

Министерство образования Республики Беларусь  
Учреждение образования  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОННИКИ  
Факультет компьютерных систем и сетей  
Кафедра электронных вычислительных машин

Лабораторная работа №2  
Создание серверной части и интерфейса программы для работы с БД

Студент:

М.А. Ходосевич

Преподаватель:

А.И. Крюков

МИНСК 2024

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	3
1 ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ.....	4
1.1 Серверное приложение.....	4
1.2 Клиентское приложение. Интерфейс .....	4
2 РАЗРАБОТКА СЕРВЕРНОЙ ЧАСТИ.....	5
3 РАЗРАБОТКА КЛИЕНТСКОЙ ЧАСТИ .....	7
ЗАКЛЮЧЕНИЕ.....	9

## **ВВЕДЕНИЕ**

В лабораторной работе выполняется разработка программного обеспечения: клиентской и серверной части приложения, для работы с базой данных Postgres.

Темой данной лабораторной работы является разработка интерфейса программы или административной панели в организации «Кафе-бар».

Серверная часть приложения написана на языке Node.js. Клиентская часть написана на JavaScript фреймворке React. Взаимодействие клиента и сервера осуществляется посредством REST API.

# **1 ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ**

Техтребования содержат принципы построения взаимодействия клиент-серверного приложения в рамках работы с базой данных, но оторвано от конкретной реализации будь то Postgres или BearkleyDB.

Техтребования подразделяются на требования для серверного приложения и требования для интерфейса клиентского приложения.

## **1.1 Серверное приложение**

1) Серверное приложение для реализации соединения с базой данных Postgres будет написано на языке NodeJS.

2) Должны быть предусмотрены CRUD операции для всех таблиц из UML-диаграммы.

3) Серверное приложение должно представлять из себя REST API сервер.

4) Серверные операции должны быть описаны обще, для дальнейшего масштабирования и наследования.

5) В серверном приложении должны быть описаны все используемые сущности базы данных.

6) Приложение должно быть оптимизированным.

## **1.2 Клиентское приложение. Интерфейс**

1) Клиентское приложение должно быть написано в SPA, для обеспечения быстродействия и реактивности. Использовать один из популярных фреймворков.

2) Интерфейс приложения должен отвечать принципам UI/UX. Дизайн должен быть удобен, понятен и однозначен.

3) Взаимодействие с серверным приложением должно происходить через REST API.

4) Приложение должно иметь минималистичный дизайн.

5) Приложение должно быть оптимизированным.

## 2 РАЗРАБОТКА СЕРВЕРНОЙ ЧАСТИ

Серверное приложение представляет собой набор функций, реализующих основные операции при работе с базами данных. Создание и удаление данных во всех таблицах схемы организации происходит посредством написания запросов в поле для пользовательских запросов. Как было сказано выше, серверная часть написана на языке Node.js. Для начала выполняется подключение к уже созданной базе данных:

```
const pool = new Pool({
  user: 'postgres',
  host: 'localhost',
  database: 'cafebar',
  password: '2223', // ваш пароль
  port: 5432,
});
```

Далее в коде прописываются GET запросы для получения данных из таблиц. Запрос возвращает все строки из заданной таблицы и их атрибуты. Пример такого запроса на примере таблицы посетителей приведен ниже.

```
const getTableData = async (req, res, table) => {
  try {
    const client = await pool.connect();
    const result = await client.query(`SELECT * FROM ${table}`);
    const data = result.rows;
    res.json(data);
    client.release();
  } catch (err) {
    console.error(err);
    res.status(500).send('Internal server error');
  }
};

app.get('/client', async (req, res) => {
  await getTableData(req, res, 'client');
});
```

По такому принципу написаны все запросы для вывода информации. Также в серверной части описан GET запрос для пользовательских запросов. Пользовательские запросы нужны для того, чтобы пользователь смог добавить или удалить данные из таблицы или осуществить необходимую ему выборку. Пользовательский запрос приведен ниже.

```
app.get('/query', async (req, res) => {
  const { query } = req.query;
  try {
    const result = await pool.query(query);
    res.json(result.rows);
  } catch (err) {
```

```
    console.error(err);  
    res.status(500).send('Internal server error');  
  }  
});
```

При успешном запуске сервера в консоль будет выводиться сообщение о том, что сервер работает и его порт. Это приведено на рисунке 2.1.

A screenshot of a terminal window with a dark background. The prompt is 'khodosevich@MacBook-Pro-Matvey server %'. The command entered is 'node server.js'. The output is 'Server is running at http://localhost:3001'. A cursor is visible on the line below the output.

```
khodosevich@MacBook-Pro-Matvey server % node server.js  
Server is running at http://localhost:3001  
█
```

Рисунок 2.1 – Запуск сервера

### 3 РАЗРАБОТКА КЛИЕНТСКОЙ ЧАСТИ

Разработка клиентской части включает в себя создание пользовательского интерфейса для удобства работы с сервером и вывода информации. Клиентская часть представляет собой страницу в браузере, написанную на React.

На главной странице находятся все таблицы из базы данных, по нажатию на кнопки «Развернуть» появляется таблица с данными. Вверху страницы находится поле для ввода пользовательского запроса и примеры запросов для добавления и удаления таблиц.

В клиентской части прописаны запросы для доступа к базе данных и настроен сам интерфейс со стилями. Ниже будет приведен запрос к базе данных для получения списка посетителей.

```
const fetchTable = () => {  
  fetch(request)  
    .then(response => response.json())  
    .then(data => setTableData(data))  
    .catch(error => console.error('Ошибка:', error));  
}
```

Далее на рисунке 3.1 приведен главный экран приложения с примерами запросов и списками таблиц.

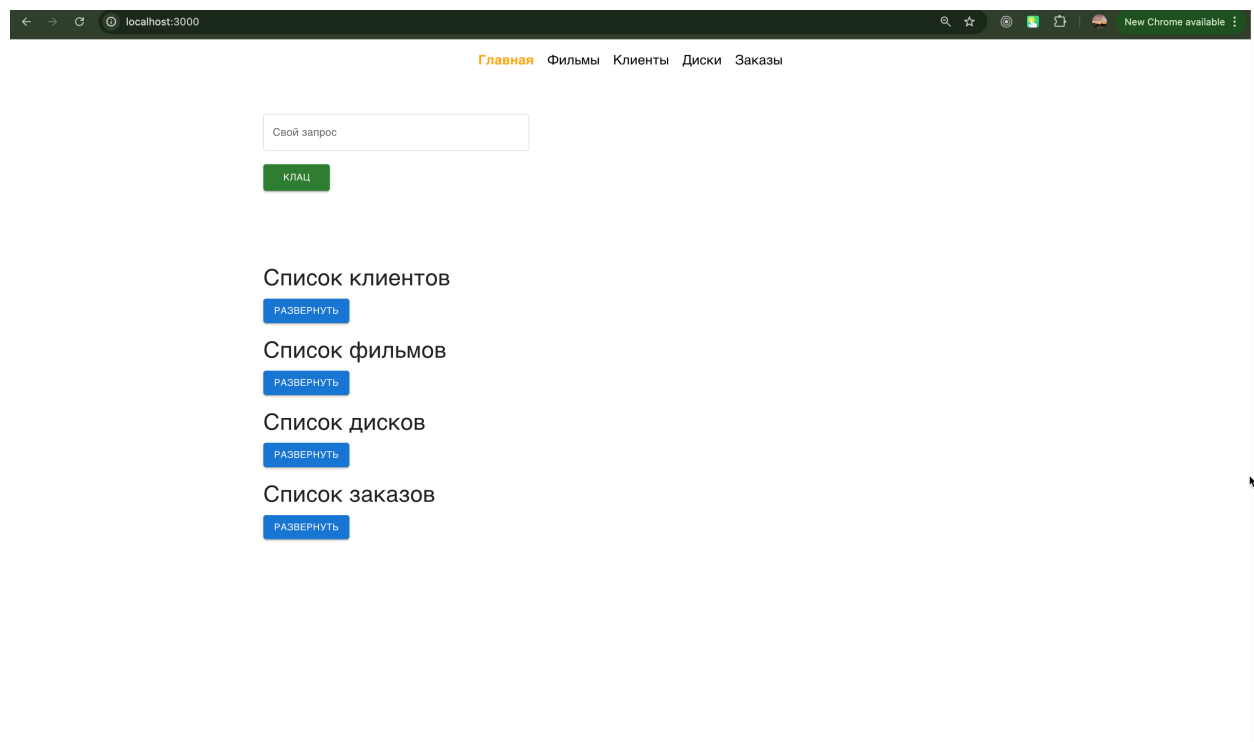
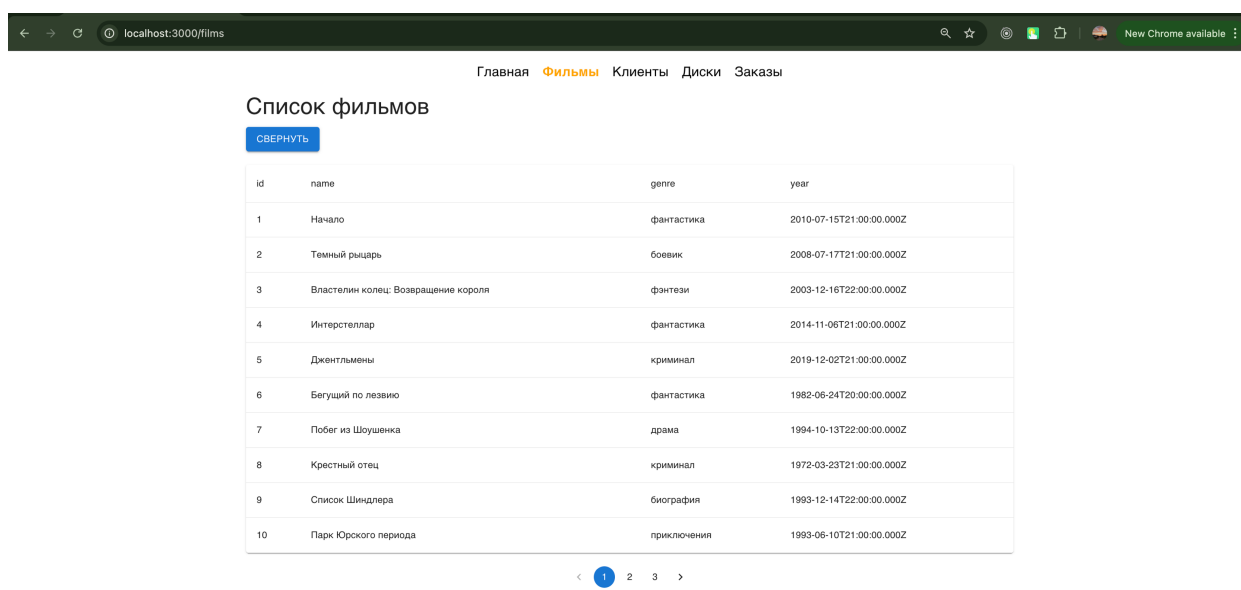


Рисунок 3.1 – Основная страница

На рисунке 3.2 будет приведен развернутый вывод таблицы фильмов (нажатие на кнопку «Развернуть»).



id	name	genre	year
1	Начало	фантастика	2010-07-15T21:00:00.000Z
2	Темный рыцарь	боевик	2008-07-17T21:00:00.000Z
3	Властелин колец: Возвращение короля	фэнтези	2003-12-16T22:00:00.000Z
4	Интерстеллар	фантастика	2014-11-06T21:00:00.000Z
5	Джентльмены	криминал	2019-12-02T21:00:00.000Z
6	Бегущий по лезвию	фантастика	1982-06-24T20:00:00.000Z
7	Побег из Шоушенка	драма	1994-10-13T22:00:00.000Z
8	Крестный отец	криминал	1972-03-23T21:00:00.000Z
9	Список Шиндлера	биография	1993-12-14T22:00:00.000Z
10	Парк Юрского периода	приключения	1993-06-10T21:00:00.000Z

Рисунок 3.2 – Развернутый список таблиц



## **ЗАКЛЮЧЕНИЕ**

В результате работы над лабораторной работой было разработано программное обеспечение с клиентской и серверной частями приложения для работы с базой данных. Приложение представляет собой платформу для организации работы кафе-бара. Были описаны как реализация серверной части, так и реализация клиентской части, приведены части кода. В итоге было получено работоспособное приложение для организации с возможностью просмотра, добавления и удаления данных.