**Assignment 01**

**Green Pastures Fostering**

**Group Project**

**Advance Database Development with Oracle APEX**

**MCOMD3ADD**

**BSc Computing Year 3 – 2020 – 2021**

**Lecturer’s Name:**

* Beth Cox

**Group Names (Blue team):**

* Jose Suan
* Asim Rai
* Chinduji Emereole
* Thomas Harrie-Thompson

**Submission Date:**

**xx/12/2020**

Contents of this report

[**1: Introduction** 4](#_Toc59300963)

[**2: How the team worked together** 4](#_Toc59300964)

[**3: Creating the Web Interface** 4](#_Toc59300965)

[**3.1: Creating population and creation scripts** 4](#_Toc59300966)

[**3.2: Managing users** 5](#_Toc59300967)

[**3.3: Creating / Designing the App** 5](#_Toc59300968)

[***3.3.1: Homepage Layout*** 6](#_Toc59300969)

[***3.3.2: User Interface (Theme, Icons, and Navigation Style)*** 7](#_Toc59300970)

[**3.4: Features** 8](#_Toc59300971)

[**3.4.1: Add new records** 8](#_Toc59300972)

[**3.4.2: View records** 16](#_Toc59300973)

[**3.4.3: Dashboard** 17](#_Toc59300974)

[**3.4.4: Approved and pending fostering list** 18](#_Toc59300975)

[**3.4.5: Edit records** 19](#_Toc59300976)

[**4: Limitation and Weaknesses** 20](#_Toc59300977)

[**4.1: Working on the same page at the same time** 20](#_Toc59300978)

[**4.2: NULL values** 20](#_Toc59300979)

[**4.3: Changing the primary key numbers** 21](#_Toc59300980)

[**4.4: Animal availability** 21](#_Toc59300981)

[**4.5: Foster application and foster tables does not update when changes made** 21](#_Toc59300982)

[**4.6: SQL Query** 21](#_Toc59300983)

[**4.7: Web server error** 22](#_Toc59300984)

[**4.8: Processing only accepts PL/SQL Expression** 22](#_Toc59300985)

[**4.9: Dynamic action does not work with Submit Page** 23](#_Toc59300986)

[**5: Improvements to Make** 23](#_Toc59300987)

[**5.1: Creating Default values for NULL values** 23](#_Toc59300988)

[**5.2:** **Application Status** 24](#_Toc59300989)

[**5.3: Date format** 24](#_Toc59300990)

[**5.4: Application Status** 24](#_Toc59300991)

[**5.5: Staff Overviewing Application** 24](#_Toc59300992)

[**5.6: Removed non-essential animal record** 25](#_Toc59300993)

[**5.7: Rename column headings to readable text** 25](#_Toc59300994)

[**5.8: Date Validation in Foster Form** 26](#_Toc59300995)

[**6: Testing** 26](file:///G:\University%20Work%20Year%203\1st%20SEM\Advance%20Database%20APEX\Assignment\Assignment%201\To%20submit\Group%20Report.docx#_Toc59300996)

[**Test code** 30](#_Toc59300997)

[**7: Conclusion** 34](#_Toc59300998)

[**8: Appendix** 35](#_Toc59300999)

[**8.1: User-Guide** 35](#_Toc59301000)

[**8.2: Logical Database Model:** 48](#_Toc59301001)

[**9: Bibliography** 49](#_Toc59301002)

[**10: Reference List** 49](#_Toc59301003)

**1: Introduction**

A fostering company called Green Pastures requires changing their fostering system to allow the staff to deal with their applications quicker. The company needs to build a friendly web Interface to their database in order to allow ease of managing the foster. This report contains evidence of creating a website application using Oracle Apex by using the existing database of the company. It includes how we manage to create the front-end interface, limitations and weaknesses discovered of the app and a conclusion to report for summarising the overall result of the development.

# **2: How the team worked together**

The team decided to divide the task before we start creating the web application. Each of the features of web app is assigned for each member. The following is the list of members with their assigned tasks:

1. Jose Suan – Homepage and Foster application record.
2. Asim Rai – Animal registration form, Animal table view, and Staff view.
3. Chinduji Emereole – Designing the web application and dashboard.
4. Thomas Harrie-Thompson – User guide and Foster Record.

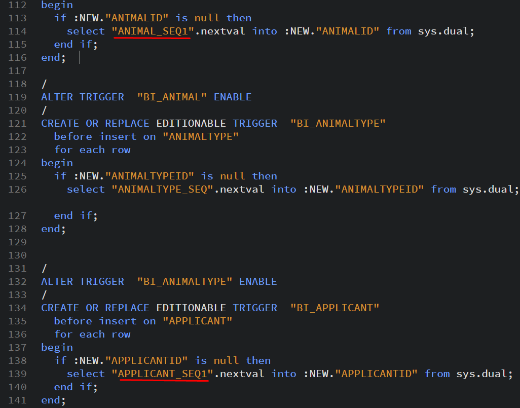
However, all members worked and help each other whenever we get stuck in a particular event.

# **3: Creating the Web Interface**

## **3.1: Creating population and creation scripts**

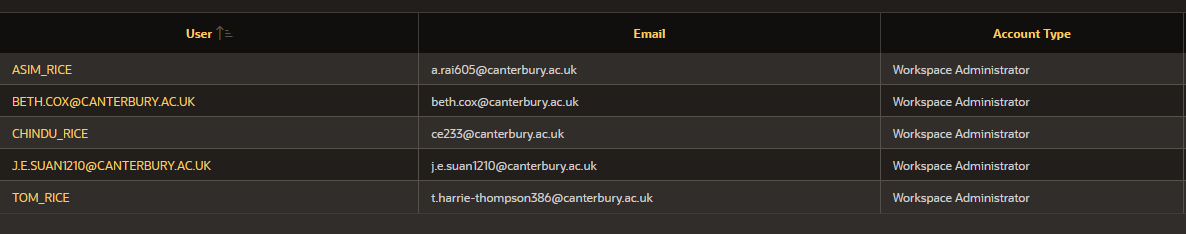
The team started to add the existing scripts that the Green Pasture provided (Population and Creation) and found how to add the scripts using oracle apex under SQL workshop/Scripts. However, an error occurred when we ran the creation script. The error was in the CREATE OR REPLACE EDITIONABLE TRIGGER code starting from line 114. The naming by calling the sequence for this conditional statement is incorrect so the team fixed it by naming the correct sequence name such as “ANIMAL\_SEQ and APPLICANT\_SEQ”.

The following is a screenshot with a red underline of the incorrect code.



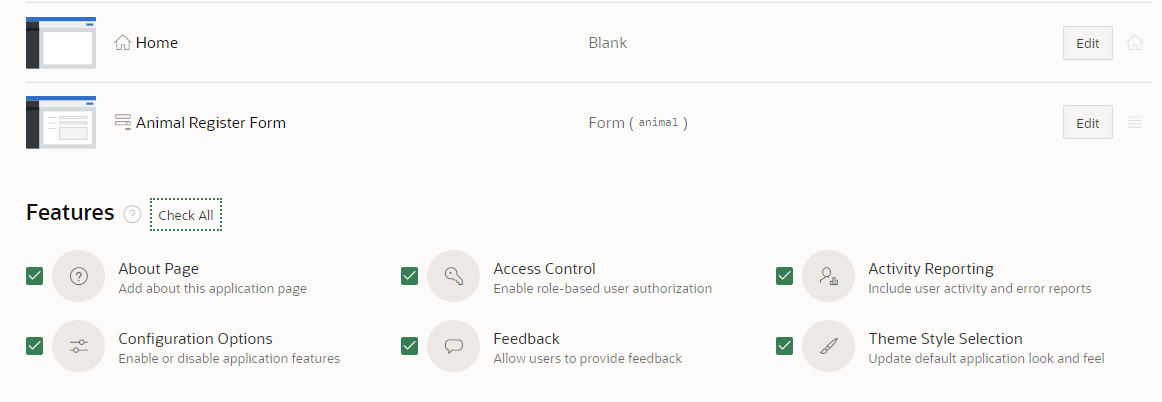
## **3.2: Managing users**

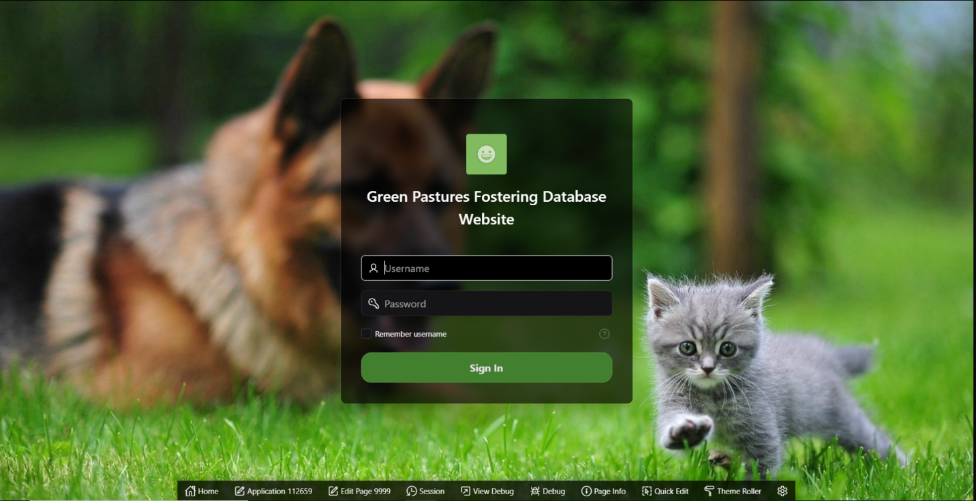
The team decided to use one workspace named “suan\_workspace”. Each member of the team has a user account and each account is set as a Workspace administrator to be able to share and have full control when creating the website.



## **3.3: Creating / Designing the App**

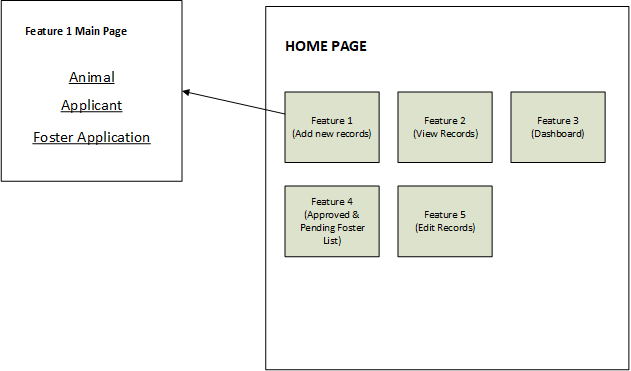
The front-end application was created by using App Builder from Oracle Apex. Homepage and animal registry were the first two pages that were made at the start and all the six optional features were checked which included (About Page, Access Control, Activity Reporting, Configuration Options, Feedback and Theme Style Selection). These features are included for future improvement for the web app. (example shown below)



The login page has a background image which appears below:   
The reason why the team has decided to include a background image on the login page is to make the app more user friendly for the staff members.

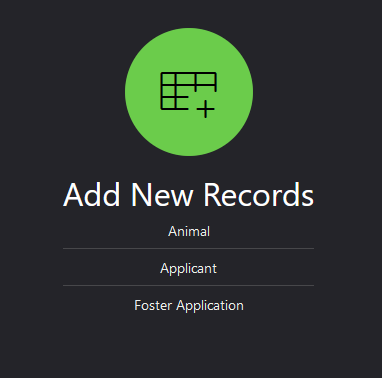
***3.3.1: Homepage Layout***

After creating the registry pages with features, the team started to create different pages to place the content requirements. The sample layout of how pages will be navigated and accessed for the user was made to help create further pages.



On the home page, there are five navigation lists and each of the lists are features of the app which is followed based on the system requirements. Feature 1, 2 and 5 have their own main pages to be able for the user to choose what specific record or table they wish to update. For example, feature 1 purpose is to add new records and show three links to choose what table the user is interested in.

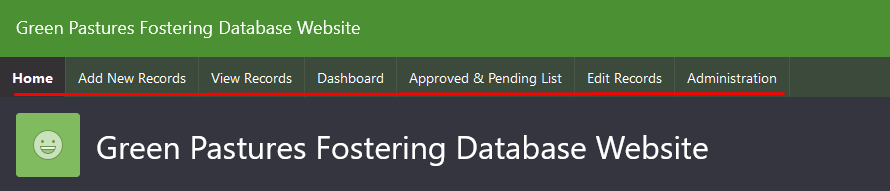
Here is an example of feature 1 main page, where users select three types of forms to add new records from Animal, Applicant, and Foster Application:



### ***3.3.2: User Interface (Theme, Icons, and Navigation Style)***

The team decided to use a dark and green theme for the app website. The colour green is chosen to follow the company name “Green Pasture fostering” and theme. Dark colour is used because it helps users to read the contents easily. Additionally, the white text on a dark background colour is easy on the eyes and stands out.

Top navigation menu is used for users be able to click and quickly navigate important pages such as the five features from any page without the needing to go back to the homepage.



Each feature of the homepage contains an icon that is related to the function of a particular feature. The team decided to add icons to help users able to quickly and intuitively recognise and find a particular feature by using a universally recognised icon for the corresponding feature.



## **3.4: Features**

This part explains how the team created the five features of the web application: Add new records, view records, dashboard, approved & pending list and edit records.

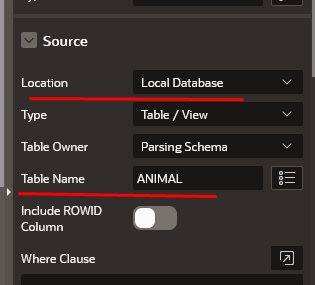
### **3.4.1: Add new records**

To add new records for the user, the team has used a region type form. There are three forms in Add new records page: Animal, Applicant and Foster Application. These three forms are the main requirement functionalities that the company wanted to add to their web app.

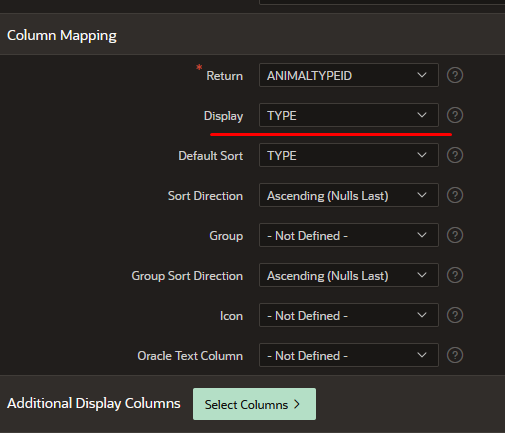
* **Animal form**

Creating the animal form in oracle apex requires managing the *source*. From the *source* we assign the location of the table, which is a local database, and specify what table in the database. Apex will then create default items for the fields needed to submit a record into the table, which we customise for a unified design, this also ensures that the items submit their data to the correct fields in the table.

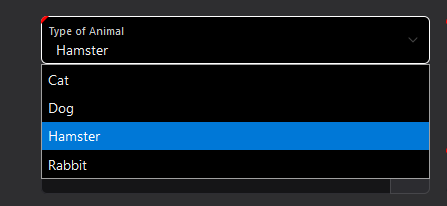
For example, the page designer of creating animal form uses a local database source named ANIMAL:



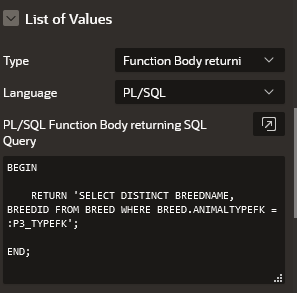
The team decided to change the view of selecting the “Type of Animal” and “Type of Breed”. Originally, these two field items are select list type items where users can click and choose a particular record from an expanded list. However, each record only shows the IDs of type animal and type breed which is not very human readable compared to the names assigned to the IDs in other tables. Using a select list will not provide the names for the IDs because these fields are foreign keys of Animal table.

To make it easier for the user to read and choose the right data, the team made a custom list to display the type but return the ID of that type, this was done easily through *column mapping*. We used the custom list of values in the select list to display in the readable format while still returning the ID when submitted, without the user needing to learn or remember the IDs and their meanings.

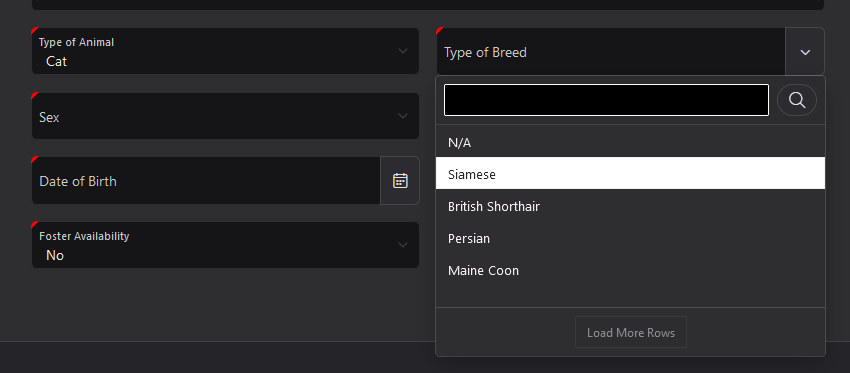
After applying the changes, this is how selecting the type of animal looks like:



For selecting the Type of Breed, we encountered an issue that the select list displays all the breed types even when the breed does not exist for the chosen animal. For example, when selecting Cat, we don’t want to display the list of breeds for dogs such as Rottweiler etc. To sort this issue, creating PL/SQL function body returning SQL query is made.

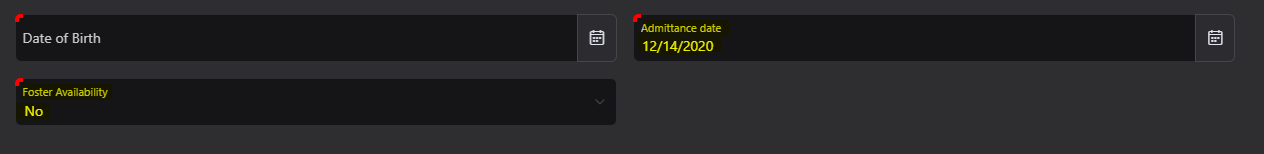
In this script code, it only displays the BREEDNAME from BREED table which a condition is applied where the BREED table’s ANIMALTYPEFK field is the same as P3\_TYPEFK’s value which is the item containing the animal type selection.

Here we have an example that when a cat is chosen it only shows all list of cat breeds:

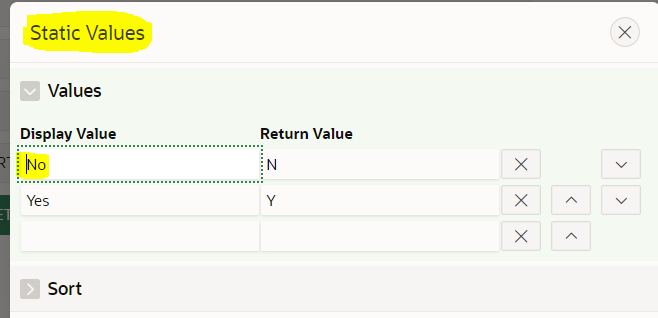


***Default Values***

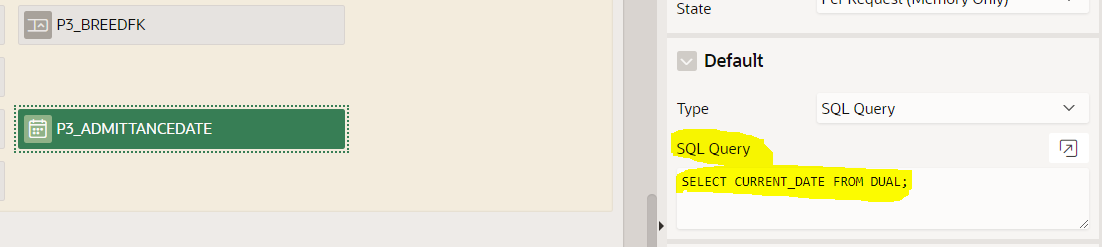
When registering new animals, the default admittance date is set to the current date and foster availability is set to No. These values can be changed if needed by dropping down the text boxes.



The Foster availability has No as the first default value, this is done by selecting the select list “P3\_AVAILABLETOFOSTER” from the page designer and having No as the first value on the List of Values, Static Values.

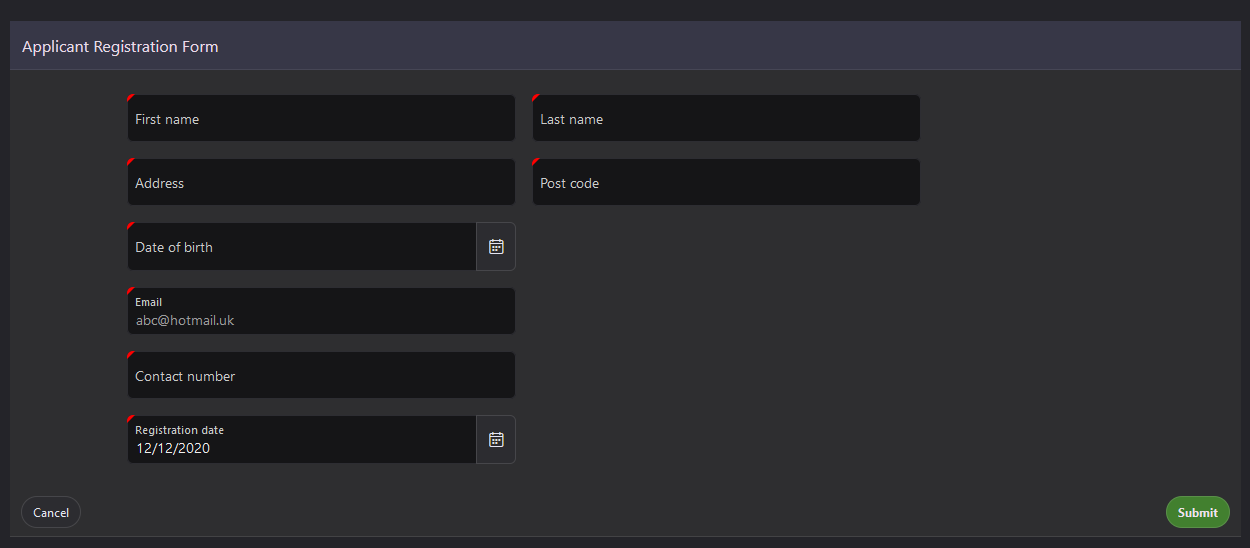


For the admittance date it uses SQL Query to get the current date. (Example below.)



* **Applicants form**

Creating the applicant form is the same as when creating the Animal form. We have added all the required fields for the applicant.

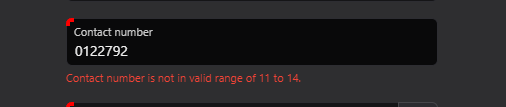


It is good to mention that all Date of Birth fields are validated to choose only the current and the past dates, not the future dates. The screenshot below shows that 13 of December and so forth are disabled to be chosen to get valid data.

***Age limit validation***

The date picker for the applicant’s date of birth excludes all dates that are less 18 years before the current date, therefore, all new applicants under the age of 18 years cannot register. [foxinfotech2014](https://www.facebook.com/foxinfotech2014) (2020)  
This was done by restricting the year range by –18 as shown below:  
  
This year range enables the user to select a maximum age of 100 years and a minimum age of 18 years.

**Contact Number Validation**

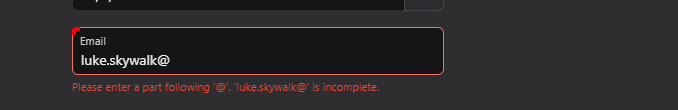


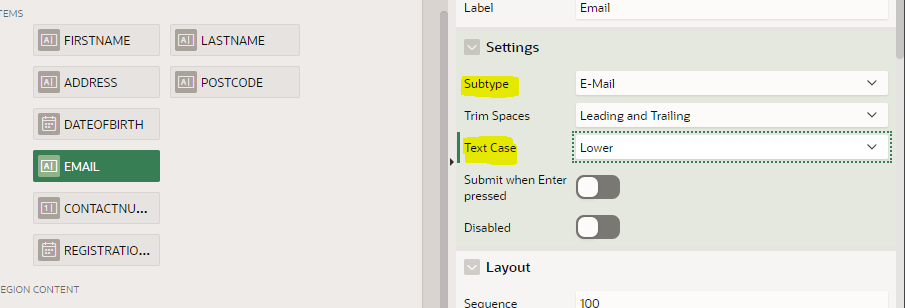
An error message will be displayed when the contact number is outside the range of 11 – 14 numbers.

The validation is done by setting the max and min values in the settings of the edit page as shown on the picture below.



***Email Validation***

The email address needs a proper format to be accepted by the system as you can see from the figure above. This is accomplished by declaring the text field as a subtype email and making the text case lower case as the picture shows below.

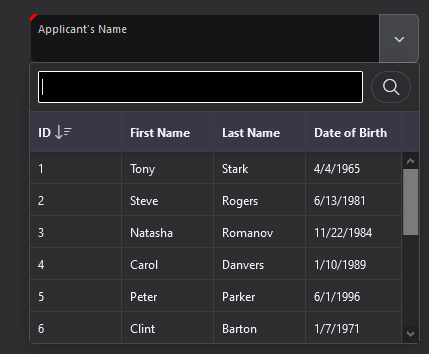


* **Foster application form**

Creating the foster application form has the most time spent as it comes with three important functions that are examples of the system requirements. These three functions are: adding new records into the FOSTER\_APPLICATION table, updating the animal’s availability to N once an application is successful and creating a new record in FOSTER table once the application status is Approved or Pending.

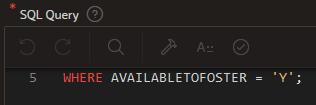
***Displaying drop down record list***

The team changed the display list for the Applicant’s name with additional column fields such as the ID, list name, last name and date of birth. With the information from additional fields, we can identify an applicant without non-human readable identifiers like ApplicantID regardless of if the applicant has fields matching other applicants such having the same first name or birthday. The same has been done for choosing an animal where now, in addition to being able to now see its name, you can see its availability.



***Displaying all available Animals***

When selecting an animal, users only need to see and choose from the list of animals that are available. In the animal table, available animals have 'Y' in the AVAILABLETOFOSTER field. Displaying the available animals is done by creating a SQL Query to the oracle’s *component list* ANIMALLIST\_AVAILABLE.



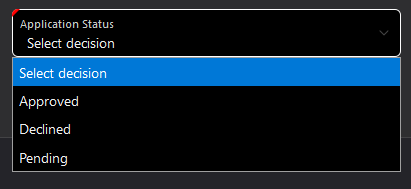
***Date of application***

In this form, the date of application’s value is set as current date. Setting the value of the date current is by creating a SQL query and querying CURRENT\_DATE from DUAL. www.w3schools.com. (n.d.). Having the date value set as current date helps user to quickly finish filling the form without the need of selecting the date.



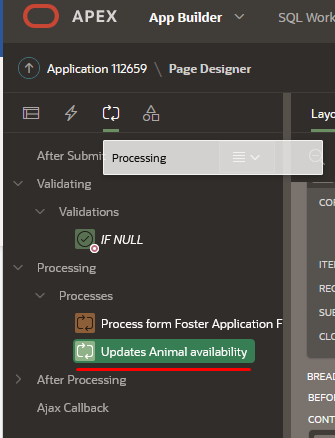
***Application status***

The team decided to add two additional default values “Declined” and “Pending”, in order to give the staff additional options to decide a concrete action for the application rather than leaving it as NULL. The team believes that adding NULL values into a record may create a misunderstanding for the staff when they view the records because the staff may presume that an application record may be pending or unintentionally forgotten to add the status.

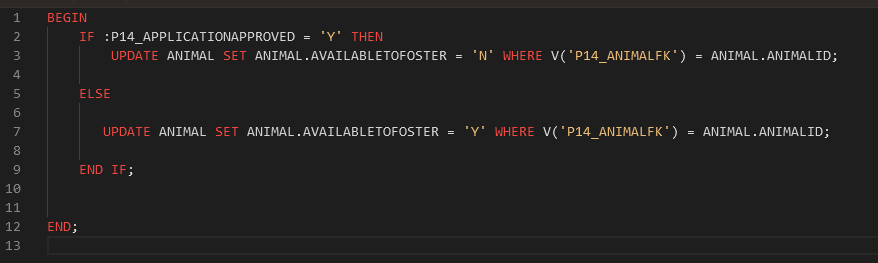


***Updating the animal’s availability***

Once the application form is submitted, the chosen animal updates its availability to ‘N’. To do this, creating a processing function for the page with PL/SQL scripts is a way to update the selected animal’s availability. Below is a screenshot of the sample process:



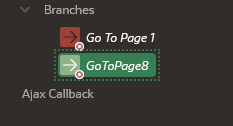
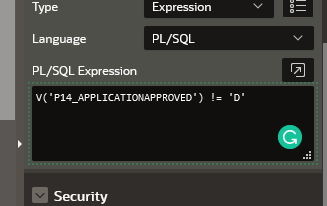
Inside this process comes with a PL/SQL code If statement that check’s if the application is approved or not. It checks using the item P14\_APPLICATIONAPPROVED value, then if the value returns as ‘Y’ it updates the animal available foster to ‘N’ with a WHERE statement to match the selected item P14\_ANIMALFK to ANIMAL.ANIMALID primary key.



***Foster form***

Once approved or pending, creates a copy of records to foster table. In order to do this, the team decided to create two *Branches:* Go to Page 1 and Go to Page 8. These two branches have their own PL/SQL code that checks whether the application status is approved.

This is done by creating *Branches* and two of these branches have their own PL/SQL code that checks if the item P14\_APPLICATIONAPPROVED is ‘Y’ or not. If the application is approved or pending the form will be submitted to the FosterApplicaion table and the user will be returned to the home page. If the application is denied, the form will still be submitted to the FosterApplication table but instead of going to the home page, the user will be taken to another page to input the rest of the details required to submit a form to the Foster table.

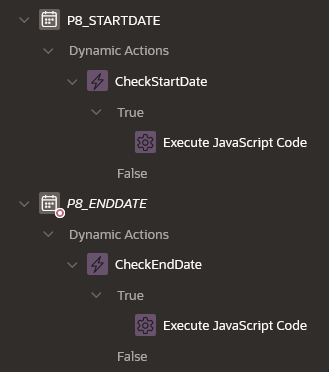
The Foster form on Page 8 is only accessible though the branch in the Foster Application from. The foster form takes the ID from the just submitted foster application and contains input fields for the rest of the required data to submit to the Foster table and when the submit button is pressed will create a record in the Foster table.

We get the Application ID through a page load event that sets the value of for the application ID item to the highest id record in FOSTERAPPLICATION table as this would be the id for the newest record added and would be the one we just submitted as we get taken to the foster form only when the foster application is approved when submitted.

On the foster form, Start date and End date are provided for the staff to choose the required dates. However, we have found that these two dates have the possibility to enter the wrong data, for example, the *start date* could be set as after the *end date* or the *end date* could be set as before the *start date*.

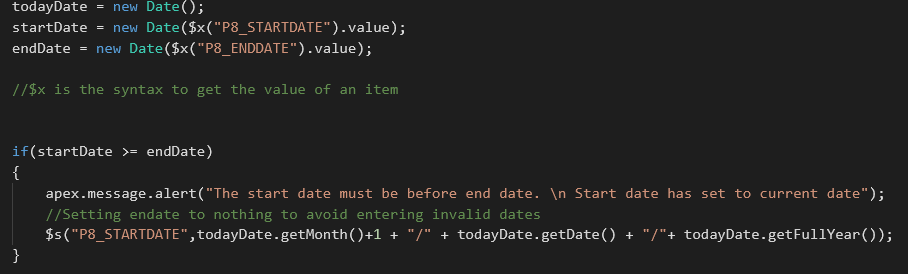
Because of it, the data entered would be illogical because we want to make sure that the start date must always be before the end date.

To avoid getting the wrong date, we have added *Dynamic action* or events to each of the date. For example, below shows that two start date and end date have their own event.

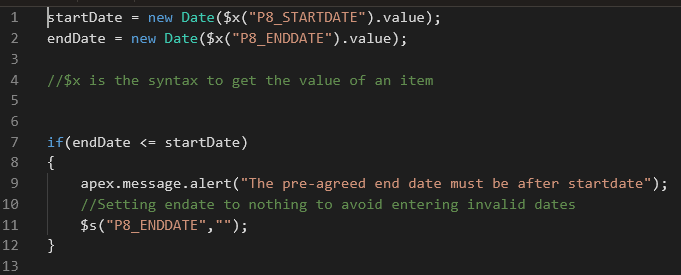


The language used here is JavaScript Code and the team have used the Date class to use dates and Conditional statement to compare both dates while not returning illogical data. The code uses get and set item value using JavaScript. Oracle SOY. (2020) The following are the two screenshots of each validation:

***Start date***



***End date***

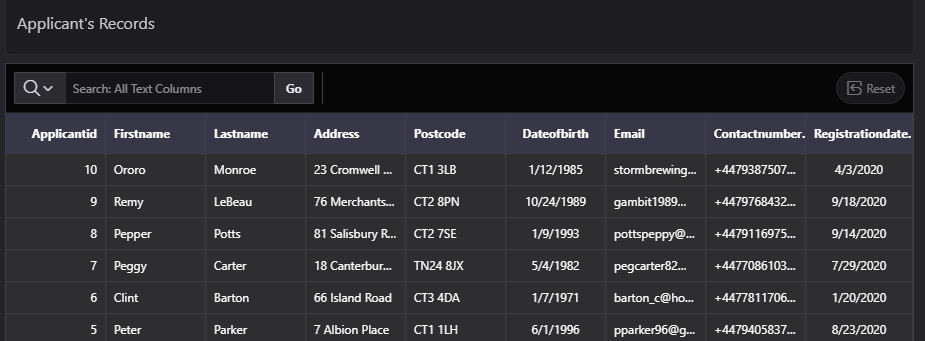


### **3.4.2: View records**

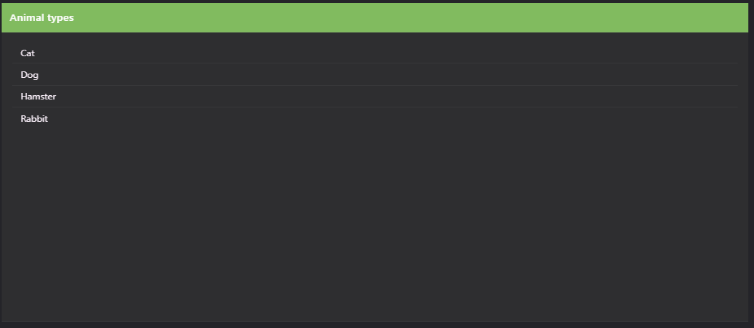
Another feature of the website is “View Records”. View records function is to view all records for Animal, Applicant, Staff and Application Foster, which is part of the system requirements. The view records page contains four links for users to choose what records they want to view.

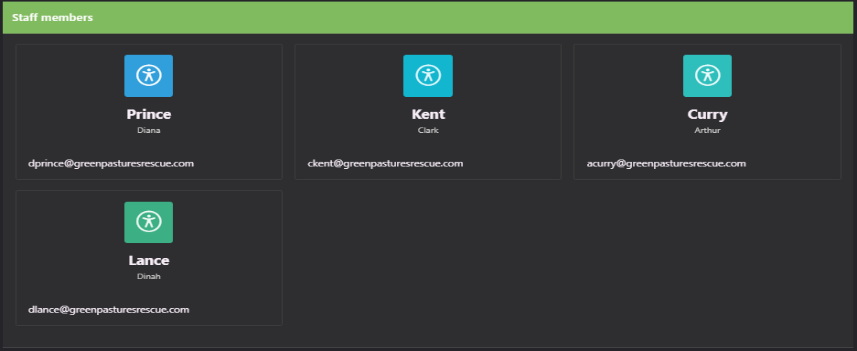


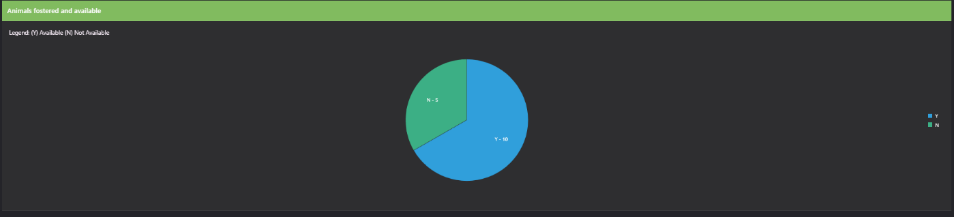
Each of the links are pages with *Interactive grid* to display all records from a particular table. This type of region is used because it already comes with a *toolbar* such as Search field to help user to find a specific record easily. (example of the view table below)

The view record pages are organised based on the most recent record, being at the top of the list. It helps the staff to check records quickly. This is done by ordering the id in a descending order. (Example below)



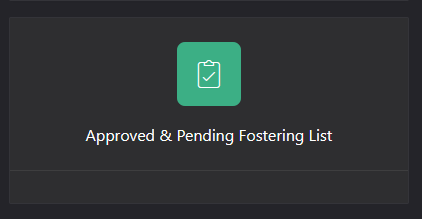
**3.4.3: Dashboard**  
 The dashboard is not essential for the app; it is an extra feature that was added to improve the user experience. The dashboard is there to improve the app in a way where it enables the user to easily see data and seeing it in a more visual way.  
  
The dashboard contains a view list that displays all types of animals the center fosters.   


It also contains a card view that shows staff members details such as their name, surname, and email address.   


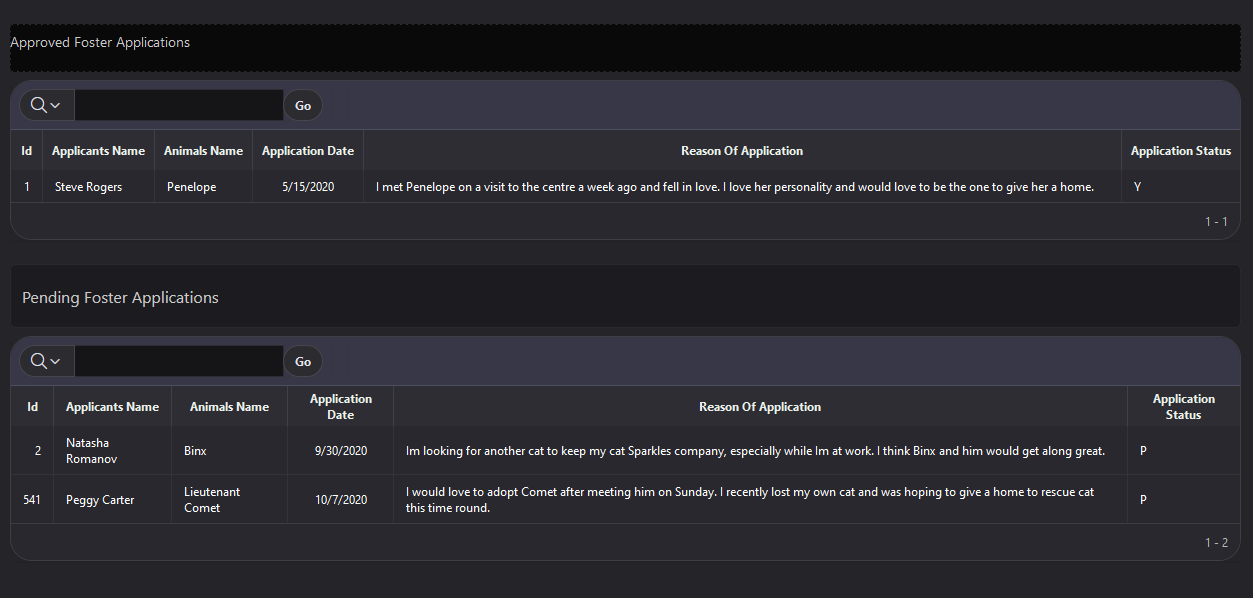
The last element of the dashboard is a pie chart that demonstrates the number of animals that are not available to be fostered in blue, and the number of animals that are available in green.   


### **3.4.4: Approved and pending fostering list**

This feature shows all the approved and pending application. It is made as an interactive grid report and comes with a search box to help users find specific record once the table has many records in the future.

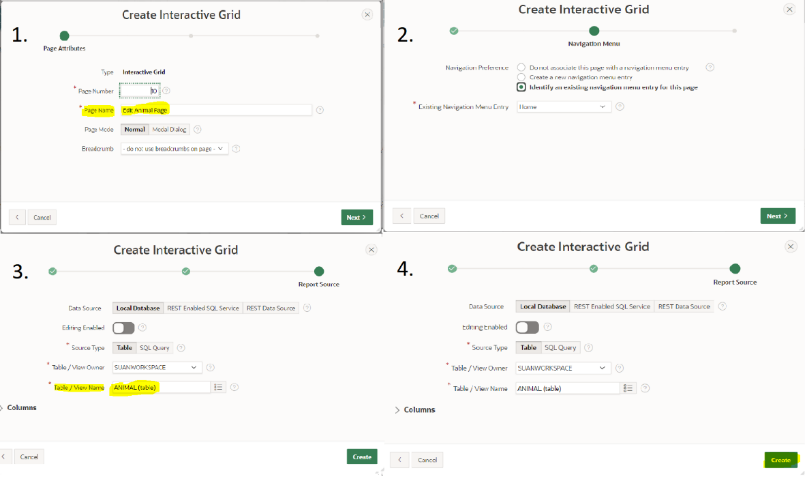


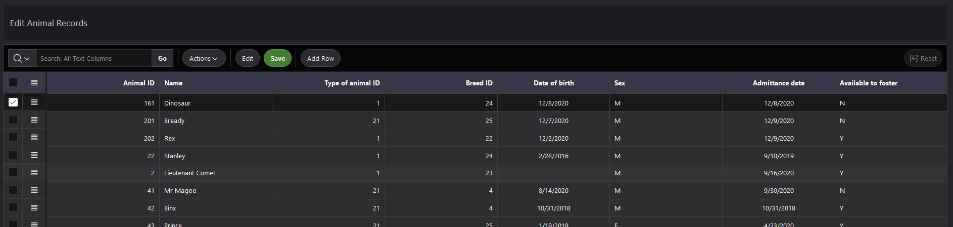
The top table shows all approved application while the bottom ones are the pending.



### **3.4.5: Edit records**

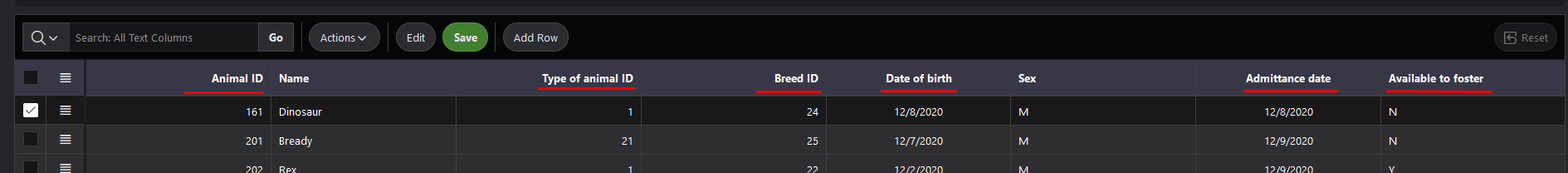
There are 7 Editing pages in the application, and they are all created by using Interactive grids*,* linking each of the grids to the corresponding tables in the App Builder. The image below shows an example of creating an editing page for the animal table. The process needs the name of the page on the 1st step and the table name on the 3rd step to complete the web page. The page will be created after pressing the create button as highlighted on step 4th:





The App Builder then creates the above page for editing tables. The page consists of rows and columns of the corresponding table with buttons and options to edit, sort, delete and add new rows. These options help to delete wrong information and make the necessary updates to the existing records.

The team has also changed all the headings of the column fields to readable text. For example, instead of *Breedfk* we have changed it to Breed ID and other fields have been added spaces for ease to read. This applies to the other editing pages as well.



A quick selection as well is added to some of the column fields such as the *Available to foster.* Staff can only select Y or N to avoid adding different data or leaving it blank.



# **4: Limitation and Weaknesses**

During the creation of the web app for Green Pasture Fostering, the team found limitations and weaknesses. The following are example points with explanation and screenshot of the limitations.

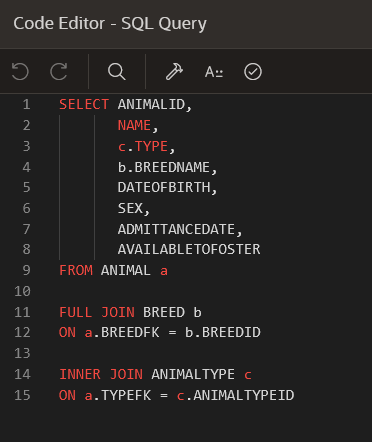
## **4.1: Working on the same page at the same time**

Oracle Apex is not suitable when multiple people are working on the same page at the same time.

The team face issues when working on the same page of the web app, this causes us some errors to occur and stops us from checking if the code or design has made any changes. However, to lessen the occurrence of this limitation, the team decides to divide each member’s work on a particular page of the web app.

## **4.2: NULL values**

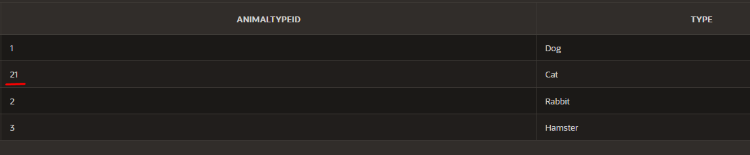
The population scripts contain fields that are nullable such as table Animal field: BREEDFK and DATEOFBIRTH. Because of this, when creating query to view all the records using INNER JOIN does not show all records with null values. However, the team manage to find another way to show null values by using FULL JOIN as it includes all values. The following screenshot is an example of SQL Query where BREED table is FULL JOIN with ANIMAL table:



## **4.3: Changing the primary key numbers**

The team noticed that on the ANIMALTYPE table, one of the animal types is id 21 which is the cat. The team wanted to change this primary key to 4 to make the data organise. However, changing this primary key violates the relationship of other tables, such as ANIMAL, BREED and FOSTERAPPLICATION that has cat, creates an error.

The following is the screenshot of the primary ID key:



## **4.4: Animal availability**

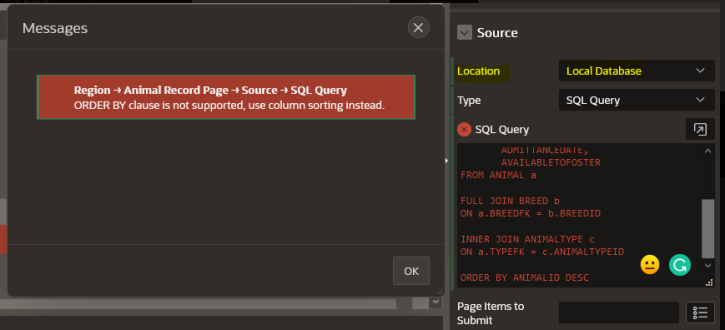
* If foster application record is deleted, the animal selected does not automatically updates its availability to ‘Y’. This would create an issue because all animals when finished fostered will remain as ‘N’ not available, which is not easy for the staff to manage especially when the foster application creates more records at some point.

## **4.5: Foster application and foster tables does not update when changes made**

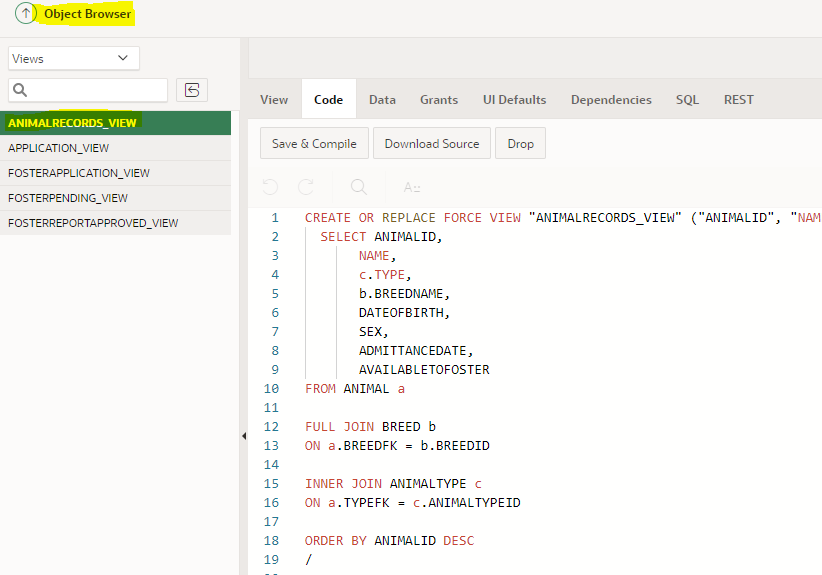
* If deleting a record from foster application that is an approved or pending status, this does not update the record in foster table. This would be an issue because when staff checks from the foster records, they may not be able to find the application record as it was been deleted.
* When a foster application record is pending and change it to Y, does not validate whether the animal chosen is available to foster.

## **4.6: SQL Query**

* There are limitations depending where the script is placed, for example of the view script located on the Local Database. This caused an error because the ORDER BY statement is not supported in the design page SQL Query.



Whereas there are no errors when the same script is used on the object browser, which we have to create views.



## **4.7: Web server error**

A web error occurs sometimes such as 502 Bad Gateway and Failure of Web Server bridge have occurred trying to access Oracle Apex website. This hinders the ability to manage the database, and without any diagnostics help it is very inconvenient. In addition, it stops the team from finishing or updating the web app quickly.

The following screenshot of the error:

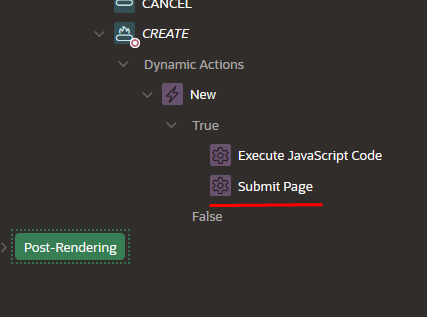
## **4.8: Processing only accepts PL/SQL Expression**

## 



Processing in the page designer is limited to PL/SQL. This means there is no ability to use JavaScript to validate the page as it only allows the use PL/SQL language. JavaScript is client-side which cannot interact with the server side and vice versa with the PL/SQL.

## **4.9: Dynamic action does not work with Submit Page**



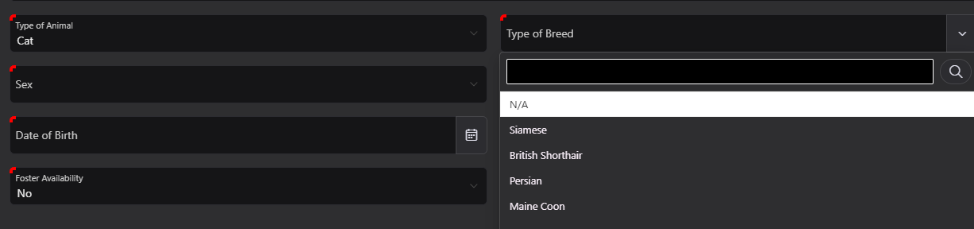
Submit page does not work with Dynamic action because the conditions will not work execute. This causes problems saving the form as the page will just transition into the homepage without saving the details.

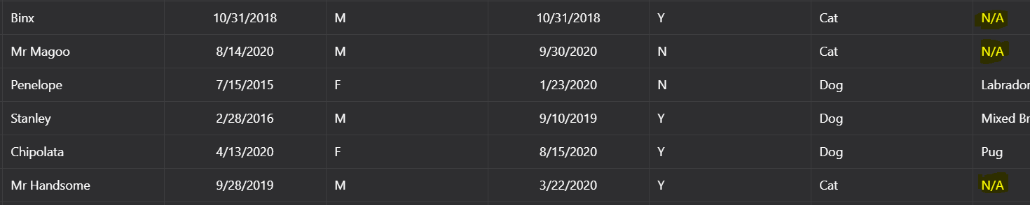
# **5: Improvements to Make**

The following are some of the improvements Team Blue thought would make the application better.

## **5.1: Creating Default values for NULL values**

When selecting a type of breed, the team found out that some records of the Animal does not have any breed. If the Breed is not known, using default value N/A will be used to help the staff to choose.

Here is an example of how N/A is used: Breed name has 3 null values from the original population script which caused errors in the animal view table. The nullable values are now changed into N/A to avoid any errors and confusion for the staff to read the record.



## **5.2:** **Application Status**

There are 3 application statuses that are given to foster applicants. D (Denied), P (Pending) and Y for (Approved). This helps clarify the application’s status instead of having empty fields.

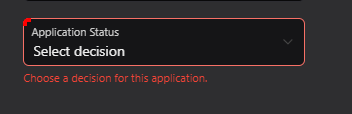


## **5.3: Date format**

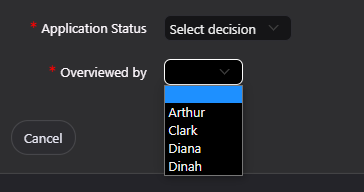
Currently the dates are in US Format which might be confusing since Green Pasture is a UK based fostering company. The Date format should be DD/MM/YYYY. 

## **5.4: Application Status**

If no Application Status is selected it does not allow the application to be completed in the Foster Application Form.



## **5.5: Staff Overviewing Application**

In the Foster Application Form “Overviewed by” section, it should assign the staff automatically depending on which staff is logged in and is completing the foster application form. 

## **5.6: Removed non-essential animal record**

Chinchilla is removed from the animal types because Green Pasture’s only fosters Dog, Cats, Rabbits and Hamsters. No existing Chinchilla are in the fostering records so the database does not need a record for Chinchillas. 

## **5.7: Rename column headings to readable text**

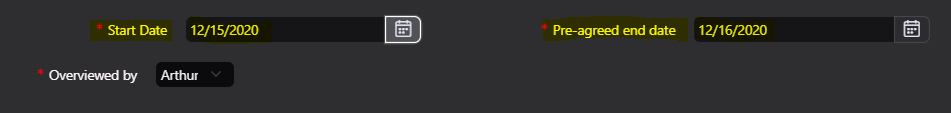
The team changed all the column headings for all views and reports tables so that staff would be able to understand and read each piece of information. Here is an example of an Animal records:

|  |  |
| --- | --- |
|  | Heading Animal Id,  Date of Birth has space so it is more readable. |
|  | The headings remain readable throughout the table. |

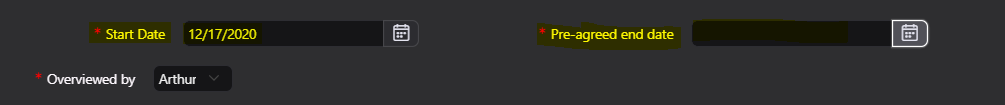
## **5.8: Date Validation in Foster Form**

The user will be prompted with error messages when the start date goes over the pre agreed end date or if the pre agreed end date is the same or less than the start date. This validation is there to help avoid any potential mistakes when inputting dates as the end date must be at least a day over the start date for the staff to be able to complete the form.

|  |  |
| --- | --- |
| Error Message 1 | Error Message 2 |

In the event of the start date going over the end date, it will prompt an error message (Error Message 1) and set the start date back to the current date. In the example below the start date was put as 12/17/2020 which exceeded the Pre agreed end date this resulted in the start date reverting back to the current date 12/15/2020 followed by Error Message 1 from the above table. 

In the event of the end date going below or being the same date as the start date, it will prompt an error message (Error Message 2) and set the pre- agreed end date to null. In the example below the pre agreed end date was put 12/16/2020 which is less than the Start Date this resulted in the pre agreed end date turning null followed by Error Message 2.



**6: Testing**

**This testing segment tests certain functionalities of the app related to the requirements that the app had to meet:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No. | Prerequisite | Steps | Input data | Expected output | Actual output | Pass/Fail |
| 1 | Staff need to access the system. | **1.** Access  to the login page **2.** Enter Username and password | -” **Username**” -” **password**” | Access approved | Access approved | PASS |
| 2 | Staff members will need to add/update/remove animals from the fostering list | **1.**On the website,  click on edit records.  **2.** Click on “Animal” **3.** Select a row to edit **4.** Double click on a textbox to edit it | Edited name:  **Rex** | Displays Rex | Rex | PASS |
| 3 |  | **1.**On the website,  click on edit records.  **2.** Click on “Animal” **3.** Select a row to edit **4.** Click on this icon to add new animals | Added animal: **Prince** | Add a new animal:  Prince | New animal: Prince | PASS |
| 4 |  | **1.**On the website,  click on edit records.  **2.** Click on “Animal” **3.** Select a row to edit **4.** Click on this icon  to delete an animal | Deleted animal: **Ajax** | Delete Ajax | Ajax has been deleted | PASS |
| 5 | ID’s cannot be edited | **1.**On the website,  click on edit records.  **2.** Click on “Animal” **3.** Double click on a Id column | Edit column Id | cannot be edited | Data remained unchanged | PASS |
| 6 | The centre fosters dogs, cats, hamsters and rabbits. | **1.** Go to “add new record” and then go to “animal” **2.** Select a type of animal | / | Only shows 4 types of animal to add | Displays 4 different types of animals | PASS |
| 7 | Applicants need to be created, applicants may, and are allowed to, foster many animals | **1.** Go to “add new record” and then go to “Applicants” | New applicant’sname:  **Tony Stark** | Tony Stark information should be entered into the system. | Information entered into the system | PASS |
| 8 | **1.** Go to “add new record” and then go to “Foster application form” | The applicant’s name selected: **Tony Stark** Animal selected: **Rex** | Tony Stark should be added on the “applicant’s name” combo box and can able to foster Rex | Tony Stark can be selected and is capable of applying to foster Rex | PASS |
| 9 | Application details from applicants need to be entered into the system, applicants will detail which animal they are interested in fostering and are required to give a reason why they should be allowed to foster that particular foster. | **1.** Go to “add new record” and then go to “Foster application form” | / | Being able to select an applicant, an animal to foster andgive a reason why the applicant wants to foster. | Textbox to select an applicant, to select an animal and to give a reason | PASS |
| 10 | When an application is approved it needs to be linked to a foster record, we only want to see current foster and pending applications | **1.** Go to “add new record” and then go to “Foster application form” | Fill out form with the application status set as “**Approved**” | Redirects to another form that be linked to foster details | Redirects to another form that will enter data into foster records | PASS |
| 11 | Fill out form with the application status set as “**Pending**” | Redirects to another form that be linked to foster details | Redirects to another form that will enter data into foster record | PASS |
| 12 | **1.** Go to “Approved & pending fostering list” | / | Approved and pending applications should be displayed. | Approved and pending applications are displayed. | PASS |
| 13 | Some animals will not be ready to be fostered, but it should be possible to update them for when they are ready. | **1.** Go to “Edit record” and “Animal” | Set “Bread”’ availability to **N** | Changes should be saved and entered into the system. | Changes successfully saved | PASS |
| 14 | Each foster will have a member of staff who overviews the whole thing and acts as a contact for the person fostering. | **1.** Go to “add new record” and then go to “Foster application form” **2.** Fill out form and submit it. | Select  **Arthur** to overview the foster application | Information enters the system. | Information succefully entered into the system. | PASS |
| 15 | Leave “Overviewed by” **\*Blank\*** | Should be denied by the system | Denied by the system | PASS |
| 16 | The applicants must be over the age of 18 to register | **1.** Go to “add new record” and then go to “Applicants” | Date of Birth: **15/03/1999** | Information should be succefully added | Information successfully added | PASS |
| 17 | Date of Birth: **19/07/2008** | Should be denied by the system | Denied by the system | PASS |
| 18 | Date of Birth: **15/07/2088** | The date should not be able to be selected. | Date cannot be selected | PASS |
| 19 | Date of Birth: **25/06/1208** | The date should not be able to be selected. | Date cannot be selected | PASS |
| 20 | Applicants must provide their address, email and a contact number. Their registration date should be defaulted to the current date if not provided in then applicants details. | **1.** Go to “add new record” and then go to “Applicants” | / | It should contain texboxes to input applicants details and the registration date should be defaulted to the current date | The registration date is defaulted to the current date and the all the texboxes to enter applicants details are there. | PASS |
| 21 | Staff profiles only need to include their first and last names and their email address. | **1.** Go to “View records” and “Staff” | / | Should display Staff members details | Displays staff members details | PASS |

**Test code**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test N°** | **Prerequisite** | **Code** **tested** | **Steps** | **Input data** | **Expected output** | **Actual output** | **Pass/Fail** |
| **1** | When animals are fostered their availability needs to be set to N. | **BEGIN**  **IF :P14\_APPLICATIONAPPROVED = 'Y' THEN**  **UPDATE ANIMAL SET ANIMAL.AVAILABLETOFOSTER = 'N' WHERE V('P14\_ANIMALFK') = ANIMAL.ANIMALID;**    **ELSE**    **UPDATE ANIMAL SET ANIMAL.AVAILABLETOFOSTER = 'Y' WHERE V('P14\_ANIMALFK') = ANIMAL.ANIMALID;**    **END IF;**      **END;** | **1.**  In” Foster application form” in “Add new record”  **2.** Select an animal on the list of animal’s name **3.** Fill out the form and submit it | Select **Penelope** | Penelope should be set to “N” and it should no be displayed in the list of Animal’s name | Penelope’s availability is set to “N” and cannot be fostered again | PASS |
| **2** | If a new animal comes to the centre, their admittance date should be defaulted to the current date (if not provided) and should not be available to foster, unless explicitly said | / | **1.** Go to “add new record” and then go to “animal” | / | Admittance date should be defaulted to the current date and the availability to foster should be defaulted to “No” | Admittance date is defaulted to the current date and the foster availability is defaulted to “No” | PASS |
| **3** | The animal breed matches the type of animal chosen. | **BEGIN**    **RETURN 'SELECT DISTINCT BREEDNAME, BREEDID FROM BREED WHERE BREED.ANIMALTYPEFK = :P3\_TYPEFK';**    **END;** |  | Type of animal: **Cat** | Type of breed should display: **-Siamese** **-British Shorthair** **-Persian** **-Maine Coon** | Displays: **-Siamese** **-British Shorthair** **-Persian** **-Maine Coon** | PASS |
| **4** | Some fosters will have a pre-agreed end date | / | **1.** Go to “add new record” and then go to “Foster application form” **2.** Fill out the form | Pre-agreed end date: **01/23/2021** | Information entered into the system | Information successfully entered into the system | PASS |
| **5** |  |  |  | Pre-agreed end date: **\*blank\*** | Information entered into the system | Information successfully entered into the system | PASS |
| **6** | The Start date must be less than the pre agreed end date | **todayDate = new Date();**  **startDate = new Date($x("P8\_STARTDATE").value);**  **endDate = new Date($x("P8\_ENDDATE").value);**    **//$x is the syntax to get the value of an item**      **if(startDate >= endDate)**  **{**  **apex.message.alert("The start date must be before end date. \n Start date has set to current date");**  **//Setting endate to nothing to avoid entering invalid dates**  **$s("P8\_STARTDATE",todayDate.getMonth()+1 + "/" + todayDate.getDate() + "/"+ todayDate.getFullYear());**  **}** |  | Start date: **01/29/2021** Pre-agreed end date: **01/23/2021** | Display error message and set start date to current date | Display error message and set start date to current date | PASS |
| **7** | The Pre-agreed end date must be greater than the start date | **startDate = new Date($x("P8\_STARTDATE").value);**  **endDate = new Date($x("P8\_ENDDATE").value);**    **//$x is the syntax to get the value of an item**      **if(endDate <= startDate)**  **{**  **apex.message.alert("The pre-agreed end date must be after startdate");**  **//Setting endate to nothing to avoid entering invalid dates**  **$s("P8\_ENDDATE","");**  **}** |  | Pre-agreed end date: **01/20/2021**  Start date: **01/250/2021** | Display error message and set end date to null | Display error message and set end date to null | PASS |

**7: Conclusion**

The team has been successful in implementing a fully functioning Web Interface. All the requirements are fulfilled in the database application, which these includes:

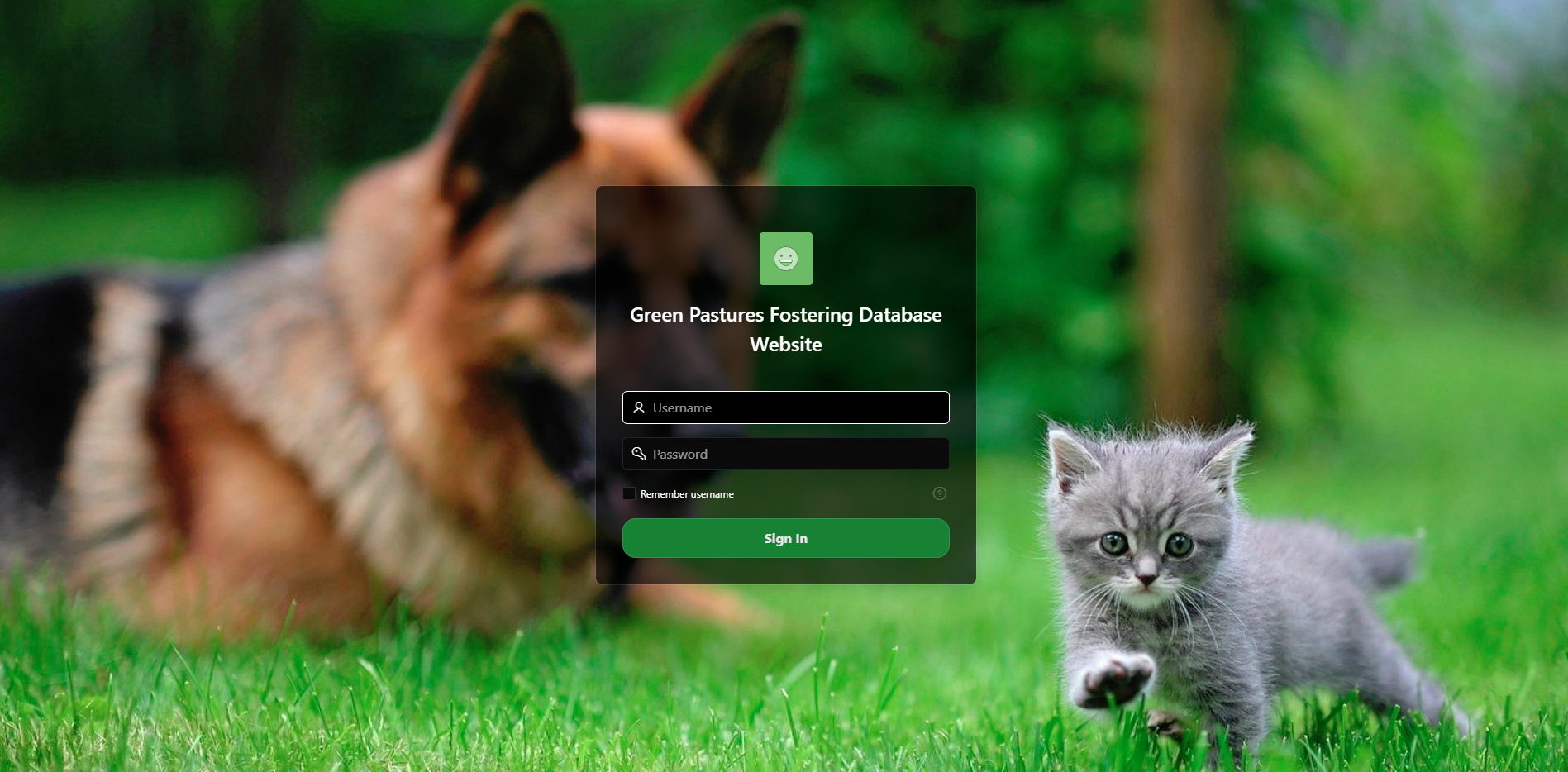
* Staff can log in with the username/passwords provided for them. Staff details such as name, email address, and staff id are displayed on the staff records page.
* Constraints are placed in the Animal Registry Form so that Cat, Dogs, Hamsters, and Rabbits can only be added to the database records.
* Animals can be added/updated/removed from the fostering list which can be done through the edit animal records page.
* Default values for admittance date and foster availability are there to help with faster and less error-prone registration.
* Applicants who are 18 can be registered on the Applicant Registration Form with the ability for them to foster many animals. Details such as an address, email, and contact number must be included on the applicant’s form to be able to complete the registration.
* Foster Application Form provides a place for the applicant’s animal interest, the reason for fostering and option to approve, deny or pend the applicant’s status.
* The Approved and Pending Foster List only shows Pending and Approved Application records for easy viewing.

In Conclusion, we have learned and understand how to use Oracle Apex, especially managing the database script code and building an app. All sections from 2 to 6 on the report provided the evidence of how we manage to go through developing the web app for Green Pasture Fostering. For example, showing how the made the application, listing the limitations and improvements we found, and testing the features that meets the system requirements.

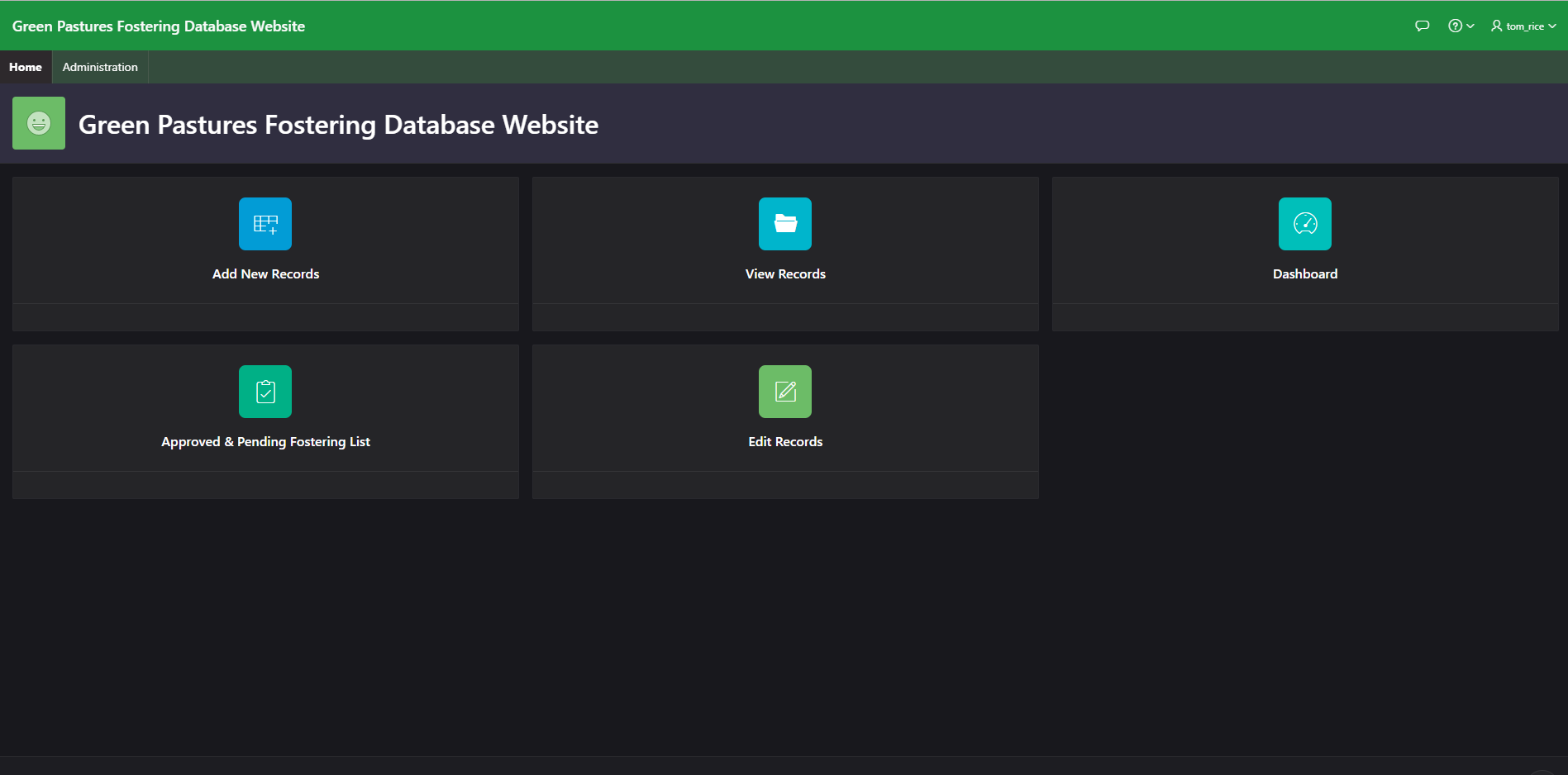
**8: Appendix**

## **8.1: User-Guide**

Login Page



Main Menu / Home screen

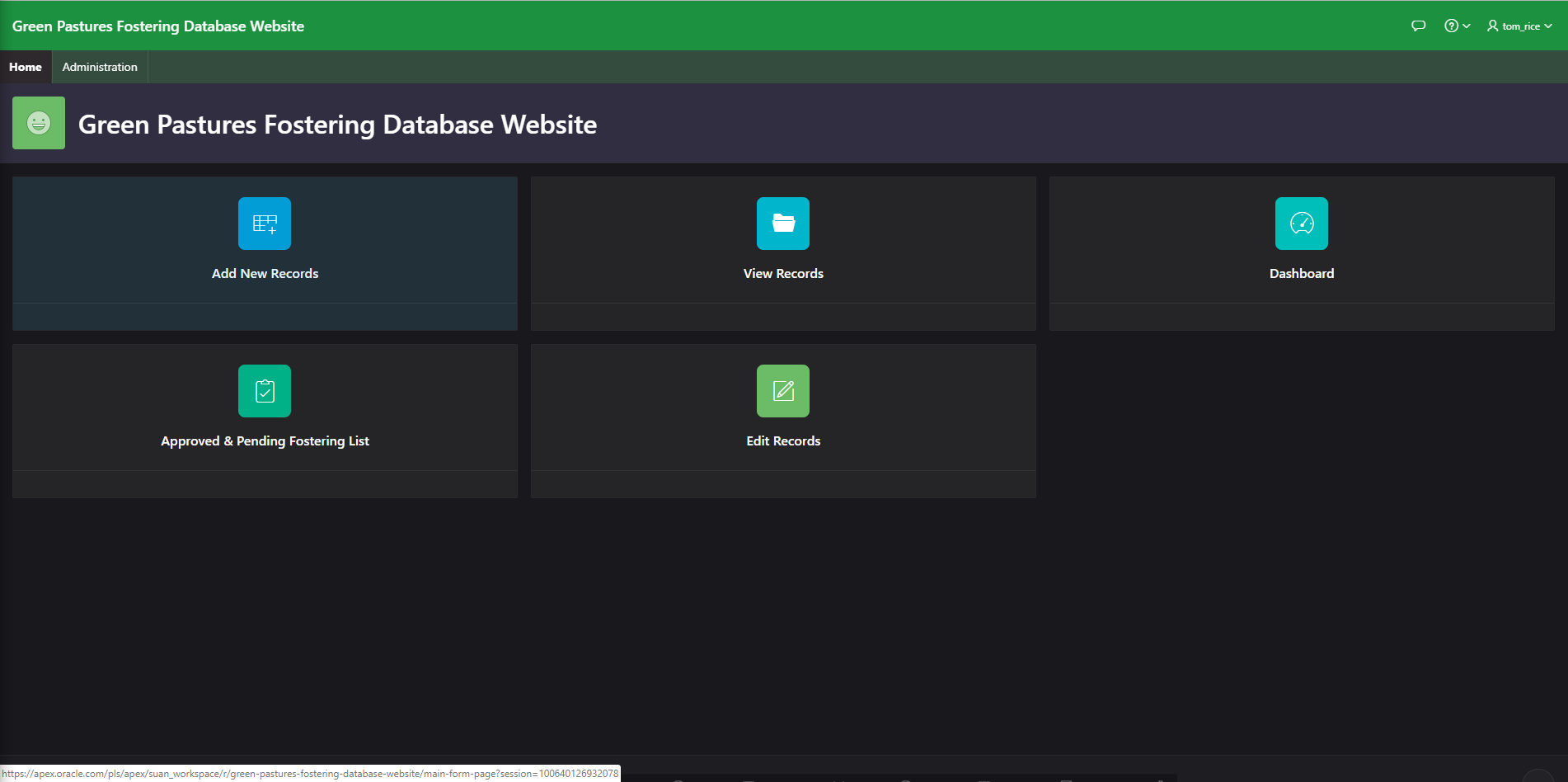


Contains the 5 key features of the database:

* Add records
* View records
* Dashboard
* Approved & pending Foster list
* Edit Records

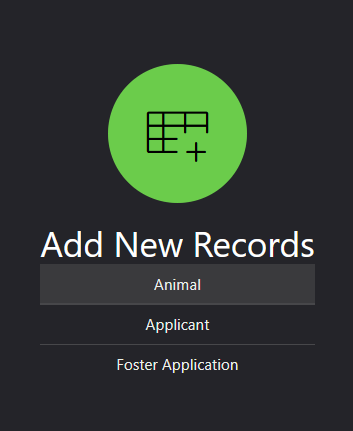
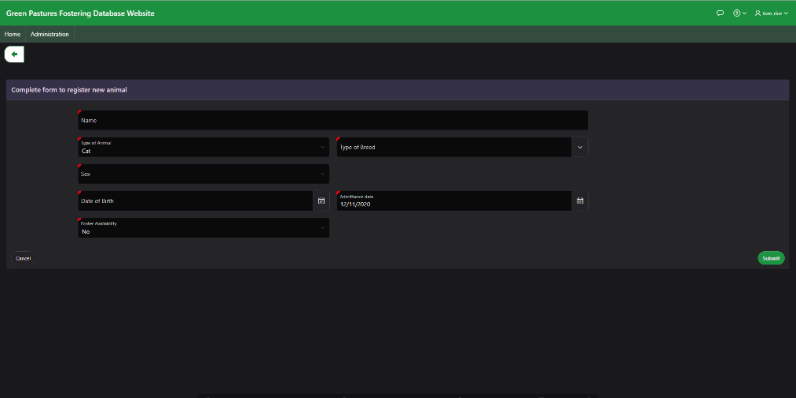
**Add new records**

Click on Add New Records to view the forms for adding new Animals, Applicants and Foster Applications.



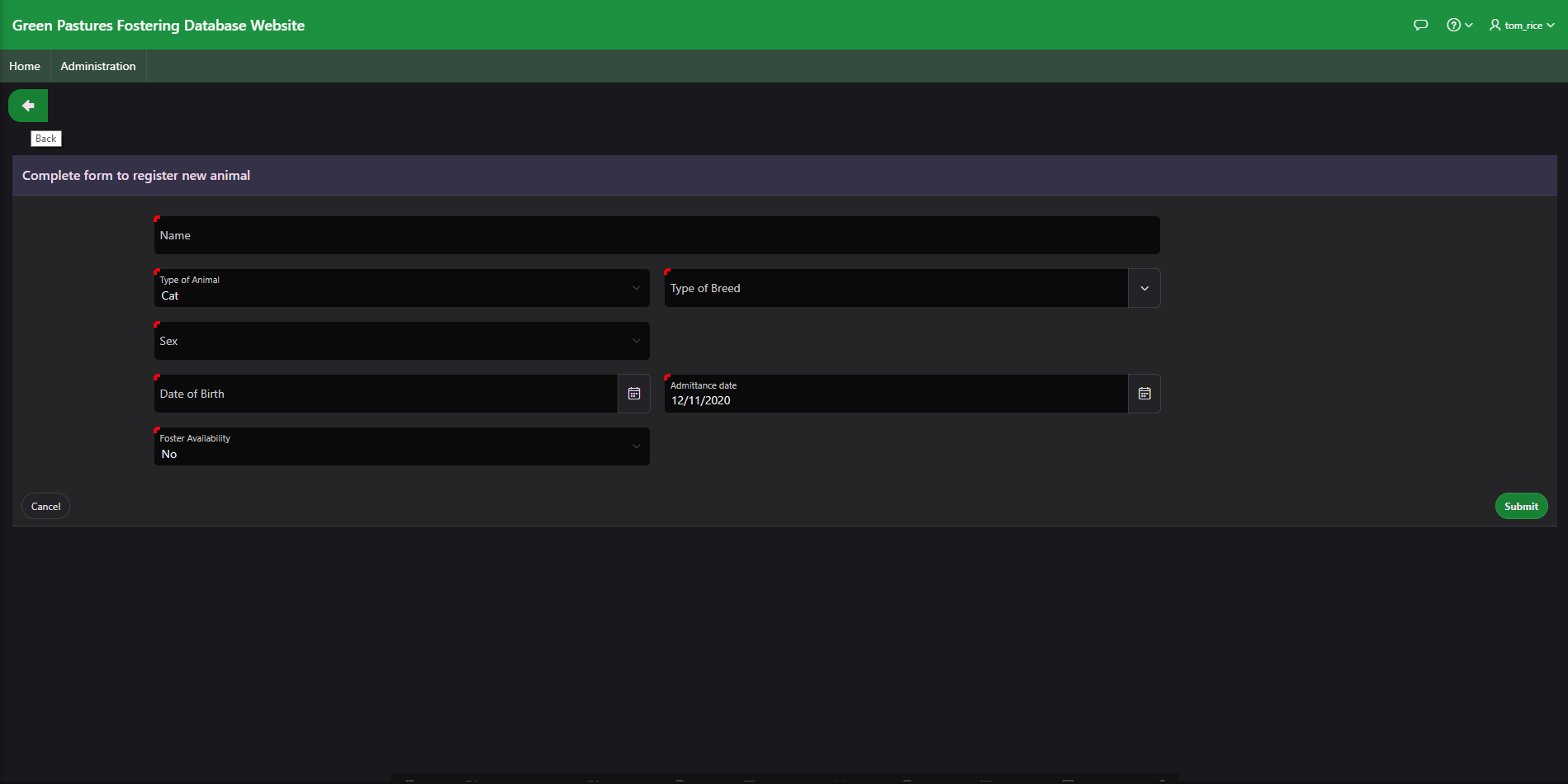
**Animal**

Click **Animal** to go to the form to add an Animal record.

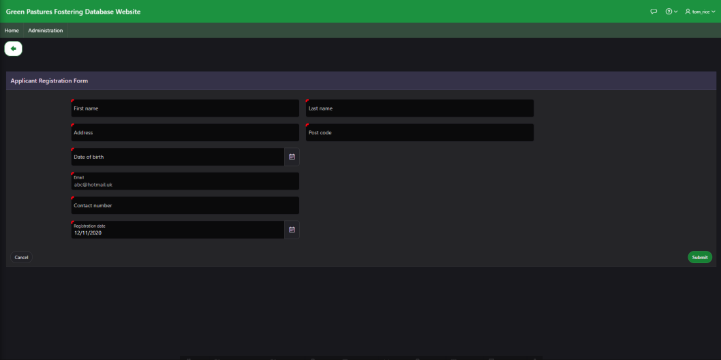


Fill in the required fields and click the submit button at the bottom right to add this record to the database.

Click on the back button at the top left to go to the previous page.



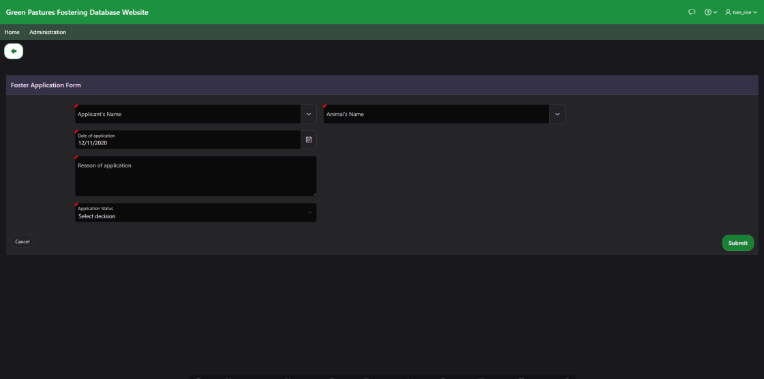
**Applicant**

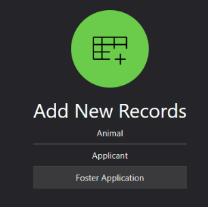
Application button will take you to the form for adding an applicant record.



Fill in the required information and click the submit button to add an Applicant record.

**Foster**

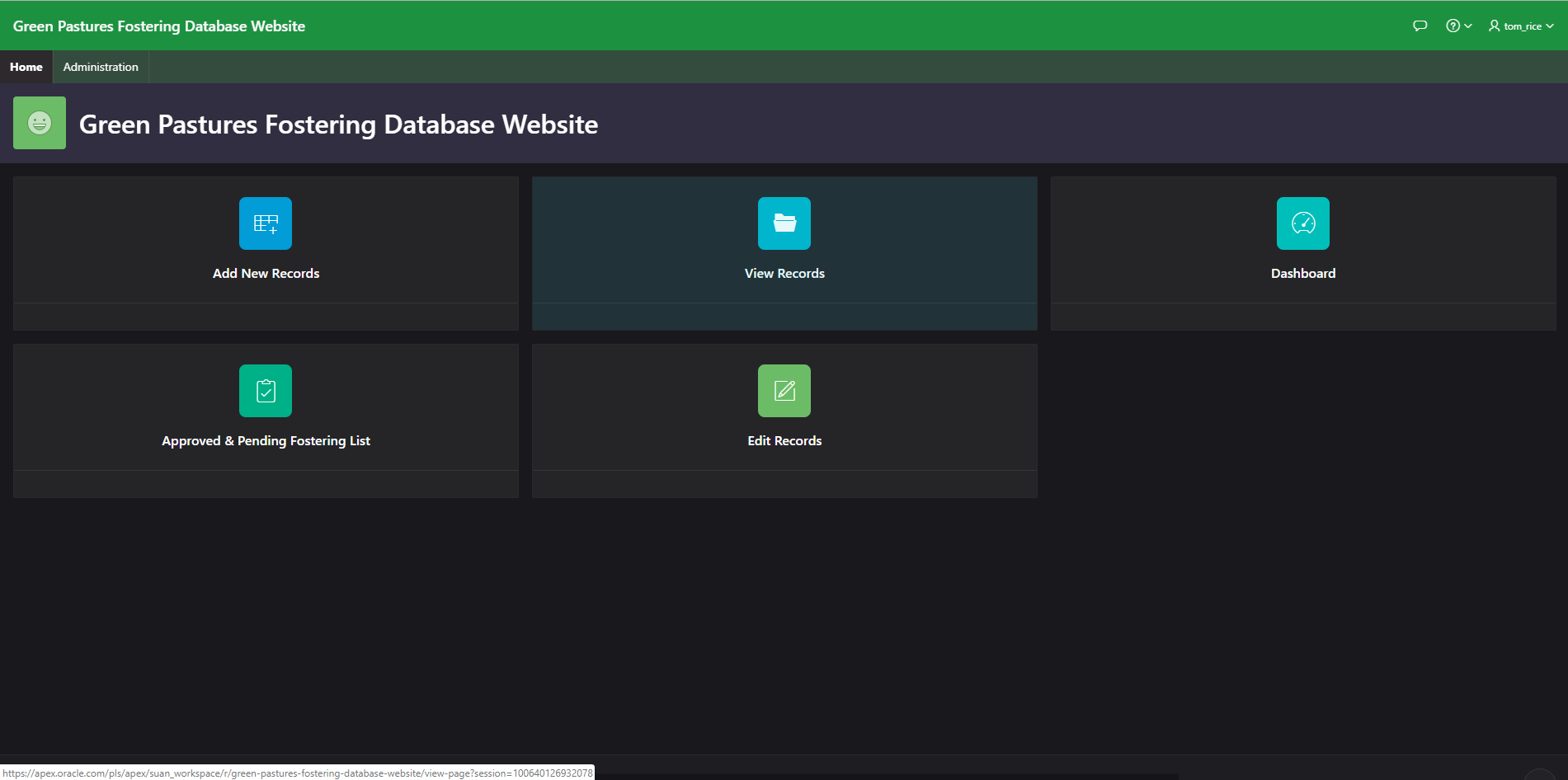
Foster Application will take you to a form for adding a Foster Application record.



Fill in the required fields and click the submit button to add a Foster Application record.

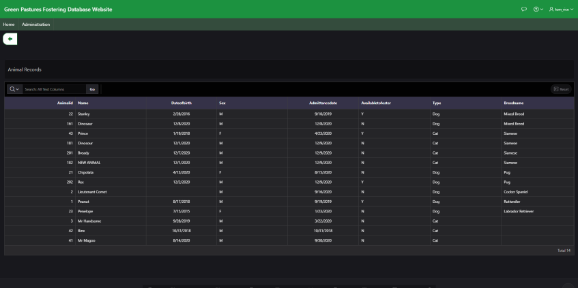
**View Records**

From the home screen, click on the View Records button to choose which record to view.



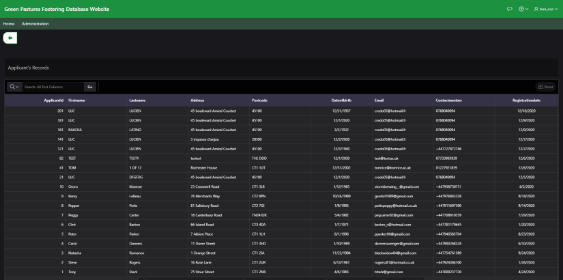
**Animal**

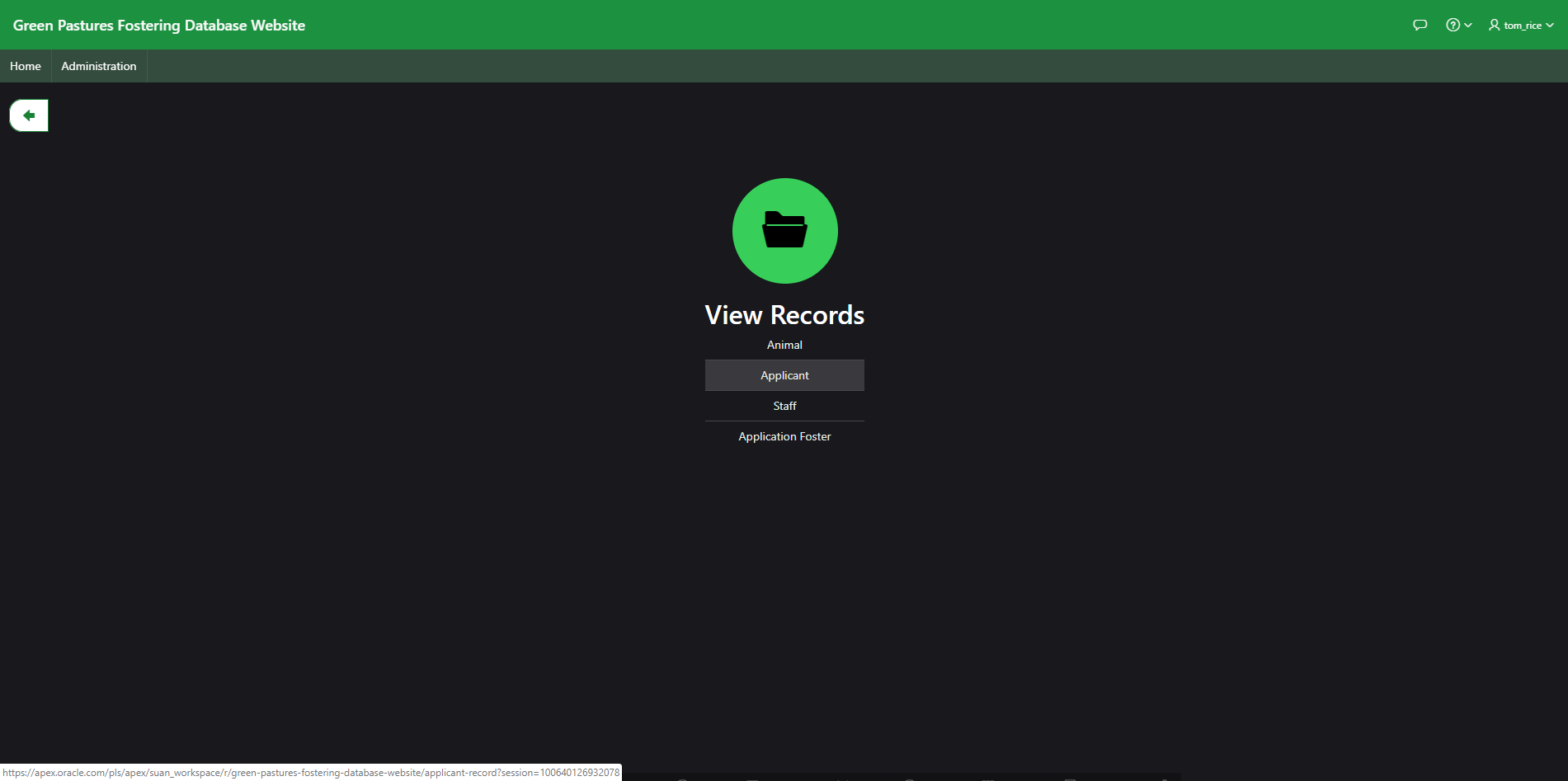
The Animal button will take you to a page showing all of the animal records.



From the Animal records page you can view, search the records and sort in ascending and descending from each field.

**Applicant**

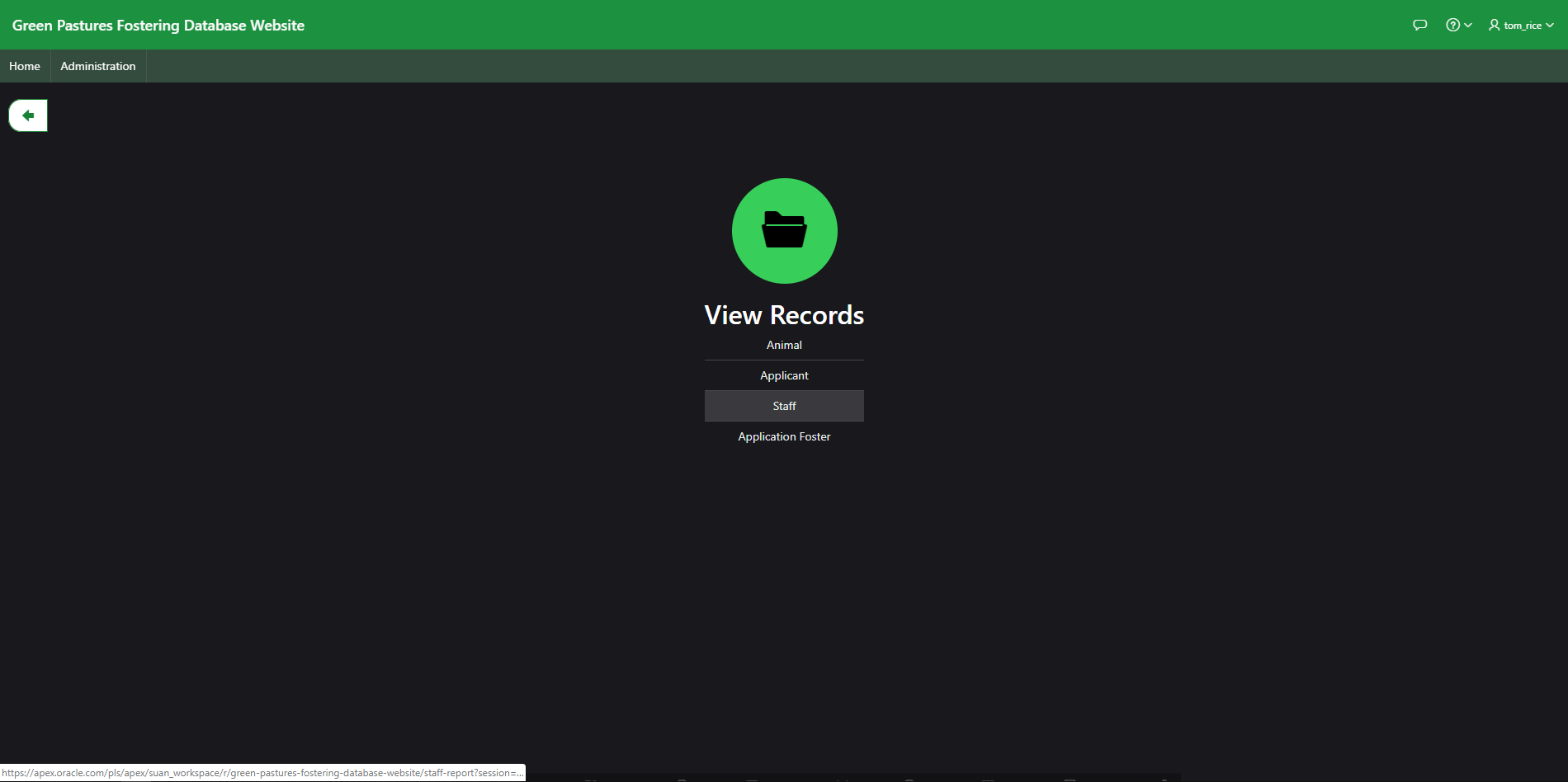
Applicant button takes you to the page to view the records containing the applicant’s data, including: personal information, contact information and registration date.



From the Applicant records page you can view, search and filter records, sorting ascending or descending by each field.

**Staff**

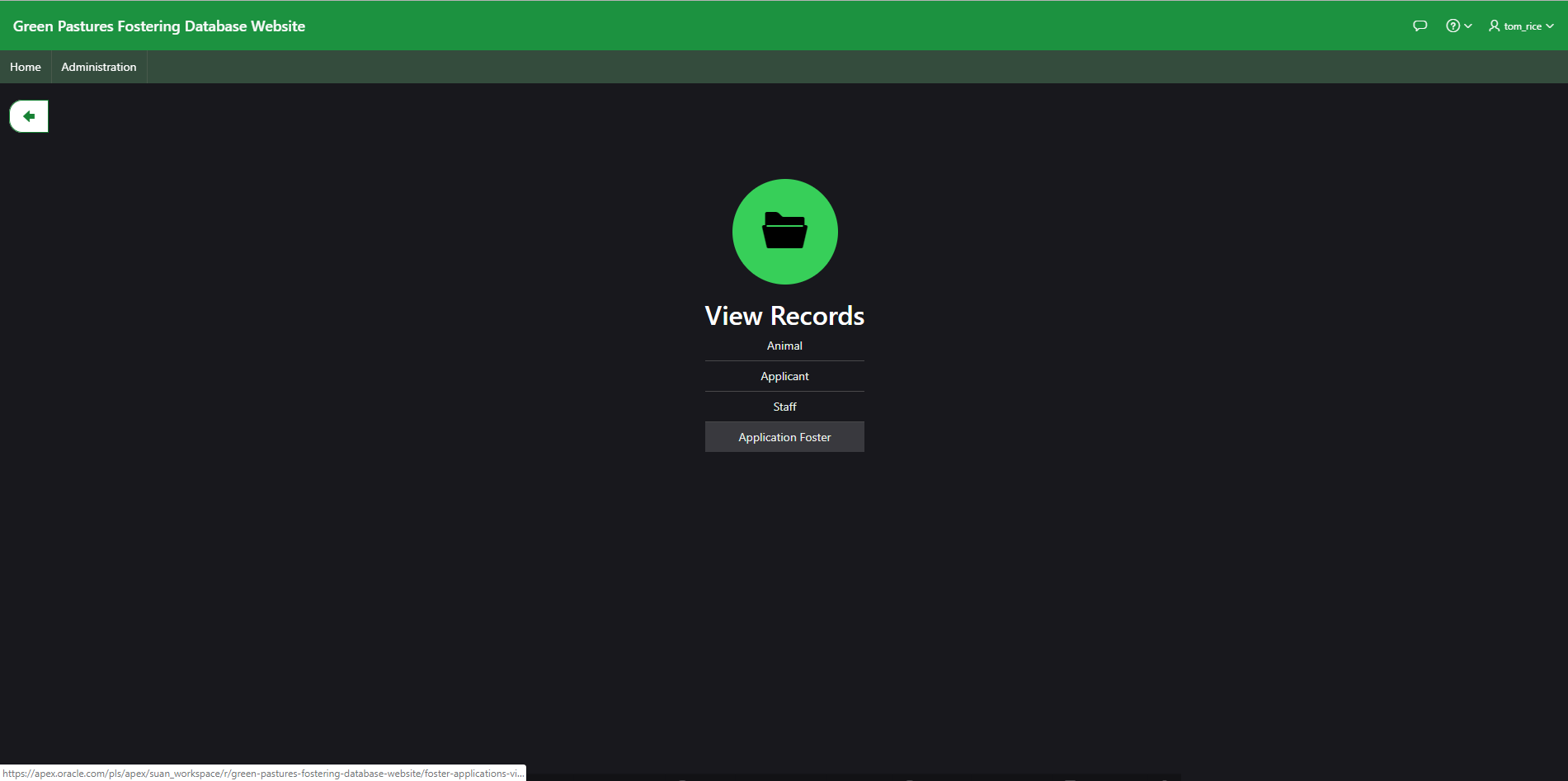
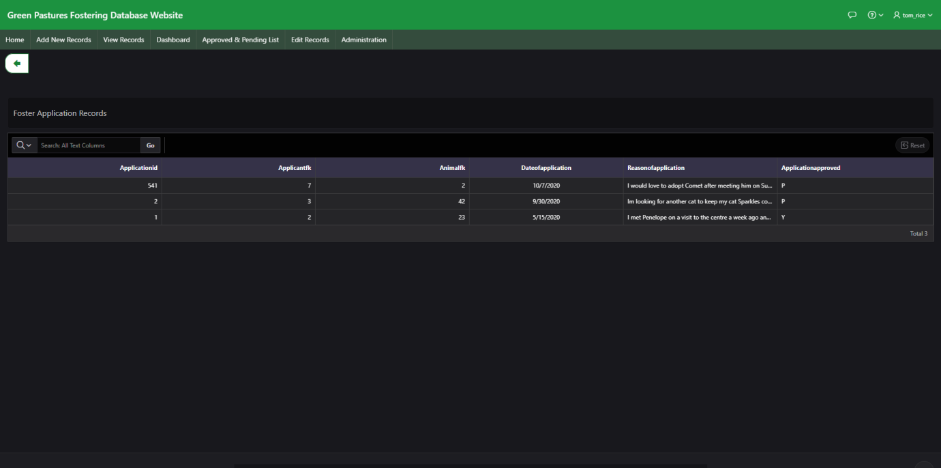
The Staff button takes you to the view page for the staff where you can view some basic details.



From the View Staff page you can see the staff, their names and contact information.

**Application Foster**

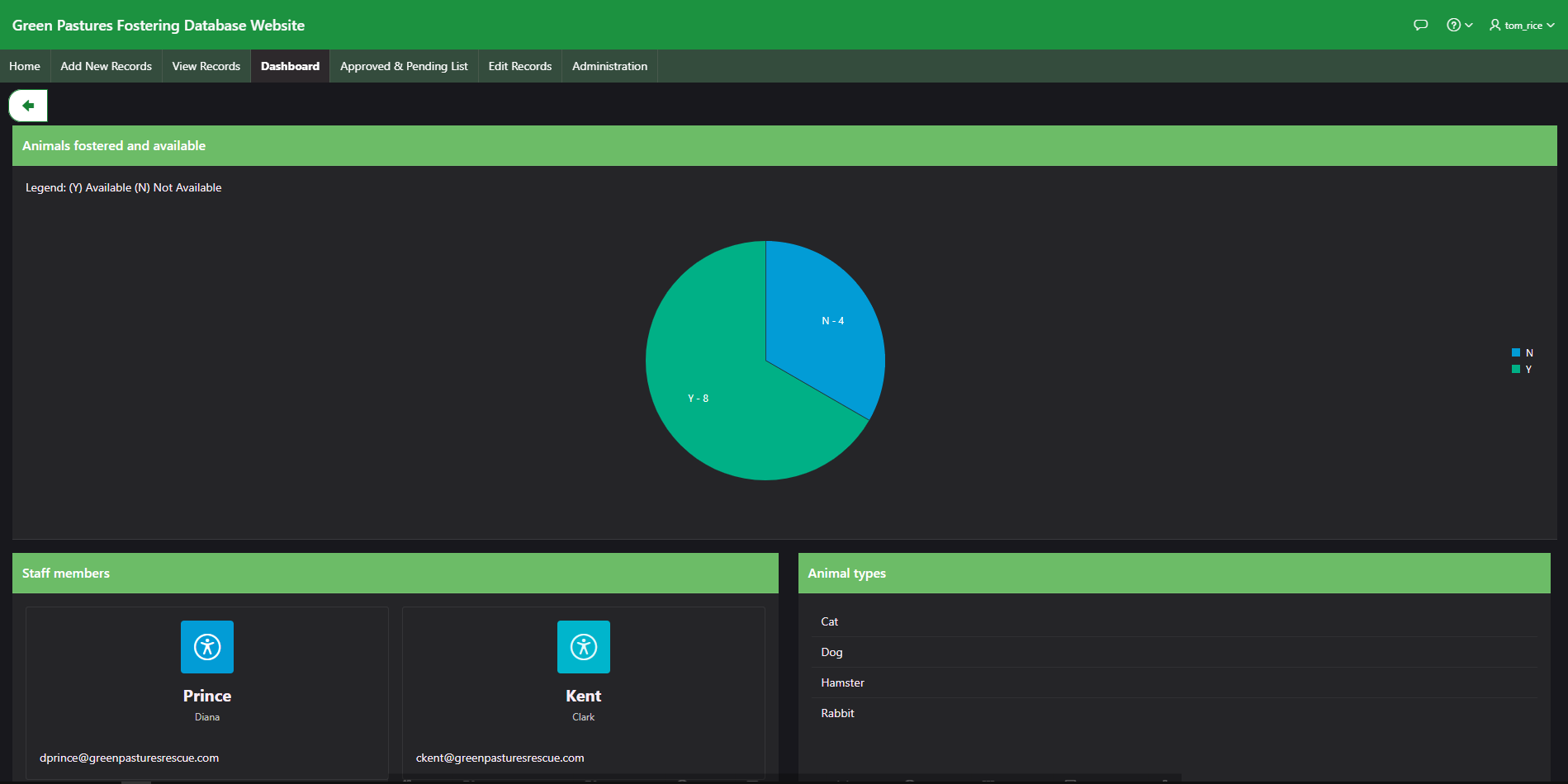
View Application Foster records by clicking the Application Foster button, to view



From the Application records you can see all of the approved and pending applications along with the reason for the application and date submitted. From this page you can search and sort the application to look for particular records.

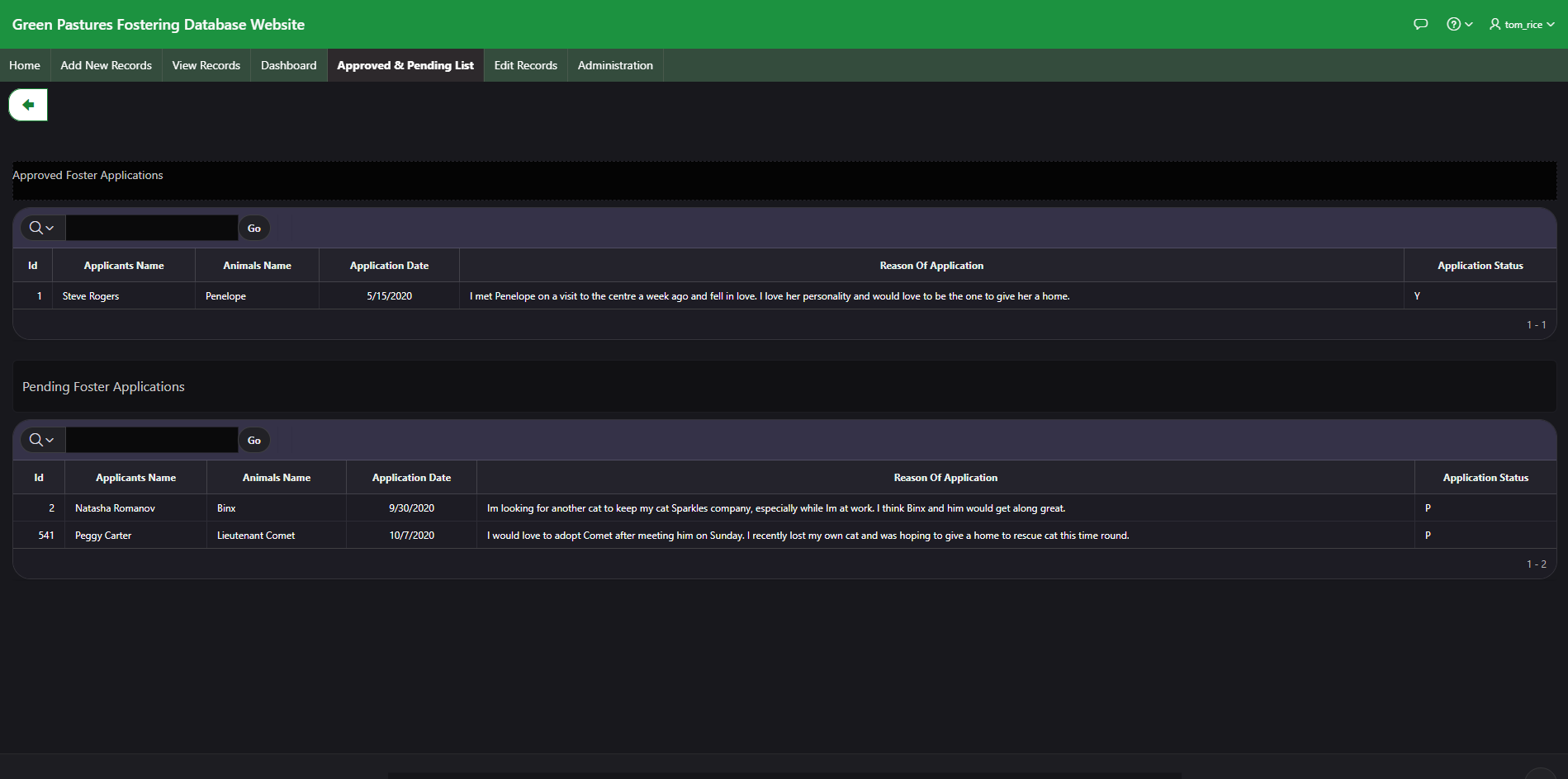
**Dashboard**

The Dashboard has some general information such as the ratio of available animals, the staff and the types of animals kept at green pastures. This page is also quickly available from the navigation at the top of the page.



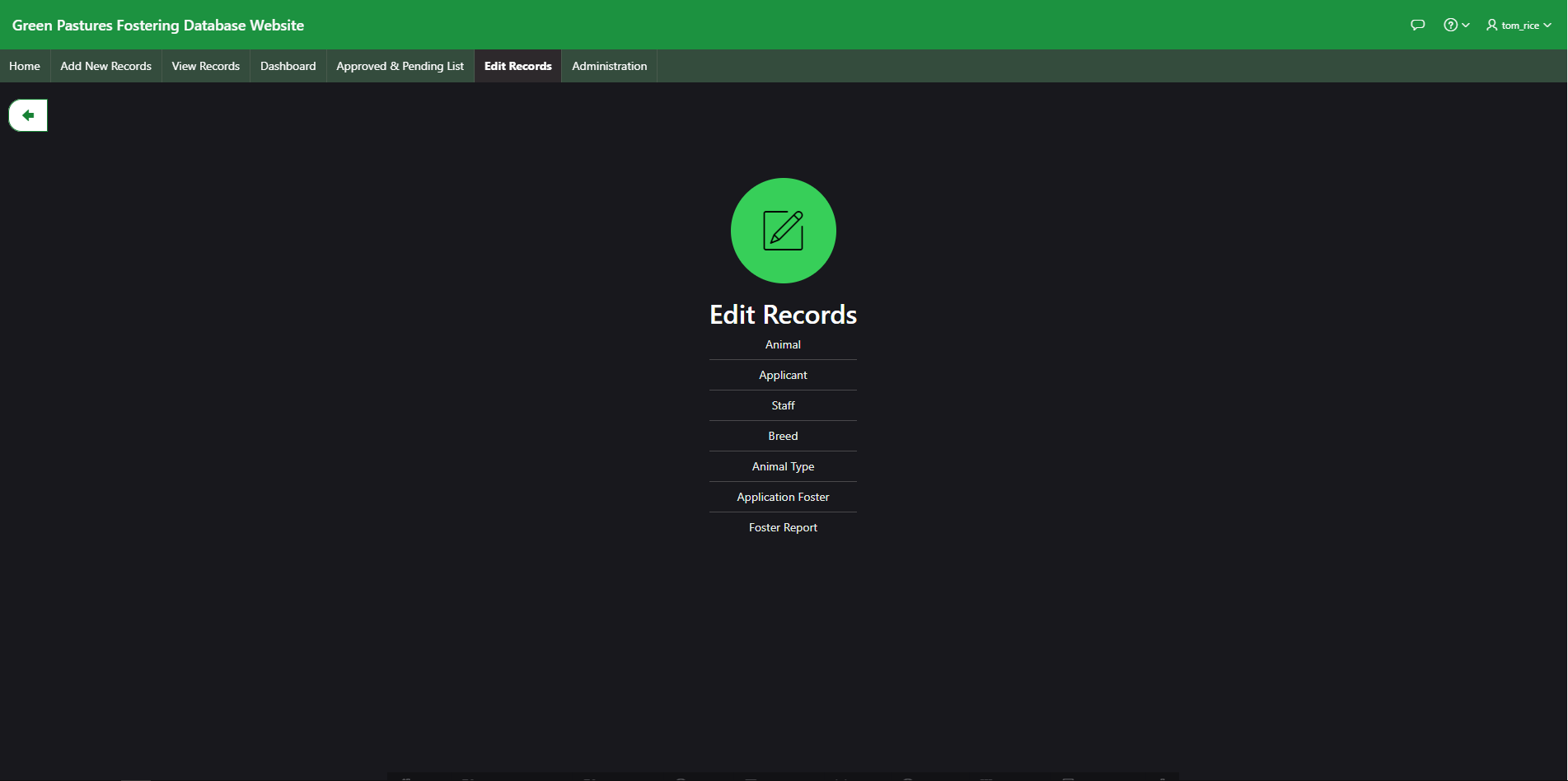
**Approved and pending list**

Approved and Pending list splits the approved application from the pending ones. From this you can see which animals are currently fostered and which applications still need to be either approved or denied.

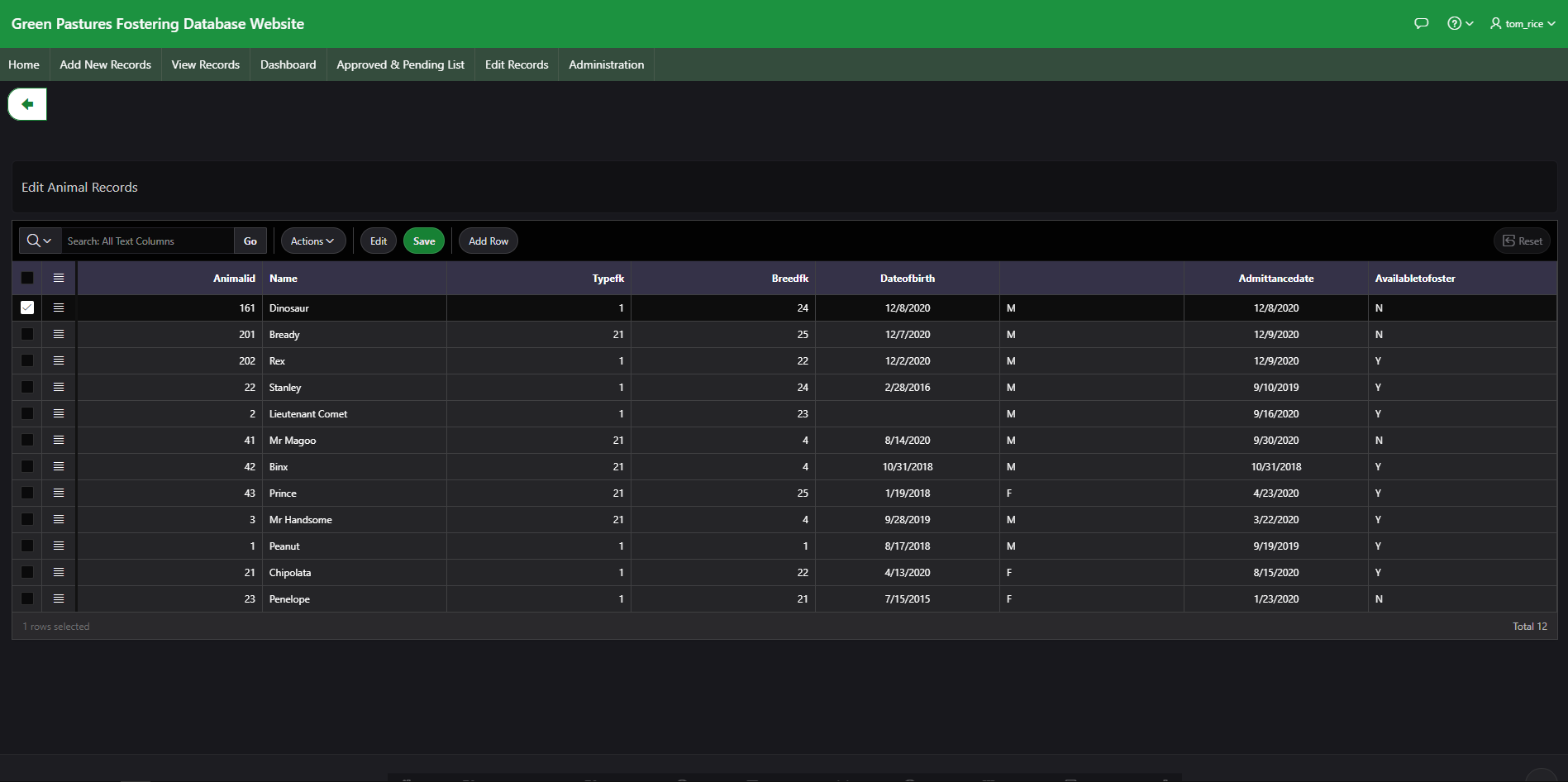


**Edit Records**

Edit records lets you edit the records for any of the tables, allowing you to expand the database in many ways when needed or delete and edit records in the event of mistakes or errors.



**Animal**

Edit Animal records, you can modify or delete any animal records.

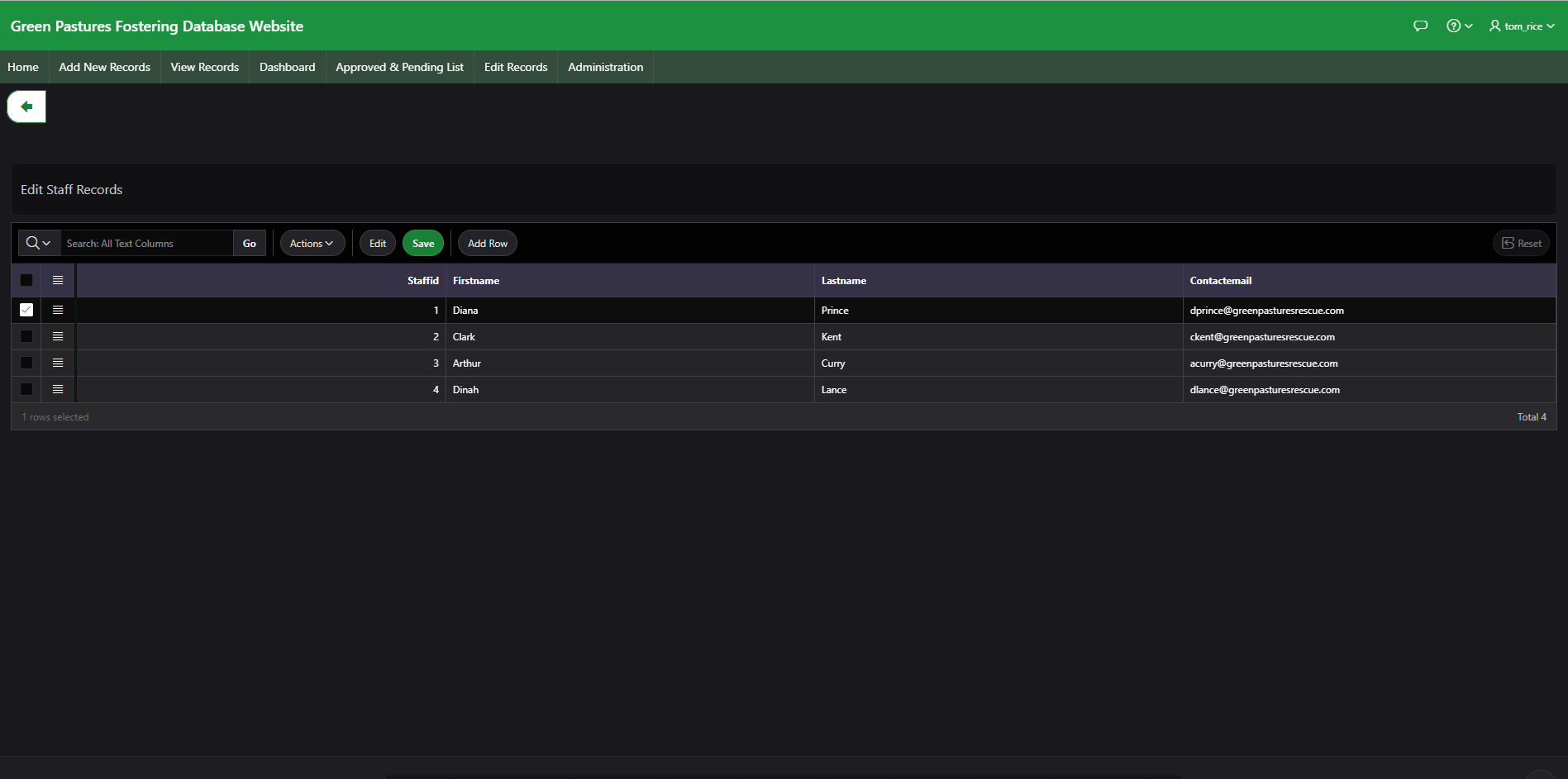
**Applicant**

Edit Applicant records, you can modify or delete any applicant records.



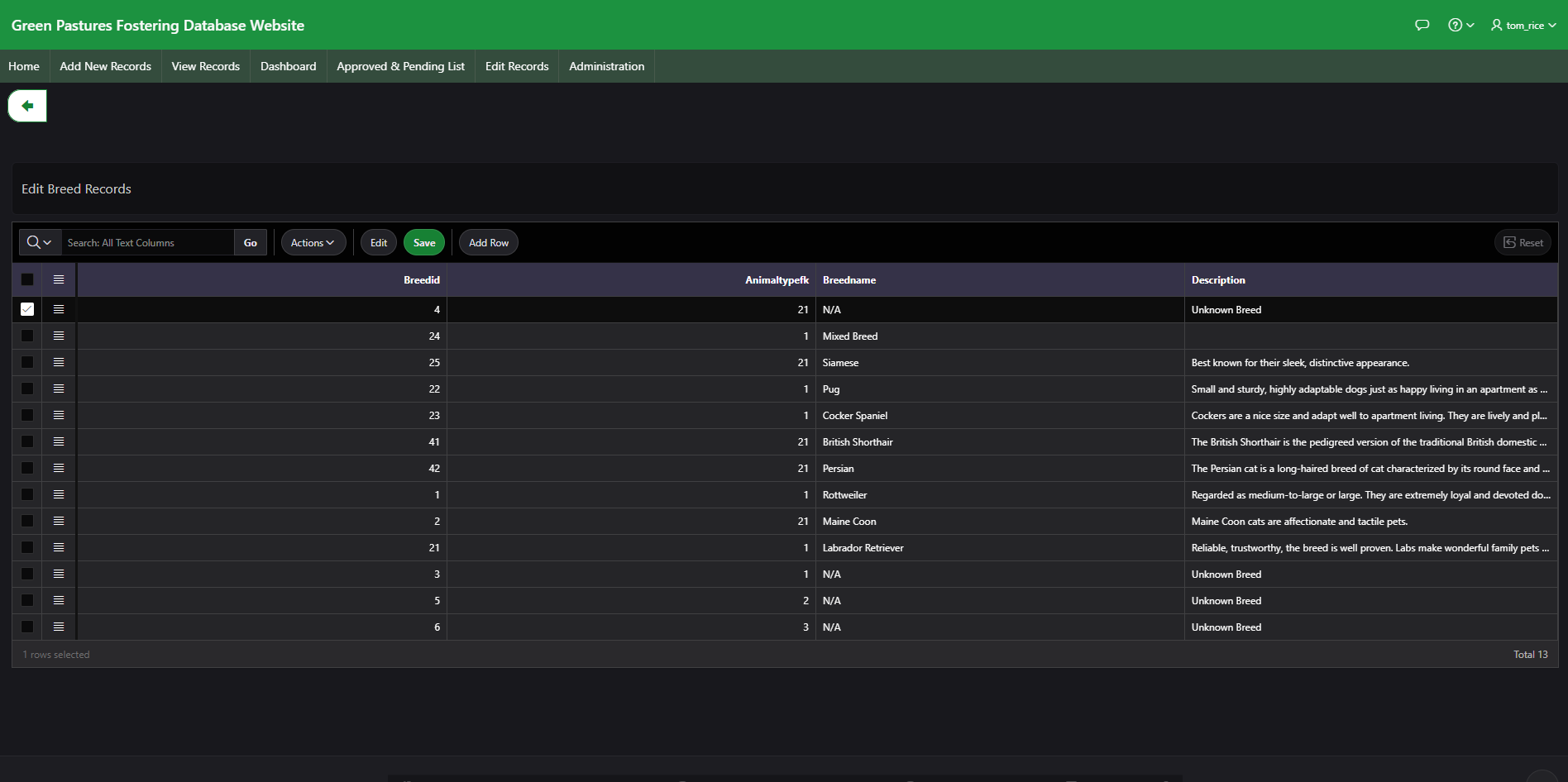
**Staff**

Edit Staff records, you can modify or delete any staff records.



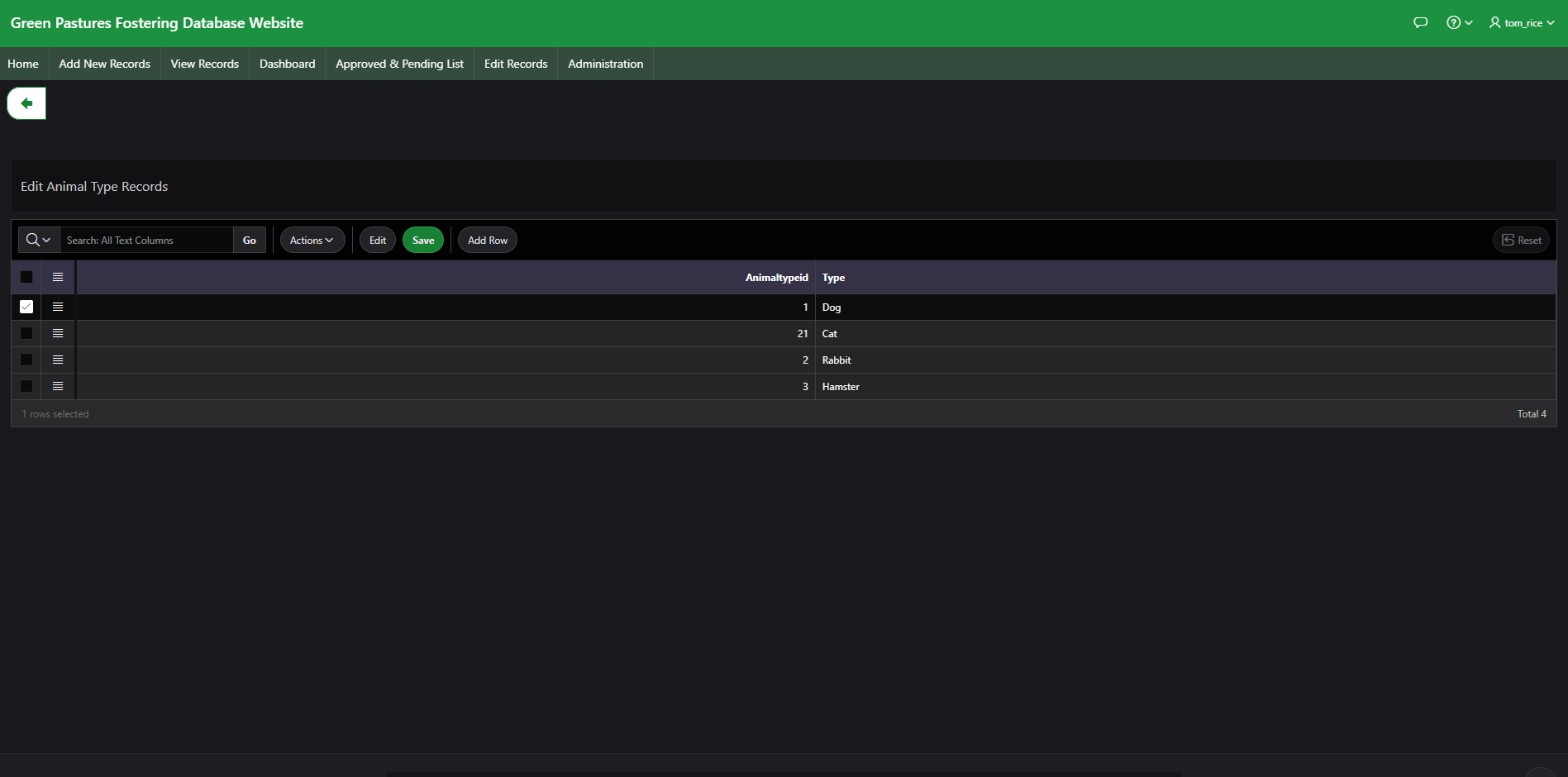
**Breed**

Edit Breed records, you can modify or delete any breed records.



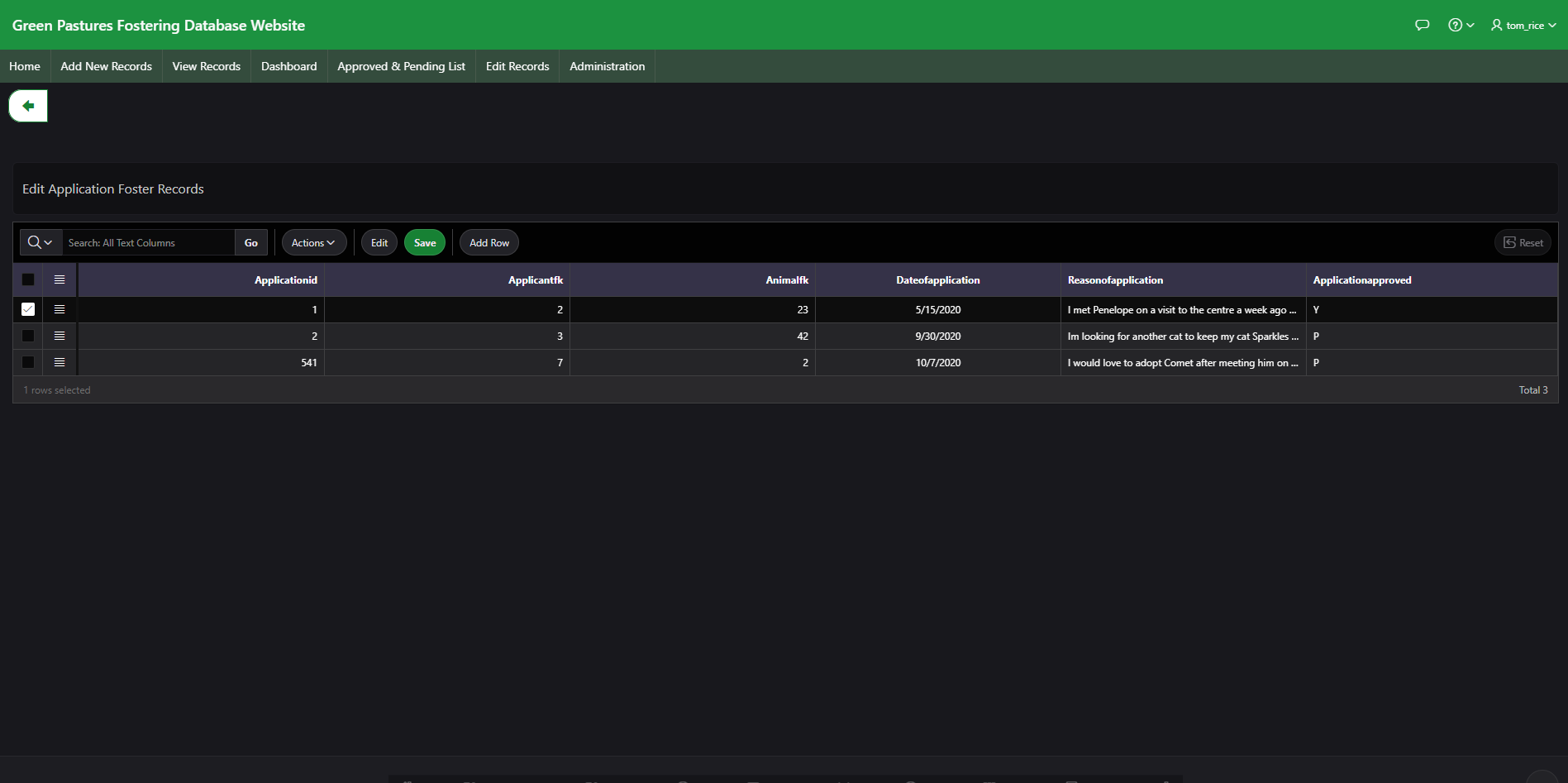
**Animal Type**

Edit Animal type records, you can modify or delete any animal type records.



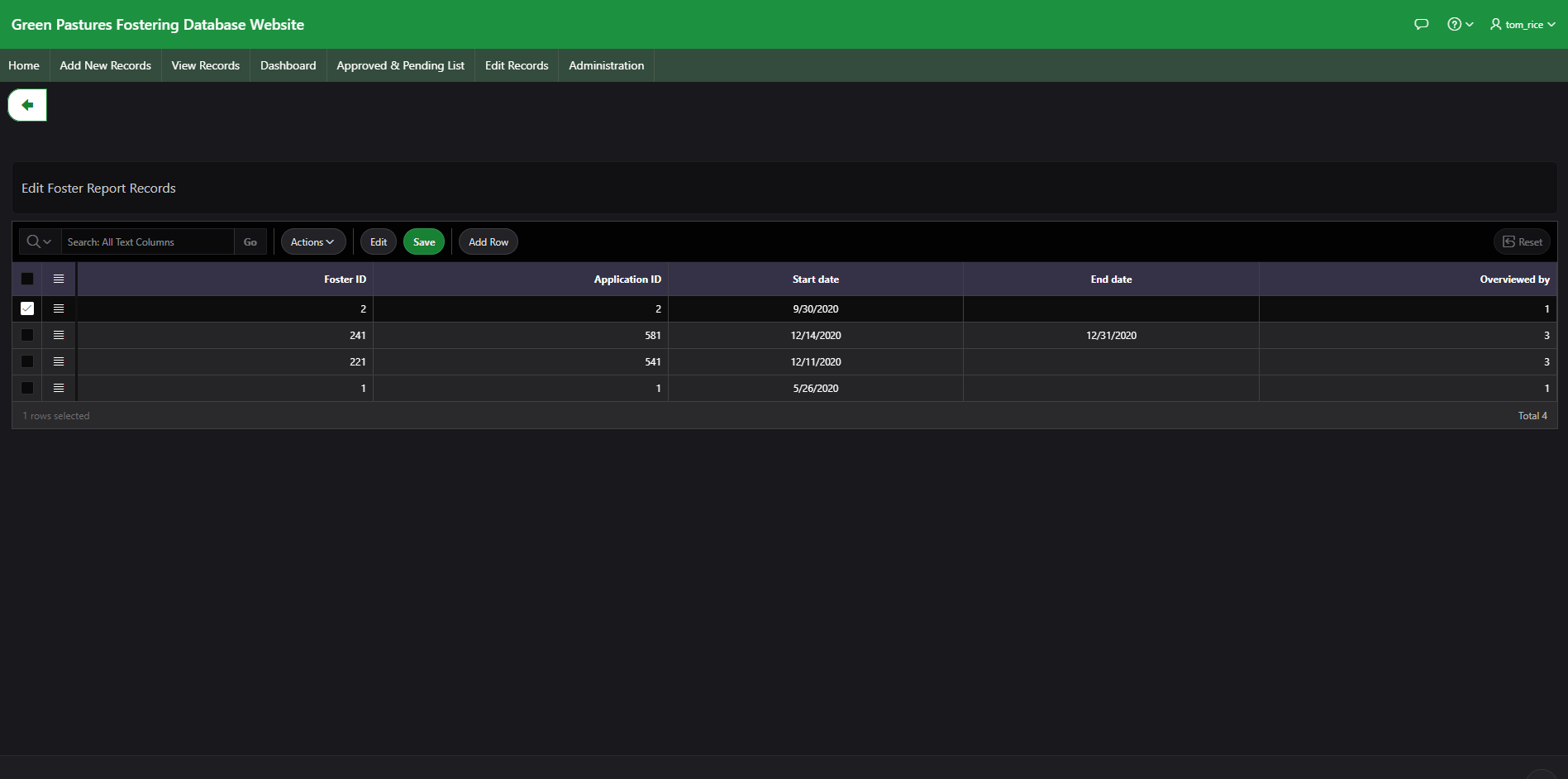
**Application Foster**

Edit Application Foster records, you can modify or delete any records for the foster applications.



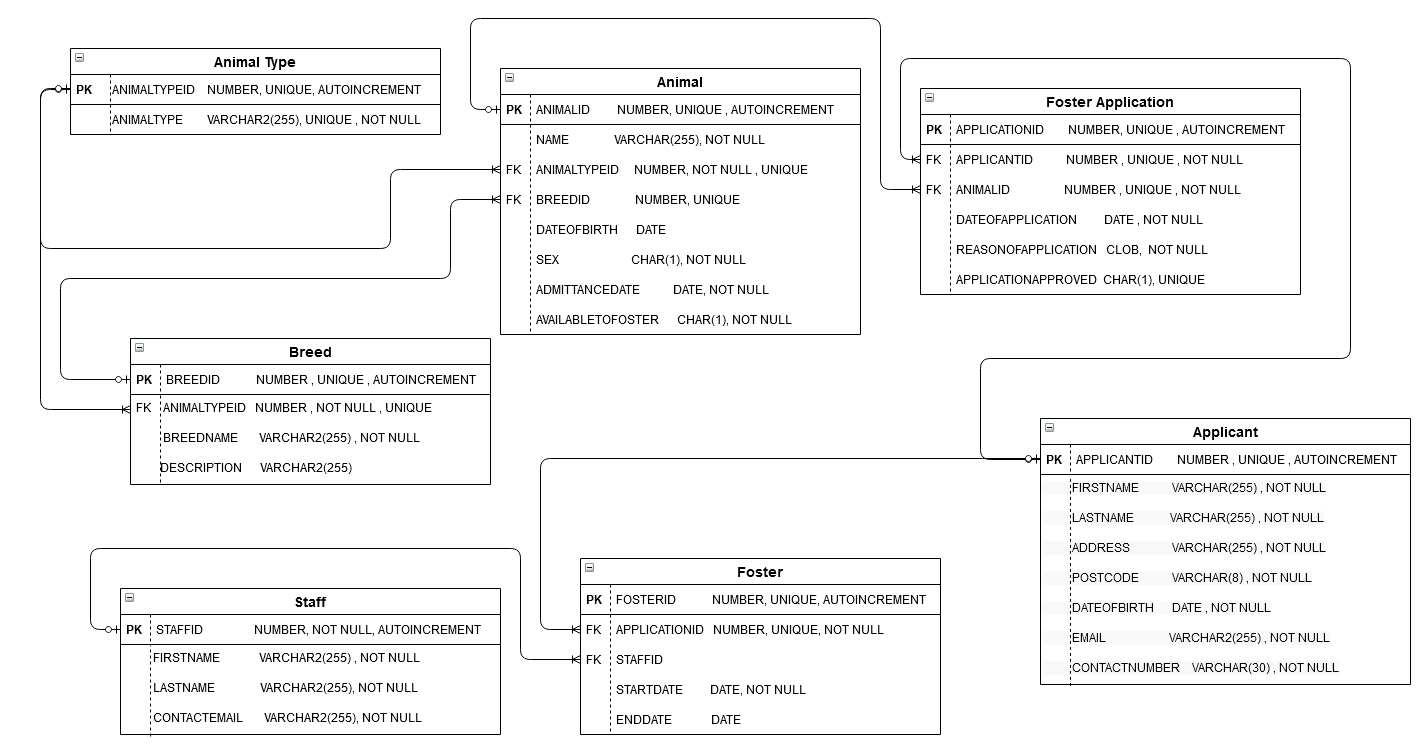
**Foster**

Edit Foster records, you can modify or delete any foster records.



## **8.2: Logical Database Model:**

The following is the logical database model using Draw.io as design tool. The design is based from what the current website app:



**9: Bibliography**

* <https://www.facebook.com/foxinfotech2014> (2019). *How to Show Alert Messages in Oracle Apex? | Vinish Kapoor’s Blog*. [online] Vinish Kapoor’s Blog. Available at: [foxinfotech.in/2019/11/how-to-show-alert-messages-in-oracle-apex.html](https://www.foxinfotech.in/2019/11/how-to-show-alert-messages-in-oracle-apex.html) [Accessed 18 Dec. 2020].
* <https://www.foxinfotech.in/2020/03/oracle-apex-function-returning-sql-query-example.html>
* Moretodogstrust.org.uk. (2020). *Volunteer Enquiry Form*. [online] Available at: <https://www.moretodogstrust.org.uk/volunteer/enquiry-form> [Accessed 18 Dec. 2020].
* W3schools.com. (2015). *JavaScript Date Formats*. [online] Available at: <https://www.w3schools.com/js/js_date_formats.asp> [Accessed 18 Dec. 2020].
* W3schools.com. (2019). *SQL INNER JOIN Keyword*. [online] Available at: <https://www.w3schools.com/sql/sql_join_inner.asp>.

# **10: Reference List**

* ‌‌www.w3schools.com. (n.d.). *MySQL CURRENT\_DATE() Function*. [online] Available at: <https://www.w3schools.com/SQl/func_mysql_current_date.asp> [Accessed 12 Dec. 2020].
* [foxinfotech2014](https://www.facebook.com/foxinfotech2014) (2020). *Oracle Apex: Working with Date Picker | Vinish Kapoor’s Blog*. [online] Vinish Kapoor’s Blog. Available at: <https://www.foxinfotech.in/2020/04/oracle-apex-date-picker.html> [Accessed 18 Dec. 2020].
* Oracle SOY. (2020). *Oracle Apex Set And Get Item Value Using Javascript - Oracle SOY*. [online] Available at: <https://www.oracle.soy/apex-set-and-get-item-value-using-javascript/> [Accessed 18 Dec. 2020].