

Санкт-Петербургский Национальный Исследовательский Университет

Информационных Технологий, Механики и Оптики

Факультет инфокоммуникационных технологий

Лабораторная работа №2

Выполнила: Ульянова Е.А.

Проверила: Марченко Е.В.

Санкт-Петербург,

2024

Оглавление

Цель работы	3
Ход работы.....	3
1. Работа с Git	3
2. Работа с gulp	4
3. Написание программы.....	6
Заключение	10

Цель работы

Целью данной лабораторной работы является приобретение основных навыков по работе с Git и gulp, а также написание программы, отображающей web-страницы из списка.

Ход работы

1. Работа с Git

Для выполнения первого задания необходимо установить Git на компьютер. Так как данное действие было выполнено ранее, достаточно проверить имеющуюся версию (рисунок 1).

```
C:\Users\Ekaterina>git --version  
git version 2.47.1.windows.1
```

Рисунок 1 – Просмотр версии git

Действительно, Git установлен и готов к работе. Теперь требуется настроить его на работу с проектом, в качестве которого была выбрана реализация предыдущей лабораторной работы по данному предмету. Для этого с помощью команды `git clone` было осуществлено копирование рассматриваемого репозитория (рисунок 2).

```
C:\Users\Ekaterina\Desktop\Учебные задания\3 курс\Web>git clone https://github.com/Ekaterina-Ulyanova/WebDevelopment_2024-2025.git -b lab1  
Cloning into 'WebDevelopment_2024-2025'...  
remote: Enumerating objects: 152, done.  
remote: Counting objects: 100% (69/69), done.  
remote: Compressing objects: 100% (51/51), done.  
remote: Total 152 (delta 29), reused 41 (delta 15), pack-reused 83 (from 1)  
Receiving objects: 100% (152/152), 28.79 MiB | 732.00 KiB/s, done.  
Resolving deltas: 100% (36/36), done.  
C:\Users\Ekaterina\Desktop\Учебные задания\3 курс\Web>cd WebDevelopment_2024-2025
```

Рисунок 2 - Клонирование репозитория

Далее в проект были внесены некоторые изменения: переименованы заголовки разделов в файле `about_buttons.html`, заменён файл с названием `button_1.jpg`, а также добавлено новое изображение `img_1.jpg`. Отображение данных процессов в командной строке представлено на рисунке 3.

```

C:\Users\Ekaterina\Desktop\Учебные задания\3 курс\Web\WebDevelopment_2024-2025>git add "work/K3322/Ульянова Екатерина
a Алексеевна/Lab1/about_buttons.html"

C:\Users\Ekaterina\Desktop\Учебные задания\3 курс\Web\WebDevelopment_2024-2025>git commit -m "rename headings"
[lab1 81681cb] rename headings
1 file changed, 5 insertions(+), 5 deletions(-)

C:\Users\Ekaterina\Desktop\Учебные задания\3 курс\Web\WebDevelopment_2024-2025>git add "work/K3322/Ульянова Екатерин
a Алексеевна/Lab1/button_1.jpg"

C:\Users\Ekaterina\Desktop\Учебные задания\3 курс\Web\WebDevelopment_2024-2025>git commit -m "change image"
[lab1 b717342] change image
1 file changed, 0 insertions(+), 0 deletions(-)

C:\Users\Ekaterina\Desktop\Учебные задания\3 курс\Web\WebDevelopment_2024-2025>git add "work/K3322/Ульянова Екатерин
a Алексеевна/Lab1/img_1.jpg"

C:\Users\Ekaterina\Desktop\Учебные задания\3 курс\Web\WebDevelopment_2024-2025>git commit -m "add image (for lab2)"
[lab1 89140fc] add image (for lab2)
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 "work/320\2323322\320\243\320\273\321\214\321\217\320\275\320\276\320\262\320\260 \320\225\320
\272\320\260\321\202\320\265\321\200\320\270\320\275\320\260 \320\220\320\273\320\265\320\272\321\201\320\265\320\26
5\320\262\320\275\320\260\Lab1\img_1.jpg"

```

Рисунок 3 - Внесение изменений в проект

После посредством команды `git push` была осуществлена синхронизация локального и удалённого репозитория (рисунок 4), а также проверено наличие внесённых изменений на GitHub (рисунок 5).

```

C:\Users\Ekaterina\Desktop\Учебные задания\3 курс\Web\WebDevelopment_2024-2025>git push origin lab1
Enumerating objects: 28, done.
Counting objects: 100% (28/28), done.
Delta compression using up to 12 threads
Compressing objects: 100% (15/15), done.
Writing objects: 100% (21/21), 191.22 KiB | 8.31 MiB/s, done.
Total 21 (delta 7), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (7/7), completed with 3 local objects.
To https://github.com/Ekaterina-Ulyanova/WebDevelopment_2024-2025.git
956c7c9..89140fc lab1 -> lab1

```

Рисунок 4 - Синхронизация локального и удалённого репозитория

rename headings Ekaterina-Ulyanova committed 12 minutes ago	81681cb <>
change image Ekaterina-Ulyanova committed 3 minutes ago	b717342 <>
add image (for lab2) Ekaterina-Ulyanova committed 3 minutes ago	89140fc <>

Рисунок 5 - Отображение сделанных коммитов на GitHub

2. Работа с gulp

В данном упражнении требуется установить `gulp`. Однако перед этим необходимо проверить наличие `node`, `npm` и `prx`. Так как они были ранее предустановлены, в консоли успешно отразились их версии (рисунок 6).

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

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

Рисунок 6 - Проверка наличия необходимых компонентов

Далее с помощью `npm install --global gulp-cli` была установлена утилита командной строки `gulp` (рисунок 7).

```
C:\Users\Ekaterina>npm install -g gulp-cli

added 80 packages in 6s

8 packages are looking for funding
run 'npm fund' for details
```

Рисунок 7 - Установка

Также с помощью команды `npm init` было создано окружение. Процесс его создания, а также информация о нём представлены на рисунке 8.

```
C:\Users\Ekaterina\Desktop\Учебные задания\3 курс\Web\Lab2>npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See 'npm help init' for definitive documentation on these fields
and exactly what they do.

Use 'npm install <pkg>' afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (lab2) lab2
version: (1.0.0)
description: task for lab2
entry point: (index.js)
test command:
git repository:
keywords:
author: Ulyanova Ekaterina
license: (ISC)
About to write to C:\Users\Ekaterina\Desktop\Учебные задания\3 курс\Web\Lab2\package.json:

{
  "name": "lab2",
  "version": "1.0.0",
  "description": "task for lab2",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Ulyanova Ekaterina",
  "license": "ISC"
}

Is this OK? (yes) yes
```

Рисунок 8 - Создание окружения

Затем через `npm install gulp --save-dev` в окружение был добавлен gulp (рисунок 9).

```
C:\Users\Ekaterina\Desktop\Учебные задания\3 курс\Web\Lab2>npm install --save-dev gulp
added 142 packages, and audited 143 packages in 9s

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

found 0 vulnerabilities
```

Рисунок 9 - Добавление gulp в окружение

После выполнения описанных ранее шагов, был создан файл `gulpfile.js`, в котором содержится код для создания простого task по выводу сообщения 'Done!' в консоль. Код представлен на рисунке 10.

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

gulp.task('test_task', function(done) {
  console.log('Done!');
  done();
});
```

Рисунок 10 - `gulpfile.js`

Результат выполнения созданной задачи представлен на рисунке 11.

```
C:\Users\Ekaterina\Desktop\Учебные задания\3 курс\Web\Lab2>gulp test_task
[18:23:04] Using gulpfile ~\Desktop\Учебные задания\3 курс\Web\Lab2\gulpfile.js
[18:23:04] Starting 'test_task'...
Done!
[18:23:04] Finished 'test_task' after 2 ms
```

Рисунок 11 - Выполнение задачи

3. Написание программы

Для выполнения данного задания необходимо написать программу клиент, которая показывает web-страницы одна за другой из списка (в программе можно задавать адреса страниц и интервал показа).

Для этого были написаны и размещены в папке `src` файлы `index.html`, `style.css` и `script.js`. В `index.html` были использованы `<form>` (для задания

адресов страниц и интервала показа через отправку форму) и `<iform>`, позволяющий встраивать контент. Код данного файла представлен на рисунке 12.



```
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   <title>Site list</title>
7   <link rel="stylesheet" href="css/style.css">
8 </head>
9 <body>
10   <div class="container">
11     <h1>Your sites:</h1>
12
13     <form id="settings-form">
14       <input type="text" id="page-url" placeholder="Enter page URL" required />
15       <input type="number" id="interval" placeholder="Interval (ms)" min="1000" required />
16       <button type="submit">Add Page</button>
17     </form>
18
19     <ul id="site-list"></ul>
20
21     <iframe id="web-viewer" src="" frameborder="0"></iframe>
22   </div>
23
24   <script src="js/script.js"></script>
25 </body>
26 </html>
```

Рисунок 12 - index.html

Файл `script.js` содержит две функции: `updateSiteList()` для обновления списка сайтов, который отображается на веб-странице, и `startRotation()` для запуска автоматической смены сайтов. Код данного файла представлен на рисунке 13.

```

1  const siteList = document.getElementById("site-list");
2  const webViewer = document.getElementById("web-viewer");
3  const settingsForm = document.getElementById("settings-form");
4  const pageUrlInput = document.getElementById("page-url");
5  const intervalInput = document.getElementById("interval");
6
7  let sites = [];
8  let currentIndex = 0;
9  let intervalTime = 5000;
10 let intervalId = null;
11
12 function updateSiteList() {
13   siteList.innerHTML = "";
14   sites.forEach((site, index) => {
15     const li = document.createElement("li");
16     li.textContent = `${index + 1}. ${site}`;
17     siteList.appendChild(li);
18   });
19 }
20
21
22 function startRotation() {
23   if (intervalId) {
24     clearInterval(intervalId);
25   }
26   if (sites.length > 0) {
27     webViewer.src = sites[currentIndex];
28     intervalId = setInterval(() => {
29       currentIndex = (currentIndex + 1) % sites.length;
30       webViewer.src = sites[currentIndex];
31     }, intervalTime);
32   }
33 }
34
35
36 settingsForm.addEventListener("submit", (e) => {
37   e.preventDefault();
38
39   const pageUrl = pageUrlInput.value.trim();
40   const interval = parseInt(intervalInput.value.trim(), 10);
41
42   if (pageUrl && interval >= 1000) {
43     sites.push(pageUrl);
44     intervalTime = interval;
45     pageUrlInput.value = "";
46     intervalInput.value = "";
47     updateSiteList();
48     startRotation();
49   } else {
50     alert("Please enter a valid URL and interval (minimum 1000 ms).");
51   }
52 });
53
54 updateSiteList();

```

Рисунок 13 - script.js

Также был написан gulpfile.js, в котором использован плагин browser-sync, который обеспечивает автоматическую перезагрузку веб-страниц при внесении изменений в файлы. Код представлен на рисунке 14.


```

1  const gulp = require("gulp");
2  const browserSync = require("browser-sync").create();
3
4  const paths = {
5    html: "src/*.html",
6    css: "src/css/*.css",
7    js: "src/js/*.js",
8  };
9
10 function serve() {
11   browserSync.init({
12     server: {
13       baseDir: "src",
14     },
15   });
16
17   gulp.watch(paths.html).on("change", browserSync.reload);
18   gulp.watch(paths.css).on("change", browserSync.reload);
19   gulp.watch(paths.js).on("change", browserSync.reload);
20 }
21
22 exports.default = serve;

```

Рисунок 14 - gulpfile.js

Запуск и результат внесения адресов представлены на рисунках 15-16.

```

C:\Users\Ekaterina\Desktop\Учебные задания\3 курс\Web\Lab2\task_3>gulp
[02:13:41] Using gulpfile ~\Desktop\Учебные задания\3 курс\Web\Lab2\task_3\gulpfile.js
[02:13:41] Starting 'default'...
[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: src

```

Рисунок 15 - Запуск gulp task

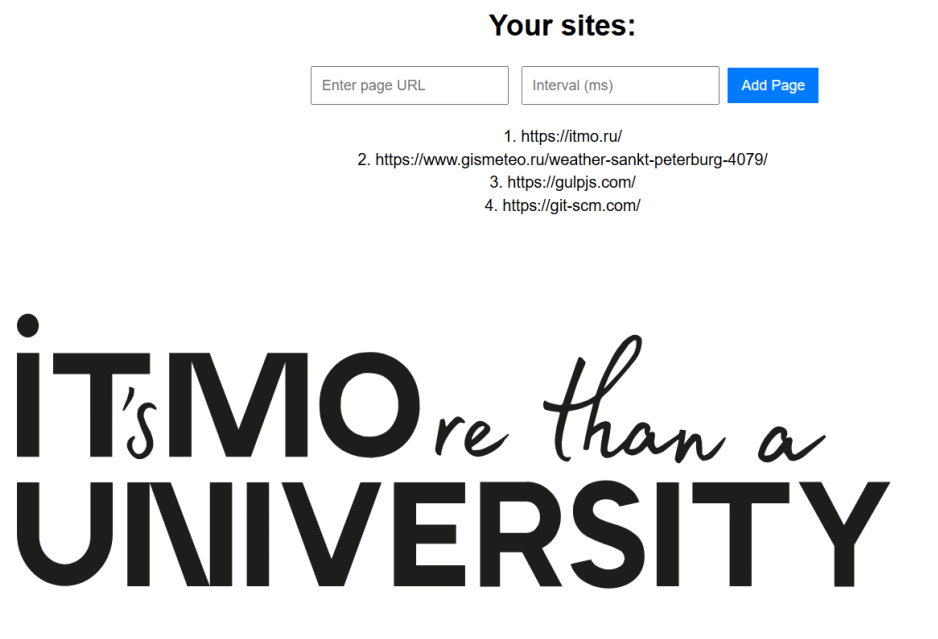


Рисунок 16 - Страница после добавления адресов сайтов

Заключение

В ходе данной лабораторной работы были приобретены основные навыки по работе с Git и gulp, а также написана программа, отображающая web-страницы из списка.