|  |
| --- |
|  |
| 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)**

**КУРСОВАЯ РАБОТА**

по дисциплине: Шаблоны программных платформ языка Джава

по профилю: Разработка программных продуктов и проектирование информационных систем

направления профессиональной подготовки: 09.03.04 «Программная инженерия»

Тема: Web-приложение «Зоомагазин»

Студент: Марданян Артём Сосович Группа: ИКБО-20-19

Работа представлена к защите (дата) /Марданян А.С./

Руководитель: доцент, к.т.н., Лобанов Александр Анатольевич

Работа допущена к защите (дата) / Зорина Н.В./

Оценка по итогам защиты:

/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

М. РТУ МИРЭА. 2021 г.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 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)**

**ЗАДАНИЕ**

**на выполнение курсовой работы**

по дисциплине: Шаблоны программных платформ языка Джава

по профилю: Разработка программных продуктов и проектирование информационных систем

направления профессиональной подготовки: Программная инженерия (09.03.04)

Студент: Марданян Артём Сосович

Группа: ИКБО-20-19

Срок представления к защите: 20.05.2021

Руководитель: Зорина Наталья Валентиновна

**Тема:** Web-приложение «Зоомагазин»

**Исходные данные:** индивидуальное задание на разработку; документация по Spring Framework и JEE, документация по языку Java (версия не ниже 8); инструменты и технологии: JDK (не ниже 8), создание Spring MVC web-приложений, RESTful web-сервисов, Spring ORM, Gradle, GitHub, IntelliJIDEA. Нормативный документ: инструкция по организации и проведению курсового проектирования СМКО МИРЭА 7.5.1/04.И.05-18.

**Перечень вопросов, подлежащих разработке, и обязательного графического материала:** 1. Провести анализ предметной области и формирование основных требований к приложению. 2. Обосновать выбранные средств ведения разработки. 3. Разработать приложение с использованием фреймворка Spring и выбранной технологии, и инструментария. 4.Провести тестирование приложения. 5. Оформить пояснительную записку по курсовой работе 6. Провести анализ текста на антиплагиат 7. Создать презентацию по выполненной курсовой работе.

Руководителем проведён инструктаж по технике безопасности, противопожарной технике и правилам внутреннего распорядка.

Зав. кафедрой ИиППО: \_\_\_\_\_\_\_\_\_\_\_/Р. Г. Болбаков/, «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2021 г.

Задание на КР выдал: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Н.В. Зорина/, «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2021 г.

Задание на КР получил: \_\_\_\_\_\_\_\_\_\_\_/А.С. Марданян/, «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2021 г.

УДК 004.4

Марданян А.С. «Web-приложение «Зоомагазин»»/ Курсовая работа по дисциплине «Шаблоны программных платформ на языке Джава» профиля «Разработка программных продуктов и проектирование информационных систем» направления профессиональной подготовки бакалавриата 09.03.04 «Программная инженерия» (4-ый семестр) / руководитель старший преподаватель Н.В. Зорина / кафедра ИППО Института ИТ МИРЭА – с. 29, табл. 6, ист. 10.

Целью работы является создания веб-приложения «Зоомагазин».

В рамках работы осуществлены краткий анализ аналогов веб-приложения «Зоомагазин», обоснование выбора технологии разработки веб-приложения, создание веб-приложения и тестирование основных функций.

Mardanyan A.S. «”pet shop” web application»/ Coursework in the discipline «Java software platform templates» of the profile «Software development and information systems design» courses undergraduate 09.03.04. "Software Engineering" (4th semester) / head associate senior lecturer N.V. Zorina / department IPPO of institute IT RTU MIREA.

The aim of the work is creation “pet shop” web application.

As part of the work, a brief analysis of the analogs of the «pet shop» web application, justification of the choice of the web application development technology, creation of the web application and testing of the main functions were carried out.

М. МИРЭА. Ин-т ИТ. Каф. ИиППО. 2021 г. Марданян А.С.

**Оглавление**

[1. СБОР И АНАЛИЗ ТРЕБОВАНИЙ К ПРОГРАММНОМУ ПРОДУКТУ 5](#_Toc73936177)

[1.1. ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ 6](#_Toc73936178)

[1.2. Вывод 8](#_Toc73936179)

[2. РАЗРАБОТКА ИНТЕРНЕТ-МАГАЗИНА 9](#_Toc73936180)

[2.1. ПРОЕКТИРОВАНИЕ ПРОДУКТА 9](#_Toc73936181)

[2.2. ПРОЕКТИРОВАНИЕ ПРОДУКТА 9](#_Toc73936182)

[2.3. СТРУКТУРА ПРОГРАММНОГО ПРОДУКТА 9](#_Toc73936183)

[2.4. Вывод 18](#_Toc73936184)

[3. ТЕСТИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА 19](#_Toc73936185)

[3.1. ЦЕЛЬ 19](#_Toc73936186)

[3.2. ОБЛАСТИ, ПОДТВЕРГАЕМЫЕ ТЕСТИРОВАНИЮ 19](#_Toc73936187)

[3.3. РЕАЛИЗАЦИЯ ТЕСТИРОВАНИЯ 19](#_Toc73936188)

[3.4. Вывод 23](#_Toc73936189)

[Заключение 24](#_Toc73936190)

[Список информационных источников 24](#_Toc73936191)

**Введение**

В наше время технологии всё быстрее развиваются: буквально десять лет назад только появились первые сенсорные телефоны, или начали развиваться автомобильные компании, которые впоследствии стали выпускать электромобили... Многие сферы IT начали так быстро развиваться, что большинство устройств прошлого десятилетия либо адаптировались под современный мир, либо совсем исчезли. Интернет тоже подвергся изменениям, которые в основном связанны с безопасностью данных пользователя, что позволяет человеку спокойно искать нужную для себя информацию.

С момента появления интернета было создано огромное количество сайтов для разных нужд:

* Интернет-порталы[1] (площадка, включающая в себя обширные возможности и услуги)
* Интернет-форумы[2] (сайты, на которых люди обсуждают различные темы)
* Социальные сети[3] (возможность общения, знакомства или передачи данных)
* Интернет-магазины[4] (возможность покупать товары из дома)

и многие другие...

Веб-приложение по теме «Зоомагазин» как раз является по своей сути интернет-магазином.

1. СБОР И АНАЛИЗ ТРЕБОВАНИЙ К ПРОГРАММНОМУ ПРОДУКТУ

# ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

Данное веб-приложение представляет собой интернет-магазин зоомагазина. Тема для разрабатываемого веб-приложения всегда актуальна, так как все больше людей заводят домашних любимцев, для которых тоже необходимо покупать правильную пищу или лекарства.

В современных реалиях из-за коронавируса простым людям тяжело посещать зоомагазины для получения большей информации товарах. Там покупатель мог бы сам лично сравнивать и изучать товары. Но теперь все заказывают необходимые вещи через интернет-магазины, что сильно упрощает покупку и поиск товара. Такие сайты как «Бетховен»[5], «Samizoo»[6] или «PetShop»[7] дают всю нужную информацию покупателям.

Проведя анализ моей предметной области, становится очевидно, что многие Интернет-магазины «Зоомагазины» имеют практически одинаковый дизайн. Рассмотрим вышеперечисленные Интернет-ресурсы.

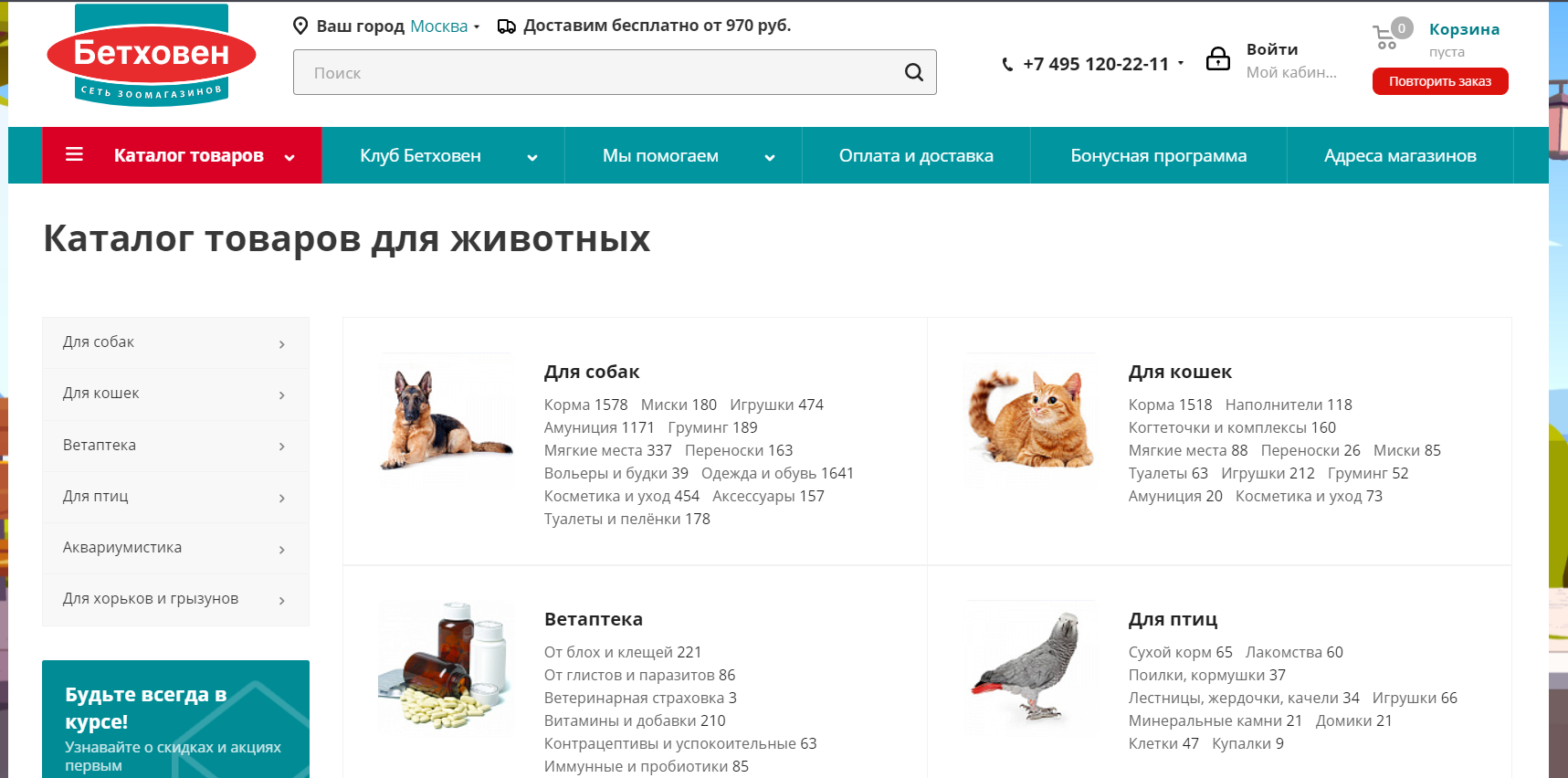


Рисунок 1.1 — интернет-магазин «Бетховен»

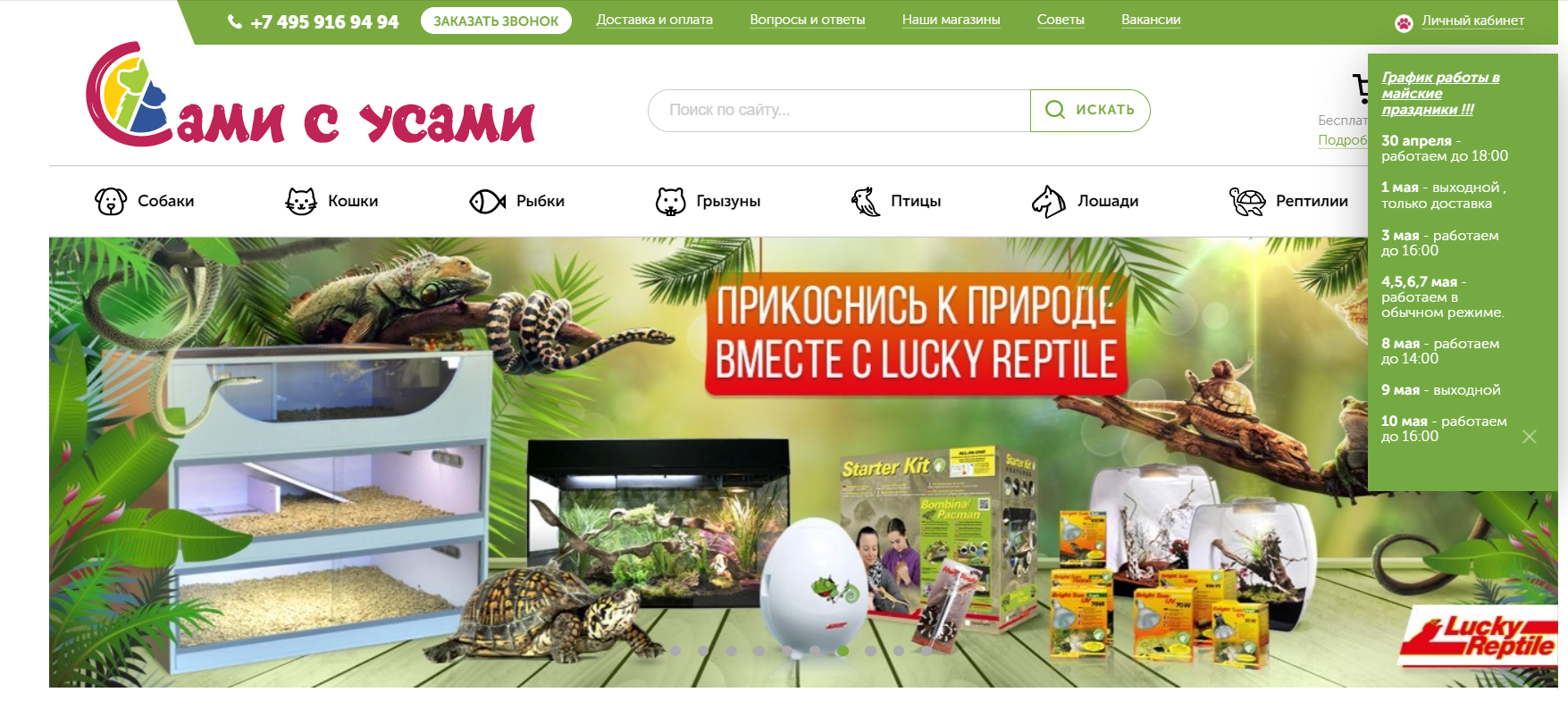


Рисунок 1.2 — интернет-магазин «Samizoo»

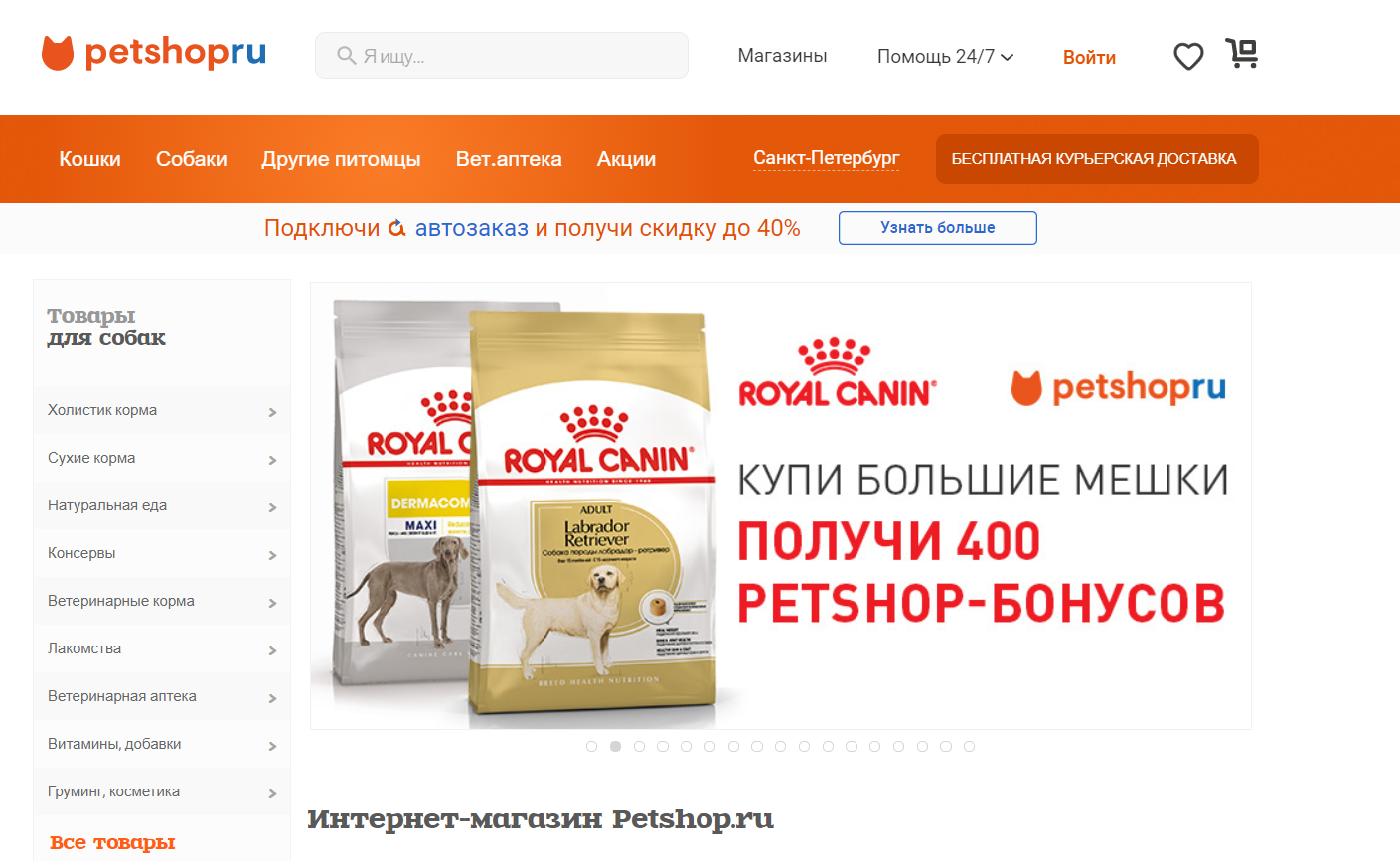


Рисунок 1.3 — интернет-магазин «PetShop»

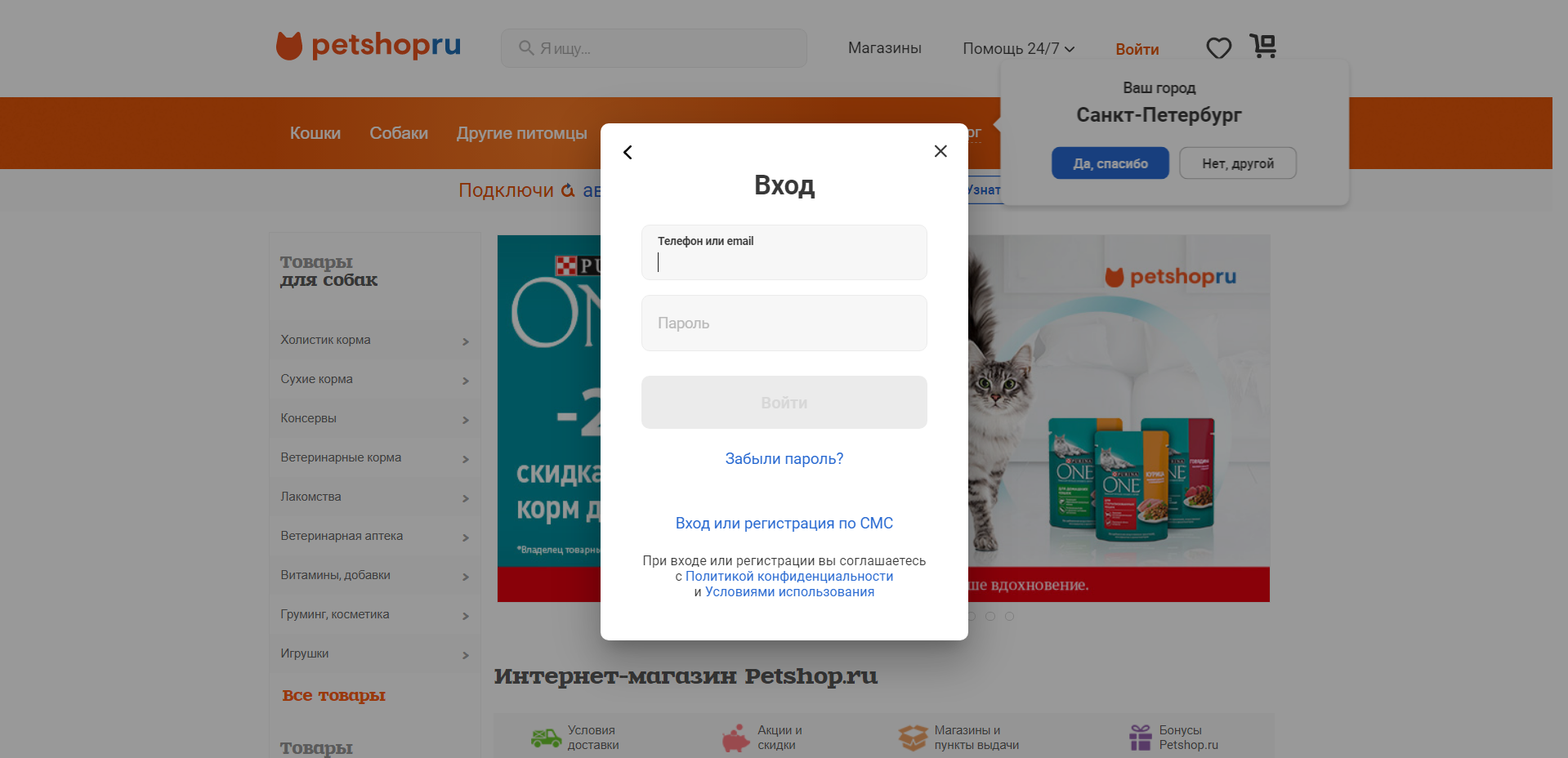


Рисунок 1.4 — регистрация и авторизация в

интернет-магазине «Samizoo»

# Вывод

Из всего вышесказанного можно сделать вывод, что данная тема актуальна и самыми важными для сайта являются:

* легкодоступная информация
* большой выбор товаров
* регистрация и вход в аккаунт пользователя
* простой дизайн

1. РАЗРАБОТКА ИНТЕРНЕТ-МАГАЗИНА

# ПРОЕКТИРОВАНИЕ ПРОДУКТА

После формирования списка общих требований в анализе предметной области необходимо сформулировать список требований заказчика, которые будут отражать его видение проекта и необходимость тестирования функционала.

Список основных требований заказчика (функции, которые должны быть в ПО и быть протестированы):

1. динамическое заполнение шаблонных мест в тексте;

2. регистрация и вход (и выход) в аккаунт пользователя;

3. возможность изменения уровня допуска пользователям и недопуск обычных пользователей на страницу администратора.

Данные функции могут быть разбиты на подфункции, например, тестирование регистрации и авторизация, т.к. они требуют от программного обеспечения проверять данные, введённые вручную

# ПРОЕКТИРОВАНИЕ ПРОДУКТА

Для разработки интернет-магазина были использованы Spring Boot[8], MySQL[9] – для управления базами данных, Thymeleaf[10] – для динамического заполнения шаблонных мест.

# СТРУКТУРА ПРОГРАММНОГО ПРОДУКТА

Структура программного продукта состоит из ряда папок:

* Config - класс, который будет читать из файлов свойств
* Controllers[11] - контроллеры обрабатывают запросы, и по этим запросам передают модель на определенные адреса
* Models – отвечает за создание базы данных
* Repository - несколько интерфейсов которые используют JPA Entity для взаимодействия с базой данных
* Templates – так называется моя папка с HTML разметкой

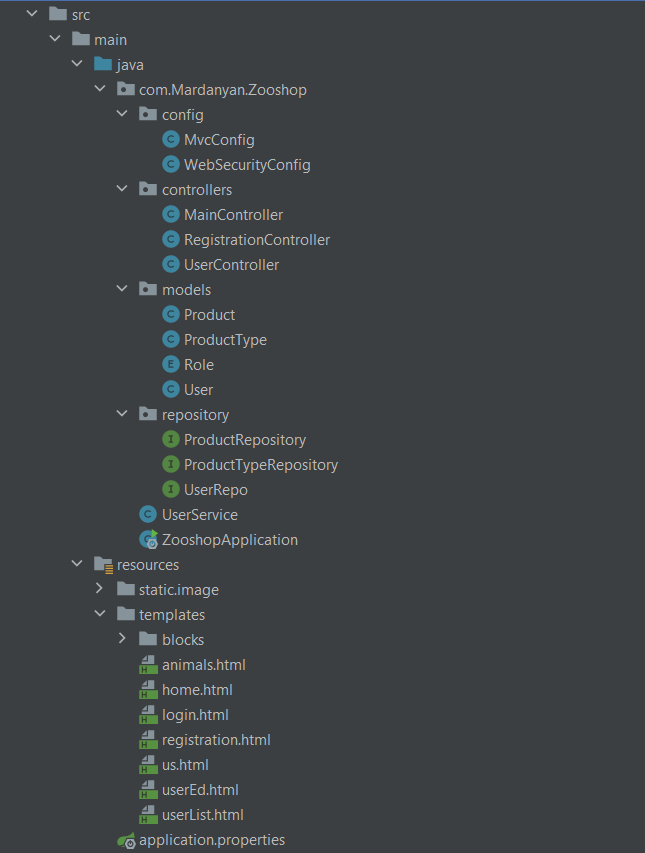


Рисунок 2.1 — структура программного продукта

package com.Mardanyan.Zooshop.controllers;  
  
import com.Mardanyan.Zooshop.models.Product;  
import com.Mardanyan.Zooshop.models.ProductType;  
import com.Mardanyan.Zooshop.repository.ProductRepository;  
import com.Mardanyan.Zooshop.repository.ProductTypeRepository;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Controller;  
import org.springframework.ui.Model;  
import org.springframework.web.bind.annotation.GetMapping;  
  
import java.util.HashMap;  
import java.util.List;  
import java.util.Map;  
  
@Controller  
public class MainController  
{  
 @Autowired  
 ProductRepository productRepository;  
  
 @Autowired  
 ProductTypeRepository productTypeRepository;  
  
 @GetMapping("/")  
 public String home(Model model)  
 {model.addAttribute("title", "Главная страница");return "home";}  
  
 @GetMapping("/login")  
 public String login(Model model)  
 {model.addAttribute("name","Авторизация");return "login";}  
  
 @GetMapping("/us")  
 public String us(Model model)  
 {model.addAttribute("name","О нас");return "us";}  
  
 @GetMapping("/animals")  
 public String animals(Model model)  
 {  
 Iterable<ProductType> types =productTypeRepository.findAll();  
 Map<ProductType, List<Product>> map = new HashMap<>();  
 types.forEach(type -> map.put(type, productRepository.findByProductType(type)));  
 model.addAttribute("map", map);  
 return "animals";  
 }  
}

Сверху можно увидеть код MainController. Здесь видно куда именно направляет каждая ссылка:

* Адрес “/” переходит на главную страницу. (см. Рис. 2.2)
* Адрес “/login” переводит на страницу авторизации. (см. Рис. 2.3)
* Адрес “/animals” переводит на страницу со всеми животными. (см. Рис. 2.4)

package com.Mardanyan.Zooshop.controllers;  
  
import com.Mardanyan.Zooshop.models.Role;  
import com.Mardanyan.Zooshop.models.User;  
import com.Mardanyan.Zooshop.repository.UserRepo;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Controller;  
import org.springframework.ui.Model;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.PostMapping;  
  
import java.util.Collections;  
  
@Controller  
public class RegistrationController  
{  
 @Autowired  
 private UserRepo userRepo;  
  
 @GetMapping("/registration")  
 public String registration() {  
 return "registration";  
 }  
  
 @PostMapping("/registration")  
 public String addUser(User user, Model model)  
 {  
 User userFromDb = userRepo.findByUsername(user.getUsername());  
 if (userFromDb != null)  
 {  
 model.addAttribute("message", "Такой пользователь уже есть!");  
 return "registration";  
 }  
 user.setActive(true);  
 user.setRoles(Collections.*singleton*(Role.*USER*));  
 userRepo.save(user);  
 return "redirect:/login";  
 }  
}

Сверху можно увидеть код RegisrtationController. Здесь видно где именно находится ссылка на регистрацию, а именно в “/registration”. Так же тут можно увидеть заметить защиту от создания двух аккаунтов с одинаковыми названиями.

(см. Рис. 2.5)

package com.Mardanyan.Zooshop.controllers;  
  
import com.Mardanyan.Zooshop.models.Role;  
import com.Mardanyan.Zooshop.models.User;  
import com.Mardanyan.Zooshop.repository.UserRepo;  
import org.springframework.stereotype.Controller;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.ui.Model;  
import org.springframework.web.bind.annotation.\*;  
  
import java.util.Arrays;  
import java.util.Map;  
import java.util.Set;  
import java.util.stream.Collectors;  
  
@Controller  
@RequestMapping("/admin/user")  
public class UserController  
{  
 @Autowired  
 private UserRepo userReposit;  
  
 @GetMapping  
 public String userList(Model model)  
 {  
 model.addAttribute("users", userReposit.findAll());  
 return "userList";  
 }  
  
 @GetMapping("{user}")  
 public String userEditForm(@PathVariable User user, Model model)  
 {  
 model.addAttribute("user", user);  
 model.addAttribute("roles", Role.*values*());  
  
 return "userEd";  
 }  
  
 @PostMapping  
 public String userSave(@RequestParam String username,  
 @RequestParam Map<String, String> form,  
 @RequestParam("userId") User user)  
 {  
 user.setUsername(username);  
  
 Set<String> roles = Arrays.*stream*(Role.*values*())  
 .map(Role::name)  
 .collect(Collectors.*toSet*());  
 user.getRoles().clear();  
 for (String key : form.keySet())  
 {  
 if (roles.contains(key)) {user.getRoles().add(Role.*valueOf*(key));}  
 }  
 userReposit.save(user);  
 return "redirect:/admin/user";  
 }  
  
 @GetMapping("/delete/{user}")  
 public String userDelete(@PathVariable("user") long id, Model model)  
 {  
 userReposit.deleteById(id);  
 model.addAttribute("user", userReposit.findAll());  
 return "redirect:/admin/user";  
 }  
}

UserController отвечает за страницу со всеми аккаунтами, где можно менять уровень пользователя или вовсе удалить его. (см. Рис 2.6-2.7)

Дальше представлены все модели программного продукта:

**Product:**

package com.Mardanyan.Zooshop.models;  
  
import lombok.Data;  
import org.hibernate.annotations.Type;  
  
import javax.persistence.\*;  
  
@Entity(name = "product")  
@Data  
public class Product  
{  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 private Long Id;  
 private String name;  
 private Double price;  
 private String image;  
  
 @Column(length = 65553)  
 @Type(type = "text")  
 private String country;  
  
 @Column(length = 65553)  
 @Type(type = "text")  
 private String weight;  
  
 @Column(length = 65553)  
 @Type(type = "text")  
 private String height;  
  
 @Column(length = 65553)  
 @Type(type = "text")  
 private String liveLong;  
  
 @OneToOne(cascade = CascadeType.*ALL*)  
 @JoinColumn(name = "product\_type\_id")  
 private ProductType productType;  
  
 public Product() {}  
  
 public Product(String name, Double price, String image)  
 {  
 this.name = name;  
 this.price = price;  
 this.image = image;  
 }  
  
 public ProductType getProductType() {  
 return productType;  
 }  
  
 public void setProductType(ProductType productType) {  
 this.productType = productType;  
 }  
}

**ProductType:**

@Entity(name = "product\_type")  
@Data  
public class ProductType  
{  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 private Long id;  
 private String name;  
 public ProductType(){}  
}

**Role:**

public enum Role implements GrantedAuthority  
{  
 *USER*,*ADMIN*;  
  
 @Override  
 public String getAuthority() {  
 return name();  
 }  
}

**User:**

@Entity  
@Table(name = "usr")  
@Data  
public class User implements UserDetails  
{  
 @Id  
 @GeneratedValue(strategy = GenerationType.*AUTO*)  
 private Long id;  
 private String username;  
 private String password;  
 private boolean active;  
  
 @ElementCollection(targetClass = Role.class, fetch = FetchType.*EAGER*)  
 @CollectionTable(name = "user\_role", joinColumns = @JoinColumn(name = "user\_id"))  
 @Enumerated(EnumType.*STRING*)  
 private Set<Role> roles;  
  
 @Override  
 public boolean isAccountNonExpired() {  
 return true;  
 }  
  
 @Override  
 public boolean isAccountNonLocked() {  
 return true;  
 }  
  
 @Override  
 public boolean isCredentialsNonExpired() {  
 return true;  
 }  
  
 @Override  
 public boolean isEnabled() {  
 return true;  
 }  
  
 @Override  
 public Collection<? extends GrantedAuthority> getAuthorities() {  
 return getRoles();  
 }  
}

Далее представлены фотографии самого программного продукта:

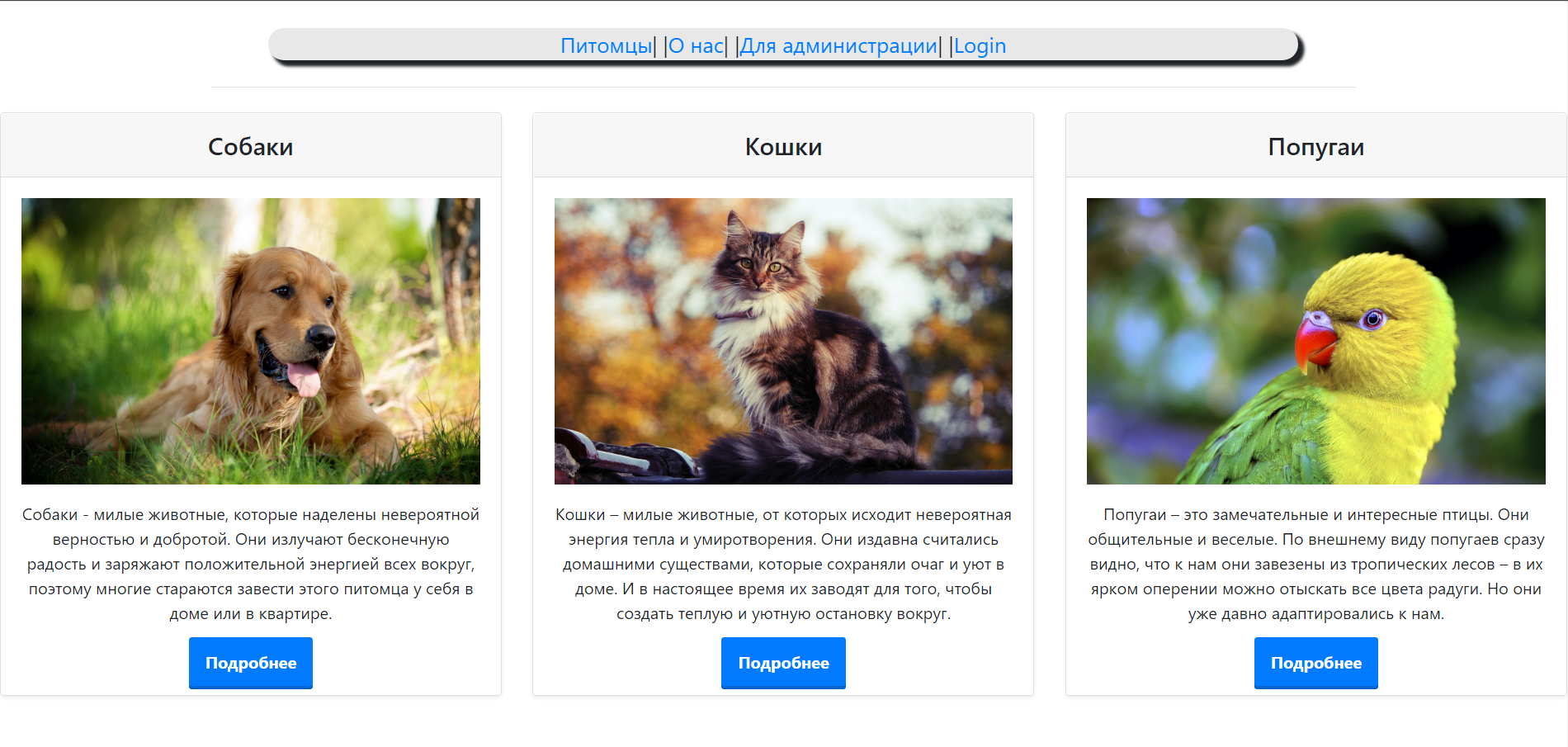


Рисунок 2.2 — главная страница

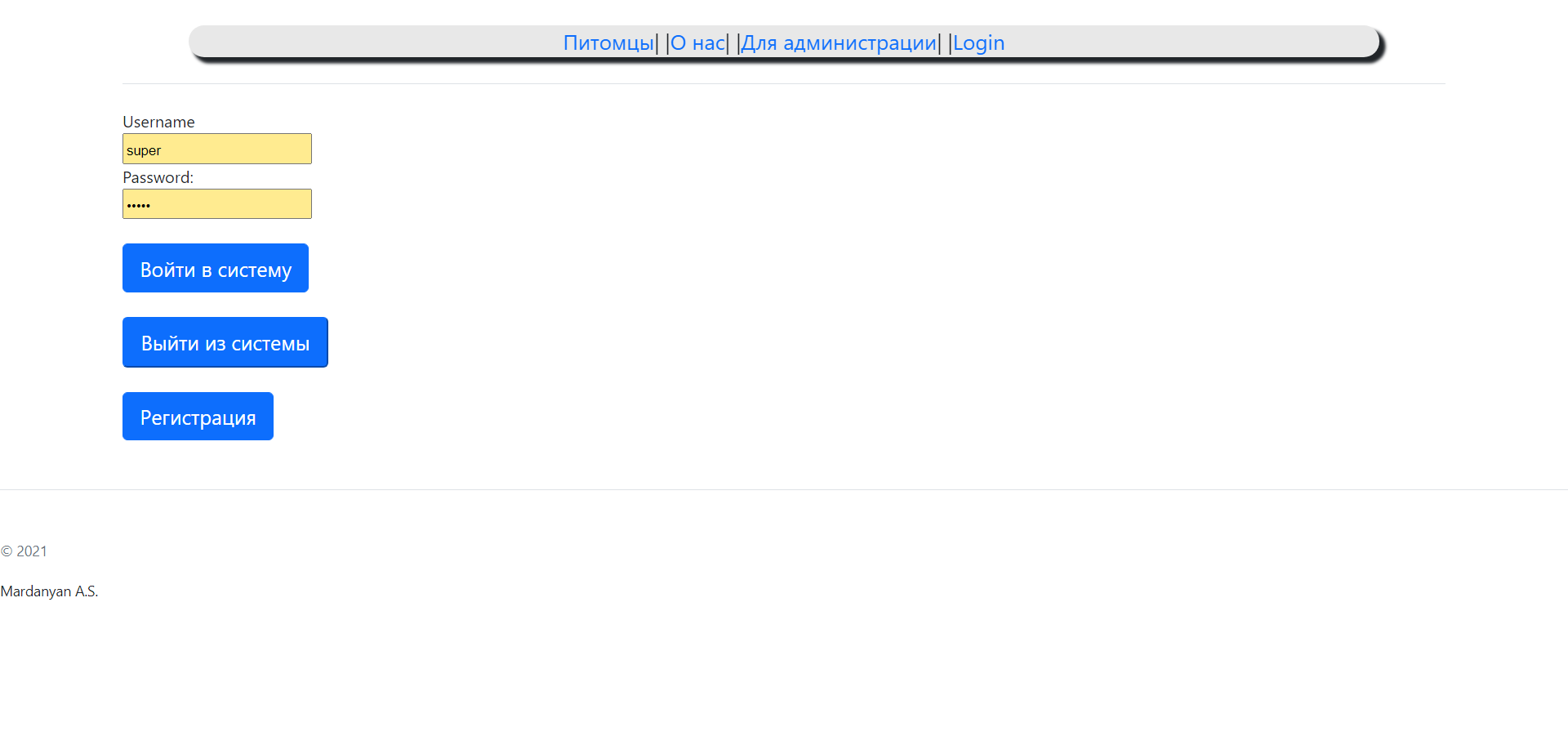


Рисунок 2.3 — страница авторизации

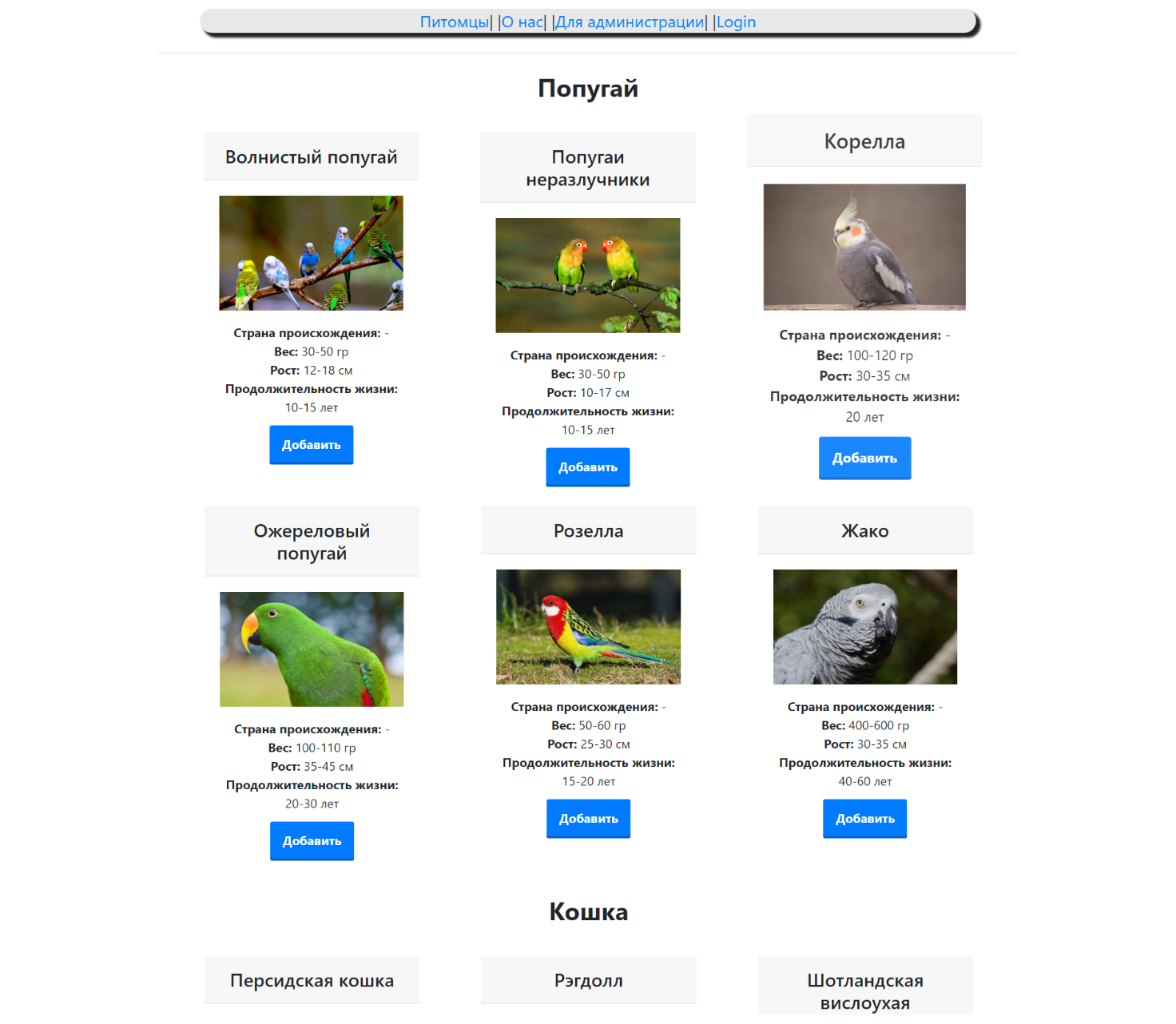


Рисунок 2.4 — страница с животными[13-15]

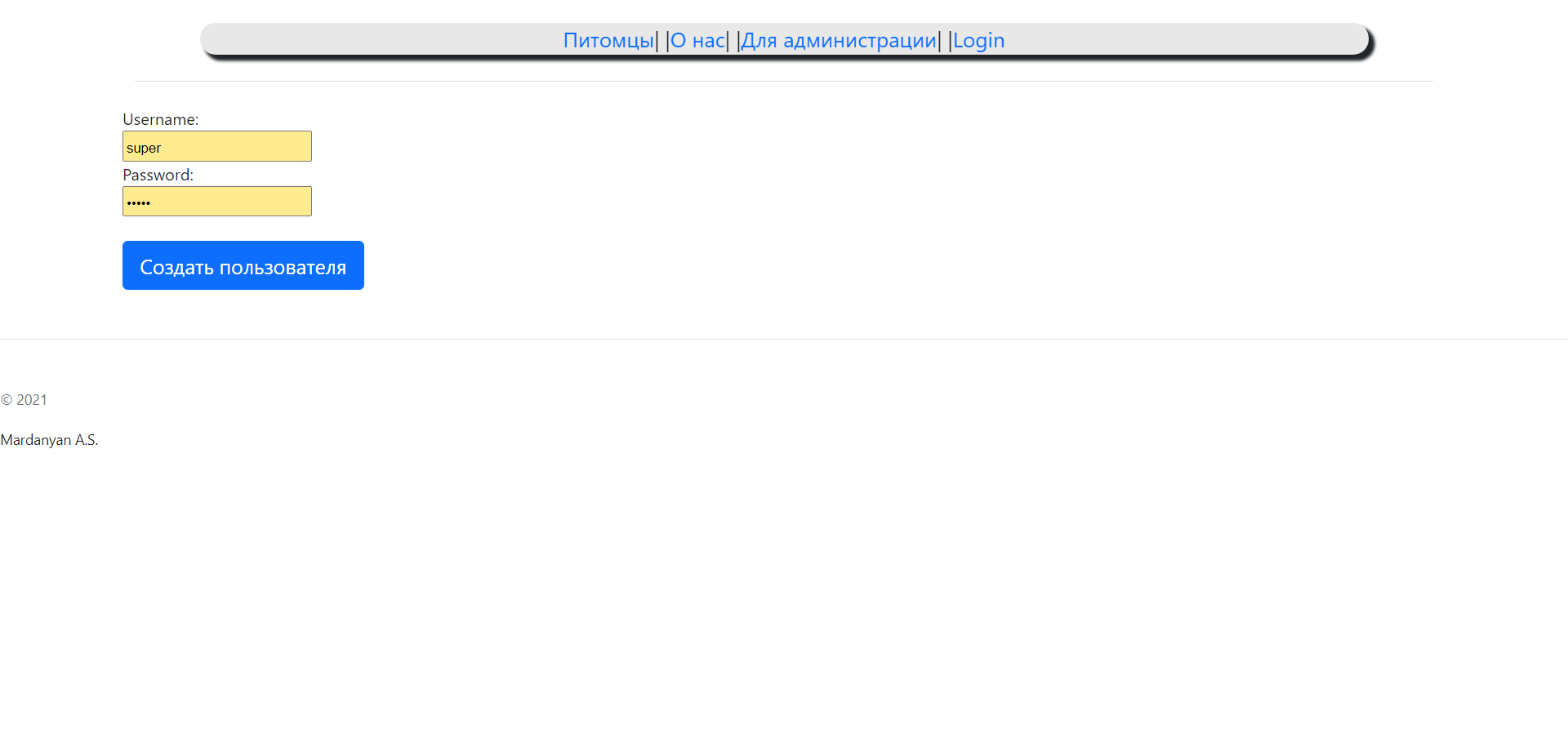


Рисунок 2.5 — страница регистрации

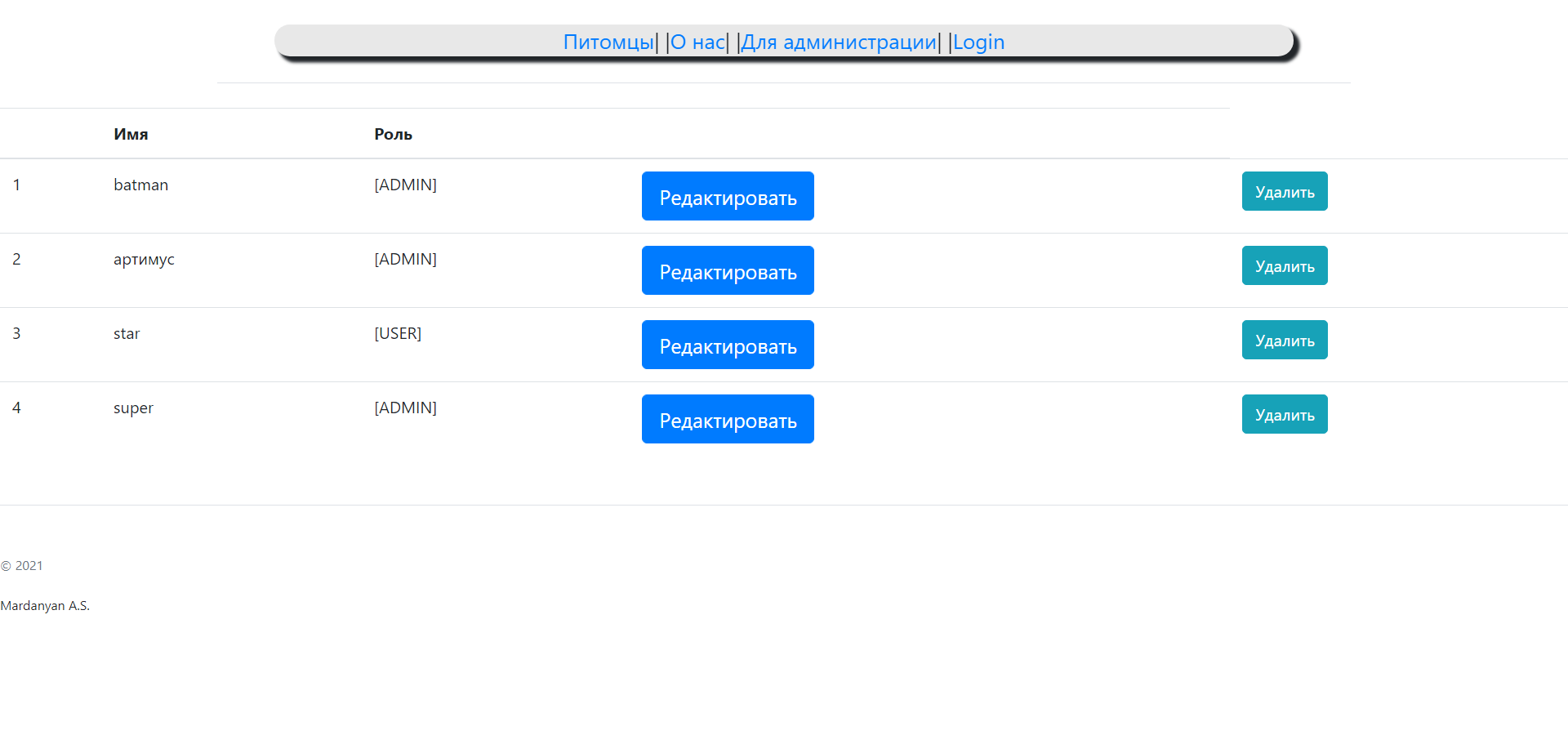


Рисунок 2.6 — страница администратора

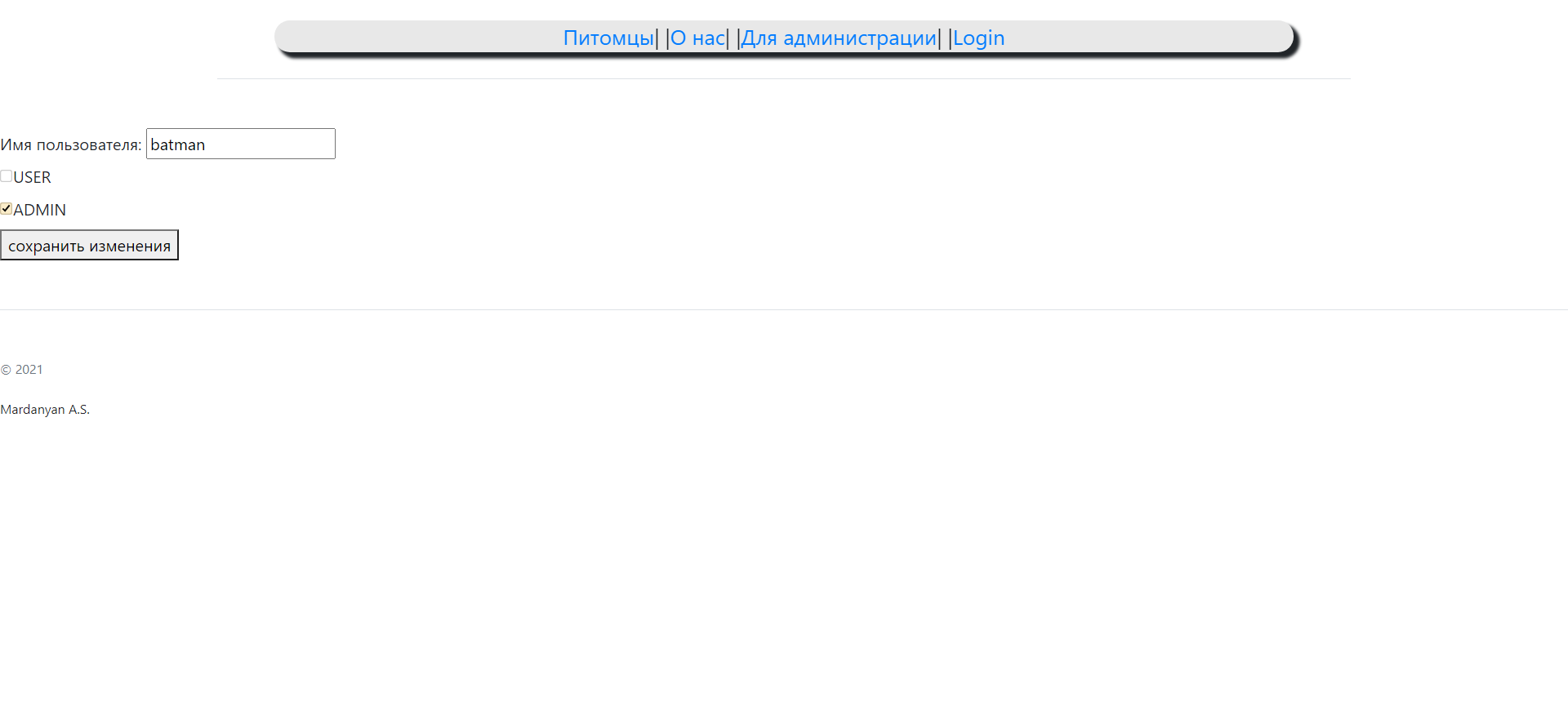


Рисунок 2.7 — страница редактирования пользователя

# Вывод

На основании всего вышесказанного можно сделать вывод, что все цели из первого раздела были выполнены. А также сумели понять, какие именно функции будут проходить тестирование.

# ТЕСТИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА

# ЦЕЛЬ

Целю тестирования является проверка работоспособности функционала программного продукта или его возможное улучшение.

# ОБЛАСТИ, ПОДТВЕРГАЕМЫЕ ТЕСТИРОВАНИЮ

* Регистрация пользователей
* Авторизация и выход пользователей
* Регистрация и удаление пользователей
* Проверка на ошибку в регистрации

# РЕАЛИЗАЦИЯ ТЕСТИРОВАНИЯ

Далее представление самого тестирования:[16]

@RunWith(SpringRunner.class)  
@SpringBootTest  
@AutoConfigureMockMvc  
class ZooshopApplicationTests  
{  
 @Autowired  
 private MockMvc mockMvc;  
  
 @Autowired  
 private MainController controller;  
  
 @Test  
 public void contextLoads() throws Exception  
 {this.mockMvc.perform(*get*("/"))  
 .andDo(*print*())  
 .andExpect(*status*().isOk())  
 .andExpect(*content*().string(*containsString*("Подробнее")));}  
  
 @Test  
 public void accessDeniedTest() throws Exception  
 {this.mockMvc.perform(*get*("/admin/user"))  
 .andDo(*print*())  
 .andExpect(*status*().is3xxRedirection())  
 .andExpect(*redirectedUrl*("http://localhost/login"));}  
  
 @Test  
 public void correctLoginTest() throws Exception  
 {  
 this.mockMvc.perform(SecurityMockMvcRequestBuilders.*formLogin*().user("super").password("super"))  
 .andDo(*print*())  
 .andExpect(*status*().is3xxRedirection())  
 .andExpect(*redirectedUrl*("/"));  
 }  
  
 @Test  
 public void badCredentials() throws Exception  
 {this.mockMvc.perform(*post*("/login").param("super","super"))  
 .andDo(*print*())  
 .andExpect(*status*().isForbidden());}  
}



Рисунок 3.1 — результаты тестирования

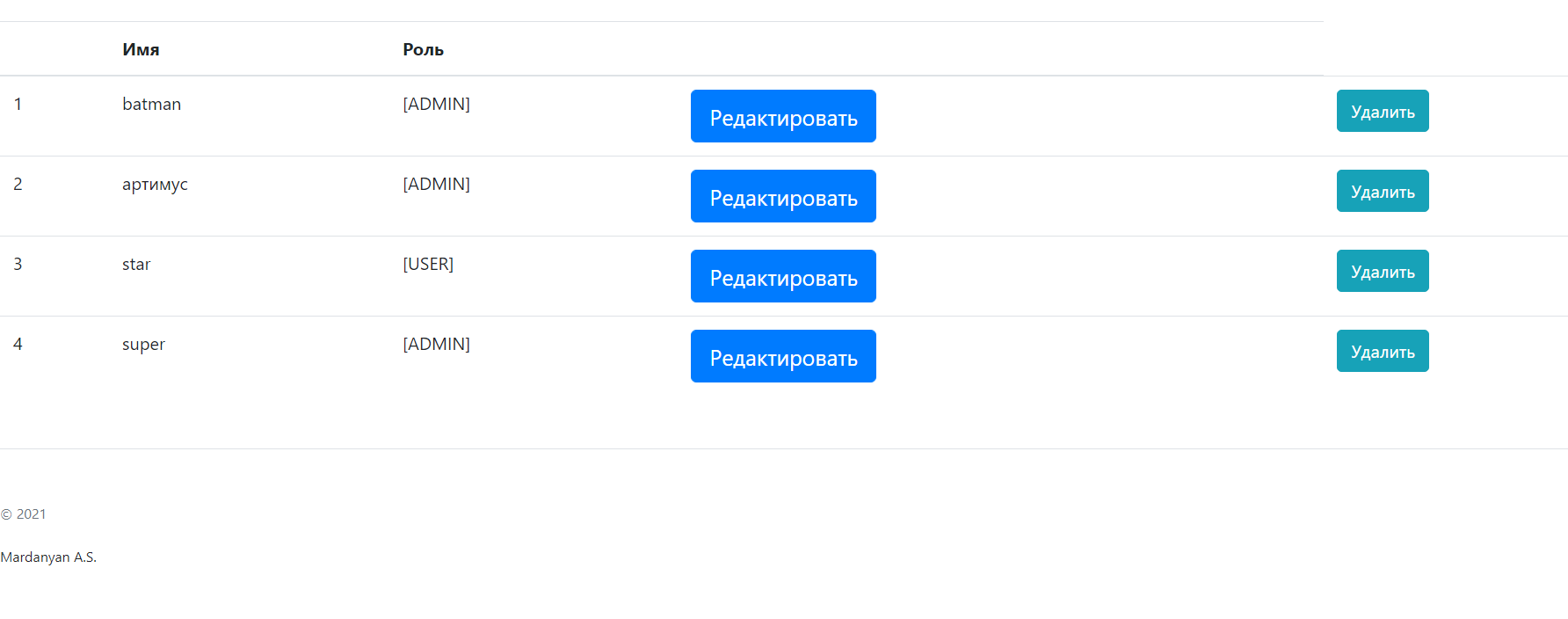


Рисунок 3.2 — до нового пользователя

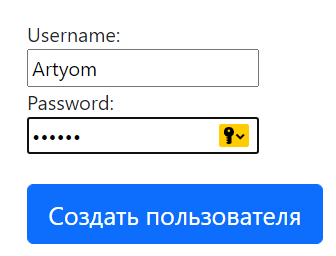


Рисунок 3.3 — создание нового пользователя

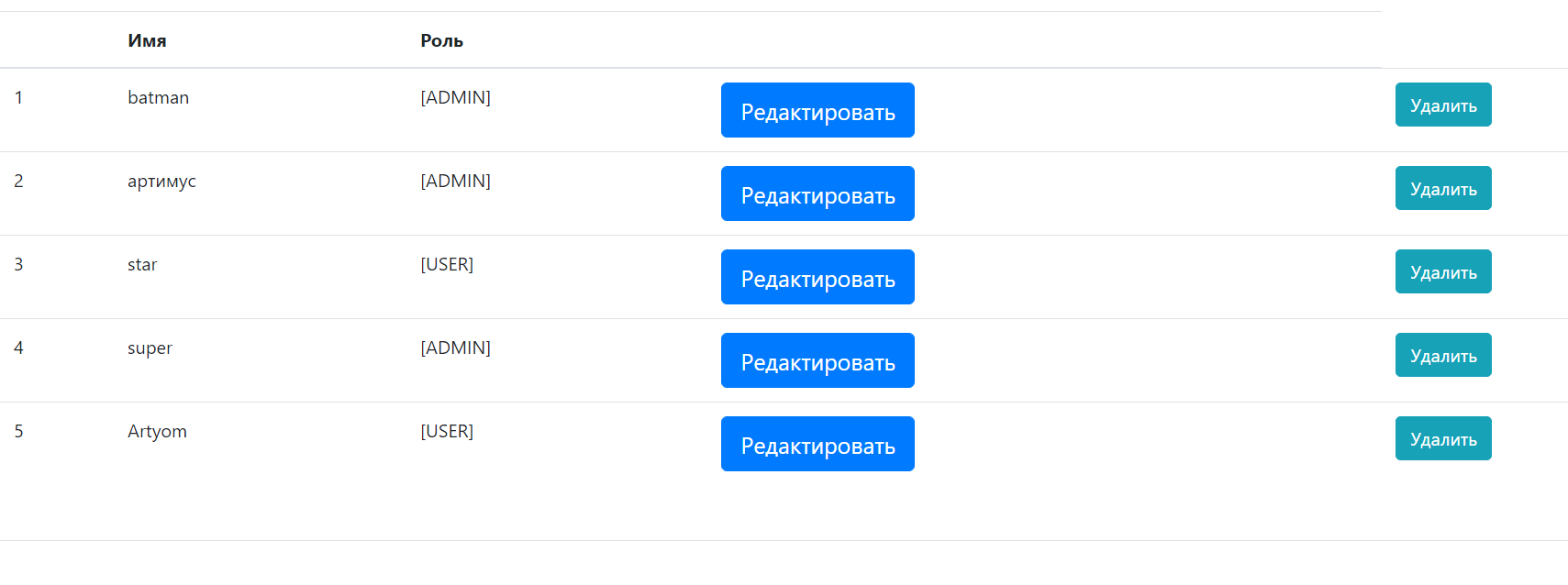


Рисунок 3.4 — новый пользователь в базе

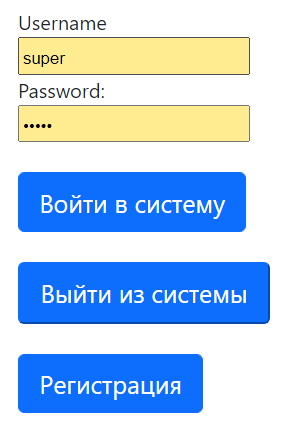


Рисунок 3.5 — авторизация

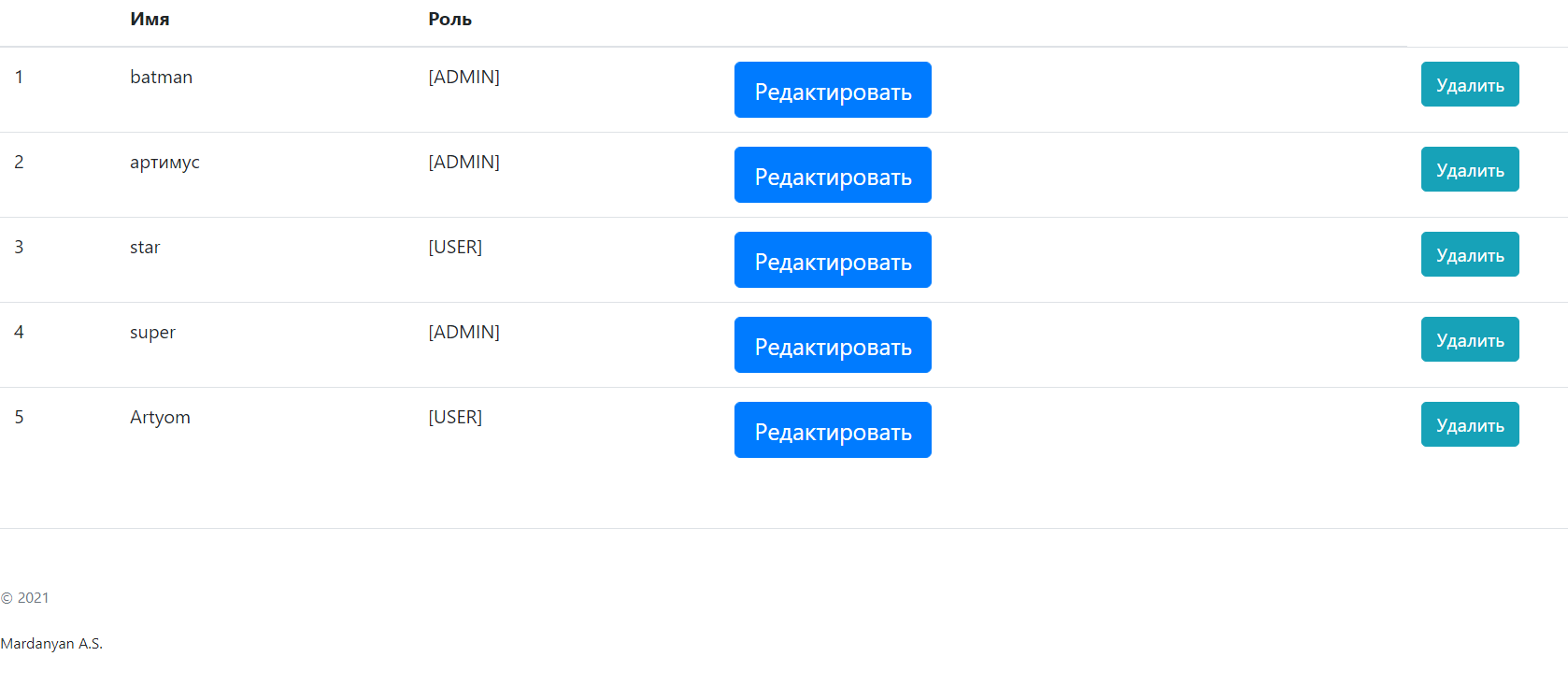


Рисунок 3.6 — до удаления

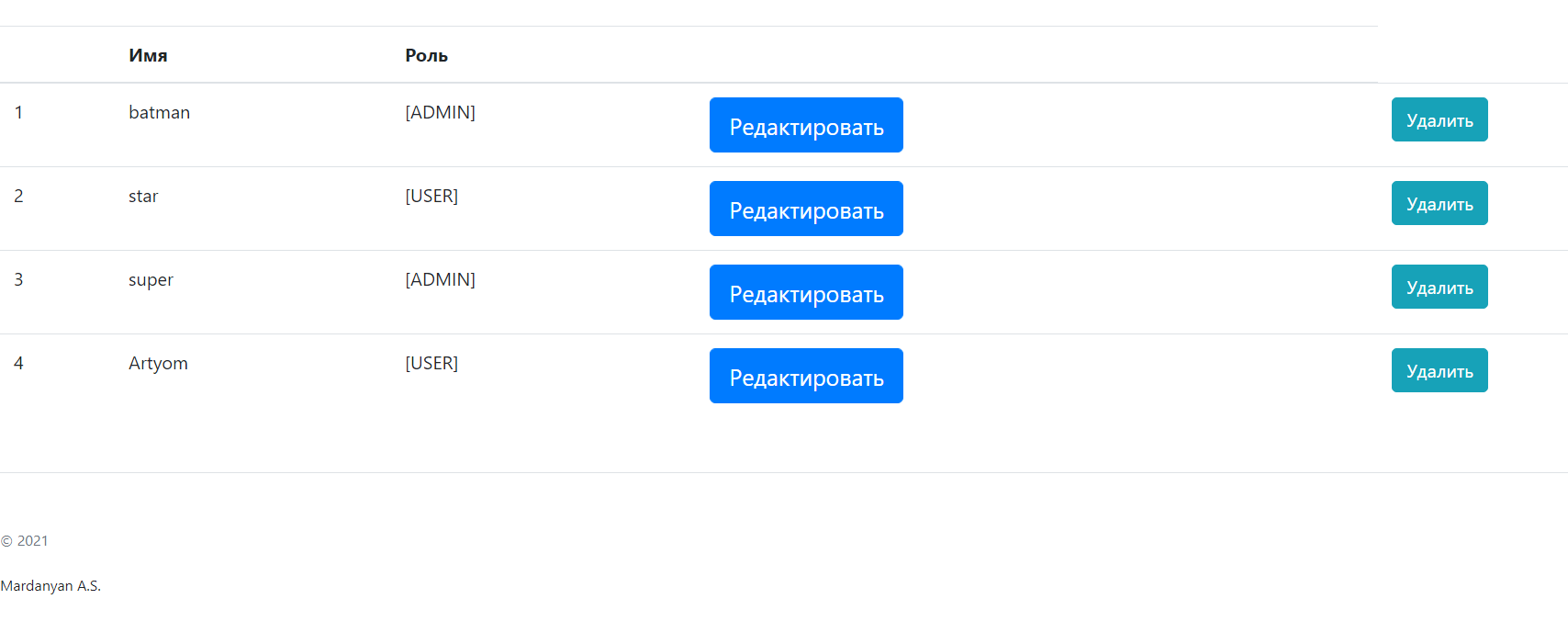


Рисунок 3.7 — после удаления пропал пользователь «артимус»

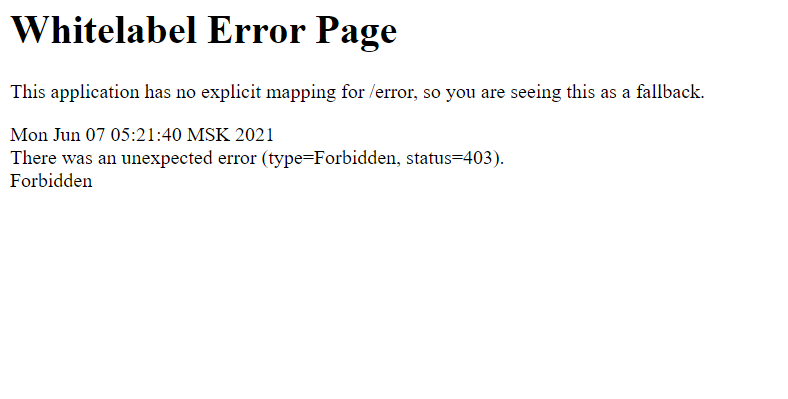


Рисунок 3.8 — ошибка при входе на страницу

администратора со страницы обычного пользователя

# Вывод

Были протестированы различные функции программного приложения, в ходе чего можно сделать вывод, что все эти функции работают.

Заключение

Во время выполнения данной курсовой работы были достигнуты все поставленные цели:

1. Были выбраны технологии разработки программного приложения
2. Создано программное приложение с учетом всех условий
3. Проведено тестирование всех функций программного приложения

В результате выполнения работы были получены навыки работы с Spring boot и MySQL.

Все требования к курсовой работе, указанные в техническом задании, реализованы в полной мере и работают без ошибок.

Github — URL: https://github.com/Mardanyan1/Mardanyan\_cursovaya

# Список информационных источников

1. Интернет-портал — URL: https://semantica.in/blog/chto-takoe-internet-portal.html#:~:text=Интернет-портал%20–%20это%20многофункциональная%20площадка,пользователям%20ссылок%20на%20другие%20сайты (Дата обращения: 15.03.2021). Текст электронный
2. Интернет-форум — URL: https://dic.academic.ru/dic.nsf/ruwiki/940938 (Дата обращения: 15.03.2021). Текст электронный
3. Социальная сеть— URL: https://wiki.rookee.ru/socialnaya-set/ (Дата обращения: 15.03.2021). Текст электронный
4. Интернет-магазин — URL: https://ru.wikipedia.org/wiki/Интернет-магазин (Дата обращения: 15.03.2021). Текст электронный
5. Бетховен: официальный сайт — URL: https://www.bethowen.ru/ (Дата обращения: 18.03.2021). Текст электронный
6. Samizoo: официальный сайт — URL: https://samizoo.ru/ (Дата обращения: 18.03.2021). Текст электронный
7. PetShop: официальный сайт — URL: <https://www.petshop.ru/> (Дата обращения: 18.03.2021). Текст электронный
8. Spring Boot: официальный сайт — URL: <https://spring.io/> (Дата обращения: 16.04.2021). Текст электронный
9. Видео-уроки Spring Boot: официальный сайт — URL: https://www.youtube.com/watch?v=FyZFK4LBjj0&t=3s (Дата обращения: 16.04.2021). Текст электронный
10. MySQL: официальный сайт — URL: https://www.mysql.com/ (Дата обращения: 17.04.2021). Текст электронный
11. Thymeleaf: официальный сайт — URL: https://www.thymeleaf.org/ (Дата обращения: 20.04.2021). Текст электронный
12. Controllers: официальный сайт — URL: https://spring.io/guides/gs/testing-web/ (Дата обращения: 21.04.2021). Текст электронный
13. Попугаи — URL: https://popugai--volnistye-ru.turbopages.org/popugai-volnistye.ru/s/popugai-domashnie-vidy-porody-dlya-doma/ (Дата обращения: 04.04.2021). Текст электронный
14. Собаки — URL: https://lapkins.ru/dog/ (Дата обращения: 04.04.2021). Текст электронный
15. Кошки — URL: https://lapkins.ru/cat/ (Дата обращения: 04.04.2021). Текст электронный
16. Видео-уроки для тестирования: официальный сайт — URL: https://www.youtube.com/watch?v=QrSR1fm9JwA&list=RDCMUC1g3kT0ZcSXt4\_ZyJOshKJQ&start\_radio=1&rv=QrSR1fm9JwA&t=5 (Дата обращения: 20.05.2021). Текст электронный