

**Московский авиационный институт  
(Национальный исследовательский университет)**

Институт: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Базы данных»

**Курсовой проект**  
**по курсу «Базы данных»**  
**Тема: База данных фитнес-клуба**

Студенты: Попов Матвей,  
Морозов Артем,  
Фаттахетдинов  
Сильвестр

Группа: М8О-308Б-20

Преподаватель: Чернышов Л.Н.

Дата:

Оценка:

Москва, 2022

## Оглавление

1. Описание проекта .....	3
2. Техническое задание.....	4
3. Диаграмма базы данных.....	5
4. Структура таблиц.....	6
5. Демонстрация работы приложения.....	7
6. Список литературы .....	9

# 1. Описание проекта

В рамках проекта необходимо спроектировать базу данных для фитнес-клуба, а также реализовать пользовательский интерфейс для администрирования этой базы данных. Для серверной части был выбран язык программирования **Golang**, пользовательский интерфейс будет представлен в виде веб-страниц, написанных с помощью **HTML** и **CSS**. Система управления базой данных — **PostgreSQL**.

## 1.1 Серверная часть (т.н. Backend)

Для разработки приложения было принято использовать популярную схему MVC (Model-View-Controller), немного преобразовав её под наши нужды. В итоге в приложении есть три основных компонента: **controller** (отвечает за взаимодействие backend-части с html-страницами, то есть считывает данные из форм, обрабатывает их, и передаёт их серверной части, также выводит данные из базы данных в html-шаблоны), **model** (содержит структуры всех необходимых сущностей из базы данных, таких как клиент, тренер группа и т.д.) и **server** (содержит функции для работы непосредственно с базой данных, именно с ним плотно взаимодействует контроллер).

## 1.2 Пользовательский интерфейс (т.н. Frontend)

Пользовательский интерфейс представлен следующими веб-страницами:

- index.html — начальная страница сайта
- client.html — страница с функционалом для управления клиентами фитнес-клуба
- trainer.html — страница с функционалом для управления тренерами фитнес-клуба
- group.html — страница с функционалом для управления группами фитнес-клуба
- timetable.html — страница с функционалом для управления расписанием занятий в фитнес-клубе
- answer.html — страница с информацией о проведённой операции над базой данных.

## 1.3 Инструкция для запуска

Для запуска приложения необходимо наличие git и Docker Desktop.

Ссылка на репозиторий проекта: <https://github.com/papey08/DB>

1. Клонировать репозиторий с помощью команды  
`git clone https://github.com/papey08/DB`
2. Переходим в директорию проекта с помощью команды  
`cd DB`
3. Запускаем Docker Desktop
4. Выполняем в командной строке/терминале команду  
`docker-compose up`
5. В любом браузере переходим по адресу localhost:8080.

## 2. Техническое задание

### 2.1 Функционал интерфейса

- Добавление клиента
- Удаление клиента
- Вывод клиентов
- Вывод неподписанных клиентов
- Изменение роста/веса клиента
- Продление абонеента клиента
- Добавление тренера
- Удаление тренера
- Вывод тренеров
- Добавление группы
- Удаление группы
- Вывод всех групп
- Вывод одной группы
- Добавление клиента в группу
- Удаление клиента из группы
- Вывод расписания для каждой группы
- Вывод расписания для каждой программы
- Вывод расписания для каждого тренера
- Добавление записей в расписание
- Удаление записей из расписания

### 2.2 Распределение обязанностей

- **Артем Морозов** (Golang Senior Backend Developer) — написание запроса на вставку тренеров для отладки программы, написание функций для вставки/удаления клиентов/тренеров/групп, для выборки клиентов/тренеров/групп.
- **Сильвестр Фаттяхетдинов** (Senior Frontend Developer) — написание запроса на вставку клиентов для отладки программы, вёрстка HTML-страниц.
- **Матвей Попов** (Golang Fullstack Teamlead Developer) — проектирование всей базы данных, проектирование архитектуры приложения, вёрстка HTML-страниц

### 3. Диаграмма базы данных



Рисунок 1 — диаграмма базы данных

## 4. Структура таблиц

- Client — таблица, содержащая описание клиентов
  - **subscription\_id** — уникальный номер подписки, первичный ключ
  - client\_second\_name — фамилия
  - client\_name — имя
  - client\_third\_name — отчество
  - sex — пол
  - birthdate — дата рождения
  - height — рост
  - weight — вес
  - subscription\_begin — дата начала действия абонемента
  - subscription\_end — дата окончания действия абонемента
- Trainer — таблица, содержащая описание тренеров
  - **trainer\_id** — уникальный номер тренера, первичный ключ
  - trainer\_second\_name — фамилия
  - trainer\_name — имя
  - trainer\_third\_name — отчество
  - trainer\_phone — номер телефона
- Program — таблица, содержащая описание программ
  - **program\_id** — уникальный номер программы, первичный ключ
  - program\_name — название программы
- Fc\_group — таблица, содержащая описание групп
  - **group\_id** — уникальный номер группы, первичный ключ
  - program\_id — внешний ключ
  - notes — примечания
  - trainer\_id — внешний ключ
  - clients\_amount — максимальное количество клиентов в группе
- Group\_client — таблица, содержащая связи клиент-группа
  - group\_id — внешний ключ
  - subscription\_id — внешний ключ
- Times — вспомогательная таблица для расписания
  - **time\_id** — первичный ключ
  - weekday — номер дня недели
  - training\_time — время
  - program\_id — внешний ключ
- Timetable — расписание всех групп
  - time\_id — внешний ключ
  - group\_id — внешний ключ

## 5. Демонстрация работы приложения

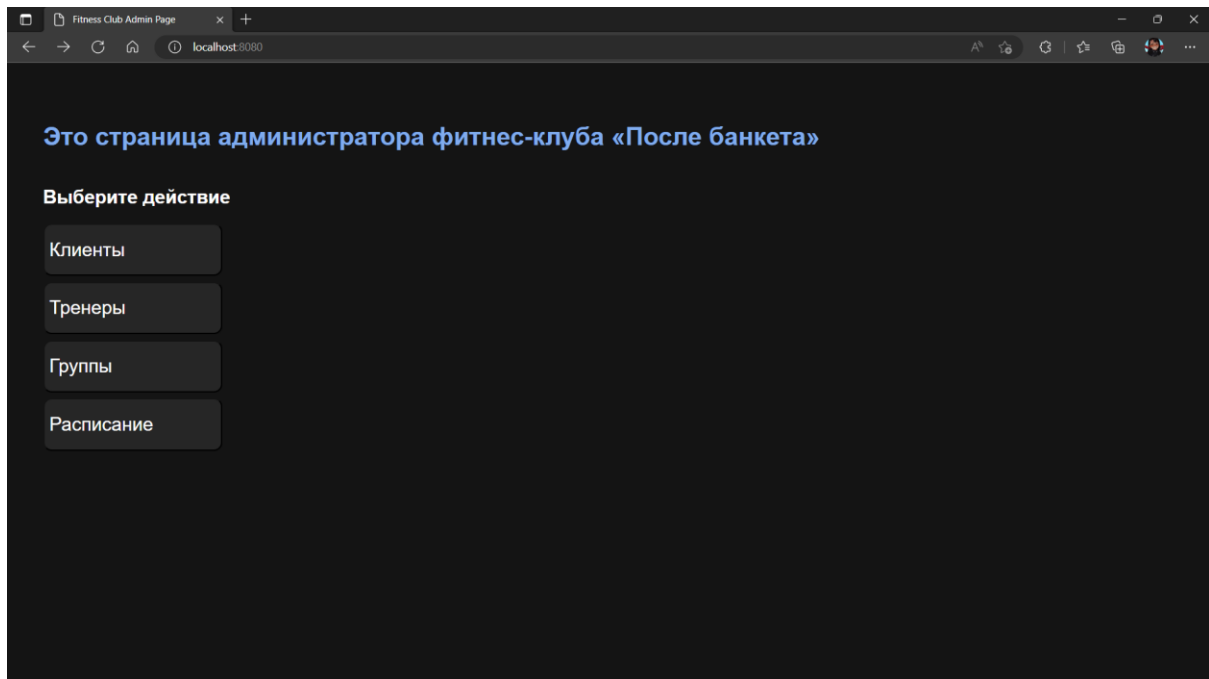


Рисунок 2 — начальная страница

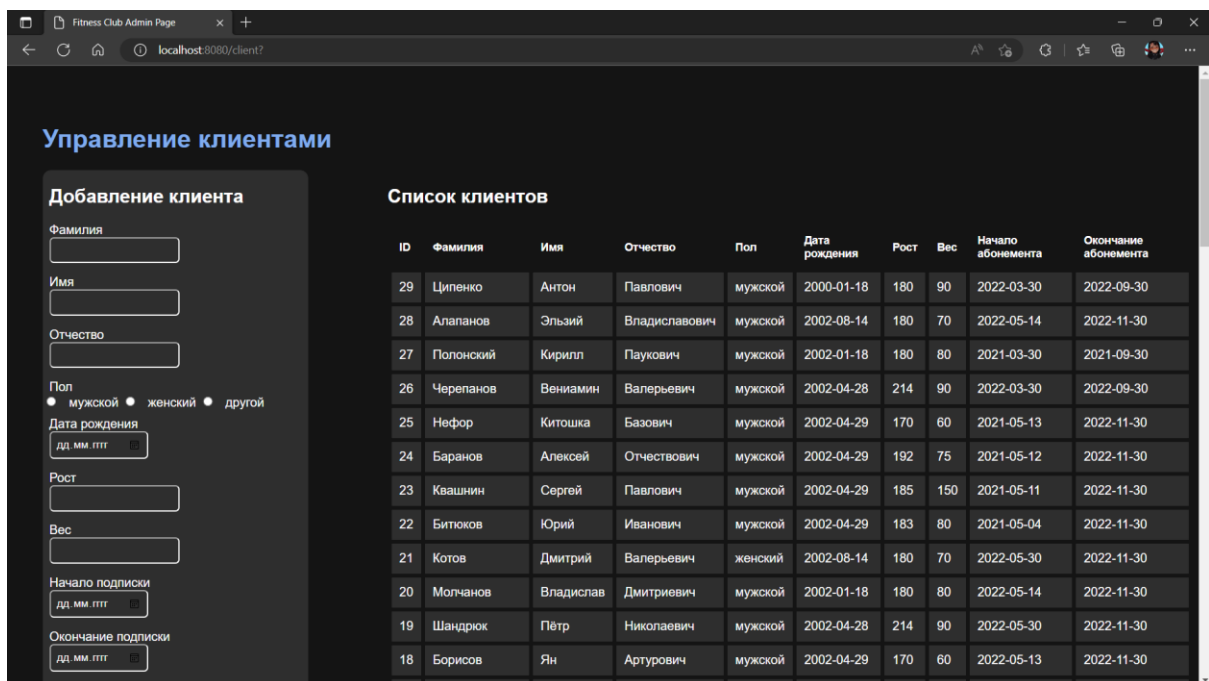


Рисунок 3 — управление клиентами фитнес-клуба

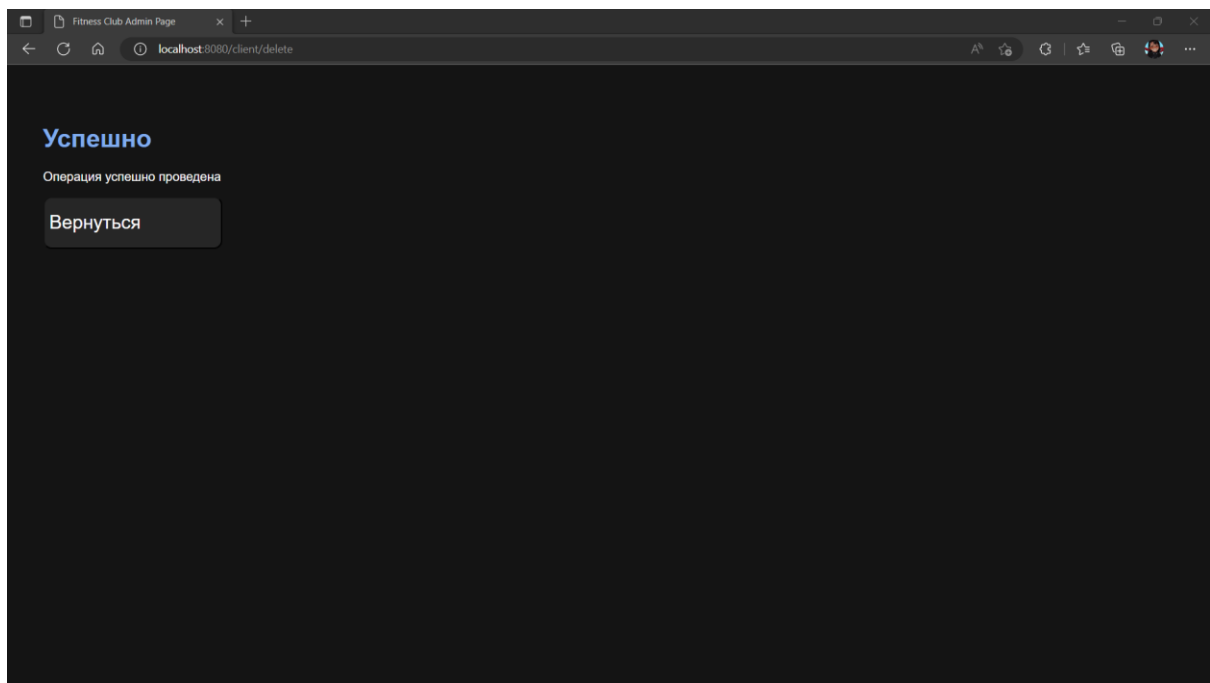


Рисунок 4 — результат операции



## 6. Список литературы

1. sqlx – Go Packages. [Электронный ресурс]  
URL: <https://pkg.go.dev/github.com/jmoiron/sqlx> (дата обращения 04.10.2022).
2. viper – Go Packages. [Электронный ресурс]  
URL: <https://pkg.go.dev/github.com/dvln/viper> (дата обращения 04.10.2022).
3. Model-view-controller. [Электронный ресурс]  
URL: <https://blog.skillfactory.ru/glossary/mvc/> (дата обращения 27.10.2022).
4. Docker Documentation. [Электронный ресурс]  
URL: <https://docs.docker.com/> (дата обращения 01.12.2022).