Christopher Poitras
Simeon Wuthier
Aaron Wiegert
Christian Gruenewald
Christin Boicourt

Sprint 3 - 2

Github: https://github.com/UCCS-CS3300/group1_spring2020
Heroku: http://dog-friendly.herokuapp.com/

This current sprint included the following feature implementation:
Login:
As a dog owner
So that I can customize my experience and quickly find what I need
I would like to have a login feature
Contact Us:
As a dog owner,
So that I can ask for help with managing my account,
I want to be able to contact the customer service team to ask about my specific needs.
Additional Tasks:
We are updating the Home Page and About Us sections of the site. For the home page we are going to add testimonials, and in the about us page we are going to add information about members of the team.

Successes:
During the last week of this sprint we were able to implement the Contact Us feature and update the Home Page and About Us Page. We were also able to implement a dark theme for the website after Jay's input in the demo and we were able to successfully implement a Jest testing environment to test our javascript code.

Set-Backs:
Implementation of the Login-In feature:
Google login feature was not successfully implemented. It almost broke our master git branch. Going forward as a team we decided it will probably be better to do our own login feature so as to not almost ruin everything. This will be discussed more in the next sprint and this feature will be pulled into the next sprint.

Testing of the Contact Us feature:
Creating test cases for the Contact Form is currently a difficult task, because the code for it (given to us by ZenHub, owned by Aaron) is obfuscated and nearly impossible to test until the source code is made available to us. The Contact Us form works by creating a new <html> tag appended to the bottom of the page, and running an obfuscated script that inserts the toggleable button. Sending a message through it triggers an HTTP request to send to ZenHub. Until we are able to control these functions manually through the code, the test cases will remain in the pending state. This will be discussed further in the next sprint. Test scenarios have been created in cucumber, but the test steps have not been created yet.

Current Testing Code Coverage:
SimpleCov:
After running the command "rake" in our environment and looking at the generated report in coverage/ .last_run.json we can see that we only have 21.29% code coverage. We believe this is due to the lack of testing that we have for the google maps api which we are just now getting started on now that we have a Jest environment set-up.
Jest:
We could check our javascript code coverage using the command "jest test --coverage" however, we are just now implementing jest testing for our javascript code.
Reek and Flog:
From our understanding Reek and Flog are to be used on .rb files. We haven't written any code in our controllers and our models for our website. We do have some code in our controllers at this time, but it was written for the login feature which has not been implemented correctly at this time.
Rake Stats:
When running the command "rake stats" we generated the image below. From the image below we can see that we have some work to do as our ratio is 1:0.5 and from the documentation we referenced
https://thoughtbot.com/blog/simple-test-metrics-in-your-rails-app-and-what-they it seems that we need to at least have a ratio of 1:1. We feel that we should be able to achieve this ratio once we get jest running for our javascript testing.

```
+------------------------+--------+--------+----------+----------+------+--------+
| Name                   | Lines  |   LOC  | Classes  | Methods  | M/C  | LOC/M  |
+------------------------+--------+--------+----------+----------+------+--------+
| Controllers            |   123  |    92  |      5   |     16   |   3  |      3 |
| Helpers                |     8  |     8  |      0   |      0   |   0  |      0 |
| Jobs                   |     2  |     2  |      1   |      0   |   0  |      0 |
| Models                 |    15  |    12  |      2   |      1   |   0  |     10 |
| Mailers                |    16  |    12  |      2   |      1   |   0  |     10 |
| Channels               |     8  |     8  |      2   |      0   |   0  |      0 |
| JavaScripts            |   535  |   412  |      0   |     19   |   0  |     19 |
| Libraries              |   126  |    92  |      0   |      0   |   0  |      0 |
| Controller tests       |    14  |     6  |      2   |      0   |   0  |      0 |
| Helper tests           |     0  |     0  |      0   |      0   |   0  |      0 |
| Model tests            |     0  |     0  |      0   |      0   |   0  |      0 |
| Mailer tests           |     0  |     0  |      0   |      0   |   0  |      0 |
| Integration tests      |     0  |     0  |      0   |      0   |   0  |      0 |
| System tests           |     0  |     0  |      0   |      0   |   0  |      0 |
| Routing specs          |    38  |    29  |      0   |      0   |   0  |      0 |
| View specs             |    91  |    75  |      0   |      0   |   0  |      0 |
| Helper specs           |    15  |     4  |      0   |      0   |   0  |      0 |
| Request specs          |    10  |     9  |      0   |      0   |   0  |      0 |
| Controller specs       |   141  |    94  |      0   |      0   |   0  |      0 |
| Model specs            |     5  |     4  |      0   |      0   |   0  |      0 |
| Cucumber features      |   173  |    67  |      0   |      0   |   0  |      0 |
+------------------------+--------+--------+----------+----------+------+--------+
| Total                  |  1320  |   926  |     14   |     37   |   2  |     23 |
+------------------------+--------+--------+----------+----------+------+--------+
  Code LOC: 638      Test LOC: 288      Code to Test Ratio: 1:0.5
```