**MVC Exercise: PetsApp**

**Objective:** Create a simple MVC application to manage pets, allowing users to register, login, and perform CRUD (Create, Read, Update, Delete) operations on their pet profiles.

**Scenario**

You are tasked with developing a PetsApp where users can register, log in, and manage their pets. Each user can have multiple pets, and each pet has attributes such as name, age, breed, and owner's details. The application should support the following functionalities:

1. **User Registration and Login:**
   * Users should be able to register by providing their name, email, and password.
   * Implement login functionality where users can log in using their email and password.
   * Use hashed passwords for security.
2. **Pet Management:**
   * Users should be able to add new pets with details such as name, age, breed, and photo.
   * Users should be able to view a list of their pets.
   * Users should be able to update pet details.
   * Users should be able to delete a pet from their profile.
3. **Profile Management:**
   * Users should be able to view and update their profile information.
   * Users should be able to change their password.

**Requirements**

1. **Models:**
   * Create a User model with properties: UserId, Name, Email, Password, and ProfilePhoto.
   * Create a Pet model with properties: PetId, Name, Age, Breed, Photo, and UserId (foreign key to User).
2. **Controllers:**
   * AccountController for user registration, login, and profile management.
   * PetController for managing pets.
3. **Views:**
   * Registration and Login views.
   * Profile view to display and update user information.
   * Pet views to display, add, update, and delete pets.
4. **Services and Repositories:**
   * Implement a service layer for business logic.
   * Implement a repository layer for data access.
5. **Security:**
   * Implement password hashing for user passwords.
   * Ensure authenticated users can only access their own pets and profile information.
6. **Validation:**
   * Implement server-side validation for all forms.

**Tasks**

1. **Create the Models:**
   * Define the User and Pet models with the appropriate properties and data annotations for validation.
2. **Implement Registration and Login:**
   * Create the registration and login views and actions in the AccountController.
   * Implement password hashing for secure storage.
3. **Profile Management:**
   * Create actions and views to display and update user profile information.
   * Implement functionality to change the password.
4. **Pet Management:**
   * Create the PetController with actions to add, view, update, and delete pets.
   * Create views for managing pets.
5. **Authorization:**
   * Ensure that only authenticated users can access the pet management and profile management features.
6. **Data Access Layer:**
   * Implement a repository pattern for accessing user and pet data.
   * Use Entity Framework (or any other ORM) to interact with the database.
7. **Service Layer:**
   * Implement a service layer to handle business logic for user and pet management.
8. **Validation and Error Handling:**
   * Add validation to ensure that all required fields are provided and valid.
   * Implement error handling to display appropriate error messages to users.

**Deliverables**

* Complete MVC application with the functionality described above.
* Source code organized in a Git repository.
* Documentation describing how to set up and run the application.