IT 309 SOFTWARE ENGINEERING

PROJECT DOCUMENTATION

Pet Adoption app

Prepared by:
Eldar Jahić

Proposed to:
Nermina Durmić, Assist. Prof. Dr.
Aldin Kovačević, Teaching Assistant

June 21st 2023

Table of Contents

1. Introduction

1.1. About the Project

The Pet Adoption app is a web-based application that seeks to pair users with animals in need of homes. This application is intended to simplify the adoption process and increase the likelihood that animals in shelters, on the streets, and in rescue organizations will find homes. Here you can find the link of my project: https://pawfriends.onrender.com/

1.2. Project Functionalities and Screenshots

1. Login

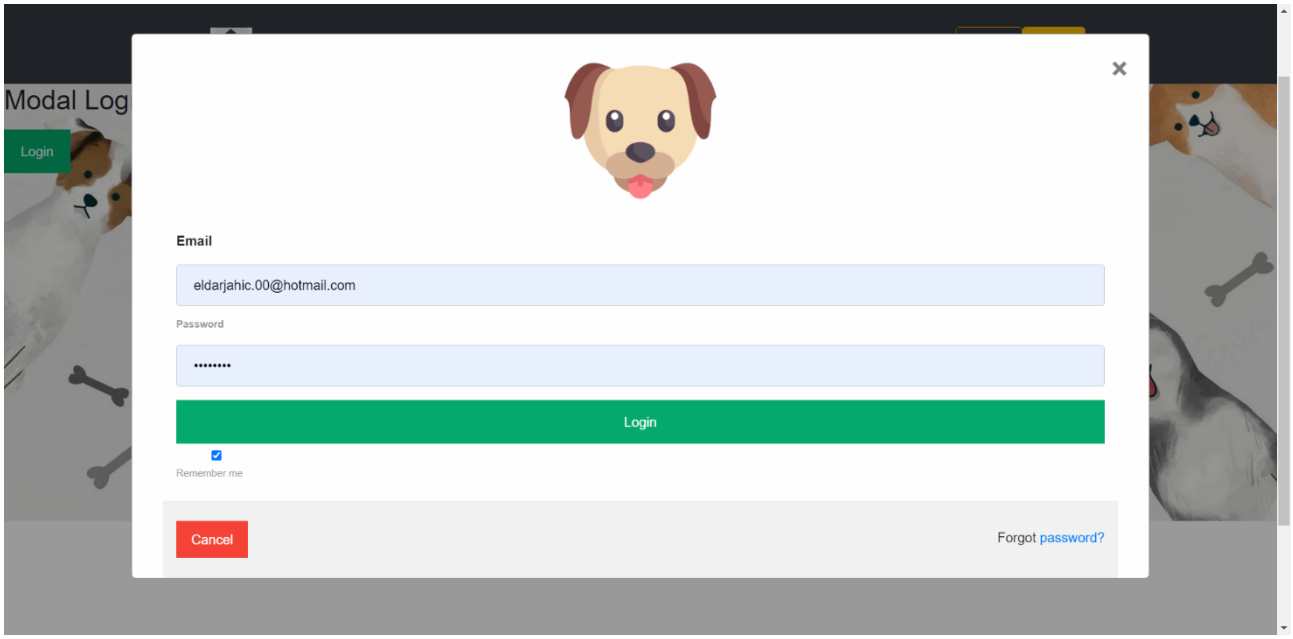

Figure 1.

2. Signout

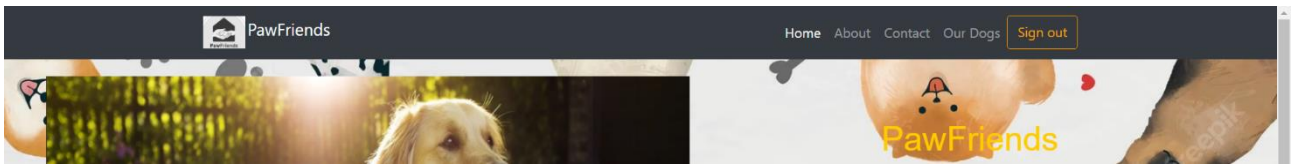

Figure 2.

3. Home/Landing page

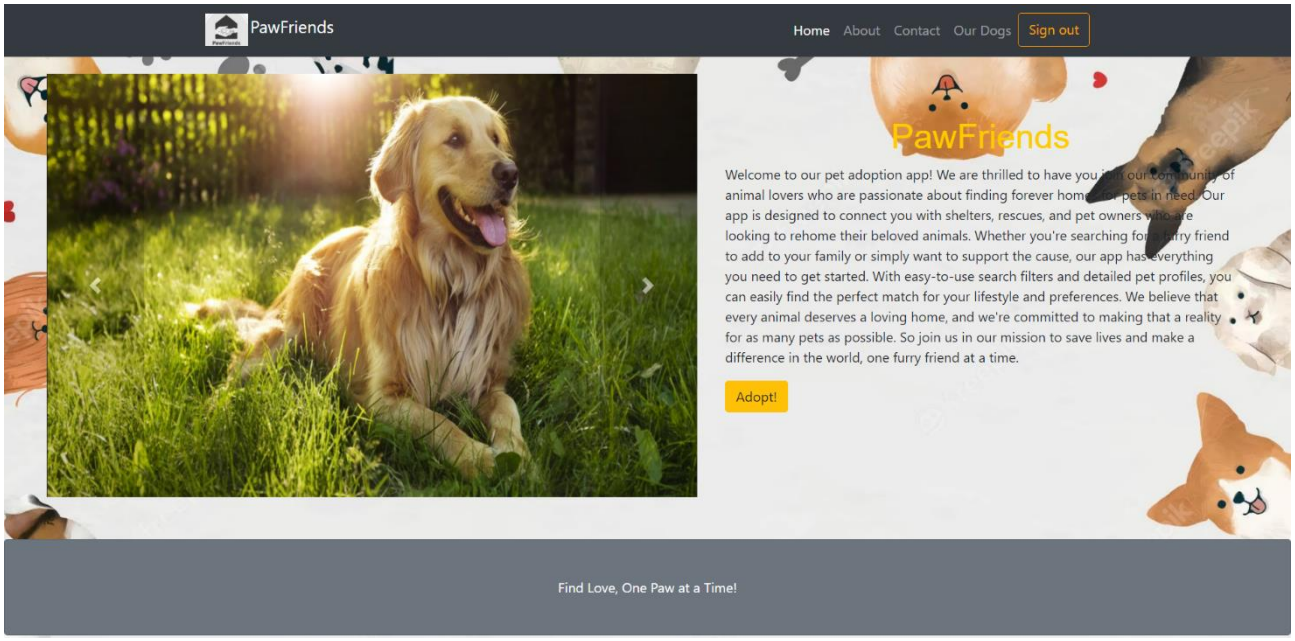

Figure 3.

4. View dogs

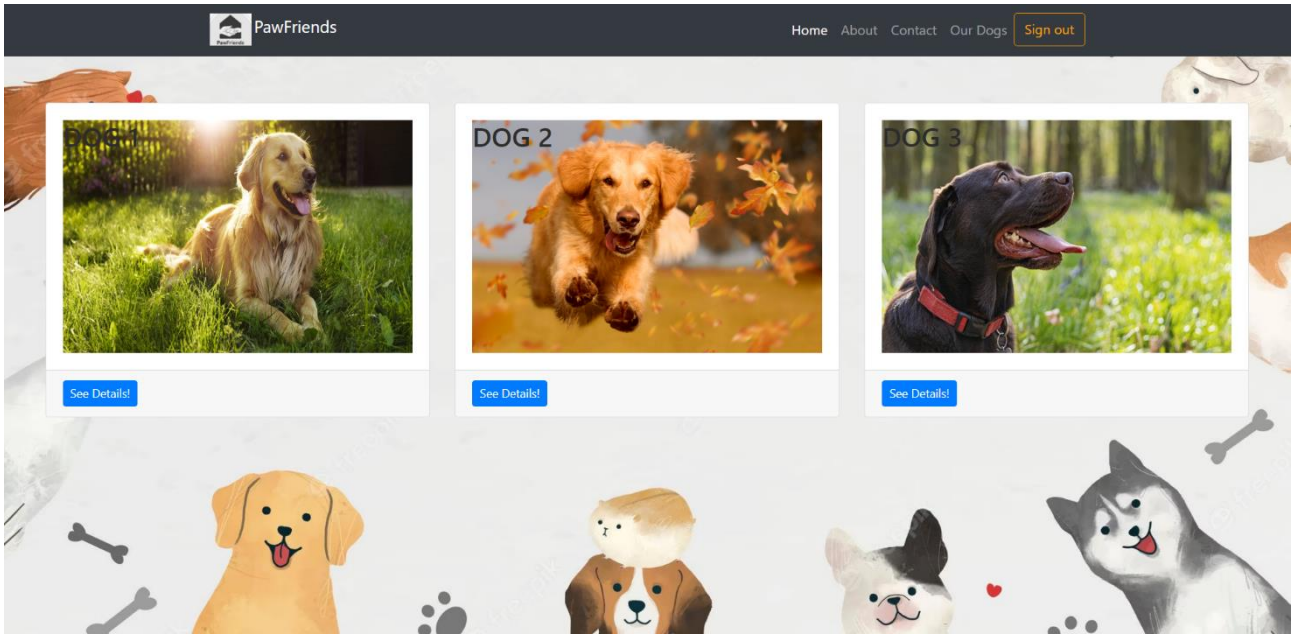

Figure 4.

5. View dog details

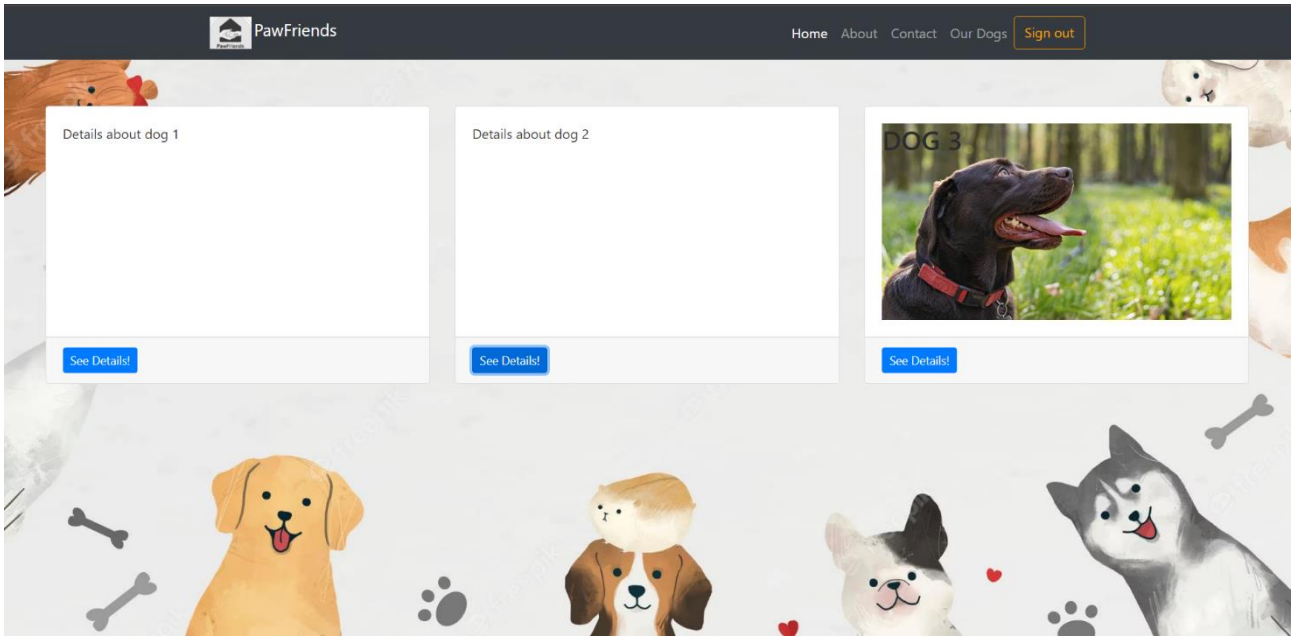

Figure 5.

Link to GitHub repository: https://github.com/eldarjahic/webapp

2. Project Structure

2.1. Technologies

List of technologies used in this project:

1. The backend of the project is done in **PHP**, using the framework **FlightPHP** for creating the RESTful API, **Swagger** for generating and showing the documentation of the API and **PHPUnit** for writing the unit tests used to test the crucial parts of the application logic
2. For the frontend part, **HTML** and **CSS** was used to create the UI of the application and **JavaScript** was used for running client side code. Alongside **JavaScript**, **jQuery** was used to handle the DOM element manipulation and validation of forms in the UI. For the API request **ajax** was used to make the process of handling API request easier and faster.
3. For the database I used **MySQL** Workbench, this decision was made because I am the most familiar with this DBMS and it is free to use.

The coding standard used in this project is psr-12 in the backend. This is the recommended coding standard for the PHP programming language. Auto-formatting was added to my Visual Studio Code text editor in order to format all files after they are saved. This ensured consistency and better readability within the backend application code.

2.2. Database Entities

1. pet
2. user
3. adoption

2.3. Design Patterns

- DAO (Data Access Object) – design pattern used on the backend within the *rest/dao* folder

- the DAO design pattern was used on the backend in order to separate the data related code into another layer. Due to this the services of the application, that are in charge of handling the business logic of the backend, are isolated from the database access code. This concept is called the Separation of Concerns.

- Singleton – design pattern used on the backend for the database connection in the rest/dao/DBConnection.class.php

- the Singleton pattern was used to prevent having a separate database connection for every DAO class. The pattern ensured that each DAO class uses the same database connection, this improves performance of the backend application and reduces the load on the database due to the fact that only one database connection is being utilized.

2.4. Tests

For backend testing, unit test were written in PHP with the PHPUnit framework. The tests are in the *test/UserServiceTest.php* file withing the repository. The testing was conducted on the UserService class that is used to handle the business logic for managing users. The login method was tested with 5 tests. The tests tested the cases where the user was not found by the DAO class, the password was incorrect and the case where the password was correct. The tests asserted that exceptions with correct messages and codes were thrown and that the success scenario returned the correct “token” key. Because there was no time to setup the database for testing, mocking was used to mock the UserDao class that interacts with the database. For each test the get_user_by_email() function was configured to return values that allowed the testing in the login business logic. This way the nuances with the database connection were skipped and it allowed for more focused and simpler test. For the frontend part of the application, manual testing was conducted by myself. Due to time constraints I wasn’t able to integrate selenium automation test on the frontend, however I look forward to adding them in the future for better sound proofing of the frontend part. Below in Figure 6. it is show that all 5 tests pass when running the PHPUnit command in the terminal.

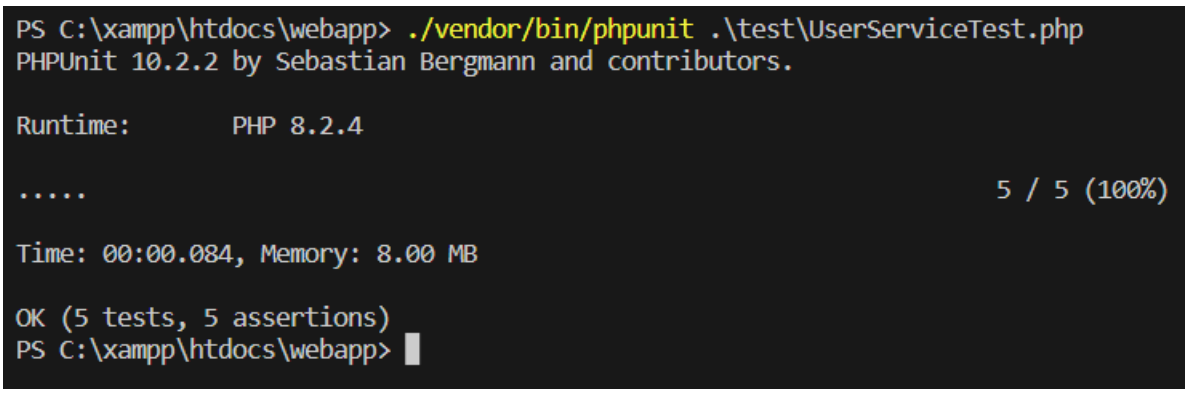

Figure 6.

3. Conclusion

My goal of this project is to simplify the overall process of pet adoption for both people wanting to adopt and for shelters looking for suitable home for pets. I am happy with the current state of my app but there is much to work on such as adding Drivers to the app that transport pets.
Luckily, there were no major problems and difficulties except some deployment issues which were resolved with some help of Teaching Assistant. I am really happy with the course and learned some new stuff that will be useful in the future.