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

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных технологий

**Отчет по первой части итогового проекта**

по дисциплине «Технологии разработки программных приложений»

**Тема практической работы:** «База данных интернет-аптеки»

|  |  |
| --- | --- |
| **Выполнили:**  Студенты группыИКБО-24-20 | Сущев Р.Д. Скотников А.В. |
| **Проверила:** | Туманова М.Б. |

Москва 2022 г.

**Оглавление**

[**РТУ МИРЭА** 1](#_Toc98862065)

[Практическая часть 3](#_Toc98862066)

Таблица 1 - Карточка

|  |  |
| --- | --- |
| Заголовок | Интернет аптека |
| Заказчик (actor) | Аптека |
| Примечание | Сделать начальный этап разработки БД |
| **Цель** | Создать таблицы сущностей в БД |

# **Первая часть**

В своей работе мы создали базу данных интернет-аптеки. Которая хранит сведения о медикаментах, докторах, рецептах, пациентов и прочих составляющих аптеки.

Первой сущностью в нашей базе данных интернет-аптеки будет таблица пациентов. О пациентах будет храниться информация: имя, фамилия, возраст и гендер.

Второй сущностью будет таблица докторов, которые выписывают пациентам рецепты на медикаменты. О докторах должно быть известно: имя, фамилия, специальность.

Третьей создаётся отельная сущностью будет таблица с рецептами. Она включает в себя информацию о пациентах и докторах. Эта таблица также будет связана с таблицей пациентов и докторов.

Четвёртой сущностью будет таблица организаций, которая выпускает медикаменты. Об этих организациях должно быть известно: вебсайт, номер, адрес.

Пятой сущностью будет таблица медикаментов. Эта таблица будет связана с таблицей производителей. О медикаментах будет известно следующие: производитель, срок годности, название, цена и нужен ли на этот препарат рецепт.

На основании данных сущностей спроектируем ER-диаграмму Рисунок 1.

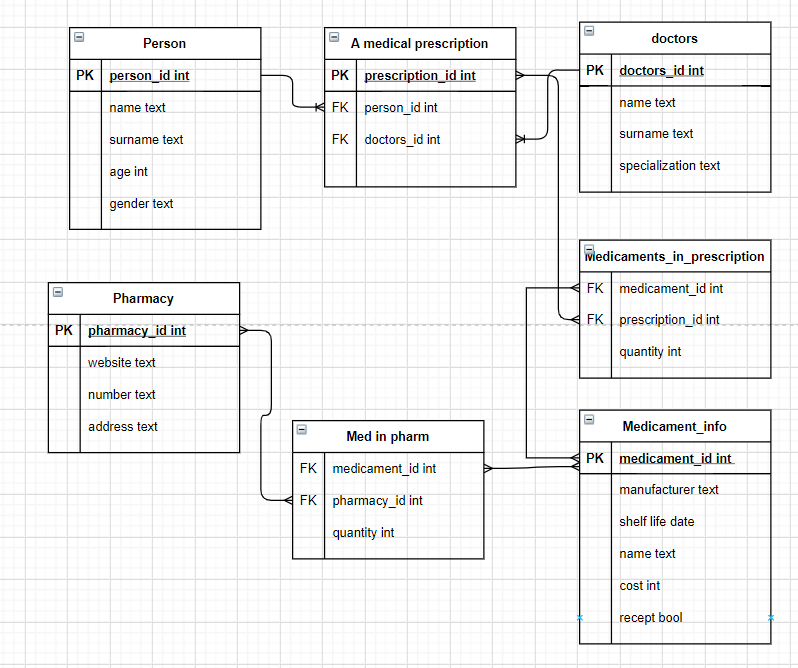


Рисунок 1- ER-диаграмма

Сущность пациентов связана с сущностью рецепт связью один-к-многим. Так как у одного пациента может быть много рецептов. Сущность доктора граждан и сущность рецептов связаны связью один-к-многим, так как один доктор может выписать несколько рецептов. Для реализации такой связи создаётся сущность «A medical prescription». Сущность организации связана с сущностью медикаментами связью многие-ко-многим, так как одна организация может выпустить много медикаментов, а один медикамент может быть произведен многими организациями. Для реализации такой связи создаётся сущность «Med in pharm». Сущность медикаменты связаны с сущностью рецепты связью многие-ко-многим, так как в одном рецепте может быть указанно много препаратов, а один медикамент может быть во многих рецептах. Для реализации такой связи создаётся сущность «Medicaments in prescription»

**Создание и настройка базы данных**

Создадим базу данных с помощью команды CREATE DATABASE. которую назовём в формате sushchev18 Лист 1.

|  |
| --- |
| CREATE DATABASE sushchev18; |

Лист 1

Создадим роль с помощью команды CREATE ROLE, которая будет использоваться для доступа и работы с создаваемой базой данных. Назовём ее в формате server18. Для простоты работы укажем атрибут LOGIN, который позволит выполнять вход от имени этой роли Лист 2.

|  |
| --- |
| CREATE ROLE server18 WITH LOGIN PASSWORD 'server18'; |

Лист 2

Далее выполним подключение к созданной базе данных от имени созданной роли с помощью мета-команды \c Лист 3.

|  |
| --- |
| \c sushchev18 server18; |

Лист 3

Далее для более удобного добавления записей в эти таблицы, следует создать последовательность – генератор чисел, который мы будем использовать лишь для того, чтобы он отсчитывал номера за нас, каждый раз выдавая новое число Лист 4.

|  |
| --- |
| CREATE SEQUENCE seq\_Person;  CREATE SEQUENCE seq\_Pharmacy;  CREATE SEQUENCE seq\_doctors;  CREATE SEQUENCE seq\_Medicament\_info; |

Лист 4

Далее необходимо создать таблицы, которые будут хранить данные в соответствии с задачей. После названия таблицы в скобках указываются имена атрибутов, их тип и прочие параметры атрибута. Уникальный ключ обозначается параметром PRIMARY KEY, а с помощью параметра DEFAULT и функции последовательности nextval укажем, что по умолчанию при добавлении новой строки в эту таблицу это поле будет равно следующему значению из созданной ранее последовательности. Остальные поля вводим в соответствии с диаграммой и типами данных, соответствующими задаче Лист 5.

|  |
| --- |
| CREATE TABLE Person  (  person\_id SERIAL PRIMARY KEY NOT NULL,  name TEXT,  surname TEXT,  age INT,  gender TEXT  );  CREATE TABLE Pharmacy  (  phrmacy\_id Serial PRIMARY KEY NOT NULL,  website TEXT default'yyyyy',  hotline TEXT default'xxxxx',  address TEXT default'zzzzz'  );  CREATE TABLE doctors  (  doctors\_id Serial PRIMARY KEY NOT NULL,  name TEXT,  surname TEXT,  specialization TEXT  );  CREATE TABLE Medicament\_info  (  medicament\_id Serial PRIMARY KEY NOT NULL,  manufacturer TEXT,  shelf\_life\_date TEXT,  name TEXT default'bbbbb',  cost INT default 55,  recept BOOLEAN  ); |

Лист 5

Также необходимо создать вспомогательную таблицу, реализующую связь «многие ко многим». Таблица будет содержать только внешние ключи на другие таблицы, что объявляется с помощью ключевого слова REFERENCES и указания таблицы, на которую этот внешний ключ будет указывать. Также добавим параметр ON DELETE RESTRICT, который будет требовать удалить строку, содержащую внешний ключ, ссылающийся на удаляемую строку, и параметр ON DELETE CASCADE, который сделает удаление за нас Лист 6.

|  |
| --- |
| CREATE TABLE A\_medical\_prescription  (  prescription\_id Serial PRIMARY KEY NOT NULL,  person\_id INT references person(person\_id) on delete restrict,  doctors\_id INT references doctors(doctors\_id) on delete restrict  );  CREATE TABLE Med\_in\_pharm  (  medicament\_id INT references Medicament\_info(medicament\_id) on delete restrict,  phrmacy\_id INT references Pharmacy(phrmacy\_id) on delete restrict,  quantity INT  );  CREATE TABLE medicament\_in\_prescription  (  medicament\_id INT references Medicament\_info(medicament\_id) on delete restrict,  prescription\_id INT references A\_medical\_prescription(prescription\_id) on delete cascade  ); |

Лист 6

Проверить создание сущностей можно с помощью мета-команды \d – эта команда выводит список таблиц, представлений и последовательностей Рисунок 2.

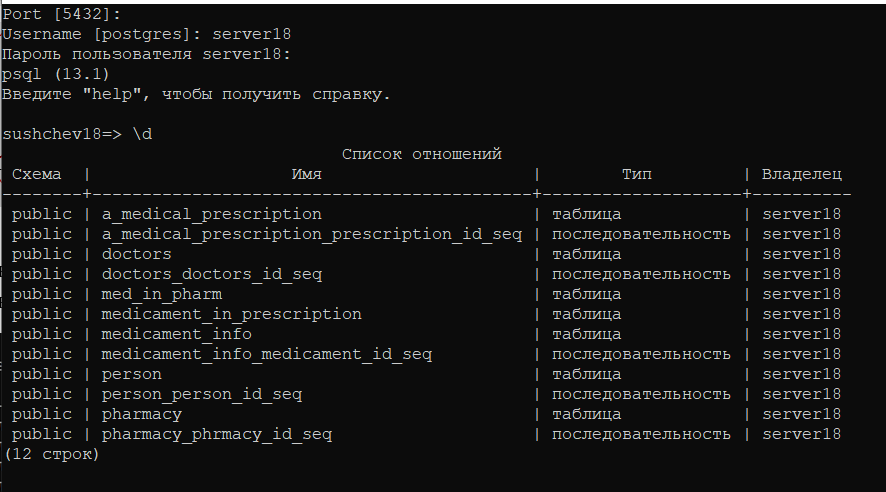


Рисунок 2 - Список таблиц, представлений и последовательностей

Проверить правильность создания конкретной таблицы можно с помощью той же мета-команды, в таком случае она выведет описание заданного элемента Рисунок 3-9.

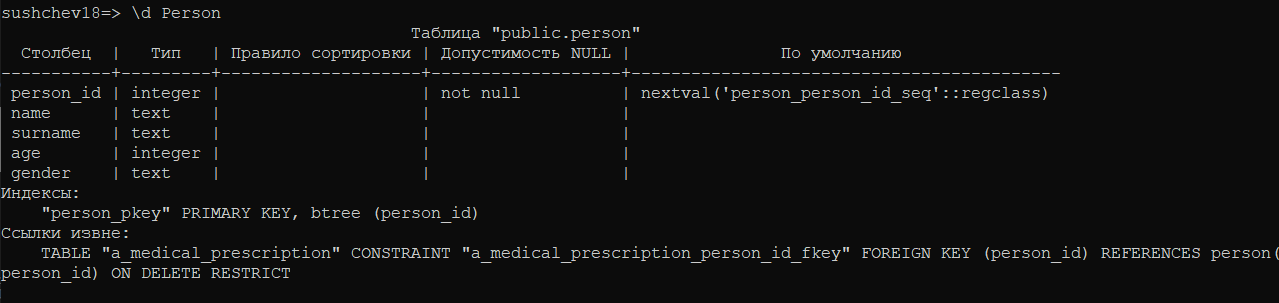


Рисунок 3 - Таблица пациентов

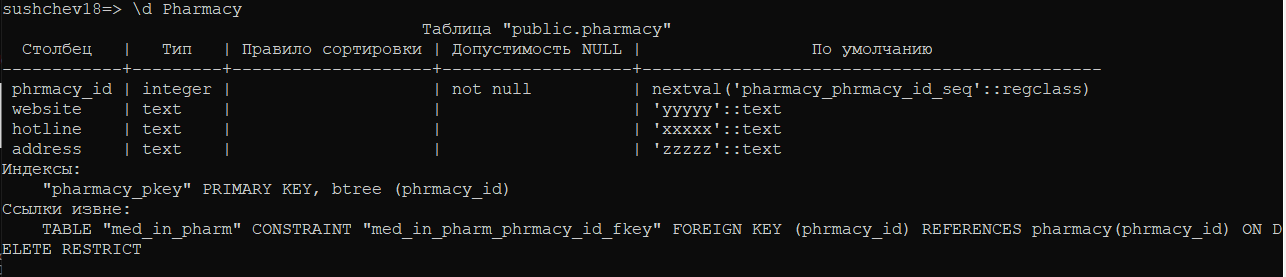


Рисунок 4 - Таблица производителей

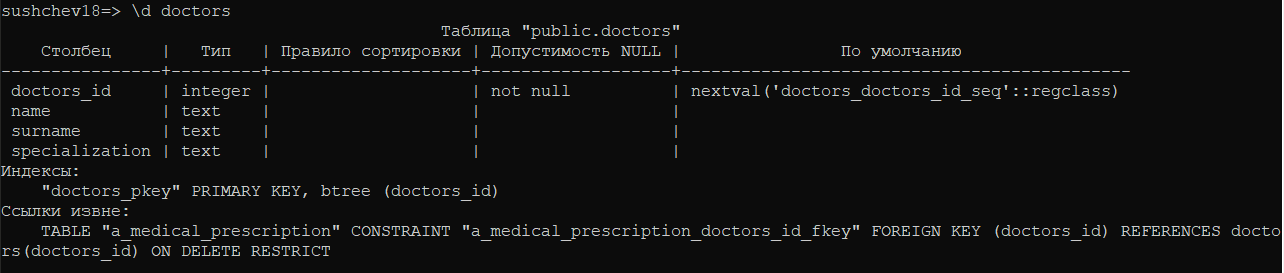


Рисунок 5 - Таблица докторов

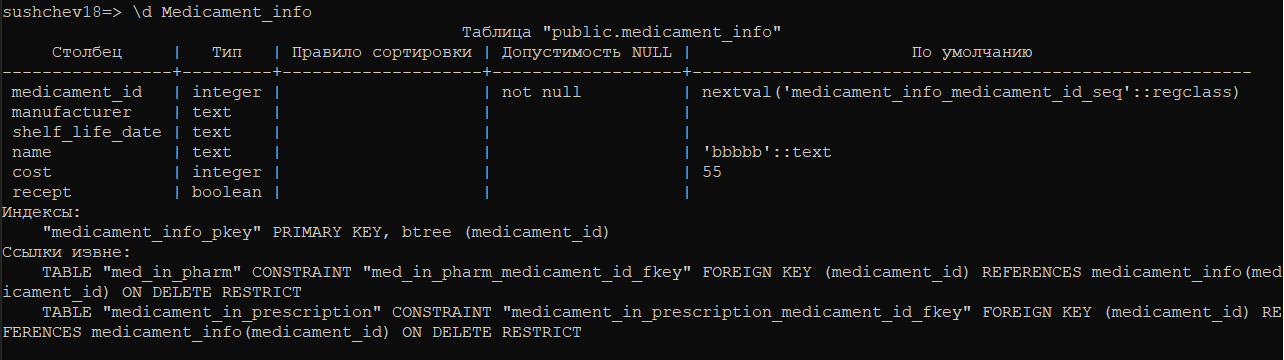


Рисунок 6 - Таблица медикаментов

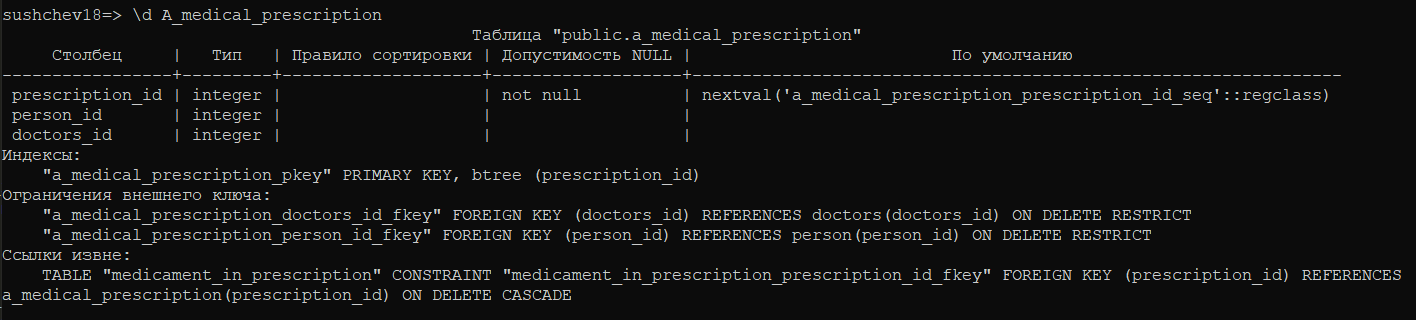


Рисунок 7 - Вспомогательная таблица для пациентов и докторов

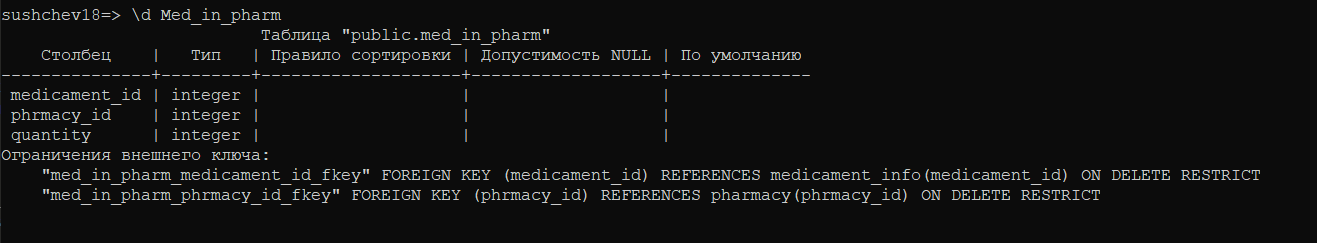


Рисунок 8 - Вспомогательная таблица для медикаментов и производителей

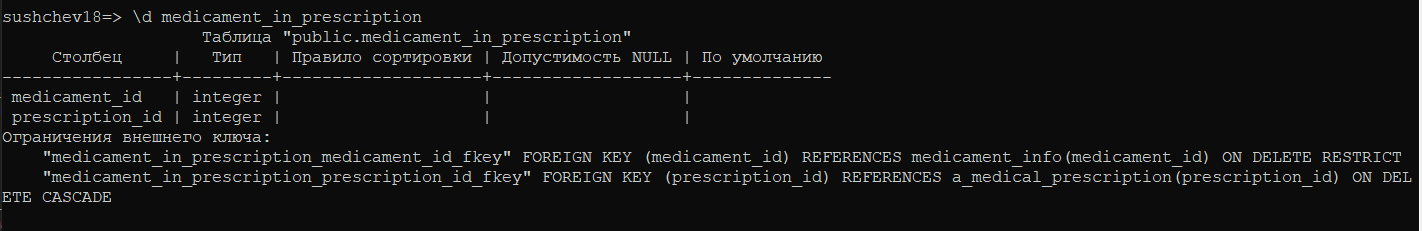


Рисунок 9 - Вспомогательная таблица для медикаментов и рецептов

**Вывод:**

В работе мы реализовали первую часть своего итогового проекта по теме «База данных интернет-аптеки». Мы сделали ER диаграмму и реализовали таблицы сущностей.

**Вторая Часть проекта**

Таблица 2 - Карточка

|  |  |
| --- | --- |
| Заголовок | Интернет аптека |
| Заказчик (actor) | Аптека |
| Примечание | Написать 4 файла: очистка базы данных – clear.sql; добавление данных – insert.sql; получение данных – select.sql; изменение данных – update.sql; |
| **Цель** | Написать начальные запросы для БД |

Теперь напишем текстовый файл, содержащий команды для создания сущностей, соответствующих нашей задаче, и выполним эти команды в psql. При возникновении ошибок, сущности можно удалить из базы данных с помощью команды DROP.

Прежде чем создавать файл для заполнения базы данных, стоит создать файл для ее очистки для быстрого исправления ошибок. Очистку таблицы можно осуществить с помощью простой sql-команды TRUNCATE. Также необходимо сбросить счетчик последовательностей, чтобы номера новых записей начинались с единицы, с помощью команды ALTER SEQUENCE.

Файл назовём clear.sql.

|  |
| --- |
| ALTER SEQUENCE a\_medical\_prescription\_prescription\_id\_seq RESTART WITH 1;  ALTER SEQUENCE doctors\_doctors\_id\_seq RESTART WITH 1;  ALTER SEQUENCE medicament\_info\_medicament\_id\_seq RESTART WITH 1;  ALTER SEQUENCE person\_person\_id\_seq RESTART WITH 1;  ALTER SEQUENCE pharmacy\_phrmacy\_id\_seq RESTART WITH 1;  TRUNCATE Person, Pharmacy, Medicament\_info, doctors, A\_medical\_prescription, Med\_in\_pharm, medicament\_in\_prescription; |

Лист 7

Вставка данных в таблицу производится sql-командой INSERT. Далее в скобках указывается имя таблицы, в которую будут вставлены данные, перечень параметров, а затем в скобках их значения в соответствующем порядке.

* Добавим 10 записей в таблицу пациентов;
* Добавим по 5 записей в таблицу, хранящие данные о производителях ;
* Добавим по 5 записей в таблицу, хранящие данные о докорах ;
* Добавим по 5 записей в таблицы хранящие связи.

Файл назовём insert.sql.

|  |
| --- |
| insert into person (name,surname,age,gender) values ('Andrey','Lobankov',18,'M');  insert into person (name,surname,age,gender) values ('Andrey','Skotnikov',30,'M');  insert into person (name,surname,age,gender) values ('Sergey','Cherniy',22,'M');  insert into person (name,surname,age,gender) values ('Pavel','Kovpack',12,'M');  insert into person (name,surname,age,gender) values ('Jo','Bidon',77,'M');  insert into person (name,surname,age,gender) values ('Donald','Trump',78,'M');  insert into person (name,surname,age,gender) values ('Vadim','Okhremenko',25,'M');  insert into person (name,surname,age,gender) values ('Timofey','Syshev',14,'M');  insert into person (name,surname,age,gender) values ('Stanislav','Vasilyev',30,'M');  insert into Pharmacy(website,hotline,address) values ('www.drio.ru','8-800-555-35-35','Lenina 34');  insert into Pharmacy(website,hotline,address) values ('www.micro.ru','8-803-355-45-35','Pushkino 31') ;  insert into Pharmacy(website,hotline,address) values ('www.dred.ru','8-840-554-65-35','ZFC 38') ;  insert into Pharmacy(website,hotline,address) values ('www.dyrin.ru','8-890-675-85-25','Shkolnaya 15') ;  insert into Pharmacy(website,hotline,address) values ('www.daran.ru','8-876-785-39-25','Arbat 101') ;  insert into doctors(name,surname,specialization) values ('Nikita','Pyetruk','Hirurg');  insert into doctors(name,surname,specialization) values ('Maxim','Kirillov','Travmotolog');  insert into doctors(name,surname,specialization) values ('Artem','Osokin','Okulist');  insert into doctors(name,surname,specialization) values ('Andrey','Naumenko','Proktolog');  insert into doctors(name,surname,specialization) values ('Vasiliy','Shabanov','urolog');  insert into Medicament\_info(manufacturer,shelf\_life\_date,name,cost,recept) values ('Biotek','14.04.2019','Analgin',345, true);  insert into Medicament\_info(manufacturer,shelf\_life\_date,name,cost,recept) values ('Garza','09.06.2018','Turinabol',35, true);  insert into Medicament\_info(manufacturer,shelf\_life\_date,name,cost,recept) values ('Inzavin','12.04.2020','Anavar',56, true);  insert into Medicament\_info(manufacturer,shelf\_life\_date,name,cost,recept) values ('FarmSib','6.03.2019','Stanozalon',500, true);  insert into Medicament\_info(manufacturer,shelf\_life\_date,name,cost,recept) values ('Pulse','16.10.2021','Vector',1000, true);  insert into A\_medical\_prescription(person\_id,doctors\_id) values (1,4);  insert into A\_medical\_prescription(person\_id,doctors\_id) values (2,1);  insert into A\_medical\_prescription(person\_id,doctors\_id) values (3,2);  insert into A\_medical\_prescription(person\_id,doctors\_id) values (4,5);  insert into A\_medical\_prescription(person\_id,doctors\_id) values (5,4);  insert into Med\_in\_pharm(medicament\_id,phrmacy\_id,quantity) values (1,1,7);  insert into Med\_in\_pharm(medicament\_id,phrmacy\_id,quantity) values (2,2,5);  insert into Med\_in\_pharm(medicament\_id,phrmacy\_id,quantity) values (3,3,6);  insert into Med\_in\_pharm(medicament\_id,phrmacy\_id,quantity) values (4,4,3);  insert into Med\_in\_pharm(medicament\_id,phrmacy\_id,quantity) values (5,5,1);  insert into medicament\_in\_prescription(medicament\_id,prescription\_id) values (5,1);  insert into medicament\_in\_prescription(medicament\_id,prescription\_id) values (2,3);  insert into medicament\_in\_prescription(medicament\_id,prescription\_id) values (3,2);  insert into medicament\_in\_prescription(medicament\_id,prescription\_id) values (1,5);  insert into medicament\_in\_prescription(medicament\_id,prescription\_id) values (2,3); |

Лист 8

Данные нужно не только хранить в базе данных, но и извлекать для чтения. Данные извлекаются из таблиц с помощью SQL-запроса SELECT. Если нужно извлечь все данные из таблицы, можно поставить символ \*, вместо того, чтобы перечислять все столбцы. Далее пунктом FROM указывается, откуда эти данные требуется извлечь (название таблицы), после чего указываются дополнительные пункты в зависимости от задачи. Например, пунктом ORDER BY указывается столбец, по которому полеченные строки будут отсортированы, а пунктом WHERE определяется условие, по которому полученные строки будут отфильтрованы.

Файл назовём select.sql.

|  |
| --- |
| SELECT \* FROM Pharmacy;  SELECT (gender,name, surname) FROM person ORDER BY age;  SELECT \* FROM medicament\_in\_prescription WHERE medicament\_id BETWEEN 1 AND 5 OR prescription\_id = 3; |

Лист 9

Данные, хранящиеся в таблицах, также можно изменять или удалять.

Изменение (обновление) данных производится с помощью SQL-запроса UPDATE. В команде указывается таблица, в которой требуется произвести изменения (обновления), затем пунктом SET определяются необходимые изменения, после чего пунктом WHERE указываются условия, определяющие какие записи требуется изменить.

Удаление записей производится с помощью SQL-запроса DELETE. Пунктом FROM указывается, в какой таблице требуется удалить записи. затем командой WHERE указываются условия, определяющие какие записи требуется удалить.

В своём задании я обновляю имена у пациентов и изменяю им возраст. Также удаляю доктора из имеющихся сущностей по id.

Файл назовём update.sql.

|  |
| --- |
| UPDATE Person SET name = 'Стас', age = 999 WHERE surname = 'Сталин';  UPDATE Person SET name = 'Стас', age = 19 WHERE surname = 'Ковпак';  UPDATE Person SET name = 'Стас', age = 23 WHERE surname = 'Охременко';  DELETE FROM A\_medical\_prescription WHERE doctors\_id = 4;  DELETE FROM doctors WHERE doctors\_id = 4;  DELETE FROM A\_medical\_prescription WHERE person\_id = 5;  DELETE FROM Person WHERE person\_id = 5; |

Лист 10

Были написаны 4 файла: очистка базы данных – clear.sql; добавление данных – insert.sql; получение данных – select.sql; изменение данных – update.sql;

Теперь необходимо выполнить написанные запросы. Для этого воспользуемся мета-командой psql \i или \include, которая считывает и исполняет весь файл.

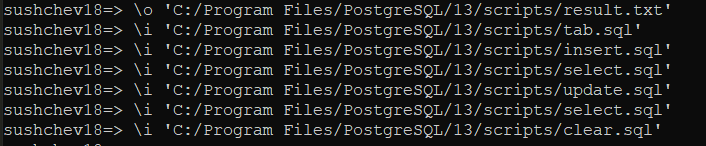


Рис. 10 Результат выполнения операций

Результат написанных запросов:

|  |
| --- |
| DROP OWNED  CREATE TABLE  CREATE TABLE  CREATE TABLE  CREATE TABLE  CREATE TABLE  CREATE TABLE  CREATE TABLE  INSERT 0 1  INSERT 0 1  INSERT 0 1  INSERT 0 1  INSERT 0 1  INSERT 0 1  INSERT 0 1  INSERT 0 1  INSERT 0 1  INSERT 0 1  INSERT 0 1  INSERT 0 1  INSERT 0 1  INSERT 0 1  INSERT 0 1  INSERT 0 1  INSERT 0 1  INSERT 0 1  INSERT 0 1  INSERT 0 1  INSERT 0 1  INSERT 0 1  INSERT 0 1  INSERT 0 1  INSERT 0 1  INSERT 0 1  INSERT 0 1  INSERT 0 1  INSERT 0 1  INSERT 0 1  INSERT 0 1  INSERT 0 1  INSERT 0 1  INSERT 0 1  INSERT 0 1  INSERT 0 1  INSERT 0 1  INSERT 0 1  INSERT 0 1  INSERT 0 1  phrmacy\_id | website | hotline | address  ------------+--------------+-----------------+-------------  1 | www.drio.ru | 8-800-555-35-35 | Ленина 34  2 | www.micro.ru | 8-803-355-45-35 | Пушкино 31  3 | www.dred.ru | 8-840-554-65-35 | ЗФС 38  4 | www.dyrin.ru | 8-890-675-85-25 | Школьная 15  5 | www.daran.ru | 8-876-785-39-25 | Арбат 101  (5 строк)  row  ------------------------  (МУЖ,Иосиф,Сталин)  (МУЖ,Павел,Ковпак)  (МУЖ,Тимофей,Сущев)  (МУЖ,Андрей,Лобанков)  (МУЖ,Сергей,Чёрный)  (МУЖ,Вадим,Охременко)  (МУЖ,Стас,Васильев)  (МУЖ,Андрей,Скотников)  (ЖЕН,Джо,Байден)  (МУЖ,Дональд,Трамп)  (10 строк)  medicament\_id | prescription\_id  ---------------+-----------------  5 | 1  2 | 3  3 | 2  1 | 5  2 | 3  (5 строк)  UPDATE 1  UPDATE 1  UPDATE 1  DELETE 2  DELETE 1  DELETE 0  DELETE 1  phrmacy\_id | website | hotline | address  ------------+--------------+-----------------+-------------  1 | www.drio.ru | 8-800-555-35-35 | Ленина 34  2 | www.micro.ru | 8-803-355-45-35 | Пушкино 31  3 | www.dred.ru | 8-840-554-65-35 | ЗФС 38  4 | www.dyrin.ru | 8-890-675-85-25 | Школьная 15  5 | www.daran.ru | 8-876-785-39-25 | Арбат 101  (5 строк)  row  ------------------------  (МУЖ,Тимофей,Сущев)  (МУЖ,Андрей,Лобанков)  (МУЖ,Стас,Ковпак)  (МУЖ,Сергей,Чёрный)  (МУЖ,Стас,Охременко)  (МУЖ,Стас,Васильев)  (МУЖ,Андрей,Скотников)  (МУЖ,Дональд,Трамп)  (МУЖ,Стас,Сталин)  (9 строк)  medicament\_id | prescription\_id  ---------------+-----------------  2 | 3  3 | 2  2 | 3  (3 строки)  ALTER SEQUENCE  ALTER SEQUENCE  ALTER SEQUENCE  ALTER SEQUENCE  ALTER SEQUENCE  TRUNCATE TABLE |

Лист 11

**Вывод:**

В работе мы реализовали вторую часть своего итогового проекта по теме «База данных интернет-аптеки». Были написаны 4 файла: очистка базы данных – clear.sql; добавление данных – insert.sql; получение данных – select.sql; изменение данных – update.sql;

**Третья Часть проекта**

Таблица 2 - Карточка

|  |  |
| --- | --- |
| Заголовок | Интернет аптека |
| Заказчик (actor) | Аптека |
| Примечание | Сгенерировать файлы веб-формы, настроить и запустить локальный веб-сервер Apache, запустить веб-форму |
| **Цель** | Создать веб-форму для управления БД |

В первую очередь настроим локальный сервер Apache. Для этого скачаем и распакуем архив с файлами сервера(Рисунок 11).

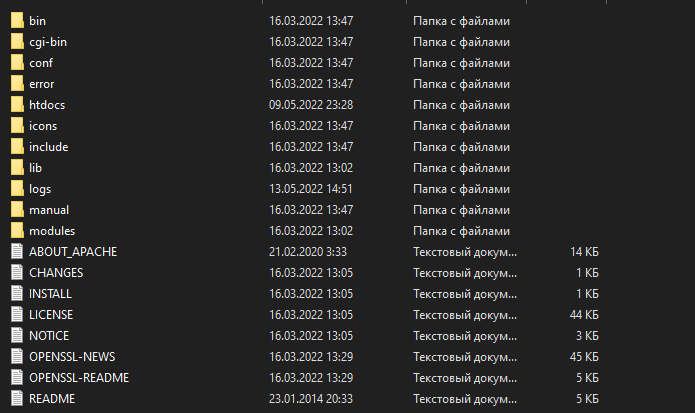


Рисунок 11 – Файлы сервера Apache

Теперь перейдём к настройке конфигурации сервера, для этого перейдём к файлу httpd.conf. В первую очередь укажем путь к папке Apache (Рисунок 12).

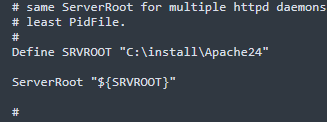


Рисунок 12 – Настройка конфигурации сервера Apache

Далее в строках ServerAdmin и ServerName укажем localhost тк именно с его помощью мы будем запускать наш сервер (Рисунок 13).

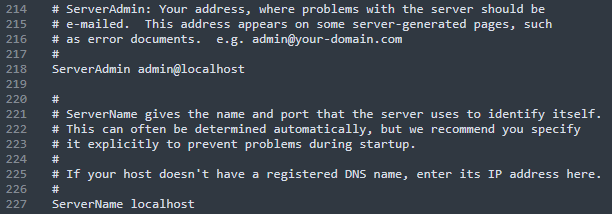


Рисунок 13 – Установка адреса сервера как localhost

Сохраняем изменения в файле и проверим их через командную в строку, которую откроем в папке bin нашего сервера (Рисунок 14).

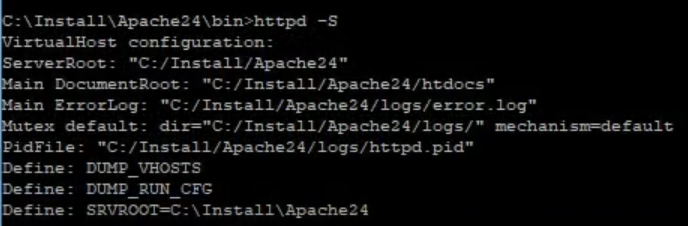


Рисунок 14 – Конфигурация сервера

Для упрощения взаимодействия с сервером установим его как службу, для этого воспользуемся той же командной строкой и выполним команду: httpd -k install (Рисунок 15).

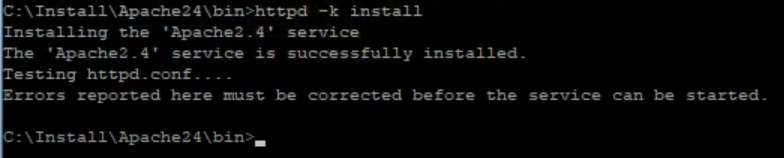


Рисунок 15 – Установка Apache как сервиса

Теперь запустим сервис Apache 2.4 и перейдём в браузере на страницу localhost, чтобы убедиться, что всё работает (Рисунок 16).

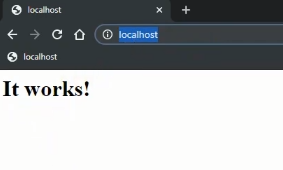


Рисунок 16 – Проверка работоспособности сервера

На странице отобразилость стандартное сообщение «It works!», которое лежит в файле в папке htdocs нашего сервера, именно в эту папку в дальнейшем мы будем загружать файлы веб-формы. Веб-форма будет работать на php, его поддержка не предусмотрена в Apache «из коробки», поэтому продолжим настройку. Скачаем и распакуем архив с файлами php 7.4.0 (Рисунок 17).

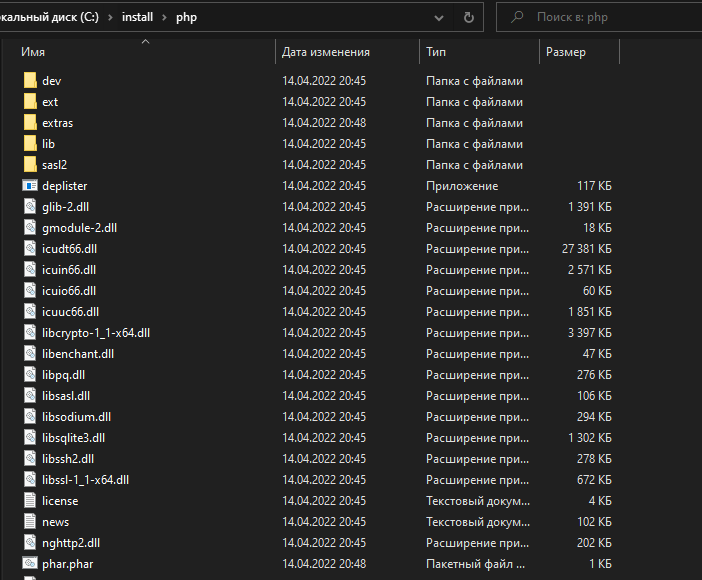


Рисунок 17 – Файлы php

В этой папке находим файл php.ini-development, создаём его копию и сохраняем как php.ini – это будет файлом конфигурации php. Чобы убедиться, что мы сделали всё верно проверим версию php через командную строку и выведем информацию о нем при помощи команды: php -i | more (Рисунок 18).

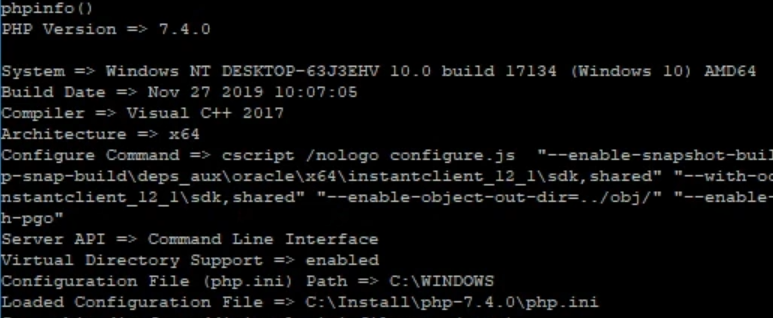


Рисунок 18 – Вывод информации о php

Видим, что установлена нужная нам версия 7.4.0 и что путь к загруженному файлу конфигурации установлен до ранее созданного php.ini.

Теперь снова перейдём к файлу httpd.conf и добавим в него строку для загрузки модуля работы с php и указываем путь к файлу php7apache2\_4.dll, который находится в папке php. Также добавим возможность работать с расширением .php и укажем директорию файла конфигурации php.ini (Рисунок 19).

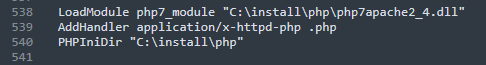


Рисунок 19 – Конфигурация сервера для работы с php

Последним шагом в настройке станет добавление поддержки работы с базами данных PostgreSQL. Для этого добавим файл с библиотекой php для pgsql в конфигурацию Apache (Рисунок 20).



Рисунок 20 – Конфигурация сервера для работы с PostgreSQL

В файле конфигурации php.ini необходимо раскомментировать строку, указывающую на папку ext с расширениями php для Windows, именно в ней лежат файлы, необходимые нам для работы с базой данных (Рисунок 21).

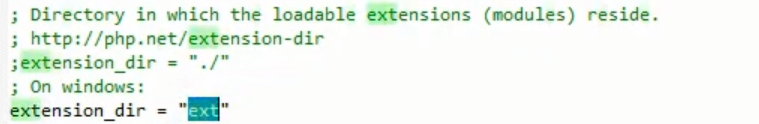


Рисунок 21 – Конфигурация php для расширений Windows

Также в блоке extensions необходимо раскомментировать строки с нужными нам расширениями pdo\_pgsql и pgsql (Рисунок 22).

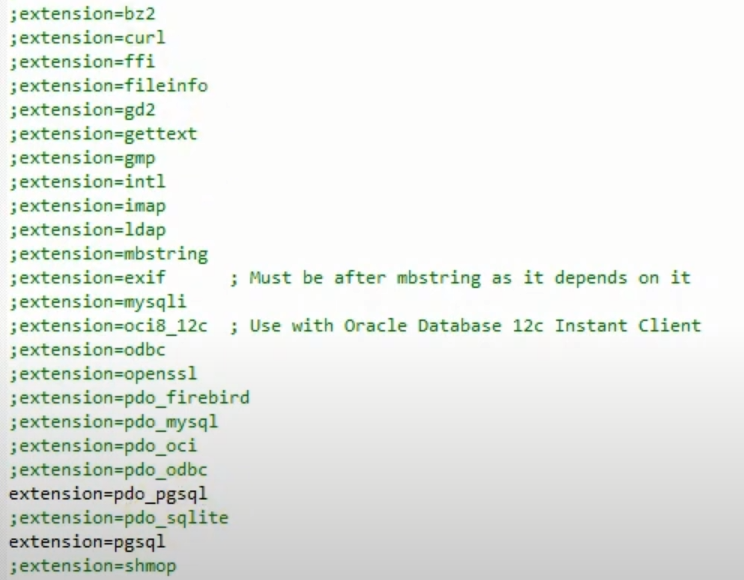


Рисунок 22 – Конфигурация php для работы с PostgreSQL

На этом настройка завершена, теперь можно перейти к созданию веб-формы. Для этого была использована программа PostgreSQL PHP Generator Lite. Запускаем её и вносим данные нашей БД (Рисунок 23).

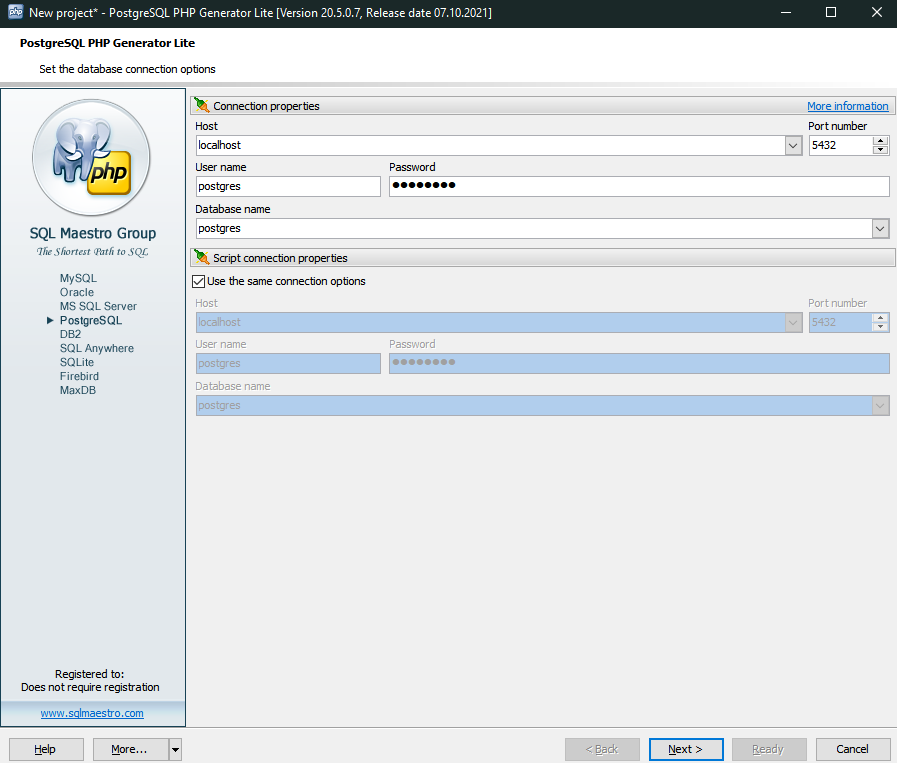


Рисунок 23 – Ввод данных БД

Далее добавляем все таблицы из нашей БД (Рисунок 24).

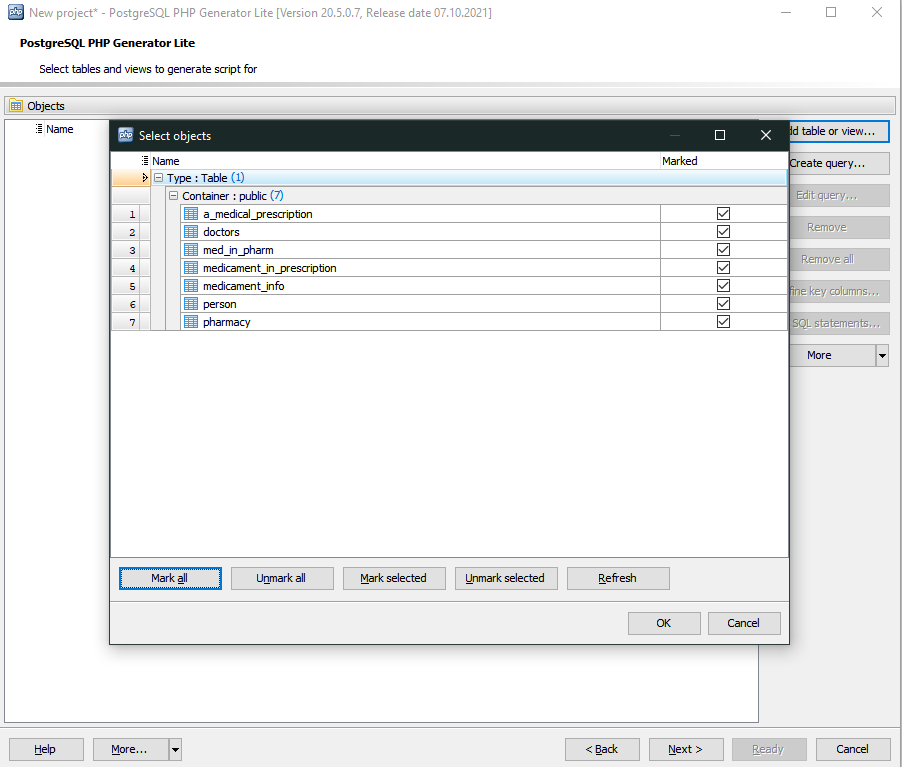


Рисунок 24 – Добавление таблиц БД

Далее указываем папку для выгрузки файлов htdocs, настраиваем пользователей и уровни доступа для них и генерируем веб-форму(Рисунок 25).

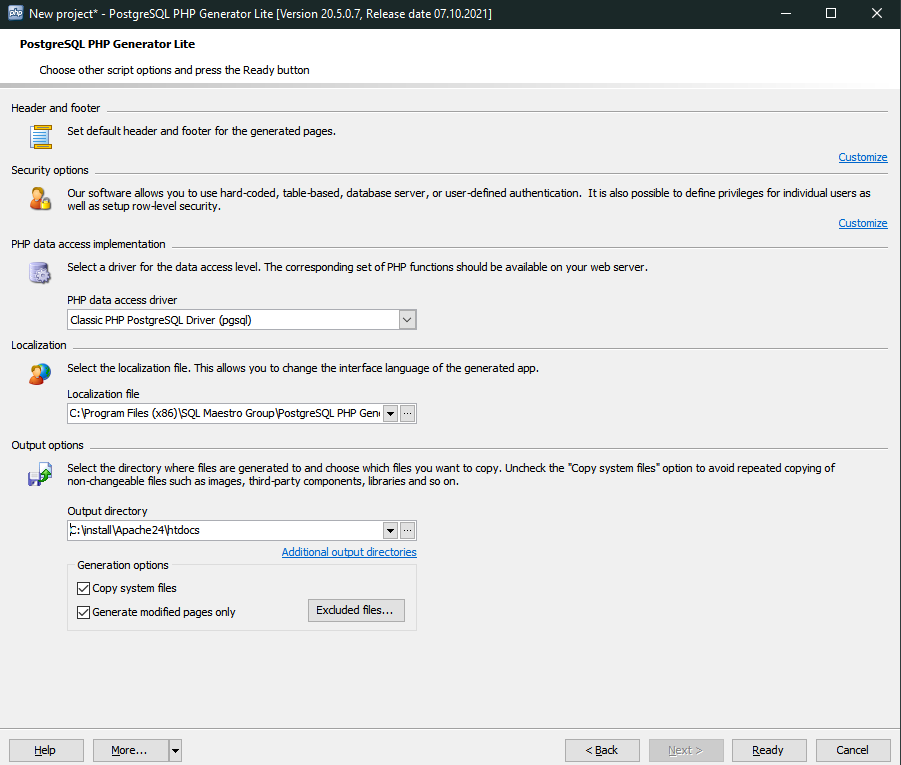


Рисунок 25 – Генерирование веб-формы

**Вывод:**

В работе мы реализовали третью часть своего итогового проекта по теме «База данных интернет-аптеки». Были настроены конфигурации Apache и php, а также сгенерирована веб-форма.

**Четвёртая Часть проекта**

Таблица 2 - Карточка

|  |  |
| --- | --- |
| Заголовок | Интернет аптека |
| Заказчик (actor) | Аптека |
| Примечание | Демонстрация и тест веб-формы |
| **Цель** | Протестировать работоспособность веб-формы |

После настройки и генерации веб-формы перезапустим сервис Apache и перейдём к тестированию. Для этого в браузере открываем страницу localhost. Открывается страница для авторизации (Рисунок 26).

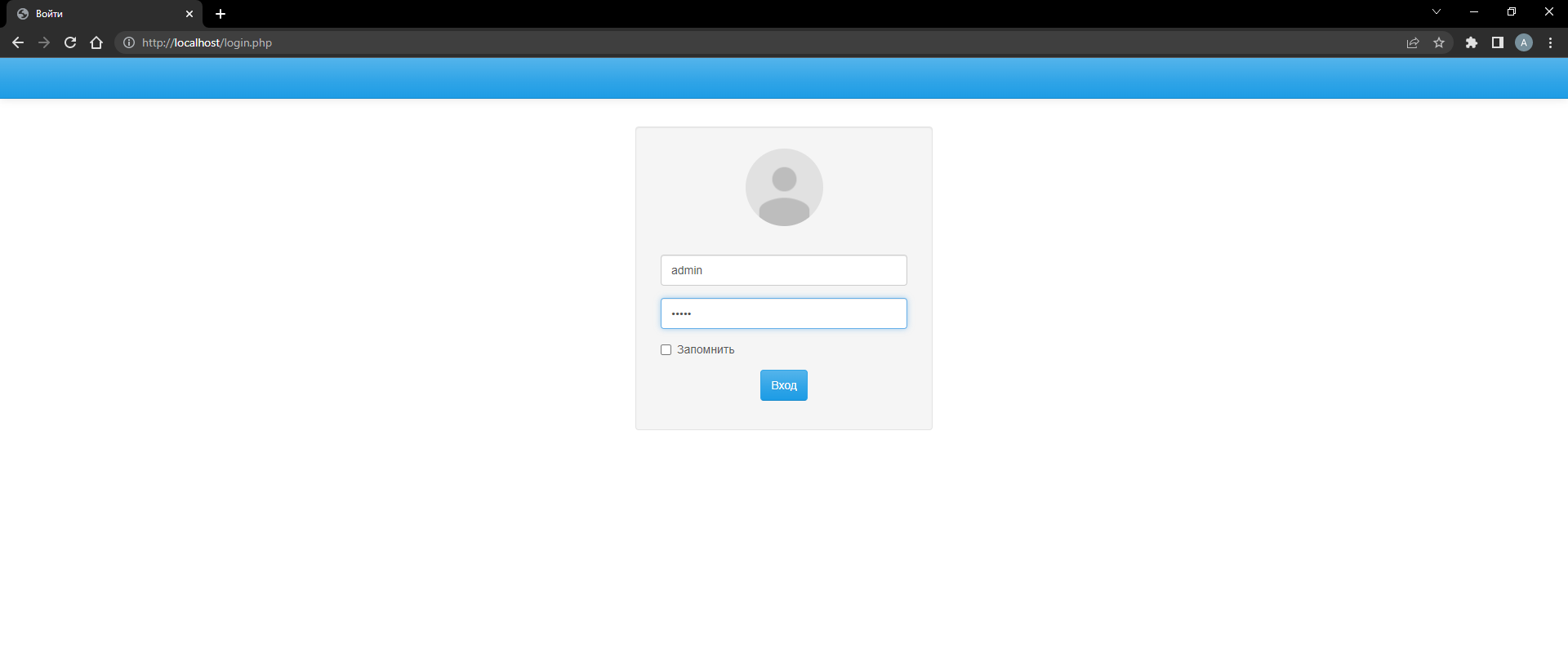


Рисунок 26 – Страница авторизации

Для работы с нашей базой данных мы сделали двух пользователей:

1) admin – пользователь с правами администратора, который имеет полный доступ ко всем аспектам работы с базой данных.

2) user – пользователь, который может только видеть данные в таблицах без возможности редактирования и удаления.

Введём данные пользователя admin и выполним вход. Открывается страница с таблицами нашей базы данных, записями, которые они содержат, а также со всеми возможными кнопками для редактирования БД (Рисунок 27).

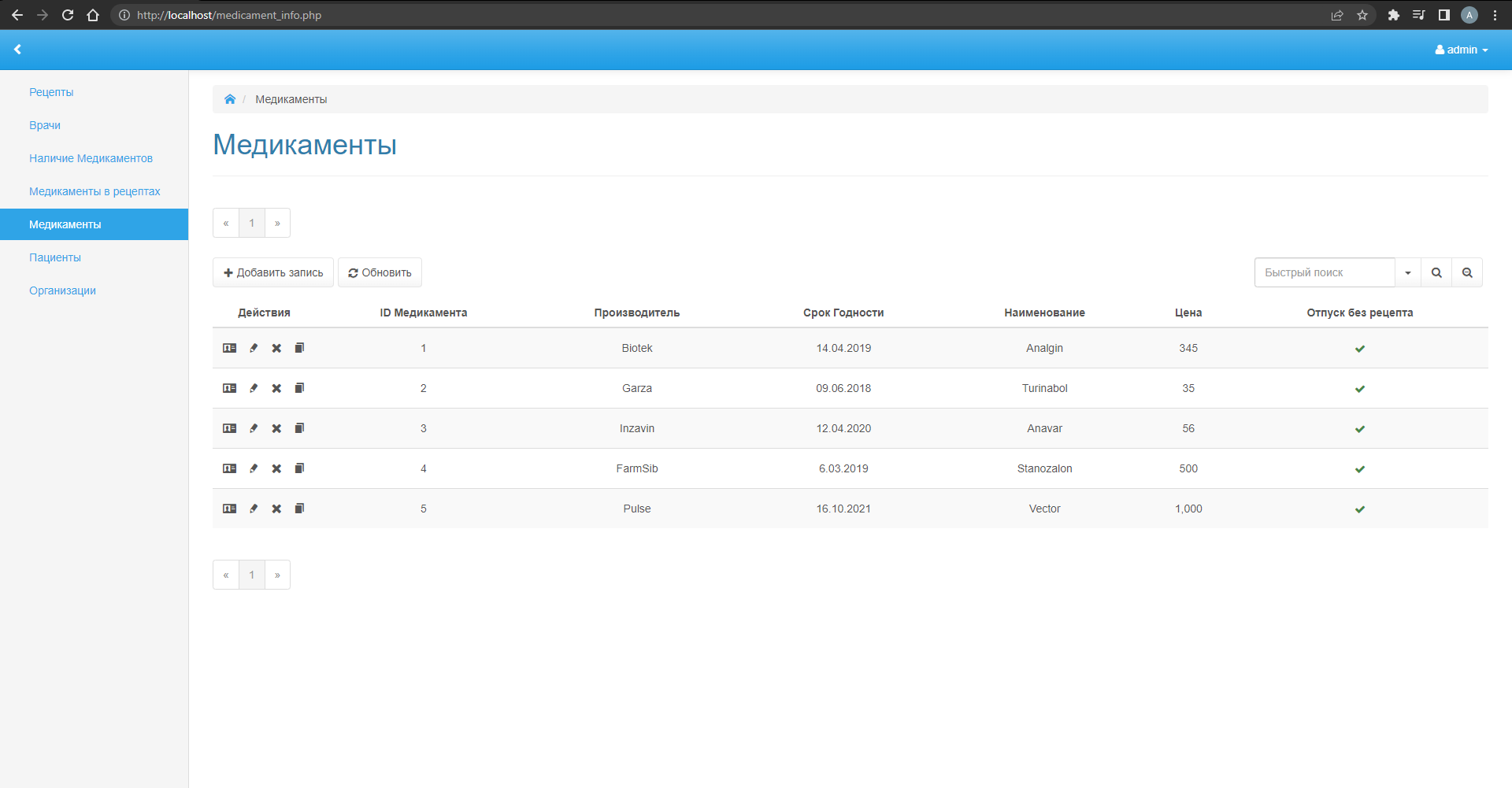


Рисунок 27 – Отображение таблиц в веб-форме

Наша веб-форма позволяет просматривать данные в таблицах, редактировать удалять, добавлять новые записи, а также имеются фильтры и поиск для удобного нахождения нужных данных в таблице.

**Редактирование**

При нажатии на значок карандаша в любой из строк таблицы можно перейти на страницу редактирования этой строки (Рисунок 28). На странице также имеются кнопка отмены редактирования и кнопка сохранения, имеющая всплывающее меню с дополнительными действиями при сохранении (Рисунок 29).

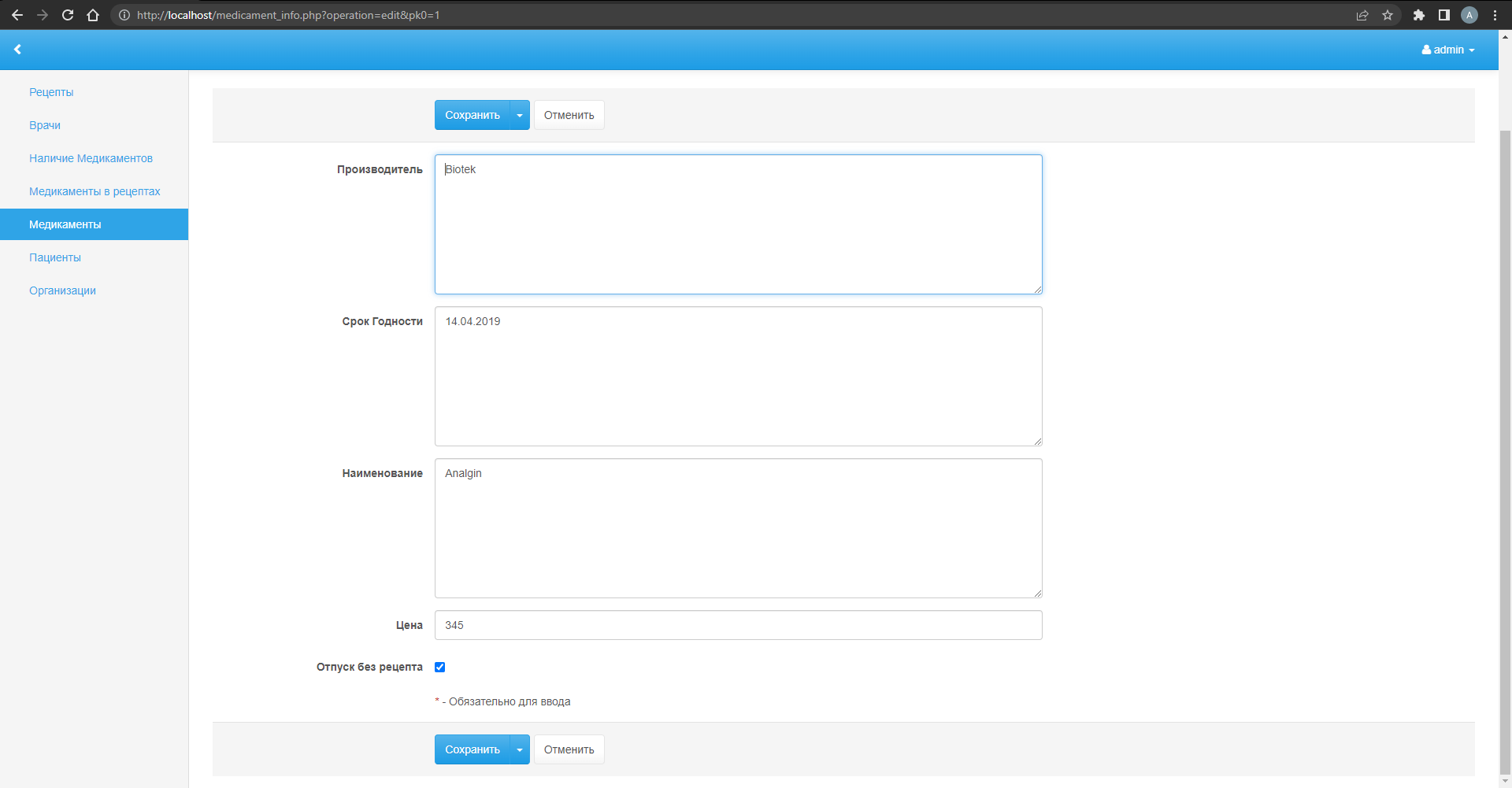


Рисунок 28 – Редактирование данных в таблице

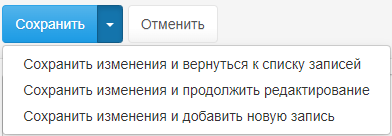


Рисунок 29 – Дополнительный функционал кнопки

**Удаление**

При нажатии на значок крестика в любой из строк таблицы откроется всплывающее окно для подтверждения удаления записи из таблицы (Рисунок 30). Если запись не связана с другими таблицами, то после подтверждения она удалится, если же она используется в других таблицах, то появится сообщение об ошибке (рисунок 31).

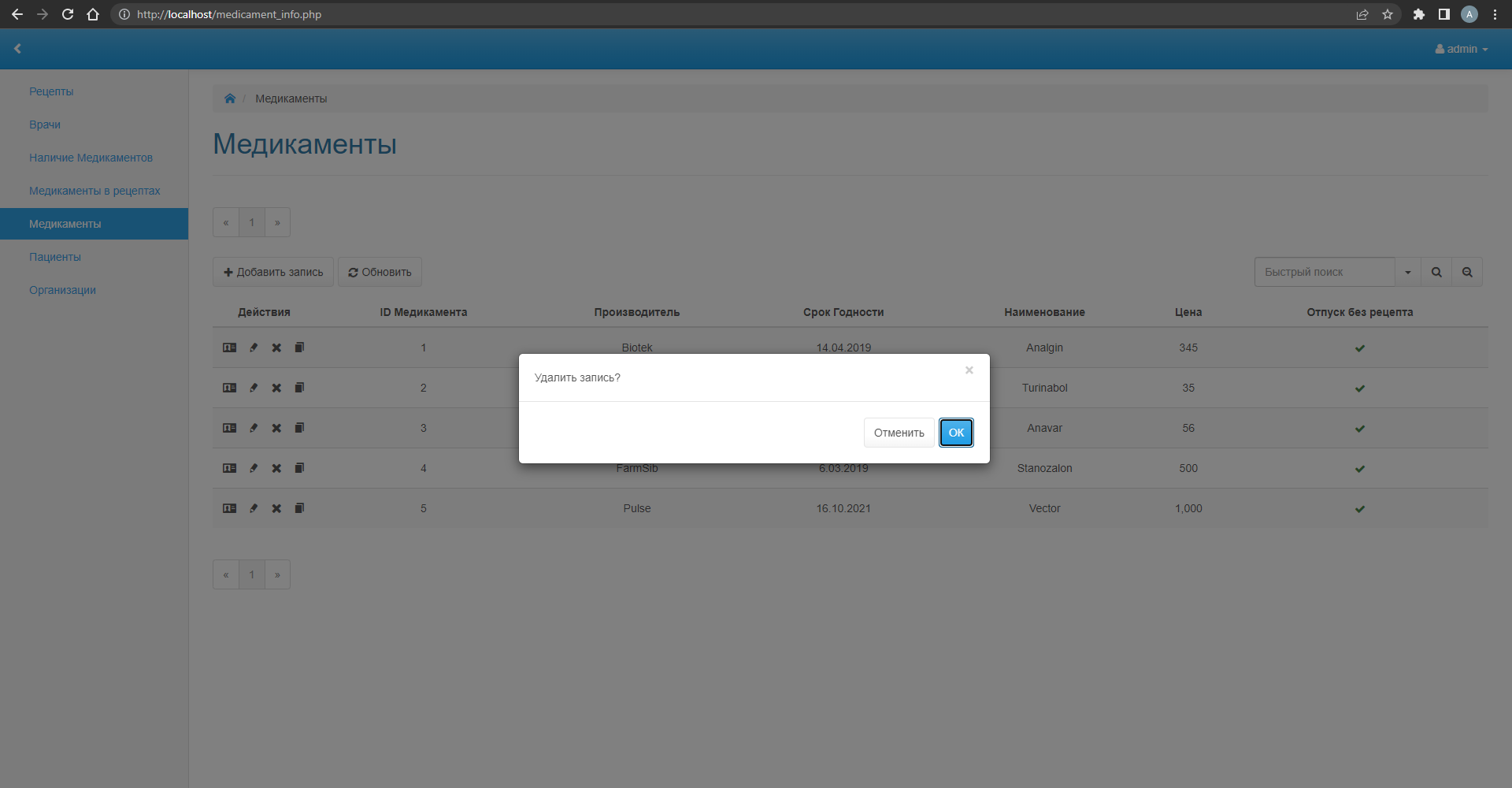


Рисунок 30 – Всплывающее окно подтверждения удаления

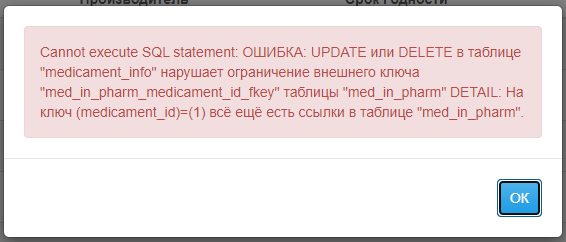


Рисунок 31 – Сообщение об ошибке при удалении т.к. на запись есть ссылка в другой таблице

**Добавление новых записей**

На странице у каждой таблицы реализована кнопка «+ Добавить запись». При нажатии на неё открывается страница с полями для заполнения, соответствующими каждому столбику таблицы (Рисунок 32). В остальном страница идентична странице редактирования записей.

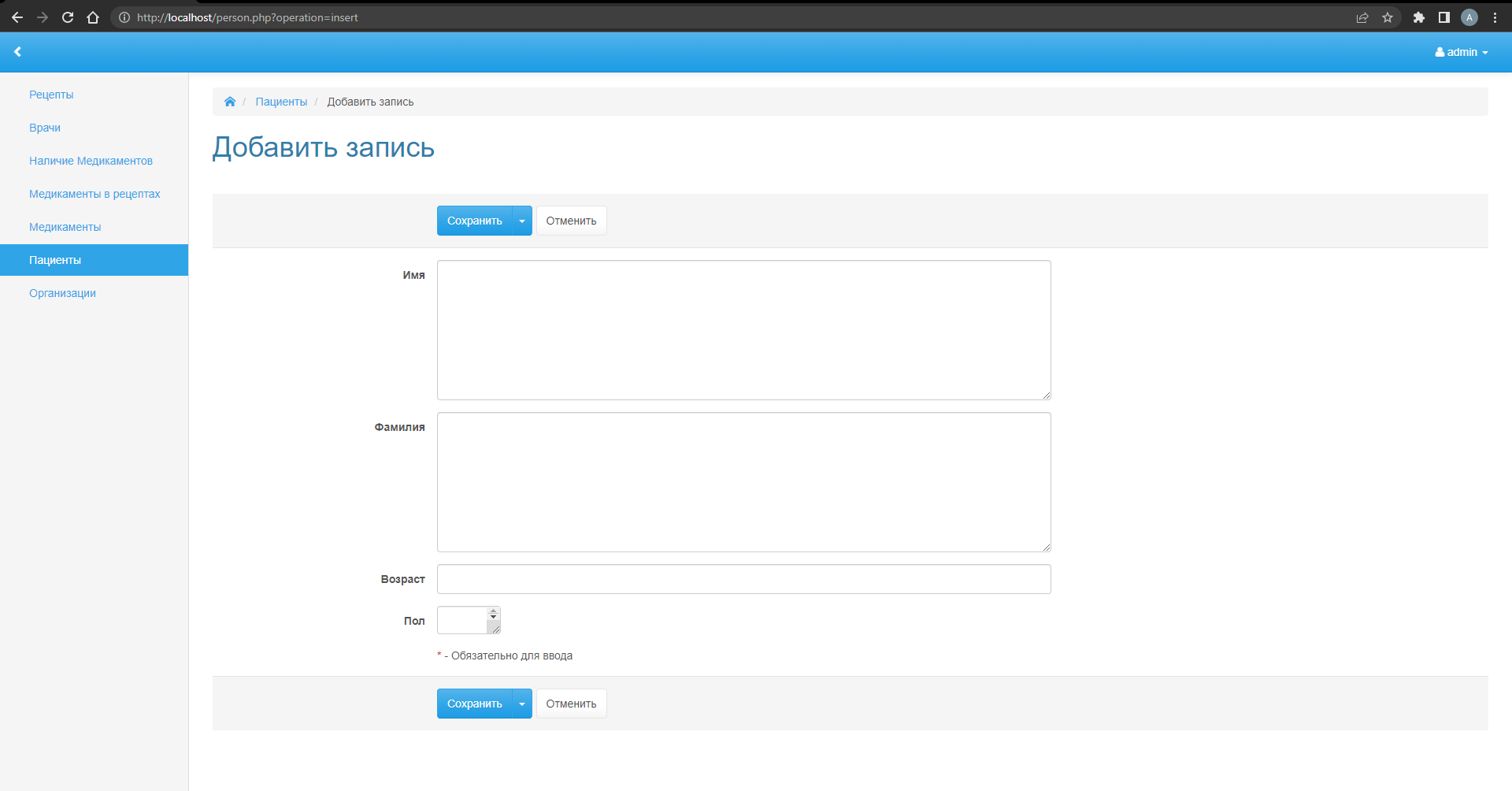


Рисунок 32 – Страница добавления новой записи

**Фильтрация и поиск данных**

На странице каждой таблицы реализован блок для фильтрации и поиска данных в таблице (Рисунок 33), он состоит из поля для ввода (быстрый поиск), выкидного меню с фильтрами для поиска (Рисунок 34), кнопки поиска и кнопки очистки поискового запроса. Для примера сделаем простой запрос (Рисунок 35).

Помимо этого блока можно сделать сортировку данных по алфавиту или по значению в определённом столбике, для этого достаточно нажать на название столбика, по которому вы хотите отсортировать записи, например, по возрасту (рисунок 36).

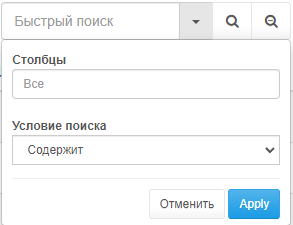


Рисунок 33 – Блок для фильтрации и поиска данных

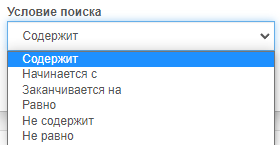


Рисунок 34 – Фильтры для поиска

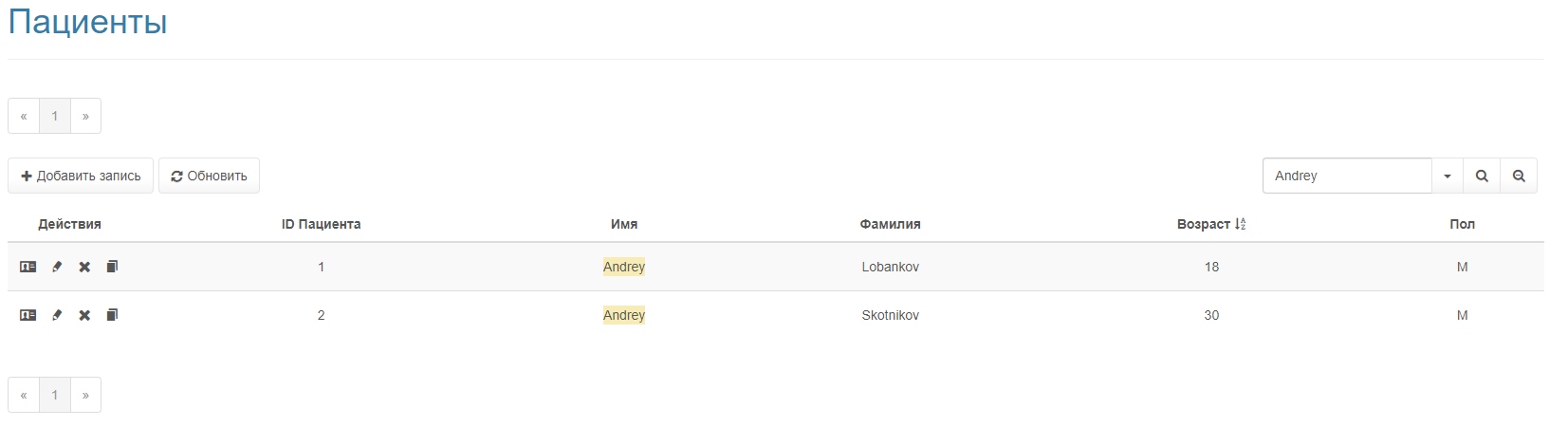


Рисунок 35 – Пример поискового запроса в таблице

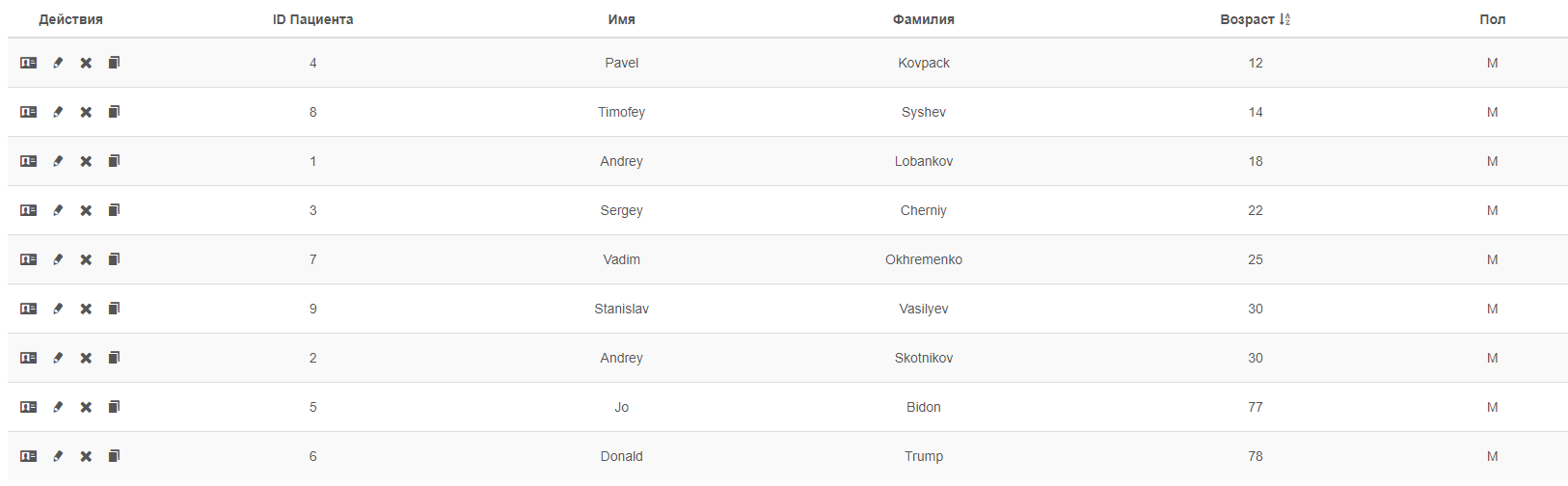


Рисунок 36 – Сортировка записей по возрастанию значения в столбике «Возраст»

Также продемонстрируем работоспособность пользователя user (Рисунок 37), как мы видим, при входе в базу данных, отсутствуют кнопки для редактирования, удаления и добавления записей, потому что у пользователя есть права только на просмотр записей.

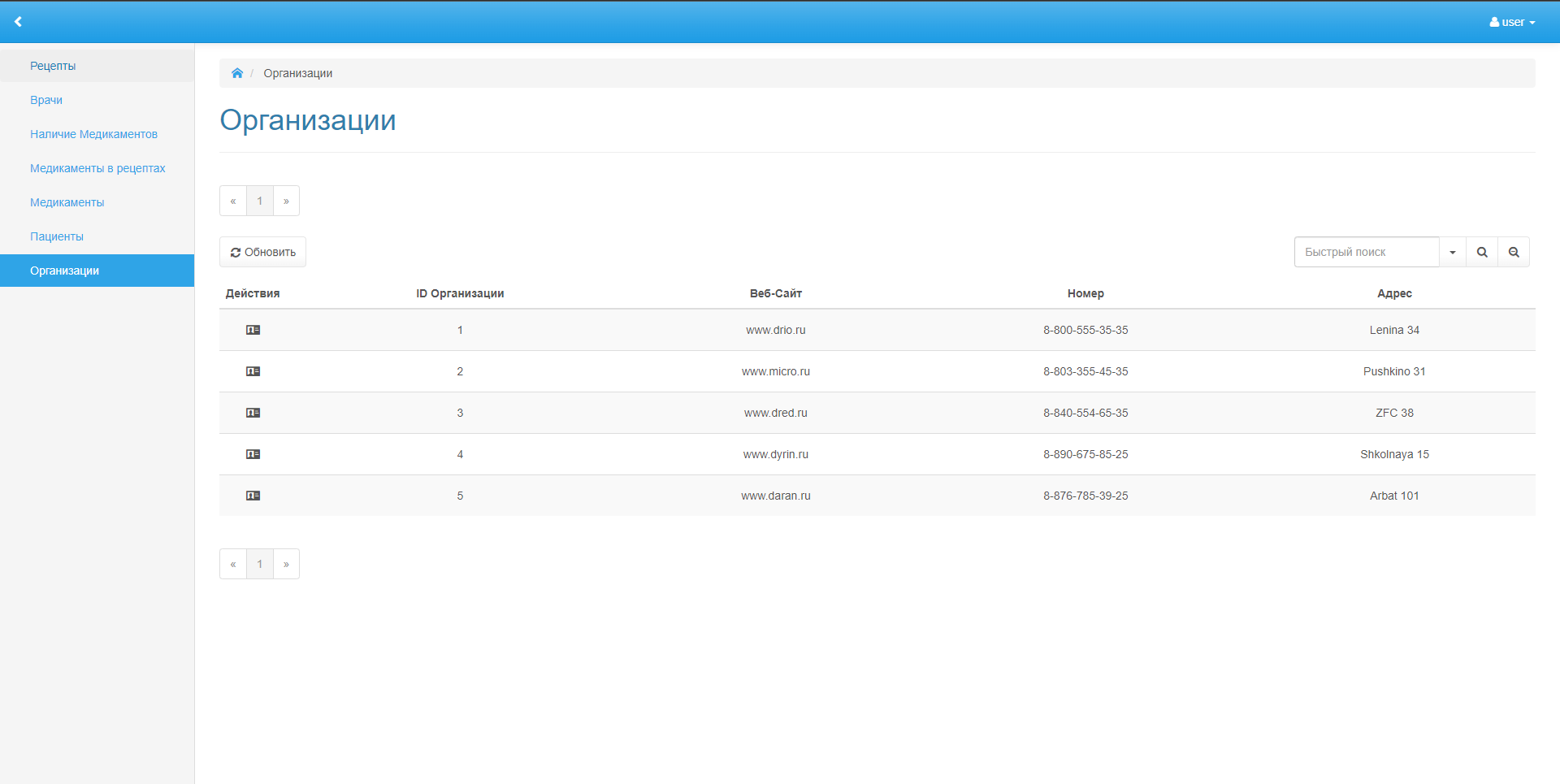


Рисунок 37 – Вид веб-формы при входе без прав администратора

**Вывод:**

В работе мы реализовали четвёртую часть своего итогового проекта по теме «База данных интернет-аптеки». Были продемонстрированы все созданные страницы, а также описан и продемонстрирован их функционал для пользователей с разными правами доступа.