**МИНИСТЕРСТВО науки и высшего ОБРАЗОВАНИЯ РОссИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ**

**УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ**

(национальный исследовательский университет)»

Институт №3 «Системы управления, информатика и электроэнергетика»

Кафедра № 304 «Вычислительные машины, системы и сети»

Web-технологии

Отчет по лабораторной работе № 19

Web-сервер на Node.js

Выполнил студент группы М3О-409Б-20:

Мокшин И. Н.

Проверил, Титов Ю.П.

Москва 2023 г.

1. Разработать программу HTTP-сервер с помощью Node.js. Для создания сервера подключить модуль http. Поднять сервер на порту 8080 в состоянии listen. Для сервера необходимо создать 4 HTML страницы (4 файла, необходимо использовать модули fs и path) с различным наполнением: Текст, Поле ввода, Таблица, Картинка. Картинка будет использоваться для отображения страницы с кодами 404 и 5xx. Оставшиеся 3 страницы будут выдаватьсяс кодом 200 при различных URL-путях. На каждой странице должен быть переход на следующую через HTML теги.

**index.js**

var http = require('http');

var fs = require('fs');

var path = require('path');

function getMimeType(extention) {

console.log(extention);

switch (extention) {

case '.html':

return 'text/html';

break;

case '.css':

return 'text/css';

break;

case '.webp':

return 'image/webp';

case '.ico':

return 'image/x-icon';

break;

}

}

http.createServer(function (request, response) {

console.log('request starting...');

var filePath = '.' + request.url;

if (filePath == './') filePath = './index.html';

var fileExtension = path.extname(filePath);

var mimeType = getMimeType(fileExtension);

fs.readFile(filePath, function(error, content) {

if (error) {

if(error.code == 'ENOENT'){

fs.readFile('./notfound.html', function(error, content) {

response.writeHead(404, { 'Content-Type': "text/html" });

response.end(content, 'utf-8');

});

}

}

else {

response.writeHead(200, { 'Content-Type': mimeType });

response.end(content, 'utf-8');

}

});

}).listen(8080);

**index.html**

<!DOCTYPE html>

<html>

<head>

<title>My First HTML Page</title>

</head>

<body>

<h1>Hello World!</h1>

<p>This is my first HTML page.</p>

<button><a href="page2.html">Next</a></button>

</body>

</html>

**page2.html**

<!DOCTYPE html>

<html>

<head>

<title>My Second HTML Page</title>

</head>

<body>

<h1>Hello World 2!</h1>

<input type="text" placeholder="Enter text">

<button><a href="page3.html">Next</a></button>

</body>

</html>

**page3.html**

<!DOCTYPE html>

<html>

<head>

<title>My Third HTML Page</title>

</head>

<body>

<h1>Hello World 3!</h1>

<table>

<tr>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

<tr>

<td>John</td>

<td>Doe</td>

<td>john.doe@example.com</td>

</tr>

<tr>

<td>Jane</td>

<td>Smith</td>

<td>jane.smith@example.com</td>

</tr>

</table>

<button><a href="page2.html">Back</a></button>

</body>

</html>

**notfound.html**

<!DOCTYPE html>

<html>

<head>

<title>Not found</title>

</head>

<body>

<h1>Oops! Page not found</h1>

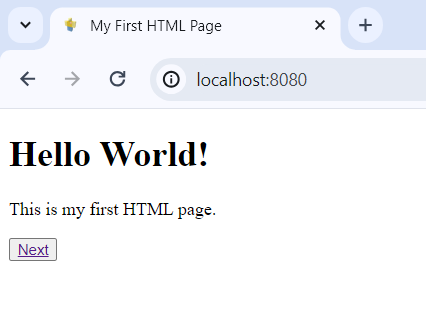
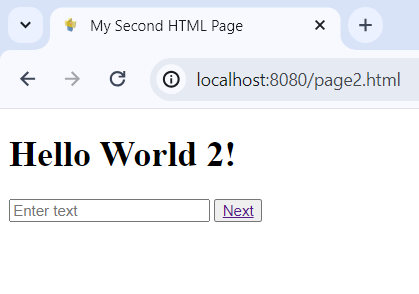
<img src="roblox.webp">

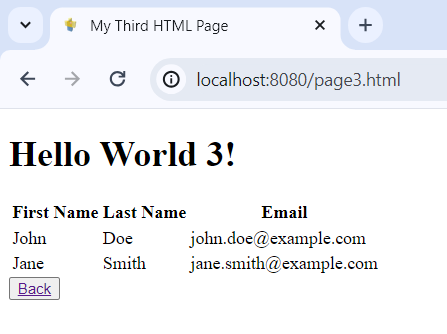
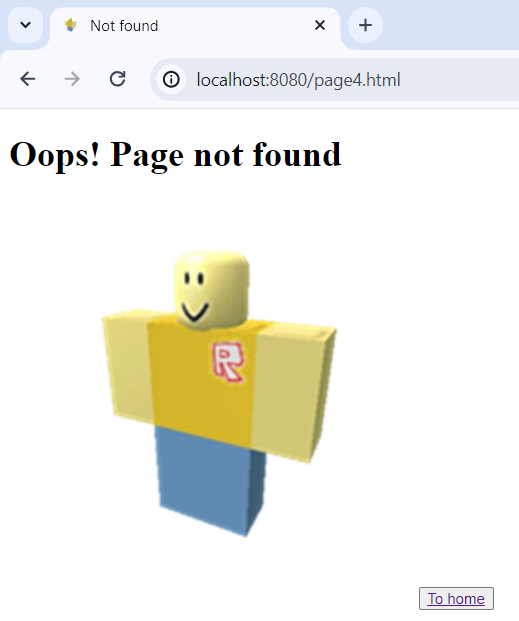
<button><a href="index.html">To home</a></button>

</body>

</html>

1. В отчете привести пример работы web-сервера и отображения всех страниц. (можно отдельные фрагменты)

1. Добавить файл стилей CSS.

**style.css**

body {

font-family: Impact, Haettenschweiler, 'Arial Narrow Bold', sans-serif;

}

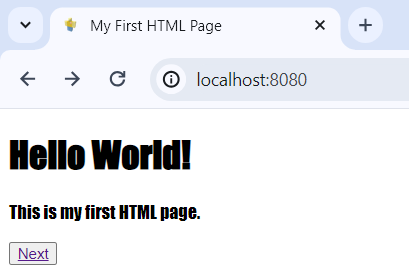
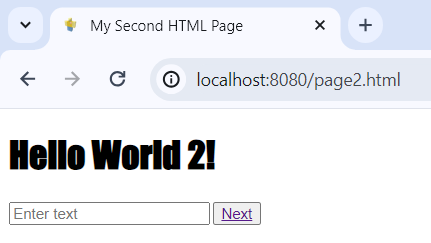
tr, td {

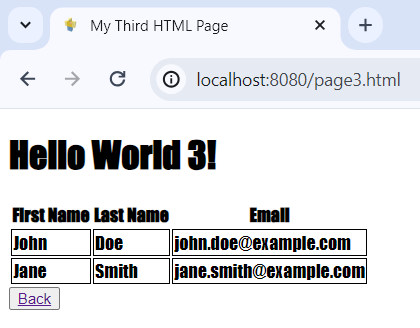
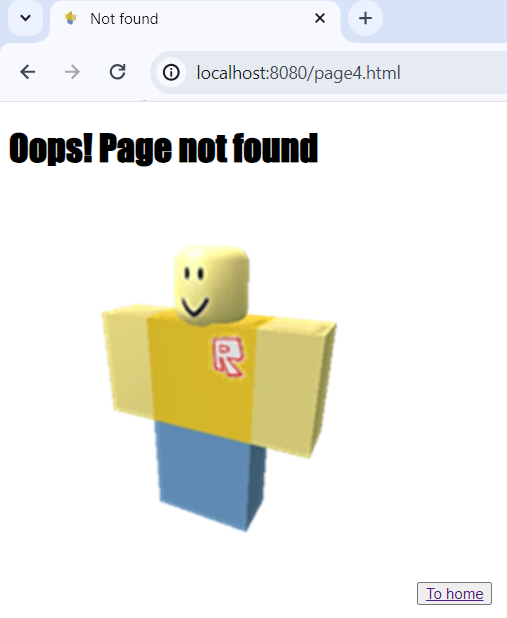
border: 1px solid black;

}

1. В отчете привести пример работы web-сервера и отображения всех страниц. (можно отдельные фрагменты) и код всех файлов сервера. Также привести запросы и ответы из вкладки Network в браузере.

Добавлен блок <link rel="stylesheet" href="style.css"> в блок head каждой страницы

Для примера загрузки взята страница, которой нет, т.е. сервер отдал 404 not found и страницу-заглушку.

