МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное

образовательное учреждение высшего образования

«Самарский национальный исследовательский университет

имени академика С.П. Королева»

(Самарский университет)

Институт дополнительного образования

Цифровая кафедра

ИТОГОВАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

«WEB-ПРИЛОЖЕНИЕ ДЛЯ ГАЛЕРЕИ ПИТОМЦЕВ»

по программе подготовки «Frontend-разработка»

дополнительной профессиональной программы профессиональной переподготовки института дополнительного образования

очно-заочной формы обучения

Обучающийся Шелудько Полина Владимировна

(подпись, дата)

Руководитель ИКР\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.А. Буслаев

(подпись, дата)

Самара 2024

Задание на итоговую квалификационную работу

слушателю дополнительной образовательной программы профессиональной переподготовки *«Frontend-разработка»*

*Слушатель: Шелудько Полина Владимировна*

*Тема работы: web-приложение для галереи питомцев.*

*Цель: реализовать web-приложение для галереи питомцев с использованием JavaScript и jQuery.*

*Задачи:*

1. создать web-приложение с помощью HTML и CSS;

2. реализовать функции добавления, удаления, редактирования записей в таблицы и навигацию между таблицами с помощью JavaScript и jQuery.

Требования к содержанию и результатам итоговой квалификационной работы:

- Перемещение между страницами должно быть реализовано через навигационную панель;

- Для каждой таблицы необходимо реализовать механизм просмотра, добавления, редактирования и удаления записей;

- При работе с зависящими таблицами необходимо давать выбирать пользователю зависимости через выпадающий список;

- Приложение должно сохранять данные из таблиц в localStorage.

Выполненная итоговая квалификационная работа является формой итоговой аттестации по дополнительной образовательной программе профессиональной переподготовки *«Frontend-разработка».*

С заданием на выполнение итоговой квалификационной работы ознакомлен:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Шелудько Полина Владимировна

СОДЕРЖАНИЕ

Оглавление

[Введение 4](#_Toc166787400)

[1. Описание предметной области 5](#_Toc166787401)

[1.1. Схема базы данных 5](#_Toc166787402)

[2. Реализация системы 6](#_Toc166787403)

[2.2. Описание интерфейса пользователя 6](#_Toc166787404)

[Заключение 9](#_Toc166787405)

[Список использованной литературы 10](#_Toc166787406)

[ПРИЛОЖЕНИЕ Код программы 11](#_Toc166787407)

### Введение

По данным 2022 года в сети Интернет насчитывалось более 1.97 миллиарда веб-сайтов [1]. Безусловно, в 2023 году это цифра заметно увеличилась и в будущем непременно продолжит расти. Следовательно, спрос на специалистов в области веб-разработки станет ещё выше, даже не смотря на возможность создания веб-сайтов с помощью онлайн-конструкторов. Потому что готовые решения чаще всего подходят лишь для реализации идей малого бизнеса, а автоматизация бизнес-процессов и сложных веб-приложений требуют непосредственного участия веб-разработчика.

Однако, благодаря современным технологиям, процесс освоения данной профессии становится всё более доступным. А потому даже начинающие специалисты могут создавать качественные макеты веб-приложений, пользуясь различными фреймворками. Опытные же веб-программисты с помощью данных структур экономят время и упрощают сам процесс разработки.

Таким образом, всё выше перечисленное обуславливает актуальность выполнения данной итоговой квалификационной работы, в ходе которой было необходимо разработать собственное веб-приложение галереи питомцев.

Объект исследования итоговой квалификационной работы – разработка веб-приложения. Предмет исследования – разработка веб-приложения с использованием библиотеки jQuery.

Целью итоговой квалификационной работы является реализация веб-приложения для галереи питомцев с использованием JavaScript и jQuery. Для достижения поставленной цели необходимо выполнить следующие задачи:

1. создать ER-модель базы данных;

2. создать web-приложение с помощью HTML и CSS;

3. реализовать функции добавления, удаления, редактирования записей в таблицы и навигацию между таблицами с помощью JavaScript и jQuery.

## Описание предметной области

Предметная область – «Галерея питомцев».

### Схема базы данных

Таблица «Питомцы»:

Имя – столбец, хранящий кличку питомцы; тип – text;

Пол – столбец, хранящий пол питомца (мальчик/девочка); тип – text;

Дата рождения – столбец, хранящий дату рождения питомца, для дальнейшего подсчета возраста; тип – date;

Возраст – столбец, хранящий значение возраста питомца в месяцах, значения получены на основе даты рождения; тип – number;

Семейство – столбец, хранящий информацию о типе питомца (кошка, собака, мышь и т.д.); тип – text;

## Реализация системы

Любой сайт состоит из двух частей – визуальной и программной. Над визуальной составляющей работают Frontend-разработчики, а над программной – Backend-специалисты. Первые используют такие технологии, как: HTML, CSS и JavaScript. Рассмотрим их подробнее [2].

### Описание интерфейса пользователя

Навигационная панель:

Галерея питомцев – ссылка ведет на главную страницу, которая по совместительству галерея, где расположены фотографии животных с указанием их кличек (рис. 1).

Реактивная форма – ссылка ведет на страницу с реактивной формой для заполнения таблицы с информацией о питомцах; (рис. 2).

Таблица «Питомцы»:

Имя – столбец, хранящий кличку питомцы;

Пол – столбец, хранящий пол питомца (мальчик/девочка);

Дата рождения – столбец, хранящий дату рождения питомца, для дальнейшего подсчета возраста;

Возраст – столбец, хранящий значение возраста питомца в месяцах, значения получены на основе даты рождения;

Семейство – столбец, хранящий информацию о типе питомца (кошка, собака, мышь и т.д.);

Форма:

Добавить нового питомца– название формы.

Пол – радио-кнопка для выбора девочка/мальчик.

Дата рождения – поле ввода даты.

Семейство – поле ввода типа питомца.

Добавить – кнопка, которая при нажатии позволяет добавить нового питомца в таблицу (рис. 3-4).

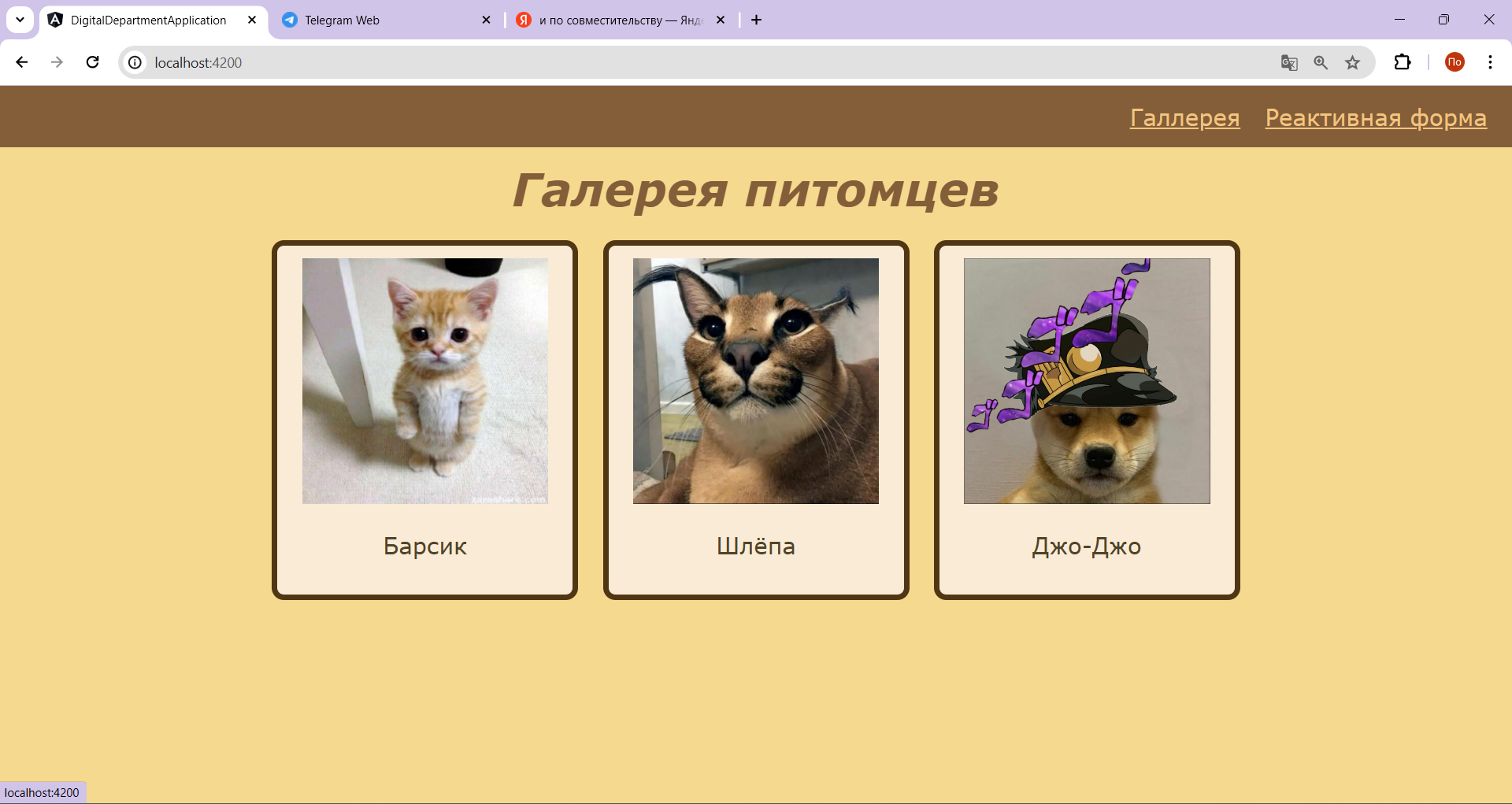


Рисунок 1 - Интерфейс web-приложения. Страница «галерея питомцев».

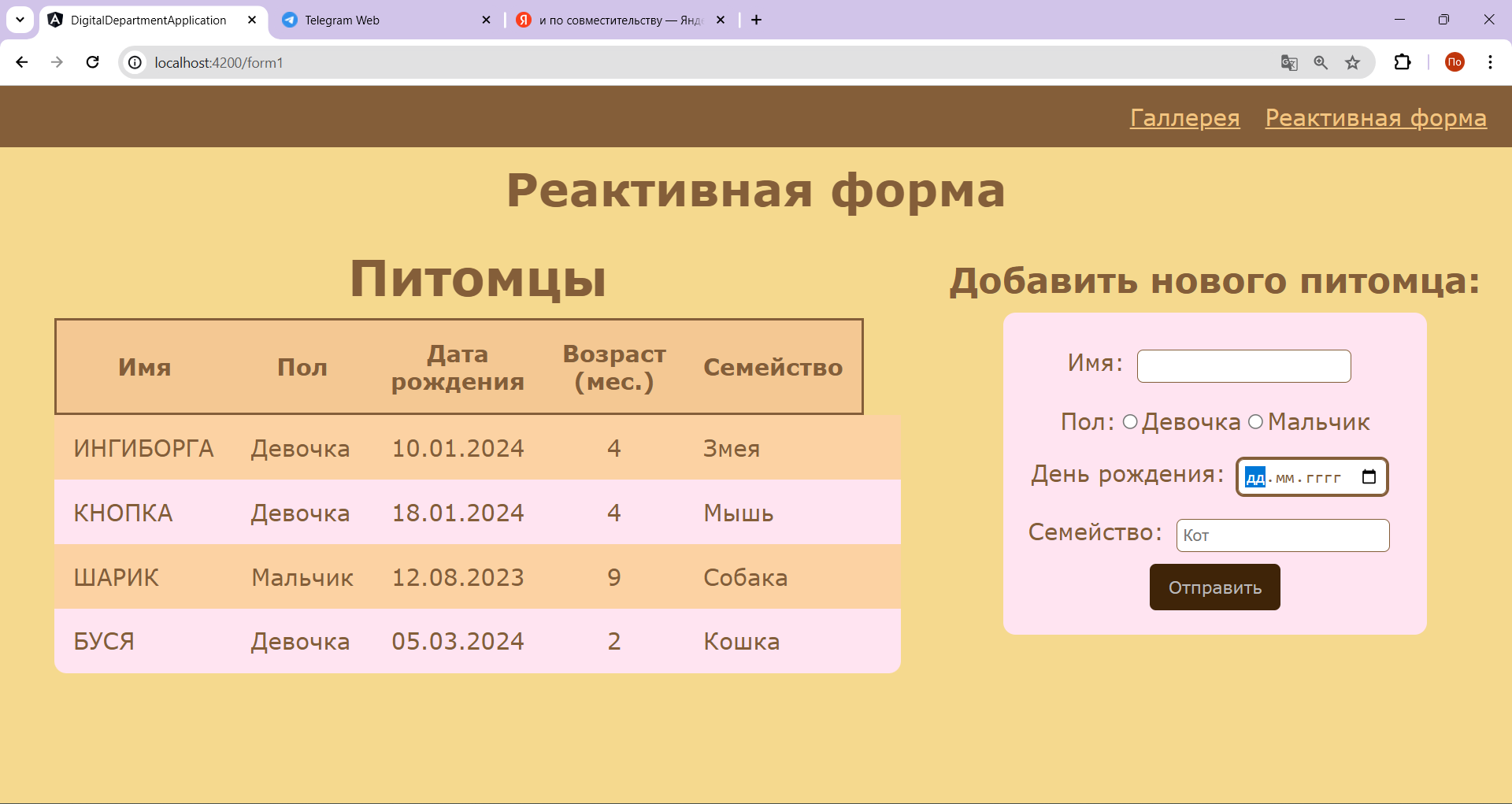


Рисунок 2 - Интерфейс web-приложения. Страница «реактивная форма».

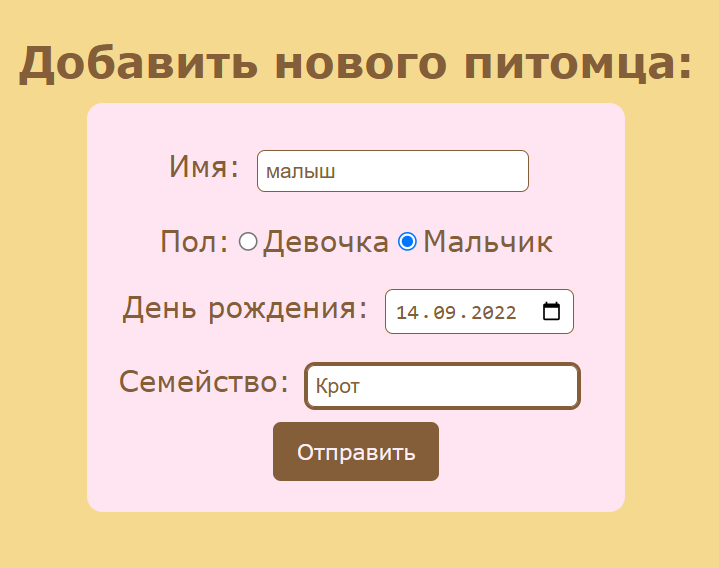


Рисунок 3 - Интерфейс web-приложения. Пример заполнения формы.

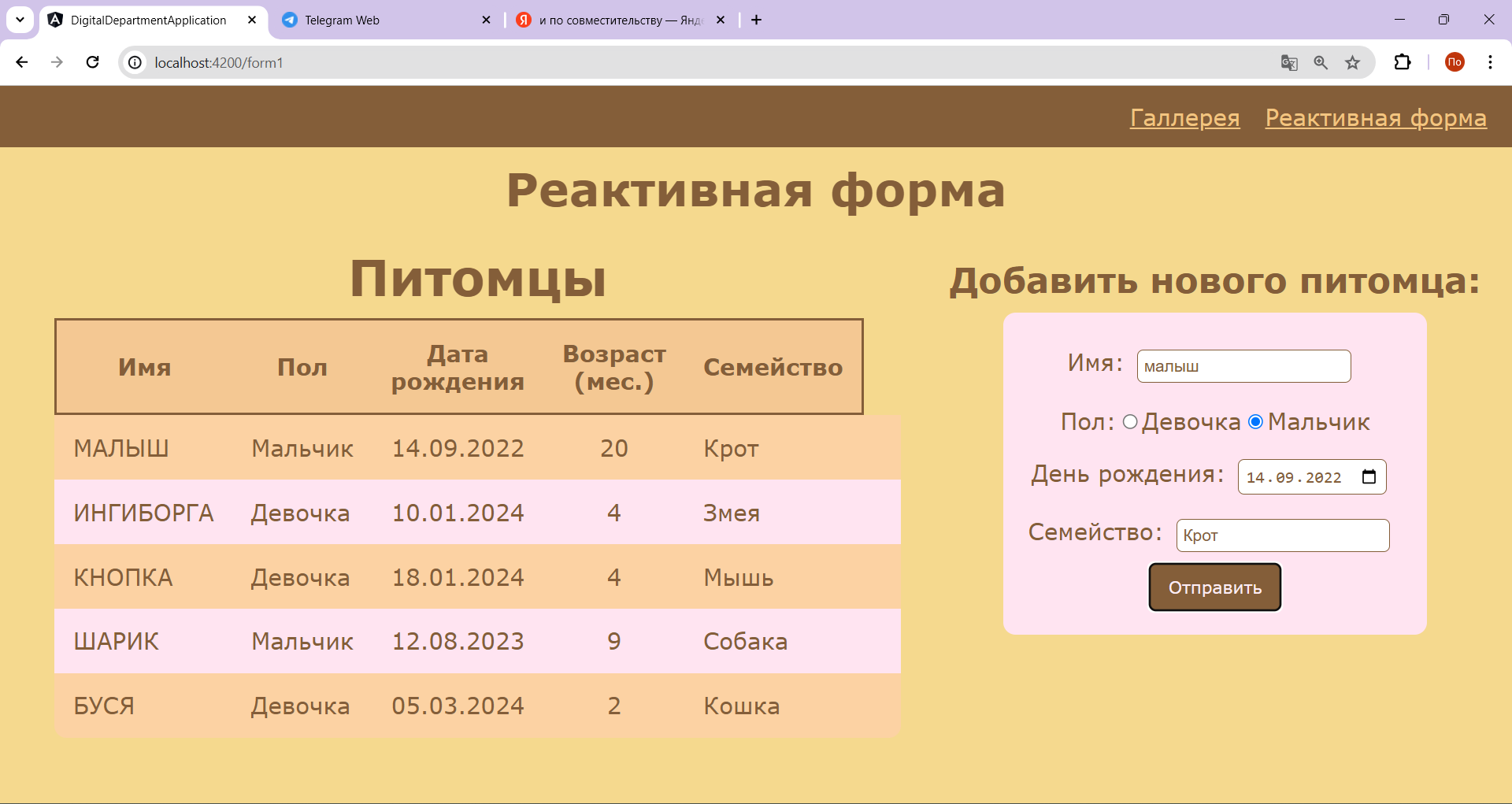


Рисунок 4 - Интерфейс web-приложения. Обновление таблицы в соответствии с информацией в форме.

## Заключение

В ходе выполнения итоговой квалификационной работы были исследованы принципы проектирования веб-приложений, на основе которых было разработано веб-приложение.

В ходе выполнения итоговой квалификационной работы были выполнены следующие задачи*:*

1. создана ER-модель базы данных;

2. создано web-приложение с помощью HTML и CSS;

3. реализованы функции добавления, удаления, редактирования записей в таблицы и навигация между таблицами с помощью JavaScript и jQuery.

В процессе выполнения итоговой квалификационной работы были закреплены теоретические и практические знания веб-программирования, полученные в ходе обучения.

В дальнейшем веб-приложение можно усовершенствовать, добавив в него новые компоненты и блоки, свидетельствующие о более богатом профессиональном опыте, а также доработать анимацию и улучшить вариант сохранения / чтения данных [2].

## Список использованной литературы

1. 100+ Интернет-статистики, фактов и тенденций на 2023 год [Электронный ресурс] URL: https://www.websiterating.com/ru/research/internet-statistics-facts/ (дата обращения: 26.05.2023).
2. Calameo [Электронный ресурс] URL: https://www.calameo.com/read/006999579ed1b87c7b6e0 (дата обращения: 26.05.2023).
3. Основы HTML [Электронный ресурс] URL: https://developer.mozilla.org/ru/docs/Learn/Getting\_started\_with\_the\_web/HTML\_basics (дата обращения: 26.05.2023).
4. Самоучитель CSS [Электронный ресурс] URL: http://htmlbook.ru/sam-css (дата обращения: 26.05.2023).
5. JavaScript [Электронный ресурс] URL: https://blog.skillfactory.ru/glossary/javascript/ (дата обращения: 26.05.2023).
6. jQuery [Электронный ресурс] URL: https://blog.skillfactory.ru/glossary/jquery/ (дата обращения: 26.05.2023).

## ПРИЛОЖЕНИЕ Код программы

Для галереи:

<h1 style="font-style: italic;" class="page-header">Галерея питомцев</h1>

<div class="gallery">

  <div class="photo-container">

    <img src="./assets/1.jpg" alt="Photo 1">

    <p>Барсик</p>

  </div>

  <div class="photo-container">

    <img src="./assets/2.jpg" alt="Photo 2">

    <p>Шлёпа</p>

  </div>

  <div class="photo-container">

    <img src="./assets/3.jpg" alt="Photo 3">

    <p>Джо-Джо</p>

  </div>

</div>

import { Component } from '@angular/core';

@Component({

  selector: 'app-page1',

  templateUrl: './page1.component.html',

  styleUrls: ['./page1.component.css']

})

export class Page1Component {

}

Для формы:

<h1 class="page-header">Реактивная форма</h1>

<div class="content">

    <table>

        <caption>Питомцы</caption>

        <tr>

            <th>Имя</th>

            <th>Пол</th>

            <th>Дата рождения</th>

            <th>Возраст (мес.)</th>

            <th>Семейство</th>

        </tr>

        <tr \*ngFor="let k of pets | reverse: {isIncreasingAge:false, isNameUppercase:true}">

            <td>{{k.name}}</td>

            <td>{{k.gender}}</td>

            <td class="age">{{k.birthday| date: "dd.MM.yyyy"}}</td>

            <td class="age">{{k.age}}</td>

            <td>{{k.type}}</td>

            <td>{{k.dateOfCreation | date: "dd.MM.yyyy hh:mm:ss"}}</td>

        </tr>

    </table>

    <div class="form">

        <h2>Добавить нового питомца:</h2>

        <form [formGroup]="myForm" (ngSubmit)="onSubmit()">

            <label for="name">Имя:</label>

            <input class="pink-input" type="text" id="name" name="name" formControlName="name"> <br>

            <div class="invalid-field-warning" \*ngIf="myForm.controls['name'].invalid && myForm.controls['name'].touched">

                Не указано имя!

            </div>

            <div style="margin:10px 0px;">

                <label for="girl">Пол:</label>

                <input type="radio" formControlName="gender" id="girl" name="gender" value="Девочка">

                <label for="girl">Девочка</label>

                <input type="radio" formControlName="gender" id="boy" name="gender" value="Мальчик">

                <label for="boy">Мальчик</label> <br>

            </div>

            <div class="invalid-field-warning" \*ngIf="myForm.controls['gender'].invalid && myForm.controls['gender'].touched">

                Не указан пол!

            </div>

            <label for="birthday">День рождения:</label>

            <input class="pink-input" type="date" id="birthday" name="birthday" formControlName="birthday"><br>

            <label for="type">Семейство:</label>

            <input class="pink-input" type="text" id="type" name="type" formControlName="type" placeholder="Кот"> <br>

            <button class="submit-button" type="submit" [disabled]="myForm.invalid">Отправить</button>

        </form>

    </div>

</div>

import { Component } from '@angular/core';

import { FormControl, FormGroup, Validators } from '@angular/forms';

import { Pet } from 'src/app/core/data-classes/pet';

interface PetForm {

  name: FormControl<string>,

  gender: FormControl<string>,

  birthday: FormControl<Date>,

  type: FormControl<string>

}

@Component({

  selector: 'app-form1',

  templateUrl: './form1.component.html',

  styleUrls: ['./form1.component.css']

})

export class Form1Component {

  pets: Pet[] = [

    new Pet("Буся", "Девочка", new Date(2024, 2, 5), "Кошка"),

    new Pet("Шарик", "Мальчик", new Date(2023, 7, 12), "Собака"),

    new Pet("Кнопка", "Девочка", new Date(2024, 0, 18), "Мышь"),

    new Pet("Ингиборга", "Девочка", new Date(2024, 0, 10), "Змея")

  ]

  myForm: FormGroup = new FormGroup<PetForm>({

    name: new FormControl('', { validators: [Validators.required], nonNullable: true, }),

    gender: new FormControl('', { validators: [Validators.required], nonNullable: true }),

    birthday: new FormControl(new Date(), { nonNullable: true }),

    type: new FormControl('', { nonNullable: true })

  })

  constructor() {

  }

  onSubmit() {

    let values=this.myForm.getRawValue();

    if(values.type==''){

      if (values.gender=='Мальчик'){

        values.type="Кот"

      }

      if (values.gender=='Девочка'){

        values.type="Кошка"

      }

    }

    let today=new Date();

    let age=Math.ceil((today.getTime()-new Date(values.birthday).getTime())/(1000 \* 3600 \* 24));

    this.pets.push(new Pet(

      values.name,

      values.gender,

      values.birthday,

      values.type));

  }

}