

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:

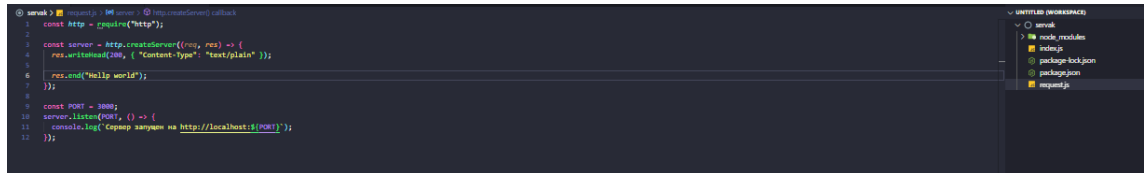


Подпись

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.

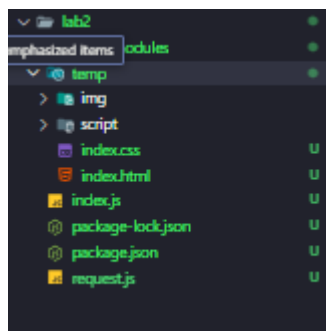


Рисунок 3 – Каталог с сервером

```

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

const server = http.createServer((req, res) => {
  const url = req.url;

  let filePath;
  let contentType = "text/html";

  // Явные маршруты
  if (url === '/' || url === '/index.html') {
    filePath = path.join(__dirname, "temp", "index.html");
  } else if (url === '/style.css') {
    filePath = path.join(__dirname, "temp", "index.css");
    contentType = "text/css";
  } else if (url === '/script.js') {
    filePath = path.join(__dirname, "temp", "script", "script.js");
    contentType = "application/javascript";
  } else if (url === '/404.jpg') {
    filePath = path.join(__dirname, "temp", "img", "404.jpg");
    contentType = "image/png";
  } else {
    filePath = path.join(__dirname, "temp", "page", "404.html");
  }

  fs.readFile(filePath, (err, content) => {
    if (err) {
      if (err.code === "ENOENT") {
        res.writeHead(404, { "Content-Type": "text/html" });
        res.end("<h1>404 - Страница не найдена</h1>");
      } 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

Так же нужно переделать код страницы, что бы он пользовался нашими роутами , которые мы прописали. Так же мы напишем страницу 404, которая будет тоже использовать роуты.

```

1 <!DOCTYPE html>
2 <html Lang="en">
3 <head>
4   <meta charset="UTF-8" />
5   <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
6   <title>Шифрование</title>
7   <link rel="stylesheet" href="/style.css">
8 </head>
9 <body>
10
11   <div class="wrapper">
12     <div class="panel">
13       <button class="najat" onclick="shifr()">Зашифровать</button>
14       <button class="info-btn" onclick="alphavit()">?</button>
15     </div>
16
17     <div class="text-container">
18       <div class="text-no-shifr">
19         <div>Обычный</div>
20         <textarea id="ne_sh"></textarea>
21       </div>
22
23       <div class="text-shifr">
24         <div>Зашифрован</div>
25         <textarea id="sh"></textarea>
26       </div>
27     </div>
28
29   </div>
30
31   <div class="sidebar" id="shifrBox">
32     <p id="put"></p>
33   </div>
34
35   <script src="/script.js"></script>
36
37 </body>
38 </html>

```

Рисунок 5 – Измененный HTML

```

<!DOCTYPE html>
<html Lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  
</body>
</html>

```

Рисунок 6 – код страницы ошибки

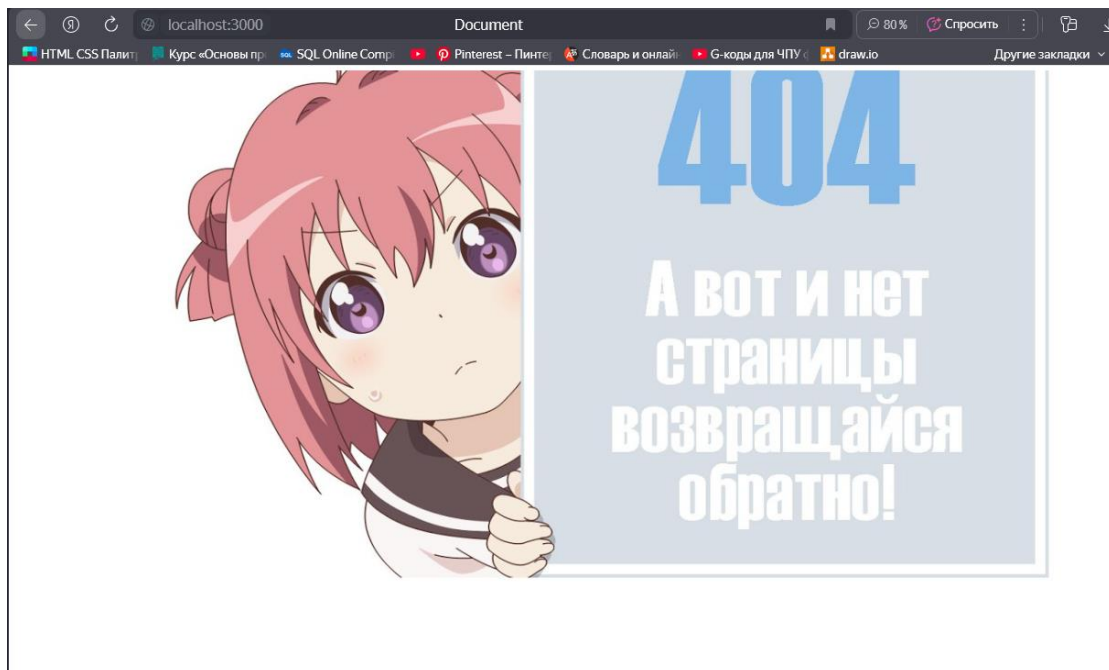


Рисунок 9 – страница ошибки 404

Далее сделаем так что бы сервер запускался с линукса , запускаем ssh сервер , и скачиваем посредством команды wseг папку с проектом с Github, а так же скачиваем необходимые для Node js библиотеки ,таким образом запускаем ssh порт и подключаемся к нему.

```
dalzhin_1isp-22@dalzhin1isp-22:~$ systemctl status ssh
• ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: enabled)
  Active: active (running) since Sun 2025-05-25 06:24:20 UTC; 1h 20min ago
  TriggeredBy: • ssh.socket
  Docs: man:sshd(8)
        man:sshd_config(5)
  Main PID: 2028 (sshd)
  Tasks: 1 (limit: 2274)
  Memory: 3.3M (peak: 4.3M)
  CPU: 551ms
  CGroup: /system.slice/ssh.service
          └─2028 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

мая 25 06:24:20 dalzhin1isp-22 systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
мая 25 06:24:20 dalzhin1isp-22 sshd[2028]: Server listening on :: port 22.
мая 25 06:24:20 dalzhin1isp-22 systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
мая 25 07:23:00 dalzhin1isp-22 sshd[21458]: Accepted password for dalzhin_1isp-22 from 10.0.2.15 port 52010 ssh2
мая 25 07:23:00 dalzhin1isp-22 sshd[21458]: pam_unix(sshd:session): session opened for user dalzhin_1isp-22(uid=1000) by dalzhin_1isp-22(uid=0)
dalzhin_1isp-22@dalzhin1isp-22:~$
```

Рисунок 10 – работа ssh

```
PS C:\Users\Яков> ssh -L 3000:127.0.0.1:3000 dalzhin_1isp-22@10.0.2.15
```

Рисунок 11 – подключение

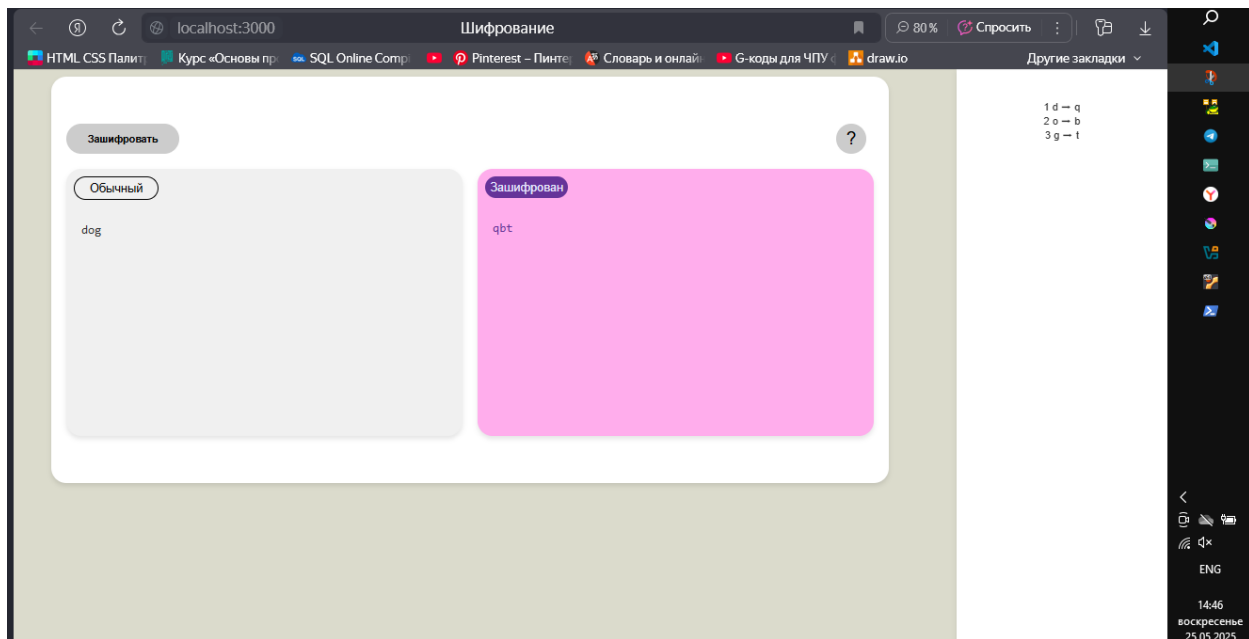


Рисунок 12 – Работа приложения

Обратим внимание что не включён VS code, через который мог бы осуществляться запуск Node js, если мы выключим виртуальную машину, то пропадёт и связь с сервером , а так же и проектом.

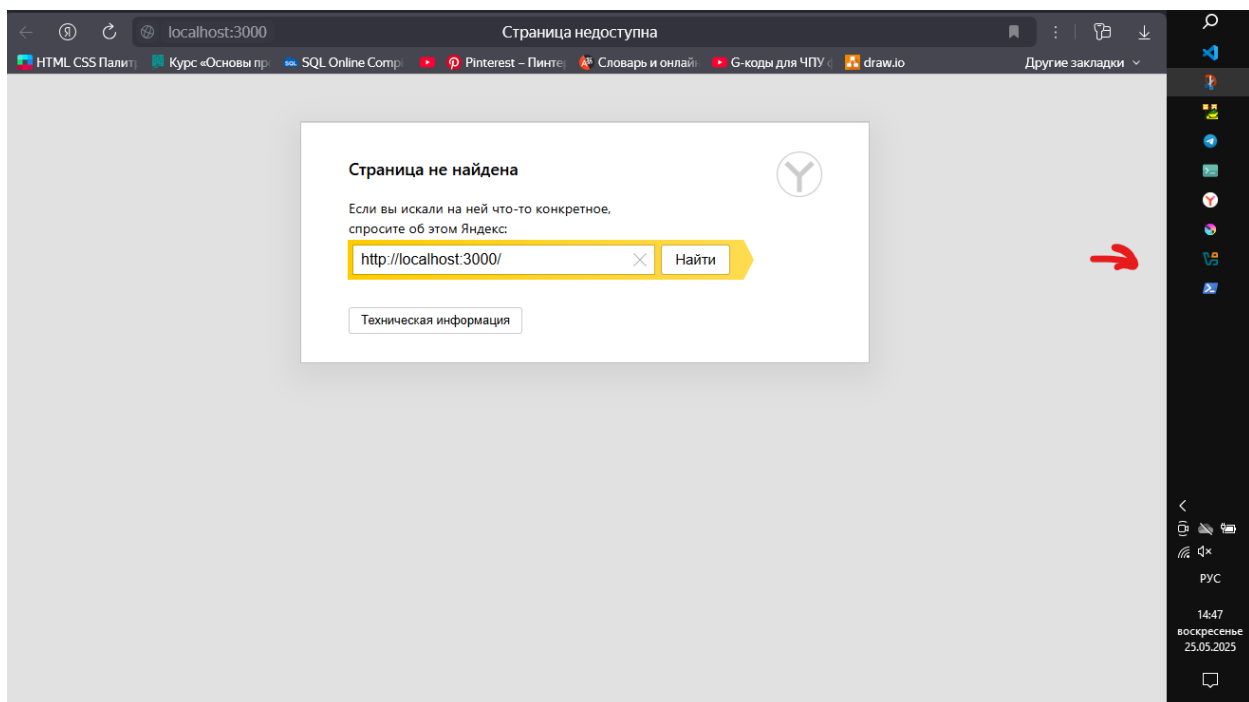


Рисунок 13 - Демонстрация

Более подробно можно ознакомиться по ссылке на Github - https://github.com/dalzhin1isp23/-_2025.git

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)