

Pet Adoption Database Application Documentation

Business Problem

Animal shelters face challenges in managing their operations efficiently, particularly in tracking available pets, adoption applications, and pet health records. This inefficiency often leads to delays in the adoption process and ineffective resource management. The Pet Adoption Database Application addresses these challenges by providing a comprehensive solution to streamline operations and improve the adoption experience.

Executive Summary

The Pet Adoption Database Application is a user-friendly web-based platform that simplifies pet adoption processes for animal shelters. The application allows users to browse available pets, submit adoption applications, and track their status. Shelters can manage their pets, applications, and health records efficiently through CRUD operations and generate detailed reports. Built with Python (Flask) and MySQL, the application integrates a robust backend with an intuitive frontend interface.

Technical Specifications

- Languages/Technologies Used:

- Frontend: HTML templates rendered via Flask.
- Backend: Python (Flask framework).
- Database: MySQL.
- Libraries: ``mysql.connector``, ``pandas``, ``flask``.

- System Requirements:

- Python 3.8 or higher.
- MySQL server.
- Required Python libraries installed via:

```
pip install flask mysql-connector-python pandas
```

- Setup Steps:

1. Run ``create_DB.py`` to initialize the MySQL database.
2. Populate tables with initial data (via SQL scripts or manual input).
3. Start the Flask server using ``python app.py``.
4. Access the application in a web browser at ``http://localhost:5000``.

Source Files Overview

1. `create_DB.py`:

- Creates the `PetAdoptionDB` MySQL database.
- Establishes an initial connection to the MySQL server.
- Ensures the database is properly set up for the application.

2. `app.py`:

- Main application file for running the Flask web server.
- Handles user interactions through routes and templates.
- Performs CRUD operations and generates reports.

Source File Interconnectivity

- `create_DB.py` initializes the database structure required for the application.
- `app.py` connects to the database using the `mysql.connector` library and defines routes to manage pets, shelters, adoption applications, and health records.
- HTML templates render data fetched from the database for user interaction, ensuring a seamless frontend experience.

Database Design

1. Tables:

- `Pets`:
Tracks pet information (e.g., `pet_id`, `name`, `species`, `status`, `shelter_id`).
- `Shelters`:
Stores shelter details (e.g., `shelter_id`, `location`, `contact_info`).
- `Adoption_Applications`:
Manages adoption application statuses (`application_id`, `user_id`, `pet_id`, `status`).
- `Users`:
Manages user credentials and contact information.
- `Pet_Health_Record`:
Maintains a health history for pets (`record_id`, `pet_id`, `checkup_date`, `vaccinations`).

2. Relationships:

- `Pets` is linked to `Shelters` through `shelter_id` (Many-to-One).

- ``Adoption_Applications`` is linked to ``Users`` and ``Pets`` through ``user_id`` and ``pet_id`` (Many-to-One).
- ``Pet_Health_Record`` links to ``Pets`` through ``pet_id`` (Many-to-One).

3. Design Choices:

- Normalized structure to prevent redundancy.
- Foreign keys to maintain referential integrity.

Database Utilization by the Application

- CRUD Operations:
 - Pets, shelters, and applications are managed through SQL queries.
 - Users can add, update, delete, and view data through the web interface.
- Dynamic Filtering:
 - Filters (e.g., species, shelter) enable users to refine their search for pets.
- Report Generation:
 - ``pandas`` library exports filtered data into Excel reports.

How to Run the Application

1. Setup Database:

- Ensure MySQL server is running.
- Execute ``python create_DB.py`` to create the ``PetAdoptionDB`` database.

2. Start the Application:

- Run ``python app.py`` to start the Flask server.
- Open a browser and navigate to ``http://localhost:5000``.

3. Using the Application:

- Browse Pets: Use the ``/pets`` route to view available pets.
- Manage Data: Add, update, or delete pets and shelters through their respective routes.
- Apply for Adoption: Submit applications via ``/apply_adoption``.
- View Reports: Generate Excel reports for available pets.

Future Enhancements

- User Authentication: Introduce roles for administrators and regular users.
- Advanced Analytics: Integrate graphical reports for better data insights.
- Mobile Compatibility: Enhance the UI for responsive design.
- Notifications: Enable email updates for adoption status changes.