

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 9

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №	4128		В. А. Тарапанов
		подпись, дата	инициалы, фамилия

Санкт-Петербург 2024

## СОДЕРЖАНИЕ

1	Цель работы и задание.....	3
1.1	Цель работы:.....	3
1.2	Задание.....	3
2	Ход работы.....	5
	ВЫВОД.....	7
	ПРИЛОЖЕНИЕ А INDEX.HTML.....	8
	ПРИЛОЖЕНИЕ Б SERVER.JS .....	11
	ПРИЛОЖЕНИЕ В INDEX.CSS .....	13
	ПРИЛОЖЕНИЕ Г WELCOME.CSS.....	16

## **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 9 - Заказ билетов в театр

- 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:

The image shows a web form for ordering theater tickets. The form is titled "Заказ билетов в театр" (Order tickets for the theater) in a bold, orange font. It is set against a light beige background. The form itself is a white rounded rectangle with a thin orange border. It contains several input fields and a submit button. The fields are labeled in Russian: "Ваше имя:" (Your name), "Электронная почта:" (Email), "Номер телефона:" (Phone number), "Выберите спектакль:" (Choose a play), "Тип места:" (Seat type), "Дополнительные услуги:" (Additional services), "Комментарии:" (Comments), and "Дата посещения:" (Date of visit). The "Выберите спектакль:" field has three radio button options: "Гамлет" (Hamlet), "Ромео и Джульетта" (Romeo and Juliet), and "Чайка" (The Cherry Orchard). The "Дополнительные услуги:" field has three checkbox options: "Снэки" (Snacks), "Напитки" (Beverages), and "Программа спектакля" (Program of the play). The "Дата посещения:" field has a date picker icon. At the bottom of the form is an orange button with the text "Оформить заказ" (Place order).

**Заказ билетов в театр**

Ваше имя:  
Введите ваше имя

Электронная почта:  
example@domain.com

Номер телефона:  
+7 999 123-45-67

Выберите спектакль:  
☐ Гамлет  
☐ Ромео и Джульетта  
☐ Чайка

Тип места:  
Стандарт

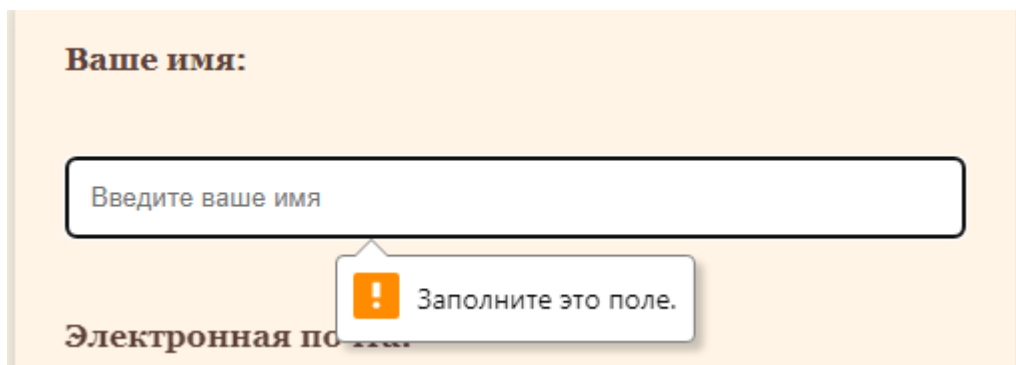
Дополнительные услуги:  
☐ Снэки  
☐ Напитки  
☐ Программа спектакля

Комментарии:  
Особые пожелания или комментарии

Дата посещения:  
ДД.ММ.ГГГГ

Оформить заказ

Рисунок 1 - Разработанная форма



Ваше имя:

Введите ваше имя

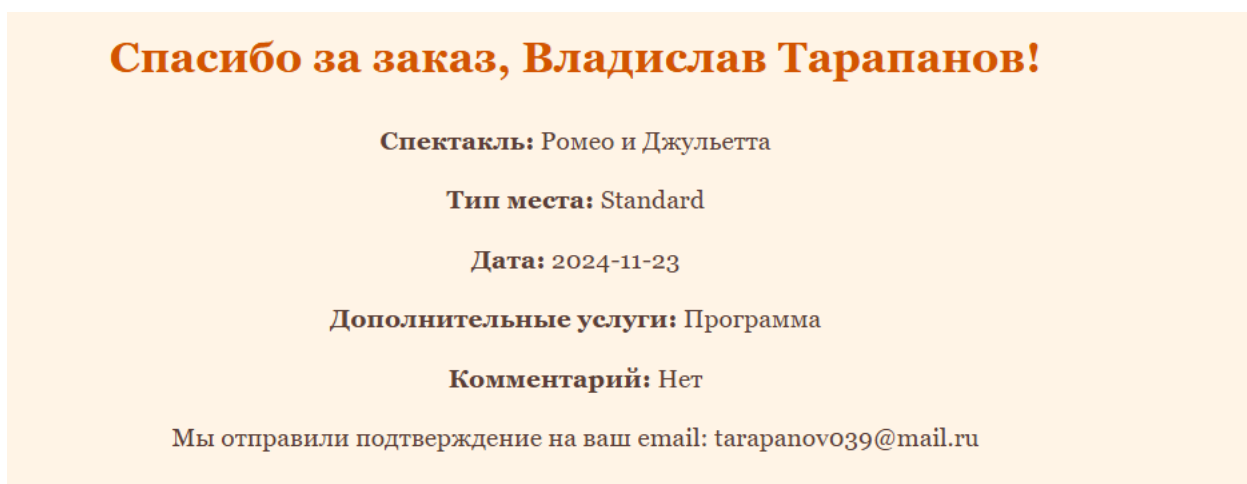
! Заполните это поле.

Электронная почта:

This screenshot shows a web form with a light orange background. At the top, it says 'Ваше имя:' followed by a text input field containing the placeholder 'Введите ваше имя'. Below the input field, a white error message box with a red exclamation mark icon and the text 'Заполните это поле.' is displayed. At the bottom, the text 'Электронная почта:' is visible.

Рисунок 2 - Пример вывода сообщения о неправильно заполненной форме

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



**Спасибо за заказ, Владислав Тарапанов!**

**Спектакль:** Ромео и Джульетта

**Тип места:** Standard

**Дата:** 2024-11-23

**Дополнительные услуги:** Программа

**Комментарий:** Нет

Мы отправили подтверждение на ваш email: tarapanov039@mail.ru

This screenshot shows a confirmation page with a light orange background. It features a large orange heading 'Спасибо за заказ, Владислав Тарапанов!'. Below this, several details are listed in a structured format: 'Спектакль: Ромео и Джульетта', 'Тип места: Standard', 'Дата: 2024-11-23', 'Дополнительные услуги: Программа', and 'Комментарий: Нет'. At the bottom, a line of text states 'Мы отправили подтверждение на ваш email: tarapanov039@mail.ru'.

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, что позволило эффективно обрабатывать ввод пользователя и формировать динамическую страницу с результатами. В процессе работы использовался модуль `body-parser` для корректного парсинга данных формы, что обеспечило корректную обработку информации, полученной от пользователя.

Форма для заказа билетов в театр была оформлена с помощью CSS, что сделало интерфейс более удобным и привлекательным для пользователя. После отправки формы, сервер генерирует страницу с результатами, на которой отображается введенная информация, а также соответствующие элементы оформления.

Данная работа позволила закрепить навыки организации серверной части приложения, обработки POST-запросов для получения данных из HTML-формы и дальнейшего формирования страницы с результатами..

## ПРИЛОЖЕНИЕ А

### 2.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Театр - Заказ билетов</title>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="2.css">
</head>
<body>
  <h2>Заказ билетов в театр</h2>
  <form action="/register" method="post">
    <label for="username">Ваше имя:</label><br>
    <input type="text" id="username" name="username" placeholder="Введите
ваше имя" minlength="3" required><br>

    <label for="email">Электронная почта:</label><br>
    <input type="email" id="email" name="email"
placeholder="example@domain.com" required><br>

    <label for="phone">Номер телефона:</label><br>
    <input type="tel" id="phone" name="phone" placeholder="+7 999 123-45-
67" required><br>

    <fieldset>
      <legend>Выберите спектакль:</legend>
      <input type="radio" id="play1" name="play" value="Гамлет" required>
      <label for="play1">Гамлет</label><br>
```



```
<input type="radio" id="play2" name="play" value="Ромео и
Джульетта">
```

```
<label for="play2">Ромео и Джульетта</label><br>
```

```
<input type="radio" id="play3" name="play" value="Чайка">
```

```
<label for="play3">Чайка</label><br>
```

```
</fieldset>
```

```
<label for="seatType">Тип места:</label><br>
```

```
<select id="seatType" name="seatType" required>
```

```
<option value="Standard">Стандарт</option>
```

```
<option value="VIP">VIP</option>
```

```
</select><br>
```

```
<fieldset>
```

```
<legend>Дополнительные услуги:</legend>
```

```
<input type="checkbox" id="snacks" name="extras" value="Снэки">
```

```
<label for="snacks">Снэки</label><br>
```

```
<input type="checkbox" id="drink" name="extras" value="Напитки">
```

```
<label for="drink">Напитки</label><br>
```

```
<input type="checkbox" id="program" name="extras"
value="Программа">
```

```
<label for="program">Программа спектакля</label><br>
```

```
</fieldset>
```

```
<label for="comments">Комментарии:</label><br>
```

```
<textarea id="comments" name="comments" rows="4" placeholder="Особые
пожелания или комментарии"></textarea><br>
```

```
<label for="date">Дата посещения:</label><br>
```

```
<input type="date" id="date" name="date" required><br>
```

```
<input type="submit" value="Оформить заказ">
</form>

<script src="2.js"></script>
</body>
</html>
```

## ПРИЛОЖЕНИЕ Б

### 2.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));

app.get('/', (req, res) => {
  res.sendFile(path.join(__dirname, '2.html'));
});

app.post('/register', (req, res) => {
  const { username, email, phone, play, seatType, extras, comments, date } =
    req.body;

  if (!username || !email || !phone || !play || !seatType || !date) {
    return res.status(400).send('Пожалуйста, заполните все обязательные
поля!');
  }

  const extrasList = Array.isArray(extras) ? extras.join(', ') : extras || 'Нет';

  res.send(`
    <!DOCTYPE html>
```

```

<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Заказ подтверждён</title>
  <link rel="stylesheet" href="2.1.css">
</head>
<body>
  <h2>Спасибо за заказ, ${username}!</h2>
  <p><strong>Спектакль:</strong> ${play}</p>
  <p><strong>Тип места:</strong> ${seatType}</p>
  <p><strong>Дата:</strong> ${date}</p>
  <p><strong>Дополнительные услуги:</strong> ${extrasList}</p>
  <p><strong>Комментарий:</strong> ${comments || 'Нет'}</p>
  <p><strong>Мы отправили подтверждение на ваш email:</strong>
  ${email}</p>
  <p><strong>Мы отправили подтверждение на ваш email:
  </strong>${email}</p>

</body>
</html>

`);
});

app.listen(port, () => {
  console.log(`Сервер запущен на http://localhost:${port}`);
});

```

## ПРИЛОЖЕНИЕ В

### 2.CSS

```
body {  
    font-family: 'Georgia', serif;  
    background-color: #f7f1e3;  
    margin: 0;  
    padding: 0;  
    display: flex;  
    justify-content: center;  
    align-items: center;  
    min-height: 100vh;  
  
}  
  
form {  
    width: 450px;  
    background-color: #fff4e6;  
    padding: 25px;  
    border-radius: 10px;  
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);  
    display: flex;  
    flex-direction: column;  
    gap: 10px;  
}  
  
h2 {  
    text-align: left;
```

```
    color: #d35400;
    margin-bottom: 200px;
    font-size: 1.8em;
    margin-right: 120px;

}

label {
    font-weight: bold;
    color: #5a3e36;
}

input[type="text"],
input[type="email"],
input[type="tel"],
textarea,
select,
input[type="date"] {
    width: 100%;
    padding: 12px;
    border: 1px solid #c8b6a6;
    border-radius: 5px;
    box-sizing: border-box;
}

fieldset {
    border: 1px solid #c8b6a6;
    padding: 15px;
```

```
border-radius: 5px;  
}
```

```
legend {  
    font-weight: bold;  
    color: #5a3e36;  
}
```

```
button, input[type="submit"] {  
    padding: 12px 20px;  
    background-color: #d35400;  
    color: #fff;  
    border: none;  
    border-radius: 5px;  
    font-size: 16px;  
    cursor: pointer;  
}
```

```
button:hover, input[type="submit"]:hover {  
    background-color: #e67e22;  
}
```

```
input:invalid {  
    border-color: #e74c3c;  
}
```

## ПРИЛОЖЕНИЕ Г

### 2.1.CSS

```
body {  
    font-family: 'Georgia', serif;  
    background-color: #fff4e6;  
    margin: 0;  
    padding: 20px;  
    color: #5a3e36;  
    text-align: center;  
}
```

```
h2 {  
    color: #d35400;  
    font-size: 2em;  
    margin-top: 20px;  
}
```

```
p {  
    font-size: 1.2em;  
    margin: 10px 0;  
    line-height: 1.8;  
}
```