Санкт-Петербургский Национальный Исследовательский Университет

Информационных Технологий, Механики и Оптики

Факультет Инфокоммуникационных Технологий

**Web-программирование**

**Лабораторная работа 2**

Выполнил

Бордюг В. Ю.

Проверила

Марченко Е. В.

Санкт-Петербург, 2024

**Цель:** научиться работать с Git, gulp и создать программу-клиент для просмотра web-страниц.

**Задачи:**

* Изучить работу с Git – клонирование репозиториев, добавление изменений, коммиты и пуш изменений.
* Изучить Gulp и создать простой task
* Создать программу-клиент для просмотра web-страниц с возможность изменений адресов страниц и интервала показа

**Задание 1**

Был клонирован репозиторий с предыдущей лабораторной:

**Изображение выглядит как текст, снимок экрана, Шрифт, черный

Автоматически созданное описание**

Рисунок 1 – клонирование репозитория

Был совершен переход в папку проекта и была переключена ветка на lab\_1.

**Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание**

Рисунок 2 – переход в папку и переключение ветки

После этого были изменены три файла, с каждым из них был создан коммит:

**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание**

Рисунок 3 – добавление изменений и коммитов

Все три коммита отобразились:

**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание**

Рисунок 4 – просмотр коммитов

После этого все изменения были запушены в Гитхаб:

**Изображение выглядит как текст, Шрифт, снимок экрана, черный

Автоматически созданное описание**

Рисунок 5 – пуш изменений

Все коммиты действительно отобразились:

**Изображение выглядит как текст, снимок экрана, линия, чек

Автоматически созданное описание**

Рисунок 6 – коммиты на GitHub

Ссылка на репозиторий: <https://github.com/vladislav-bordiug/WebDevelopment_2024-2025/tree/lab_1>

**Задание 2**

Сначала были проверены на предмет установки node, npm, npx:

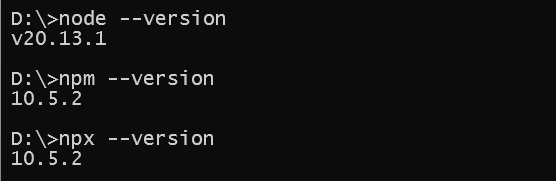
****

Рисунок 7 – проверка версий node, npm, npx

После этого была установлена утилита командной строки gulp:

**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание**

Рисунок 8 – установка gulp-cli

После этого была создана директория проекта и перемещена консоль в эту папку:

**Изображение выглядит как текст, Шрифт, снимок экрана, черный

Автоматически созданное описание**

Рисунок 9 – создание папки проекта и перемещение консоли

После этого был создан файл package.json:

**Изображение выглядит как текст, снимок экрана, Шрифт, черный

Автоматически созданное описание**

Рисунок 10 – инициализация package.json

После этого был установлен gulp в зависимостях (папка с зависимостями была удалена из пул реквеста из-за слишком большого размера, нужно будет заново прописать при клонировании этого репозитория):

**Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание**

Рисунок 11 – установка gulp в зависимости проекта

Была проверена версия gulp:

**Изображение выглядит как текст, Шрифт, снимок экрана, Графика

Автоматически созданное описание**

Рисунок 12 – проверка версии gulp

После этого был установлена gulp-htmlmin в зависимостях (папка с зависимостями была удалена из пул реквеста из-за слишком большого размера, нужно будет заново прописать при клонировании этого репозитория):

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 13 – установка gulp-htmlmin в зависимости проекта

Был создан gulpfile для минификации html (удаления лишних пробелов и удаления комментариев) и их сохранения в другую директорию:

Изображение выглядит как текст, Шрифт, снимок экрана, типография

Автоматически созданное описание

Рисунок 14 – создание и редактирование gulpfile.js

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок 15 – код gulpfile.js

Был запущен этот task;

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 16 – запуск gulp

Исходный html файл:

Изображение выглядит как текст, электроника, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 17 – исходный html файл

Полученный:

Изображение выглядит как текст, снимок экрана, программное обеспечение, компьютер

Автоматически созданное описание

Рисунок 18 – полученный сжатый html файл

Были удалены все ненужные пробелы.

**Задание 3**

Для выполнения третьего задания были созданы task\_3.html (html-разметка), task\_3.css (стили) и javascript.js (логика).

Код task\_3.html:

Изображение выглядит как текст, электроника, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 19 – код task\_3.html

Код task\_3.css:

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Автоматически созданное описание

Рисунок 20 – код task\_3.css

Код javascript.js:

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок 21 – код javascript.js

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

Рисунок 22 – код javascript.js

В результате получается такая страница, позволяющая просматривать другие страницы (которые доступны для отображения в тэге iframe):

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок 23 – task\_3.html в браузере

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок 24 – task\_3.html в браузере

Можно добавлять страницы, перемещаться между ними кнопками Prev и Next, начинать и останавливать автоматическое переключение кнопками Start и Stop, удалять ссылки кнопками Remove и устанавливать значение таймера в нижнем поле.

**Вывод:** были изучены Git, gulp и была создана программа-клиент для просмотра web-страниц.