САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

Факультет инфокоммуникационных технологий

Направление подготовки «Программирование в инфокоммуникационных системах»

Лабораторная работа №2

«Использование Git и Gulp для решения задач web-разработки»

Выполнил:

Смирнов К.А.

Группа №33201

Проверила:

Марченко Е.В.

Санкт-Петербург

2023

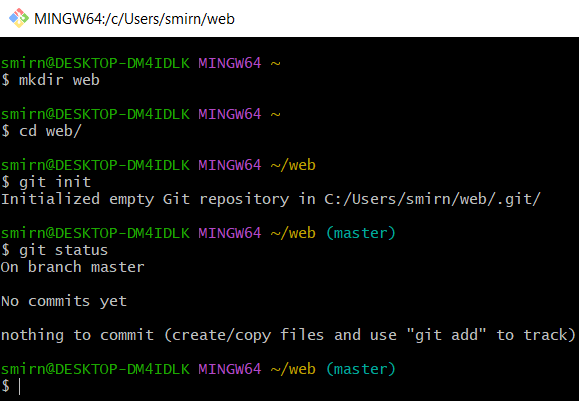
Цель работы:

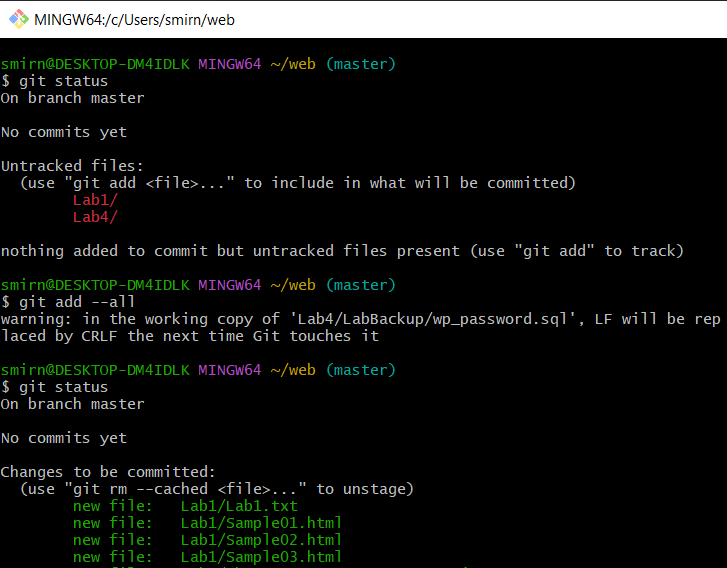
Повторить базовые правила работы с Git, создав не менее трех коммитов и синхронизировав локальный репозиторий с удаленным. Освоить установку и настройку Gulp, создав минимум одну задачу, а также рассмотреть возможность использования BrowserSync. Разработать программу-клиент, способную последовательно отображать web-страницы из предоставленного списка, предоставляя возможность задавать адреса страниц и интервал между их показом.

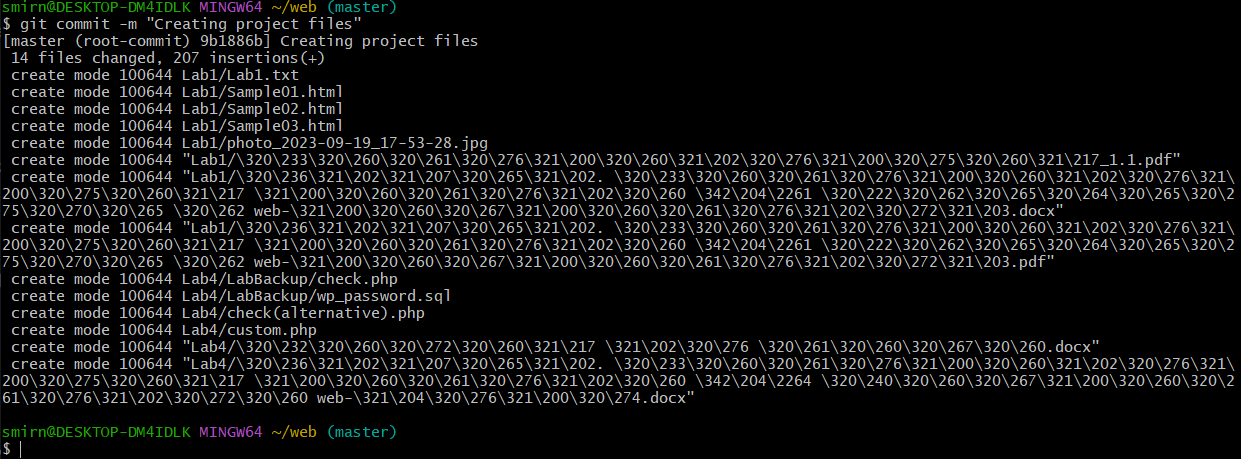
Ход работы:

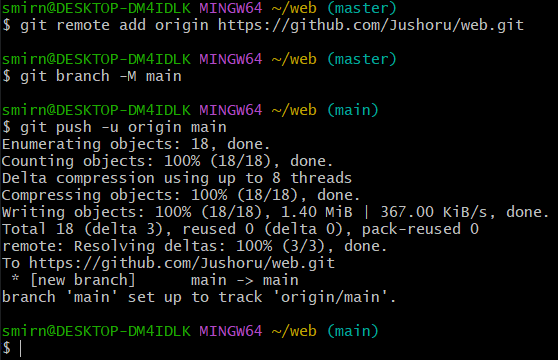
Задание 1

При помощи Git был создан локальный репозиторий - «web», который был инициализирован командой «git init» (см. рис. 1). Далее в него были добавлены две лабораторные и при этом было добавлено изменение из рабочего каталога в раздел проиндексированных файлов при помощи команды «git add -all» (см. рис. 2). Были закоммичены добавленные лабораторные работы (см. рис. 3). Создан удалённый репозиторий «web» на GitHub после чего сделано связывание локального и удаленного репозиториев при помощи команд представленных на рисунке 4. После чего в удалённом репозитории появились добавленные лабораторные и отобразился первый коммит – «Creating project files». Ссылка на репозиторий: <https://github.com/Jushoru/web.git>.

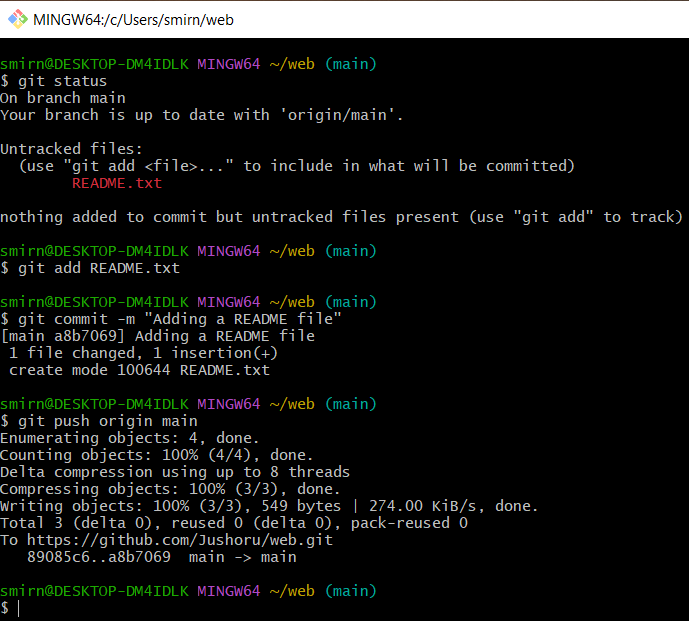
  
Рисунок 1 – создание локального репозитория

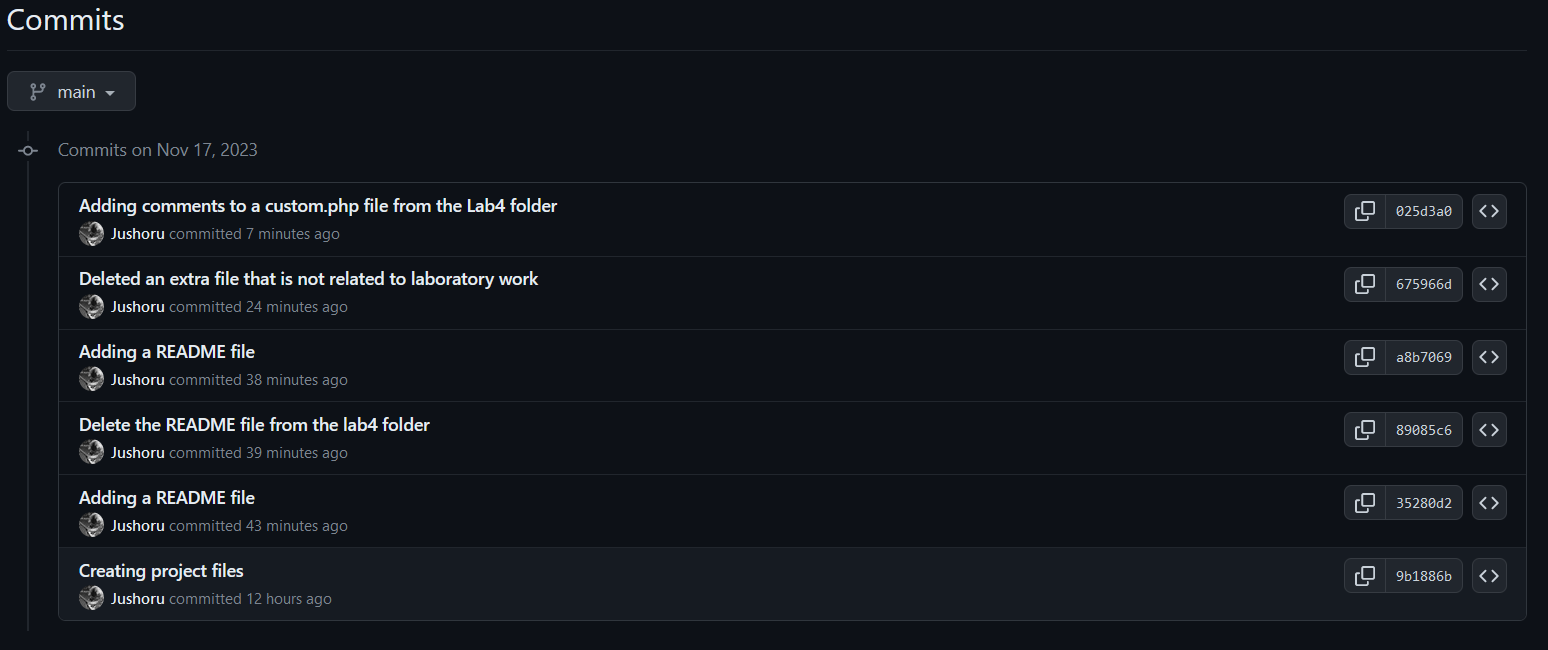
  
Рисунок 2 – создание изменений в локальном репозитории

  
Рисунок 3 – коммит добавления файлов в репозиторий

  
Рисунок 4 - связывание локального и удаленного репозиториев

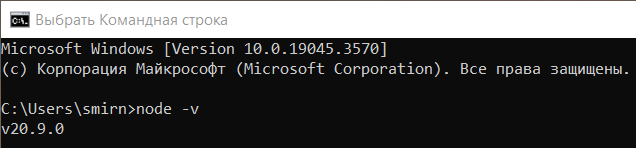
В репозиторий были добавлены изменения, которые были закоммичены и пропушены на удалённый репозиторий. 1. Был добавлен файл README (см. рис. 5). 2. Удален лишний файл из папки Lab4 3. Изменён файл custom.php в папке lab4. На рисунке 6 представлены все коммиты, связанные с удалённым репозиторием.

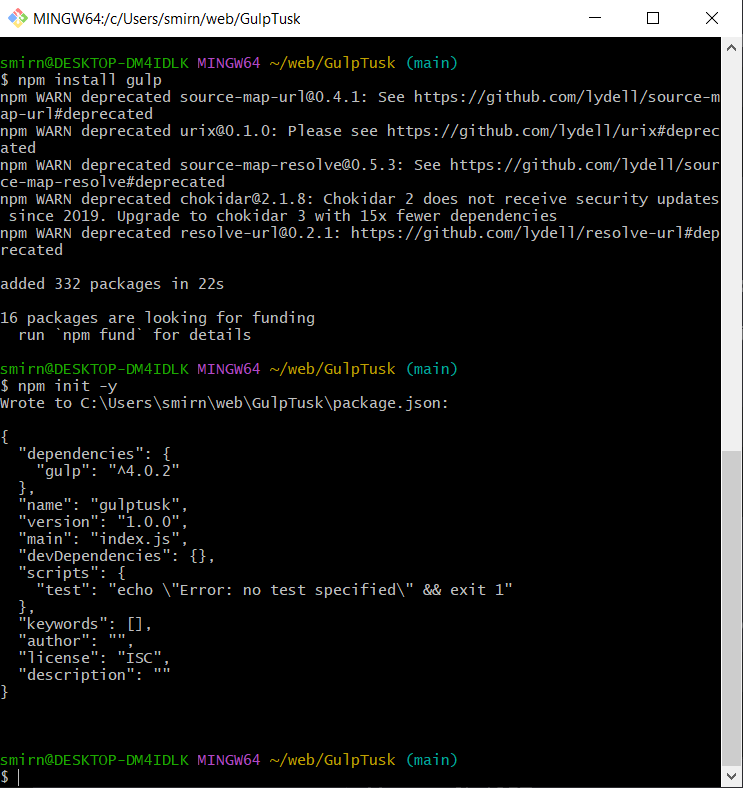
  
Рисунок 5 – добавление файла README

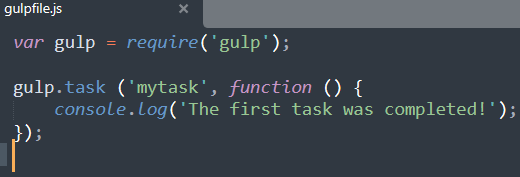
  
Рисунок 6 – коммиты

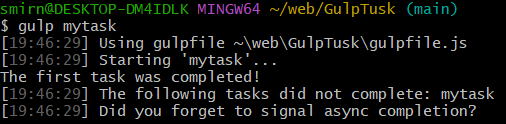
Задание 2

Для установки Gulp первым делом была произведена установка Node js. Установка прошла успешно, установленная версия проверена при помощи команды <node -v> (см. рис. 7). Установка самого Gulp делается при помощи команды, указанной на рисунке 8, после ввода команды Gulp устанавливается локально в директорию проекта. Далее был инициализирован файл package.json командой «npm init -y» (см. рис. 8), флаг «y» автоматически принимает все значения по умолчанию. После был создан файл gulpfile.js, именно этот файл будет содержать задачи для Gulp. В вышеуказанном файле была написана простейшая задача для gulp – вывод заданного сообщения в консоли (см. рис. 9). При вводе в консоли команды «gulp mytask», отображается заданное сообщение (см. рис. 10).

  
Рисунок 7 – версия node js

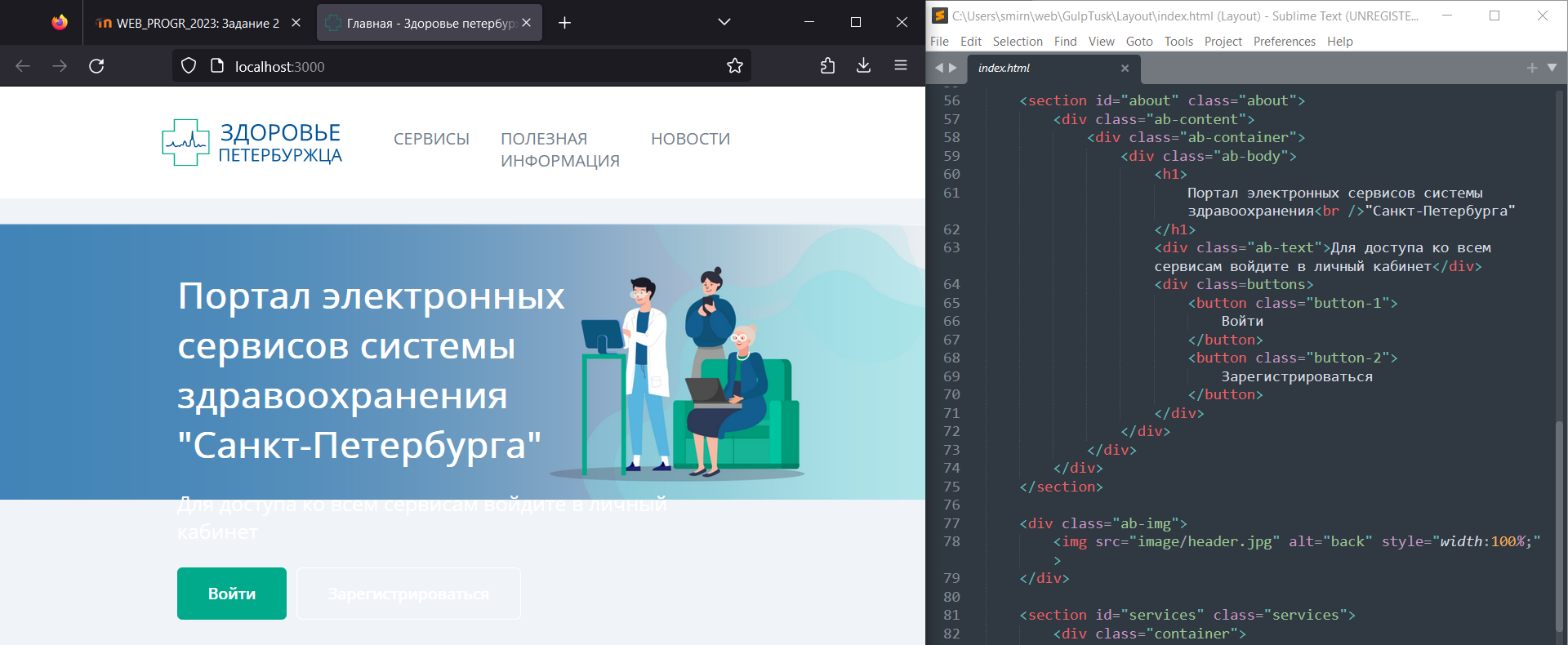
  
Рисунок 8 – установка Gulp в директорию проекта и инициализация файла package.json

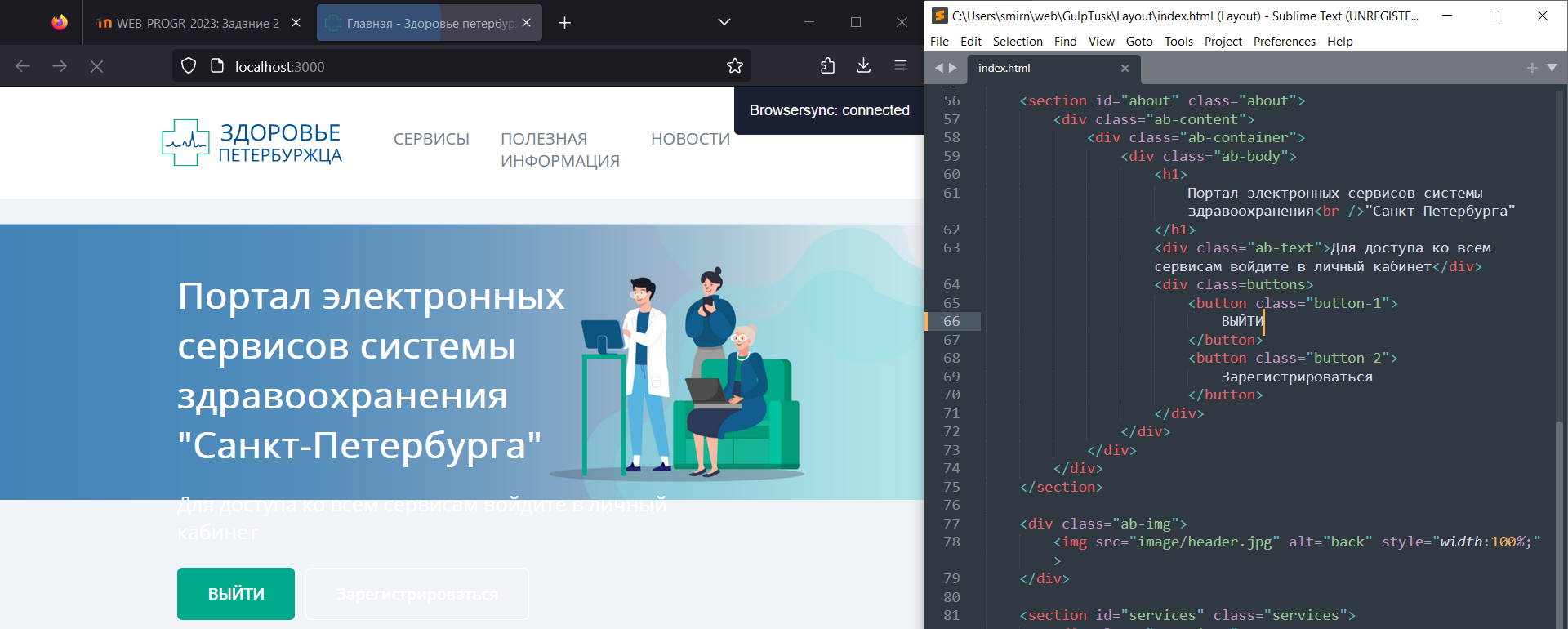
  
Рисунок 9 – код gulptask.js, предназначенный для реализации задачи Gulp

  
Рисунок 10 – Проверка работы задачи Gulp

Также был создана задача для работы с BrowserSync (Обновление web-страницы в реальном времени). Для этого был установлен плагин Browser-sync командой «npm install browser-sync --save-dev». После чего в файле gulpfile.js был написан новый код, позволяющий корректно работать плагину Browser-sync с html файлом проекта (см. рис. 11). При запуске данной gulp задачи во время сохранения изменений в файле index.html происходит автоматическое обновление html страницы в браузере (см. рис. 12-13).

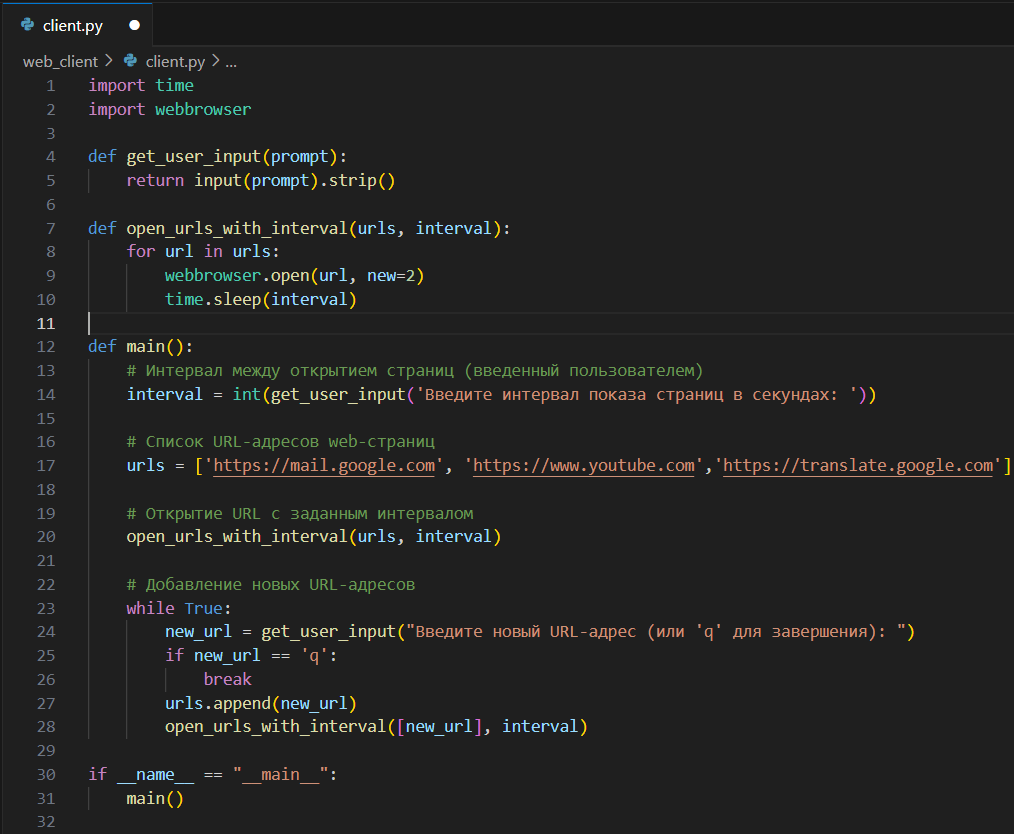
  
Рисунок 11 – код gulpfile.js

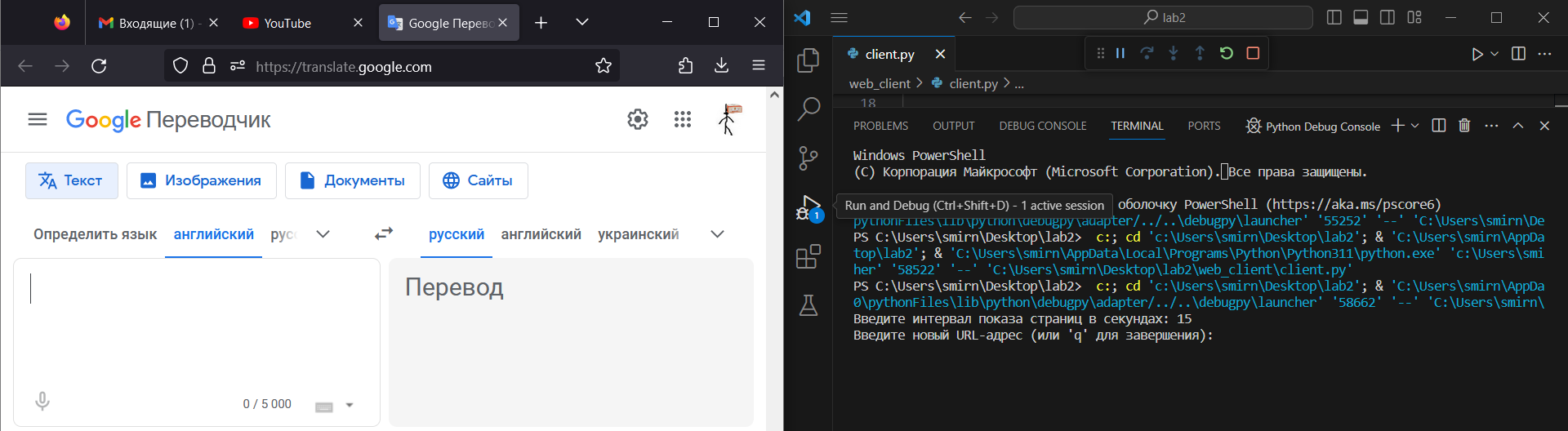
Рисунок 12 – страница до обновления данных в index.html (кнопка – «Войти»)

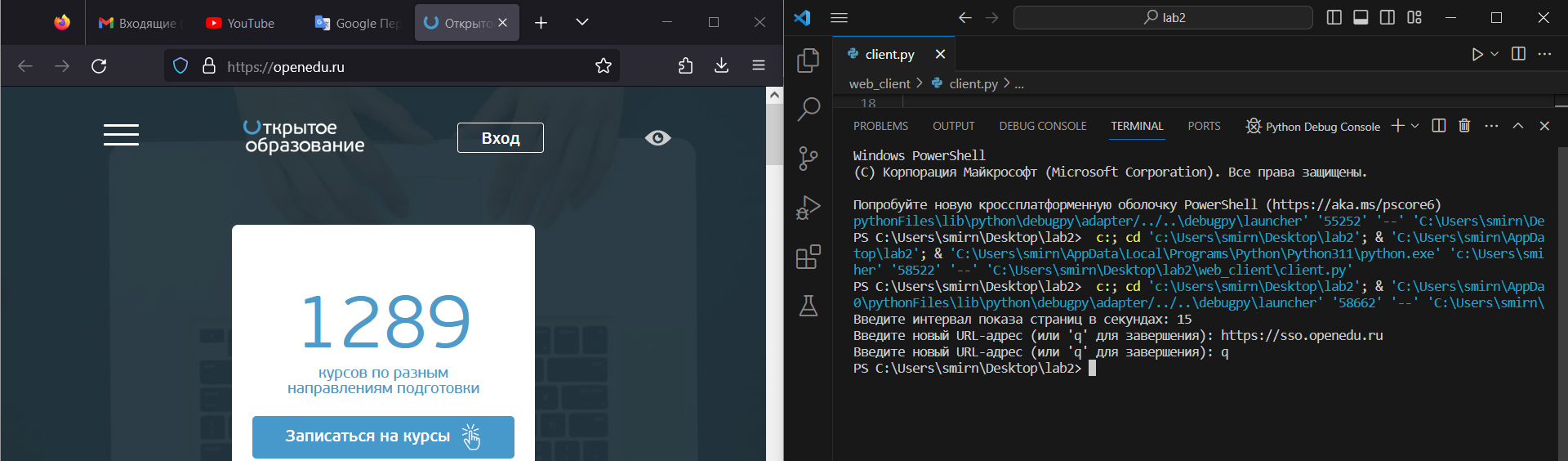
  
Рисунок 13 – страница после обновления данных в index.html (кнопка – «ВЫЙТИ»)

Задание 3

Программа-клиент для показа web-страниц (см. рис. 14) была написана при помощи языка программирования python, отладка происходила в редакторе кода VisualStudioCode. При запуске программа запрашивает интервал открытия web-страниц, после чего в браузере, установленном по умолчанию, открываются страницы, указанные в списке urls (см. рис. 15). Далее, по окончании списка, программа предлагает задать новые URL, либо завершить её выполнение при помощи написания «q» (quit) (см. рис 16).

  
Рисунок 14 – программа-клиент

  
Рисунок 15 – открытие web-страниц, заданных в списке urls

  
Рисунок 16 – открытие заданной web-страницы и завершение работы программы

Вывод

В ходе выполнения лабораторной работы были повторены основы Git, был успешно настроен и установлен Gulp и созданы две задачи: одна выводит текстовую строку в консоль, а вторая запускает HTML-проект с BrowserSync. Кроме того, была разработана программа-клиент, позволяющая последовательно отображать веб-страницы с возможностью настройки адресов и интервалов между показами. Полученные навыки обеспечивают эффективную работу в области веб-разработки. Лабораторная работа дала необходимые web-разработчику базовые навыки работы с Git, Gulp и python модулем web browser.