|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение  высшего образования  «Алтайский государственный технический университет им. И. И. Ползунова»    Факультет университетский технологический колледж  Кафедра информационных систем в экономике          Отчет защищен с оценкой  Преподаватель *С. В. Умбетов*  « » 2025 г.            Отчёт по лабораторной работе №2  по дисциплине «Производственная практика»  «Шифрование ROT13» ЛР 09.02.07.21.000 Студент группы 1ИСП-21 Д.В.Соколов | | | | |
|  |  | группа |  | и.о.,фамилия |
| Преподаватель *ассистент,* к. *т. н.* С.В.Умбетов | | | | |
| должность, ученая степень | | | | и.о.,фамилия |

БАРНАУЛ 2025

**Лабораторная работа №2**

**Шифрование ROT13**

**Цели и задачи работы**: Вспомнить основы javascript, html и css.

**Задания к работе:**

Необходимо создать html страницу и js код, страница должна быть валидной стандарту HTML5.

В коде реализовать шифрование ROT13 для четного варианта и расшифровку для нечетного

ROT13 — это простой шифр замены букв, который заменяет букву буквой, находящейся через 13 букв после неё в алфавите. Если в строку включены цифры или специальные символы, их следует вернуть в исходном виде. Смещать следует только буквы латинского/английского алфавита.

Использовать любые строковые операции например (replace, charCodeAt, fromCharCode, join, split ) запрещено. Алфавит должен быть помещен в массив. Вывод алфавита и операций шифрования\расшифровки вместе с овтет должен быт на странице.

Дизайн, слева тексе до шифрования, справа зашифрованный текст



**Задание принял:** Соколов Д.В.

Подпись ФИО

**Ход работы.**

Для последующей работы нам понадобится установленный node.js на нашем устройстве. Возьмём установочный файл с https://nodejs.org/en/download и начнём.

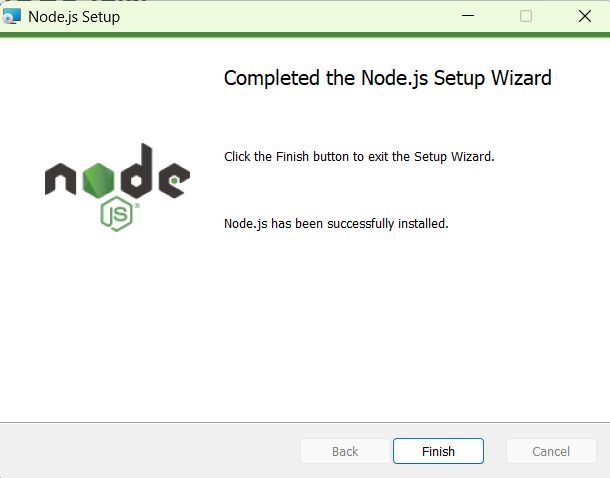


Рисунок 1 – Установка node.js.

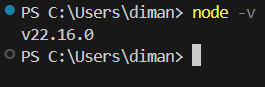


Рисунок 2 – Версия node.js.

Напишем 404.html

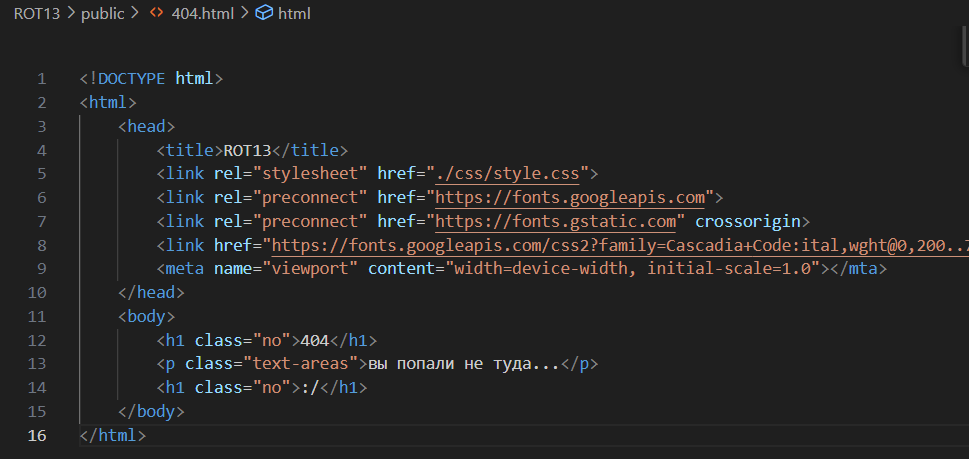


Рисунок 3 – 404.html

Изображение выглядит как белый, снимок экрана, дизайн

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 4 – 404.html

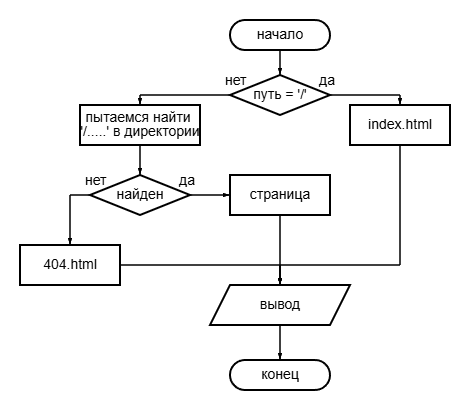


Рисунок 5 – Логика будущего server.js

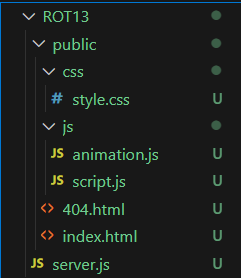


Рисунок 6 – Измененная структура проекта.

Напишем server.js, исходя из логики на Рисунке 5.



Рисунок 7 – Код server.js.

Запустим сервер командой node server.js.

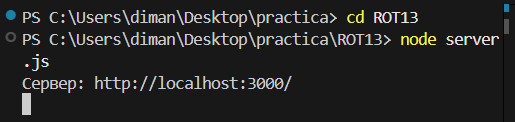


Рисунок 8 – Запуск сервера.

Протестируем:

Откроем http://localhost:3000/ в браузере.

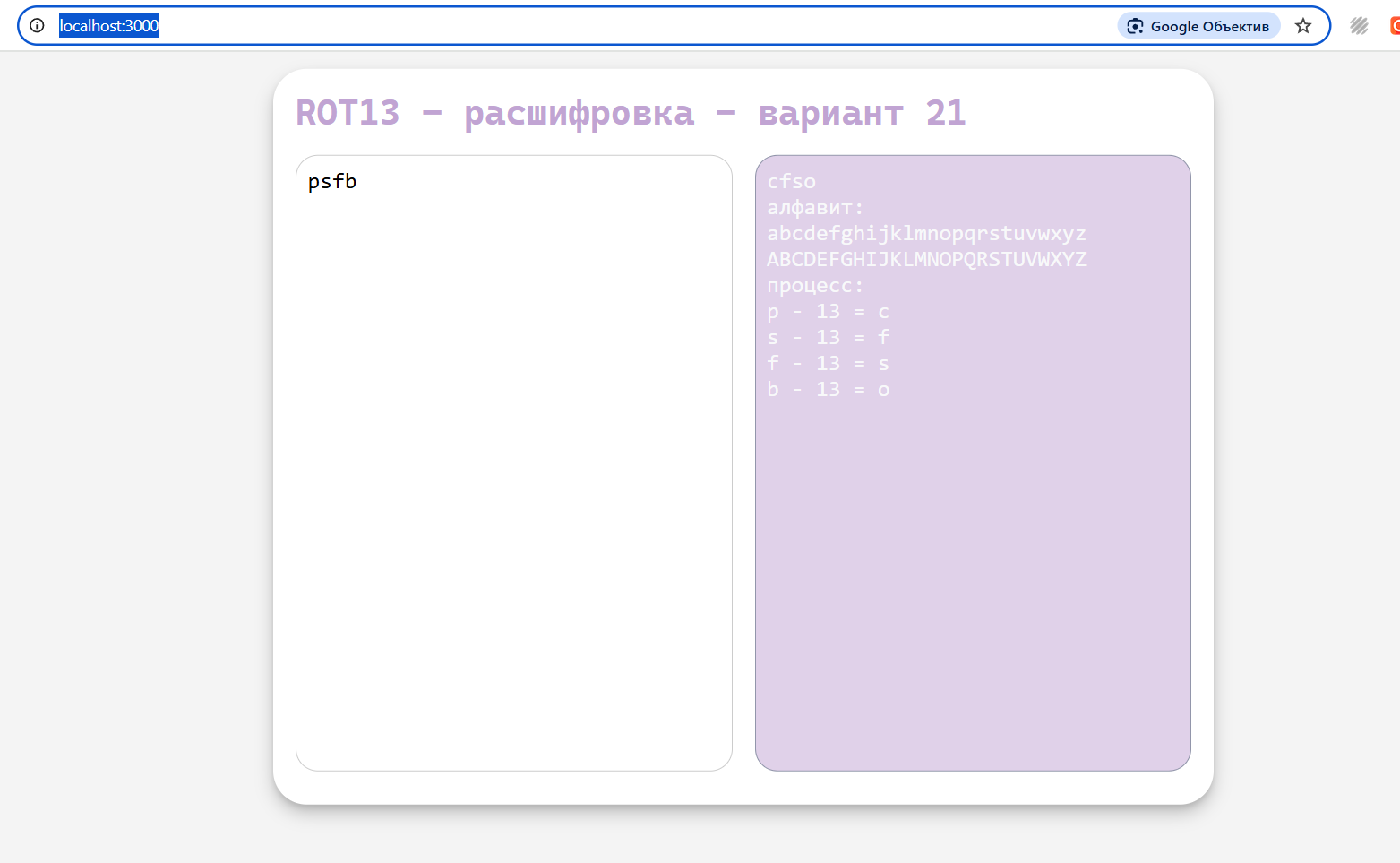


Рисунок 9 – Тест: http://localhost:3000/.

Работает! Все стили и скрипты работают, значит mime мы написали правильно. Теперь попробуем ввести другой адрес, например http://localhost:3000/qwerty

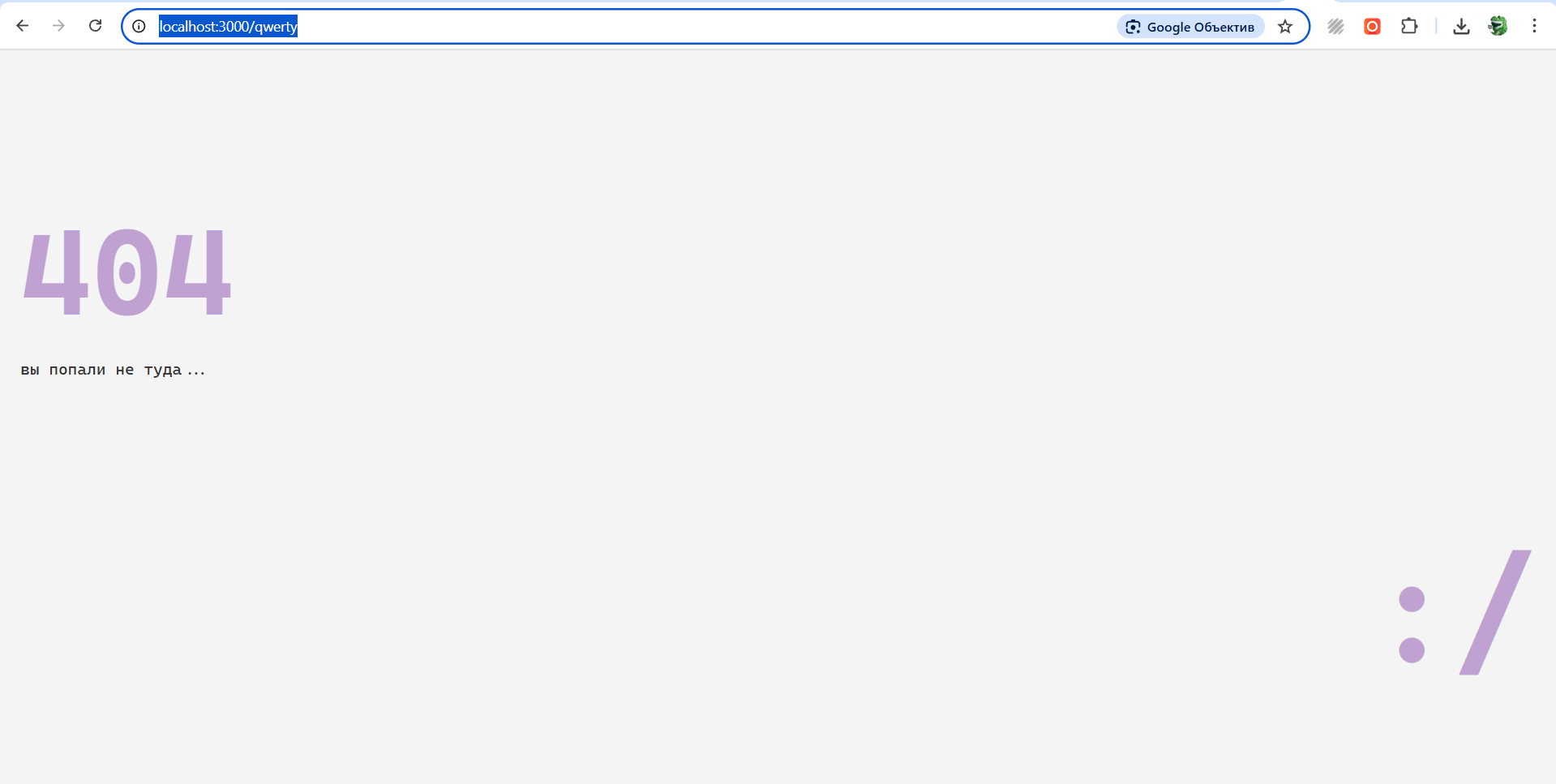


Рисунок 10 – Тест: http://localhost:3000/qwerty.

И снова всё работает правильно! Теперь попробуем создать файл qwerty.html и посмотрим, найдет ли его сервер.

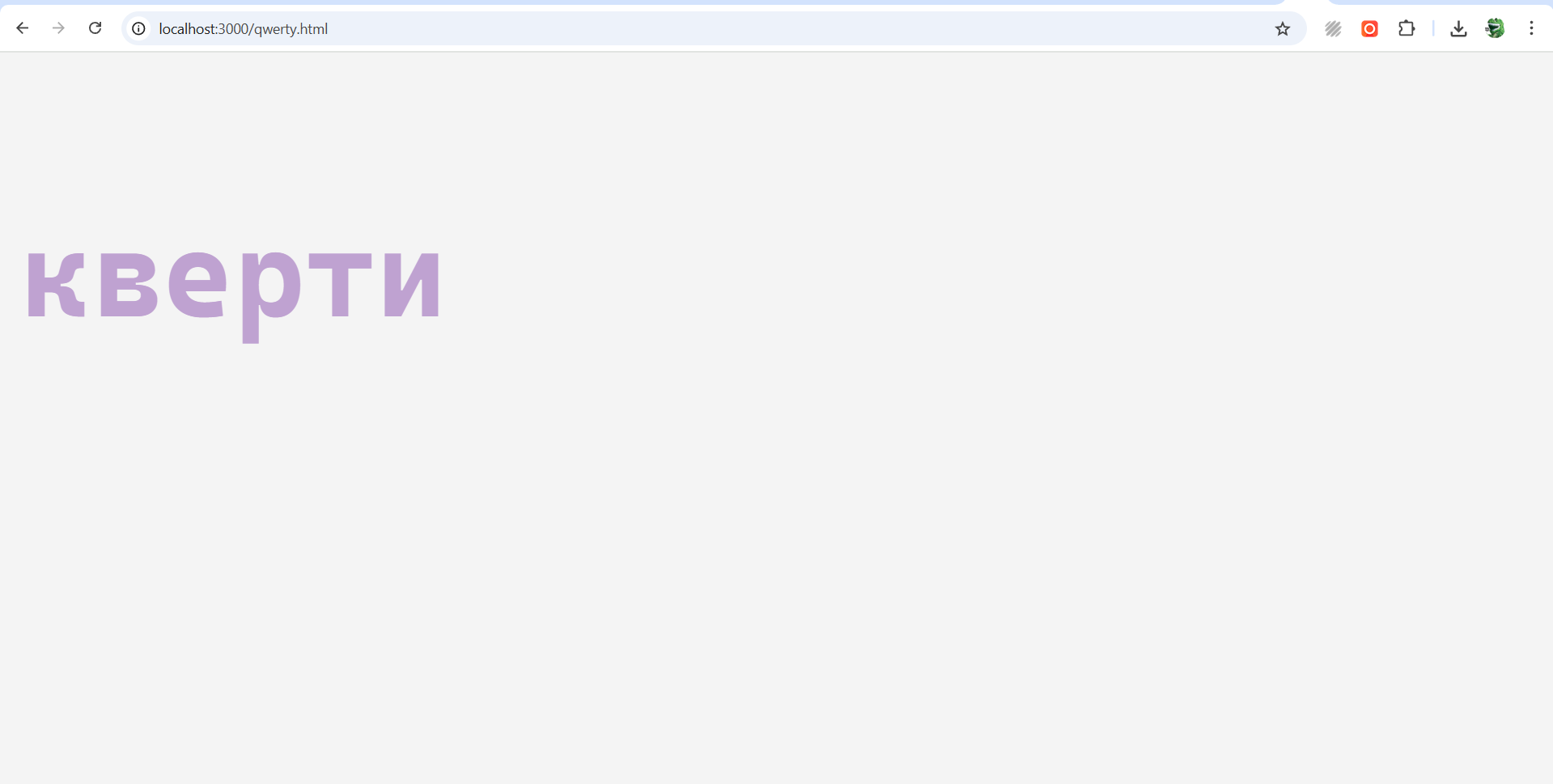


Рисунок 11 – Тест: http://localhost:3000/qwerty.html.

Сервер нашел наш новый файл, а значит всё работает исправно. Удалим этот тестовый qwerty.html.

Версии отслеживается по адресу: <https://github.com/dimasoko/lov_practika>

**Вывод**

В ходе выполнения данной практической работы я поупражнялся писать http серверы. Это было нелегко, так как в бэкэнде я разбираюсь плохо и больно. В процессе выполнения я попробовал написать самый простой сервер, для одного html и у меня получилось! А вот для решения данной задачи мне пришлось обращаться к одногруппникам за помощью. Конструкцию с mime удалось сократить до трех строк, и, в связи с этим, я благодарен javascript.

Все пункты задания реализованы и работают корректно, что подтвердили тесты.