

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Алтайский государственный технический университет им. И.И. Ползунова»

Факультет информационных технологий

Кафедра информационных технологий

Отчёт защищён с оценкой _____

Преподаватель _____ С.В. Умбетов

(подпись)

(и.о. фамилия)

« ____ » _____ 2025 г.

Отчёт по лабораторной работе №2

ЛР 12.03.01.15 000 О

Студент группы ИИСП-22 _____ Я.А. Далжин

Преподаватель к.т.н., доцент _____ С.В. Умбетов

БАРНАУЛ 2025

Задания к лабораторной работе:

1. Написать серверную часть на Node js и наладить связь с шифратором, созданным в первой лабораторной работе, при попытке перейти на любую другую страницу вывести страницу 404.

Задание принял:

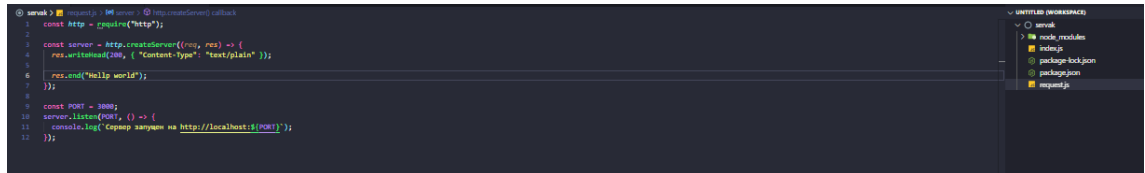


Подпись

Я.А.Далжин
ФИО

1. Написать серверную часть на Node js и наладить связь с шифратором, созданным в первой лабораторной работе, при попытке перейти на любую другую страницу вывести страницу 404.

После установки требуемых ресурсов (Node js , модуль express) , пробуем написать тестовую страничку для Node js



```
1 const http = require("http");
2
3 const server = http.createServer((req, res) => {
4   res.writeHead(200, { "Content-Type": "text/plain" });
5   res.end("Hello world");
6 });
7
8 const PORT = 3000;
9 server.listen(PORT, () => {
10   console.log(`Сервер запущен на http://localhost:${PORT}`);
11 });
```

Рисунок 1 – Код тестового сервера

Hello world

Рисунок 2 – результат

Далее начнём уже создание основного сервера, так же добавим каталог с шифратором в каталог с сервером, что бы было удобнее до него тянуться.

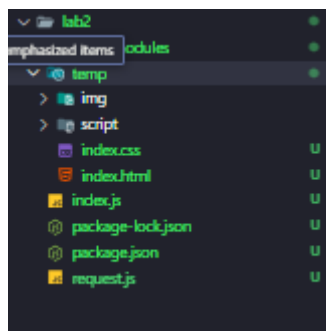


Рисунок 3 – Каталог с сервером

```

const fs = require("fs");
const http = require("http");
const path = require("path");

const server = http.createServer((req, res) => {
  const url = req.url === "/" ? "/index.html" : req.url;
  const filePath = path.join(__dirname, "temp", url);

  const extname = path.extname(url);
  const mimeTypes = {
    ".html": "text/html",
    ".css": "text/css",
    ".js": "application/javascript",
    ".png": "image/png",
    ".jpg": "image/jpeg"
  };
  const contentType = mimeTypes[extname] || "text/html";

  fs.readFile(filePath, (err, content) => {
    if (err) {
      if (err.code === "ENOENT") {
        fs.createReadStream(path.join(__dirname, "temp/page/404.html")).pipe(res);
      } else {
        res.writeHead(500);
        res.end("Ошибка сервера");
      }
    } else {
      res.writeHead(200, { "Content-Type": contentType });
      res.end(content);
    }
  });
});

const PORT = 3000;
server.listen(PORT, () => {
  console.log("Сервер запущен на http://localhost:${PORT}");
});

```

Рисунок 4 - Новый код

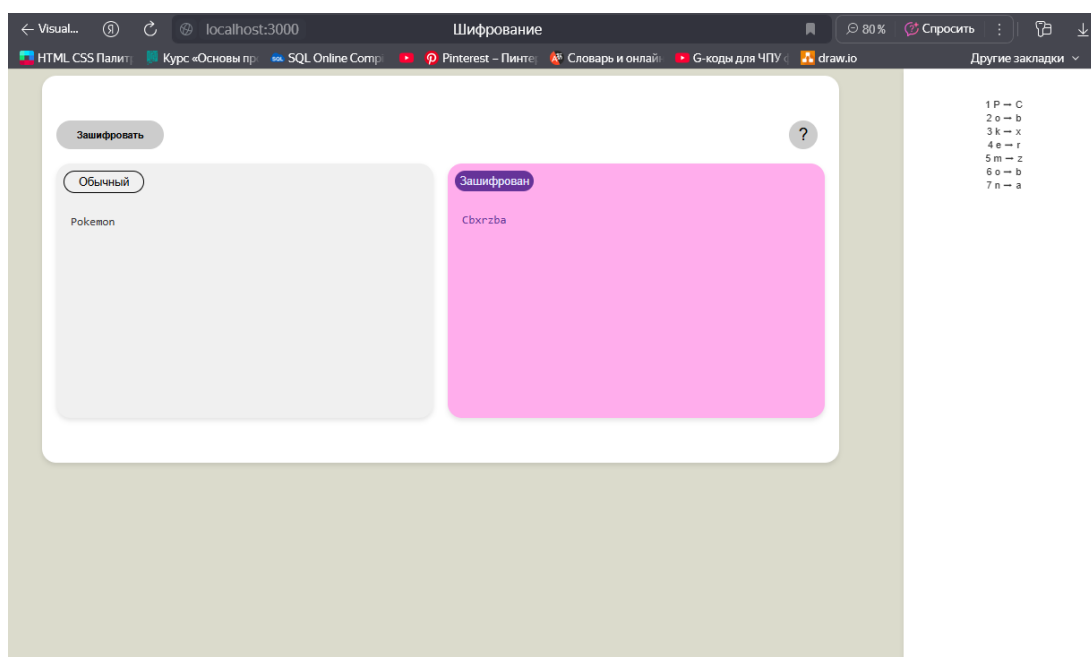


Рисунок 5 – Демонстрация работы

При написании чего либо другого в поле url (после указания хоста), кроме / , то выведет страницу ошибки 404.

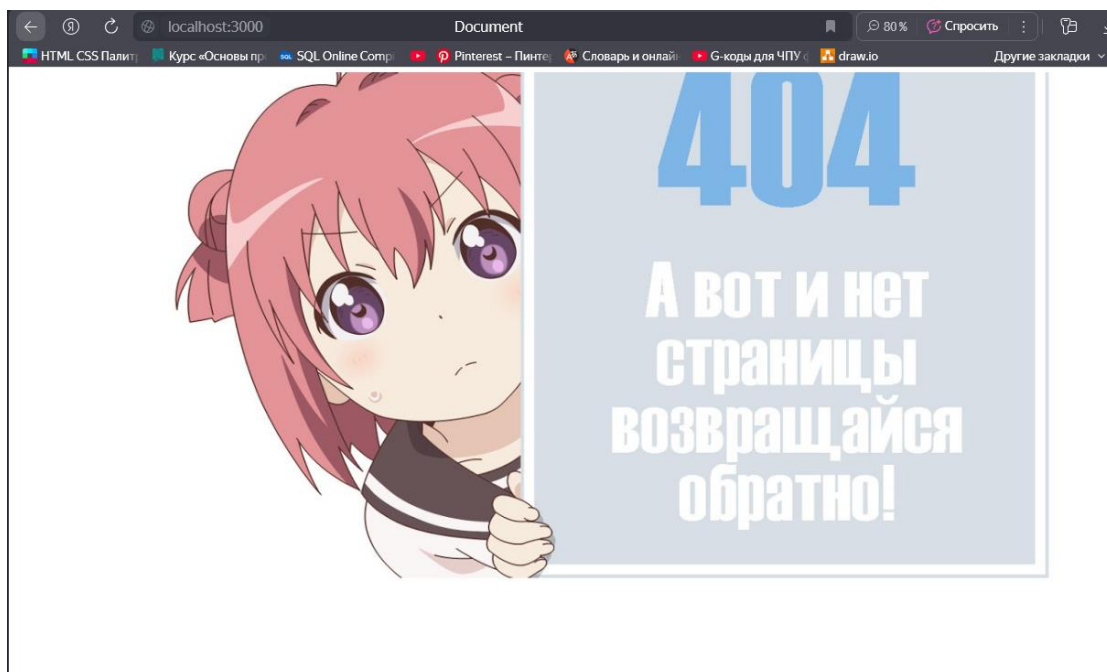


Рисунок 6 – страница ошибки 404

Вывод:

В ходе лабораторной работы были изучены азы работы с Node.js, так же в результате лабораторной работы был создан сервер который позволяет запускать сайт и обрабатывать страницы, а так же контактировать файлам html подключенному к серверу с их файлами (без определенной настройки будет HTML и даже если что то прописано в документе, то документ их не увидит)