**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информатика»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 4

по дисциплине **«Разработка приложений для Интернет»**

на тему: **«Http/Https»**

Выполнил: студент гр. ИП-31

Коваленко А.И.

Принял: ст. преподаватель Самовендюк Н. В.

Дата сдачи отчета: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата допуска к защите: ­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата защиты: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Гомель 2023

**Цель работы:** освоить возможности NodeJs по созданию локальных серверов с выводом статической информации, ознакомиться с фреймворком Express.

**Задание 1** Создать сервер на NodeJs без использования фреймворка express. На базе сервера разработать статический web-сайт, состоящий из нескольких страниц. Тема выбирается студентом самостоятельно. К страницам сайта применить стилизацию. Предусмотреть переходы между станицами. Страницы храняться как статические файлы.

**Задание 2** Выполнить задание1 с использованием фреймворка Express. Дополнительно: все переходы логгировать в отдельный файл server.log.

**Задание 3** Используя фреймворк Express, разработать REST API. В качестве предметной области взять задание на курсовую работу. В качестве источника данных создать файл в формате JSON. Установить Postman и протестировать все разработанный REST API.

Листинг программы  
**contacts.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Контакты</title>

</head>

<style>

body {

background-color: rgb(125, 230, 99);

}

.menu {

width: 19%;

}

.nav {

width: 100%;

}

.content {

width: 100%;

margin-top: 20px;

margin-left: 20px;

font-size: 40px;

}

.section {

width: 80%;

margin-top: 20px;

margin-left: 20px;

}

.inks>ul>li,

.nav>ul>li {

list-style-type: none;

padding: 8px 10px;

margin-top: 7px;

}

.section p {

width: 100%;

text-align: justify;

}

</style>

<body>

<div class="content">

<div class="menu">

<div class="nav">

<ul>

<li><a href="/" target="\_self">Безопасность</a></li>

<li><a href="/info" target="\_self">О нас</a></li>

<li><a href="/contacts" target="\_self">Контакты</a></li>

</ul>

</div>

</div>

<div class="section">

<p>Установа адукацыі «Гомельскі дзяржаўны тэхнічны ўніверсітэт імя П.В.Сухога» (ГДТУ iмя П.В.Сухога)

Пр-т Кастрычнiка, 48, 246029, г. Гомель, Рэспублiка Беларусь</p>

<p> Учреждение образования «Гомельский государственный технический университет имени П.О.Сухого» (ГГТУ им. П.О. Сухого)

Пр-т Октября, 48, 246029, г. Гомель, Республика Беларусь</p>

<p>Sukhoi State Technical University of Gomel

Prospect Octiabria, 48, 246029, Gomel, Republic of Belarus</p>

<p> Телефон: (+375 232) 22-46-36

Факс: (+375 232) 26-02-87

http://www.gstu.by

РЕЖИМ РАБОТЫ: понедельник - пятница 8.00 - 12.00; 13.00 - 17.00</p>

</div>

</div>

</body>

</html>

**info.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>О нас</title>

</head>

<style>

body {

background-color: rgb(125, 230, 99);

}

.menu {

width: 19%;

}

.nav {

width: 100%;

}

.content {

width: 100%;

margin-top: 20px;

margin-left: 20px;

font-size: 40px;

}

.section {

width: 80%;

margin-top: 20px;

margin-left: 200px;

}

.inks>ul>li,

.nav>ul>li {

list-style-type: none;

padding: 8px 10px;

margin-top: 7px;

}

.section p {

width: 100%;

text-align: justify;

}

</style>

<body>

<div class="content">

<div class="menu">

<div class="nav">

<ul>

<li><a href="/" target="\_self">Безопасность</a></li>

<li><a href="/info" target="\_self">О нас</a></li>

<li><a href="/contacts" target="\_self">Контакты</a></li>

</ul>

</div>

</div>

<div class="section">

<p>О ком писать в разделе «О нас»

Добавим немного когнитивного диссонанса.

Страница «О компании» не должна рассказывать о вашей компании. «О нас» должна в первую очередь рассказывать о клиенте и его интересах, а во вторую – о вашем бизнесе.

Помните английскую поговорку о джентльменах? Так называют людей, которые никогда не употребляют слово «джентльмен». Держите это в уме: текст «О нас» – это история о нуждах клиента, а не о том, как прошли ваше детство и юность.

Но как узнать, в чем нуждается клиент? Мы подошли к важной теме. Но обо всем по порядку.</p>

</div>

</div>

</body>

</html>

**safety.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Безопасность</title>

</head>

<style>

body {

background-color: rgb(125, 230, 99);

}

.menu {

width: 19%;

}

.nav {

width: 100%;

}

.content {

width: 100%;

margin-top: 20px;

margin-left: 20px;

font-size: 40px;

}

.section {

width: 80%;

margin-top: 20px;

margin-left: 20px;

}

.inks>ul>li,

.nav>ul>li {

list-style-type: none;

padding: 8px 10px;

margin-top: 7px;

}

.section p {

width: 100%;

text-align: justify;

}

</style>

<body>

<div class="content">

<div class="menu">

<div class="nav">

<ul>

<li><a href="/" target="\_self">Безопасность</a></li>

<li><a href="/info" target="\_self">О нас</a></li>

<li><a href="/contacts" target="\_self">Контакты</a></li>

</ul>

</div>

</div>

<div class="section">

<p>В ГГТУ им. П.О. Сухого прошла встреча

с сотрудниками МЧС на тему «Правила безопасности и меры поведения на воде в осенне-зимний период»</p>

</div>

</div>

</body>

</html>

**l4\_1.js**

const http = require("http");

const fs = require("fs");

const hostname = "localhost";

const port = 3000;

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

res.statusCode = 200;

res.setHeader("Content-Type", "text/html");

if (req.url == "/") {

fs.createReadStream("extras/safety.html").pipe(res);

} else if (req.url == "/info") {

fs.createReadStream("extras/info.html").pipe(res);

} else if (req.url == "/contacts") {

fs.createReadStream("extras/contacts.html").pipe(res);

}

});

server.listen(port, hostname, () => {

console.log(`Server running at http://${hostname}:${port}/`);

});

**l4\_2.js**

const express = require("express");

const exp = express();

const fs = require("fs");

const hostname = "localhost";

const port = 3000;

function logging(data) {

fs.writeFile("server.log", data + "\n", { flag: "a" }, (err) => {

if (err) console.log(err);

});

}

exp.use(function (req, res, next) {

let now = new Date();

let data = `${now} ${req.url} ${req.method} ${req.get("user-agent")}`;

logging(data);

next();

});

exp.get("/", (req, res) => {

res.sendFile(\_\_dirname + "/extras/safety.html");

});

exp.get("/info", (req, res) => {

res.sendFile(\_\_dirname + "/extras/info.html");

});

exp.get("/contacts", (req, res) => {

res.sendFile(\_\_dirname + "/extras/contacts.html");

});

exp.listen(port, hostname, () => {

console.log(`Server running at http://${hostname}:${port}/`);

});

**l4\_3.js**

const express = require("express");

const fs = require("fs");

const exp = express();

const jsonParser = express.json();

const filePath = "cinemas.json";

const hostname = "localhost";

const port = 3000;

exp.get("/cinemas", function (req, res) {

const data = fs.readFileSync(filePath, "utf8");

const cinemas = JSON.parse(data);

res.send(cinemas);

});

exp.put("/cinemas/:id", jsonParser, function (req, res) {

const cinemaID = req.params.id;

const cinemaName = "New";

const cinemaLocation = "Могилев";

const cinemaCost = "7р";

let cinemas = JSON.parse(fs.readFileSync(filePath, "utf8"));

let cinema = cinemas.find((el) => el.id == cinemaID);

if (cinema) {

cinema.name = cinemaName;

cinema.location = cinemaLocation;

cinema.cost = cinemaCost;

fs.writeFileSync(filePath, JSON.stringify(cinemas, null, 2), "utf-8");

res.send(cinema);

} else {

res.status(404).send();

}

});

exp.post("/cinemas", jsonParser, function (req, res) {

const cinemaName = "Красный октябрь";

const cinemaLocation = "Минск";

const cinemaCost = "10р";

let cinema = {

name: cinemaName,

location: cinemaLocation,

cost: cinemaCost,

};

let cinemas = JSON.parse(fs.readFileSync(filePath, "utf8"));

const id = Math.max(

...cinemas.map(function (cin) {

return cin.id;

})

);

cinema.id = id + 1;

cinemas.push(cinema);

fs.writeFileSync(filePath, JSON.stringify(cinemas, null, 2), "utf-8");

res.send(cinema);

});

exp.delete("/cinemas/:id", function (req, res) {

const id = req.params.id;

let cinemas = JSON.parse(fs.readFileSync(filePath, "utf8"));

let index = cinemas.find((el) => el.id == id).id;

if (index > -1) {

const cinema = cinemas.splice(index, 1)[0];

fs.writeFileSync(filePath, JSON.stringify(cinemas, null, 2), "utf-8");

res.send(cinema);

} else {

res.status(404).send();

}

});

exp.listen(port, hostname, () => {

console.log(`Server running at http://${hostname}:${port}/`);

});

**Результат выполнения.**

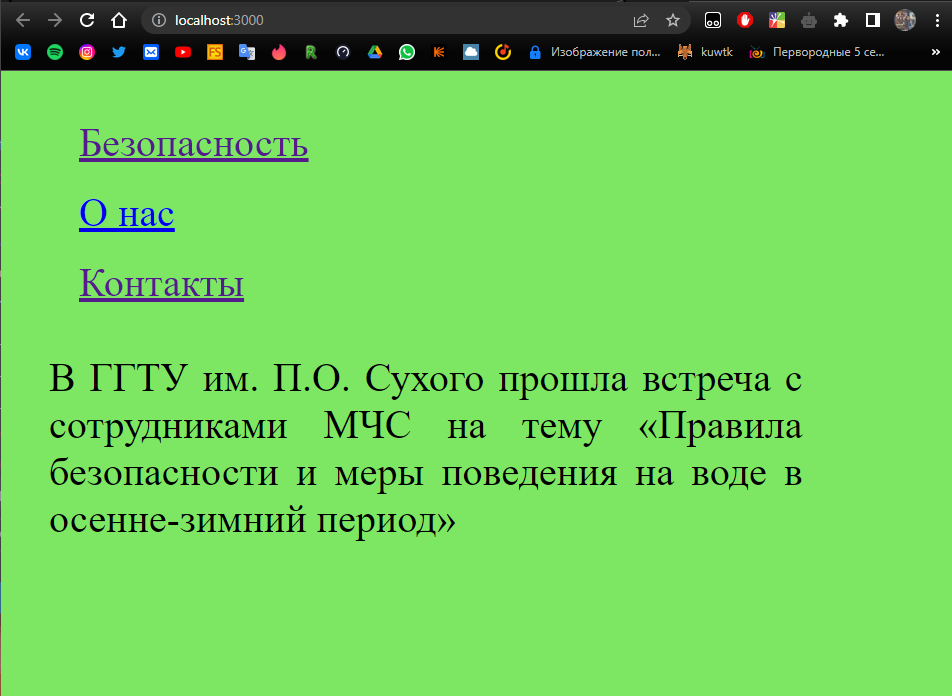
****

Рисунок 1 – Страница безопасность

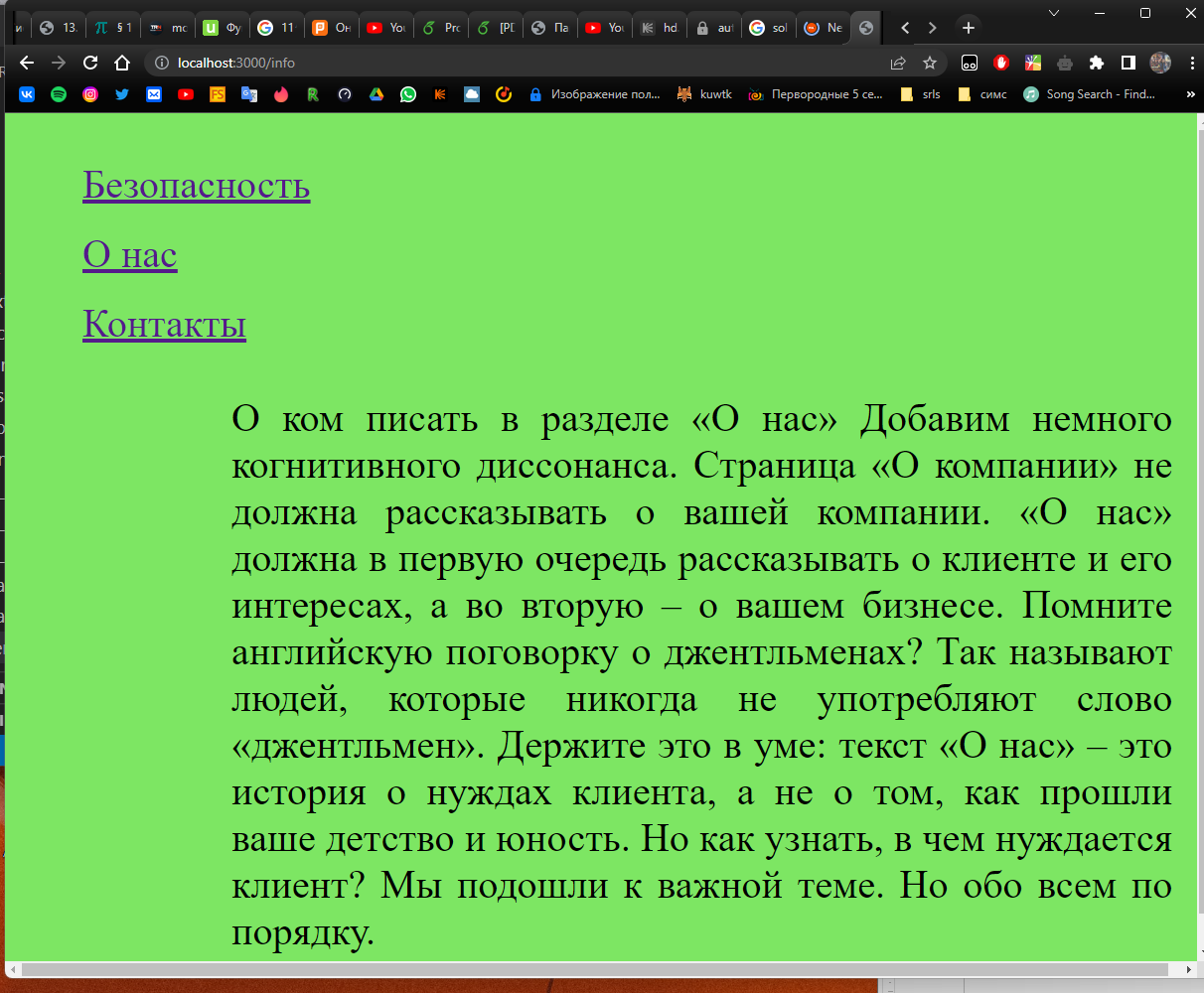
****

Рисунок 2 – Страница о нас

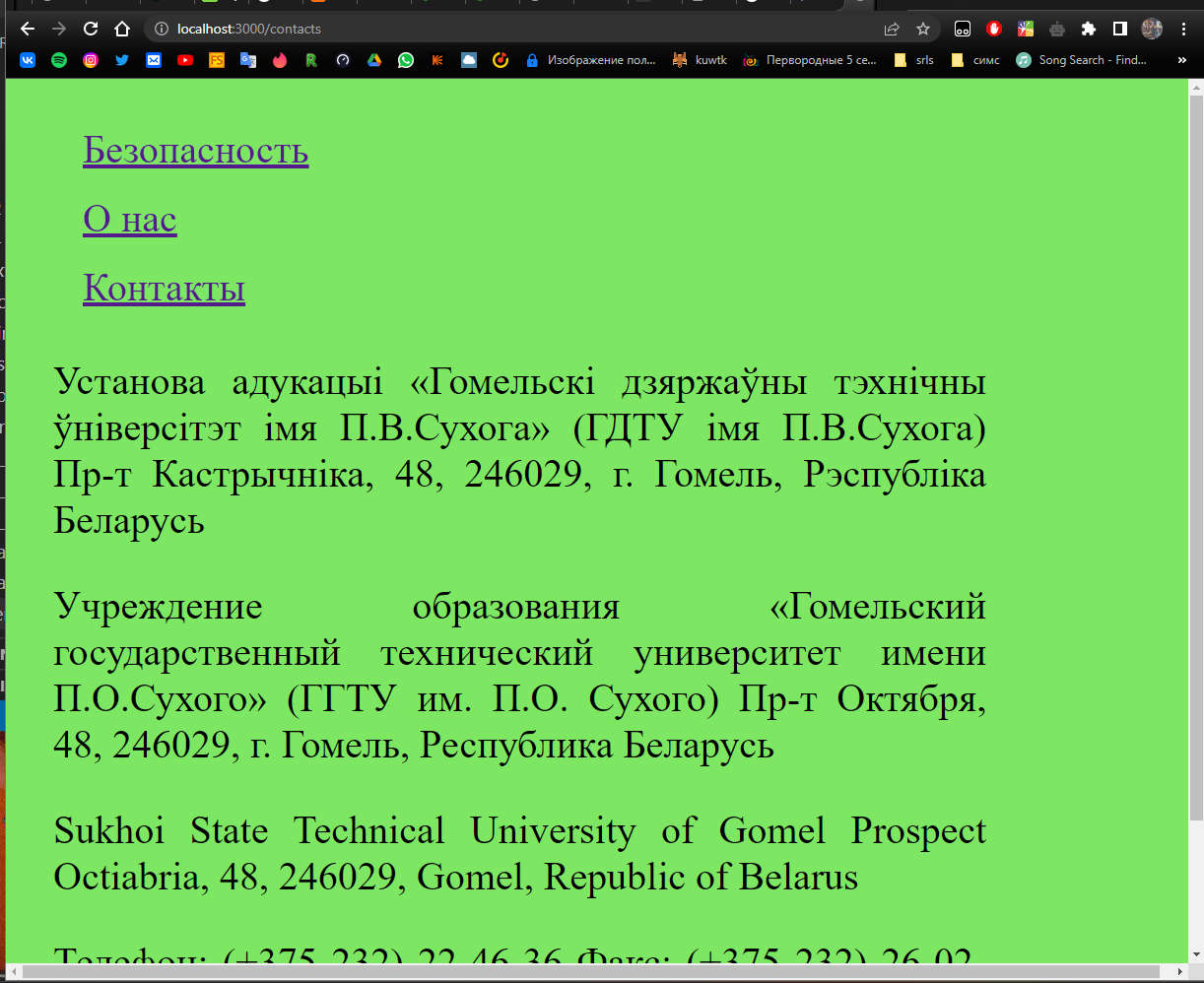
****

Рисунок 3 – Страница Контакты

**Результат выполнения.**

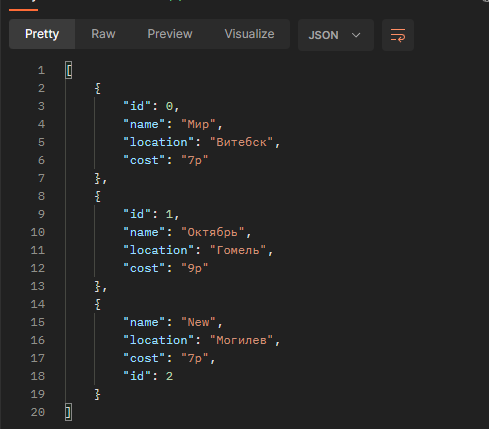


Рисунок 4 – GET запрос

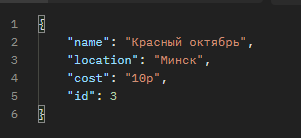


Рисунок 5 – POST запрос



Рисунок 6 – PUT запрос



Рисунок 7 – Результат выполнения запросов get, put, post



Рисунок 7 – DELETE запрос

**Вывод.**

В ходе лабораторной работы освоилт возможности NodeJs по созданию локальных серверов с выводом статической информации, ознакомился с фреймворком Express.