

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

Факультет Инфокоммуникационных Технологий  
11.03.02 Инфокоммуникационные технологии и системы связи

Лабораторная работа №2  
«Git, Gulp»

Выполнила:  
Бахтина Анастасия Вячеславовна  
Группа: К3320  
Проверила:  
Марченко Елена Вадимовна

Санкт-Петербург  
2024

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	3
1 Задание 1. Установка и работа с Git .....	4
2 Задание 2. Установка и работа с Gulp .....	5
3 Задание 3. Написание программы клиент.....	7
ЗАКЛЮЧЕНИЕ .....	10

## ВВЕДЕНИЕ

**Цель работы:** Установить Git на компьютер и выполнить изменения в файлах проекта. А также установить gulp, проверить процесс установки и отметить основные этапы. Написать программу клиент, которая показывает web-страницы одна за другой из списка.

## 1 Задание 1. Установка и работа с Git

Для работы с данным заданием была установлена специальная программа GitKraken, чтобы было удобно работать с GitHub. В данной программе был выбран проект и настроен к нему доступ. На рисунке 1.1 представлены созданные ранее несколько коммитов.

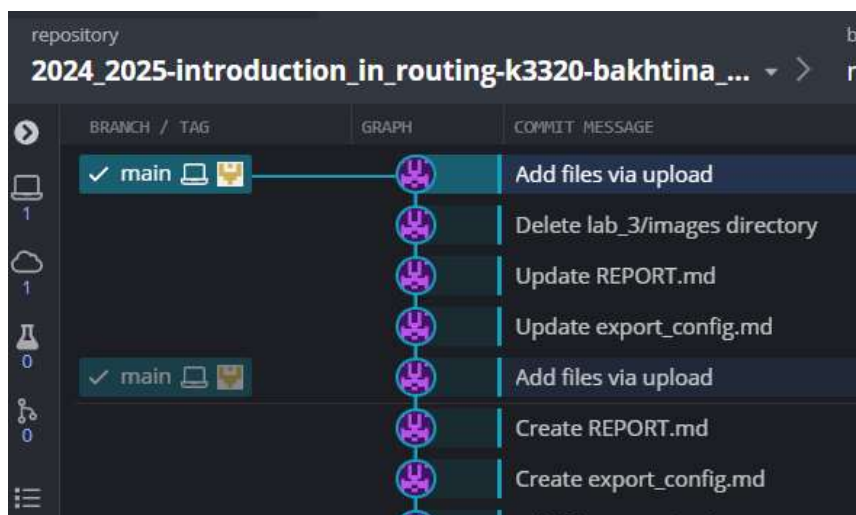


Рисунок 1.1 – Созданные коммиты

На рисунке 1.2 показаны те же коммиты только в проекте GitHub.

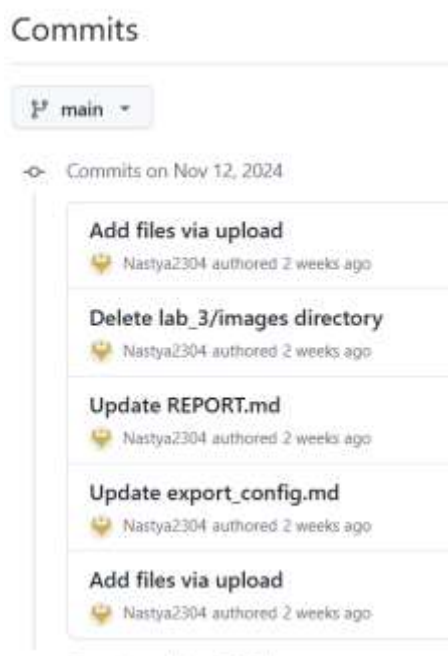


Рисунок 1.2 – Коммиты в проекте GitHub

Ссылка на репозиторий данного проекта в гитхабе:

[https://github.com/Nastya2304/2024\\_2025-introduction\\_in\\_routing-k3320-bakhtina\\_a\\_v](https://github.com/Nastya2304/2024_2025-introduction_in_routing-k3320-bakhtina_a_v)

## 2 Задание 2. Установка и работа с Gulp

Для работы со вторым заданием сначала на компьютер был установлен сервер Node JS с официального сайта <https://nodejs.org/en/>. На рисунке 2.1 представлена проверка успешной установки данного сервера с информацией о версиях.

```
C:\Users\123>node -v
v22.11.0

C:\Users\123>npm -v
10.9.0
```

Рисунок 2.1 – Проверка установки сервера

Далее с помощью специальных команд из данного сайта <https://gulpjs.com/docs/en/getting-started/quick-start/> был установлен Gulp. Используя данную команду **npm install-D gulp** была произведена установка Gulp в качестве NPM пакета, также предварительно была произведена настройка npm (рисунок 2.2).

```
C:\Users\123>npm install -D gulp

added 142 packages in 6s

14 packages are looking for funding
  run `npm fund` for details

C:\Users\123>_
```

Рисунок 2.2 – Установка Gulp в качестве NPM пакета

Затем с помощью специальных команд в проект было установлено все необходимое для работы с Gulp (рисунок 2.3).

```
PS C:\Users\123\Desktop\works\K3320\Бахтина Анастасия Вячеславовна\lab2\Task 2> npm install --save-dev gulp

added 142 packages, and audited 143 packages in 7s

14 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Users\123\Desktop\works\K3320\Бахтина Анастасия Вячеславовна\lab2\Task 2> gulp -v
CLI version: 3.0.0
Local version: 5.0.0
```

Рисунок 2.3 – Установка Gulp в проект

На рисунке 2.4 представлен итоговый код файла `package.json`.

```
package.json > ...
1  {
2    "name": "task-2",
3    "version": "1.0.0",
4    "main": "index.js",
5    "scripts": {
6      "test": "echo \"Error: no test specified\" && exit 1"
7    },
8    "keywords": [],
9    "author": "",
10   "license": "ISC",
11   "description": "",
12   "devDependencies": {
13     "gulp": "^5.0.0"
14   }
15 }
```

Рисунок 2.4 – Код файла

После всех установок был создан код файла **gulpfile.js** для простого task (рисунок 2.5).

```
gulpfile.js > gulp.task('Task2') callback
1  const gulp = require('gulp');
2
3  gulp.task('Task2', function (done) {
4    console.log('Выполняется простая задача на Gulp!');
5    done();
6  });
```

Рисунок 2.5 – Код файла

На рисунке 2.6 представлен результат работы данного файла в консоле.

```
PS C:\Users\123\Desktop\works\K3320\Бахтина Анастасия Вячеславовна\lab2\Task2> gulp Task2
[20:20:03] Using gulpfile ~\Desktop\works\K3320\Бахтина Анастасия Вячеславовна\lab2\Task2\gulpfile.js
[20:20:03] Starting 'Task2'...
Выполняется простая задача на Gulp!
[20:20:03] Finished 'Task2' after 2.31 ms
```

Рисунок 2.6 – Результат работы файла

### 3 Задание 3. Написание программы клиент

В третьем задании второй лабораторной работы необходимо было написать программу клиент, которая должна показывать web-страницы одна за другой из списка. Кроме того, в данной программе нужно задать адрес страниц и интервал показа сайта выбранной страницы.

Для работы с данным заданием были созданы три файла: **script\_web.js**, **index\_web.html** и **index\_web.css**.

Первый код файла **index\_web.html** создает веб-страницу, которая позволяет пользователям вводить ссылку на сайт и указывать продолжительность показа страницы. После отправки формы, введенная ссылка добавляется в список ранее введенных сайтов. На странице также есть элемент **iframe**, который отображает содержимое указанного сайта. В шапке страницы содержится запрос для ввода ссылки и продолжительности, в основной части расположена форма для ввода данных, секция для отображения списка введенных ссылок и содержимого **iframe**. Также подключены внешние файлы CSS и JavaScript для стилей и функционала.

Второй код файла **index\_web.css** оформляет внешний вид веб-страницы. Он задает основные стили для элементов страницы, таких как **body**, **header**, **footer**, **main**, формы и других секций. В частности, он определяет расположение элементов, цвета фона и текста, отступы, границы и стили для кнопок.

Третий код файла **script\_web.js** управляет формой на веб-странице, позволяя пользователям вводить URL сайта и длительность его показа. При отправке формы, URL и длительность добавляются в массив **websiteList**. Если сайты не показываются, запускается функция **displayNextPage**, которая задает URL для **iframe** и использует **setTimeout** для переключения на следующий сайт после указанного времени. Также данный код обновляет список введенных URL на странице.

На рисунке 3.1 представлен код файла **index\_web.html**.

```

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      <link rel="stylesheet" href="index_web.css">
7      <title>Task #3 Website</title>
8  </head>
9  <body>
10     <header>
11         <div class="header-content">
12             <h1>Введите в поле ссылку на сайт и укажите время показа страницы:</h1>
13         </div>
14     </header>
15     <main>
16         <form>
17             <input type="url" name="url" placeholder="Введите ссылку на сайт" required>
18             <input type="number" name="duration" placeholder="Введите время (в секундах)" required>
19             <input type="submit" value="Добавить">
20         </form>
21         <section class="url-list-section">
22             <h2>Сайты, которые мы вводили ранее:</h2>
23             <div id="sites"></div>
24         </section>
25         <section class="iframe-section">
26             <h2>Просмотр сайта:</h2>
27             <iframe id="frame" title="site"></iframe>
28         </section>
29     </main>
30     <footer>
31         <p>© 2024 Бахтина Анастасия. К3320</p>
32     </footer>
33     <script src="script_web.js"></script>
34 </body>
35 </html>

```

Рисунок 3.1 – Код файла

На рисунке 3.2 представлен результат запуска файла в браузере.

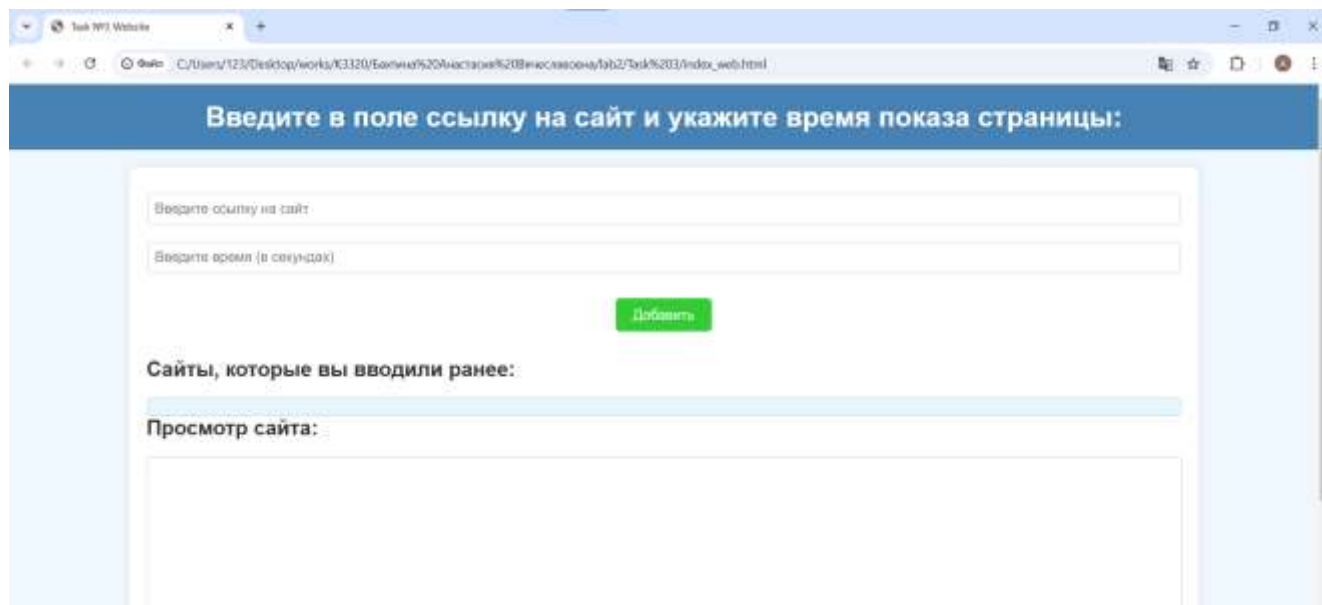


Рисунок 3.2 – Запуск файла в браузере

А на рисунке 3.3 представлен итоговый результат работы данного файла с заполненными полями.



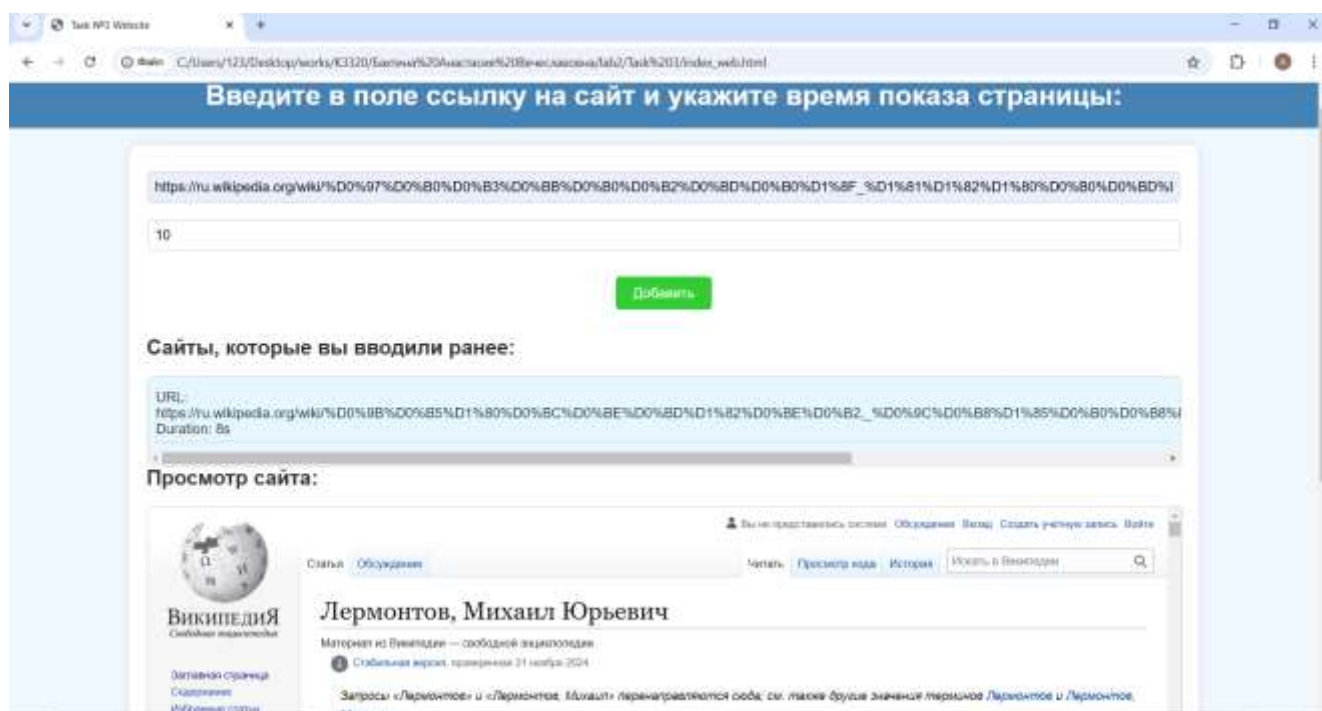


Рисунок 3.3 – Итоговый результат работы файла

## **ЗАКЛЮЧЕНИЕ**

В данной лабораторной работе цель была достигнута: был установлен Git на компьютер и выполнены изменения в файлах проекта, также был установлен gulp, проверен процесс установки и отмечены основные этапы. Была написана программа клиент, которая показывает web-страницы одна за другой из списка.