

# Project Plan Document: Animal Shelter Database

Kriti Bharadwaj (PES1UG22CS295)

Keya Kesani (PES1UG22CS285)

September 30, 2024

## 1 Lifecycle Model

We have chosen the **Agile development model** for this project due to its flexibility and iterative nature.

### 1.1 Justification

- **Flexibility:** Agile allows for evolving requirements, which is important for dynamic projects like an animal shelter database.
- **Incremental Progress:** The project is broken into sprints, allowing us to deliver features iteratively, making scope management easier.
- **User Involvement:** Agile encourages continuous feedback from stakeholders, ensuring alignment with their needs.

## 2 Tools Throughout the Lifecycle

We have selected the following tools for each phase of the lifecycle:

- **Planning Tool:** Trello for task tracking and sprint management.
- **Design Tool:** Lucidchart for database schema and ER diagram design.
- **Version Control:** GitHub for source code management and version control.
- **Development Tool:** VSCode for coding and development (SQL, Python).
- **Bug Tracking:** JIRA for logging and tracking bugs.
- **Testing Tool:** Postman for API testing and JUnit for unit testing.

## 3 Deliverables and Reuse/Build Components

- **Database Schema** (reuse/build): Built based on standard database designs, with some portions reused from templates.
- **Animal Management System** (build): A custom-built feature to manage animal intake, medical records, and adoptions.
- **Adoption Portal** (build): A custom-built UI for users to browse animals for adoption.
- **Reports Module** (reuse/build): Reused with some customization for generating adoption reports.

### 3.1 Justification

Building custom features like the Animal Management System and Adoption Portal ensures the system caters to the shelter's needs. Reusing components such as the reports module saves development time.

## 4 Work Breakdown Structure (WBS)

### 1. Database Setup

- Design the schema for animals, adopters, medical records, etc.
- Create tables and define relationships.
- Implement triggers and stored procedures.

### 2. Backend Development

- Implement API to interact with the database.
- Develop authentication and user roles.
- Build endpoints for animal intake, adoption, and medical history.

### 3. Frontend Development

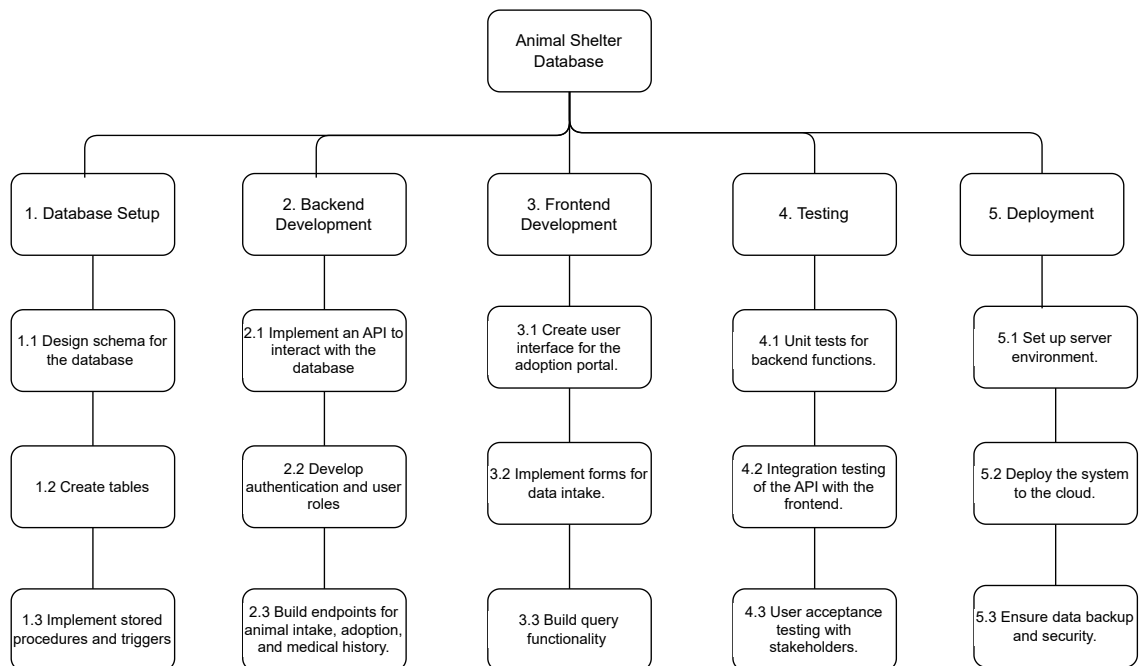
- Create a user interface for the adoption portal.
- Implement forms for intake and medical record entry.
- Build search functionality for available animals.

### 4. Testing

- Unit tests for backend functions.
- Integration testing of the API with the frontend.
- User acceptance testing with stakeholders.

### 5. Deployment

- Set up server environment.
- Deploy the system to the cloud (AWS/Heroku).
- Ensure data backup and security.



Phase based work breakdown structure.

## 5 Effort Estimation Gantt Chart

### 5.1 Effort Estimation (Person Months)

- **Database Setup:** 1 person-month
- **Backend Development:** 2 person-months
- **Frontend Development:** 2.5 person-months
- **Testing:** 1 person-month
- **Deployment:** 0.5 person-months
- **Total Effort:** 7 person-months

### 5.2 Gantt Chart

A Gantt Chart showing the schedule for each task is created using Microsoft Project or GanttProject, with the following key milestones:

- Database setup
- Backend API implementation
- Frontend UI development
- Integration and testing
- Final deployment

