ADT Final Project Technical Report
Pawesome Adoptions: Where Every Paw Finds a Home

Shraddha Gupta Harsh Lotia Sarvesh Soni

Application URL : https://shraddhagupta.pythonanywhere.com/

GitHub URL : https://github.iu.edu/gupta37/PawesomeAdoptions

PROJECT SUMMARY :

Our project's objective is to use data obtained from national pet shelters in the US via the PetFinders.com API to build a database and online interface for pet adoption. The database, which contains information about animals that are available for adoption, will make the adoption process easier for prospective pet owners. Users will be able to search, filter, and browse through available pets using the interactive interface, which will streamline the process of matching creatures with loving homes. The purpose of this project is to provide a user-friendly platform for people and organizations involved in pet adoption, with the aim of fostering lasting relationships between adoptive families and their pets.

DESCRIPTION

PURPOSE:

The purpose of our project is to utilize data from national pet shelters in the US via the PetFinders.com API to build a database and online interface for pet adoption. The platform aims to simplify the adoption process for prospective pet owners by providing an interactive interface to search, filter, and browse available pets, fostering lasting relationships between adoptive families and their pets.

Our project serves a vital purpose in the realm of pet adoption by harnessing the power of technology to streamline the process of finding loving homes for animals in need. By utilizing data from national pet shelters via the PetFinders.com API, we have constructed a comprehensive database that houses information about available pets. Through our online interface, prospective pet owners can easily search, filter, and browse through the database to find their ideal companion. Our platform is designed to simplify the adoption process, offering a user-friendly experience that facilitates successful matches between pets and adoptive families. Beyond merely facilitating adoptions, we aim to foster lasting relationships between pets and their new owners, promoting the well-being of both animals and humans alike. Overall, our project represents a significant step forward in promoting responsible pet ownership and ensuring that every paw finds a loving home.

DATA :

Information on pet adoption is efficiently arranged amongst four main entities: Pet, Request, User, and Organization. Complete information on animals up for adoption, such as breed, age, and status, is stored in the Pet table. The organization_id provides a link to the relevant organization. Through distinct IDs, status updates, and timestamps, the Request table monitors the advancement of adoption requests, establishing a foreign key connection between individual pets and users. In order to facilitate contact and updates during the adoption process, the User table keeps track of the personal information of people looking for adoption. Last but not least, the Organization table facilitates cooperation and communication with possible adopters by managing contact information and representing shelters. All parties engaged in the pet adoption process can be assured of reliable data management and seamless operations thanks to this schema.

We Performed some preprocessing steps on the dataset such as -
1. We read and normalize JSON data from certain DataFrame columns by using `pd.json_normalize` to flatten nested structures into individual columns and transform JSON strings into dictionaries. Following extraction, the new columns are combined with the old DataFrame and the original columns containing JSON are deleted. By offering comprehensive, understandable insights on characteristics like pet types, colors, and addresses, this improves data analysis capabilities.
2. Using data wrangling techniques, we cleaned and reformatted a DataFrame by filtering out 'None' values, converting boolean strings to integer representations, merging attributes into single columns, and eliminating redundant columns. These modifications improve the DataFrame's usefulness for upcoming analytical and modeling operations.
3. Two specially filtered DataFrames are processed and data is inserted into SQLite database tables. The 'Pet' and 'Organization' tables are successfully updated in 'pet_adoption.db' by using the \`to_sql} method with the \`if_exists='replace'} option and \`index=False}. This allows for improved data management and retrieval for additional analysis.

TOOLS :

We used Python for backend coding and Visual Studio Code for editing.

Web Framework - The application is built on Flask.

Frontend Technologies - HTML, CSS, Bootstrap, jQuery, Popper.js
The structure of the web pages is defined using HTML while the application's style is managed using CSS and bootstrap is used for responsive and styled UI components.

Templating - Jinja2
Jinja2 is used for dynamic content rendering on the frontend. It is used with Flask applications for integrating server-side data with client-side presentations.

Database - SQLite
Flask is integrated with sqlite using the Flask-SQLAlchemy extension.

Authentication - CSRF
This is used to protect against Cross-Site Request Forgery attacks.

PROJECT DEVELOPMENT:
Our project is built on a foundation of meticulous planning, collaborative effort, and a keen focus on utilizing the right tools and technologies to achieve our objectives. At the heart of our application lies the Flask web framework, providing a robust and flexible platform for developing web applications in Python. We leveraged Flask's simplicity and extensibility to create a backend infrastructure that seamlessly handles user requests, database interactions, and authentication processes. With Flask-SQLAlchemy, we seamlessly integrated SQLite into our application, allowing for efficient data storage and retrieval. On the frontend, we employed a combination of HTML, CSS, Bootstrap, and jQuery to craft visually appealing and responsive user interfaces that ensure a smooth browsing experience across devices. Jinja2 templating facilitated dynamic content rendering, enabling us to inject server-side data into client-side presentations with ease. Additionally, we implemented CSRF protection to safeguard against potential security threats, ensuring the integrity and security of our application. Through careful selection and implementation of these technologies, we have built a robust and user-friendly platform that fulfills our project's objectives with efficiency and elegance.

FUNCTIONALITIES :

The web application has the following functionalities -
1. There are pages for users to login and sign up. New users can sign up or create a new account by filling details like name, email and password. These users can then login using their email and password.
2. In the home page, there is a list of available pets where the user can filter them by name, breed, age, gender and size suggesting a dynamic way to search.
3. Once the user clicks on the desired pet link, the pet details page opens up where all the information of the pet is available such as its breed, name and whether it's available for adoption or not.
4. The authenticated users can submit a request to adopt a pet and once the users send the request, they can complete or update personal details like phone number and address, suggesting a user profile management component. Along with the users' details, the page also includes the adoption history where the user and admin can check the adoption status and date.
5. There is an 'About Us' page wherein the information about the website has been provided and the users or visitors can connect to us through the contact details mentioned on the page.
6. The website features a navigation bar that makes it simple for visitors to navigate between sections like Home, Login, About Us, My Profile, Sign up, and Logout.
7. We have also created an admin account where the admin can see the entire database. Here, the admin can create, delete or update the entries as needed. For example, If a user has requested a pet for adoption then the admin can update the same in the database which in return will reflect on the website for both the users as well as the admin.

ISSUES FACED :

Our team faced some technical difficulties when deploying our project on Render via GitHub, Heroku, **PythonAnywhere** and Glitch, mostly with regard to the requirements.txt configuration file and command configurations. To address this, we sought solutions through educational videos and further internet searches that provided insights into the correct usage and configuration. By applying the knowledge gained, we were able to successfully resolve the error and proceed with the deployment of our project on **PythonAnywhere**.
These experiences not only helped us overcome immediate technical hurdles but also enhanced our team's problem-solving skills and deepened our understanding of web application deployment workflows.

GROUP CONTRIBUTION :

Name	**Tasks**	**Average Time Spent**
Shraddha Gupta	● Created wireframes for the Home page and About Us page. ● Developed the frontend code for both pages. ● Implemented backend code for database creation ● Programmed data fetching with filtering query functions. ● Compiled the project report. ● Deployed the application on PythonAnywhere.	25 hours
Harsh Lotia	● Designed wireframes for the Pet Details Page and User Profile page. ● Programmed the frontend code for both pages. ● Executed backend coding to populate the database with a Kaggle dataset. Created code for user authentication across pages ● Developed functions for managing data and requests within the pages. ● Compiled the project report.	25 hours
Sarvesh Soni	● Designed the admin panel for the website ● Created Admin Views and Model Views for all the tables in the database ● Granted all CRUD privileges to Admin and connected it to the database. ● Developed functions to authenticate and reflect on various requests on the website from Admin-end ● Compiled the project report.	25 hours