Санкт-Петербургский Национальный Исследовательский Университет
Информационных Технологий, Механики и Оптики

Факультет инфокоммуникационных технологий

Отчет
по лабораторной работе №3

Выполнила: Гусева
Екатерина Михайловна
Группа №K3322

Проверил: Марченко
Елена Вадимовна

Санкт-Петербург,
2024

Цель работы:

Познакомиться с основами gulp, php, wordpress и инструментарием для отладки проектов

Задачи:

1.1. Настроить два gulp-таска на последовательное и параллельное выполнение.

1.2. Настроить отображение файлов проекта в браузере и автоматическую перезагрузку при изменении одного из контролируемых файлов проекта.

2. Разработать файл с формой для отправки информации по обратной связи то пользователя сайта и php скрипт. Познакомиться с использованием методов get и post

3. Установить инструментарий для отладки проектов. Настроить портал http://test.site

Результаты выполнения работы:

Задание 1:

Создадим две таски, в рамках которых выводятся в консоль слова «first task» и «second task» соответственно. Далее создадим таску «serial», которая запускает таски first и second последовательно, и таску «parallel», которая запускает first и second параллельно.

work > К3322 > Гусева_Екатерина > lab3 > task1.1 > gulpfile.js > ...

```
const gulp = require('gulp');

gulp.task('first', function (done) {
    console.log('first task');
    done();
})

gulp.task('second', function (done) {
    console.log('second task');
    done();
})

gulp.task('serial', gulp.series('first', 'second'))
gulp.task('parallel', gulp.parallel('first', 'second'))
```

Рисунок 1 - файл gulpfile.js для задания 1.1

Запустим таску «serial». В терминале видно, что таска «second» запускается только после завершения таски «first».

```
PS C:\Users\grina\Desktop\веб\WebDevelopment_2024-2025\work\К3322\Гусева_Екатерина\lab3\task1.1> gulp serial
[06:03:53] Using gulpfile ~\Desktop\веб\WebDevelopment_2024-2025\work\К3322\Гусева_Екатерина\lab3\task1.1\gulpfile.js
[06:03:53] Starting 'serial'...
[06:03:53] Starting 'first'...
first task
[06:03:53] Finished 'first' after 1.16 ms
[06:03:53] Starting 'second'...
second task
[06:03:53] Finished 'second' after 1.06 ms
[06:03:53] Finished 'serial' after 4.97 ms
PS C:\Users\grina\Desktop\веб\WebDevelopment_2024-2025\work\К3322\Гусева_Екатерина\lab3\task1.1>
```

Рисунок 2 - Выполнение таски "serial"

Запустим таску «parallel». В терминале можно увидеть, что таска «second» запустилась до завершения «first».

```
PS C:\Users\grina\Desktop\веб\WebDevelopment_2024-2025\work\К3322\Гусева_Екатерина\lab3\task1.1> gulp parallel
[06:05:52] Using gulpfile ~\Desktop\веб\WebDevelopment_2024-2025\work\К3322\Гусева_Екатерина\lab3\task1.1\gulpfile.js
[06:05:52] Starting 'parallel'...
[06:05:52] Starting 'first'...
[06:05:52] Starting 'second'...
first task
[06:05:52] Finished 'first' after 1.47 ms
second task
[06:05:52] Finished 'second' after 1.71 ms
[06:05:52] Finished 'parallel' after 3.86 ms
PS C:\Users\grina\Desktop\веб\WebDevelopment_2024-2025\work\К3322\Гусева_Екатерина\lab3\task1.1>
```

Рисунок 3 - Выполнение таски "parallel"

Задание 1.2

Настроим отображение файлов проекта в браузере и автоматическую перезагрузку при изменении одного из контролируемых файлов проекта. Для этого создадим таск «update»

```
work > К3322 > Гусева_Екатерина > lab3 > task1.2 > gulpfile.js > ...
1   const gulp = require('gulp');
2   const browserSync = require('browser-sync').create();
3
4   gulp.task('update', function () {
5       browserSync.init({
6           server: {
7               baseDir: './'
8           }
9       });
10
11      gulp.watch('*.html').on('change', browserSync.reload);
12      gulp.watch('*.css').on('change', browserSync.reload);
13      gulp.watch('*.js').on('change', browserSync.reload);
14  });
```

Рисунок 4 - Таск "update"

В терминале можно увидеть, что после запуска таски «update» начинают отслеживаться любые изменения в указанных в таске файлах.

```
PS C:\Users\grina\Desktop\веб\WebDevelopment_2024-2025\work\К3322\Гусева_Екатерина\lab3\task1.2> gulp update
[06:23:34] Using gulpfile ~\Desktop\веб\WebDevelopment_2024-2025\work\К3322\Гусева_Екатерина\lab3\task1.2\gulpfile.js
[06:23:34] Starting 'update'...
[Browsersync] Access URLs:
 ------------------------------------
       Local: http://localhost:3000
    External: http://192.168.56.1:3000
 ------------------------------------
          UI: http://localhost:3001
 UI External: http://192.168.56.1:3001
 ------------------------------------
[Browsersync] Serving files from: ./
[Browsersync] Reloading Browsers...
[Browsersync] File event [change] : styles.css
[Browsersync] File event [change] : styles.css
[Browsersync] File event [change] : styles.css
[Browsersync] File event [change] : styles.css
[Browsersync] File event [change] : styles.css
```

Рисунок 5 - Выполнение таски "update"

Задание 2

Создадим форму для отправки информации по обратной связи от пользователя сайта. Для обработки информации напишем php-скрипт, сохраняющий данные из формы в файл feedback.csv.

work > K3322 > Гусева_Екатерина > lab3 > task2 > index.html > html > body

```
<html>

<head>
    <title>Форма обратной связи</title>
    <link rel="stylesheet" href="styles.css">
</head>
</head>

<body>
    <form action="process.php" method="get">
        <label for="name">Имя:</label>
        <input type="text" id="name" name="name" required><br>

        <label for="surname">Фамилия:</label>
        <input type="text" id="surname" name="surname" required><br>

        <label for="email">Email:</label>
        <input type="email" id="email" name="email" required><br>

        <label for="message">Сообщение:</label>
        <textarea id="message" name="message" rows="5" required></textarea><br>

        <label>Выберите пол:</label>
        <label><input type="radio" name="option" value="man" required> Мужчина</label>
        <label><input type="radio" name="option" value="woman" required> Женщина</label><br>

        <label>Выберите направление:</label>
        <label><input type="checkbox" name="checkbox1" value="Frontend"> Frontend</label>
        <label><input type="checkbox" name="checkbox2" value="Backend"> Backend</label>
        <label><input type="checkbox" name="checkbox3" value="Mobile"> Mobile</label>
        <label><input type="checkbox" name="checkbox4" value="ML"> ML</label>
        <label><input type="checkbox" name="checkbox5" value="DevOps"> DevOps</label><br>

        <input type="submit" value="Отправить">
    </form>

</body>
```

Рисунок 6 - Файл index.html с формой для обратной связи

work > K3322 > Гусева_Екатерина > lab3 > task2 > process.php

```php
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $data = $_POST["name"] . "; ";
    $data .= $_POST["surname"] . "; ";
    $data .= $_POST["email"] . "; ";
    $data .= $_POST["message"] . "; ";
    $data .= $_POST["option"] . "; ";

    $checkboxes = [];
    if (isset($_POST["checkbox1"]))
        $checkboxes[] = $_POST["checkbox1"];
    if (isset($_POST["checkbox2"]))
        $checkboxes[] = $_POST["checkbox2"];
    if (isset($_POST["checkbox3"]))
        $checkboxes[] = $_POST["checkbox3"];
    $data .= implode(separator: ", ", array: $checkboxes) . "\n";


    $file = "feedback.csv";
    $fp = fopen(filename: $file, mode: "a"); // Открыть файл в режиме добавления

    if ($fp) {
        fwrite(stream: $fp, data: $data);
        fclose(stream: $fp);
        echo "Данные успешно сохранены!";
    } else {
        echo "Ошибка при записи данных в файл.";
    }
} else {
    echo "Некорректный запрос.";
}
?>
```

Рисунок 7 - php-скрипт для обработки формы

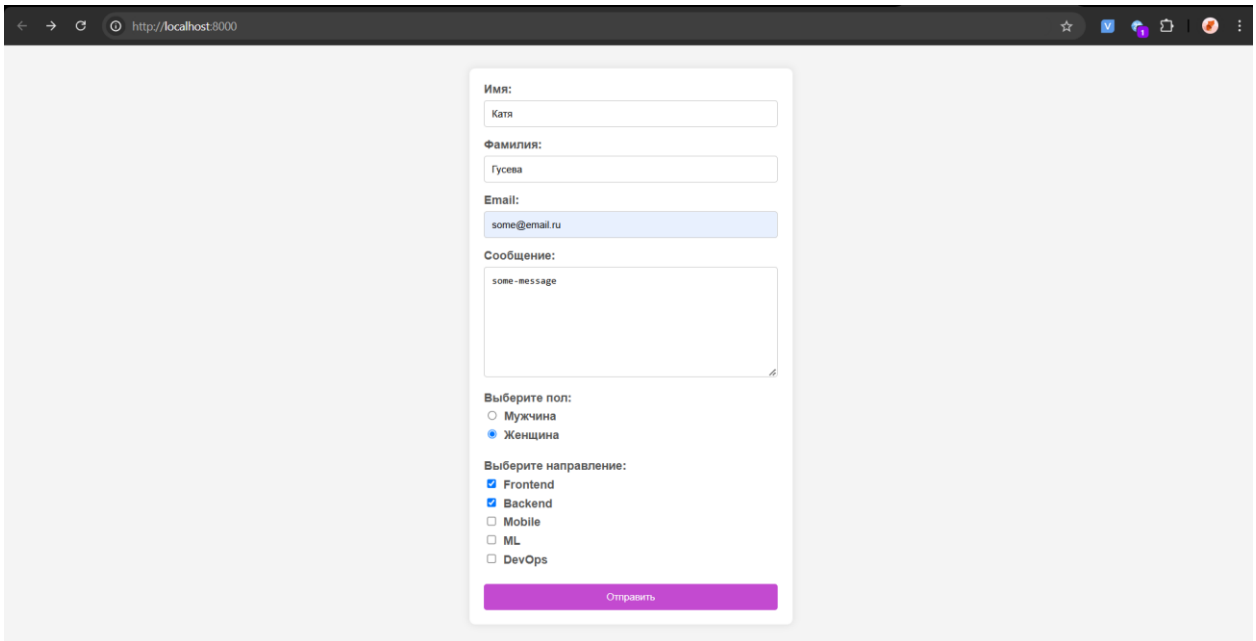

Рисунок 8 - Форма в браузере

Методы GET и POST – ключевые методы HTTP. GET используется для получения данных от сервера. Если при передаче данных формы указать метод GET, то данные будут передаваться как параметры в URL адресе и потому будут видны в адресной строке браузера, истории браузера и кэше, что небезопасно.

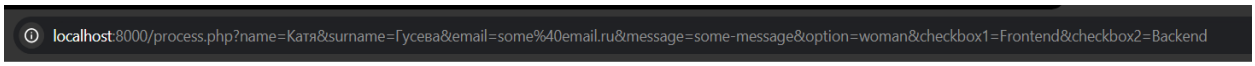

Рисунок 9 - Передача данных с помощью GET метода

POST используется для отправки данных на сервер. В отличие от GET, POST передает данные не в URL, а в теле HTTP-запроса, что делает его более безопасным.

Задание 3

Напишем docker-compose.yml файл для автоматического развертывания Nginx, MySql и Wordpress. Настроим nginx так, что бы по адресу http://test.site открывался наш портал.

work > K3322 > Гусева_Екатерина > lab3 > task3 > docker-compose.yml

```yaml
services:
  nginx:
    image: nginx:latest
    volumes:
      - ./nginx:/etc/nginx/conf.d
      - ./data/html:/var/www/html
      - ./logs/nginx:/var/log/nginx
    ports:
      - "8080:80"
    links:
      - wordpress

  wordpress:
    depends_on:
      - db
    image: wordpress:latest
    volumes:
      - ./data/html:/var/www/html
    ports:
      - "80:80"
    restart: always
    environment:
      WORDPRESS_DB_HOST:  db:3306
      WORDPRESS_DB_USER: test
      WORDPRESS_DB_PASSWORD: test
      WORDPRESS_DB_NAME: wordpress

  db:
    image: mysql:5.7
    volumes:
      - ./data/mysql:/var/lib/mysql
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: test
      MYSQL_DATABASE: wordpress
      MYSQL_USER: test
      MYSQL_PASSWORD: test
```

Рисунок 10 - docker-compose.yml

В браузере можно увидеть, что сайт успешно открывается

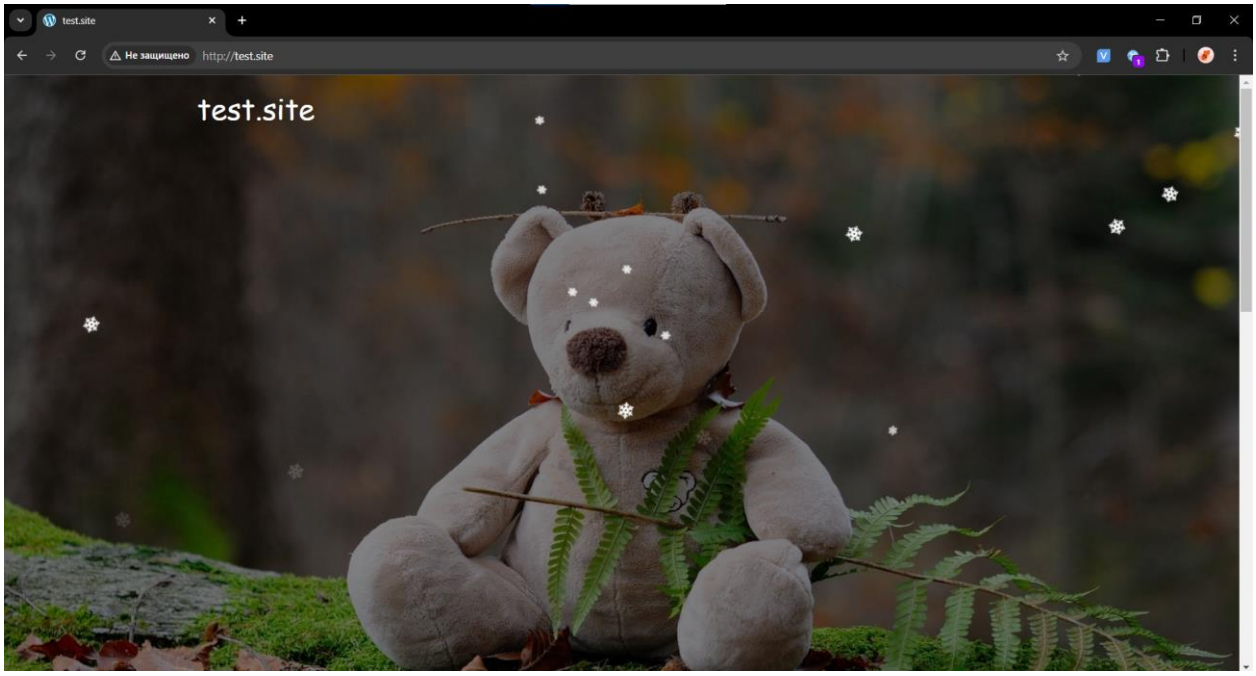

Рисунок 11 - http://test.site

Вывод:

В ходе выполнения данной лабораторной работы были изучены основы gulp, php, wordpress и получены навыки работы с инструментарием для отладки проектов.