ГУАП

КАФЕДРА № 42

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

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

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

РАБОТА С ФАЙЛАМИ В NODE.JS

по курсу: WEB-ПРОГРАММИРОВАНИЕ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТГР. № 4128		В.А.Тарапанов
	подпись, дата	инициалы, фамилия

Санкт-Петербург 2024

Цель работы

Получение навыков программирования на языке программирования JavaScript в среде Node.JS. Изучение функций для работы с файлами (файловые функции) в среде Node.JS.

Задание

Используя web-страницу, полученную в ходе выполнения лабораторной работы 2, дополнить ее следующими возможностями.

1. Написать JS-скрипт в среде Node.JS, позволяющий загружать файл на сервер, который содержит дополнительную информацию по предметной области лабораторной работы 1. Скрипт должен выводить сообщение с результатом загрузки (успех/неуспех). Необходимо реализовать дополнительным полем в HTML-форме (интегрировать в форму).
2. Дополнить в JS-скрипт из пункта 1 возможностью проверки типа и размера загружаемого файла. Тип файла определяется по номеру варианта из таблицы 1, а размер файла – из таблицы 2.
3. Добавить в получившуюся в пункте 1 web-страницу кнопку, которая позволяет записать всю текстовую информацию из полей формы с названиями полей формы в текстовый файл (можно *.txt). Для работы кнопки необходимо дополнить JS-скрипт из пункта 2. После сохранения информации в файл форма должна предложить просмотреть/скачать получившийся файл. Когда скачивание будет завершено, необходимо выдать сообщение согласно номеру варианта из таблицы 3. Текст необходимо структурировать.

Вариант: 9	Тематика HTML-формы: Заказ билетов в театр
	Тип файла: .jpg
	Размер файла: 100-200 кб
	Содержание сообщения: Время последнего доступа к файлу

Выполнение работы

На рисунке 1 представлена начальная форма, с заполненными полями.

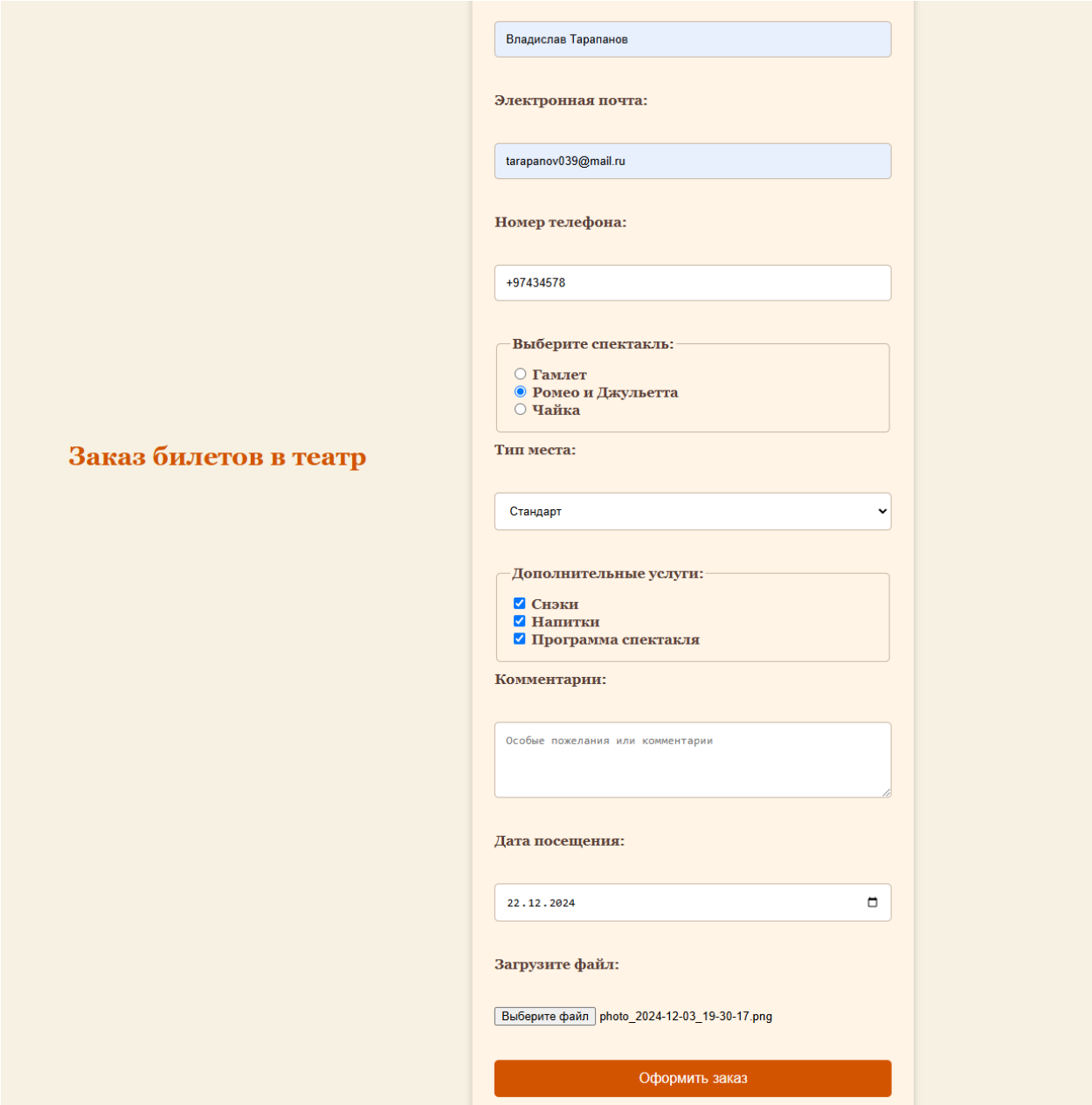

Рисунок 1 – Тестовая форма

На рисунке 2 представлено сообщение об ошибке , которое оповещает пользователя, о том, что он загрузил файл не того формала.

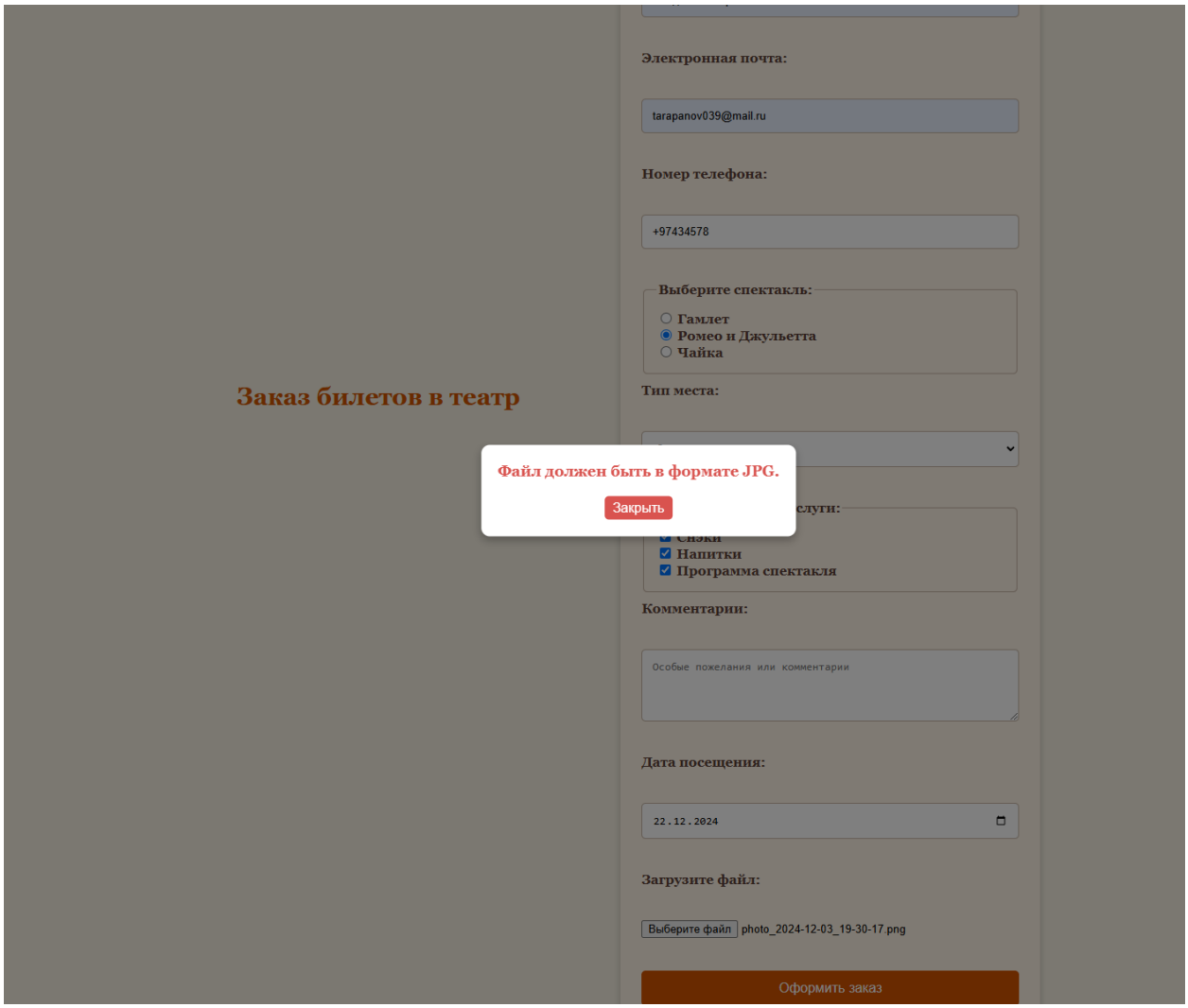

Рисунок 2 – Модальное окно с ошибкой

После нажатия на кнопку «Оформить заказ» с файлом правильного формата, мы переходим на страницу подтверждения, которая представлена на рисунке 3. Файл, который мы загружаем, сохраняется на сервере в каталог uploads.

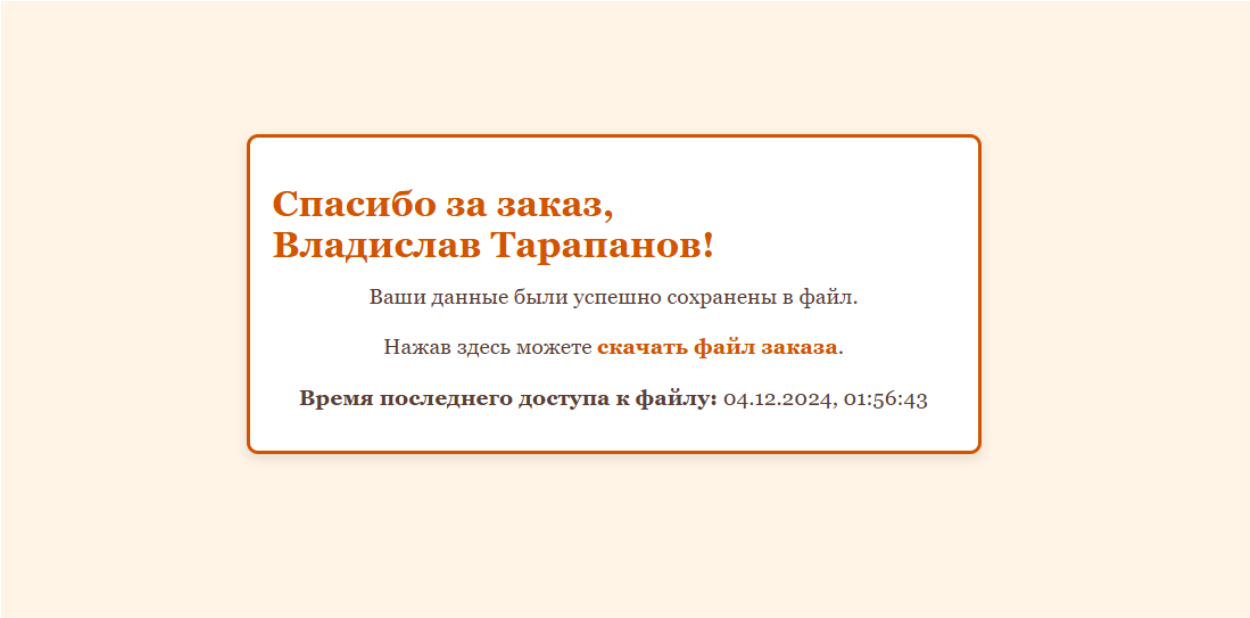

Рисунок 3 – Форма подтверждения

При нажатии кнопки “Скачать файл заказа” мы получаем текстовый файл, содержащий информацию о заказе. На рисунке 4 представлено содержимое файла order.txt.

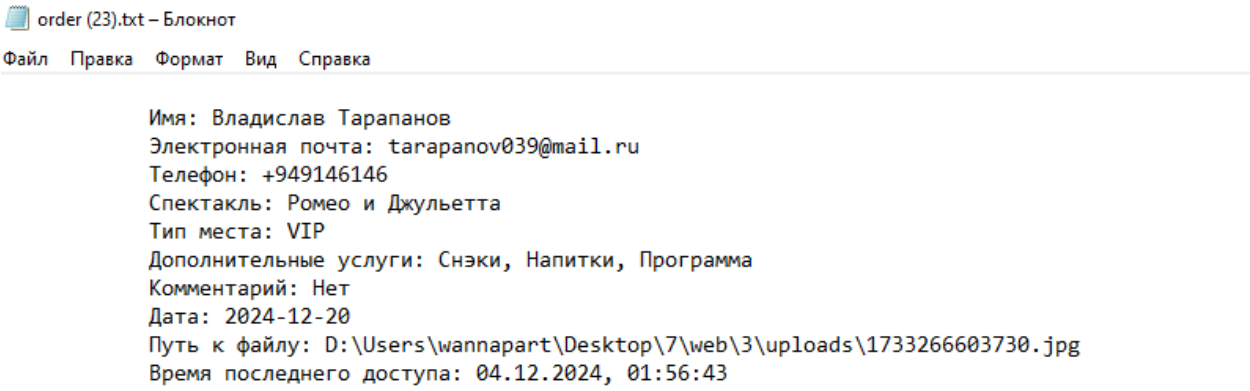

Рисунок 4 – Содержание файла order.txt

Вывод

В результате выполнения лабораторной работы были успешно реализованы различные функциональные возможности для веб-страницы, направленные на улучшение взаимодействия с пользователем и добавление новых возможностей для работы с файлами на сервере. В ходе работы был создан и интегрирован JavaScript-скрипт в среде Node.js, который позволил загружать файл на сервер и проверять тип и размер этого файла.

Была разработана система для обработки ошибок, которая обеспечивает отображение модальных окон с информацией о возникших проблемах при загрузке файлов или других действиях на странице. Это повышает удобство и информативность интерфейса для пользователя, предотвращая недоразумения и улучшая опыт взаимодействия с веб-приложением.

Также был реализован функционал для скачивания информации в виде файла. Реализованный механизм позволяет пользователю скачать необходимую информацию с сервера в формате текстового файла, что расширяет возможности использования веб-страницы. В случае возникновения ошибок при скачивании файла, пользователю отображается соответствующее сообщение об ошибке через модальное окно.

Добавлена возможность для пользователя записывать данные в файл, что дает дополнительную гибкость в работе с веб-страницей. Эта функция предоставляет пользователю возможность сохранять выбранные данные, что может быть полезно для различных сценариев использования.

Таким образом, в ходе лабораторной работы были достигнуты поставленные цели, улучшены функциональные возможности веб-страницы, обеспечено более удобное взаимодействие с пользователем, а также реализована система работы с файлами на сервере, включающая как загрузку, так и скачивание файлов.

ПРИЛОЖЕНИЕ А

ЛИСТНИГ 3.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="3.css">
</head>
<body>
    <h2>Заказ билетов в театр</h2>
    <form id="ticketForm" enctype="multipart/form-data">
        <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>

            <label for="file">Загрузите файл:</label><br>
            <input type="file" id="file" name="file" accept=".jpg"
required><br>

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

        <div id="errorModal">
            <div id="errorModalContent">
                <h3 id="errorText"></h3>
                <button id="closeErrorModal">Закрыть</button>
            </div>
        </div>

        <script>
            const ticketForm = document.getElementById('ticketForm');
            const errorModal = document.getElementById('errorModal');
            const errorText = document.getElementById('errorText');
            const closeErrorModal =
document.getElementById('closeErrorModal');

            ticketForm.addEventListener('submit', async (event) => {
                event.preventDefault();

                const formData = new FormData(ticketForm);
```

```
            try {
                const response = await fetch('/register', {
                    method: 'POST',
                    body: formData,
                });

                if (!response.ok) {
                    const errorMessage = await response.text();
                    showErrorModal(errorMessage);
                } else {
                    const successHtml = await response.text();
                    document.body.innerHTML = successHtml;
                }
            } catch (err) {
                showErrorModal('Произошла ошибка. Попробуйте снова.');
            }
        });

        closeErrorModal.addEventListener('click', () => {
            errorModal.style.display = 'none';
        });

        function showErrorModal(message) {
            errorText.textContent = message;
            errorModal.style.display = 'block';
        }
    </script>
</body>
</html>
```

ПРИЛОЖЕНИЕ Б

Листинг 3.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;
}

#errorModal {
    display: none;
    position: fixed;
    z-index: 1000;
    left: 0;
    top: 0;
```

```
    width: 100%;
    height: 100%;
    background-color: rgba(0, 0, 0, 0.5);
}

#errorModalContent {
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
    background: white;
    padding: 20px;
    border-radius: 10px;
    text-align: center;
    box-shadow: 0 4px 10px rgba(0, 0, 0, 0.3);
}

#errorModalContent h3 {
    margin: 0 0 10px;
    font-size: 18px;
    color: #d9534f;
}

#errorModalContent button {
    margin-top: 10px;
    padding: 5px 10px;
    background: #d9534f;
    color: white;
    border: none;
    border-radius: 5px;
```

```
    cursor: pointer;
}

#errorModalContent button:hover {
    background: #c9302c;
}
```

ПРИЛОЖЕНИЕ В

ЛИСТИНГ 3.1.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;
    margin-bottom: 10px;
}

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

a {
    color: #d35400;
    text-decoration: none;
    font-weight: bold;
}
```

```
a:hover {
    text-decoration: underline;
}

.container {
    max-width: 600px;
    margin: 0 auto;
    background: #fff;
    padding: 20px;
    border-radius: 10px;
    box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1);
    border: 3px solid #d35400; /* Добавляем рамку */
}
```

ПРИЛОЖЕНИЕ Г

Листинг 3.JS

```
const express = require('express');
const bodyParser = require('body-parser');
const path = require('path');
const multer = require('multer');
const fs = require('fs');

const app = express();

const port = 3000;

const storage = multer.diskStorage({
    destination: (req, file, cb) => {
        const uploadPath = path.join(__dirname, 'uploads');
        if (!fs.existsSync(uploadPath)) {
            fs.mkdirSync(uploadPath);
        }
        cb(null, uploadPath);
    },
    filename: (req, file, cb) => {
        cb(null, Date.now() + path.extname(file.originalname));
    }
});

const upload = multer({
    storage: storage,
    limits: { fileSize: 200000 },
    fileFilter: (req, file, cb) => {
```

```
        const allowedTypes = ['image/jpeg'];
        if (!allowedTypes.includes(file.mimetype)) {
            return cb(new Error('Файл должен быть в формате JPG.'));
        }
        cb(null, true);
    }
}).single('file');

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

app.get('/', (req, res) => {
    res.sendFile(path.join(__dirname, '3.html'));
});

app.post('/register', (req, res) => {
    upload(req, res, (err) => {
        if (err instanceof multer.MulterError) {
            if (err.code === 'LIMIT_FILE_SIZE') {
                return res.status(400).send('Размер файла должен быть не
более 200 кб.');
            }
        } else if (err) {
            return res.status(400).send(err.message);
        }

        if (!req.file) {
            return res.status(400).send('Файл обязателен для загрузки.');
        }
```

```
        if (req.file.size < 100000) {
            return res.status(400).send('Размер файла должен быть не менее 100 кб.');
        }

        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';

        // Сохраняем данные в файл
        const orderDetails = `
            Имя: ${username}
            Электронная почта: ${email}
            Телефон: ${phone}
            Спектакль: ${play}
            Тип места: ${seatType}
            Дополнительные услуги: ${extrasList}
            Комментарий: ${comments || 'Нет'}
            Дата: ${date}
            Путь к файлу: ${req.file.path}
            Время последнего доступа: ${new Date().toLocaleString()}
        `;
```

```
        const filePath = path.join(__dirname, 'order.txt');
        fs.writeFileSync(filePath, orderDetails);

        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="3.1.css"> <!-- Ссылка на файл
стилей -->
            </head>
            <body>
                <div class="container">
                    <h2>Спасибо за заказ, ${username}!</h2>
                    <p>Ваши данные были успешно сохранены в
файл.</p>
                    <p>
                        Нажав здесь можете <a href="/order.txt"
download>скачать файл заказа</a>.
                    </p>
                    <p><strong>Время последнего доступа к
файлу:</strong> ${new Date().toLocaleString()}</p>
                </div>
            </body>
            </html>
        `);
```

```
    });
});

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