**Министерство науки и высшего образования Российской Федерации**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**ITMO University**

**Отчёт по лабораторной работе 2**

**По дисциплине** Web-программирование

**Тема работы** Отчёт по лабораторной работе 2

**Обучающийся** Шимченко Александра Сергеевна

**Факультет** факультет инфокоммуникационных технологий

**Группа** К3320

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

**Образовательная программа** Программирование в инфокоммуникационных системах

**Обучающийся**  Шимченко А.С.

(дата) (подпись) (Ф.И.О.)

**Руководитель**  Марченко Е.В.

(дата) (подпись) (Ф.И.О.)

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc180413911)

[ХОД РАБОТЫ 4](#_Toc180413912)

[1.1 Работа с Git 4](#_Toc180413913)

[1.2 Работа с gulp 6](#_Toc180413914)

[1.3 Программа клиент для показа web-страниц 8](#_Toc180413915)

[ЗАКЛЮЧЕНИЕ 12](#_Toc180413916)

## **ВВЕДЕНИЕ**

Цель работы: установить и поработать в Git и GULP, создать программу-клиент для просмотра web-страниц.

## **ХОД РАБОТЫ**

# **1.1 Работа с Git**

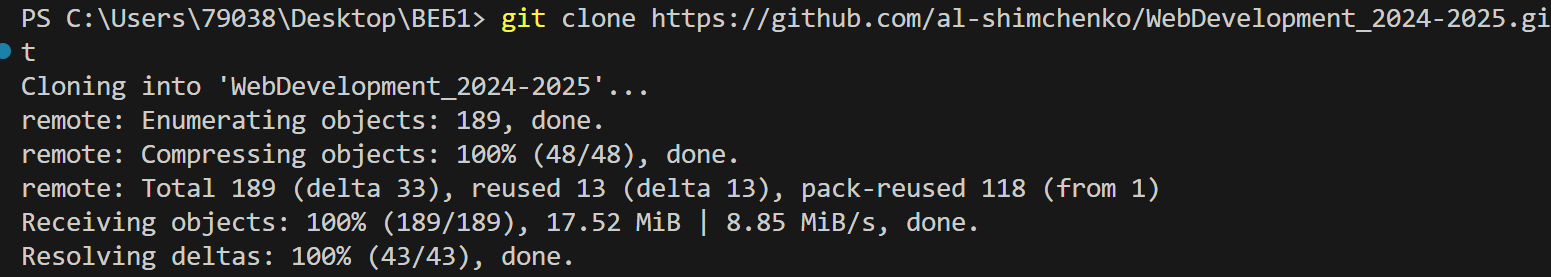
Для работы был склонирован репозиторий (рисунок 1) и совершён переход в ветку lab\_1 (рисунок 2). С помощью команды git status проверили, что никаких изменений пока не проводилось. 

Рисунок 1 — Клонирование репозитория

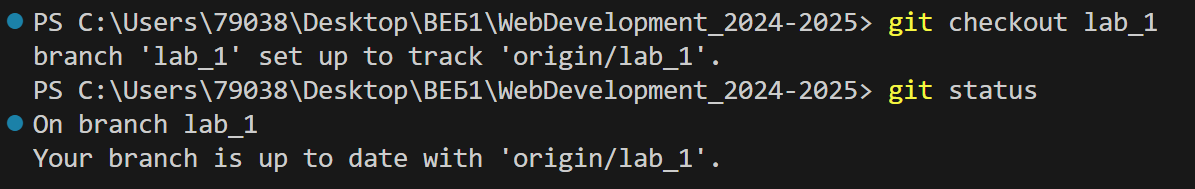


Рисунок 2 — Переход в ветку lab\_1

Изменения вносились в файл ex1.html. По очереди были сделаны 3 коммита (рисунок 3). После внесения изменений в файл применялась команда git commit, затем git add. Все изменения отправились в репозиторий с помощью команды git push.

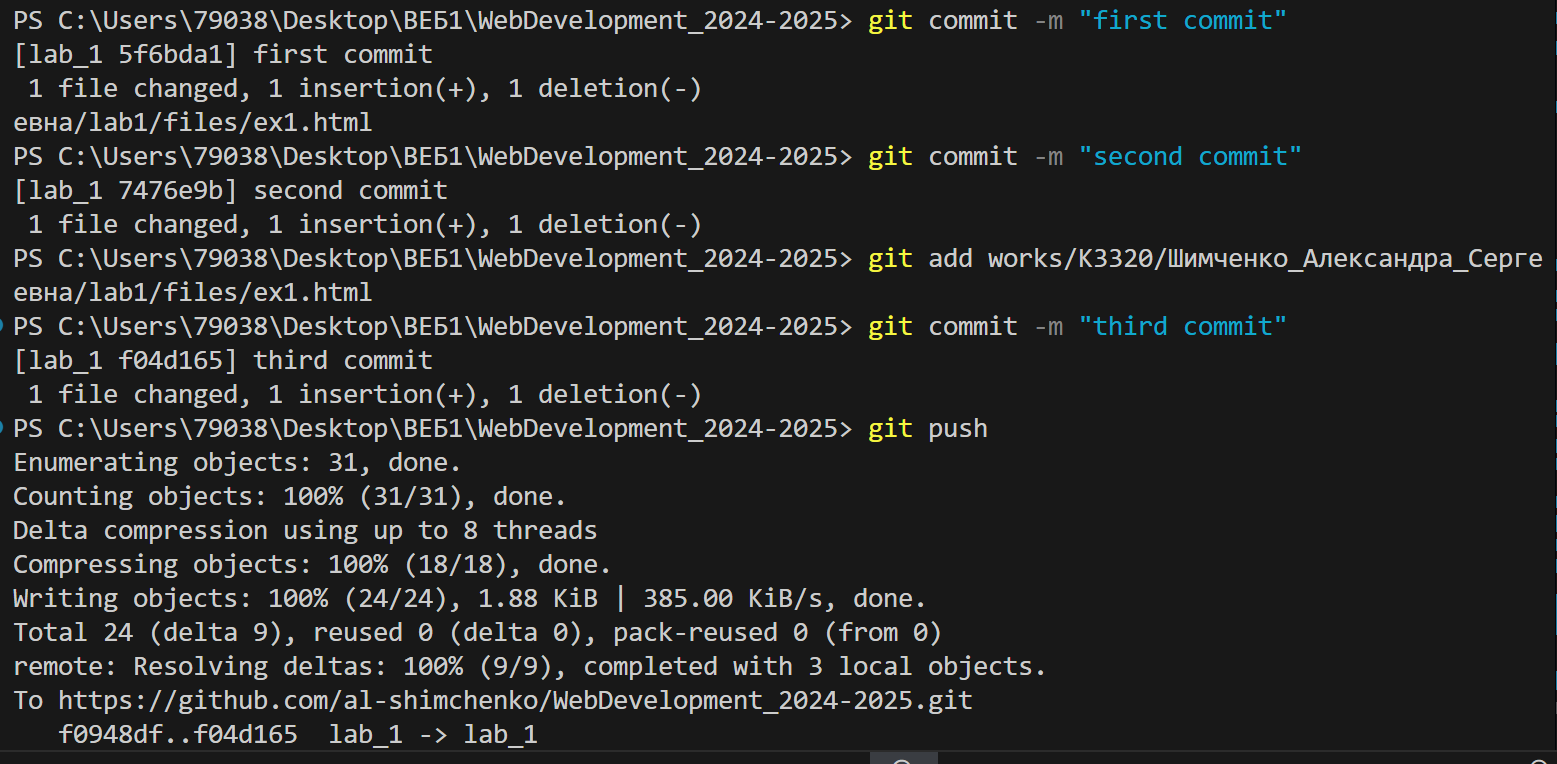


Рисунок 3 — Коммиты

Все коммиты отразились на сайте (рисунок 4).

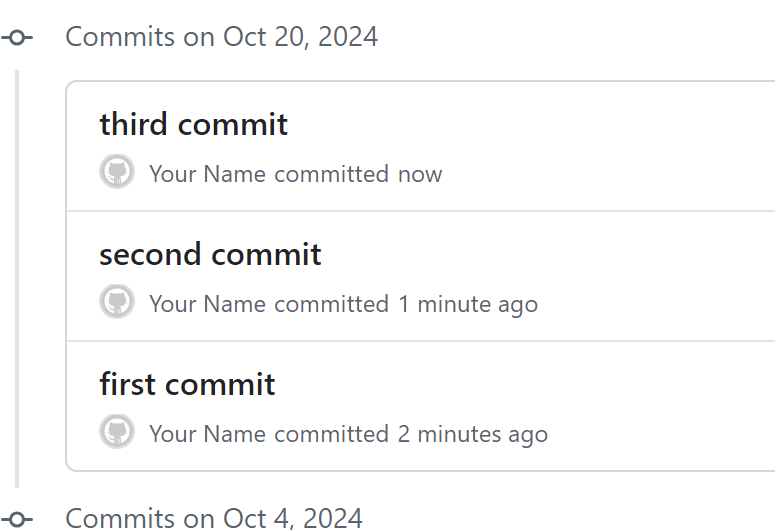


Рисунок 4 — Отображение коммитов

Проект с коммитами доступен по ссылке: <https://github.com/al-shimchenko/WebDevelopment_2024-2025/blob/lab_1/works/K3320/Шимченко_Александра_Сергеевна/lab1/files/ex1.html>

# **1.2 Работа с gulp**

В начале работы было проверено наличие установленных версий node, npm, nvm (рисунок 5). Все программы уже установлены на компьютер с актуальными версиями.

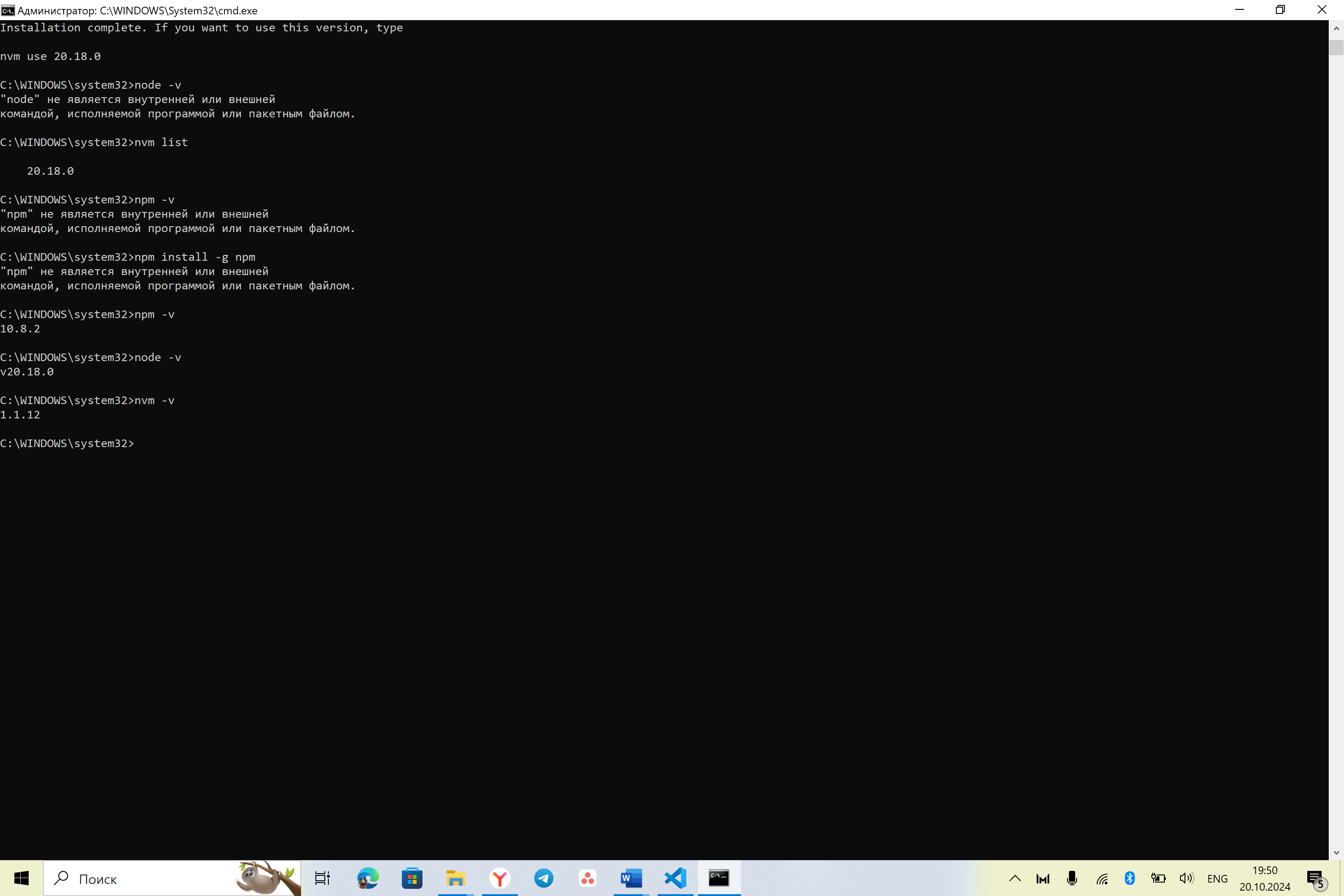


Рисунок 5 — Установленные программы

После был глобально установлен gulp для возможности его использования в любом каталоге (рисунок 6).

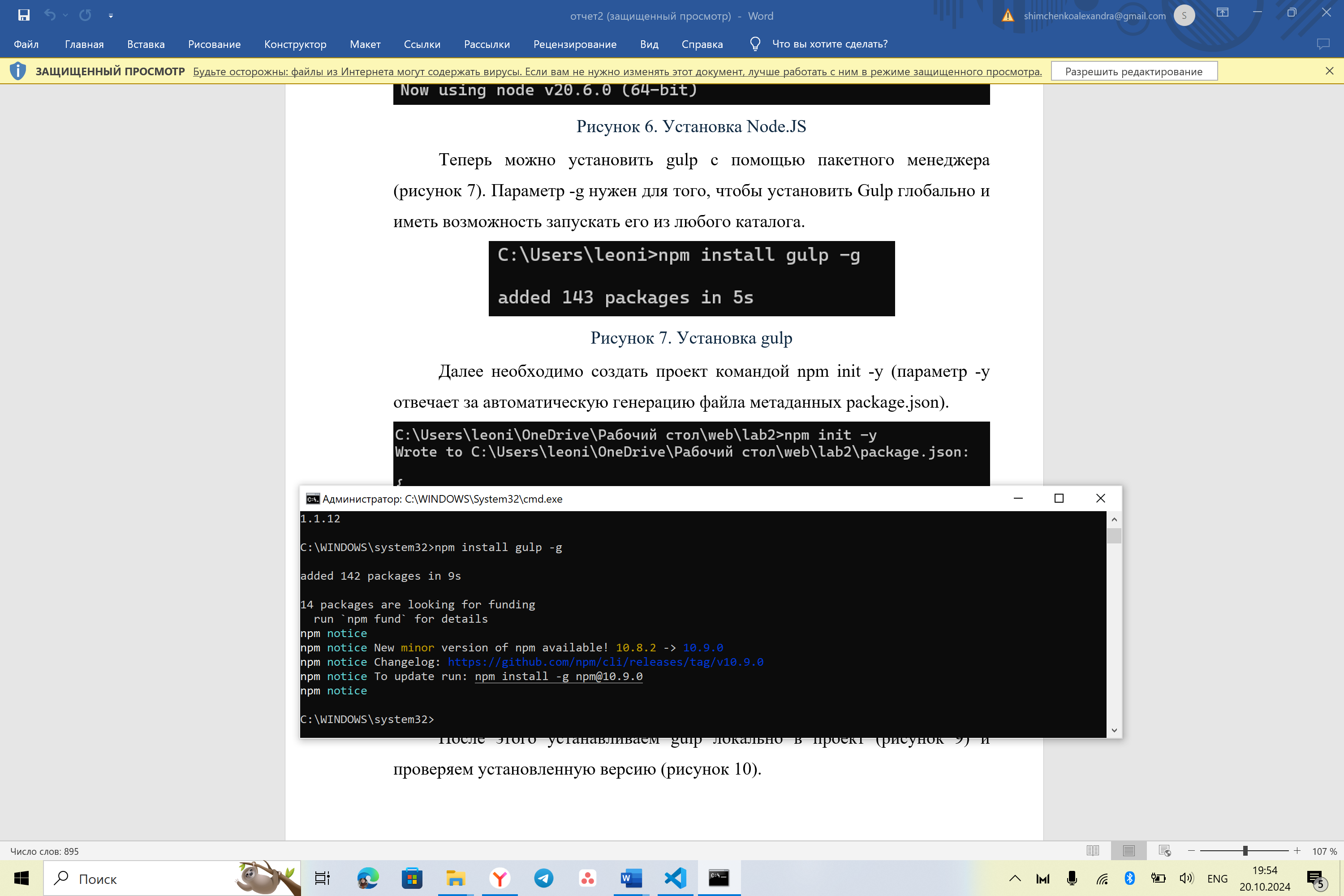


Рисунок 6 — Установка gulp

Далее с помощью команды npm init -y был создан проект, в котором автоматически был создан файл package.json (рисунок 7).

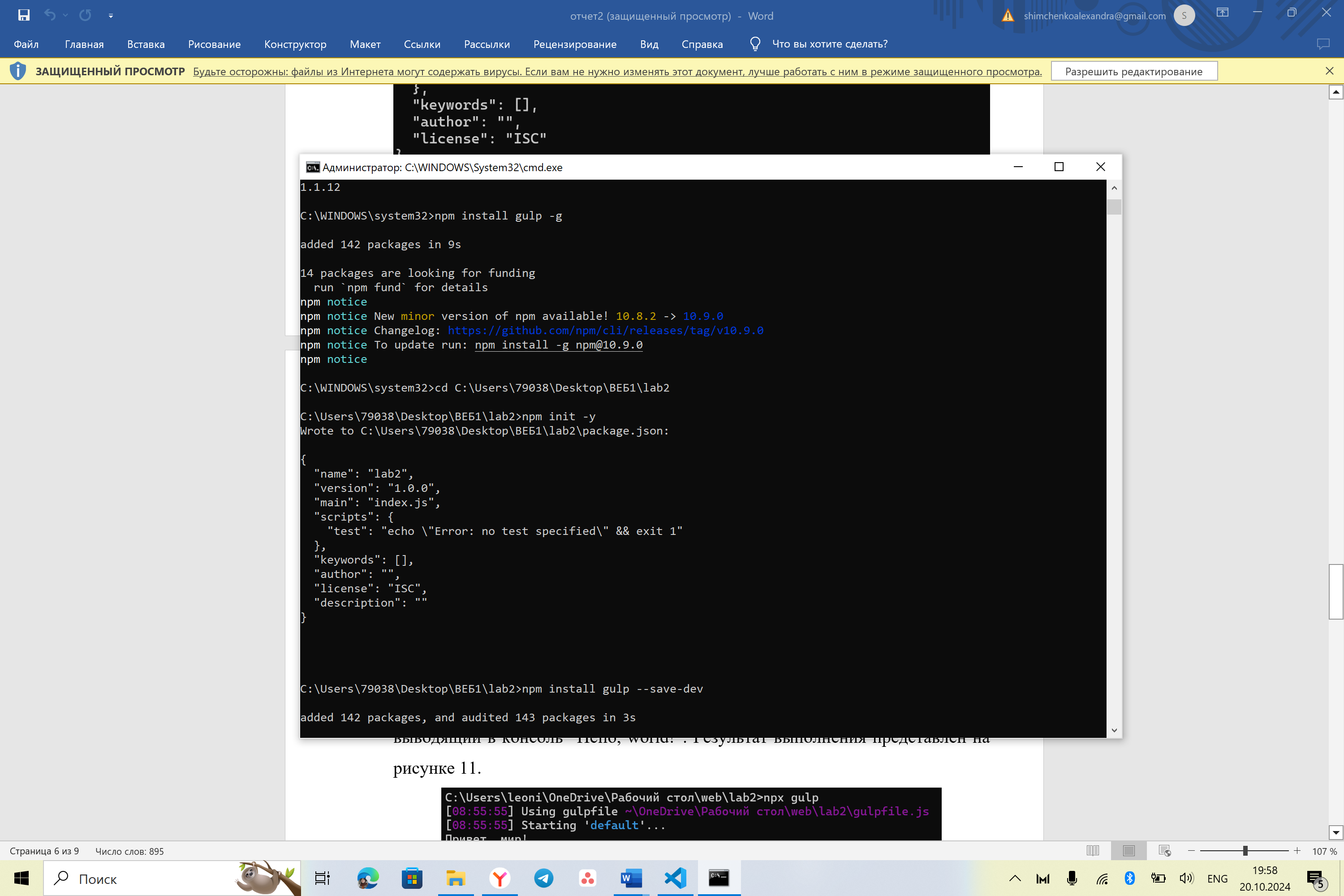


Рисунок 6 — Создание проекта

В директории проекта был создан файл, в котором прописан task (рисунок 7). При вводе команды gulp test в терминале появляется текст “Hello World”.



Рисунок 7 — Простой task

# **1.3 Программа клиент для показа web-страниц**

В задании была реализована программа для показа страниц по заданному интервалу. Был написан код index.js (рисунок 8).

Сначала был найден элемент формы на странице и добавлен обработчик события submit, который вызывал функцию onSubmit при отправке формы. Затем были инициализированы переменные: isShowing для отслеживания, показываются ли страницы, pages для хранения массива объектов с URL и временем показа, currentIndex для отслеживания текущего индекса страницы, и iframe для нахождения элемента iframe на странице.

Когда форма отправлялась, функция onSubmit предотвращала стандартное поведение отправки формы с помощью event.preventDefault(). Затем находился элемент с id site-list, и в него добавлялся новый div с текстом из первого поля формы. Объект с URL и временем из формы добавлялся в массив pages. Если страницы еще не показывались (`isShowing` было равно false`), запускался показ страниц, вызывая функцию `showNextPage. После этого форма сбрасывалась.

Функция showNextPage устанавливала src iframe на текущий URL из массива pages. Затем использовался setTimeout для ожидания указанного времени, после чего обновлялся currentIndex, и функция showNextPage вызывалась снова, создавая цикл показа страниц.

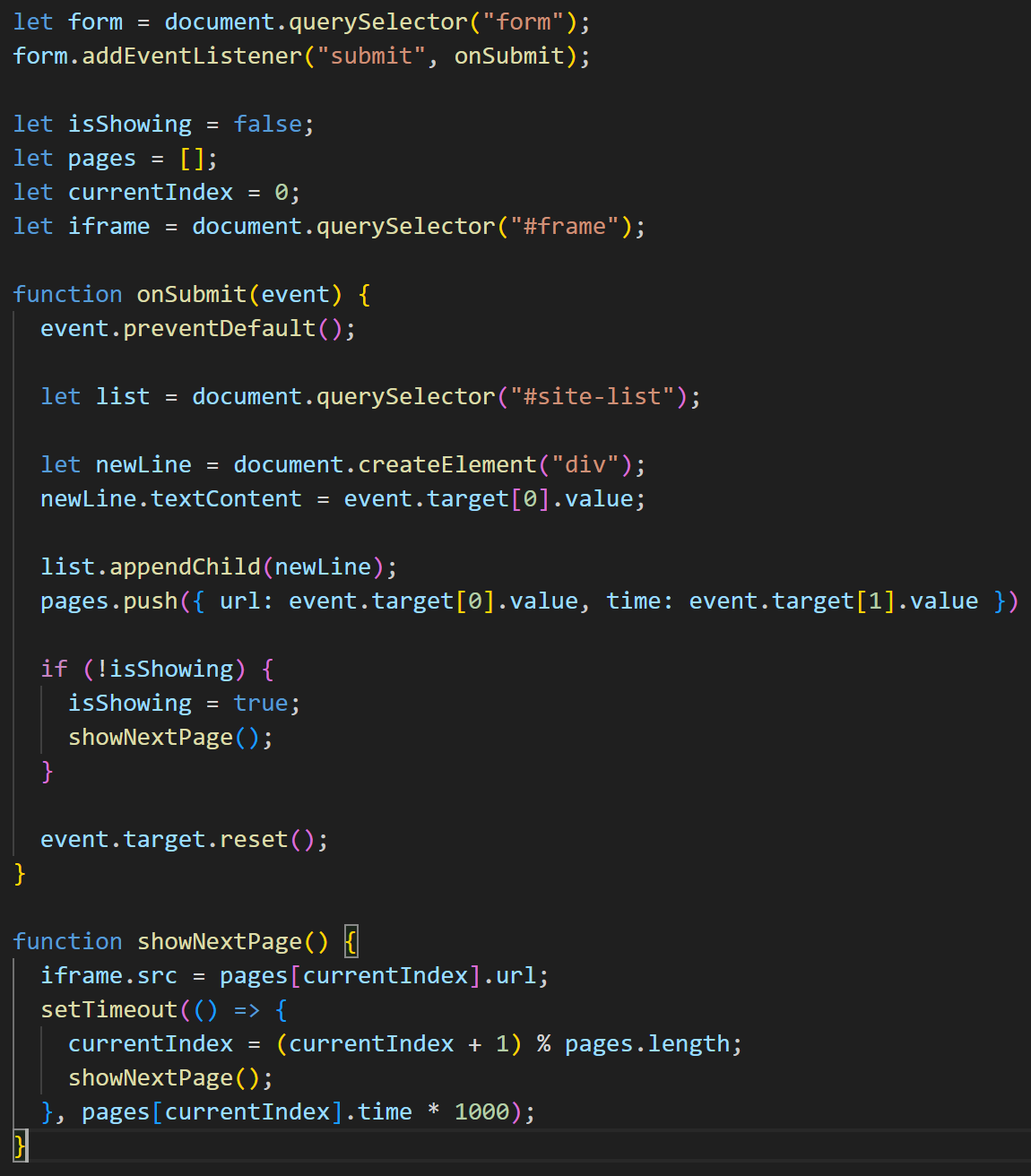


Рисунок 8 — Код index.js

Также были созданы html и css файлы для визуального оформления сайта (рисунок 9).

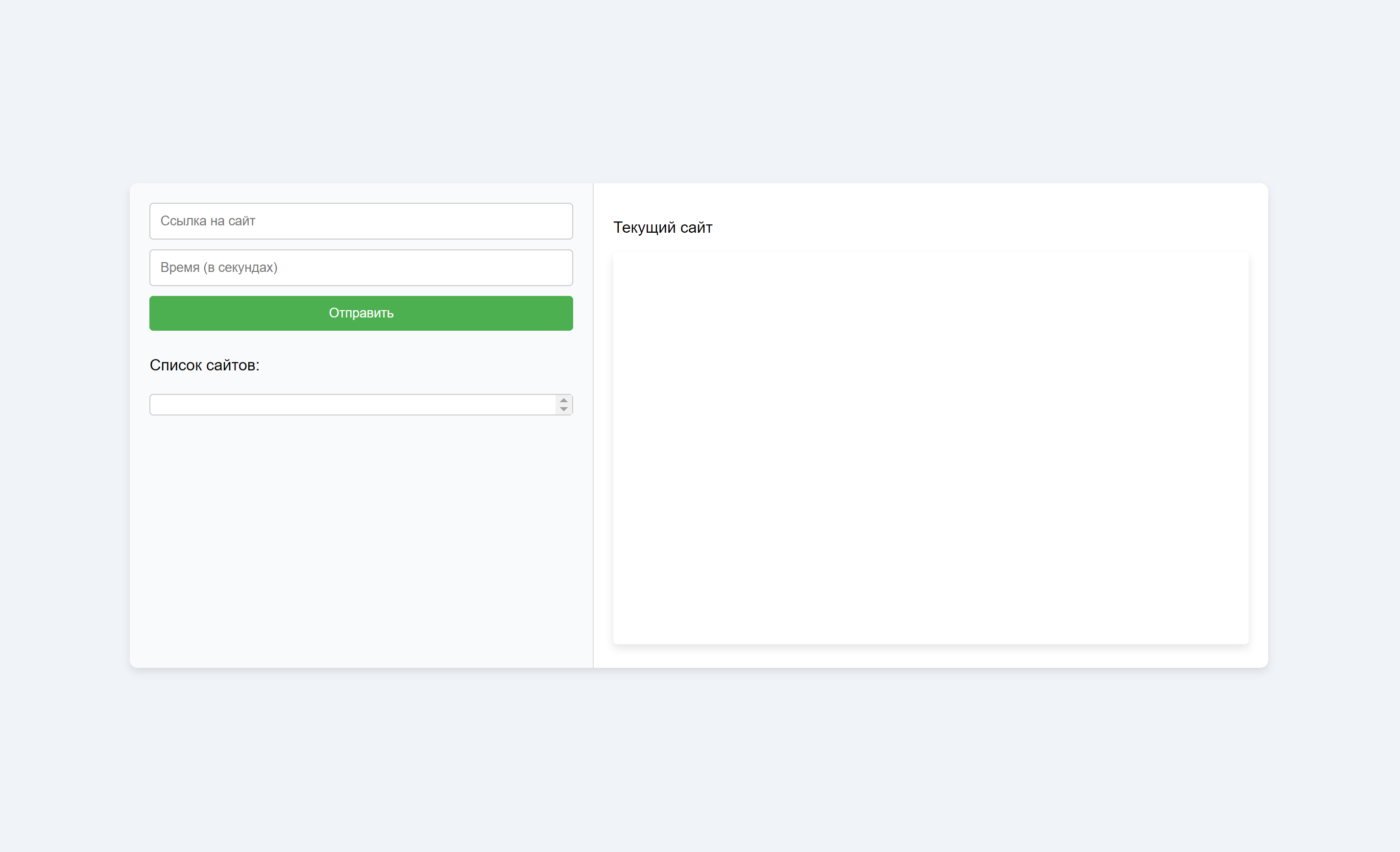


Рисунок 9 — Вид страницы

После заполнения полей формируется очередь на показ сайтов в соответствии с введённым временным интервалом (рисунки 10-11).

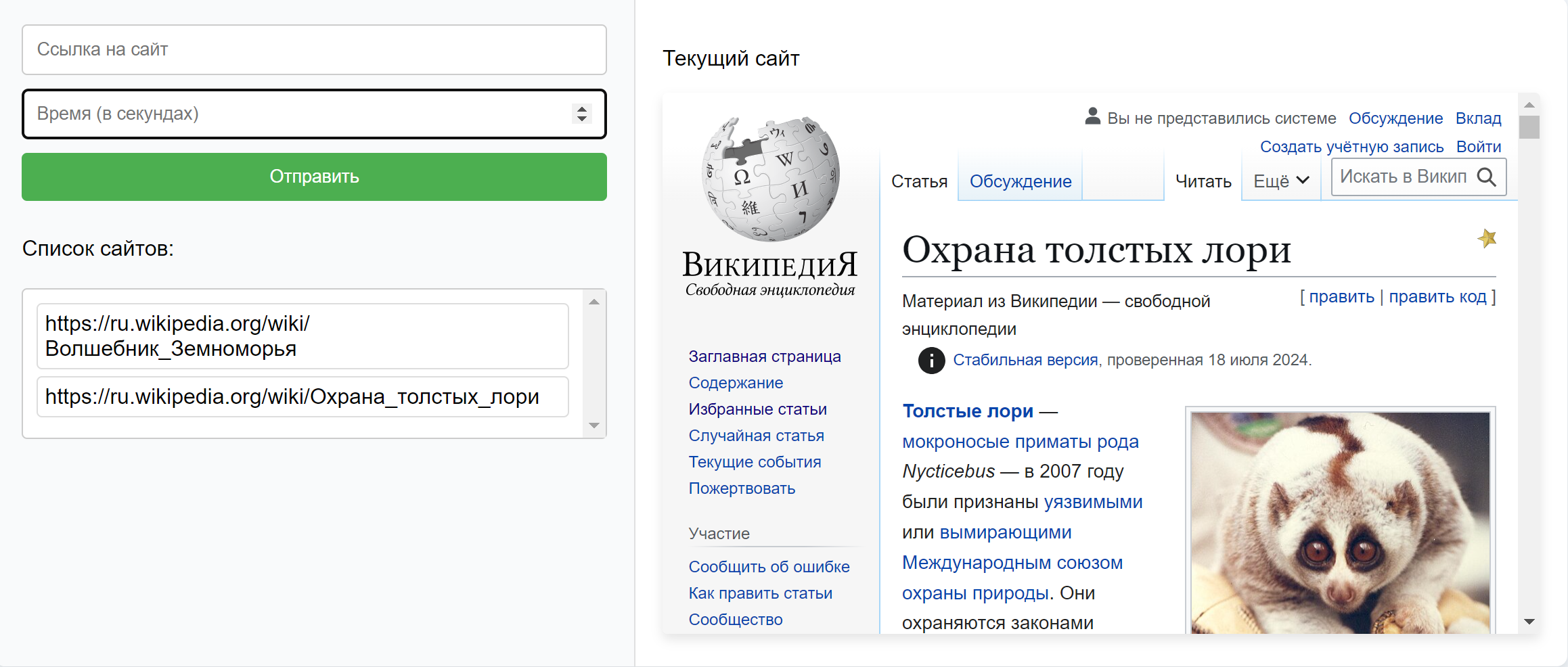


Рисунок 10 — Вид страницы

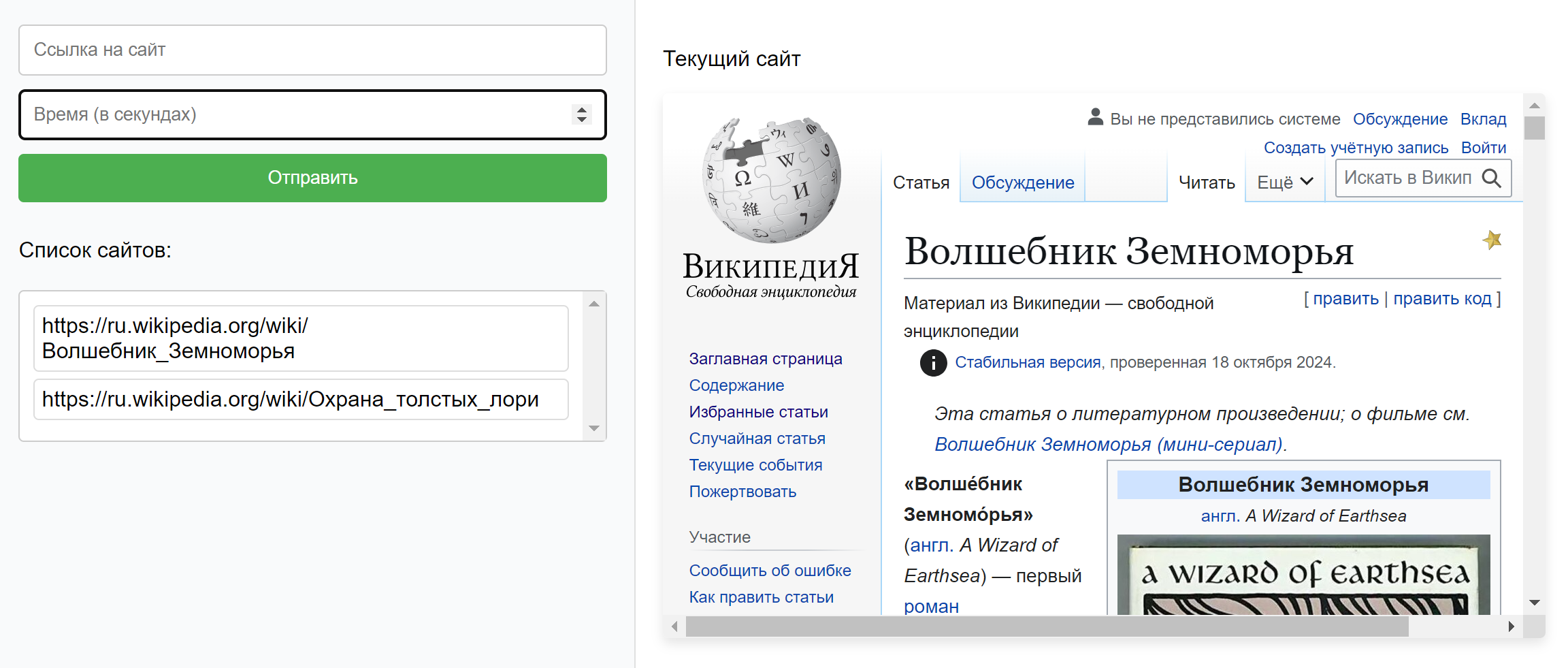


Рисунок 11 — Вид страницы

## **ЗАКЛЮЧЕНИЕ**

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