

Implementation Plan Document: Animal Shelter Database

Kriti Bharadwaj (PES1UG22CS295)

Keya Kesani (PES1UG22CS285)

September 30, 2024

1 Tech Stack

1.1 Backend

- **Languages:** Python (FastAPI)
- **Frameworks:** FastAPI for API development
- **Libraries:** Pandas for data manipulation, Jinja2 for templating, WeasyPrint or ReportLab for PDF generation
- **Integration:** SQLAlchemy for database interaction, JWT for authentication
- **Deployment:** Docker for containerization

1.2 Database

- **Type:** MySQL
- **Integration:** SQLAlchemy for ORM (Object-Relational Mapping)

1.3 Frontend

- **Languages:** JavaScript, HTML, CSS
- **Frameworks:** React.js for frontend development, Tailwind for UI styling
- **Integration:** Axios/Fetch for API communication
- **Deployment:** Netlify or Vercel for frontend hosting

1.4 Version Control

- **Platform:** Git for version control and collaboration
- **Hosting:** GitHub/GitLab for repository management

2 System Architecture

2.1 Database Design

- **Animals:** Stores information about each animal, including their ID, name, date of birth, gender, and whether they have a passive adopter.
- **Type:** Stores the type of animals, including breed, frequency of check-up, and other type-specific details (e.g., bath for dogs).
- **Food Inventory:** Stores details about food items, including their type, stock, and cost per kilogram.

- **Medical Records:** Stores medical records for each animal, including follow-up information, treatment, diagnosis, and doctor details.
- **Medicine Inventory:** Stores details about available medicines, including name, stock, expiry, and date of purchase.
- **Adopter:** Stores information about people who adopt animals, including their contribution.
- **Employee:** Stores information about employees, including name, role, and date of joining.
- **Credentials:** Stores login information for employees, including username and password.

2.2 Relationships

- **Animals - Type:** One type of animal can be associated with multiple animals, but each animal belongs to one type.
- **Animals - Medical Records:** One animal can have multiple medical records, and each medical record is linked to one animal.
- **Animals - Food Inventory:** Multiple animals can consume different types of food, and each food type can be consumed by multiple animals.
- **Medical Records - Medicine Inventory:** A medical record can reference multiple medicines, and each medicine can be associated with multiple medical records.
- **Animals - Medicine Inventory:** Multiple animals can use different medicines, and each medicine can be used by multiple animals.
- **Adopter - Animals:** One adopter can adopt multiple animals, but each animal has only one adopter.
- **Employee - Credentials:** Each employee has a unique set of credentials, and each set of credentials belongs to one employee.

2.3 API Endpoints

2.3.1 Authentication

- **POST /register:** Allows a new user (employee) to create an account by providing a username, password, and role.
- **POST /login:** Authenticates an existing user by verifying their credentials (username and password) and returns a JWT token for future requests.

2.3.2 Add Information

- **POST /animals:** Adds a new animal to the database by providing its details such as name, date of birth, breed, and passive adopter status.
- **POST /medical-records:** Adds a new medical record for an animal by providing the animal ID, treatment, diagnosis, follow-up date, and doctor's details.
- **POST /employees:** Adds a new employee to the system by providing their name, date of joining, role, and credentials (username and password).

2.3.3 Extract Information and Generate Reports

- **GET /animals/id:** Retrieves detailed information about a specific animal based on its ID, including medical records and adoption status.
- **GET /medical-records/animal_id:** Retrieves all medical records for a specific animal.
- **GET /employees/id:** Retrieves detailed information about a specific employee, including their role and date of joining.

- **GET /reports/animal-health:** Generates a comprehensive report on the health status of animals, aggregating data from medical records.
- **GET /reports/food-inventory:** Generates a report on food stock levels and consumption by animals.
- **GET /reports/medicine-inventory:** Generates a report on medicine usage and current stock.

2.4 Care Notifications

- **GET /notifications/bath-schedule:** Retrieves a list of dogs that need a bath based on their care schedule and triggers notifications to staff.
- **GET /notifications/health-checkups:** Retrieves a list of animals that are due for a scheduled health check-up and sends alerts to staff.
- **GET /notifications/medicine-expiry:** Retrieves a list of medicines nearing expiry and sends notifications to the staff for timely action.

2.5 Financial Reporting

- **GET /reports/expenditure:** Generates a monthly report of expenditures, including costs related to food, medicine, and animal care, helping track and manage the shelter's budget.

2.6 User Interface Interaction

- **GET /animals/search:** Allows staff to search for animals based on various criteria such as name, breed, or medical condition.
- **PUT /animals/update/id:** Allows staff to update details of a specific animal, such as medical records or adoption status.
- **PUT /medical-records/update/id:** Allows staff to update the medical records for a specific animal.
- **GET /reports/generate:** Allows staff to generate various reports such as animal health, food inventory, and medicine stock reports.

3 Frontend Design

3.1 UI Components

- **Forms for Input:**
 - Animal registration form: A form to capture animal details such as name, date of birth, breed, and passive adopter status.
 - Medical record entry form: Allows staff to input medical records for an animal, including treatment, diagnosis, follow-up date, and doctor's details.
 - Employee registration form: Allows administrators to add a new employee by providing their name, date of joining, role, and credentials.
 - Search and filter forms: Components to search for animals, medical records, or employees based on specific criteria.
- **Login and Registration Page:**
 - A user-friendly login page with fields for username and password, offering form validation and error messages for incorrect credentials.
 - A registration page for new users to create accounts with fields for username, password, and role selection.
- **Interactive Notifications:**

- Visual alerts for upcoming check-ups, expiring medicines, and other care notifications. Displayed prominently on the dashboard for easy access.

- **Dashboard and Report Generation:**

- A main dashboard providing quick access to animal records, notifications, and a summary of ongoing activities such as upcoming health check-ups or expiring medicine.
- Report generation options integrated into the UI, allowing staff to generate and download reports related to animal health, food inventory, and monthly expenditures.

3.2 Responsive Website

- **Mobile-Friendly Design:** Ensures that the system's interface adjusts dynamically to various screen sizes, providing an optimal experience on desktops, tablets, and smartphones.
- **Cross-Browser Compatibility:** Ensures the UI works seamlessly across major browsers such as Chrome, Firefox, Safari, and Edge.

3.3 Error Handling and Validation

- **Form Validation:** Implements client-side validation for all input fields, ensuring data is complete and valid before submission. Displays error messages and visual indicators when fields are left blank or contain invalid data.