Pet Adoption Application - Requirements

Functional Requirements

- Display a list of adoptable pets, including images, breed, age, and description.
- Allow users to view detailed information about each pet.
- Provide an adoption questionnaire that prospective adopters must complete.
- Enable PAO managers to review, approve, or reject adoption applications. - Allow approved adopters to select a pet.
- Provide a system for PAO managers to resolve multiple adoption requests per pet.
- Require users to log in before accessing adoption features.
- Enable searching and filtering of pets by breed, age, size, and other criteria.
- Update pet availability when an adoption is confirmed.

Non-Functional Requirements

- Data validation should be implemented to prevent SQL injection and security risks. - The system should be responsive and work on desktop and mobile devices.
- Data should be stored in a relational database (SQLite) with structured relationships.
- The backend API should provide JSON responses for all requests.
- The application must be deployed using Docker with separate frontend and backend containers.

Technology Stack

The web application will be developed using the following technologies:

- Frontend: React (SPA) using Vite, React Router for navigation
- Backend: Python Flask API for handling data operations
- Database: SQLite for storing and managing pet and user data
- Deployment: Docker containers for frontend and backend
- CI/CD: GitHub Actions for automated testing and deployments

Git Workflow

- Create a private repository under 'CSE2102-Spring25'.
- Maintain a structured folder organization (frontend, backend, documentation).
- Use GitHub Actions for CI/CD automation.
- Use 'develop' branch for active development, merging feature branches via pull requests. - Merge 'develop' into 'main' for milestone submissions.