

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

Факультет Инфокоммуникационных Технологий

Web-программирование

Практическая работа №3

Выполнил:

Гайдаренко Никита Кириллович

Группа:

K3321

Проверила:

Марченко Елена Вадимовна

Санкт-Петербург,

2024

Цель: настроить gulp, разработать web-страницу, которая получает некоторую информацию, научиться работа с WordPress

Задачи:

- Настроить gulp и создать task согласно заданию
- Создать страницу, которая необходимо для отправки информации об обратной связи
- Настроить Wordpress.

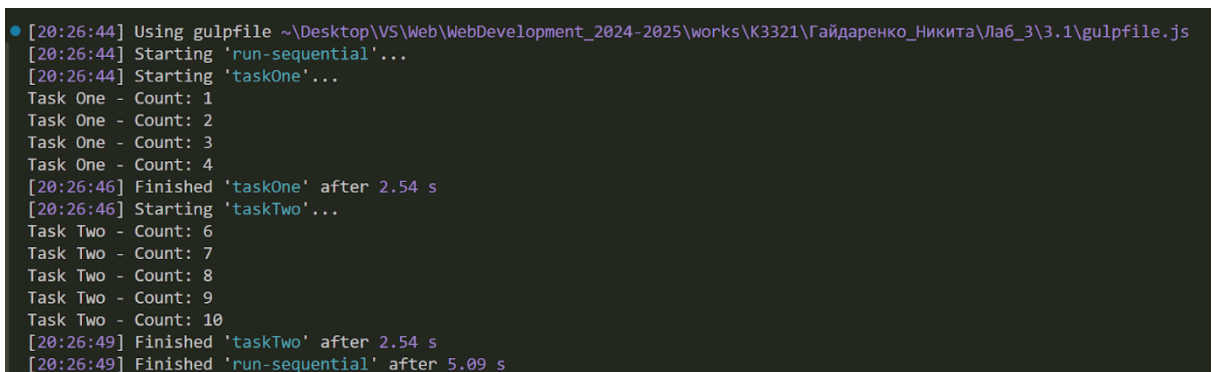
Ход работы

Задание 1:

Для выполнения первой части задания 1, необходимо было настроить gulp на параллельное и последовательное выполнение задач.

Для начал необходимо было установить Node.js и gulp.

Для последовательное выполнения задач, необходимо было указать параметр **series** и те задачи, которые необходимо выполнить. Результат работы представлен на рисунке 1. TaskOne выводит числа от 1 до 4. После вывода все чисел, выполняется TaskTwo, которая выводит значения от 6 до 10. Можно сделать вывод, что при последовательном выполнении задач, одна задача дожидается остальных перед тем, как начать выполнение и в один момент времени работает только одна задача.



```
[20:26:44] Using gulpfile ~\Desktop\VS\Web\WebDevelopment_2024-2025\works\K3321\Гайдаренко_Никита\Лаб_3\3.1\gulpfile.js
[20:26:44] Starting 'run-sequential'...
[20:26:44] Starting 'taskOne'...
Task One - Count: 1
Task One - Count: 2
Task One - Count: 3
Task One - Count: 4
[20:26:46] Finished 'taskOne' after 2.54 s
[20:26:46] Starting 'taskTwo'...
Task Two - Count: 6
Task Two - Count: 7
Task Two - Count: 8
Task Two - Count: 9
Task Two - Count: 10
[20:26:49] Finished 'taskTwo' after 2.54 s
[20:26:49] Finished 'run-sequential' after 5.09 s
```

Рисунок 1 – Последовательное выполнение задач

Для параллельного выполнения задач, необходимо указать параметр **parallel**. Результат работы представлен на рисунке 2. В этом случае две задачи выполняются одновременно, и одна задача не дожидается выполнения остальных. После каждой итерации в цикле идет засыпание на 500 мс, что придает большей наглядности работе параллельного выполнения задач.

```
PS C:\Users\Никита\Desktop\VS\Web\WebDevelopment_2024-2025\works\K3321\Гайдаренко_Никита\Лаб_3\3.1> gulp run-parallel
[20:26:56] Using gulpfile ~\Desktop\VS\Web\WebDevelopment_2024-2025\works\K3321\Гайдаренко_Никита\Лаб_3\3.1\gulpfile.js
[20:26:56] Starting 'run-parallel'...
[20:26:56] Starting 'taskOne'...
[20:26:56] Starting 'taskTwo'...
Task One - Count: 1
Task Two - Count: 6
Task One - Count: 2
Task Two - Count: 7
Task One - Count: 3
Task Two - Count: 8
Task One - Count: 4
Task Two - Count: 9
Task One - Count: 5
[20:26:59] Finished 'taskOne' after 2.52 s
Task Two - Count: 10
[20:26:59] Finished 'taskTwo' after 2.52 s
[20:26:59] Finished 'run-parallel' after 2.52 s
PS C:\Users\Никита\Desktop\VS\Web\WebDevelopment_2024-2025\works\K3321\Гайдаренко_Никита\Лаб_3\3.1> █
```

Рисунок 2 - Параллельное выполнение задач

Во второй части первого задания, необходимо реализовать автоматическую перезагрузку браузера при изменении одного из файлов проекта. Для этого был создан gulp файл, в котором реализована **parallel task**. Для компиляции scss в css файл.

При создании scss файла будет создаваться css файл при условии, если он не еще создан, иначе он будет просто обновляться. На рисунке 3 представлен результат запуска default task.

```

PS C:\Users\Никита\Desktop\VS\Web\WebDevelopment_2024-2025\works\K3321\Гайдаренко_Никита\Лаб_3\3.1> gulp
[20:29:05] Using gulpfile ~\Desktop\VS\Web\WebDevelopment_2024-2025\works\K3321\Гайдаренко_Никита\Лаб_3\3.1\gulpfile.js
[20:29:05] Starting 'default'...
[20:29:05] Starting 'sass'...
[20:29:05] Starting 'browser-sync'...
[20:29:05] Starting 'watch'...
Deprecation Warning: The legacy JS API is deprecated and will be removed in Dart Sass 2.0.0.

More info: https://sass-lang.com/d/legacy-js-api

[Browsersync] Access URLs:
-----
    Local: http://localhost:3000
  External: http://169.254.136.169:3000
-----
    UI: http://localhost:3001
  UI External: http://169.254.136.169:3001
-----
[Browsersync] Serving files from: application
[Browsersync] 1 file changed (style.css)
[20:29:05] Finished 'sass' after 258 ms
[20:29:19] Starting 'sass'...
Deprecation Warning: The legacy JS API is deprecated and will be removed in Dart Sass 2.0.0.

More info: https://sass-lang.com/d/legacy-js-api

[Browsersync] 1 file changed (style.css)
[20:29:19] Finished 'sass' after 62 ms
[20:29:38] Starting 'sass'...
Deprecation Warning: The legacy JS API is deprecated and will be removed in Dart Sass 2.0.0.

More info: https://sass-lang.com/d/legacy-js-api

[Browsersync] 1 file changed (style.css)
[20:29:38] Finished 'sass' after 38 ms
[20:29:46] Starting 'sass'...
Deprecation Warning: The legacy JS API is deprecated and will be removed in Dart Sass 2.0.0.

More info: https://sass-lang.com/d/legacy-js-api

[Browsersync] 1 file changed (style.css)
[20:29:46] Finished 'sass' after 33 ms

```

Рисунок 3 – Результат запуска gulp файла

Задание 2

В этом задании необходимо было создать форму для отправки информации по обратной связи. Пользователь вводит информацию о себе имя, фамилия, email, отзыв, также выбирает необходимый функционал в радиокнопках и чекбоксах. На основе задания была создана страница, которая представлена на рисунке 4.

Обратная связь

Имя:

Фамилия:

Электронная почта:

Отзыв:

Выберите вашу оценку:

☒ ★★★★★

☐ ★★★★★

☐ ★★★

☐ ★★

☐ ★

Хотите получать информацию о нас:

☒ Получать обновления

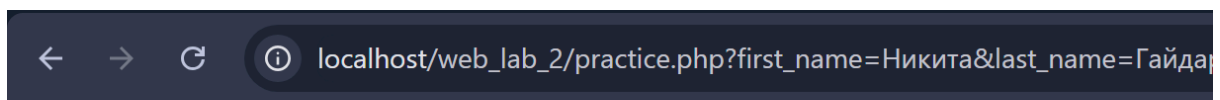
☒ Получать новости

☐ Получать специальные предложения

Рисунок 4 – Форма отправки информации

Необходимо было скачать XAMP, который позволяет локально запускать сервер. Внесенная информация в поля формы отправляются при помощи POST-запроса.

После нажатия кнопки “*Отправить (GET)*”, данные отправятся при помощи GET-запроса. Результат работы представлен на рисунке 5. Можно заметить, что при GET-запросе, все данные, которые мы отправляли в форме, отображаются в URL-е страницы после вопросительного знака. На рисунке 6, можно увидеть часть URL странице.



Спасибо за ваш отзыв, Никита!

Электронная почта: nicita.gaidarenko@yandex.ru

Ваш отзыв: Все прекрасно

Ваша оценка: ★★★★★

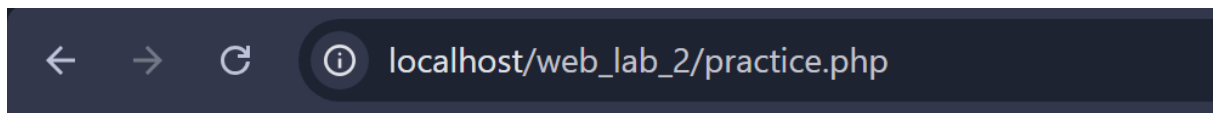
Ваши предпочтения: Получать обновления, Получать новости

Рисунок 5 – Результат отправки формы при GET запросе

/practice.php?first_name=Никита&last_name=Гайдаренко&email=nicita.gaidarenko%40yandex.ru&feedback=Все+прекрасно

Рисунок 6 – URL страницы

При нажатии кнопки “*Отправить (POST)*”, данные отправляются при помощи POST-запроса. Результат работы представлен на рисунке 7. В отличие от GET запроса, в URL-е страницы не отображаются параметры, которые мы указывали в форме.



Спасибо за ваш отзыв, Никита!

Электронная почта: nicita.gaidarenko@yandex.ru

Ваш отзыв: Все прекрасно

Ваша оценка: ★★★★★

Ваши предпочтения: Получать обновления, Получать новости

Рисунок 7 - Результат отправки формы при POST запросе


Можно сделать вывод, что метод POST безопаснее, чем GET-запрос. В действительности метод POST отправляет данные на серверную часть приложения, в теле запроса.

Метод GET передает данные в URL, и это подходит, когда данные не конфиденциальные. На практике GET использует для получения данных с сервера.

Задание 3

В данном задании необходимо было установить WordPress. Также необходимо отображение странице по URL <http://test.site>.

Для начала необходимо установить WordPress, после перенести все файлы в папку htdocs программы XAMPP. После необходимо было настроить WordPress. Один из этапов настройки представлен на рисунке 8.



Введите здесь информацию о подключении к базе данных. Если вы в ней не уверены, свяжитесь с поддержкой вашего хостинга.

Имя базы данных	<input type="text" value="wordpress"/>	
	Имя базы данных, в которую вы хотите установить WordPress.	
Имя пользователя	<input type="text" value="root"/>	
	Имя пользователя базы данных.	
Пароль	<input type="password" value="password"/>	Показать
	Пароль пользователя базы данных.	
Сервер базы данных	<input type="text" value="localhost"/>	
	Если localhost не работает, нужно узнать правильный адрес в службе поддержки хостинг-провайдера.	
Префикс таблиц	<input type="text" value="wp_"/>	
	Если вы хотите запустить несколько копий WordPress в одной базе, измените это значение.	
<input type="button" value="Отправить"/>		

Рисунок 8 – Настройка WordPress

После необходимо было добавить виртуальный порт в файл `httpd-vhosts.conf`. Результат изменений представлен на рисунке 9.


```

"
##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>

```

Рисунок 9 – Файл *httpd-vhosts.conf*

После сайт был доступен по ссылке <http://test.site>, результат представлен на рисунке 10.

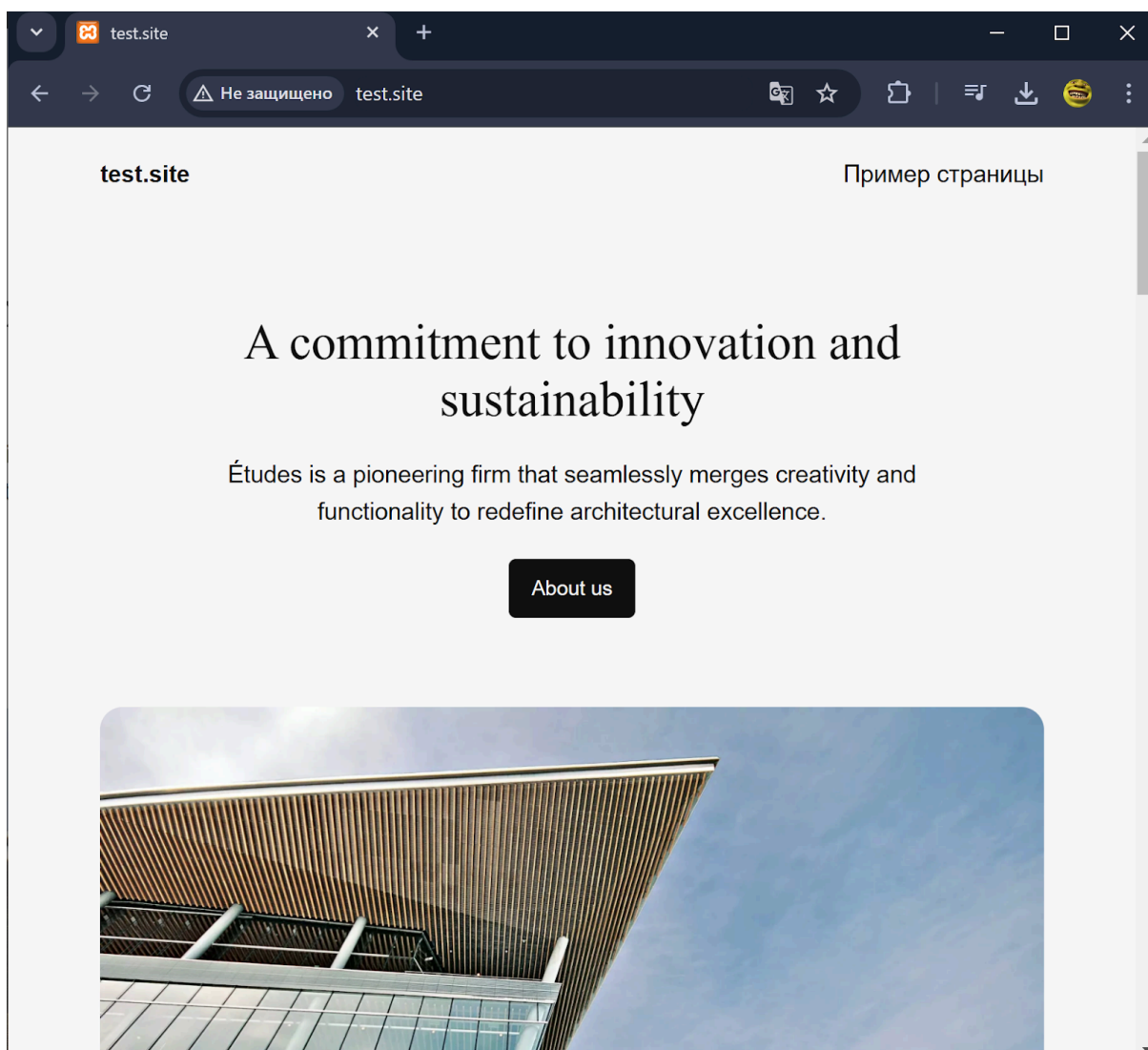


Рисунок 10 – Переход по <http://test.site>

Вывод

Настроил gulp. Разработал web-страницу, которая получает информацию от клиента. Научился работать с WordPress