

Санкт-Петербургский Национальный Исследовательский
Университет

Информационных Технологий, Механики и Оптики

Факультет Инфокоммуникационных Технологий

Web-программирование

Лабораторная работа 3

Выполнил

Скворцов И. В.

Проверила

Марченко Е. В.

Санкт-Петербург, 2024

Цели работы

Цели:

- написать gulp скрипты
- создать форму для отправки обратной связи с использованием php скрипта
- изучить основы инструментария для отладки проектов

Задание 1

Первым делом надо инициализировать npm в папке задания для работы с gulp. Для этого введем команду `npm init` и создадим `gulpfile`. Кроме того установим необходимые зависимости командой `npm install --save-dev gulp gulp-series browser-sync`. На рисунке 1 будет представлен код написанных скриптов.

```
4
5 // Функция для динамического импорта модуля `del`
6 async function clean() {
7   const del = (await import('del')).deleteSync;
8   console.log('Очистка папки dist...');
9   del(['dist/**', '!dist']);
10 }
11
12 // Задача для копирования HTML файлов в папку dist
13 function copyHtml(done) {
14   gulp.src('src/*.html')
15     .pipe(gulp.dest('dist'))
16     .pipe(browserSync.stream()); // Обновляем браузер
17   done();
18 }
19
20 // Задача для копирования CSS файлов
21 function copyCss(done) {
22   gulp.src('src/css/*.css')
23     .pipe(gulp.dest('dist/css'))
24     .pipe(browserSync.stream()); // Обновляем браузер
25   done();
26 }
27
28 // Задача для запуска локального сервера и отслеживания изменений
29 function serve(done) {
30   browserSync.init({
31     server: {
32       baseDir: './dist' // Указываем директорию для отображения в браузере
33     }
34   });
35
36   // Наблюдаем за изменениями в файлах и перезагружаем браузер
37   gulp.watch('src/*.html', copyHtml);
38   gulp.watch('src/css/*.css', copyCss);
39
40   done();
41 }
42
```

Рисунок 1 - написанные скрипты

Далее на рисунке 2 можно увидеть, что часть задач запущена параллельно, а часть последовательно

```
42 }
43
44 // Задачи для последовательного выполнения
45 gulp.task('build', gulp.series(clean, copyHtml, copyCss));
46
47 // Задачи для параллельного выполнения
48 gulp.task('parallel-build', gulp.parallel(copyHtml, copyCss));
49
50 // Задача по умолчанию для запуска проекта с отслеживанием изменений
51 gulp.task('default', gulp.series('build', serve));
52
```

Рисунок 2 - запуск скриптов

Теперь запустим и проверим, что все работает и изменения в html автоматически синхронизируются. Результат можно видеть на рисунке 3.

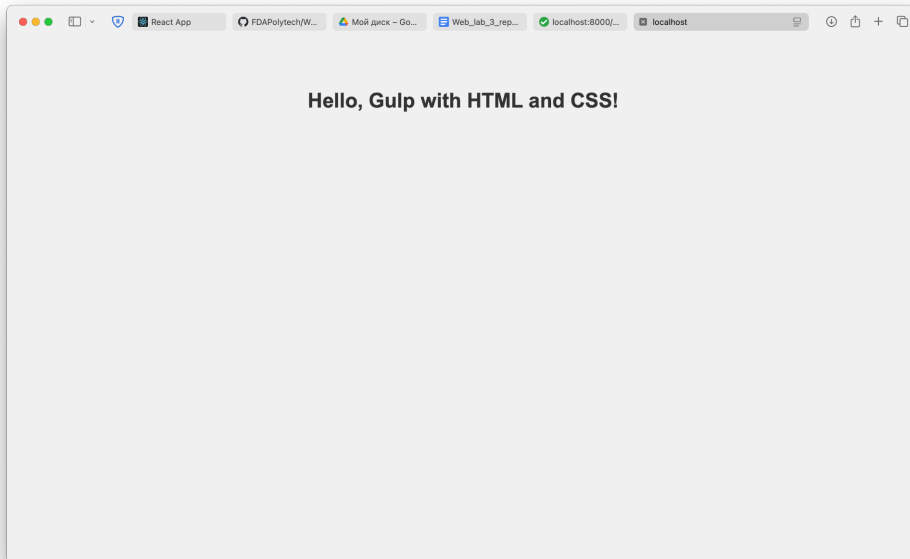


Рисунок 3 - запуск скриптов

Теперь поменяем html код и проверим, будет ли окно в браузере обновлено.

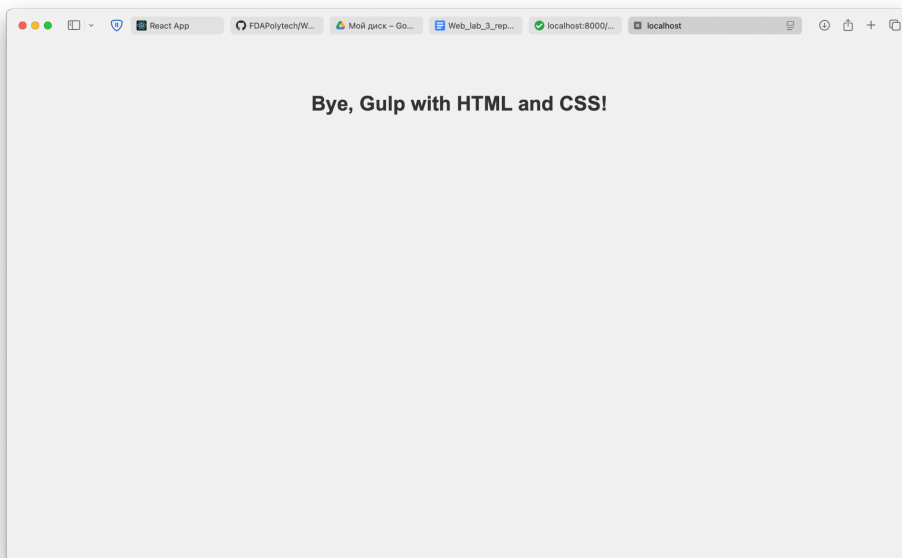


Рисунок 4 - результат изменения кода

Как мы видим, при изменении кода, страничка в браузере автоматически синхронизировалась

Задание 2

В этом задании нам надо создать форму для отправки обратной связи и сделать php скрипт для обработки запросов. Для отправки формы будет использоваться метод POST. Метод GET используется для получения данных. Метод POST применяется для отправки данных в теле запроса, он подходит для операций, изменяющих состояние сервера. Ниже на рисунке 5 можно увидеть код отправки обратной связи со всеми необходимыми элементами.

```
1  <!DOCTYPE html>
2  <html lang="ru">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Форма обратной связи</title>
7  </head>
8  <body>
9      <h1>Форма обратной связи</h1>
10     <form action="handle_form.php" method="POST">
11         <label for="first_name">Имя:</label>
12         <input type="text" id="first_name" name="first_name" required><br><br>
13
14         <label for="last_name">Фамилия:</label>
15         <input type="text" id="last_name" name="last_name" required><br><br>
16
17         <label for="email">Электронная почта:</label>
18         <input type="email" id="email" name="email" required><br><br>
19
20         <label for="feedback">Обратная связь:</label><br>
21         <textarea id="feedback" name="feedback" rows="4" cols="50" required></textarea><br><br>
22
23         <p>Как вы узнали о нас?</p>
24         <input type="radio" id="internet" name="source" value="internet" required>
25         <label for="internet">Интернет</label><br>
26         <input type="radio" id="friend" name="source" value="friend" required>
27         <label for="friend">Рекомендация друга</label><br><br>
28
29         <p>Какие услуги вас интересуют?</p>
30         <input type="checkbox" id="service1" name="services[]" value="design">
31         <label for="service1">Дизайн</label><br>
32         <input type="checkbox" id="service2" name="services[]" value="development">
33         <label for="service2">Разработка</label><br>
34         <input type="checkbox" id="service3" name="services[]" value="seo">
35         <label for="service3">SEO</label><br><br>
36
37         <input type="submit" value="Отправить">
38     </form>
39 </body>
40 </html>
```

Рисунок 5 - html код формы обратной связи

Теперь напишем php скрипт который будет это обрабатывать. Его код можно увидеть на рисунке 6. Для того, чтобы убедиться в корректности работы скрипты, был добавлен вывод в командную строку полученных параметров.

```

1 <?php
2 if ($_SERVER["REQUEST_METHOD"] == "POST") {
3     // Получаем данные из формы
4     $first_name = htmlspecialchars($_POST['first_name']);
5     $last_name = htmlspecialchars($_POST['last_name']);
6     $email = htmlspecialchars($_POST['email']);
7     $feedback = htmlspecialchars($_POST['feedback']);
8     $source = htmlspecialchars($_POST['source']);
9     $services = isset($_POST['services']) ? $_POST['services'] : [];
10
11     // Выводим данные для демонстрации
12     echo "<h1>Спасибо за обратную связь!</h1>";
13     echo "<p><strong>Имя:</strong> $first_name</p>";
14     echo "<p><strong>Фамилия:</strong> $last_name</p>";
15     echo "<p><strong>Электронная почта:</strong> $email</p>";
16     echo "<p><strong>Обратная связь:</strong> $feedback</p>";
17     echo "<p><strong>Источник:</strong> $source</p>";
18     echo "<p><strong>Интересующие услуги:</strong> " . implode(", ", $services) . "</p>";
19 } else {
20     echo "Форма должна быть отправлена методом POST.";
21 }
2 ?>

```

Рисунок 6 - php скрипт

Теперь запустим командой `php -S localhost:8000` и перейдем по указанному в команде адресу. На рисунке 7 можно увидеть, что все успешно запустилось.

Форма обратной связи

Имя: bvz

Фамилия: njnjdf

Электронная почта: test@test.test

Обратная связь:
s/koiopdsfjsdf

Как вы узнали о нас?

☒ Интернет

☐ Рекомендация друга

Какие услуги вас интересуют?

☐ Дизайн

☒ Разработка

☒ SEO

Отправить

Рисунок 7 - форма обратной связи

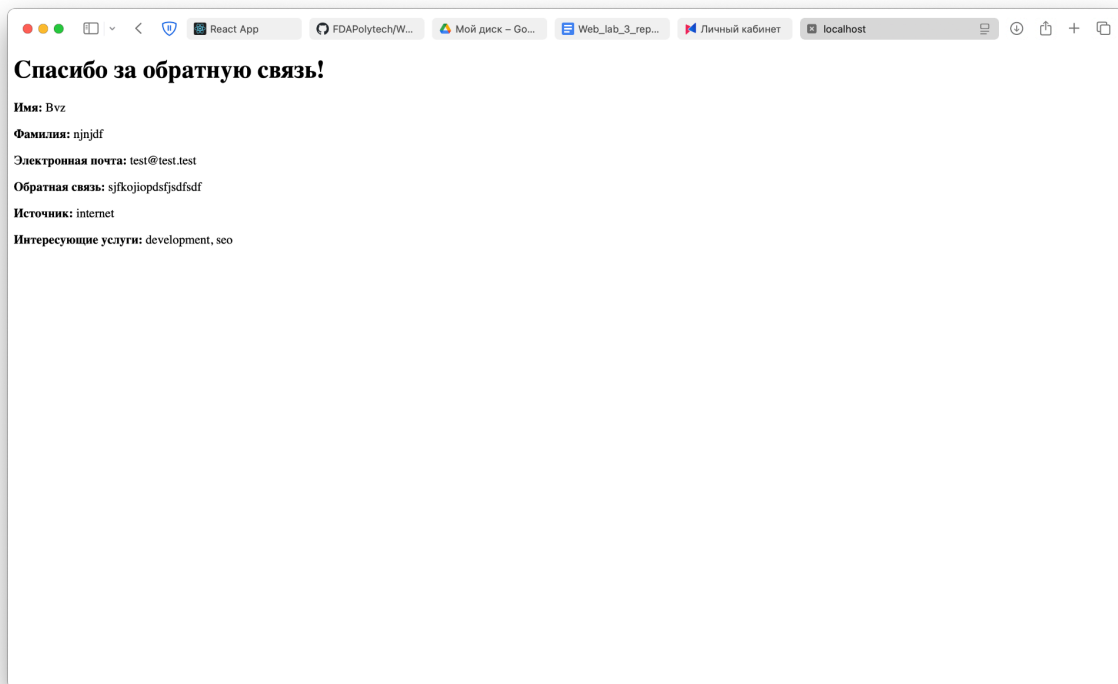


Рисунок 8 - результат отправки формы

Как мы видим, php скрипт успешно сработал.

Задание 3

В данном задании нам надо настроить wordpress и сделать чтобы по адресу <http://test.site> отвечал наш портал. Для этого будем использовать docker-compose. Снизу на рисунке 9 можно увидеть файл docker-compose.yaml.

```
docker-compose.yml - The Compose specification establishes a st
1  services:
2      wordpress:
3          image: wordpress
4          restart: always
5          ports:
6              - 8080:80
7          environment:
8              WORDPRESS_DB_HOST: db
9              WORDPRESS_DB_USER: exampleuser
0              WORDPRESS_DB_PASSWORD: examplepass
1              WORDPRESS_DB_NAME: exampledb
2          volumes:
3              - wordpress:/var/www/html
4
5      db:
6          image: mysql:8.0
7          restart: always
8          environment:
9              MYSQL_DATABASE: exampledb
0              MYSQL_USER: exampleuser
1              MYSQL_PASSWORD: examplepass
2              MYSQL_RANDOM_ROOT_PASSWORD: '1'
3          volumes:
4              - db:/var/lib/mysql
5
6  volumes:
7      wordpress:
8      db:
```

Рисунок 9 - docker-compose.yaml

Запустим командой `docker compose up`. Теперь по адресу `localhost:8080` открывается наш портал wordpress. Теперь надо сделать, чтоб он открывался по адресу `http://test.site/`. Для этого добавим запись в `/etc/hosts`

```
127.0.0.1 test.site
```


Заключение

В данной работе мы познакомились с php скриптами и настроили портал wordpress