|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования«МИРЭА - Российский технологический университет»РТУ МИРЭА |

Институт информационных технологий (ИТ)

Кафедра инструментального и прикладного программного обеспечения (ИиППО)

|  |  |
| --- | --- |
| **ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №7** | |
| **по дисциплине** | |
| «Разработка клиент-серверных приложений»  на тему  **«Создание CRUD приложения на React»** | |
| Выполнил студент группы ИКБО-20-19 | Эртек Х.И. |
|  |  |
| Принял | ассистент, Коваленко М.А. |

Практическая работа выполнена «25» апреля 2022 г.

(подпись студента)

Зачтено «\_\_» апреля 2022 г.

(подпись студента)

Москва 2022

**Цель практической работы:**

Создать React приложение с базой данных PostgreSQL.

**Задачи:**

* Реализовать запись и удаление информации из базы данных PostgreSQL в соответствии с полученными запросами HTTP.

**Описание выполнения работы:**

Перед созданием React приложения необходимо задать новый профиль в СУБД PostgreSQL через консоль psql и создать в новом профиле для работы новую базу данных, которая представлена на рисунке 1.

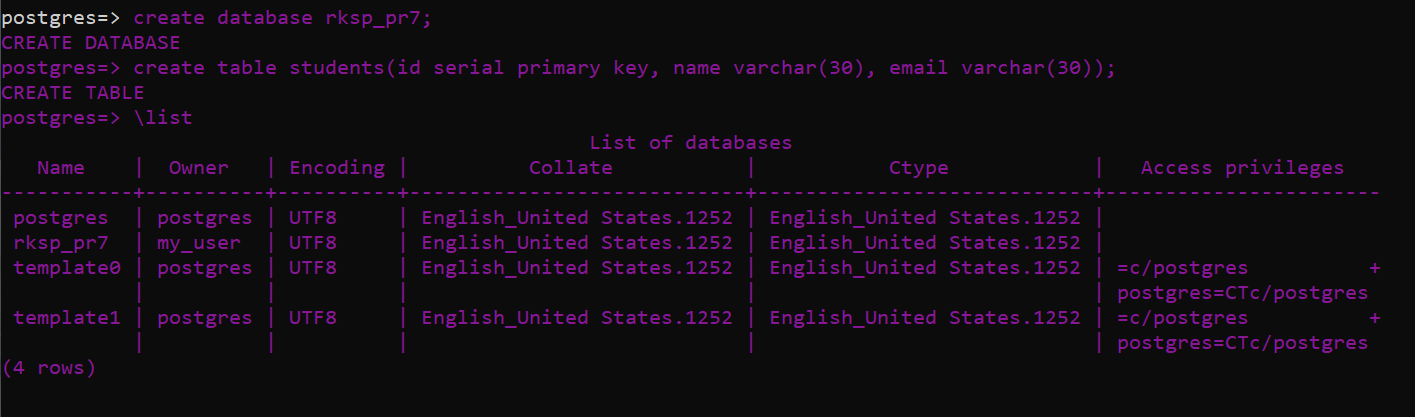


Рисунок 1 ‒ Создание профиля my\_user и создание базы данных rksp\_pr7

После того, как база данных была создана и был произведён вход с созданного профиля my\_user для последующих проверок корректной работы взаимосвязи React приложения с PostgreSQL, внесём несколько значений в базу данных rksp\_pr7, в таблицу students с параметрами name и email.

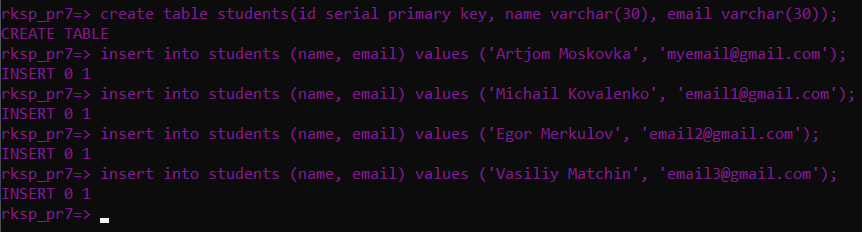


Рисунок 2 ‒ Внесение новых студентов в таблицу

Далее необходимо создать папку node-postgres (основа для связи React и PostgresSQL) и в командной строке пропишем следующую команду **npm init**, которая представлена на рисунке 3.

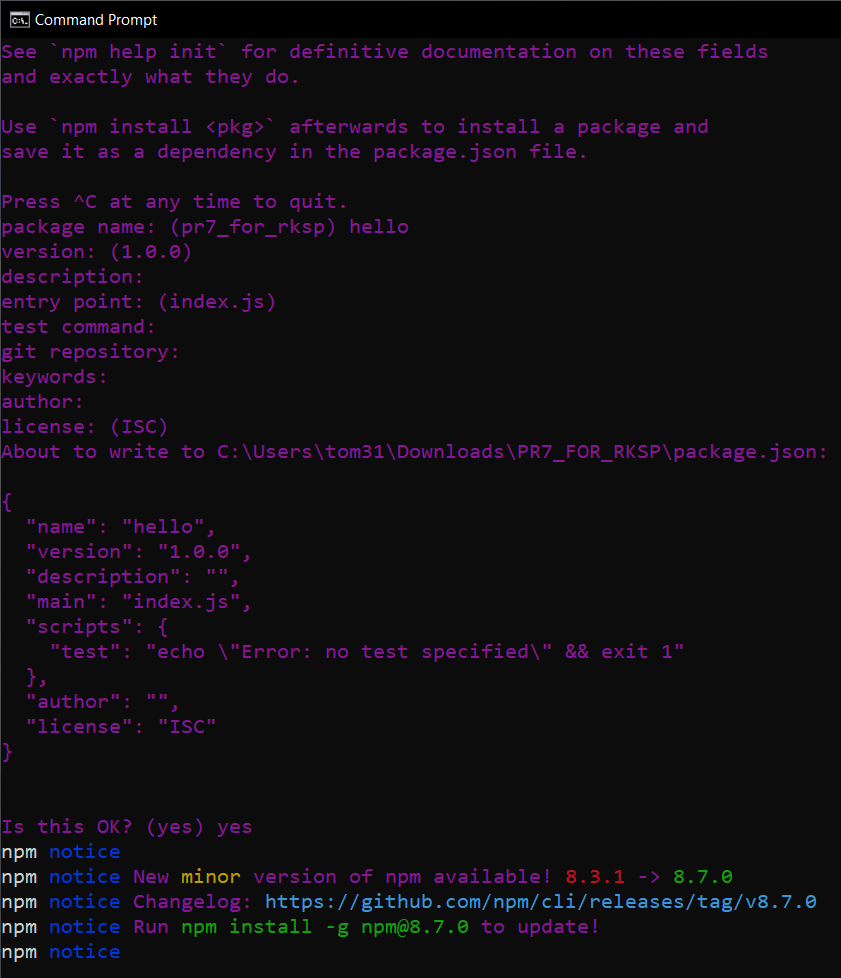


Рисунок 3 ‒ Создание package.json

После создания package.json необходимо установить пакеты для связей React с PostgresSQL через веб-фреймворк express. Как только все подготовлено, переходим в IDE и создаем файл index.js, заполняя его следующим кодом, представленным на рисунке 4.

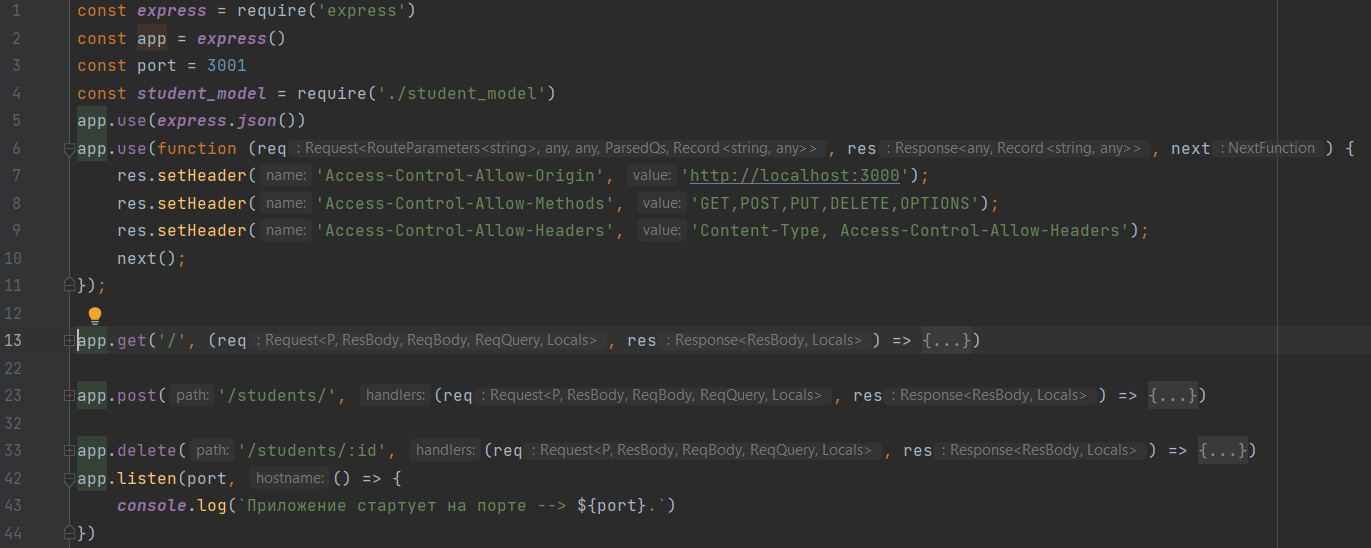


Рисунок 4 ‒ Листинг файла index.js

Был написан код для тестового сервера связи. Проверим работоспособность и запустим сервер на порте 3001.

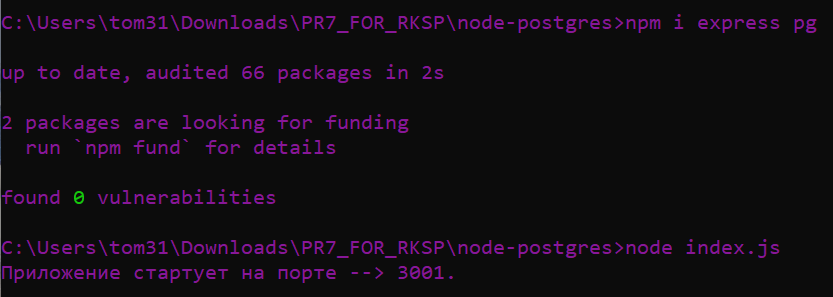


Рисунок 5 ‒ Запуск приложения на порте 3001

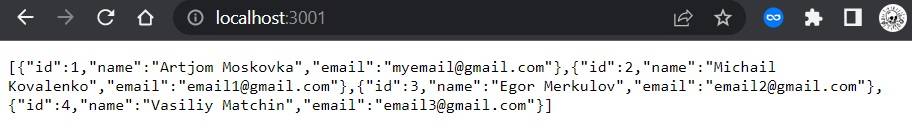


Рисунок 6 ‒ Результат вывода на странице в браузере

Проверка пройдена, теперь необходимо разобраться с доступом к созданной базе данных. Создадим файл student\_model и заполним его следующим кодом:

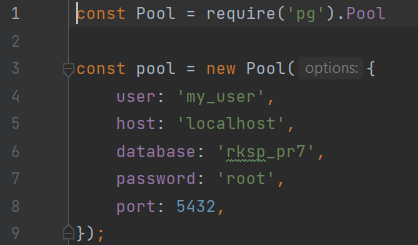


Рисунок 7 ‒ Создание и заполнение файла student\_model.js

Получив доступ к базе данных rksp\_pr7, нужно прописать функции, с помощью которых можно будет манипулировать всеми данными в базе данных. Прописанные функции представлены на рисунке 8.

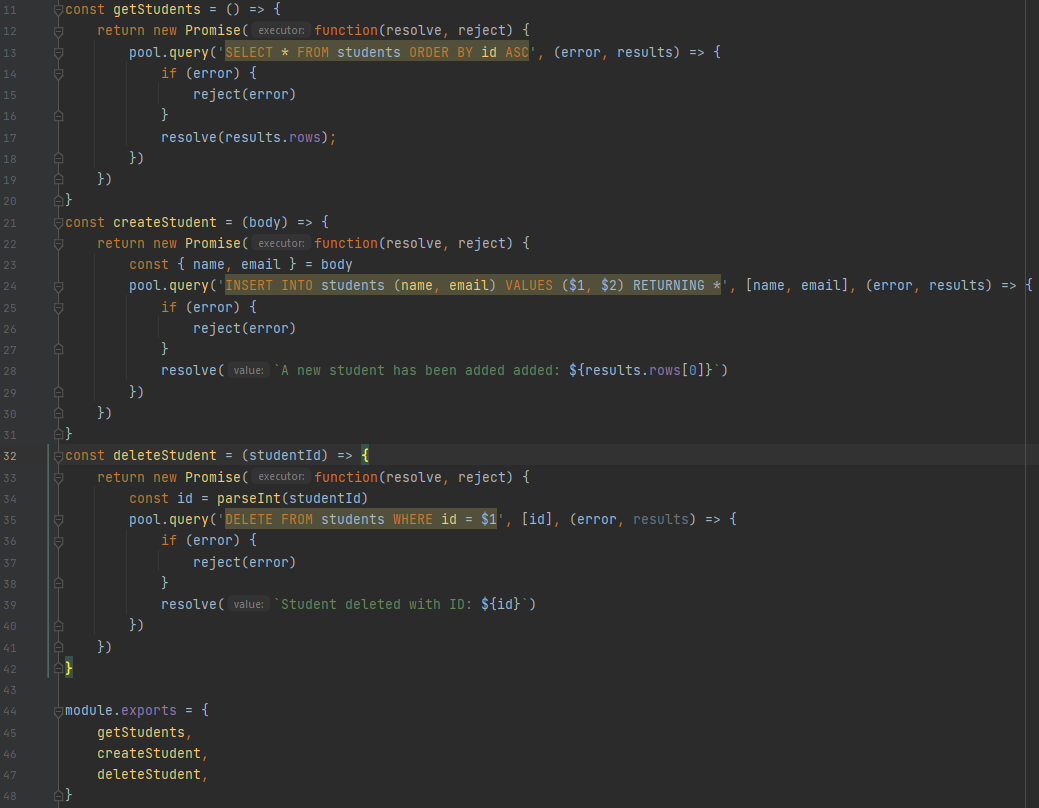


Рисунок 8 ‒ Функции взаимодействия с данными в базе данных

Перезаполним файл index.js, который будет обращаться к модели студента для манипулирования данными внутри базы данных

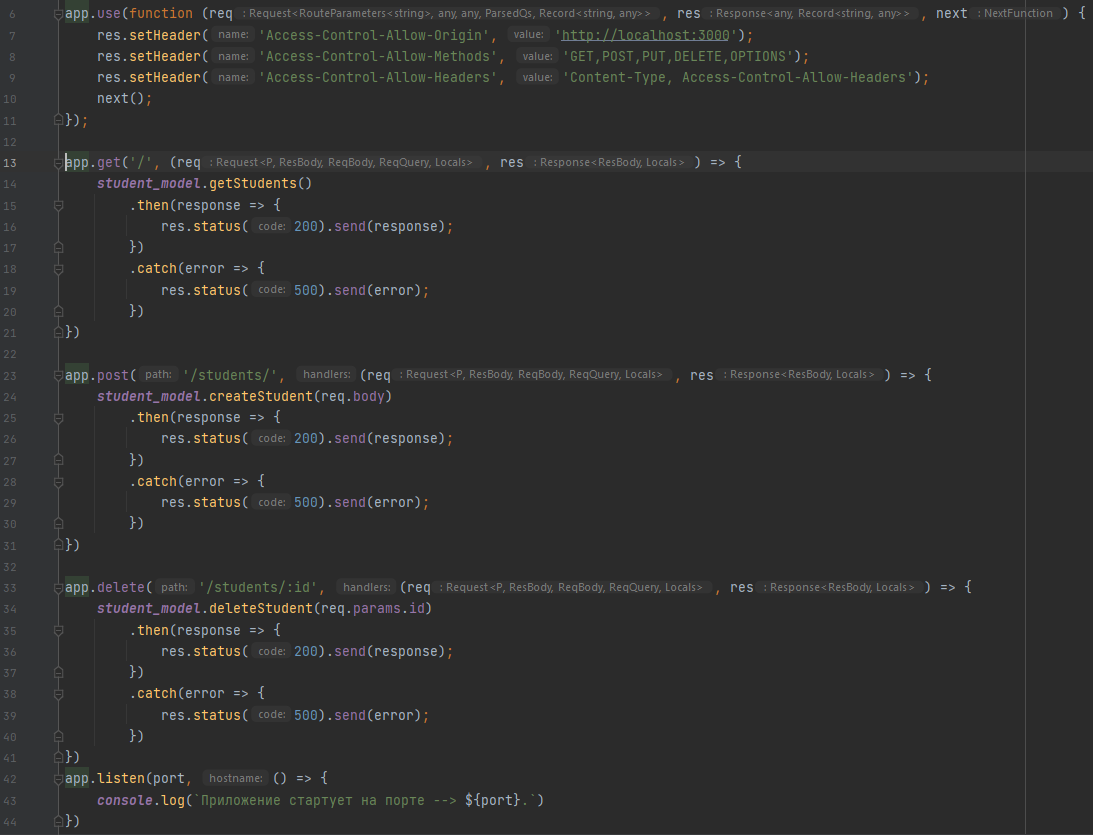


Рисунок 9 ‒ Новый index.js

Для дальнейшей работы и вывода данных необходимо создать папку, в котором настроим API и запросы к базе данных. Для этого выполним команду в командной строке **npx create-react-app react-postgres**. Для дальнейшей корректной работы нужн полностью очистить директорию src. Внутри неё создается новый файл App.js и заполняется следующим образом.



Рисунок 10 ‒ Создание и заполнение App.js (часть 1)

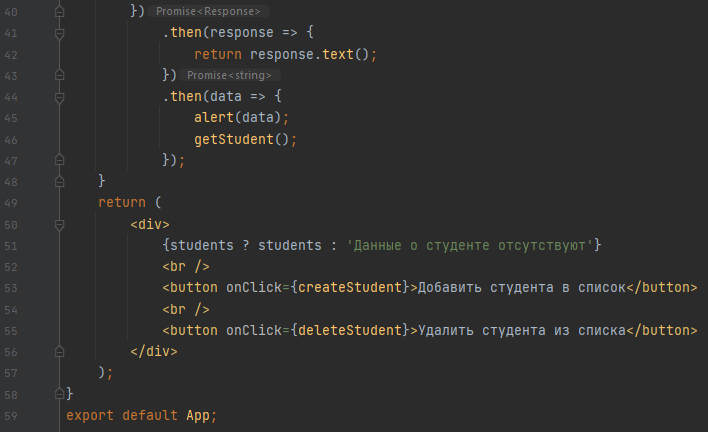


Рисунок 11 ‒ Создание и заполнение App.js (часть 2)

Для связи со второй частью проекта внутри директории src пропишем ещё один файл index.js. Код представлен на рисунке 12.

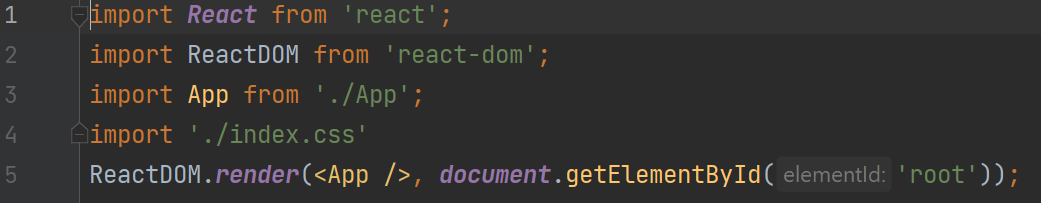


Рисунок 12 ‒ Создание и заполнение файла index.jx

Все готово. Остаётся запустить.

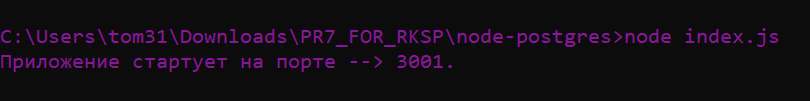


Рисунок 13 ‒ Запуск сервера

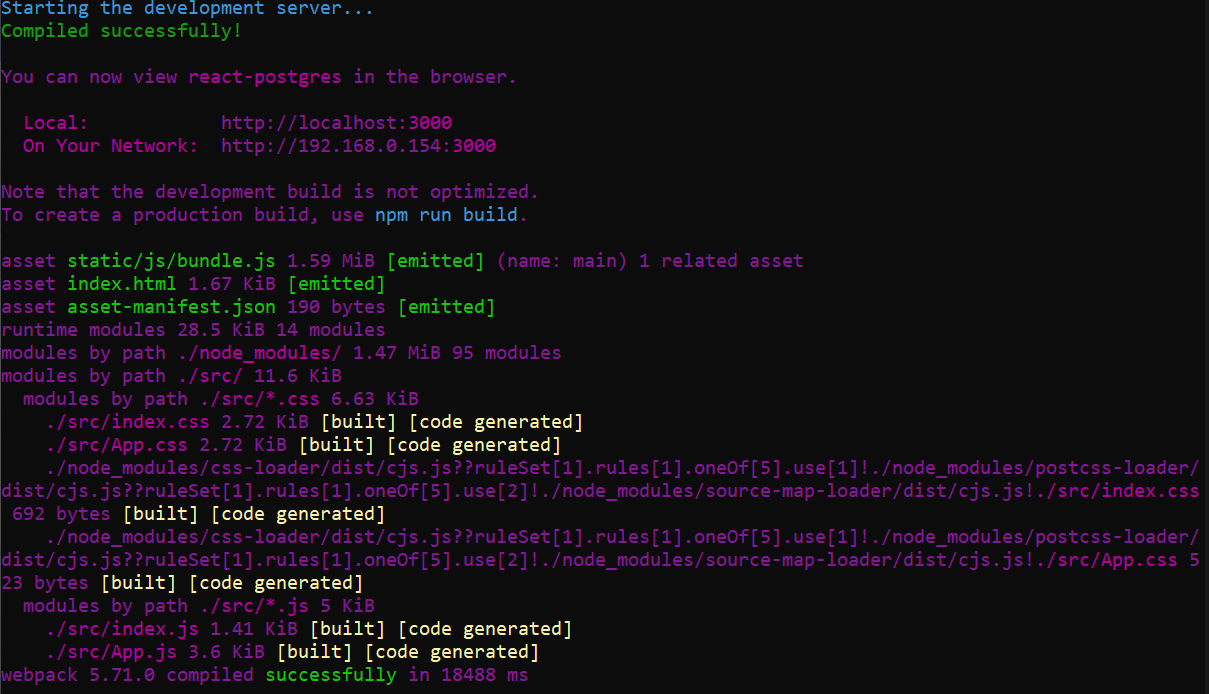


Рисунок 14 ‒ Запуск приложения

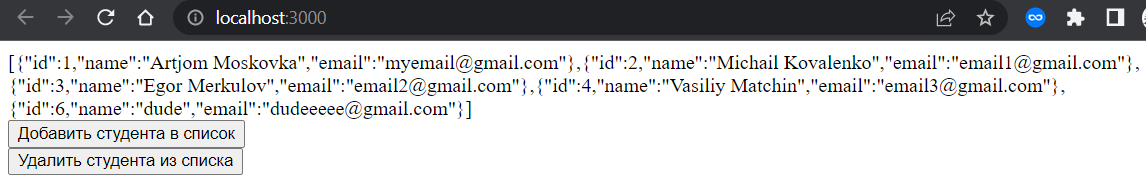


Рисунок 15 ‒ Отображение React приложения на странице браузера

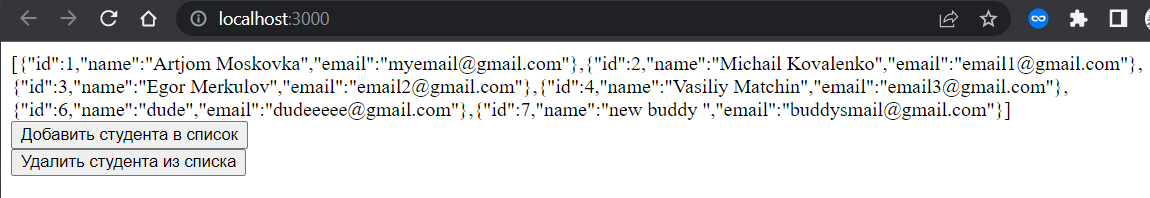


Рисунок 16 ‒ Добавление нового студента в базу данных (id 7)

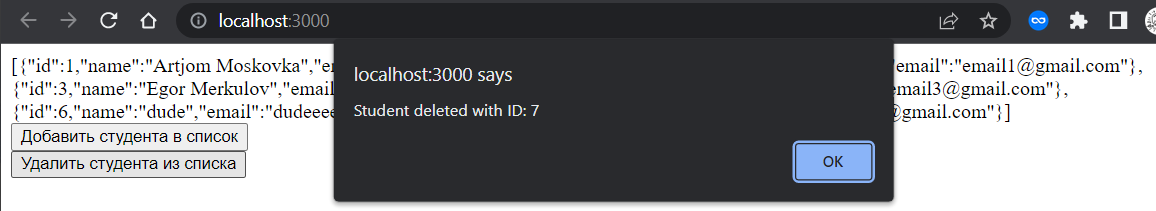


Рисунок 17 ‒ Удаление студента из базы данных по id

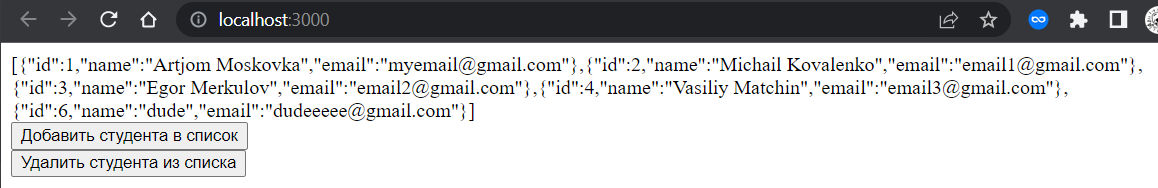


Рисунок 18 ‒ Обновлённая база данных

**Вывод:** В данной практической работе было проведено знакомство с созданием взаимосвязями между React приложением и базами данных PostgresSQL. Были реализованы стандартные функции записи и удаления информации из базы данных.