**Министерство науки и высшего образования Российской Федерации**

федеральное государственное автономное образовательное учреждение высшего образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

Факультет Инфокоммуникационных Технологий

Направление подготовки

«Программирование в Инфокоммуникационных Системах»

Лабораторная работа №2

**«Введение в Git, Gulp»**

Выполнил:

Фатин Алексей Андреевич

Группа №К3320

Проверила:

Марченко Елена Вадимовна

Санкт-Петербург

2024

# **Введение**

**Цель:** познакомиться с базовым функционалом технологий Git, Github, Gulp.

**Задачи:**

1. Выполнить коммиты в удаленный репозиторий Git
2. Создать базовую таску автоматизации с помощью Gulp
3. Написать веб-приложение, показывающее другие страницы одну за другой

# **Ход работы**

**Задание 1. Знакомство с Git.**  
 Для сдачи лабораторной работы 1 была необходимость сделать fork репозитория преподавателя, в отдельной ветке выполнить лабораторную работу, внести изменения в свой удаленный репозиторий и создать Pull Request в главную ветку main.

Был сделан fork репозитория, при использовании графического интерфейса сайта github.com. Затем репозиторий был склонирован на рабочий компьютер локально, с помощью команды git clone. Команда git checkout с флагом -b была использована для создания отдельной ветки, в которой выполнялась лабораторная работа.

Для того, чтобы внести изменения в коммит, была использована команда git add, для создания коммита – git commit, для добавления изменений в удаленный репозиторий – git push origin <название ветки>.

Так были созданы 3 коммита в репозиторий (рис. 1).

A black rectangular object with a blue border

Description automatically generated

Рисунок 1 – отображение коммитов в Pull Request

Ссылка на Pull Request: <https://github.com/FDAPolytech/WebDevelopment_2024-2025/pull/5>

**Задание 2. Знакомство с Gulp.**

Используя документацию быстрого старта (<https://gulpjs.com/docs/en/getting-started/quick-start> ), был установлен Gulp в качестве NPM-пакета и глобально (рис. 2).

A screen shot of a computer

Description automatically generated

Рисунок 2 – установка Gulp

Используя документацию BrowserSync (<https://browsersync.io/docs/gulp> ), был создан базовый таск, который следит за изменениями в JS и HTML файлах.

Для тестирования создадим соответствующие файлы и проверим, что при изменениях в этих файлах наша страница в браузере будет перезагружаться (рис. 3).

A screenshot of a computer

Description automatically generated

Рисунок 3 – тестирование функционала перезагрузки страницы

При изменениях в файлах в терминал выводятся сообщения об обновлении (рис. 4).

A screenshot of a computer screen

Description automatically generated

Рисунок 4 – вывод в терминале при изменениях в файлах

**Задание 3. Разработка веб-приложения**

С помощью утилиты create-react-app создадим приложение с шаблоном TypeScript. С использованием связки React и MobX для упрощенной работы с данными в приложении было создано веб-приложение, в котором пользователю предлагается выбрать из 3 статей с сайта wikipedia.org, настроить длительность показа каждой из них и запустить последовательный показ. Возникла проблема с возможностью предоставления пользователю ввода ссылок, так как не каждый сайт позволяет подключаться с другого хоста и это приводит к ошибкам CORS (Cross-Origin Resource Sharing).

Была создана модель, управляющая данными и пользовательскими интеракциями. Изначально в приложение добавлено 3 статьи из Википедии, при нажатии на соответствующую кнопку веб-страница добавляется в очередь показа, а пользователю предлагается настроить, сколько секунд он хочет видеть эту страницу. Лишь при соблюдении условия, что список показа не пуст и длительность показа каждой страницы больше 0, появляется кнопка начала демонстрации веб-страниц (рис. 5).

A screenshot of a computer

Description automatically generated

Рисунок 5 – настройка показа и кнопка начала демонстрации

Далее, при помощи observable-полей нашей модели, через соответствующий интервал меняется ссылка в <iframe>, через который и происходит показ страниц (рис. 6).

A screenshot of a computer

Description automatically generated

Рисунок 6 – показ веб-страниц

Также пользователь может удалить страницы из очереди показа, если ему что-то покажется лишним, нажав на крестик.

**Заключение**

В ходе выполнения лабораторной работы были изучены базовые возможности технологий Git и Gulp. Результатом выполнения лабораторной работы стали созданные коммиты в репозитории, таск для отслеживания изменений в JS и HTML файлах в Gulp. Также было разработано веб-приложений для показа веб-страниц через интервал.