BMU

BAKI MÜHƏNDİSLİK UNİVERSİTETİ

# Pet Caring Application

Teachers name: Yusif Yusifov

Student’s name: Sama Baliyeva

Student id: 220106036

Specialization: Information technologies

Subject : OOP2

Course: 3

# Introduction to the PetPal App Frontend

**Purpose:**
The PetPal frontend application provides a seamless and user-friendly interface for pet owners. It allows users to manage their pets efficiently through a highly intuitive design. The app focuses on delivering a responsive and engaging experience while maintaining accessibility and usability.

**Key Features:**

- Interactive UI: Easy navigation for managing pets and viewing user-specific details.
- Error Feedback: Displays helpful messages for better user guidance.
- Visual Cues: Utilizes animations and intuitive layouts for enhanced usability.

# Authentication and Security in Frontend

**User Authentication Workflow:**

- Secure login integrated with backend JWT-based authentication.
- User roles (SuperAdmin, Admin, Regular User) dynamically adjust frontend access.

**Frontend Implementation Highlights:**

- Form Validation: Ensures correct data input using real-time feedback mechanisms.
- Secure Token Handling: Stores JWT securely in SharedPreferences for authenticated API calls.
- Role-Based Views: Dynamically renders features based on the user's role embedded in the JWT.

# User Interaction and Features

**Home Page:**

- Displays a personalized welcome message and a list of registered pets.
- Interactive buttons for viewing, editing, or deleting pet details.

**Pet Registration:**

- User-friendly forms to add pet details, such as name, breed, and age.
- Drop-down options for selecting the pet's gender.

**Update Pet Details:**

- Allows users to modify pet details with pre-filled forms for easy updates.
- Real-time validation ensures no incomplete or incorrect data is submitted.

**Error Handling:**

- Centralized error-handling framework to catch issues like invalid inputs or server errors.
- Toast messages and dialog boxes provide immediate feedback to the user.

# Frontend Tech Stack

**Programming Language:**

- Kotlin: Provides concise and robust code for Android development.

**UI Components:**

- XML-based layouts for designing intuitive interfaces.
- Material Design principles for consistent and visually appealing UI elements.

**State Management:**

- ViewModel and LiveData for observing and managing application states.

**API Integration:**

- Retrofit for seamless communication with the backend APIs.
- Swagger for understanding backend endpoints and ensuring compatibility.

# Challenges & Solutions

**Challenge 1: Creating Role-Based Views**

Solution:

Dynamically render UI components based on the role extracted from JWT.

**Challenge 2: Handling API Errors Gracefully**

Solution:

Integrated a centralized error-handler that uses ApiErrorHandler to map backend error codes to user-friendly messages.

# Future Enhancements

**Improved User Experience:**

- Introduce animations and transitions for smoother interactions.

**Offline Functionality:**

- Add support for offline pet management with local storage, syncing changes once online.

**Cross-Platform Expansion:**

- Develop a web-based version of the app to complement the Android version.

# Thank you!