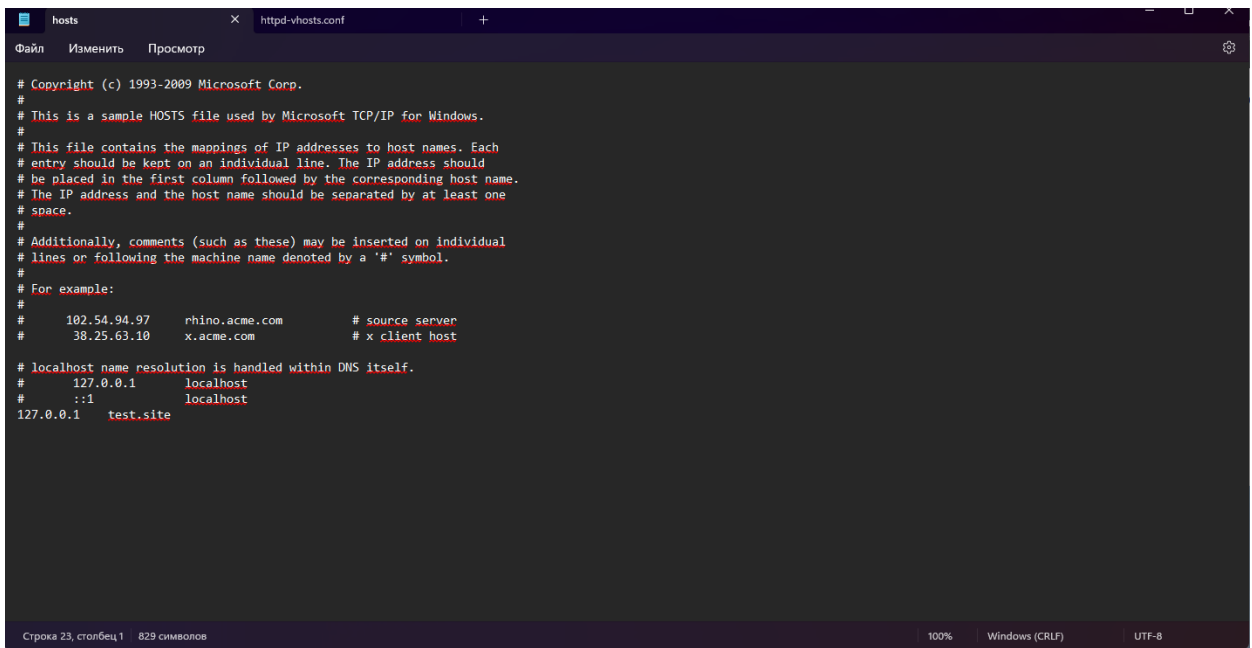


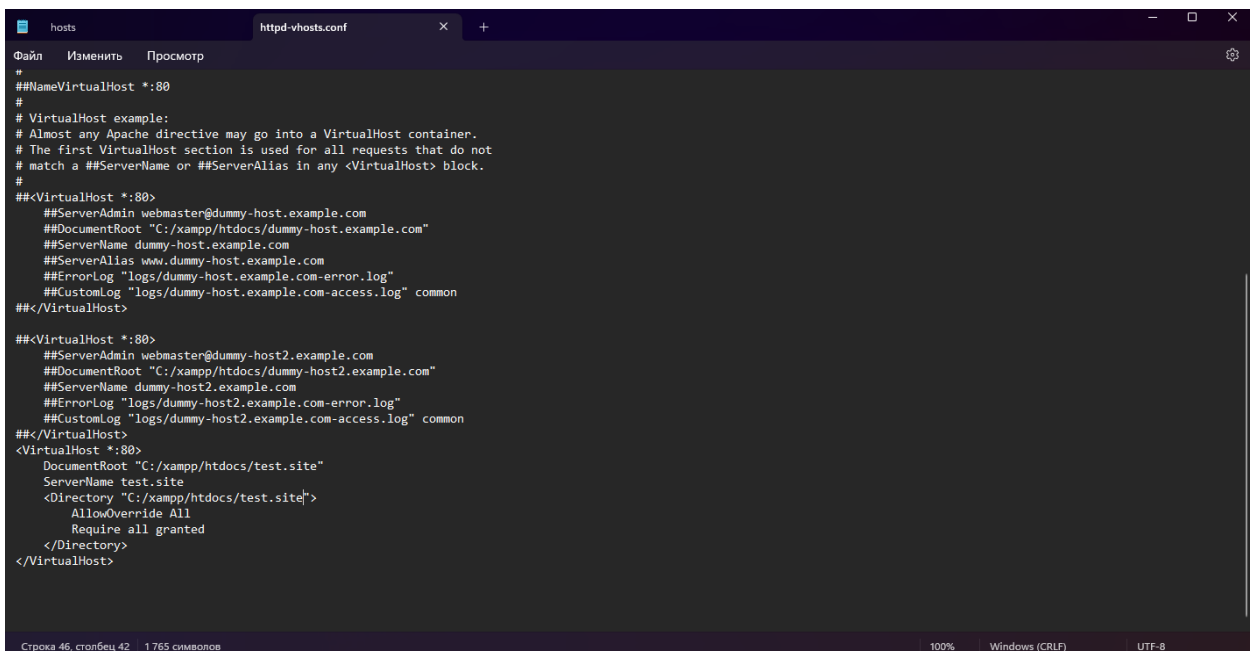
Рисунок 10 – Создание БД



```
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#       102.54.94.97       rhino.acme.com   # source server
#       38.25.63.10       x.acme.com      # x client host

# localhost name resolution is handled within DNS itself.
#       127.0.0.1         localhost
#       ::1               localhost
127.0.0.1   test.site
```

Рисунок 11 – Настройка файла hosts



```
##NameVirtualHost *:80
#
# VirtualHost example:
# Almost any Apache directive may go into a VirtualHost container.
# The first VirtualHost section is used for all requests that do not
# match a ##ServerName or ##ServerAlias in any <VirtualHost> block.
#
##<VirtualHost *:80>
#   ##ServerAdmin webmaster@dummy-host.example.com
#   ##DocumentRoot "C:/xampp/htdocs/dummy-host.example.com"
#   ##ServerName dummy-host.example.com
#   ##ServerAlias www.dummy-host.example.com
#   ##ErrorLog "logs/dummy-host.example.com-error.log"
#   ##CustomLog "logs/dummy-host.example.com-access.log" common
##</VirtualHost>

##<VirtualHost *:80>
#   ##ServerAdmin webmaster@dummy-host2.example.com
#   ##DocumentRoot "C:/xampp/htdocs/dummy-host2.example.com"
#   ##ServerName dummy-host2.example.com
#   ##ErrorLog "logs/dummy-host2.example.com-error.log"
#   ##CustomLog "logs/dummy-host2.example.com-access.log" common
##</VirtualHost>

<VirtualHost *:80>
    DocumentRoot "C:/xampp/htdocs/test.site"
    ServerName test.site
    <Directory "C:/xampp/htdocs/test.site">
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

Рисунок 12 – Настройка файла httpd-vhosts

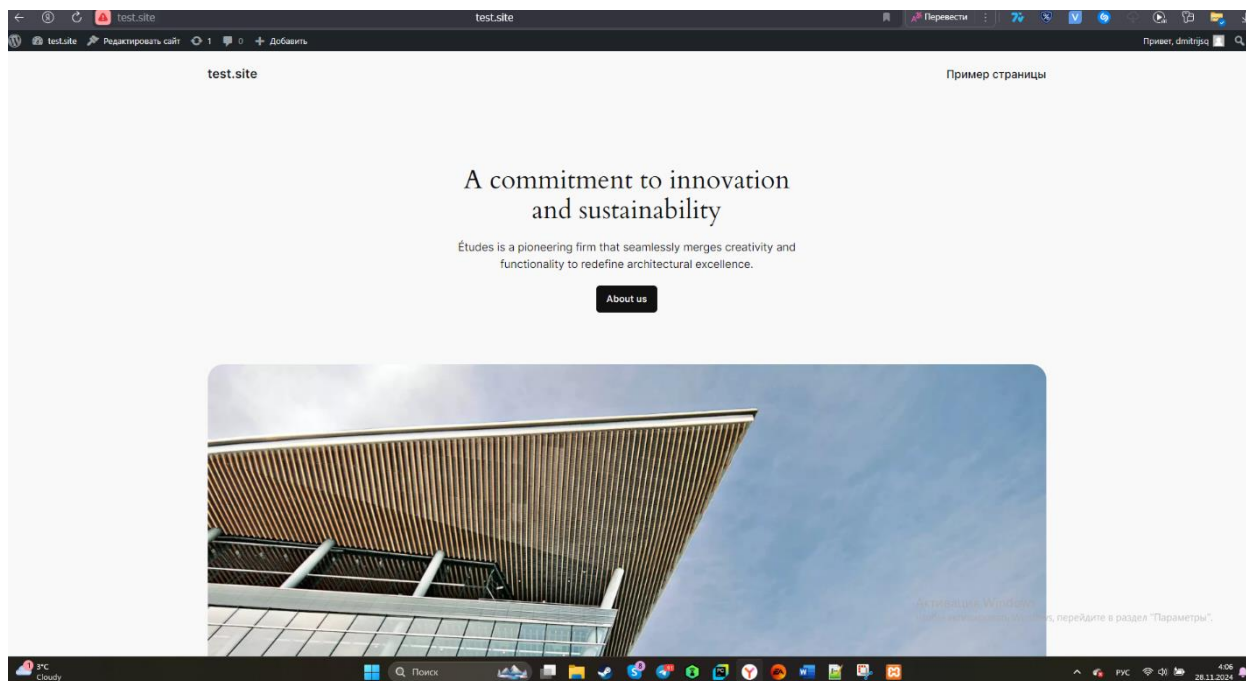


Рисунок 13 – Вид сайта

## **ВЫВОД**

В процессе выполнения лабораторной работы были созданы два таска в gulp настроенные на параллельное и последовательное выполнение, а также был создан таск который отслеживает изменение файлов проекта и автоматически обновляет их отображение в браузере. Также была разработана форма обратной связи с использованием html и php. После был установлен XAMPP и настроен движок WordPress.