Austin Animal Shelter Dashboard

CS-499 Computer Science Capstone - Category Three: Databases Enhancement

Author: Ericka Resendez

Project Overview

This dashboard helps the Grazioso Salvare animal rescue organization identify and filter dogs suitable for search and rescue training. The application features user authentication, role-based access control, and advanced analytics using MongoDB aggregation pipelines.

Enhancement Details

Original Artifact: CS-340 Grazioso Salvare Dashboard
Enhancement Focus: Database security and advanced analytics

Key Enhancements:

1. **JWT Authentication** - Secure token-based authentication with password hashing (bcrypt)
2. **Role-Based Access Control** - Three user roles: Admin, Analyst, and Viewer
3. **MongoDB Aggregation Pipelines** - Complex data analytics for rescue eligibility
4. **Audit Logging** - Tracks all database operations for security compliance
5. **User Management** - Admin interface for creating and managing users

Prerequisites

Required Software:

- **Python 3.8+**
- **MongoDB 4.4+**
- **Web Browser** (Chrome, Firefox, Safari, or Edge)

Required Python Packages:

pip install pymongo pandas dash dash-leaflet plotly bcrypt pyjwt

Installation Instructions

Step 1: Install MongoDB

macOS:

brew install mongodb-community
brew services start mongodb-community

Windows: Download and install from: https://www.mongodb.com/try/download/community

Linux:

sudo apt-get install mongodb
sudo systemctl start mongodb

Step 2: Clone or Download Project Files

Ensure you have these files in your project directory:

- Artifact_Three_animal_shelter_CRUD.py - Enhanced CRUD module
- ArtifiactThreeDashboard.ipynb - Dashboard application
- aac_shelter_outcomes.csv - Animal shelter data
- README.md - This file

Step 3: Set Up the Database

Run the setup script to import data and create users:

python setup_database.py

This will:

- Import the CSV data into MongoDB
- Create default user accounts
- Set up database indexes
- Verify the installation

Step 4: Run the Dashboard

Option A: Using Jupyter Notebook

jupyter notebook
Open ArtifiactThreeDashboard.ipynb and run all cells

Option B: Convert and run as Python script

jupyter nbconvert --to python ArtifiactThreeDashboard.ipynb
python ArtifiactThreeDashboard.py

Step 5: Access the Dashboard

Open your web browser and navigate to:

http://127.0.0.1:8050

Default User Credentials

Username	Password	Role	Permissions
admin	admin234	Admin	Full access + user management
analyst	analyst456	Analyst	Analytics and data viewing
user	user123	Viewer	Read-only access

Features Demonstration

1. Authentication System

- Login with username and password
- JWT token generation and validation
- Automatic token expiration (24 hours)
- Role-based interface customization

2. User Management (Admin Only)

- Create new users with specific roles
- Deactivate user accounts
- View all users in the system
- Password hashing for security

3. Advanced Analytics

- **Water Rescue Analytics** - Identifies dogs eligible for water rescue training
- **Wilderness Rescue Analytics** - Filters dogs for mountain/wilderness operations
- **Disaster Rescue Analytics** - Finds dogs suitable for disaster response
- **Breed Performance Metrics** - Aggregated adoption and success rates by breed
- **Monthly Adoption Trends** - Time-series analysis of adoption patterns
- **Demographics Analysis** - Animal type distribution and statistics

4. Interactive Data Table

- Filter by rescue type (Water, Wilderness, Disaster)
- Sort by any column
- Pagination for large datasets
- Search functionality

5. Geolocation Mapping

- Interactive map showing animal locations
- Click markers to view animal details
- Centered on Austin, Texas

Database Schema

Animals Collection

```
{
  animal_id: String,
  animal_type: String,          // "Dog", "Cat", etc.
  breed: String,
  name: String,
  age_upon_outcome_in_weeks: Number,
  sex_upon_outcome: String,     // "Intact Male", "Spayed Female", etc.
  outcome_type: String,         // "Adoption", "Transfer", etc.
  datetime: Date,
  location_lat: Number,
  location_long: Number
}
```

Users Collection

```
{
  username: String,
  password: String,             // bcrypt hashed
  role: String,                 // "admin", "analyst", "viewer"
  email: String,
  created_at: Date,
  active: Boolean
}
```

Audit Logs Collection

```
{
  username: String,
  action: String,               // "LOGIN_SUCCESS", "CREATE_RECORD", etc.
  details: String,
  timestamp: Date,
  ip_address: String
}
```

MongoDB Aggregation Pipeline Examples

Breed Performance Metrics

```
db.animals.aggregate([
  { $match: { animal_type: "Dog", outcome_type: { $in: ["Adoption", "Transfer"] }}},
  { $group: {
      _id: "$breed",
```

```
    total_animals: { $sum: 1 },
    adoption_count: { $sum: { $cond: [{ $eq: ["$outcome_type", "Adoption"]}, 1, 0]}}
  }},
  { $project: {
    breed: "$_id",
    adoption_rate: { $multiply: [{ $divide: ["$adoption_count", "$total_animals"]}, 100]}
  }},
  { $sort: { adoption_rate: -1 }}
])
```

Rescue Type Analytics

```
// Water Rescue Eligibility
db.animals.aggregate([
  { $match: {
    animal_type: "Dog",
    breed: { $in: ["Labrador Retriever Mix", "Chesapeake Bay Retriever", "Newfoundland"]},
    sex_upon_outcome: "Intact Female"
  }},
  { $group: {
    _id: "$breed",
    count: { $sum: 1 },
    avg_age_weeks: { $avg: "$age_upon_outcome_in_weeks" }
  }}
])
```

Files Structure

```
project/
├── Artifact_Three_animal_shelter_CRUD.py    # CRUD module
├── ArtifiactThreeDashboard.ipynb            # Main dashboard application
├── aac_shelter_outcomes.csv                 # Source data
└── Grazioso-Salvare-Logo.png                    # Logo
```

Last Updated: October 2025
