Happy Paws: Pet Adoption Website

Submitted in partial fulfillment of the requirements of the degree of

Bachelor of Engineering (Information Technology)

By

Mahvish Siddiqui Roll No – 56

Department of Information Technology

VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY,

Chembur, Mumbai 400074

(An Autonomous Institute, Affiliated to University of Mumbai)

April 2024

Contents:

Happy Paws

Name of student	Mahvish Siddiqui
Class_Roll no	D15A_56
D.O.P	20/03/25
D.O.S	27/03/25
Sign and Grade	

Title: Happy Paws

Project Description:

The rise in stray animals and overcrowded shelters has highlighted the urgent need for streamlined adoption systems. Happy Paws is a web-based platform designed to simplify and promote pet adoption by connecting potential adopters with registered shelters and pet owners. The objective of this project is to bridge the gap between homeless pets and loving families through an intuitive, user-friendly interface that encourages responsible pet ownership. By leveraging technology, Happy Paws aims to enhance visibility for adoptable pets and improve the efficiency of the adoption process.

Requirement gathering: The requirement gathering phase was a critical step in the development of the Happy Paws pet adoption website, as it established a clear understanding of the project goals and user expectations. Functional requirements were identified to ensure core features such as user authentication (login/registration), pet profile management, advanced search and filtering capabilities, real-time adoption request submission, and an admin dashboard for shelters to manage listings and view applicant details. Non-functional requirements included responsiveness across devices, secure data handling (especially user and pet data), fast load times, and a user-friendly interface optimized for accessibility. These requirements were documented using use-case diagrams and user stories to guide the design and development process, ensuring that the final product would be both technically robust and aligned with user needs.

System Requirements:

1. Hardware Requirements:

1. **Processor:** Intel Core i5 / AMD Ryzen 5 or higher (dual-core, 2.0 GHz or faster)
2. **RAM:** Minimum 8GB (16GB recommended)
3. **Storage:** At least 1GB free space (256GB SSD recommended)
4. **Network:** Stable internet connection (especially for MongoDB Atlas users)

2. Software Requirements:

- **Operating System:** Windows 10/11, macOS 10.15+, or Ubuntu 20.04+
- **Code Editor:** Visual Studio Code or compatible IDE
- **Version Control:** Git 2.25+

Technologies Used:

Development	VS Code, Postman, Git
Frontend	React
Backend	Flask (Python 3.8+)
Database	MongoDB Atlas
Styling	CSS
APIs	RESTful Flask APIs

Setup Instructions:

- **Node.js** and **Angular CLI:** To set up Happy Paws, first ensure that Node.js is installed. Visit the official Node.js website and download the LTS version suitable for your operating system. After installation, verify it using node -v and npm -v in your terminal. Once Node.js is installed, open a terminal or command prompt to create a new react app.

- **Python 3.8+ :** Next, install Python 3.8 or higher by visiting the official Python website. Download the appropriate installer for your OS and during installation (especially on Windows), ensure you check the option "Add Python to PATH." After installation, verify it by running python --version and pip --version in the terminal. Optionally, for better environment management, you can create a virtual environment using python -m venv venv' and activate it with venv\Scripts\activate on Windows or source venv/bin/activate on macOS/Linux.

- **MongoDB** (Local or Cloud - **MongoDB Atlas**): Finally, set up **MongoDB** either locally or via **MongoDB Atlas**. To use MongoDB Atlas, go to https://www.mongodb.com/cloud/atlas, create a free cluster, set up a database and obtain the connection string for use in your Flask backend. Add the connection string in the .env file of your flask backend folder.

Backend Setup :

1. **Navigate to backend folder:**
   cd backend

2. **Create a virtual environment:**
   python -m venv venv

   venv\Scripts\activate # For Windows

3. **Install dependencies:**
   pip install -r requirements.txt

4. **Start the Flask server:**
   python app.py

Backend will run at: http://localhost:5000

Frontend Setup

1. **Navigate to frontend folder:**
   cd frontend

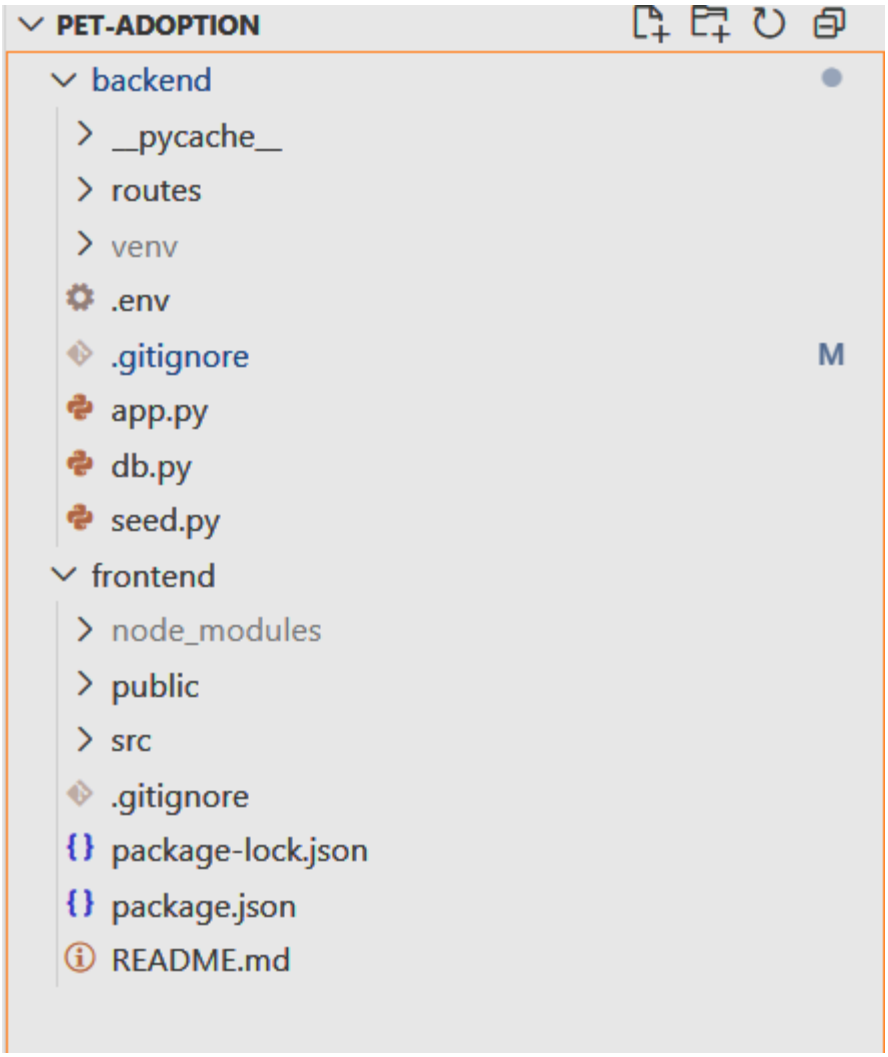
2. **Install dependencies:**
   npm install

3. **Start React development server:**

   Npm start

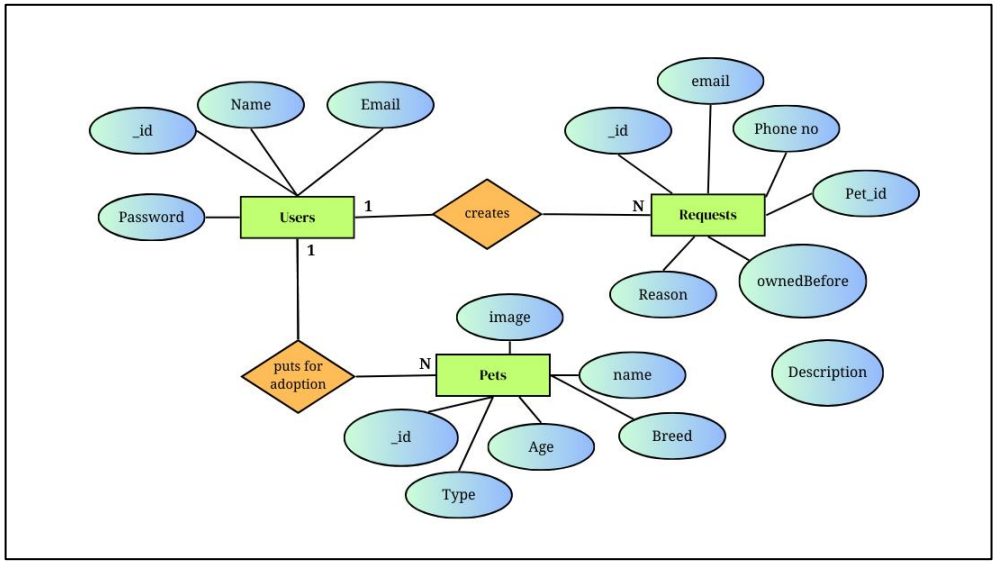
Frontend will run at: http://localhost:3000

Project Structure

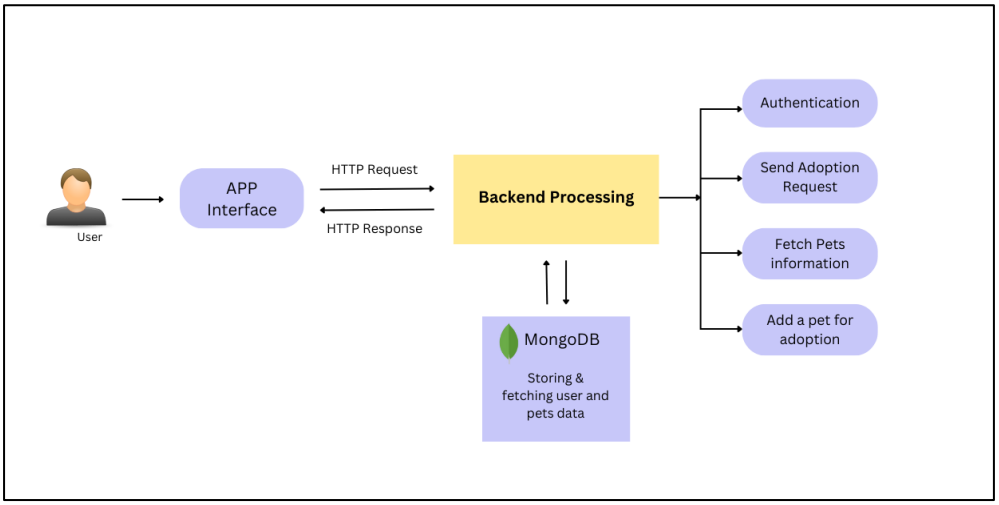


Architectural Diagrams :

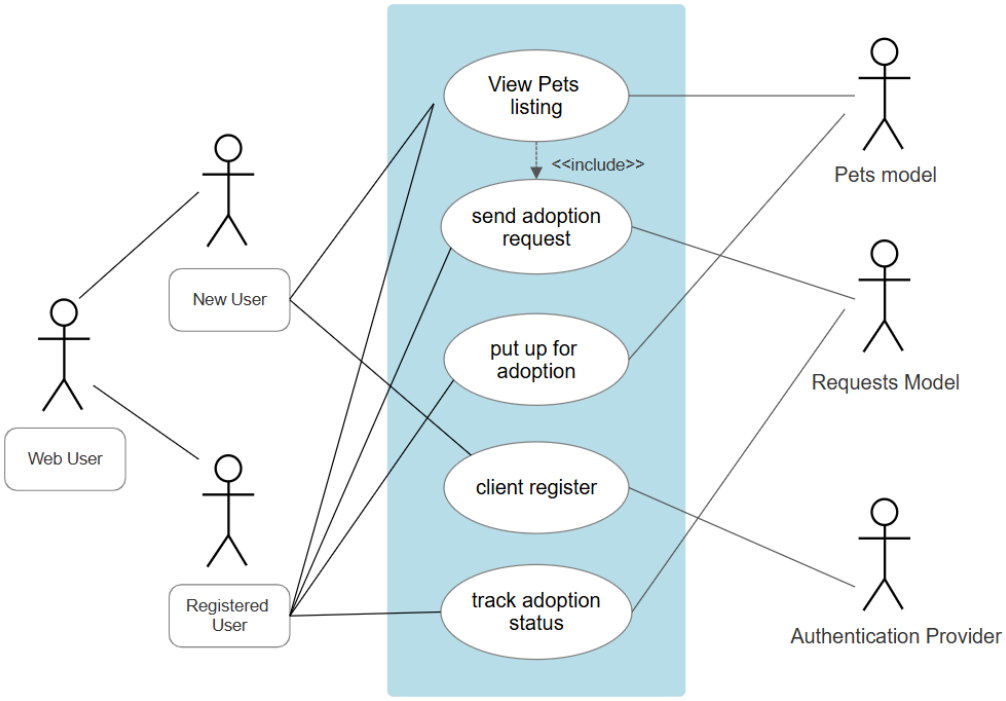
a) ER Diagram -



b) Software Architecture –

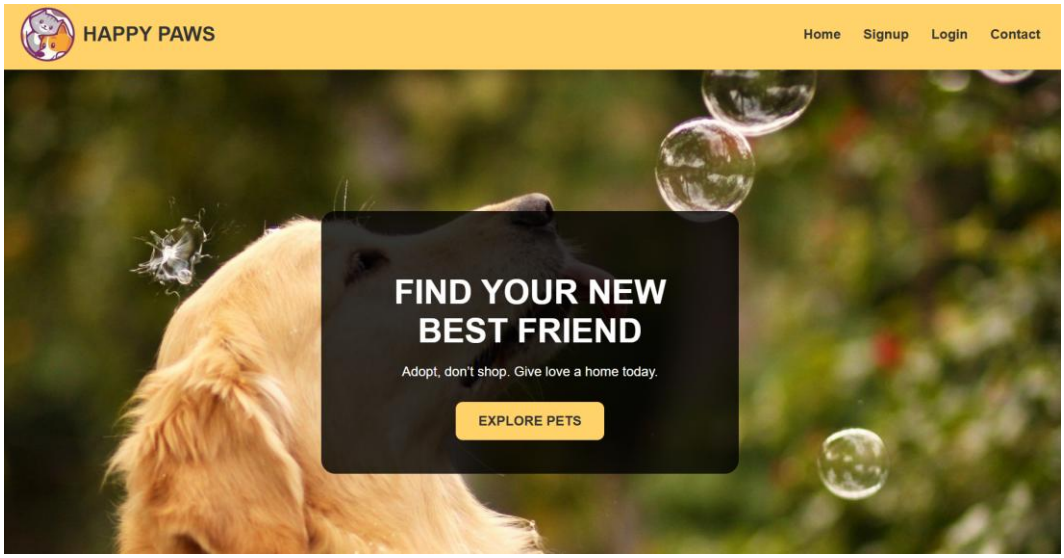


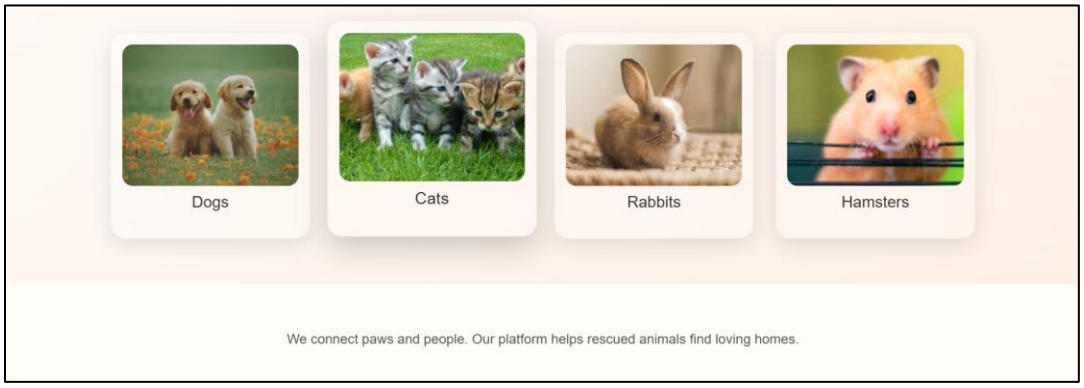
c) Use Case Diagram



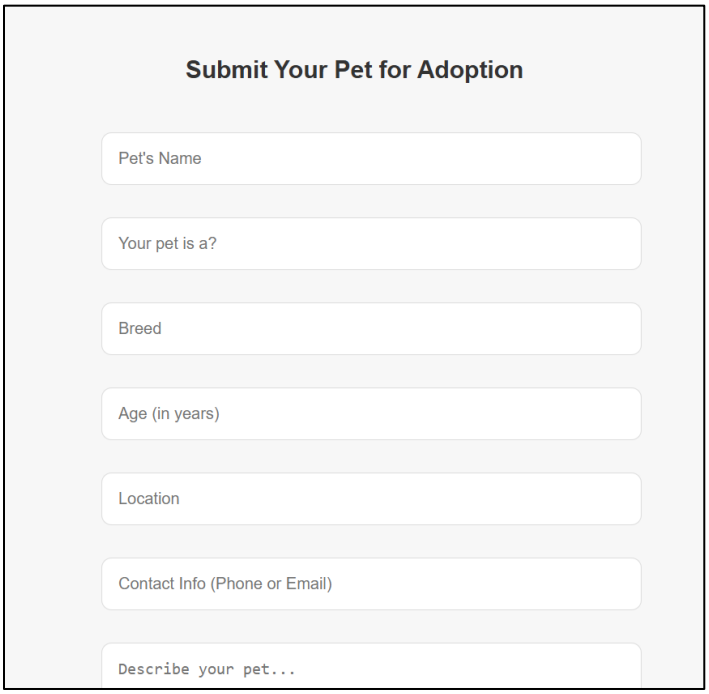
Screenshots of implementation:

Home Page -

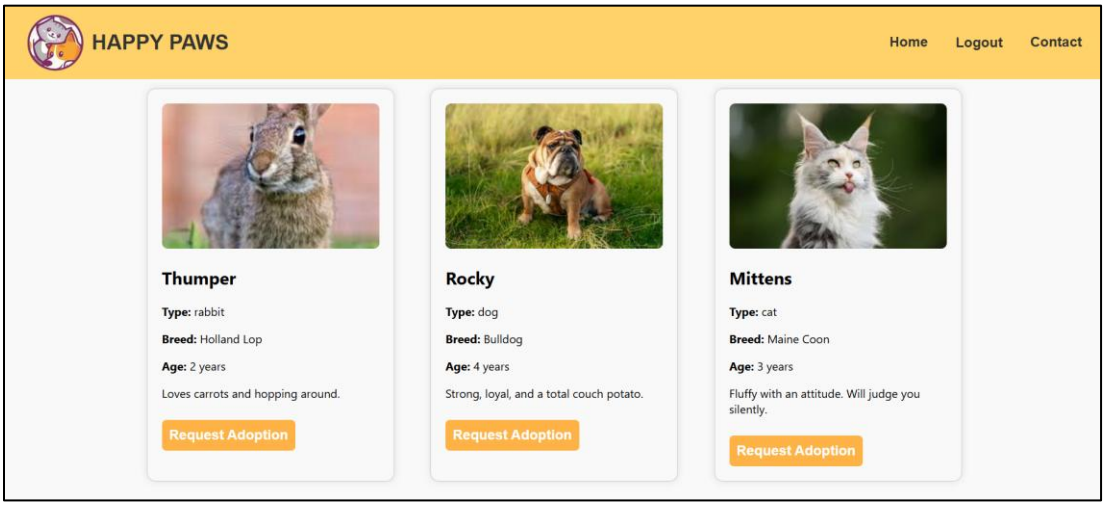




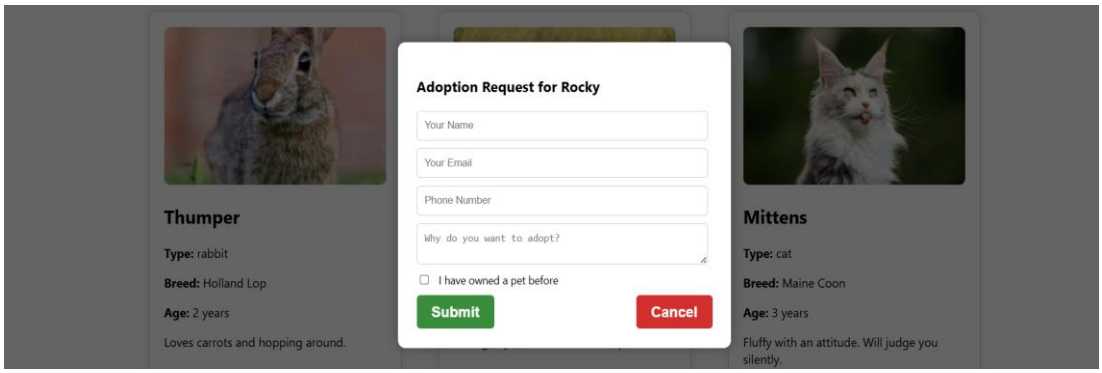
Put your pet for adoption-



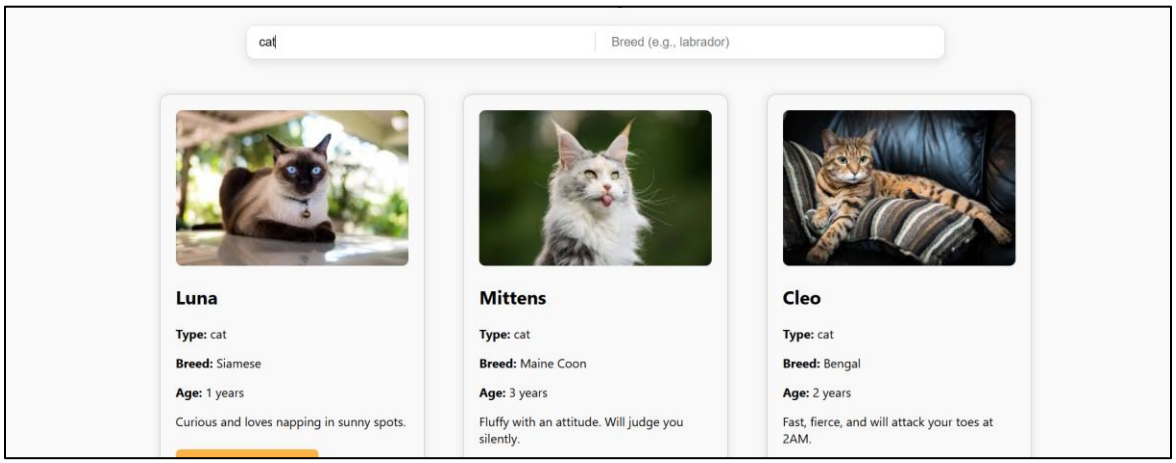
See all pets available for adoption



Request for adoption of pet



Search Functionality – by breed and type



Future Scope: The *Happy Paws* platform has significant potential for future development and scalability. Planned enhancements include the integration of AI-powered recommendation systems to match adopters with pets based on preferences and lifestyle, thereby improving adoption success rates. A mobile application version can be developed to increase accessibility and allow users to browse and apply for adoptions on-the-go. Additionally, incorporating real-time chat support between adopters and shelter representatives can streamline communication and reduce adoption processing time. Features such as vaccination reminders, post-adoption follow-ups, and pet care resources can be added to support adopters beyond the adoption phase. In the long term, the platform could expand to include volunteering opportunities, foster care programs, and integration with government animal welfare databases to promote responsible and large-scale pet adoption.

GitHub Link: https://github.com/MahvishLS/happy-paws

Conclusion:

Happy Paws demonstrates how digital platforms can play a pivotal role in addressing real-world problems such as pet homelessness. Through features like filtered searches, detailed pet profiles, and a streamlined adoption request system, the website ensures a seamless experience for both adopters and shelters. This project not only showcases technical competencies in web development but also emphasizes the societal value of technological solutions. In the future, enhancements such as AI-based pet matching and mobile app integration could further expand the platform's reach and impact.