

BbPets Adoption Application

Project Overview

Javier Bastande Leon

23176989



-Visit [Live Application](#) ()

NOTE: Since it is on a free hosting service, some features might take some time to perform their task. It might feel as non-response, just wait and be patient 😊

- **Spring Back End App** repository [Spring/Baby_Pets_Adoption](#) deployed on [Render.com](#)

- **React Front End App** repository [React/Baby_Pets_Adoption](#) deployed on [Github Pages](#)

Table of Contents

1. [Project Overview](#)
2. **Features**
 - [Authentication and Autorization](#)
 - [Searching and Sorting](#)
 - [Pagination](#)
 - [Pet Details View](#)
 - [Adoption Application Request](#)
 - [Pet Listing ManagementR](#)
 - [Adoption Applications Management](#)
 - [Notifications](#)
 - [Profile Management](#)
3. [Future Implementations & Improvements](#)
4. **How To Run the Application**
 - [Run Spring App Locally](#)
 - [Run React App Locally](#)
4. **Site Map**
 - [View Site Map here](#)
5. **SEO**
 - [Metatags](#)
 - [Robots.text file](#)
 - [Sitemap.xml File](#)
 - [Meaningful keywords](#)
6. **Tests**
 - [Manual Test](#)
 - [LightHouse Browser Extension](#)
 - [W3 Validator](#)
 - [Automated Tests](#)
 - [Unit Test Specifications](#)
 - [Integration Test Specification](#)

1.Overview

Pet adoption application designed to solve 2 issues. It is designed for Pet owners with pregnant pets looking for giving away the expected baby pets in advance rather than once they are already born, or for Pet Risers looking for selling pets of pure breed, and designed for users who are only interested in fostering newly born pets.

It is designed with a minimalistic approach to facilitate user interaction with a very easy and straightforward interface.

In order to ensure user security, and provide with the necessary features, the adoption services require the registration of users allowing them to submit adoption application and to have access to management features:

- Create, update and delete pet listings
- Upload images of the mother's pet and optionally the father's pet image
- Submit and drop adoption applications
- Manage adoption requests by updating the status of these "Viewed", "Accepted", "Selected", or dismiss them.
- List and sort the adoption applications by their status. When the pet owner updates an application status, this application is moved to the list of applications sorted by their status.
- Manage their personal details such as name, location, and email address, as well as deleting their profile and all data related to them.

Furthermore, the application provides with other features:

- Search of pets divided in categories
- Sorting by location or pet's breed
- Pagination of search results allows users to paginate their search and enhance user experience and system performance by only loading the necessary data and images.
- Provides with useful information about adoption process and fee (for pets on sale)
- Notifications. Users receive a notification of any action taken in their pet listings or adoption application, this is, when a user submits an adoption application the pet owner receives a notification when they login back and the notification will remain till they "Mark as Viewed" the notification. Also applicants receive a notification when the pet owner updates their application status as mentioned earlier or when the owner decides to remove it from the list of applicants.

2. Features

2.1.Authentication

- Registration

SignUp

Name	Last name
<input type="text"/>	<input type="text"/>
Email Address	Confirm Email Address
<input type="text" value="1@5.com"/>	<input type="text"/>
Password	Confirm Password
<input type="password" value="...."/>	<input type="password"/>
<input type="button" value="Go!"/>	

- Login

es

← SignIn

Email Address

1@5.com

Password

....

Login! [SignUp](#)

[Forgot password](#)

- **Password Reset**

Reset Password

New Password

....

Confirm new password

....

Passwords match

Submit

2.2 Searching and Sorting

- User can choose a pet category to search for



- Users can sort their search for pets by location, breed...



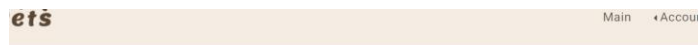
2.3. Pagination

- Users can load pets by chunks of 6 pets each time



2. 4. Pet Details View

- Users can select to view a specific pet details on a separated page where they can find more information about the pet and the adoption service if they are previously authenticated



2.5 Adoption Application

- Only authenticated users will have access to adoption services



Price:
Free Adoption

Owner's Name:
Javier

Comments

Such as crazy cat!! Gens could be passed down to kitties.. you have chances of adoption a cat you'll never be bored with!

Only subscribed users have access to adoption services!

- Authenticated user will see the adoption section down the pet details view page and they can submit adoption applications from the “Pet details view page”

Adoption Request

Such a beautiful decision from you!

We will provide the pet provider with the user name associated to your account.

Your preferred contact email address

County of your current location

lovepets@gmail.com

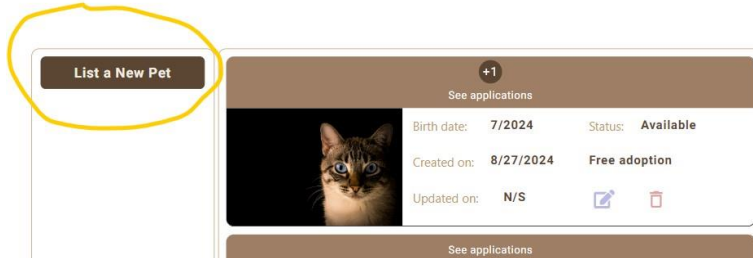
Mayo

Some other details about you to make you stand out!

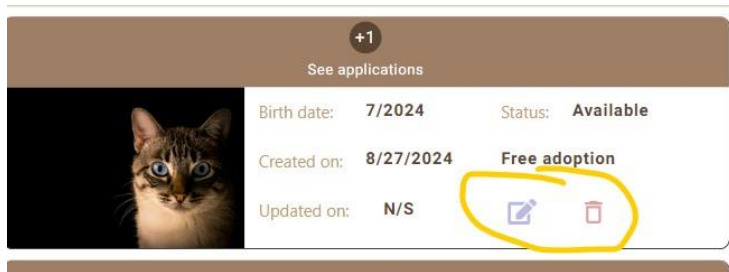
Submit

2.6 Pet listing Management (My Listings page)

- Users can list a New Pet

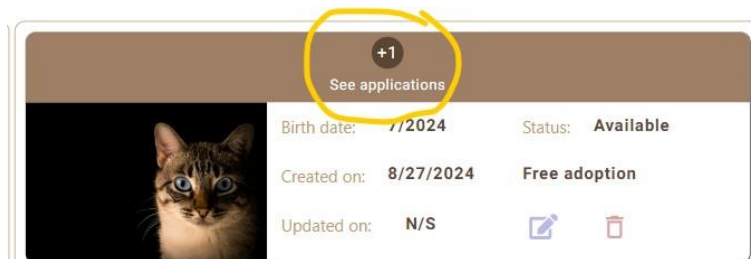


- Pet providers can update a specific pet listing or deleted from the “My listings page”



-

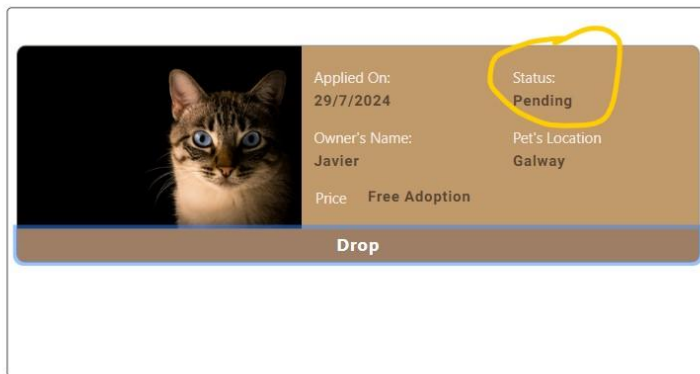
- Pet providers can view adoption applications of the pet that are to be reviewed



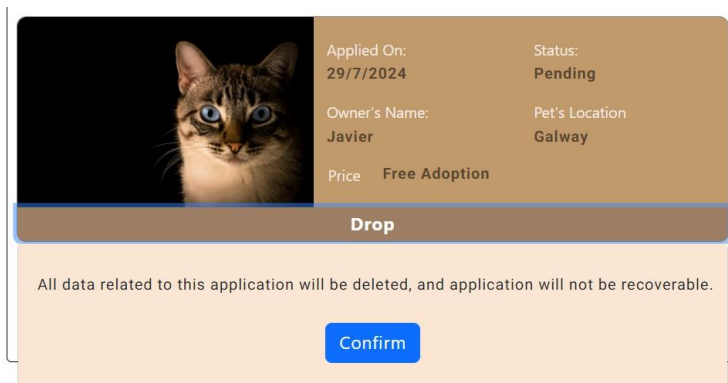
2.7 Applications Management (My Applications page)

2.6.1 Adoption Applicants

- Adoption applicants can view the list of their submitted applications and the status of it. When the pet owner changes the status, applicants receive a notification and will see the status changed



- Adoption applicants can drop a specific application. The pet owner will be notified.



2.6.2 Pet Providers

- Pet providers can view all applications sorted by their status.

A screenshot of a web interface showing four tabs: "Pending", "Viewed", "Accepted", and "Selected". The "Viewed" tab is highlighted in a light orange color. A red hand-drawn circle highlights the entire row of tabs. Below the tabs, a card displays application details for a user named Javier. The card includes fields for Name, Email address, Applied On, and Status, along with "Drop" and "View" buttons. The status is currently "Pending".

Pending	Viewed	Accepted	Selected
ref: 3c0c			
Name: Javier		Email address: notri80@gmail.com	
Applied On: 8/29/2024		Status: Pending	
		<button>Drop</button>	
		<button>View</button>	

- Pet providers can view any comments made by the applicants in order to decide what to do with the application and change the status of it based on that. The applicant user will be notified of the new status update.

A screenshot of a web interface showing the same four tabs as the previous image. Below the tabs, a card displays application details for a user named Javier. The card includes fields for Name, Email address, Applied On, and Status, along with "Drop" and "View" buttons. The status is currently "Pending". Below the card, there is a section titled "Update Status" with three radio buttons: "Viewed", "Accept", and "Select". The "Accept" radio button is selected. A red hand-drawn circle highlights the "Update Status" section. Below the radio buttons is an "Update" button.

Pending	Viewed	Accepted	Selected
ref: 3c0c			
Name: Javier		Email address: notri80@gmail.com	
Applied On: 8/29/2024		Status: Pending	
		<button>Drop</button>	
		<button>View</button>	
The applicant did not leave any comment			
Update Status			
<input type="radio"/> Viewed <input checked="" type="radio"/> Accept <input type="radio"/> Select			
<button>Update</button>			

- **Pet providers can drop the application, and the applicant user will be notified of that.**

Pending	Viewed	Accepted	Selected
<div>Name: Javier</div> <div>Email address: notri80@gmail.com</div> <div>Applied On: 8/29/2024</div> <div>Status: Pending</div>			<div>ref: 3c0c</div> <div>Drop</div> <div>View</div>

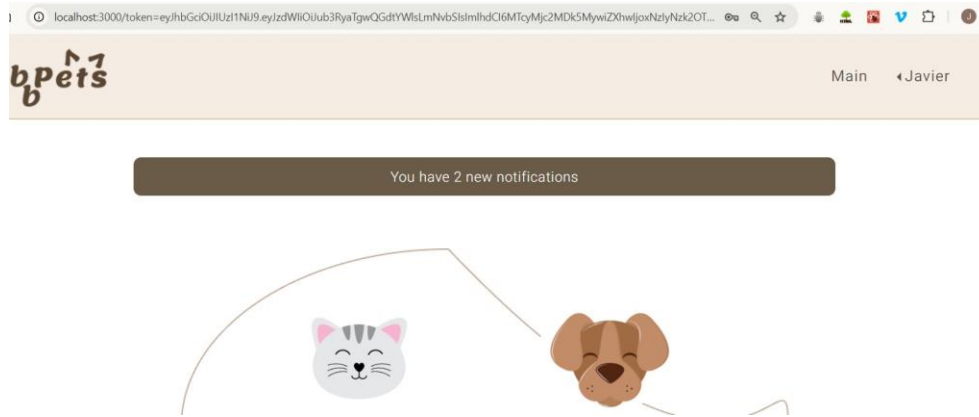
- **Pet providers receive feedback when they update the status of an application**

Application status was updated. You can now see it within 'Selected' tab

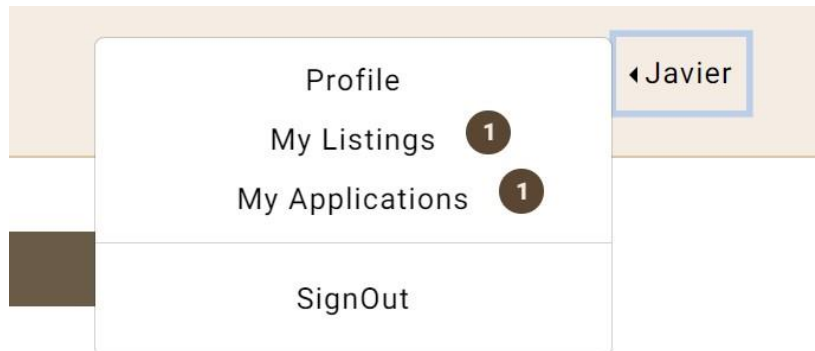
Pending	Viewed	Accepted	Selected
No applications with status 'Accepted'			

2.8 Notifications

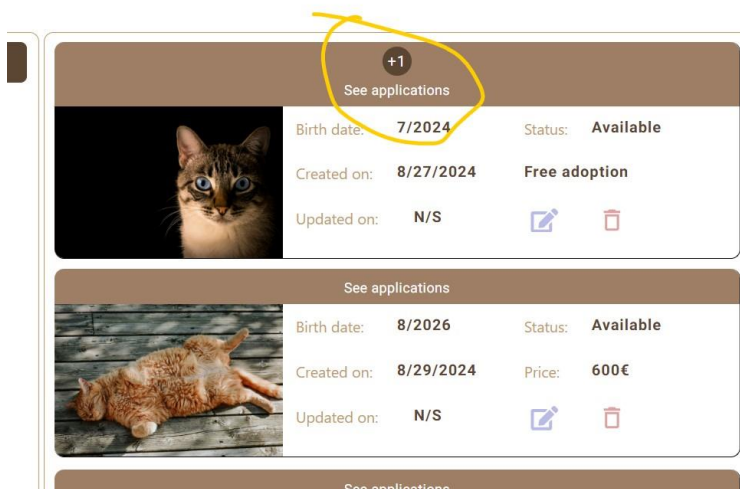
- A warning message of unviewed notifications when the user logs back in.



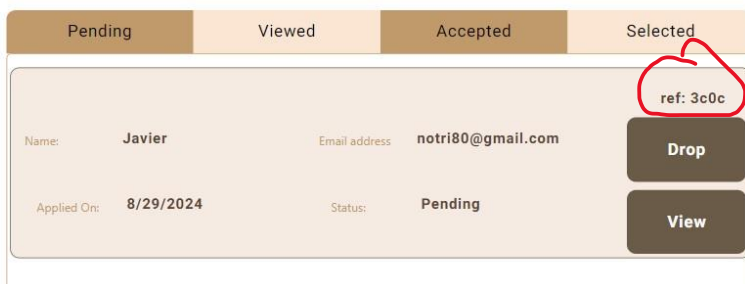
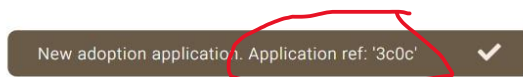
- Persistent notifications badges on the navbar menu



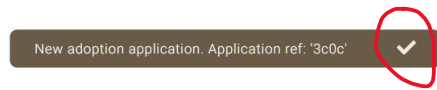
- Pet providers can see a notification badge with the number of notifications unviewed related to a pet listing



- Within the pet applications page, pet providers will see a message of the for each of the incoming adoption applications that have not been viewed just yet, and their reference to easily track them down



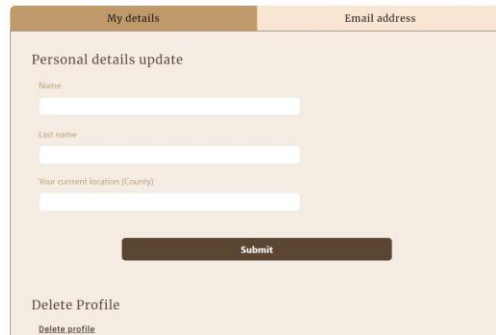
- User must mark notifications as “viewed” or they will persist accross the app otherwise.
Whne marked as “viewed” notification will be deleted from database too.



Pending	Viewed	Accepted	Selected
<div>ref: 3c0c</div> <div><div>Name: Javier</div><div>Email address: notri80@gmail.com</div><div>Applied On: 8/29/2024</div><div>Status: Pending</div><div>Drop</div><div>View</div></div>			

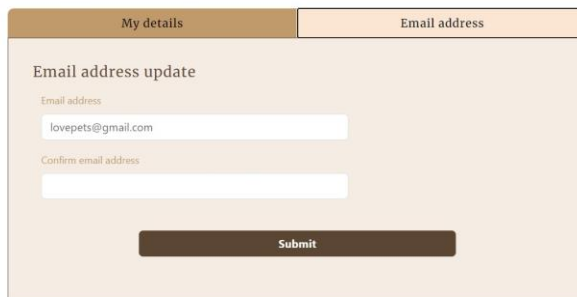
2.9 Profile Management

- **Users can update their personal information**



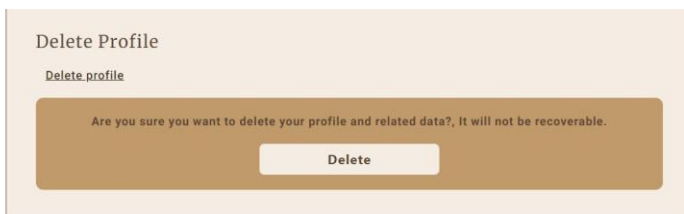
The form is titled "Personal details update" and is part of a "My details" tab. It contains three input fields: "Name", "Last name", and "Your current location (Country)". Below the fields is a "Submit" button. At the bottom, there is a "Delete Profile" link and a "Delete profile" button.

- **Users can update their email addresses associated with the account**



The form is titled "Email address update" and is part of an "Email address" tab. It contains two input fields: "Email address" (with the example "lovepets@gmail.com") and "Confirm email address". Below the fields is a "Submit" button.

- **Users can delete their profile and all data related such as pet listings, adoption applications etc...**



The form is titled "Delete Profile" and contains a "Delete profile" link. Below the link is a confirmation message: "Are you sure you want to delete your profile and related data?, It will not be recoverable." and a "Delete" button.

3.Future Implementations and Improvements

There are several potential improvements and new features to provide a better user experience and ensure the app is more maintainable and scalable.

1.Additional pet categories

- Extend the variety of pets available categories for adoption such as birds, reptiles, etc.

2.User Interface refactor and redesign

- Improve the overall user experience with a more redesigned user interface, focusing on modern design principles, usability, and accessibility.

3.Page loading performance

- Optimize page loading times to ensure faster and smoother user interactions, particularly on pages with heavy content, since the current loading performance is very poor.

4.Notification service refactor

- Refactor the existing notification service on the backend to improve maintainability, and performance.
- Implement proper react state handling of notifications in the React application to ensure timing and reliability.
- events.

5. Real-time messaging System:

- Implement a messaging system that allows applicants and pet providers to communicate directly, enhancing overall communication.

6.Distributed Application architecture:

- Turn the current backend app into a microservice distributed architecture to improve maintainability, scalability, and fault tolerance.

3. How to Start Up the Application

3.1 Running the Spring Boot Application Locally

Prerequisites:

- **Java Development Kit (JDK):** Ensure you have JDK 11 or higher installed.
- **Maven:** Ensure Maven is installed and configured in your system.
- **MongoDB:** Ensure MongoDB is installed and running.

Steps:

1. Clone the Repository:

1. Open a terminal
2. Navigate to the folder directory you'd like to store the project
3. Once there enter the command on your terminal
`git clone https://github.com/JavierBL89/Baby_Pets_Adoption.git`
4. Open you IDE to run Java

2. Configure Environment Variables: Create an .env file in the root directory of the Spring Boot application with the following environment variables:

```
SPRING_DATASOURCE_URL=mongodb://localhost:27017/bb_pets_adoption
SPRING_MAIL_HOST=smtp.your-email-provider.com
SPRING_MAIL_PORT=587
SPRING_MAIL_USERNAME=your-email@example.com
SPRING_MAIL_PASSWORD=your-email-password
```

NOTE: You can run it without the _MAIL_ vars, only you will not have access to certain features that require authentication.

In order to make it fully functionall, you'll need to configure your Gmail settings.

```
SPRING_MAIL_HOST=smtp.gmail.com
SPRING_MAIL_PORT=587
SPRING_MAIL_USERNAME=your-gmail-address@gmail.com
SPRING_MAIL_PASSWORD=your-gmail-password
```

Gmail Configuration:

- Enable "Less secure app access" in your Google account settings or use an app-specific password if you have 2-Step Verification enabled.

3. Build and Run the Application:

1. Open the terminal of you IDE
2. Navigate to the project root directory
`<path-to-spring-app>/bb-pets-adoption`
3. Run -> mvn clean install
4. Run -> mvn spring-boot:run

4. Access the Application:

- Once the application is running, it will be accessible at <http://localhost:8080>.

3.2 Running the React Application Locally

Prerequisites:

- **Node.js and npm:** Ensure Node.js and npm are installed.

Steps:

1. Open your IDE to run javascript code
2. In the IDE terminal navigate to the React Application Directory:
`<path-to-react-application>/bb-pets-adoption\bb-pets-adoption-frontend`
3. Go to 'scripts' folder and ensure your axios config looks like this

```
const instance = axios.create({
  baseURL: 'http://localhost:8080',
  headers: {
    'Content-Type': 'application/json',
  },
});
```

4. Go to App.js file and ensure you top <Router> tag is set like this

```

  ✓ function App() {
  ✓   return (
    // <Router basename="/Frontend_React_Baby_Pets_Adoption_App">
    ✓   <Router>
      <ScrollToTop />
      <NotificationsProvider >

```

5. Run -> npm install
6. Run -> npm start

Access the Application: The React application will be accessible at <http://localhost:3000>.

5.SEO

Metatags

Head meta tags are key for search engine optimization (SEO) and to enhance user experience.

- Meta Description

```

<!-- Meta Description -->
<meta name="description"
  content="Adopt a pet with confidence through our trusted pet adoption services.
  Connect with pet providers, learn about the adoption process, and find your perfect pet today.">

```

It provides a brief explanation of what the site is about which helps search engines to know the topic of the content.

It also appears on the search listings and can be used to lure users with an appealing description.

- Meta Keywords

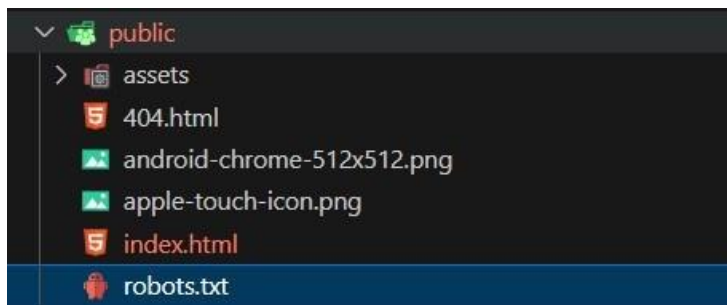
```
<!-- Meta Keywords -->
<meta name="keywords"
  content="adopt a pet, pet adoption, pet providers, adoption services, pet adoption process, adopt a cat, adopt a dog">
<link rel="apple-touch-icon" href="%PUBLIC_URL%/apple-touch-icon.png" />
```

Adding keywords to provide relevant sites' content which also helps the search engine for quickly identify the topic of the site, less significant nowadays with modern search engines.

It also helps to target user queries when looking for pet adoption services in this case.

Robots.txt file

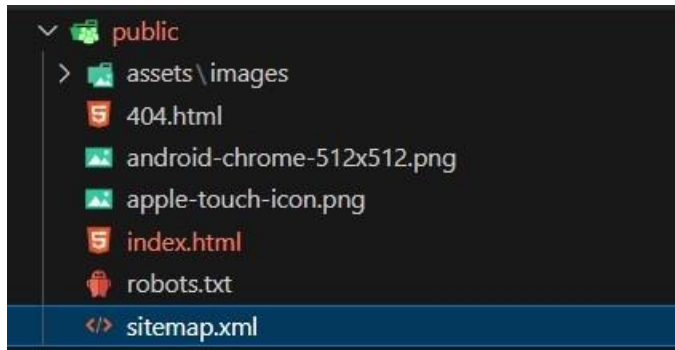
The robots.txt file is used to tell web crawlers (like those from search engines) which pages or files they can or cannot request access of your site. Mostly used to keep certain parts of site private or non-indexed.



Sitemap.xml file

The sitemap.xml file is an XML file that lists the URLs of the site. It allows to include information such as when an URL was last updated, how often it usually changes, and how important it is relative to other URLs in the site.

It helps search engines to crawl the site.



Meaningful keywords

Use HTML elements to provide meaningful keywords related to the site's topic or services.

Adopt a Pet with Confidence

We understand that people have specific pr

Adoption Process

- **Application:** Pet providers have the f
decision, we will ask for some informat
net providers can confidently hand ove

Welcome to **Baby Pet Adoption!**

At Baby Pet Adoption, we believe that every pet deserves a loving home and a companion. Our mission is to make pet adoption easy, transparent, and enjoyable.

6. Tests

While developing Backend side, I've been using **Logger class** to provide feedback on every step to report of outcomes and provide proper feedback on the logs for easy debugging when application is on production.

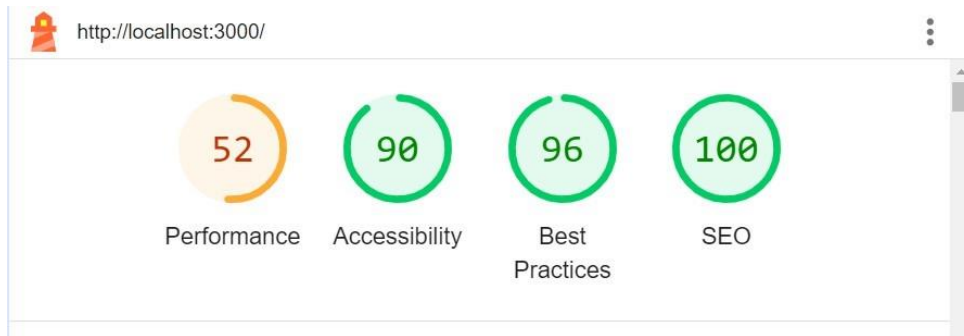
When testing Http requests, before the front end was available, and also when frontend components were available to speed things up, I've been using Postman and its feature of *collections* storing of http request for each of the different API service URLs

While developing the React frontend application, in order to monitor errors, the messages returned from backend app, and to view the structure of the data returned on the http responses as well as to try and style html elements on the fly, have been using Chrome DevTools.

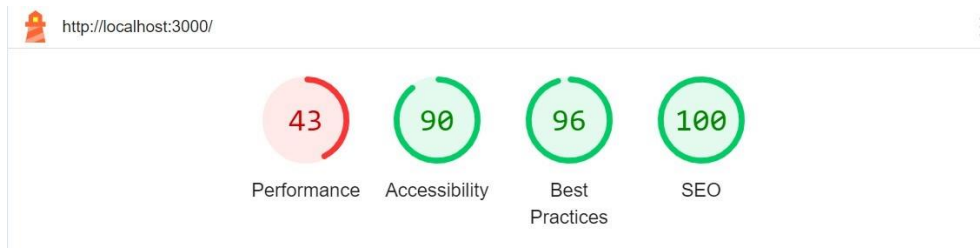
6.1 Manual Test

6.1.1 Lighthouse Browser Extension

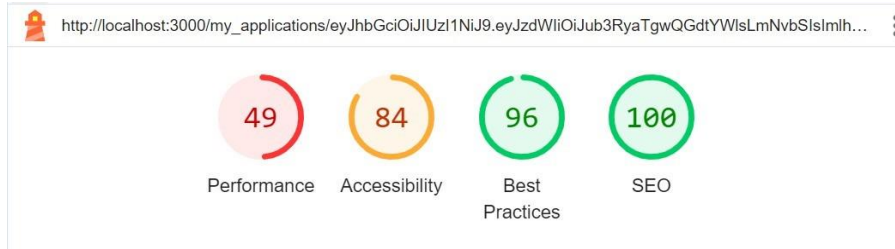
- **Desktop: Home page**



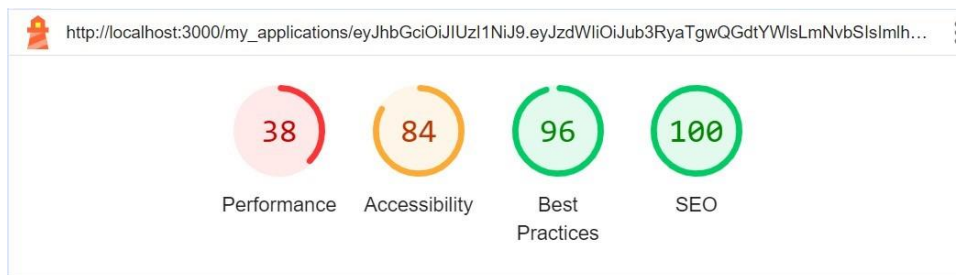
- **Mobile: Home page**



- **Desktop: My Applications page**



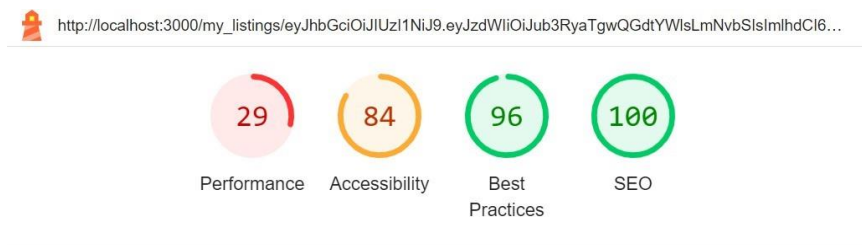
- **Mobile: My Applications page**



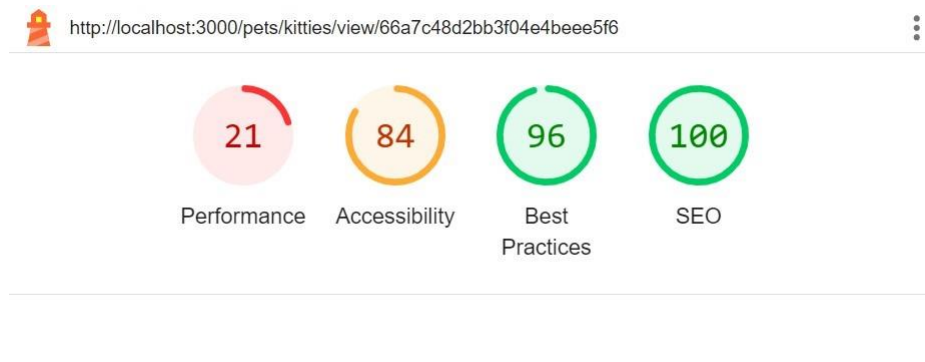
- **Desktop: My Listings page**



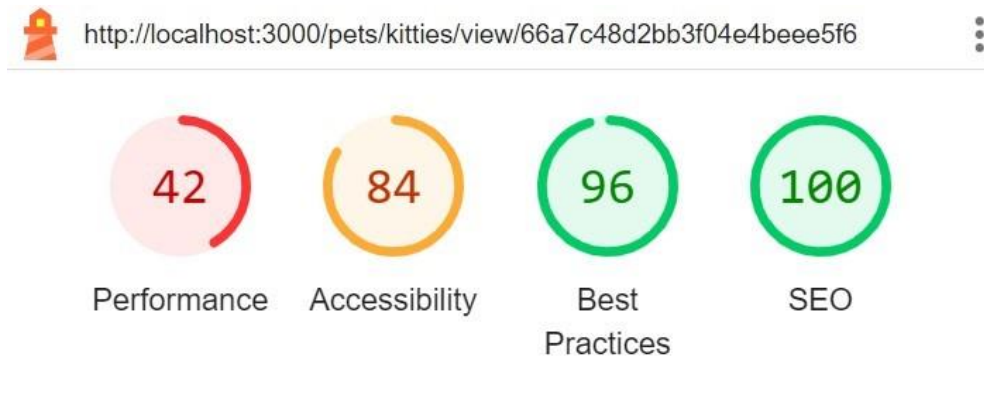
- **Mobile: My Listings page**



- **Desktop: Pet Details View page**



- **Mobile: Pet Details View page**



6.1.2 W3 Validator

- [W3 HTML Validator](#)

No html error were found

- [CSS Validator](#)

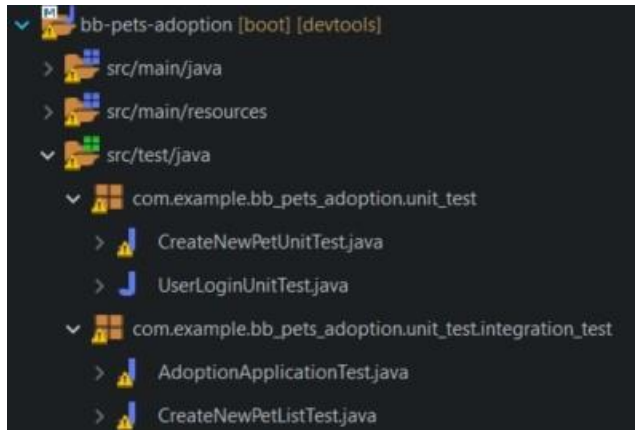
Some errors were found and corrected

6.2 Automated Tests

In order to test just the key functionalities of the appliation, I have created **Unit Tests** to test small parts in isolation, and Integration Test to test the interaction of multiple components to achieve a task.

Automated tests can be found in the project directory folder

Bb-pets-adoption/src/test/java/



6.2.1. Unit Test Specifications

- User Login

Unit Test--Test Case 1:

Title: User Login

Primary Actor: QA Tester

Description: Verify that a new pet can be created and saved in the database.

Pre-conditions:

- System up and running
- The application context is set up, and the necessary dependencies are injected

Test Steps:

1. Set up the test environment by creating a new pet instance with the required details
2. Save the pet instance using the `petServiceImpl`
3. Verify that the saved pet instance is not null
4. Check that the saved pet's details match the expected values

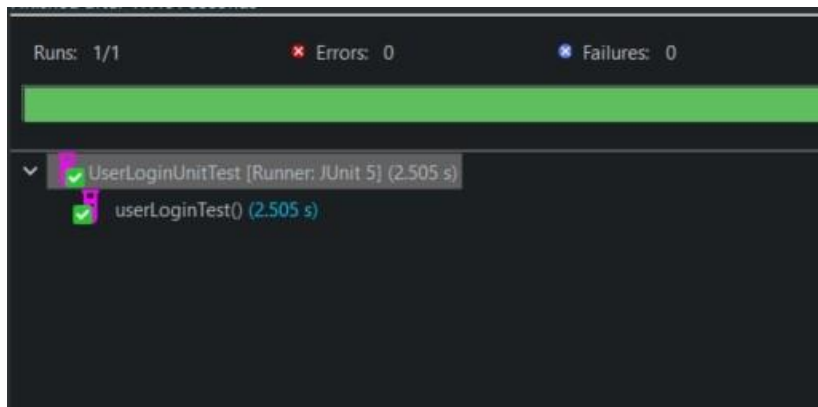
Postconditions:

- The pet instance is removed from the repository to clean up the test environment.

Result:

- PASS

Result



- Create a New Pet

Unit Test--Test Case 2:

Title: Create a New Pet

Primary Actor: User

Description: Verify that a new pet can be created and saved in the database.

Pre-conditions:

- System up and running
- The application context is set up, and the necessary dependencies are injected

Test Steps:

1. Set up the test environment by creating a new pet instance with the required details
2. Save the pet instance using the `petServiceImpl`
3. Verify that the saved pet instance is not null
4. Check that the saved pet's details match the expected values

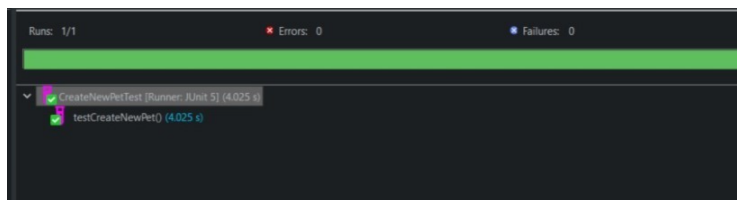
Postconditions:

- The pet instance is removed from the repository to clean up the test environment.

Result:

- PASS

Result



6.2.2 Integration Test Specifications

- Add a new Pet Listing

Components involved:

- Authentication Service
- PetListing Service
- MongoDB

Integration Test -- Test Case 1:

Title: `TC_CreateNewPetList`

Primary Actor: QA Tester

Description: Verify that a new pet can be created and associated with the authenticated user.

Pre-conditions:

- User exists in the system.
- User is authenticated, and a valid current session JWT token is given

Test Steps:

- Encode the password and save the user to the database.
- Authenticate the user and obtain a JWT token.
- Create multipart files for pet, mother, and father images.
- Make a POST request to `/pets/list_pet` with the required parameters and files.
- Validate the HTTP status of the response.

Expected results:

- The response status should be HTTP 200 OK
- The pet should be successfully created and associated with the user

Post Conditions:

- The user and associated pets are deleted from the database to ensure a clean state for subsequent tests

Result:

- PASS

Result



- Submit an Adoption Application

Components involved:

- Authentication Service
- PetListing Service
- Adoption Application Service
- MongoDB

Integration Test -- Test Case 2:

Title: Adoption Application Integration Test

Primary Actor: QA Tester

Description: Verify that a user can successfully apply for pet adoption

Pre-conditions:

- Network connection
- A user is logged in and a JWT token is obtained
- A pet is created and saved in the database
- A pet list is also created and saved in db

Test Steps:

1. Ensure the token is not null
2. Ensure the pet is not null
3. Perform a multipart POST request to the /adoption/apply endpoint with the necessary parameters, including the JWT token, pet category, pet ID, and location
4. Verify that the request is successful by checking the HTTP status.

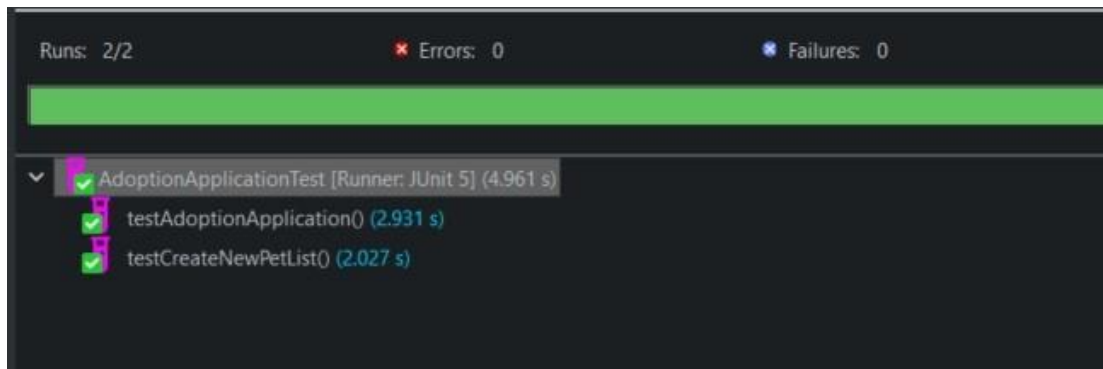
Postconditions:

- Postconditions
- The adoption application should be successfully created and saved in the database

Result:

- PASS

Result



THE END