

Министерство науки и высшего образования Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

ITMO University

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ 3

По дисциплине Web программирование

Тема работы Gulp, GET POST requests, wordpress

Обучающийся Телунц Эдуард Рубенович

Факультет Факультет инфокоммуникационных технологий

Группа K3321

Направление подготовки 11.03.02 Инфокоммуникационные технологии и системы связи

Образовательная программа Программирование в инфокоммуникационных системах

Обучающийся

10.12.2024

(дата)

\_\_\_\_\_

(подпись)

Телунц Э.Р.

(Ф.И.О.)

Руководитель

\_\_\_\_\_

(дата)

\_\_\_\_\_

(подпись)

Марченко Е.В.

(Ф.И.О.)

Санкт-Петербург  
2024 г.

## ЦЕЛЬ РАБОТЫ:

Научиться работать с Gulp, GET POST запросами и научиться поднимать собственный сервер.

## ЗАДАНИЕ 1:

Был создан файл gulpfile.js для пункта а:

```
C: > Users > Эдуард > HTML > Lab3 > Task1 > a > gulpfile.js > ...
1  const { series, parallel } = require('gulp');
2
3  function firstTask (done){
4    console.log("Первое задание");
5    done();
6  };
7
8  function secondTask(done){
9    console.log('Вторая задание');
10   done();
11 }
12
13 exports.parallel = parallel(firstTask, secondTask)
14 exports.series = series([firstTask, secondTask])
```

После чего в консоли были запущены задача parallel:

```
C:\Users\Эдуард\HTML\Lab3\Task1\a>gulp parallel
[15:03:04] Using gulpfile ~\HTML\Lab3\Task1\a\gulpfile.js
[15:03:04] Starting 'parallel'...
[15:03:04] Starting 'firstTask'...
[15:03:04] Starting 'secondTask'...
Первое задание
[15:03:04] Finished 'firstTask' after 1.82 ms
Вторая задание
[15:03:04] Finished 'secondTask' after 2.36 ms
[15:03:04] Finished 'parallel' after 4.91 ms
```

И series соответственно:

```
C:\Users\Эдуард\HTML\Lab3\Task1\>gulp series
[15:04:14] Using gulpfile ~\HTML\Lab3\Task1\>gulpfile.js
[15:04:14] Starting 'series'...
[15:04:14] Starting 'firstTask'...
Первое задание
[15:04:14] Finished 'firstTask' after 1.16 ms
[15:04:14] Starting 'secondTask'...
Вторая задание
[15:04:14] Finished 'secondTask' after 1.31 ms
[15:04:14] Finished 'series' after 5.46 ms
```

Как можем увидеть - все отработало верно.

Далее делаем пункт б и пишем соответствующий gulpfile:

```
C: > Users > Эдуард > HTML > Lab3 > Task1 > b > gulpfile.js > ...
1  var gulp = require('gulp');
2  var browserSync = require('browser-sync').create();
3  const sass = require('gulp-sass')(require('sass'));
4
5  gulp.task('sass', function compileSass() {
6    return gulp
7      .src('src/sass/*.scss')
8      .pipe(sass())
9      .pipe(gulp.dest('src/css'))
10     .pipe(browserSync.stream());
11  });
12
13  gulp.task('serve', function () {
14    gulp.watch("src/css/*.css", gulp.parallel('sass'));
15    gulp.watch("src/html/*.html").on('change', browserSync.reload());
16  });
17
18  gulp.task('init', function Goaaaaal() {
19    browserSync.init({
20      server: "./src/html",
21      injectChanges: false
22    });
23  });
24
25  gulp.task('default', gulp.series('init', 'serve'));
```

Здесь можем увидеть инициализацию browsersync сервера и установку слежки за изменяющимися файлами, а конкретно за файлами css и html. И определяем задачу перевода sass кода в css код.

Проследим за процессом запуска gulpfile:

```
C:\Users\Эдуард\HTML\Lab3\Task1\b>gulp
[15:09:10] Using gulpfile ~\HTML\Lab3\Task1\b\gulpfile.js
[15:09:10] Starting 'default'...
[15:09:10] Starting 'init'...
[Browsersync] Access URLs:
    -----
    Local: http://localhost:3000
    External: http://172.26.64.1:3000
    -----
    UI: http://localhost:3001
    UI External: http://172.26.64.1:3001
    -----
[Browsersync] Serving files from: ./src/html
```

Увидим результат в браузере:

## Несколько историй

[История первая](#)  
[История вторая](#)

### История первая и последняя kjdvjvjccjkcj

Давняя и неизменная часть истории костюма - **пуговица** - это и "скромная статистика", и яркая деталь на страницах многих произведений художественной литературы, отражающая не только веяния моды, но и социальное положение, вкус, воспитание.

#### Греческая загадка

В Европе первые пуговицы появились в IV веке до н.э. у греческих воинов. Видимо, возникла нужда в чем-то, чем можно было (кроме булавок) скреплять части одежды. Кожаные ремни на их "униформе" застегивались спереди на несколько металлических пуговиц на ножке. Среди археологических находок древнегреческого искусства с IV века до н.э. по I век н.э. попадаются пуговицы из золота.

Древние греки умели точно и основательно придумывать всяческим предметам и явлениям названия, которым суждено было сохраниться по сей день. Для пуговицы тоже существовало какое-то слово, но русский язык оно миновало. Слово "пуговица" попало на Русь с Востока, пришло оно из древнеиндийского языка и буквально означает "ком, куча, масса". До русских модников пуговица, которую тогда называли "пугвицей", "докатилась" в конце XV - начале XVI века.

До знаменитой петровской реформы костюма и приказа носить иностранное платье русские бояре и боярыни не скупясь платили золотых дел мастерам и иностранным купцам большие деньги за пуговицы из драгоценных металлов со вставками из бирюзы, жемчуга, кораллов, эмалей. Очень модными были круглые, похожие на шарики пуговицы с петелькой для пришивания - их называли кляшшами.

#### Пуговичный бал

Пуговицы, словно призраки, кочевали по перепутьям европейской моды начиная с конца XIII века, исчезая на несколько веков, пока XVII век, с его рационализмом и легким отношением к новациям, не

Изменим файл index.html и получим в браузере следующее:

## Несколько историй

[История первая](#)  
[История вторая](#)

### История первая и последняя

Давняя и неизменная часть истории костюма - **пуговица** - это и "скромная статистика", и яркая деталь на страницах многих произведений художественной литературы, отражающая не только веяния моды, но и социальное положение, вкус, воспитание.

#### Греческая загадка

В Европе первые пуговицы появились в IV веке до н.э. у греческих воинов. Видимо, возникла нужда в чем-то, чем можно было (кроме булавок) скреплять части одежды. Кожаные ремни на их "униформе" застегивались спереди на несколько металлических пуговиц на ножке. Среди археологических находок древнегреческого искусства с IV века до н.э. по I век н.э. попадаются пуговицы из золота.

Древние греки умели точно и основательно придумывать всяческим предметам и явлениям названия, которым суждено было сохраниться по сей день. Для пуговицы тоже существовало какое-то слово, но русский язык оно миновало. Слово "пуговица" попало на Русь с Востока, пришло оно из древнеиндийского языка и буквально означает "ком, куча, масса". До русских модников пуговица, которую тогда называли "пугвицей", "докатилась" в конце XV - начале XVI века.

До знаменитой петровской реформы костюма и приказа носить иностранное платье русские бояре и боярыни не скупясь платили золотых дел мастерам и иностранным купцам большие деньги за пуговицы из драгоценных металлов со вставками из бирюзы, жемчуга, кораллов, эмалей. Очень модными были круглые, похожие на шарики пуговицы с петелькой для пришивания - их называли кляшшами.

#### Пуговичный бал

Пуговицы, словно призраки, кочевали по перепутьям европейской моды начиная с конца XIII века, исчезая на несколько веков, пока XVIII век, с его рационализмом и легким отношением к новациям, не

## ЗАДАНИЕ 2:

Была реализована Razor страница с формой, которая будет посылать POST метод на сервер, который в свою очередь будет сохранять данные.

```
@page
@model WebLab3Task2.Pages.IndexModel
<form method="post">
    <div>
        <label asp-for="UserProfile.LastName">Фамилия:</label>
        <input name="UserProfile.LastName" />
    </div>

    <div>
        <label asp-for="UserProfile.FirstName">Имя:</label>
        <input name="UserProfile.FirstName" />
    </div>

    <div>
        <label asp-for="UserProfile.Email">Электронная почта:</label>
        <input type="email" name="UserProfile.Email" />
    </div>

    <div>
        <p>Выберите животное:</p>
        <input type="radio" id="cat" name="UserProfile.Animal" value="Кот" />
        <label for="cat">Коты</label><br />
        <input type="radio" id="dog" name="UserProfile.Animal" value="Собака" />
        <label for="dog">Собаки</label>
    </div>

    <div>
        <p>Выберите транспортное средство:</p>
        <input type="checkbox" id="bicycle" name="UserProfile.Transport" value="Велосипед" />
        <label for="bicycle">Велосипед</label><br />
        <input type="checkbox" id="motorcycle" name="UserProfile.Transport" value="Мотоцикл" />
        <label for="motorcycle">Мотоцикл</label><br />
        <input type="checkbox" id="car" name="UserProfile.Transport" value="Машина" />
        <label for="car">Автомобиль</label>
    </div>
</form>
```

```
<h1>Профиль пользователя</h1>

<table class="table">
    <thead>
        <tr>
            <th>Поле</th>
            <th>Значение</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>Фамилия</td>
            <td>@Model.UserProfile.LastName</td>
        </tr>
        <tr>
            <td>Имя</td>
            <td>@Model.UserProfile.FirstName</td>
        </tr>
        <tr>
            <td>Электронная почта</td>
            <td>@Model.UserProfile.Email</td>
        </tr>
        <tr>
            <td>Животное</td>
            <td>@Model.UserProfile.Animal</td>
        </tr>
        <tr>
            <td>Транспортные средства</td>
            <td>
                @foreach (var transport in Model.UserProfile.Transport)
                {
                    <span>@transport</span>
                    <br />
                }
            </td>
        </tr>
    </tbody>
</table>
```

```

    }
    </td>
  </tr>
</tbody>
</table>

```

Далее приведу код на C#, который связан с этой Razor страницей и обрабатывает код:

```

[IgnoreAntiforgeryToken]
public class IndexModel : PageModel
{
    [BindProperty]
    public UserProfile UserProfile { get; set; }

    public void OnGet()
    {
        if (UserProfile == null)
        {
            UserProfile = new UserProfile();
        }
    }

    public IActionResult OnPost()
    {
        return Page();
    }
}

```

Также приведу файл Program.cs, который служит настройкой для Asp Net Core приложения:

```

using WebLab3Task2.Data;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.
builder.Services.AddRazorPages();
builder.Services.AddControllers();
// Learn more about configuring Swagger/OpenAPI at https://aka.ms/aspnetcore/swashbuckle
builder.Services.AddEndpointsApiExplorer();
builder.Services.AddSwaggerGen();

var app = builder.Build();

// Configure the HTTP request pipeline.
if (app.Environment.IsDevelopment())
{
    app.UseSwagger();
    app.UseSwaggerUI();
}

app.UseHttpsRedirection();
app.UseAuthorization();
app.MapRazorPages();
app.MapControllers();
app.Run();

```

## Интерфейс программы:

Фамилия:   
Имя:   
Электронная почта:

Выберите животное:

- ☐ Коты  
☐ Собаки

Выберите транспортное средство:

- ☐ Велосипед  
☐ Мотоцикл  
☐ Автомобиль

## Профиль пользователя

Поле	Значение
Фамилия	
Имя	
Электронная почта	
Животное	
Транспортные средства	

## Интерфейс после заполнения:

Фамилия:   
Имя:   
Электронная почта:

Выберите животное:

- ☐ Коты  
☐ Собаки

Выберите транспортное средство:

- ☐ Велосипед  
☐ Мотоцикл  
☐ Автомобиль

## Профиль пользователя

Поле	Значение
Фамилия	Телунц
Имя	Эдуард
Электронная почта	edtelunts@gmail.com
Животное	Кот
Транспортные средства	Велосипед Мотоцикл

## ЗАДАНИЕ 3:

Для создания сервера был написан docker-compose.yaml:

```

version: '3'
services:
  nginx:
    image: nginx:latest
    container_name: test-site-nginx
    restart: unless-stopped
    ports:
      - "80:80"
    volumes:
      - ./nginx.conf:/etc/nginx/conf.d/default.conf
      - ./var/www/root/html:/usr/share/nginx/html
    networks:
      - app-network

networks:
  app-network:
    driver: bridge

```

Также был настроен nginx веб-сервер:

```

server {
    listen 80;
    server_name test.site;

    root /usr/share/nginx/html;
    index index.html;

    location / {
        try_files $uri $uri/ =404;
    }
}

```

И был написан соответствующий index.html:

```

<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Приветствие</title>
</head>
<body>
  <div>
    <h1>Добро пожаловать!</h1>
    <p>Мы рады видеть вас на нашей странице.</p>
  </div>
</body>
</html>

```



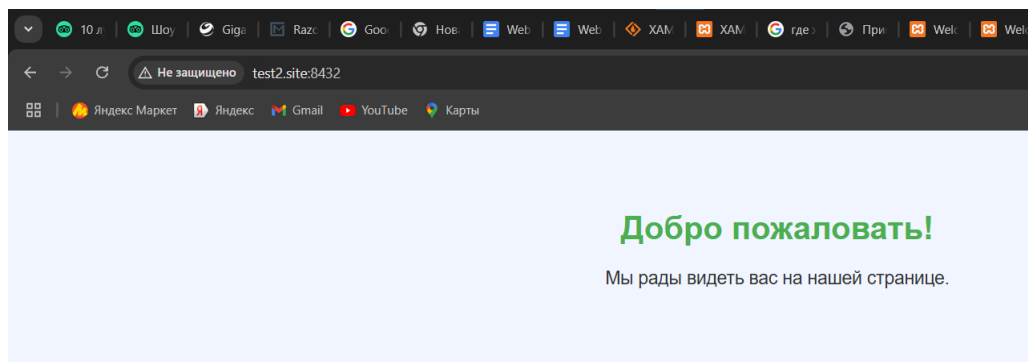
Также для DNS был прописан адрес в файле hosts:

```
# 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
163.172.167.207 bt.t-ru.org
163.172.167.207 bt2.t-ru.org
163.172.167.207 bt3.t-ru.org
163.172.167.207 bt4.t-ru.org
# Added by Docker Desktop
192.168.31.252 host.docker.internal
192.168.31.252 gateway.docker.internal
# To allow the same kube context to work on the host and the container:
127.0.0.1 kubernetes.docker.internal
# End of section

127.0.0.1 test.site
```

Можем наблюдать, что по нужному домену отображается index.html:



## ЗАКЛЮЧЕНИЕ:

В процессе выполнения лабораторной работы были приобретены навыки для работы с Gulp, Browser-sync, GET POST запросами. Были отточены навыки для написания docker-compose файлов.