**Министерство науки и высшего образования Российской Федерации**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**ITMO University**

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2**

**По дисциплине** Web-программирование

**Обучающийся** Петрова Виктория Владимировна

**Факультет** инфокоммуникационных технологий

**Группа** К3323

**Направление подготовки** 11.03.02 Инфокоммуникационные технологии и системы связи

**Образовательная программа** Программирование в инфокоммуникационных системах

**Обучающийся**  Петрова В.В.

(дата) (подпись) (Ф.И.О.)

**Руководитель**  Марченко Е.В.

(дата) (подпись) (Ф.И.О.)

Содержание

[Содержание 2](#_Toc180431987)

[Введение 2](#_Toc180431988)

[Ход работы 2](#_Toc180431989)

[Задание 1 2](#_Toc180431990)

[Задание 2 3](#_Toc180431991)

[Задание 3 5](#_Toc180431992)

[Заключение 6](#_Toc180431993)

Введение

Цель работы — освоить навыки работы с Git для управления версиями, автоматизации задач с помощью Gulp и создания простых веб-программ на JavaScript.

Ход работы

Задание 1

Был установлен Git. Далее для настройки были прописаны ключи «user.name» и «user.email», результат представлен на рисунке 1.



Рисунок 1 – Ввод ключей «user.name» и «user.email»

В папке лабораторной работы 1 инициализирован Git репозиторий. В него добавлены все файлы первой лабораторной (Смотри Рисунок 2), далее сделаны три коммита с удалением лишних файлов (Смотри Рисунок 3)

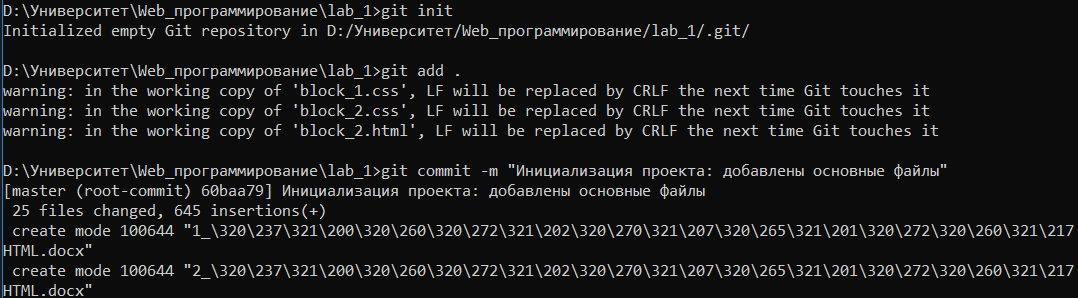


Рисунок 2 – Инициализация Git репозитория

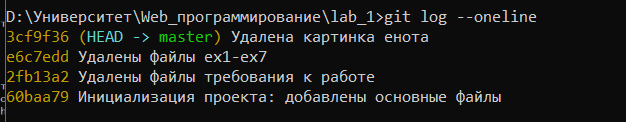


Рисунок 3 – История создания коммитов

На сайте GitHub был создан репозиторий <https://github.com/notpetrusha/ForWeb2Lab> к которому был подключен локальный репозиторий, подключение представлено на рисунке 4

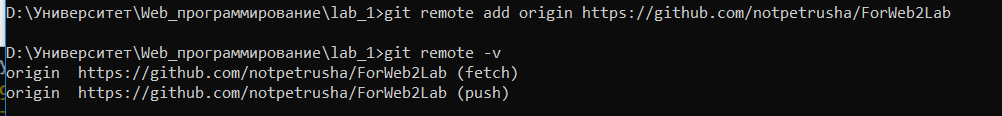


Рисунок 4 – Подключение удалённого репозитория к локальному

Результат сделанного Задания 1 представлен на рисунке 5

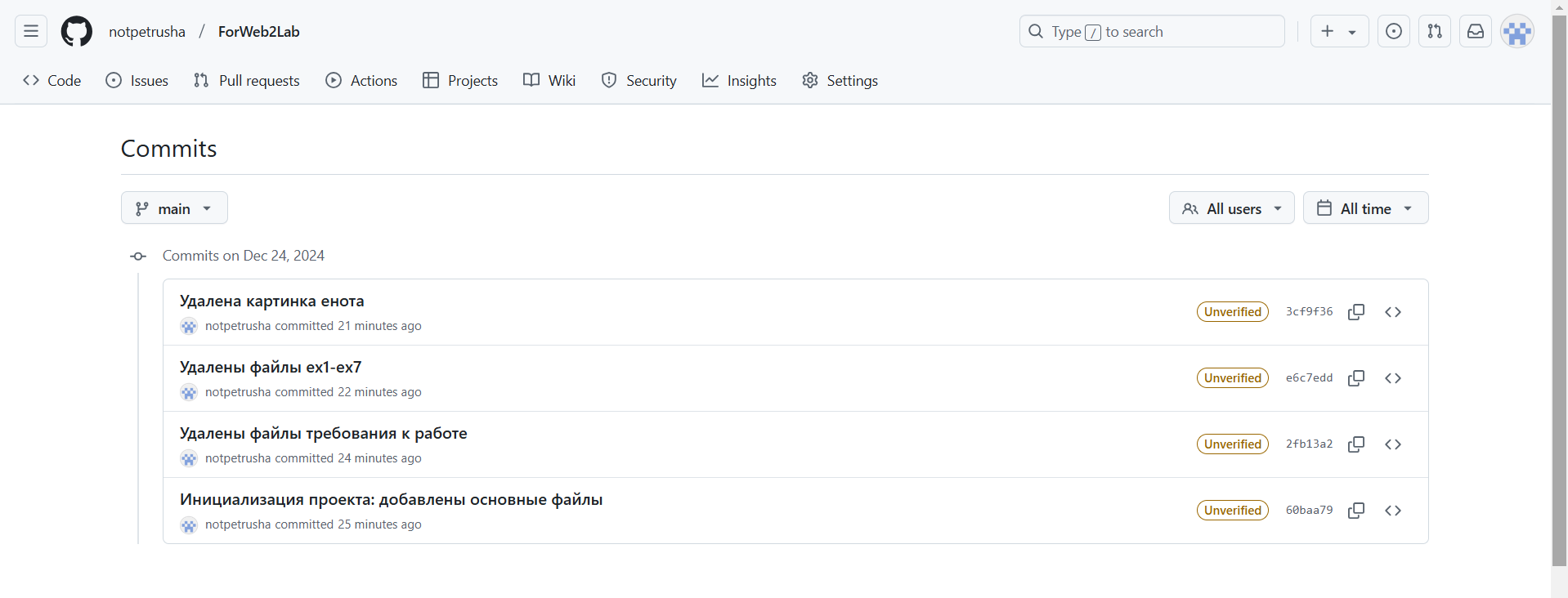


Рисунок 5 – Отображение коммитов проекта на GitHub

Задание 2

Для установки gulp также был скачан NodeJS. Версии NodeJS и npm представлены на рисунке 6.



Рисунок 6 – Версии NodeJS и npm

Далее gulp-cli был установлен глобально в систему (рисунок 7).

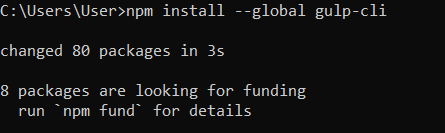


Рисунок 7 – Установка gulp-cli

Затем был инициализирован проект (рисунок 8).

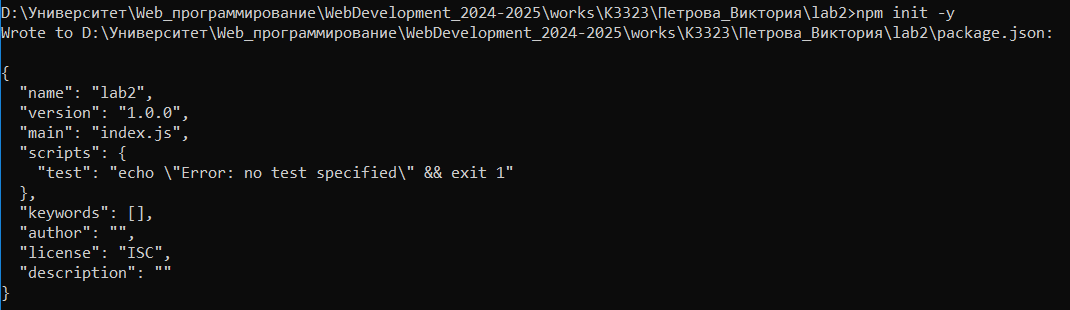


Рисунок 8 – Инициализация проекта

Далее был установлен gulp в проект (рисунок 9). Версия gulp представлена на рисунке 10.

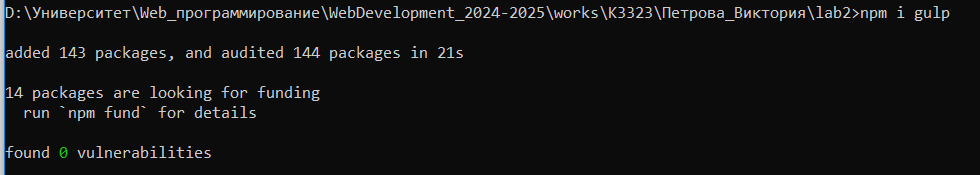


Рисунок 9 – Установка gulp в проект

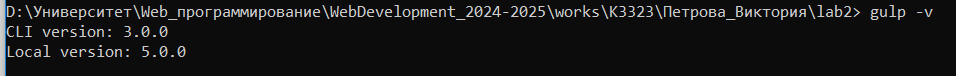


Рисунок 10 – Версия gulp

В файле gulpfile.js были созданы следующие задачи:

1. Task для работы с BrowserSync, в рамках которого происходит инициализация синхронизации браузера с заданными параметрами.
2. Task для отслеживания изменений в файлах, который автоматически перезагружает сервер при изменении содержимого.
3. Задача по умолчанию, которая запускает сервер с настройками BrowserSync.  
   Результат работы представлен на рисунке 11.

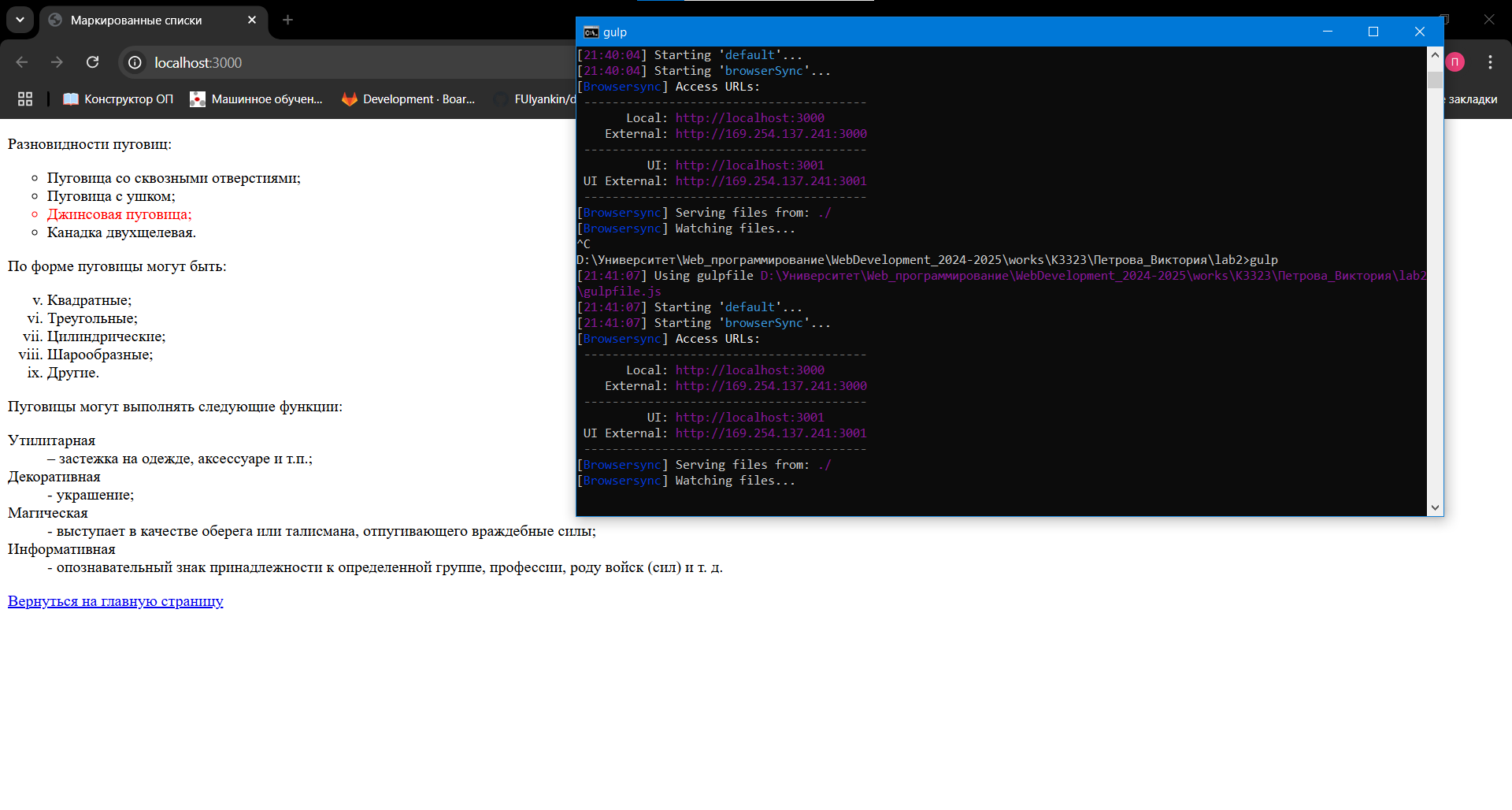


Рисунок 11 – Результат работы BrowserSync

Задание 3

В этом задании была разработана программа-клиент, которая последовательно отображает web-страницы из заранее заданного списка. На странице предусмотрены следующие элементы:

* поле для ввода URL-адреса страницы,
* кнопка для добавления ссылки в список,
* поле для указания интервала между показами страниц,
* кнопка «Отправить».

Также была реализована функциональность отображения списка добавленных ссылок. При добавлении ссылки она автоматически появляется в списке на странице, а при нажатии на ссылку — происходит ее удаление. В файле style\_url.css были заданы стили оформления страницы. Внешний вид программы показан на рисунке 12.

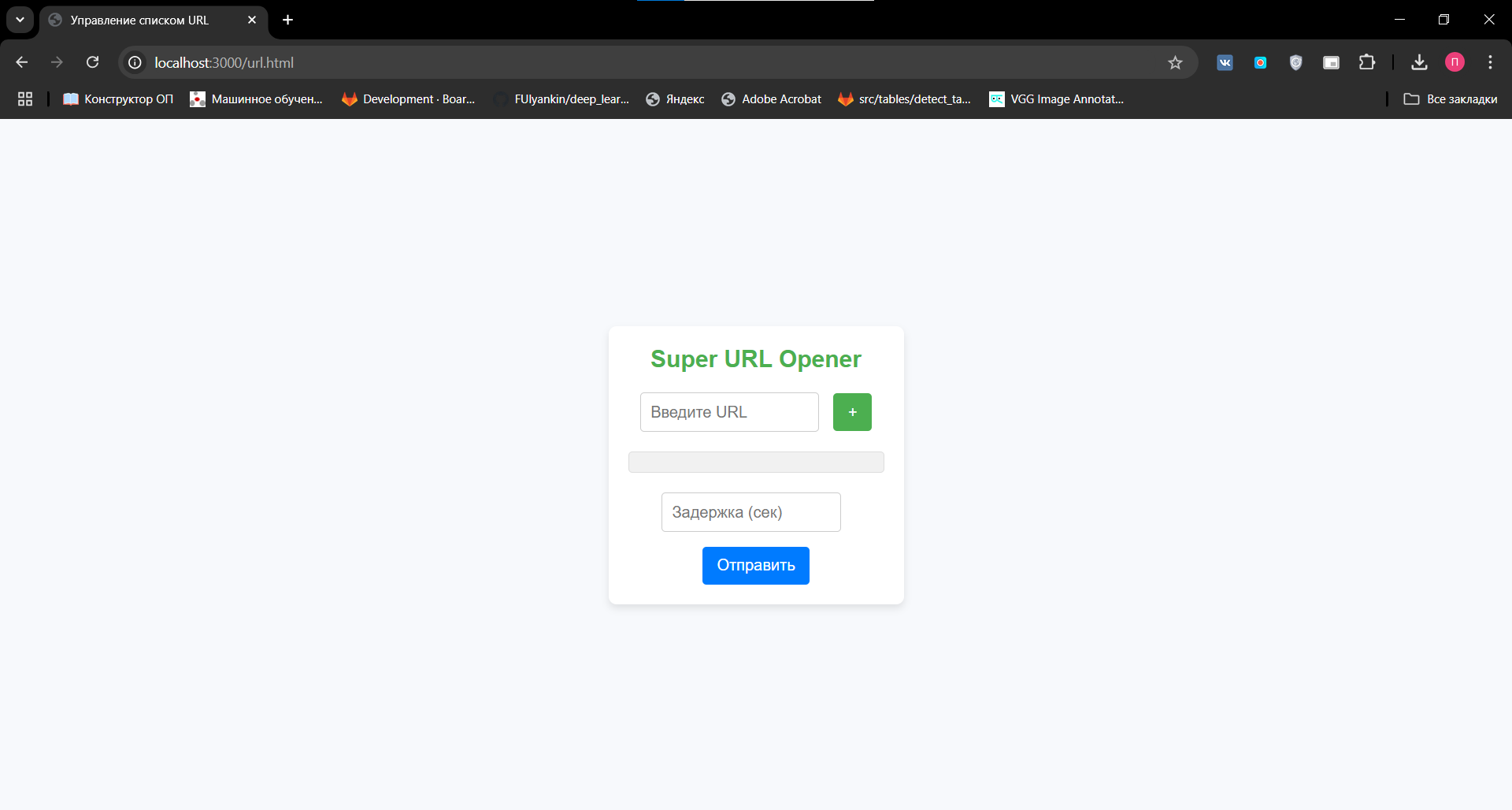


Рисунок 12 - Программа-клиент, которая показывает web-страницы по очереди из списка

Заключение

В ходе выполнения лабораторной работы были освоены основные возможности git и gulp, создан task для работы с BrowserSync, а также разработана программа-клиент, которая поочередно отображает web-страницы из списка с функциями добавления адресов и настройки интервала показа.