ГУАП

КАФЕДРА № 42

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ _____

ПРЕПОДАВАТЕЛЬ

ассистент		В. В. Жукалин
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №2

ОБРАБОТКА ДАННЫХ HTML-ФОРМ В NODE.JS

по курсу: WEB-ПРОГРАММИРОВАНИЕ

Вариант 3

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. № 4128		Д. И. Вититников
	подпись, дата	инициалы, фамилия

Санкт-Петербург 2024

СОДЕРЖАНИЕ

1 Цель работы и задание

1.1 Цель работы:

Получение навыков программирования на языке программирования JavaScript в среде Node.JS. Изучение возможностей получения данных из HTML-формы и их обработки в среде Node.JS.

1.2 Задание

1 Задание

Создать папку с проектом, содержащую страницу с HTML-формой, каскадной таблицей стилей (CSS) и JS-скриптом в среде Node.JS.

2 Разработать web-страницу, которая должна предлагать пользователю HTML-форму, включающую различные элементы (текстовые поля, списки, кнопки и т.д.). HTML-форма должна содержать не менее 10 различных элементов. HTML-форма должна иметь дизайн, разработанный средствами CSS. Тематика, для которой создается форма, определяется в соответствии с приложением А. HTML-форма должна иметь обязательные поля (input type="text") в соответствии с разделом «Варианты для лабораторных работ». РЕКОМЕНДУЕТСЯ использовать форму из Лабораторной работы № 5 «Создание форм в web-документах» по дисциплине «Web-технологии».

№ варианта

1) Дистанционное обучение
2) Электронный магазин бытовой техники
3) Электронный магазин компьютерной техники
4) Фитнес-клуб
5) Бронирование мест в гостинице

6) Туристическое агентство

7) Агентство недвижимости (продажа объектов)

8) Агентство недвижимости (аренда объектов)

9) Заказ билетов в театр

10) Заказ билетов на самолет (поезд)

3 Разработать JS-скрипт в среде Node.JS, который получает данные из HTML-формы методом POST и, в зависимости от полученного содержимого, формирует новую страницу. Итоговая страница должна содержать текстовые и графические элементы, связанные с результатом заполненной HTML формы. Количество элементов новой страницы должно быть не меньше количества полей в HTML-форме. Для поля типа radio или checkbox предусмотреть возможность изменения минимум одного изображения в HTML-форме в зависимость от выбранного пользователем значения поля.

4 4 JS-скрипт должен проверить полноту введенных пользователем данных и корректность их ввода. Также полученная с помощью JS-скрипта страница должна иметь разработанный дизайн средствами CSS.

2 Ход работы

В результате получилось создать страницы, изображенные на рис. 1-3. При неправильном вводе или отсутствии ввода там, где это необходимо, выводится соответствующее сообщение об этом. Для создания таблицы стилей для формы была использована технология flexbox. Результирующий код представлен в Приложении. Пример представлен на рис. 2:

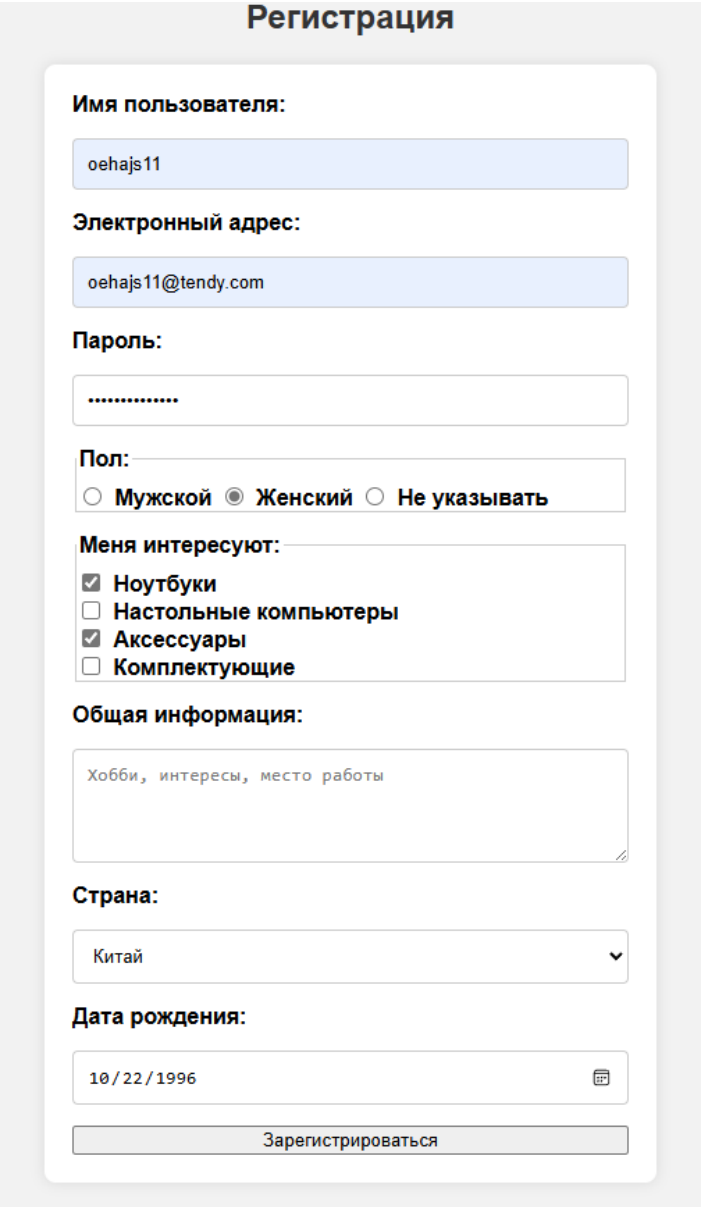

Рисунок 1 - Разработанная форма

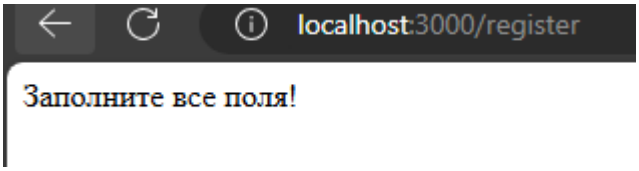

Рисунок 2 - Пример вывода сообщения о неправильно заполненной форме

В случае, если пользователь форму заполняет правильно, сервер возвращает страницу, изображённую на рис. 3:

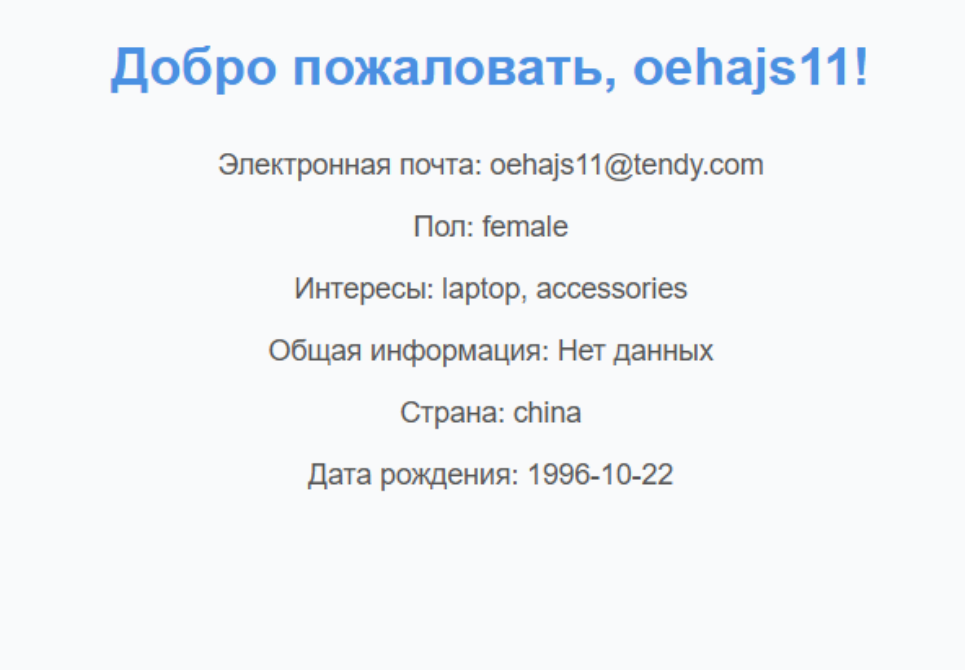

Рисунок 3 - Сгенерированная страница

ВЫВОД

В ходе выполнения лабораторной работы были изучены основные принципы работы с сервером на базе Node.js. Был настроен сервер для обработки данных, отправленных через HTML-форму методом POST, и обеспечено его взаимодействие с фронтенд-частью через маршрутизацию и раздачу статических файлов. Применен модуль body-parser для корректного парсинга данных формы, что позволило эффективно обрабатывать ввод пользователя.

Также реализована структура проекта, где файлы сервера и фронтенда расположены таким образом, чтобы сервер автоматически обслуживал статические ресурсы, такие как стили CSS и изображения. Это упростило организацию кода и улучшило пользовательский интерфейс. На выходной странице представлена информация, введенная пользователем, с дополнительными элементами оформления, что сделало взаимодействие более интуитивным и привлекательным.

Данная работа позволила закрепить навыки организации серверного кода, обработки POST-запросов для получения данных из HTML формы и дальнейшей их обработки на сервере.

ПРИЛОЖЕНИЕ А
INDEX.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Электронный магазин компьютерной техники</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="index.css">
</head>
<body>
    <h2>Регистрация</h2>
    <form action="/register" method="post">
        <label for="username">Имя пользователя:</label><br>
        <input type="text" id="username" name="username" placeholder="Введите ваше имя пользователя (латиница)" pattern="[a-zA-Z0-9]+" minlength="3"><br>

        <label for="email">Электронный адрес:</label><br>
        <input type="email" id="email" name="email" placeholder="example@form.com"><br>

        <label for="password">Пароль:</label><br>
        <input type="password" id="password" name="password" placeholder="Минимальная длина: 8 символов" minlength="8"><br>
```

```
<fieldset>
    <legend>Пол:</legend>
    <input type="radio" id="male" name="gender" value="male">
    <label for="male">Мужской</label><br>
    <input type="radio" id="female" name="gender" value="female">
    <label for="female">Женский</label><br>
    <input type="radio" id="unidentified" name="gender" value="unidentified">
    <label for="female">Не указывать</label><br>
</fieldset>

<fieldset>
    <legend>Меня интересуют:</legend>
    <input type="checkbox" id="laptop" name="interests" value="laptop">
    <label for="laptop">Ноутбуки</label><br>
    <input type="checkbox" id="desktop" name="interests" value="desktop">
    <label for="desktop">Настольные компьютеры</label><br>
    <input type="checkbox" id="accessories" name="interests" value="accessories">
    <label for="accessories">Аксессуары</label><br>
    <input type="checkbox" id="components" name="interests" value="components">
    <label for="components">Комплектующие</label><br>
</fieldset>
```

```
        <label for="about">Общая информация:</label><br>
        <textarea id="about" name="address" rows="4" cols="50" placeholder="Хобби, интересы, место работы"></textarea><br>

        <label for="country">Страна:</label><br>
        <select id="country" name="country">
            <option value="russia">Россия</option>
            <option value="usa">США</option>
            <option value="china">Китай</option>
            <option value="other">Другая</option>
        </select><br>

        <label for="birthdate">Дата рождения:</label><br>
        <input type="date" id="birthdate" name="birthdate"><br>

        <input type="submit" value="Зарегистрироваться">
    </form>
</body>
</html>
```

ПРИЛОЖЕНИЕ Б
SERVER.JS

```
const express = require('express');
const bodyParser = require('body-parser');
const path = require('path');
const app = express();
const port = 3000;

app.use(bodyParser.urlencoded({ extended: true }));
app.use(express.static(__dirname));

// получаем html страницу с формой
app.get('/', (req, res) => {
    res.sendFile(path.join(__dirname, 'index.html'));
});

// применяем post к action "/register"
app.post('/register', (req, res) => {
    const { username, email, password, gender, interests, about, country, birthdate } =
  req.body;

    // проверка полей
    if (!username || !email || !password || !gender || !country || !birthdate) {
        return res.status(400).send('Заполните все поля!');
    }
```

```
  if (password.length < 8 || username.length < 3) {
    return res.status(400).send('Некорректный ввод: Пожалуйста, проверьте требования к логину и паролю.');
  }

  // динам. рендер
  const interestsText = Array.isArray(interests) ? interests.join(', ') : interests || 'None';

  // ответ от сервера - сгенерированная страница
  res.send(`
    <html>
    <head>
      <title>Welcome, ${username}</title>
      <link rel="stylesheet" href="/welcome.css">
    </head>
    <body>
      <h2>Добро пожаловать, ${username}!</h2>
      <p>Электронная почта: ${email}</p>
      <p>Пол: ${gender}</p>
      <p>Интересы: ${interestsText}</p>
      <p>Общая информация: ${about || 'Нет данных'}</p>
      <p>Страна: ${country}</p>
      <p>Дата рождения: ${birthdate}</p>
    </body>
```

```
        </html>
    `);
});

// для быстрого подключения через консоль
app.listen(port, () => {
    console.log(`join http://localhost:${port}`);
});
```

ПРИЛОЖЕНИЕ В
INDEX.CSS

```
body {
    font-family: Arial, sans-serif;
    background-color: #f2f2f2;
    margin: 0;
    padding: 0;
    display: flex;
    justify-content: center;
    align-items: center;
    min-height: 100vh;
}

form {
    width: 400px;
    background-color: #fff;
    padding: 20px;
    border-radius: 8px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    display: flex;
    flex-direction: column;
    gap: 15px;
}

h2 {
```

```
    text-align: center;
    color: #333;
    margin-bottom: 20px;
    flex-direction: column;
}

label {
    font-weight: bold;
}

input[type="text"],
input[type="email"],
input[type="password"],
textarea,
select,
input[type="date"] {
    width: 100%;
    padding: 10px;
    border: 1px solid rgb(204, 204, 204);
    border-radius: 4px;
    box-sizing: border-box;
}

fieldset {
```

```
    border: none;
    padding: 0;
    flex-direction: column;
    border-style: solid;
    border-color: rgb(204, 204, 204);
    border-width: 1px;
}

legend {
    font-weight: bold;
    margin-bottom: 8px;
}

.radio-group,
.checkbox-group {
    display: flex;
    flex-direction: row;
    gap: 10px;
}

input[type="radio"],
input[type="checkbox"] {
    margin-right: 5px
}
```

```
button {
    padding: 12px 20px;
    background-color: #007bff;
    color: #fff;
    border: none;
    border-radius: 4px;
    cursor: pointer;
    font-size: 16px;
}

button:hover {
    background-color: #0056b3;
}

input:invalid {
    border-color: #ff0000;
}

#container {
    flex-direction: column;
}
```

ПРИЛОЖЕНИЕ Г
WELCOME.CSS

```
body {
    font-family: Arial, sans-serif;
    background-color: #f9fafb;
    margin: 0;
    padding: 20px;
    color: #333;
    text-align: center;
}

h2 {
    color: #4a90e2;
    font-size: 2em;
    margin-top: 20px;
}

p {
    font-size: 1.1em;
    margin: 10px 0;
    line-height: 1.6;
    color: #555;
}

img {
```

```
        width: 150px;
        height: 150px;
        border-radius: 50%;
        margin: 20px 0;
        box-shadow: 0px 4px 10px rgba(0, 0, 0, 0.1);
    }

    button {
        background-color: #4a90e2;
        color: white;
        padding: 10px 20px;
        border: none;
        border-radius: 5px;
        font-size: 1em;
        cursor: pointer;
        transition: background-color 0.3s ease;
    }

    button:hover {
        background-color: #357ab8;
}
```