

Министерство науки и высшего образования Российской Федерации
Муромский институт (филиал)
федерального государственного бюджетного образовательного учреждения высшего образования
«Владимирский государственный университет
Имени Александра Григорьевича и Николая Григорьевича Столетовых»
(МИВлГУ)

Факультет _____ ИТР _____

Кафедра _____ ПИИ _____

КУРСОВАЯ РАБОТА

По _____ Разработке корпоративных приложений _____

Тема _____ Веб-приложение «Ветеринарная клиника» _____

Руководитель

Кульков Я.Ю.
(фамилия, инициалы)

(подпись) (дата)

Студент ПИИ-121
(группа)

Лямина М.А.
(фамилия, инициалы)

(подпись) (дата)

Муром 2025

задание

В данной курсовой работе представлено Веб-приложение «Ветеринарная клиника». Приложение разработано на языке программирования Java в среде разработки IntelliJ IDEA с использованием фреймворка Spring. На заключительном этапе работы произведено тестирование разработанного продукта.

In this coursework, the web application "Veterinary Clinic" is presented. The application is developed using the Java programming language in the IntelliJ IDEA development environment with the Spring framework. In the final stage of the project, the developed product was tested.

Содержание

Введение	6
1 Анализ технического задания.....	7
2 Разработка моделей данных	9
3 Проектирование работы системы.....	11
4 Разработка и реализация системы.....	14
5 Тестирование системы	17
Заключение.....	21
Список используемой литературы.....	22
Приложение 1 Текст программы	23
Приложение 2 Снимки окон программы (скриншоты программы).....	24

					МИВУ 09.03.04-17.000 ПЗ		
Изм.	Лист	№ докум.	Подпись	Дата			
Разраб.		Лямина М.А.			Веб-приложение «Ветеринарная клиника»	Лит.	Лист
Провер.		Кульков Я.Ю.					Листов
Реценз.							5
Н. Контр.						МИ ВлГУ ПИН-121	
Утверд.							
							35

Введение

Эффективное управление ветеринарной клиникой в современных условиях невозможно без внедрения и использования информационных технологий. Внедрение распределённой информационной системы позволяет автоматизировать ключевые процессы, такие как запись на приём, ведение электронной медицинской карты пациентов, контроль за запасами медикаментов и оборудование, а также управление работой персонала.

Распределённая система – это система, в которой функции и ресурсы распределены между множеством узлов, а отсутствие единого управляющего центра обеспечивает её устойчивость и непрерывность работы. Применение такого подхода в разработке веб-приложения «Ветеринарная клиника» позволяет создать отказоустойчивую систему, обеспечивающую доступ пользователей к необходимым функциям в режиме реального времени.

Автоматизация работы ветеринарной клиники значительно повышает её эффективность, снижает вероятность ошибок при ведении документации и ускоряет процесс оказания медицинских услуг.

Целью курсовой работы является проектирование и разработка веб-приложения для управления деятельностью ветеринарной клиники. Для достижения поставленной цели необходимо выполнить следующие задачи:

- проанализировать предметную область;
- разработать модель данных;
- создать базу данных;
- разработать приложение;
- протестировать программный продукт.

1 Анализ технического задания

Функциональные возможности и требования:

Веб-приложение предназначено для удобного взаимодействия сотрудников ветеринарной клиники с внутренними процессами управления. Оно упрощает доступ к информации о клиентах и их питомцах, помогает организовать рабочий процесс врачей и персонала клиники, а также автоматизирует рутинные задачи, связанные с регистрацией приемов, ведением медицинских карт и формированием отчетов.

Приложение предоставляет следующие функциональные возможности:

- Регистрация и управление учетными записями сотрудников клиники: добавление, редактирование и удаление сотрудников.
- Назначение ролей (администратор и оператор).
- Управление информацией о врачах: ФИО, должность, контактная информация и другие характеристики.
- Ведение расписания работы врачей с возможностью редактирования.
- Управление услугами клиники.
- Регистрация клиентов и их питомцев: добавление данных о клиентах клиники (ФИО, контактные данные) и об их питомцах (кличка, порода, вид животного и другие характеристики), управление этими данными.
- Ведение медицинских карт питомцев: хранение истории приемов питомца с возможностью просмотра подробной информации о каждом приеме.
- Управление приемами: запись питомца на прием, отмена записи на прием, добавление приема с указанием такой информации, как дата, причина, диагноз, рекомендации врача, оказанные услуги и др, и с возможностью прикрепления файлов к приему, редактирование и удаление приема.
- Генерация сводных отчетов по различным критериям, сохранение отчетов в виде файлов.

Требования к системе:

					МИВУ 09.03.04-17.000 ПЗ	Лист
						7
Изм.	Лист	№ докум.	Подпись	Дата		

– Авторизация пользователей: доступ к системе только для зарегистрированных пользователей, использование аутентификации с помощью логина и пароля.

– Рольное разделение: администратор (доступ ко всем функциям) и оператор (ограниченные возможности).

– Адаптивный интерфейс.

– Валидация данных на стороне клиента (на предмет корректности формата данных) и на стороне сервера (на предмет соответствия данных логике работы).

– Формирование файловых отчетов.

					МИВУ 09.03.04-17.000 ПЗ	Лист
						8
Изм.	Лист	№ докум.	Подпись	Дата		

2 Разработка моделей данных

При проектировании модели данных учитывались требования предметной области и функциональные особенности системы. Для приложения были выделены основные сущности и взаимосвязи между ними.

Описание таблиц и связей:

Таблица users: хранит информацию о пользователях (ID, ФИО, дата рождения, контактные данные, логин и пароль). Связана с таблицей clients (один к одному) и с таблицей employees (один к одному).

Таблица files: содержит информацию о файлах (ID, путь). Связана с таблицами employees и animals (по одному файлу для сотрудника и животного), а также с таблицей appointments (многие ко многим).

Таблица positions: хранит данные о должностях сотрудников (ID, название). Связана с таблицей employees (один ко многим).

Таблица employees: содержит информацию о сотрудниках (ID, описание, связи с файлами (главное изображение), должность и пользователь). Связана с таблицами positions (многие к одному), users (один к одному), schedules

Таблица clients: содержит данные о клиентах (ID). Связана с таблицами animals (многие к одному), users (один к одному), с файлами (главное изображение).

Таблица animals: хранит данные о животных (ID, имя, дата рождения, пол, порода). Связана с таблицами animal_types (многие к одному), clients (многие к одному), files (один к одному), schedules (многие ко многим) и appointments (многие ко многим).

Таблица animal_types: содержит виды животных (ID, название). Связана с таблицей animals (один ко многим).

Таблица appointments: хранит информацию о визитах животных к врачу (ID, дата, причина, диагноз, рекомендация). Связана с таблицами animals

(многие к одному), employees (многие ко многим), и procedures (многие ко многим).

Таблица procedures: содержит данные об услугах (ID, название, цена). Связана с таблицей appointments (многие ко многим).

Таблица schedules: хранит данные о расписаниях сотрудников и записях на прием (ID, дата). Связана с таблицами employees (многие к одному) и animals (многие ко многим).

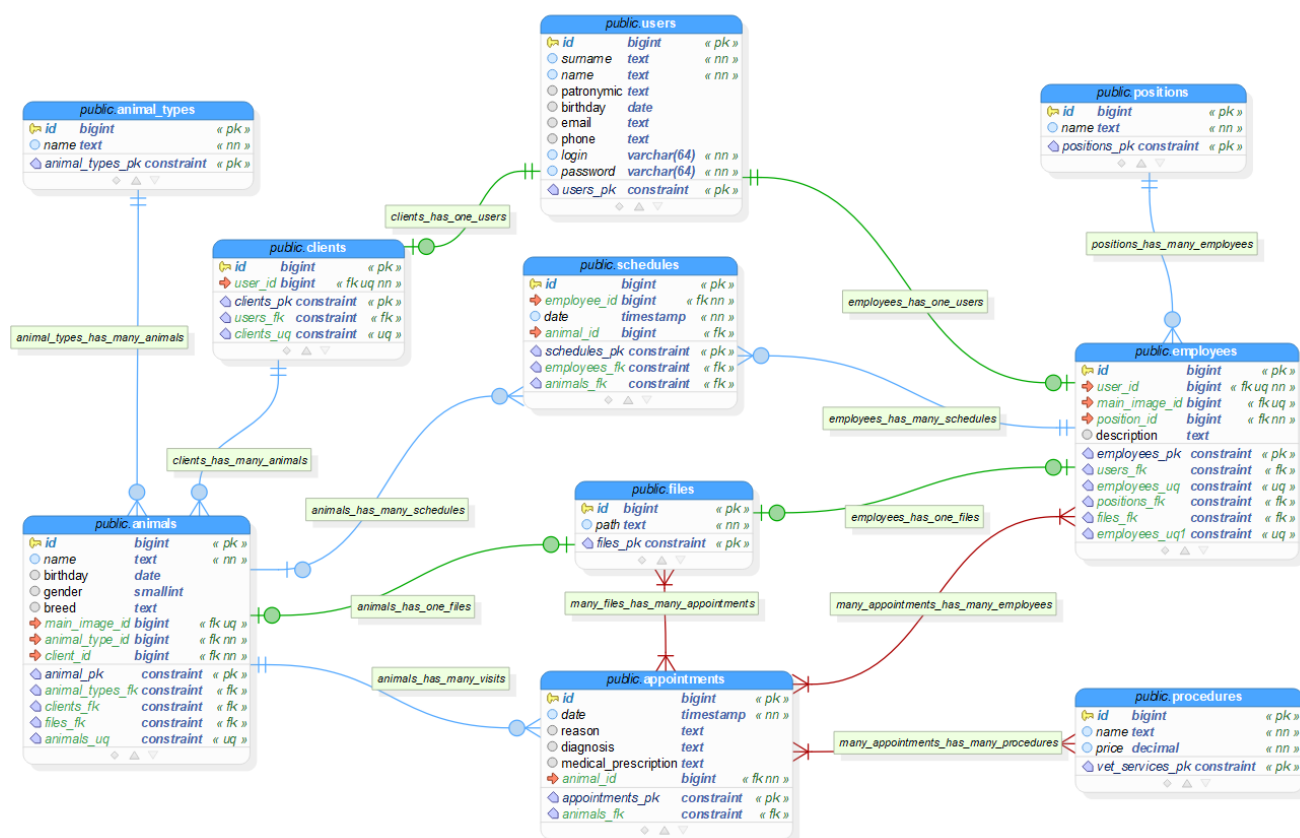


Рисунок 1 – Физическая модель данных

3 Проектирование работы системы

Приложение построено на основе Spring Boot, который предоставляет мощный инструмент для создания веб-приложений с использованием аннотаций и упрощает работу с базой данных.

Система спроектирована с использованием архитектуры MVC (Model-View-Controller), что обеспечивает разделение логики приложения, пользовательского интерфейса и взаимодействия с данными.

Основные компоненты системы:

- Модели (Entities):

Модели данных представляют собой сущности в базе. Каждая модель соответствует таблице в базе данных, где поля модели отражают столбцы этой таблицы.

- Контроллеры (Controllers):

Контроллеры обрабатывают входящие HTTP-запросы, управляют маршрутизацией страниц и взаимодействуют с сервисами для обработки данных.

- Сервисы (Services):

Сервисы выполняют бизнес-логику приложения. Они взаимодействуют с репозиториями, обрабатывают данные и передают их в контроллеры. Сервисы также выполняют дополнительные задачи, такие как валидация данных, проверка уникальности и обработка ошибок.

- Репозитории (Repositories):

Репозитории отвечают за взаимодействие с базой данных. Они используют Spring Data JPA для выполнения операций CRUD (создание, чтение, обновление, удаление). Репозитории абстрагируют работу с базой данных, позволяя работать с сущностями как с обычными Java-объектами. Для хранения и управления данными используется система управления базами данных PostgreSQL.

- DTO:

DTO используются для передачи данных между слоями приложения,

					МИВУ 09.03.04-17.000 ПЗ	Лист
						11
Изм.	Лист	№ докум.	Подпись	Дата		

особенно когда требуется передать только часть информации из более сложных объектов. Они используются для передачи данных от формы пользователя в контроллер или для отображения результатов в представлении.

- HTML-шаблоны (Thymeleaf):

HTML-шаблоны используются для отображения данных пользователю. В рамках приложения используется шаблонизатор Thymeleaf, который позволяет динамически вставлять данные в HTML-страницы. Шаблоны обрабатываются контроллерами и используются для отображения информации.

Разделение ролей в системе:

В системе предусмотрены две роли:

- Администратор (ADMIN) – имеет полный доступ к функционалу системы.

- Оператор (OPERATOR) – имеет ограниченный доступ.

Авторизация реализована с помощью Spring Security, который предоставляет готовые решения для реализации аутентификации и авторизации. Пользователь вводит логин и пароль, которые проверяются на сервере. После успешной аутентификации пользователь получает доступ к системе в соответствии с его ролью. При добавлении нового пользователя пароль кодируется перед записью в БД.

В приложении настроено разграничение доступа к маршрутам в зависимости от роли пользователя. Неавторизованным пользователям закрыты все старницы приложения. Если пользователь имеет соответствующие права доступа, то запрос будет успешным. В противном случае будет возвращен код ошибки 403 («запрещено»).

Доступные функциональные возможности в зависимости от роли:

Функциональные возможности доступные только админу:

- создание, обновление и удаление видов животных;
- создание, обновление и удаление должностей сотрудников;
- создание, обновление и удаление услуг;

					МИВУ 09.03.04-17.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		12

- создание, обновление, удаление, изменения роли в системе сотрудника;
- создание и удаление расписаний (свободных записей) для сотрудников;
- удаление приемов.

Функциональные возможности доступные как админу, так и оператору:

- создание, обновление и удаление питомцев;
- создание, обновление и удаление клиентов;
- создание и обновление приемов.

					МИВУ 09.03.04-17.000 ПЗ	Лист
						13
Изм.	Лист	№ докум.	Подпись	Дата		

4 Разработка и реализация системы

Реализация контроллера на примере контроллера питомца.

1. `getAnimalDetails(Long id, Model model)`

HTTP Метод: GET

Принимает:

`id` — ID питомца (через путь URL `/animals/details/{id}`).

Возвращает:

HTML-страницу с деталями питомца.

Валидация:

Проверка на наличие питомца с указанным ID в базе данных. Если питомец не найден, перенаправляет на страницу списка клиентов.

2. `createGet(Long clientId, Model model)`

HTTP Метод: GET

Принимает:

`clientId` — ID клиента, для которого создается питомец (параметр запроса `/animals/create?clientId={clientId}`).

Возвращает:

HTML-страницу для создания нового питомца с предзаполненными данными клиента и списком доступных типов животных.

Валидация:

Проверка наличия клиента с переданным ID. Если клиент не найден, перенаправляет на список клиентов.

3. `createPost(Long clientId, CreateAnimalDTO animalDTO, BindingResult bindingResult, Model model)`

HTTP Метод: POST

Принимает:

`clientId` — ID клиента (параметр запроса).

`animalDTO` — Объект DTO для создания питомца.

`bindingResult` — Результаты валидации данных из формы.

					МИВУ 09.03.04-17.000 ПЗ	Лист
						14
Изм.	Лист	№ докум.	Подпись	Дата		

Возвращает:

В случае ошибок валидации — повторно отображает форму с ошибками.

В случае успеха — перенаправление на страницу клиента с новым питомцем.

Валидация:

Валидация данных формы через аннотации в CreateAnimalDTO.

Используется объект BindingResult, чтобы обработать ошибки и показать пользователю уведомления о неправильных данных.

4. updateGet(Long id, Model model)

HTTP Метод: GET

Принимает:

id — ID питомца для редактирования (через путь URL /animals/update/{id}).

Возвращает:

HTML-страницу для редактирования данных питомца с предзаполненными значениями.

Валидация:

Проверка наличия питомца с переданным ID в базе данных. Если питомец не найден, перенаправляет на страницу списка клиентов.

5. updatePost(UpdateAnimalDTO animalDTO, BindingResult bindingResult, Model model)

HTTP Метод: POST

Принимает:

animalDTO — Объект DTO для обновления питомца.

bindingResult — Результаты валидации данных из формы.

Возвращает:

В случае ошибок валидации — повторно отображает форму с ошибками.

					МИВУ 09.03.04-17.000 ПЗ	Лист
						15
Изм.	Лист	№ докум.	Подпись	Дата		

В случае успеха — обновляет информацию о питомце и перенаправляет на страницу деталей питомца.

Валидация:

Валидация данных формы через аннотации в UpdateAnimalDTO.

Проверка существования питомца по ID. Если питомец не найден, перенаправляет на страницу списка клиентов.

6. deleteGet(Long id, Model model)

HTTP Метод: GET

Принимает:

id — ID питомца для удаления (через путь URL /animals/delete/{id}).

Возвращает:

HTML-страницу для подтверждения удаления питомца. Также проверяет, есть ли у питомца связанные приемы или записи на приемы, и если есть, показывает предупреждение.

Валидация:

Проверка наличия питомца с переданным ID в базе данных. Проверка на наличие приемов или записей на приемы. Если есть, отображается предупреждение.

7. deletePost(Long id, Model model)

HTTP Метод: POST

Принимает:

id — ID питомца для удаления (через путь URL /animals/delete/{id}).

Возвращает:

В случае наличия приемов или записей на приемы — повторно отображает форму с предупреждением.

В случае отсутствия записей — удаляет питомца и перенаправляет на страницу клиента.

Валидация:

Проверка наличия питомца с переданным ID в базе данных. Проверка на наличие назначенных визитов или расписаний перед удалением.

					МИВУ 09.03.04-17.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		16

5 Тестирование системы

Одним из важнейших этапов создания приложения является его тестирование и отладка. Целью тестирования является проверка работоспособности программы, правильности выполнения всех функций, а также правильности обработки всех исключений, возникающих в ходе работы программы.

Таблица 1 - Тестирование приложения

Удаление объектов		
№	Проверка	Результат
1	Удаление вида животного	Доступно только админу. Успешное удаление в случае, если вид не связан с питомцами.
2	Удаление должности	Доступно только админу. Успешное удаление в случае, если должность не назначена сотрудникам.
3	Удаление услуги	Доступно только админу. Успешное удаление в случае, если услуга не связана с приемами.
4	Удаление сотрудника	Доступно только админу. Успешное удаление в случае, если сотрудник не имеет приемов и/или активного расписания.
5	Удаление клиента	Успешное удаление в случае, если у клиента нет питомцев.
6	Удаление питомца	Успешное удаление в случае, если у питомца нет приемов и записей на

		приемы.
7	Удаление записей на прием у питомцев	Успешное удаление.
8	Удаление приемов	Доступно только админу. Успешное удаление.
9	Удаление расписания сотрудника	Доступно только админу. Успешное удаление.
Создание объектов		
10	Добавление вида животного	Доступно только админу. Успешное добавление. Проверка на ввод поля и уникальность названия.
11	Добавление должности	Доступно только админу. Успешное добавление. Проверка на ввод поля и уникальность названия.
12	Добавление услуги	Доступно только админу. Успешное добавление. Проверка на ввод поля и уникальность названия.
13	Добавление сотрудника	Доступно только админу. Успешное добавление. Проверка на ввод обязательных полей, уникальность логина, номера и email, проверка на корректный ввод.
14	Добавление клиента	Успешное добавление. Проверка на ввод обязательных полей, уникальность логина, номера и email, проверка на корректный ввод.
15	Добавление питомца	Успешное добавление. Проверка на ввод обязательных полей, проверка на корректный ввод.

16	Добавление записи на прием.	Успешное добавление. Проверка на выбор врачей и даты.
17	Добавление приема	Успешное добавление. Проверка на выбор врача и даты.
18	Добавление свободной записи для врача	Доступно только админу. Успешное добавление. Проверка ввода даты.
Изменение объектов		
19	Изменение вида животного	Доступно только админу. Успешное изменение. Проверка на ввод поля и уникальность названия.
20	Изменение должности	Доступно только админу. Успешное изменение. Проверка на ввод поля и уникальность названия.
21	Изменение услуги	Доступно только админу. Успешное изменение. Проверка на ввод поля и уникальность названия.
22	Изменение сотрудника	Доступно только админу. Успешное изменение. Проверка на ввод обязательных полей, уникальность логина, номера и email, проверка на корректный ввод.
23	Изменение клиента	Доступно только админу. Успешное изменение. Проверка на ввод обязательных полей, уникальность логина, номера и email, проверка на корректный ввод.
24	Изменение питомца	Успешное добавление. Проверка на ввод обязательных полей, проверка на корректный ввод.

25	Изменение приема	Успешное изменение.
26	Загрузка и отображение фото	Успешная загрузка и отображение фото.
27	Скачивание отчета	Успешное формирование отчетов в формате excel.
28	Изменение роли сотрудника	Доступно только админу. Успешное изменение.

Заключение

В рамках данной курсовой работе в соответствии с заданием было - приложение для управления деятельностью ветеринарной клиники. Для достижения поставленной цели были выполнены следующие этапы:

- спроектированы и разработаны модели данных;
- создана база данных;
- разработано и протестировано приложение.

Программа содержит следующие функциональные возможности:

- авторизация пользователей;
- Управление информацией о сотрудниках.
- Ведение расписания работы врачей с возможностью редактирования.
- Управление услугами клиники.
- Управление клиентами и их питомцами.
- Ведение медицинских карт питомцев.
- Управление приемами.
- Генерация сводных отчетов по различным критериям, сохранение отчетов в виде файлов.

Программа соответствует требованиям, указанным в техническом задании.

Для ознакомления с данным проектом можно использовать ссылку на репозиторий (приложение 1).

Список используемой литературы

1. Блох, Дж. Jva. Эффективное программирование / Дж. Блох; перевод В.Стрельцов ; под редакцией Р. Усманов. — 2-е изд. - Саратов : Профобразование, 2019. — 310 с. — ISBN 978-5-4488-0127-3. - Текст : электронный // Электронно-библиотечная система IPR BOOKS: [сайт].
2. Свистунов, А. Н. Построение распределенных систем на Java : учебное пособие / А.Н.Свистунов. — 3-е изд. - Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 316 с. — ISBN 978-5-4497-0940-0. — Текст: электронный // Цифровой образовательный ресурс IPR SMART : [сайт].
3. Мухамедзянов, Р. Р. JAVA. Серверные приложения / Р. Р. Мухамедзянов — Москва : СОЛОН-Р, 2016. - 336 с. - ISBN 5-93455-134-5. - Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт].

Приложение 1. Текст программы

Ссылка на GitHub страницу репозитория с исходным кодом программы:

https://github.com/Marina-Lyamina/Spring_VetClinic

					МИВУ 09.03.04-17.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		23

Приложение 2. Снимки окон программы

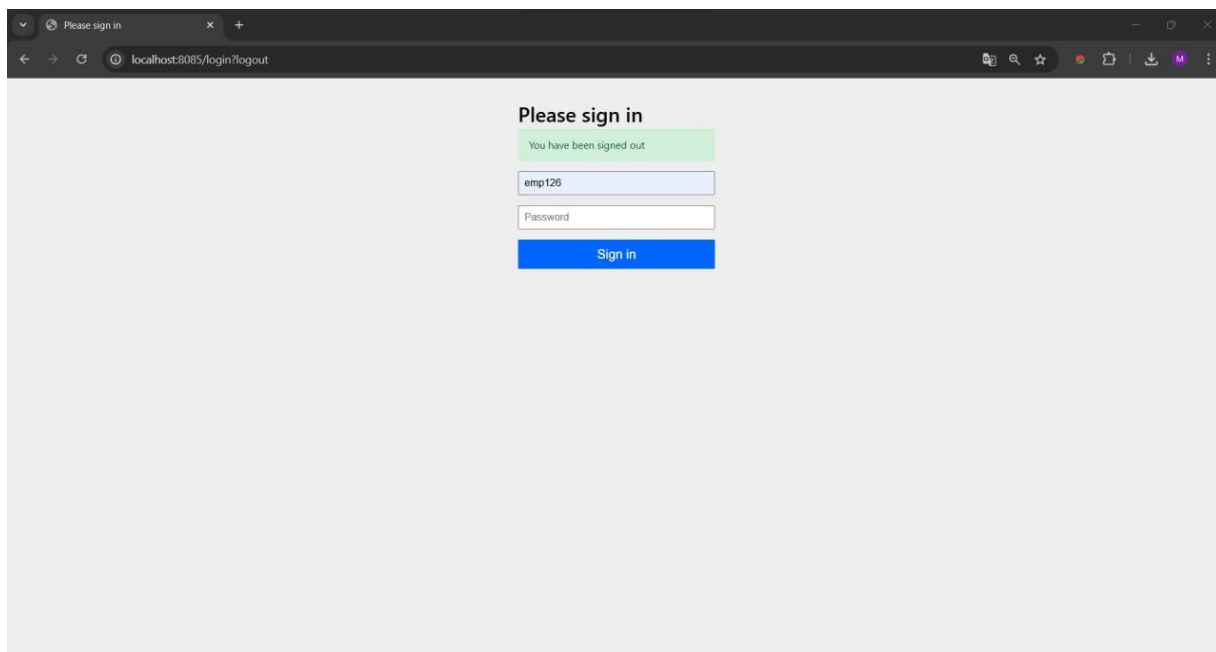


Рисунок 2 – скриншот приложения

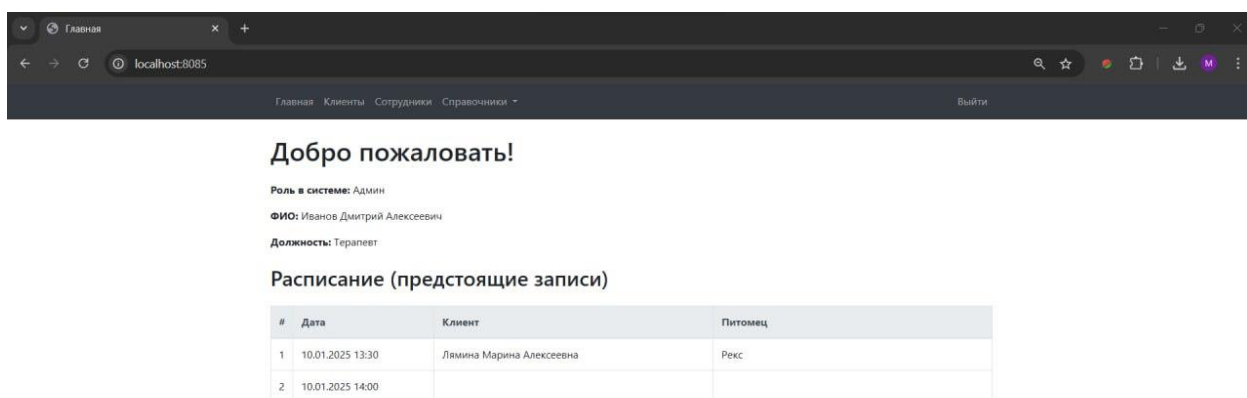


Рисунок 3 – скриншот приложения

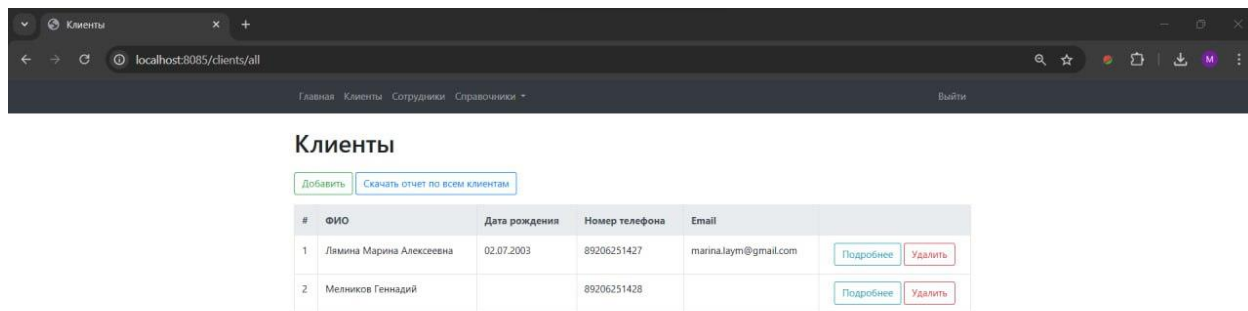


Рисунок 4 – скриншот приложения

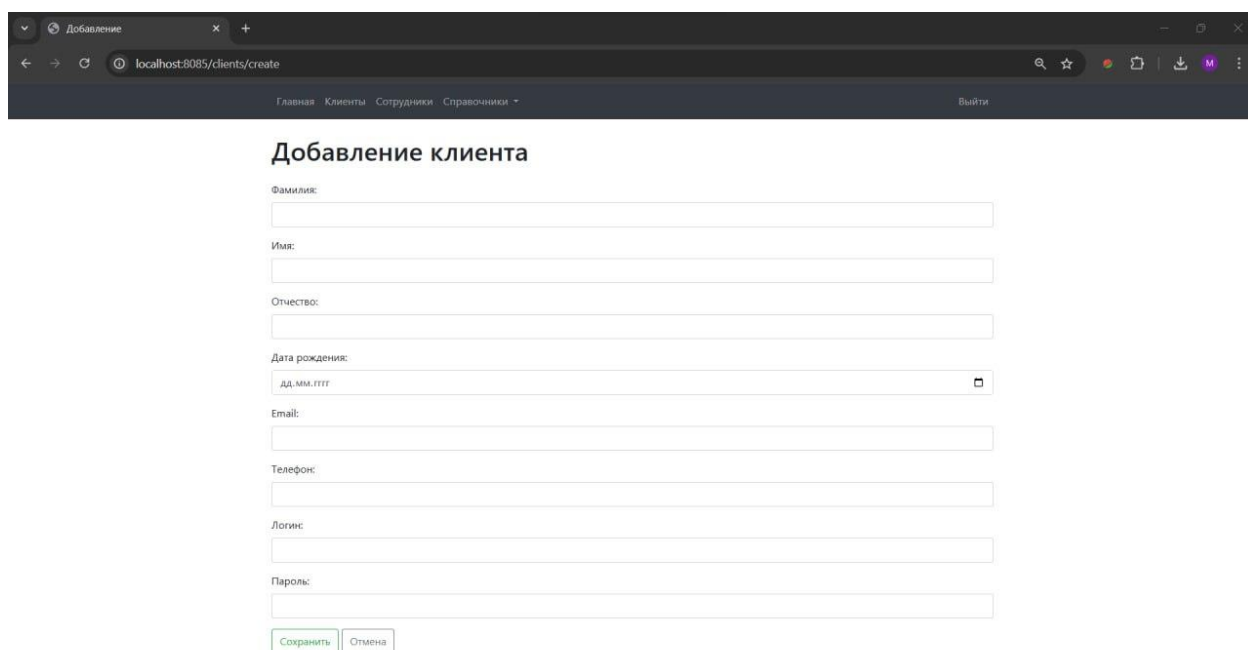


Рисунок 5 – скриншот приложения

clients_report (1).xlsx - Excel

Вход

Файл Главная Вставка Разметка страницы Формулы Данные Рецензирование Вид Справка Что вы хотите сделать?

Вставить

Буфер обмена

Шрифт

Выравнивание

Число

Общий

Условное форматирование

Форматировать как таблицу

Стили ячеек

Вставить

Удалить

Формат

Ячейки

Редактирование

Надстройки

Надстройки

A1

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	#	Фамилия	Имя	Отчество	Дата рождения	Номер телефона	Email							
2	1	Лямина	Марина	Алексеевна	2003-07-02	89206251427	marina.laym@gmail.com							
3	2	Мельников	Геннадий			89206251428								
4														
5														
6														
7														
8														
9														
10														
11														
12														
13														
14														
15														
16														
17														

Клиенты

Готово Специальные возможности: все в порядке

100 %

Рисунок 6 – отчет

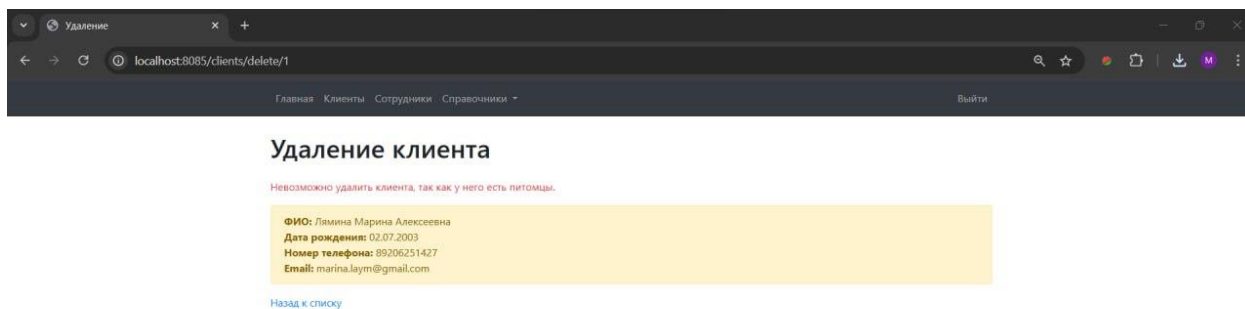


Рисунок 7 – скриншот приложения

					МИВУ 09.03.04-17.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		26

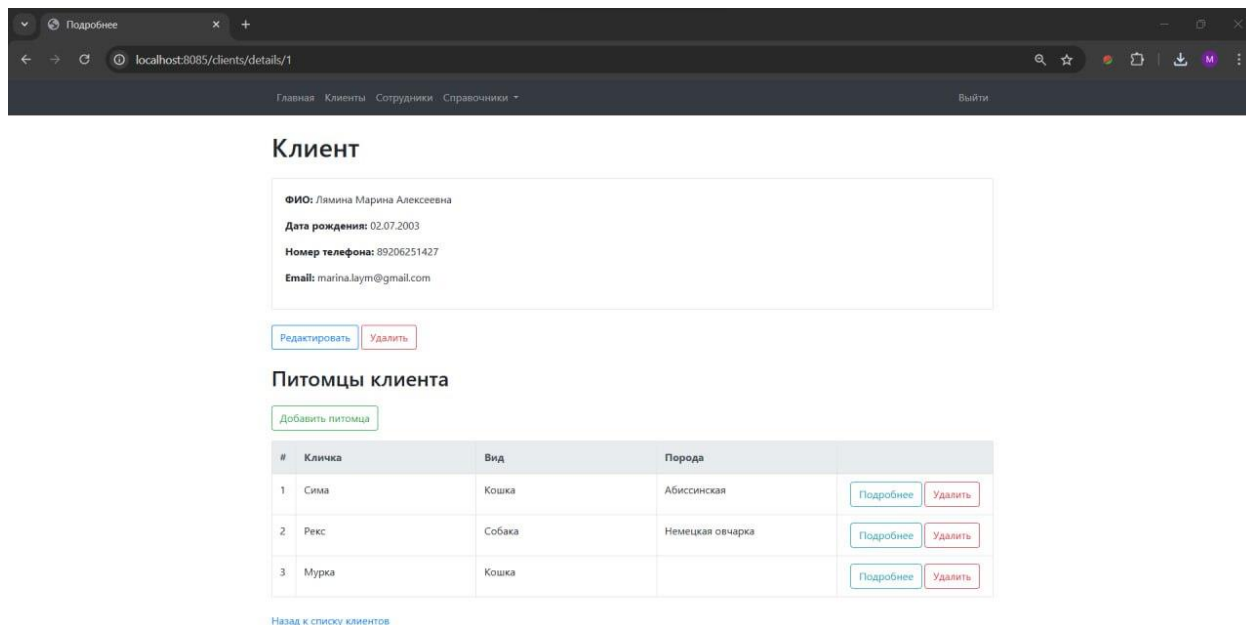


Рисунок 8 – скриншот приложения

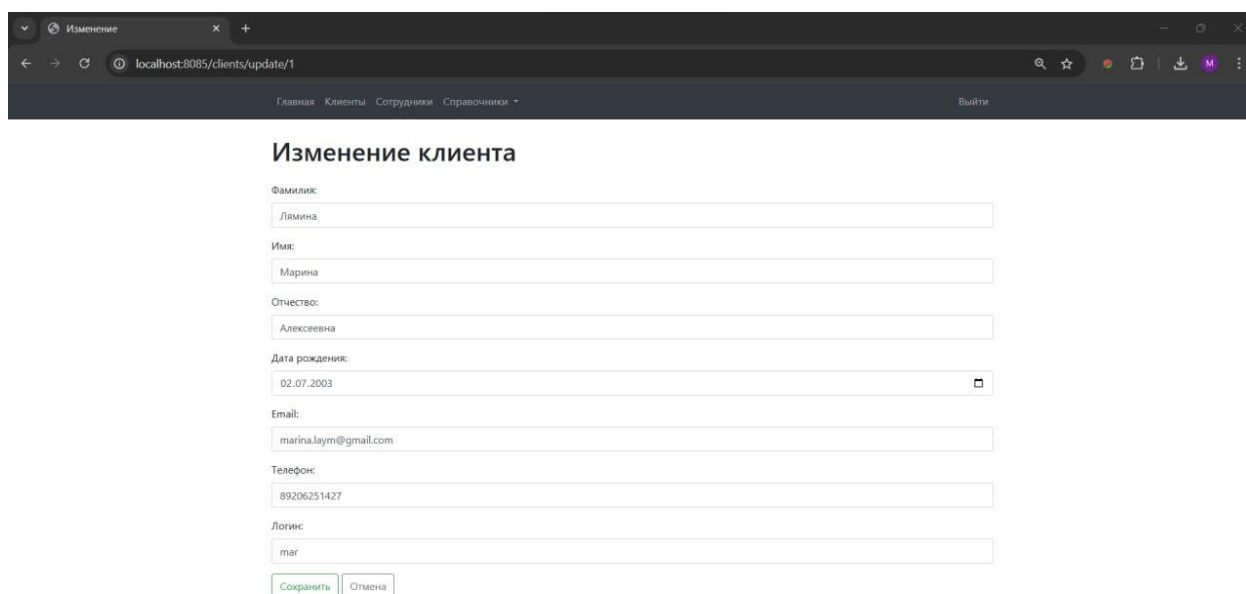


Рисунок 9 – скриншот приложения

Добавление

localhost:8085/animals/create?clientId=1

Главная Клиенты Сотрудники Справочники

Выйти

Добавление питомца

Клиент: Лямина Марина Алексеевна

Кличка:

Вид:

Собака

Дата рождения:

ДД.ММ.ГГГГ

Пол:

М

Порода:

Сохранить Отмена

Рисунок 10 – скриншот приложения


Подробнее

localhost:8085/animals/details/12

Главная Клиенты Сотрудники Справочники

Выйти

Питомец



Кличка: Сима
Вид: Кошка
Дата рождения: 17.12.2021
Пол: Ж
Порода: Абиссонская

Редактировать Загрузить фото Удалить

Записи на приемы

Добавить запись на прием

#	Дата	Сотрудник (должность)
---	------	-----------------------

Приемы

Добавить прием

#	Дата	Сотрудник (должность)	
1	23.11.2024 14:00	Евгений Константинович Сергеевич (ветеринар)	Подробнее Удалить
2	14.10.2021 14:00	Младшая Чиндэлай Романовна (ветеринар) Евгений Константинович Сергеевич (администратор)	Подробнее Удалить
3	09.01.2023 19:05	Евгений Константинович Сергеевич (ветеринар) Младшая Чиндэлай Романовна (ветеринар)	Подробнее Удалить

Назад к списку

Рисунок 11 – скриншот приложения

Измение

localhost:8085/animals/update/12

Главная

Клиенты

Сотрудники

Справочники

Выйти

Изменение питомца

Клиент: Лямина Марина Алексеевна

Кличка:

Вид:

Дата рождения:

Пол:

Порода:

Сохранить

Отмена

Рисунок 12 – скриншот приложения

Измение

localhost:8085/animals/updatePhoto/12

Главная

Клиенты

Сотрудники

Справочники

Выйти

Изменение фото питомца

Выберите файл

Файл не выбран

Сохранить

Отмена

Рисунок 13 – скриншот приложения

					МИВУ 09.03.04-17.000 ПЗ	Лист
						29
Изм.	Лист	№ докум.	Подпись	Дата		

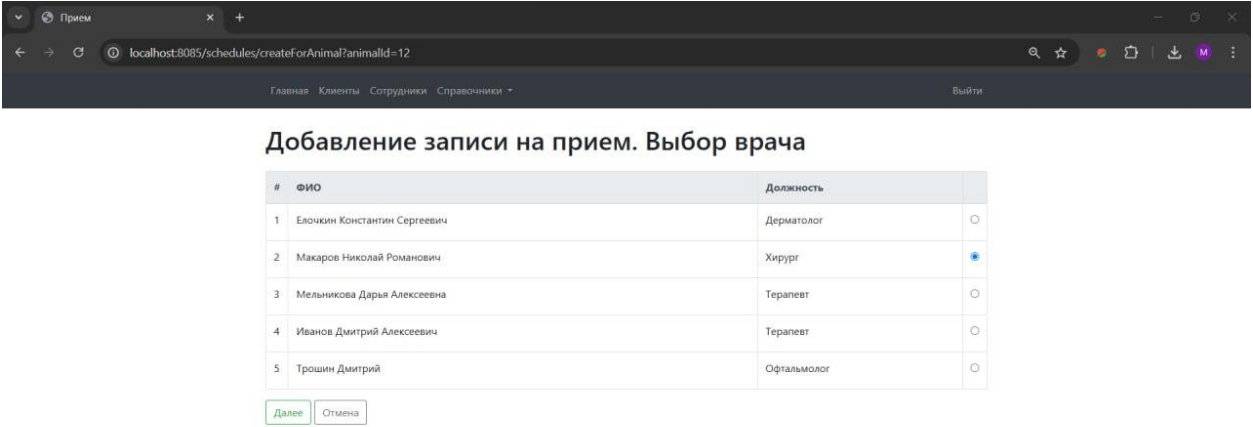


Рисунок 14 – скриншот приложения

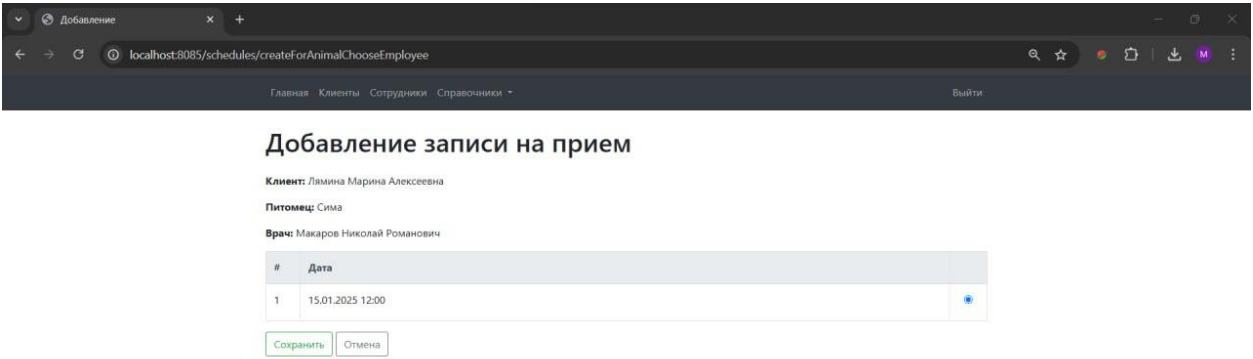


Рисунок 15 – скриншот приложения

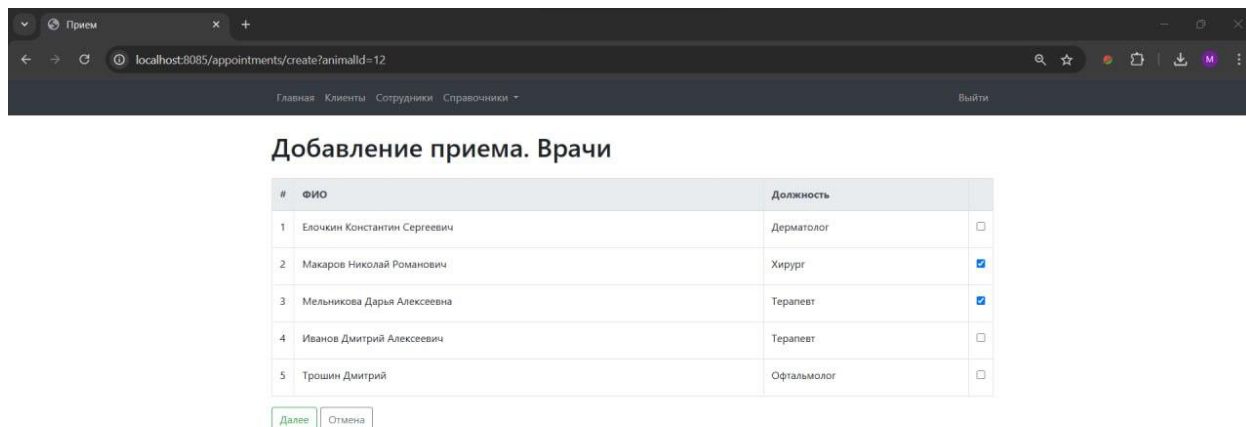


Рисунок 16 – скриншот приложения

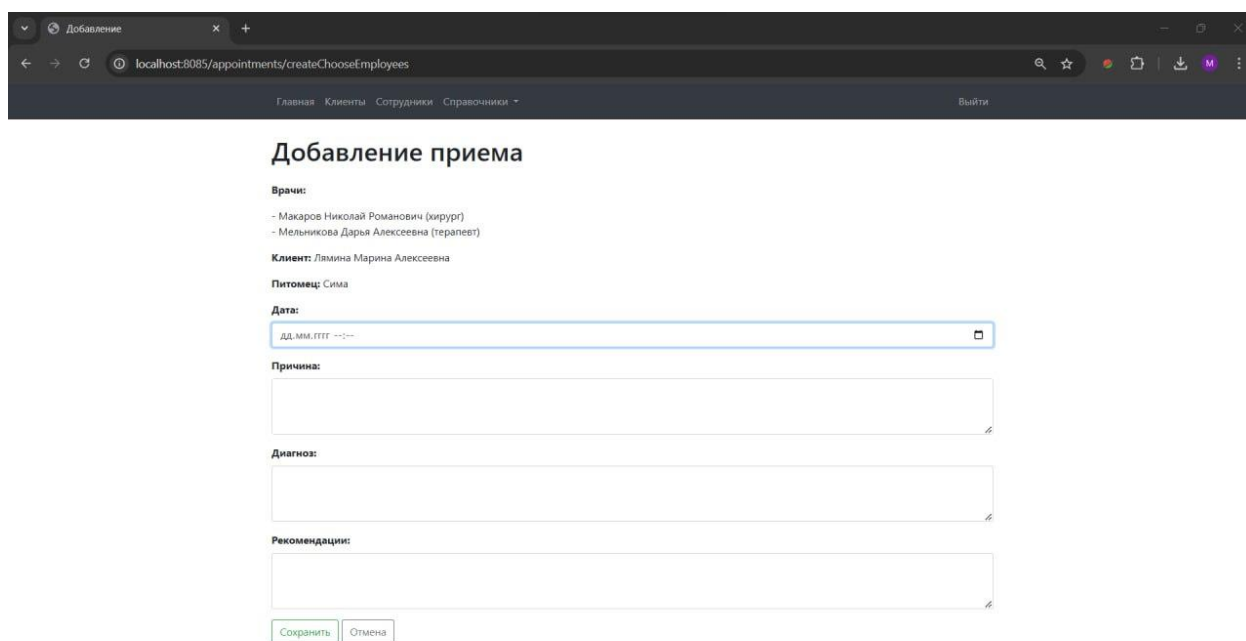


Рисунок 18 – скриншот приложения

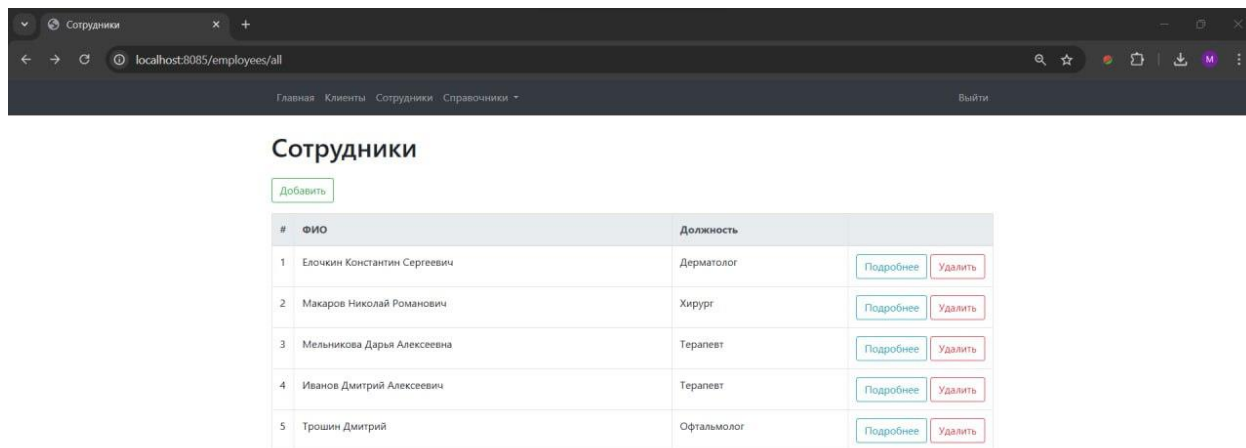


Рисунок 19 – скриншот приложения

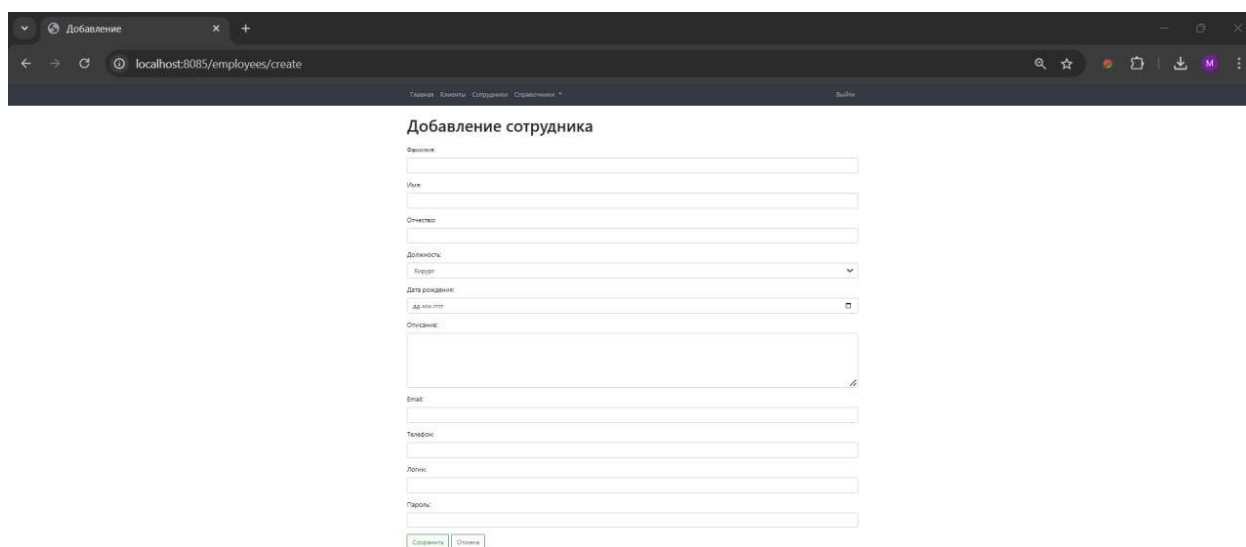


Рисунок 20 – скриншот приложения

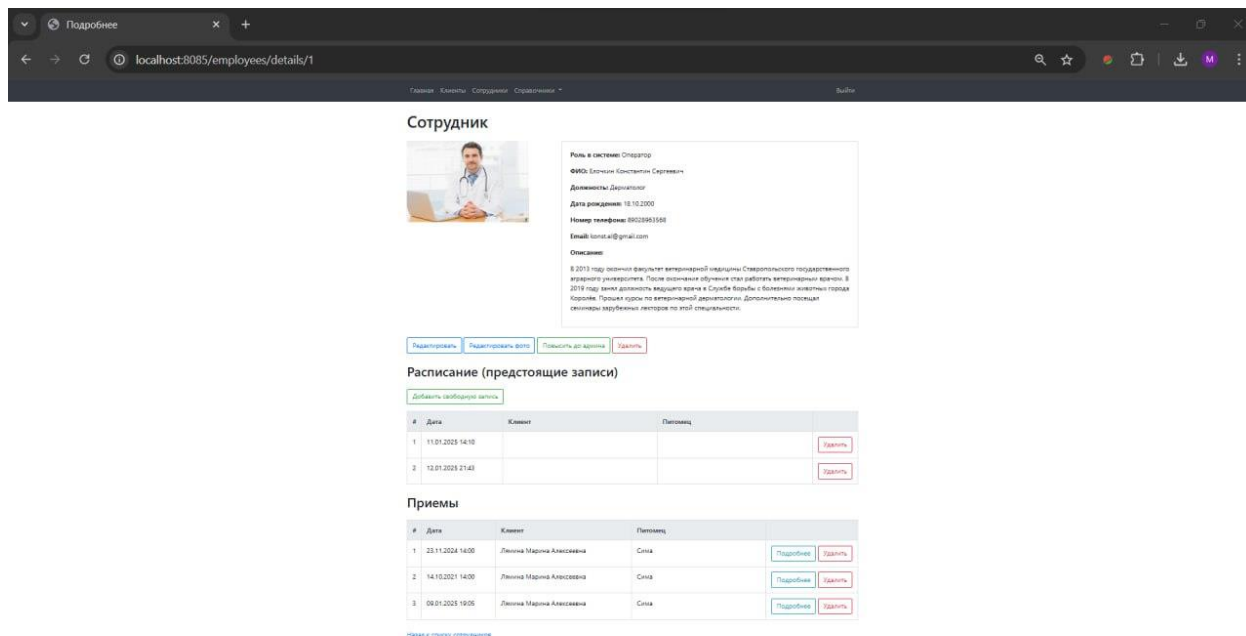


Рисунок 21 – скриншот приложения

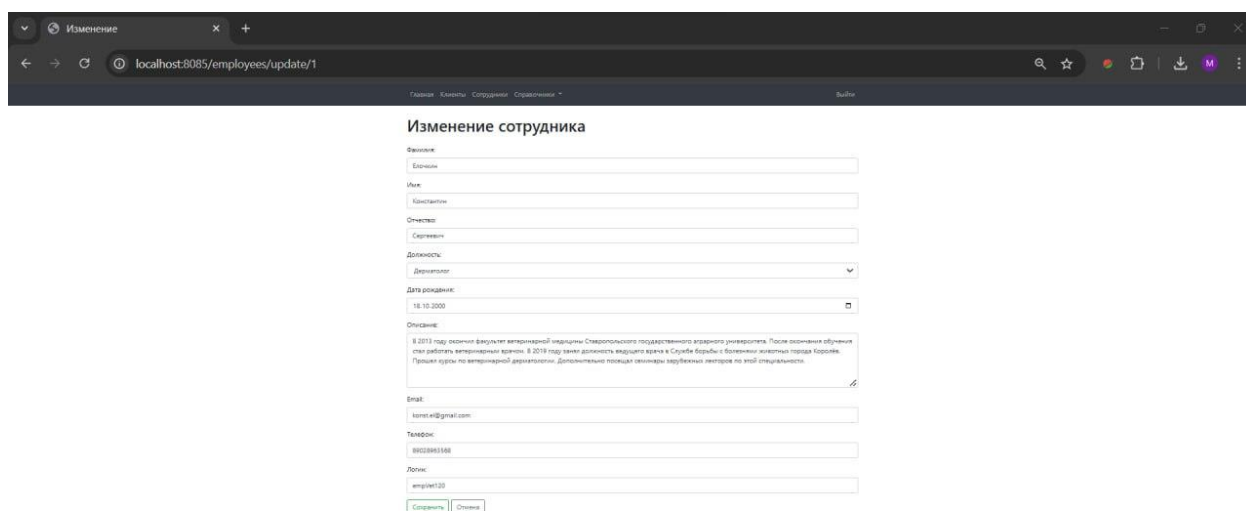


Рисунок 22 – скриншот приложения

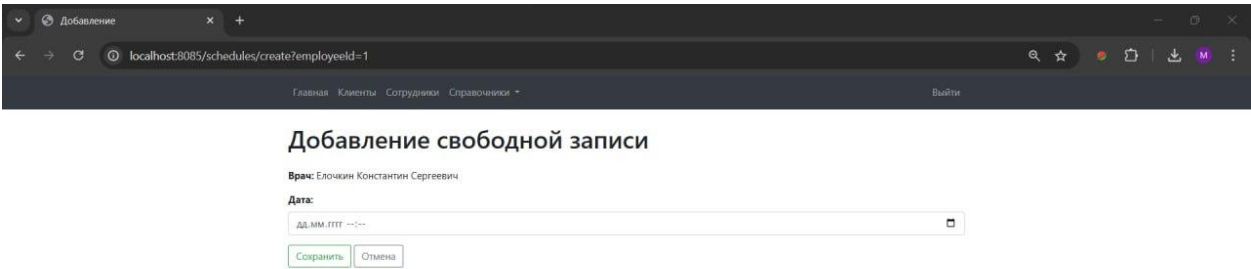


Рисунок 23 – скриншот приложения

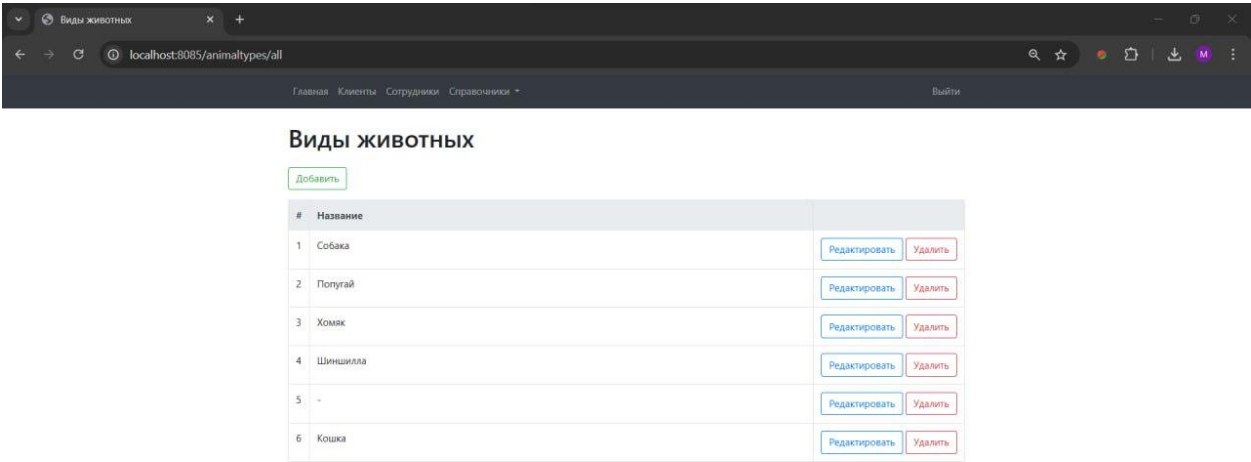


Рисунок 24 – скриншот приложения

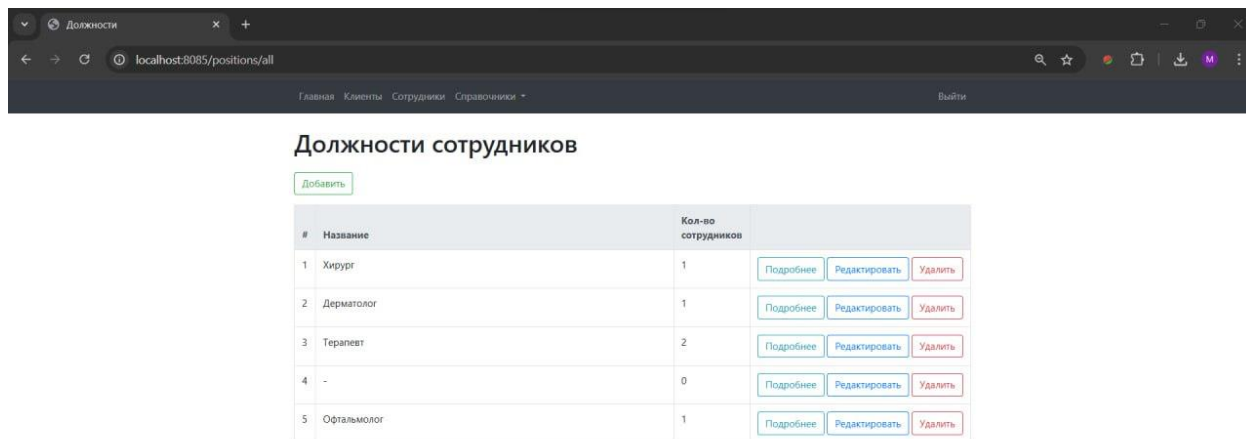


Рисунок 25 – скриншот приложения

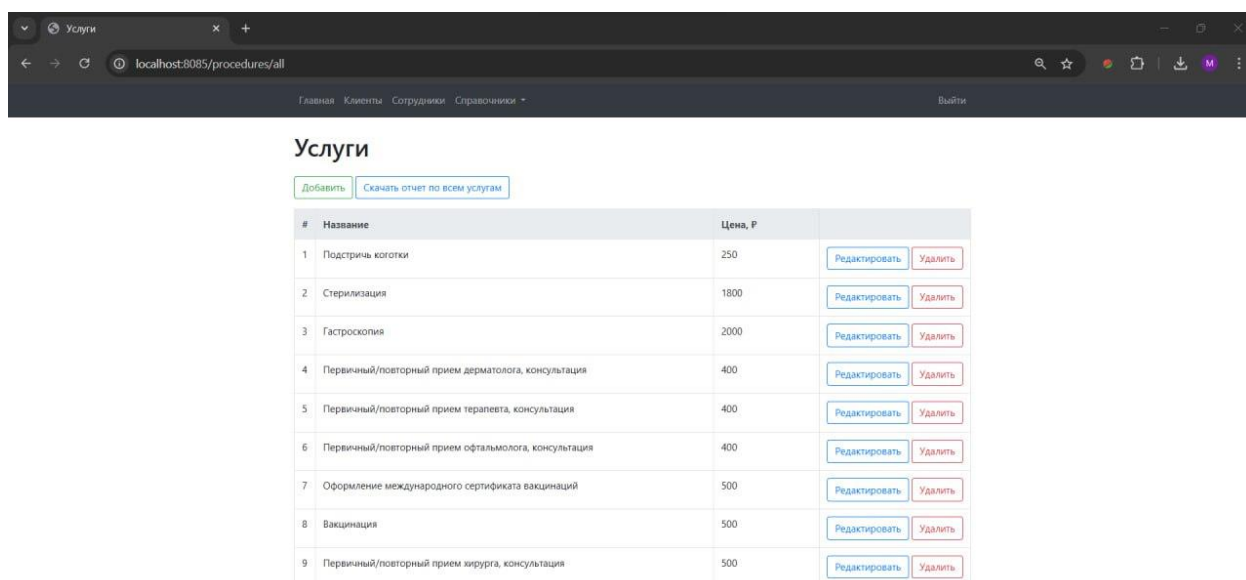


Рисунок 26 – скриншот приложения