**VISUAL PROGRAMMING**

**PROJECT DOCUMENTATION**

**PET ADOPTION WEB APP**

**PRESENTED TO:**

**Ms. Farkhanda Zafar**

**PRESENTED BY:**

**Javeria Rehman – 210976**

**Mah Rukh - 210910**

# TABLE OF CONTENTS

1. **Project Summary**

# Key Project Features

1. **Project Backend Summary**

# Project Backend Queries Details

# Front-End

## PROJECT SUMMARY

Our “Pet Planet” focuses on the target audience, which are the people who want to help stray animals so that they can have a loving home. This Web App has a main goal of making it easier for adoptees to look through an available database of adoptable pets while also allowing the caretakers and shelters of those pets maintain the list of the animals within their care.

Our further goals also include making this a reliable platform for adoptees and shelters. The application provides all the necessary details one adoptee wishes to know about the pet they wish to adopt and at the same time the shelters can make sure the adoptee is reliable enough to have the pet in their future care.

It makes it easier for the adoptees to traverse a wide range of databases especially if they do not have the help they need to adopt a pet or any people to get help from.

## The Pet Planet is a user-friendly and intuitive web application aimed at facilitating the adoption process for individuals seeking to find a new pet. The application provides a convenient platform that connects potential pet owners with animal shelters, rescue organizations, and pet foster homes.

* 1. **KEY PROJECT FEATURES**

1. User Registration and Profile Creation: Users can create their accounts and set up profiles with information.
2. Pet Listings and Search: The application provides a comprehensive database of available pets for adoption, including dogs, cats, and other small animals. Users can search and filter the listings based on their preferences to find pets that match their criteria.
3. Detailed Pet Profiles: Each pet has a dedicated profile page with information about their breed, age, health condition, and photos. Users can view this information to get a better understanding of the pet's suitability for adoption.
4. Adoption Application and Approval: Users can submit adoption applications directly through the application. The application process may involve filling out forms, providing references, and agreeing to adoption terms.
5. Add Pet: Users can also add any pet into the database which they want to let others adopt. Users can perform crud operations here like adding a new pet, updating the info and also they can delete the pet if it is no longer available.
6. Live Safe: You can also look at the nearby shelters or clinics from your location to go for the checkup of your pet. It is done using google map API integration.

7. Additional Features: The application may incorporate additional features such as pet care guides, veterinarian directories, vaccination reminders, and integration with social media platforms for sharing pet profiles.

**Benefits:**

* Convenience: The web app simplifies the pet adoption process by providing a centralized platform for searching, and applying with shelters and organizations.
* Increased Adoption Rates: By making it easier for potential pet owners to find suitable pets, the application can contribute to increased adoption rates and help more animals find loving homes.
* Improved User Experience: The user-friendly interface and intuitive design of the application enhance the overall experience for both potential pet owners and shelters/organizations.

Overall, the pet adoption web application aims to bridge the gap between individuals looking to adopt a pet and organizations providing rescue and adoption services. By leveraging technology, the application simplifies the adoption process, facilitates communication, and promotes successful pet adoptions, ultimately improving the lives of both animals and their owners.

Also, this kind of application is not currently available in Pakistan although OLX has the functionality but it is not completely based on Pet Adoption. Our project aims at building this gap.

## 3. PROJECT BACKEND SUMMARY

Pet Adoption Web Application utilizing SQL Database:

The Pet Adoption Web Application utilizes a SQL Database as its backend solution to manage and store data related to pet listings, user profiles, adoption applications, and other pertinent information. The SQL Database provides a flexible, scalable, and real-time environment for efficient data management within the application.

**Data Organization:**

1. Tables and Records: The SQL Database organizes data into tables and records, mirroring the functionality of collections and documents in Firestore.
2. Pet Listings Table: Each pet listing is stored as a record within the “Pet” table. This table contains attributes such as breed, age, gender, location, name, weight, and images for each pet. Real-time synchronization ensures that pet listings are updated consistently, providing users with the most current information.
3. Owner Profiles Table: Owner profiles, encompassing registration details like 'user-name' and 'password' are stored as records within the “Users” table. Each profile has a unique record associated with it, simplifying access to their pet data.
4. Adoption Applications Table: When a user submits an adoption application, the application data is stored as a separate record within the "Applications" table. This facilitates efficient review, status tracking, and communication between shelters/organizations and applicants.
5. Shelter Information Table: This table stores URLs of shelters, aiding in displaying shelter locations on Google Maps during runtime.

**Core Functionalities:**

1. Real-time Updates:

Utilizing triggers and stored procedures, the SQL Database ensures real-time synchronization across devices. Any modifications made to records are instantly propagated to all connected clients, guaranteeing users access to up-to-date information.

2. Scalability and Performance:

The SQL Database offers scalability through efficient indexing and load balancing strategies. It distributes the workload across multiple servers to ensure high performance and responsiveness, even during periods of increased traffic and data growth.

3. Security and Permissions:

Robust security measures are implemented within the SQL Database to safeguard data integrity and prevent unauthorized access. Defined access control rules ensure that only authenticated users can read and modify data within specific tables and records.

4. Integration with Backend Services:

The SQL Database seamlessly integrates with other backend services and APIs for various functionalities. For instance, integration with authentication services enables user authorization, while backend logic implemented via server-side scripts handles specific database events.

5. Offline Support:

The SQL Database provides mechanisms to support offline functionality within the application. Changes made while offline are queued and synchronized with the database once the user reconnects, ensuring consistency and data integrity.

6. Monitoring and Analytics:

Though SQL databases might not inherently provide analytics, external monitoring tools can be employed to track database performance, user behavior, and error tracking, contributing to optimization and enhancement of the application's user experience.

**Code Organization:**

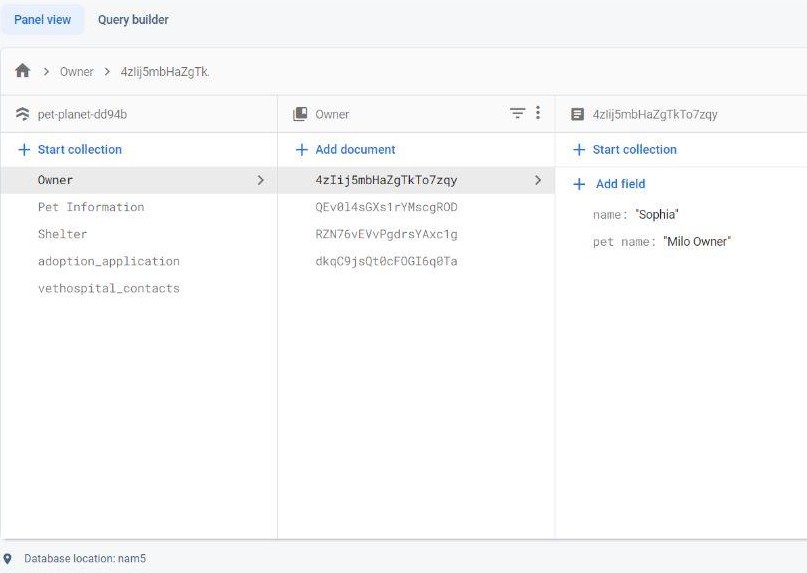
1. MVC Architecture:

We’ve used the MVC architecture to build this website. We’ve separated the backend database. The model classes are separate, the services are separate, we’ve also kept the API integration separate from all of this as well.

In the Blazor pages, we’ve used razor components to make our code clean and understandable while keeping the HTML and CSS code separate.

We’ve also passed parameters between different Blazor pages.

In conclusion, the utilization of a SQL Database as the backend solution for the Pet Adoption Web Application offers a robust, scalable, and efficient data storage environment. It empowers the application with real-time updates, secure data access, seamless integration capabilities, and the ability to handle diverse functionalities, contributing significantly to the success and performance of the pet adoption platform.



## Project Backend Queries Details

Here are the backend queries used to retrieve the data from firestore to the application:

* 1. This query will get all the data that is stored about the pet such as its name, location, age etc.

db.collection("Pet Information")

.get()

.then((querySnapshot) => { querySnapshot.forEach((doc) => { console.log(doc.id, " => ", doc.data());

});

})

.catch((error) => {

console.log("Error getting documents: ", error);

});

* 1. This query will only get the important information about the shelters such as its location and its contact information as well as the visit hours or the contact and name of the vet hospitals.

db.collection("vethospital\_contacts")

.get()

.then((querySnapshot) => { querySnapshot.forEach((doc) => { const { breed, age } = doc.data();

console.log("name:", name, "phone number:", phonenumber);

});

})

.catch((error) => {

console.log("Error getting documents: ", error);

});

* 1. This query will get the data after the user filters their requirement of the pet such as limiting the search by breed or color etc.

db.collection("Pet Information")

.where("breed", "==", "Persian")

.get()

.then((querySnapshot) => { querySnapshot.forEach((doc) => { console.log(doc.id, " => ", doc.data());

});

})

.catch((error) => {

console.log("Error getting documents: ", error);

});

* 1. This query will limit the number of pet data displayed to the user on the cards for swiping such as the pet nearest to the user in terms of location.

1. **FRONT – END:**