

CS 320 Course Project Final Report

for

Animal Database

Prepared by

Group Name: Team 20

|  |  |  |
| --- | --- | --- |
| Vinh Duong | 11613642 | vinh.duong@wsu.edu |
| Seth Lanante | 11458512 | seth.lanante@wsu.edu |
| Yevgeniy Diriyenko | 11679399 | Yevgeniy.diriyenko@wsu.edu |
| John Stevenson | 11639605 | john.stevenson@wsu.edu |

|  |  |
| --- | --- |
|  |  |
| Date: | December 16, 2020 |
|  |  |
|  |  |
|  |  |

Content

Contents ii

1 Introduction 1

1.1 Project Overview 1

1.2 Definitions, Acronyms and Abbreviations 1

1.3 References and Acknowledgments 1

2 Design 2

2.1 System Modeling 2

2.2 Interface Design 2

3 Implementation 3

3.1 Development Environment 3

3.2 Task Distribution 3

3.3 Challenges 3

4 Testing 4

4.1 Testing Plan 4

4.2 Tests for Functional Requirements 4

4.3 Tests for Non-functional Requirements 4

4.4 Hardware and Software Requirements 4

5 Analysis 5

6 Conclusion 6

Appendix A - Group Log 7

# Introduction

## Project Overview

This project is an animal database that a user can either make an account to log in so that they can create and interact with the animal or browse as a guest to view the animals. The logged in user will also be able to play with and feed the animals. When looking at a list of animals you should be able to sort the listed animals by one of the fields used to make the animal. We will just do a sequence diagram for this project.

## Definitions, Acronyms and Abbreviations

|  |  |
| --- | --- |
| SRS | Software Requirements Specification |
| Database | Where the information for animals will be stored |
| UML | Unified Modeling Language, used to create visuals of how systems works.. |
| Account | How the user’s information will be stored. |

## References and Acknowledgments

Used the Lecture 13 – System Modeling Part 2 slides for the definitions of the diagrams. Other than that, we did not cite or paraphrase any other source to the best of our knowledge for this documentation.

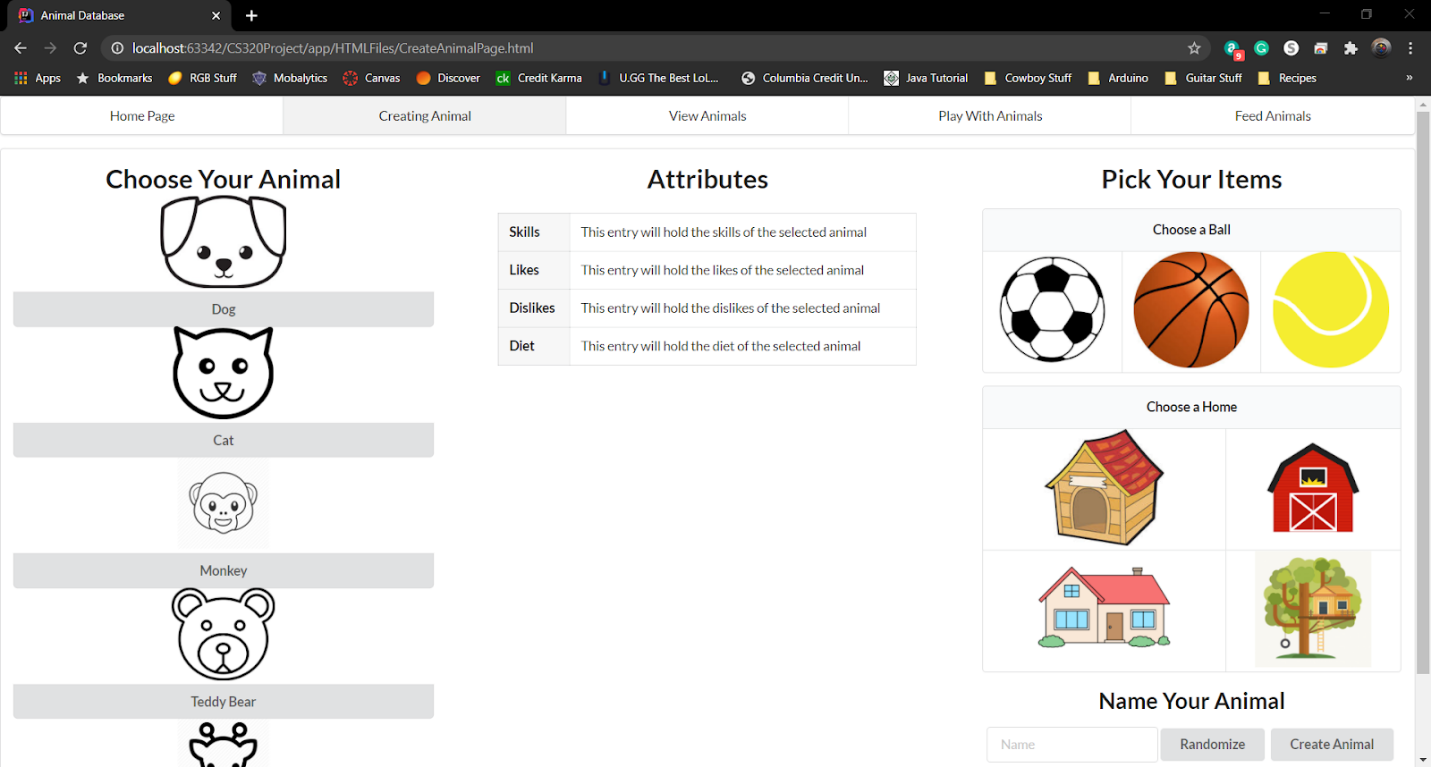
# Design

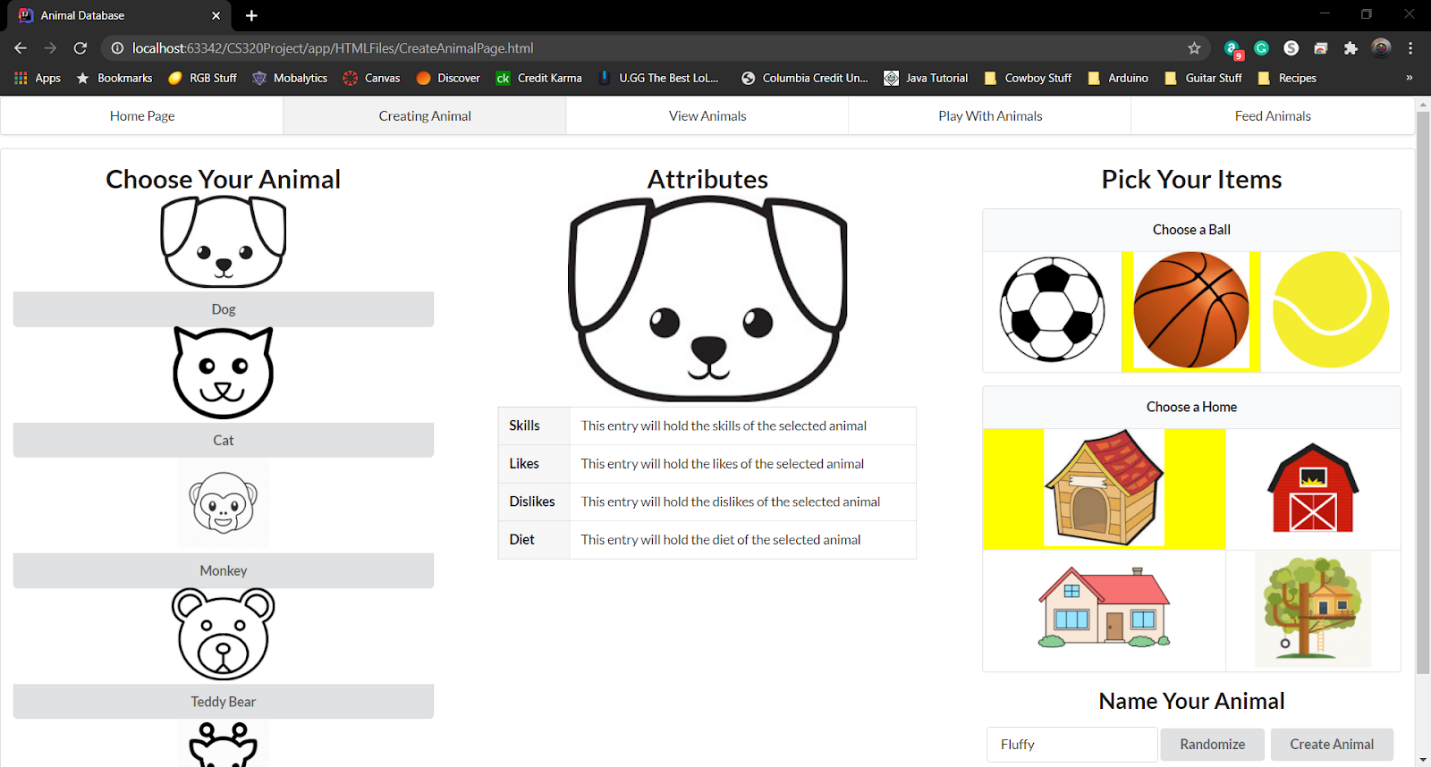
## System Modeling

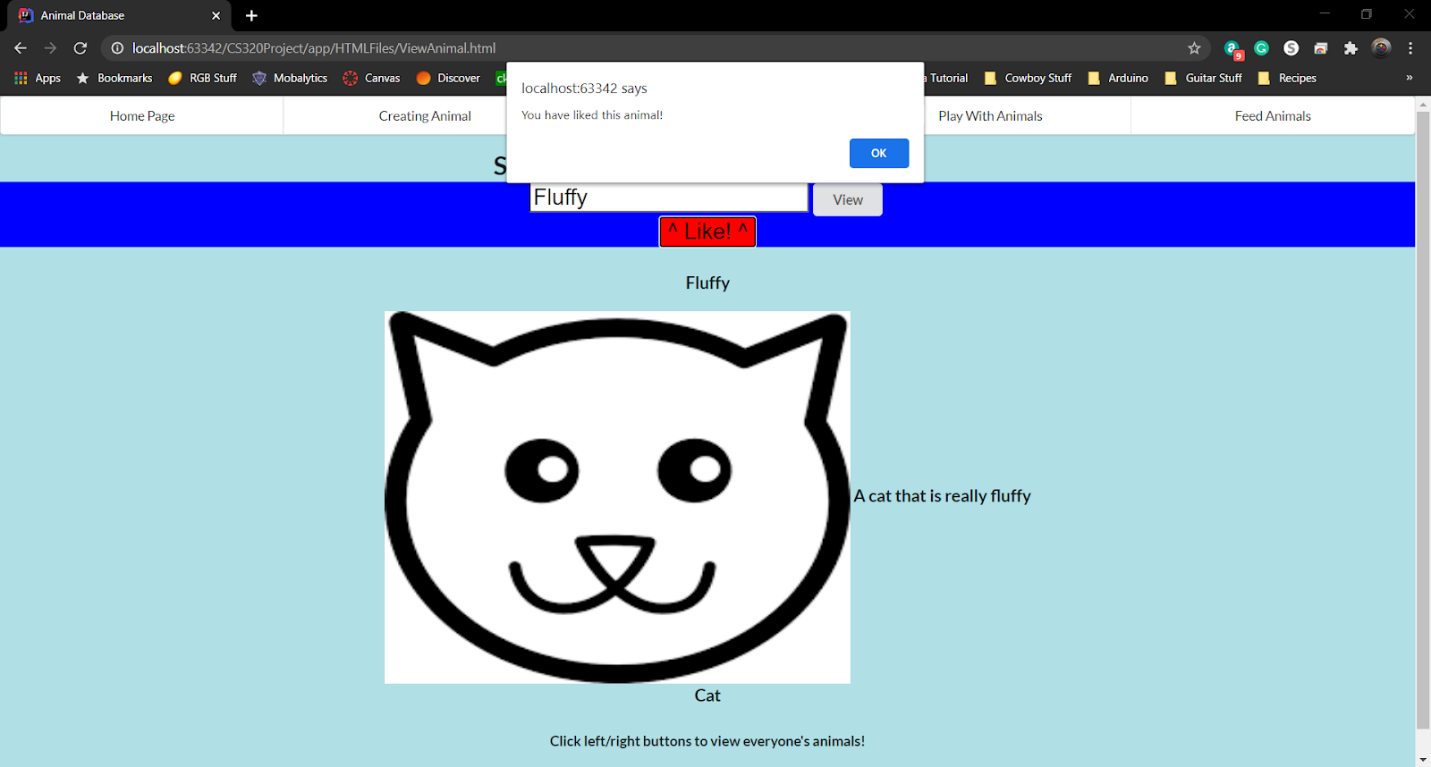
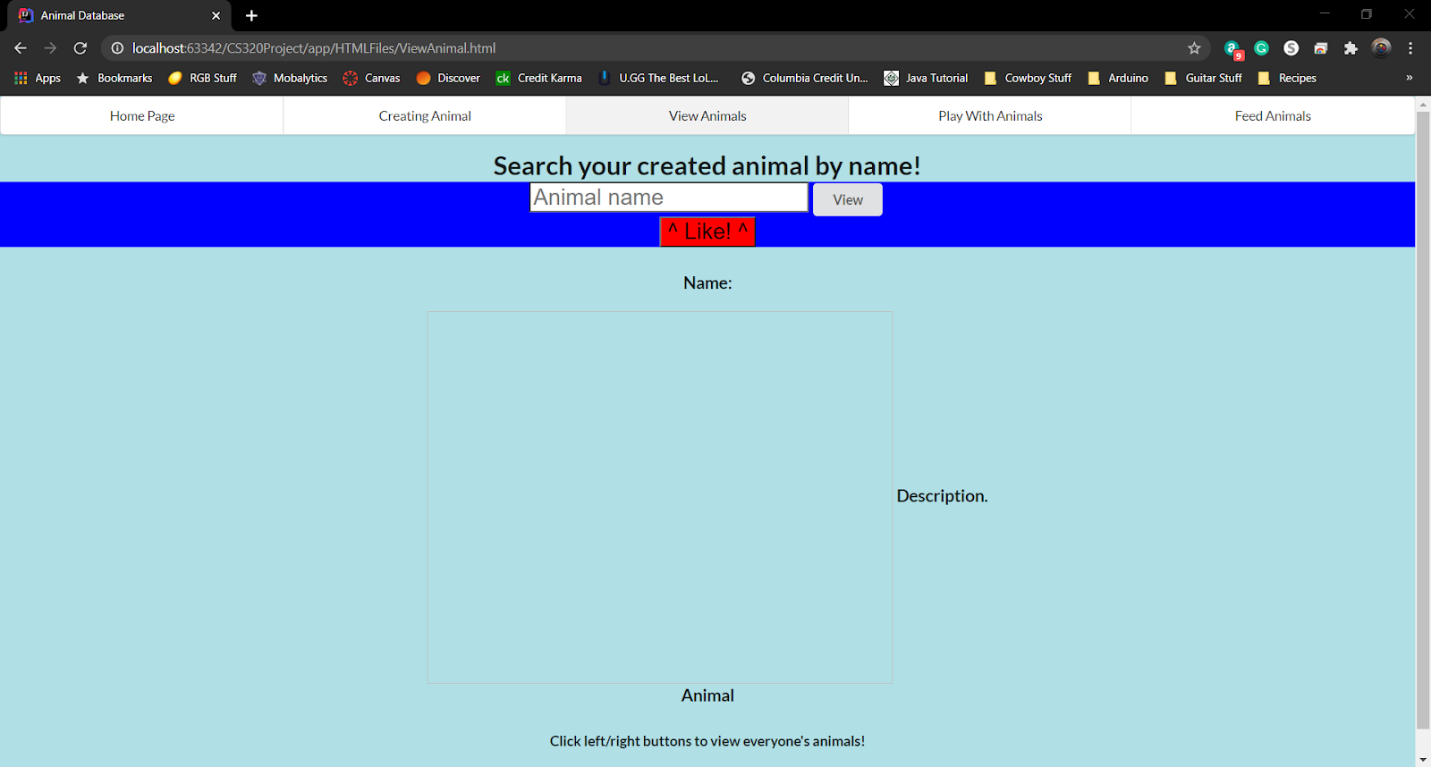
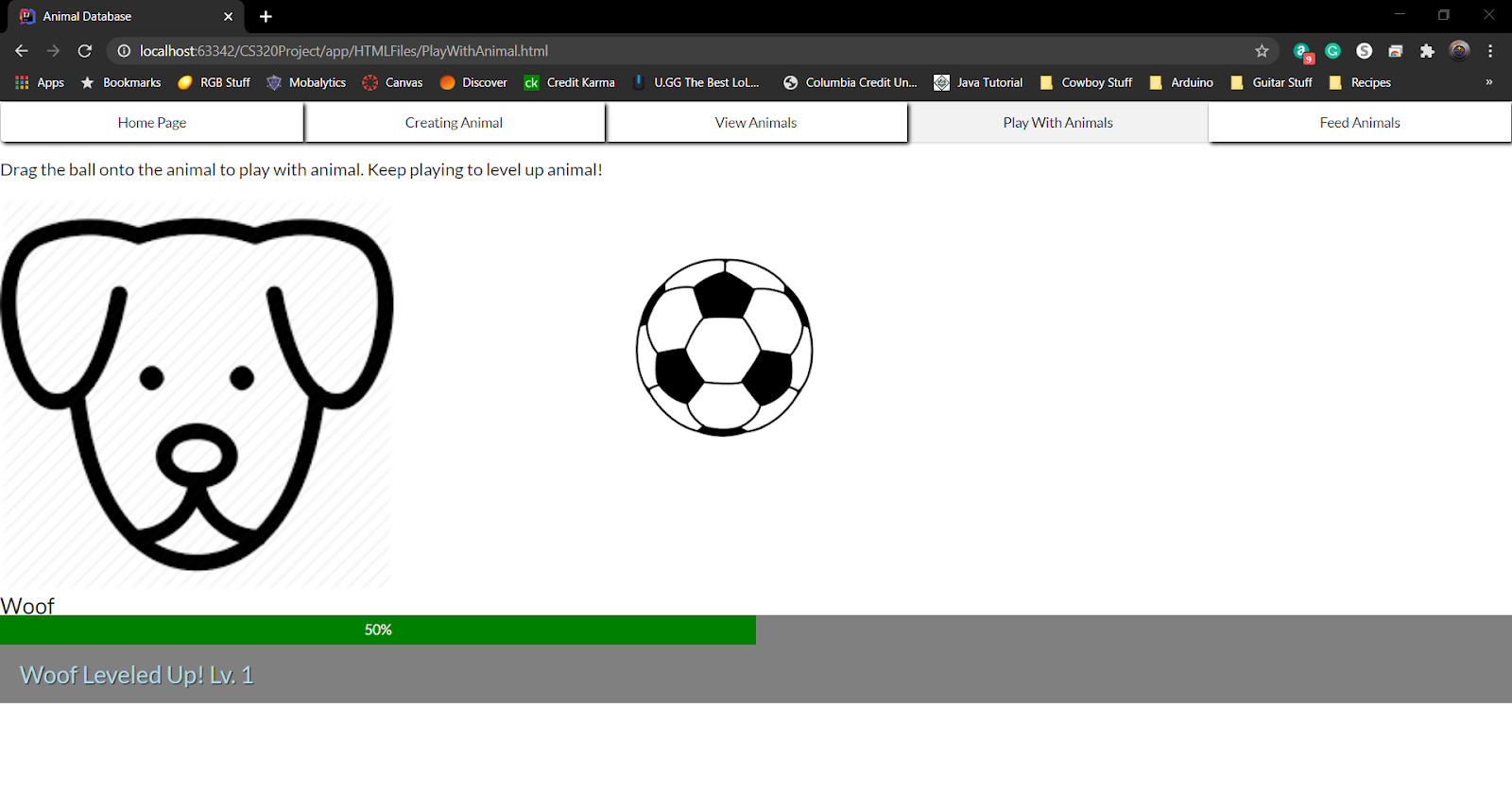
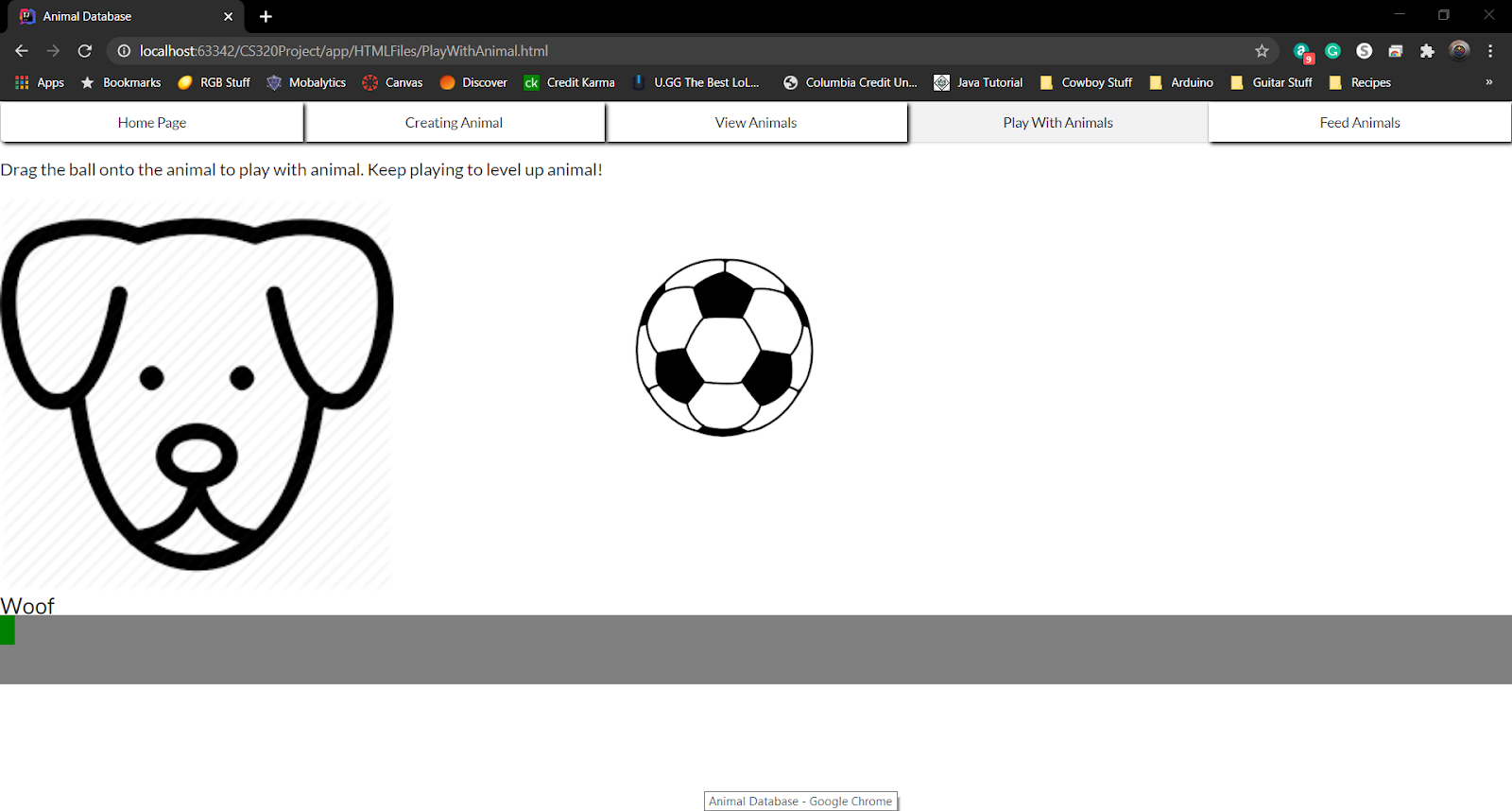
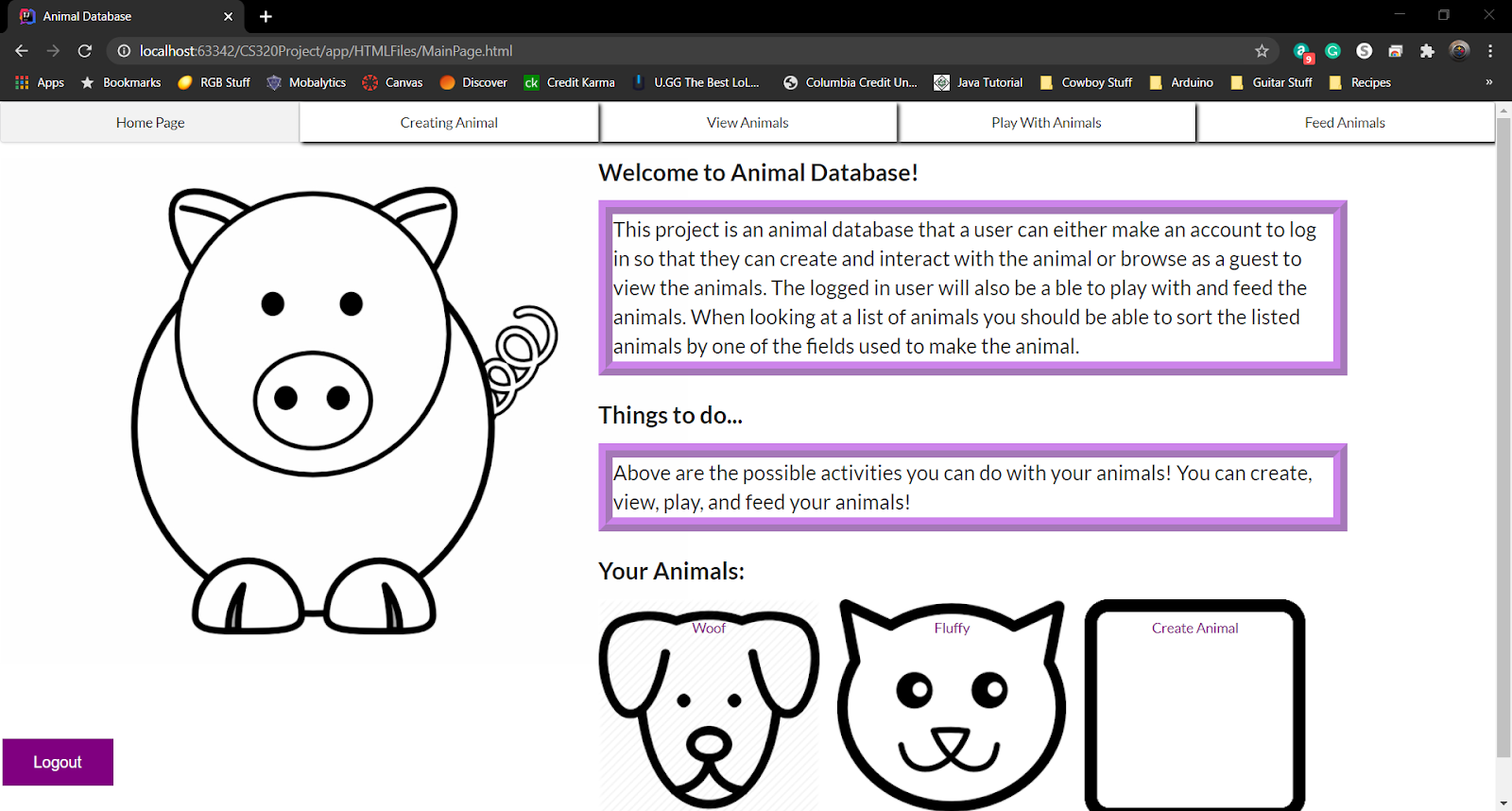
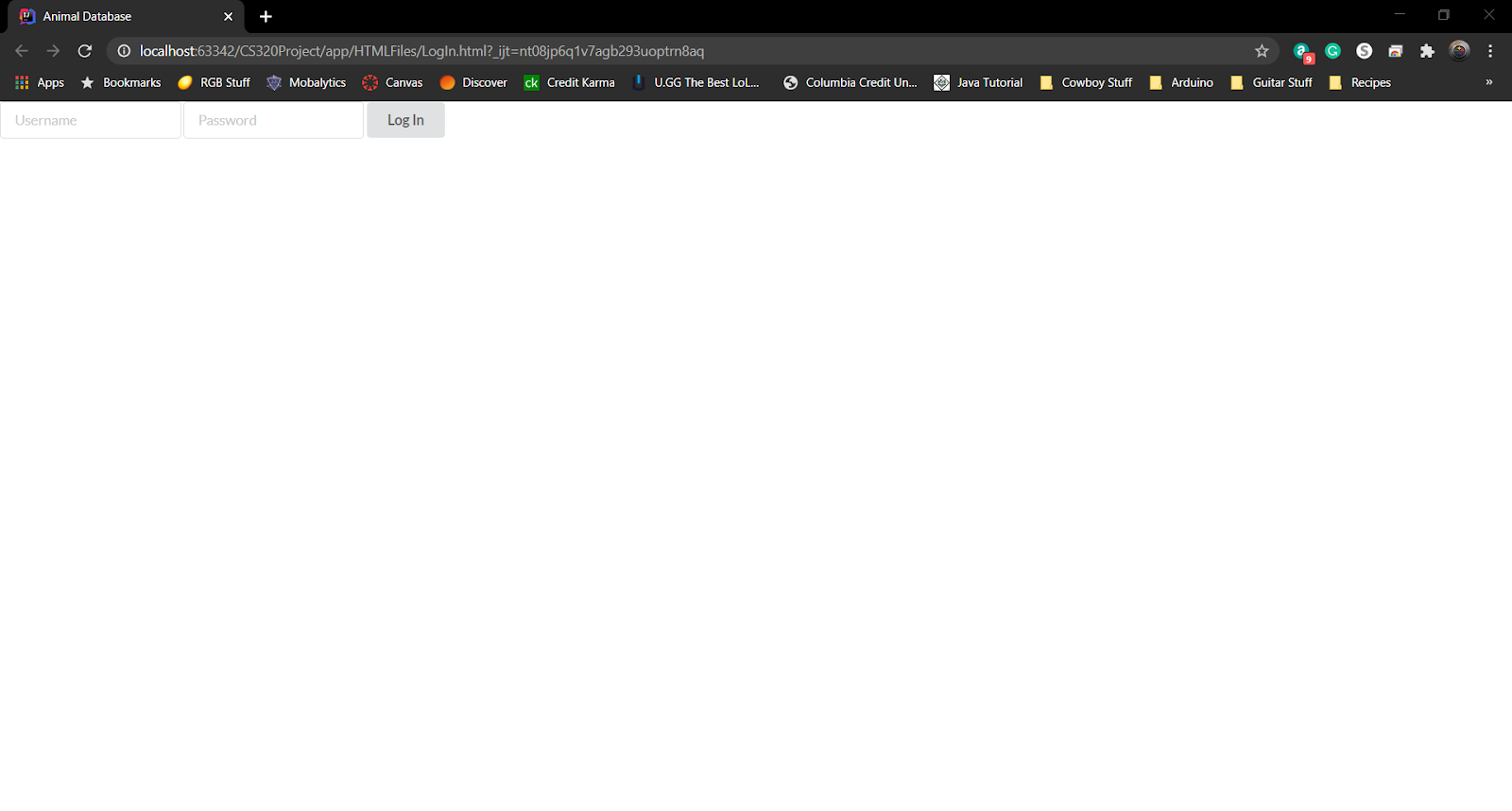
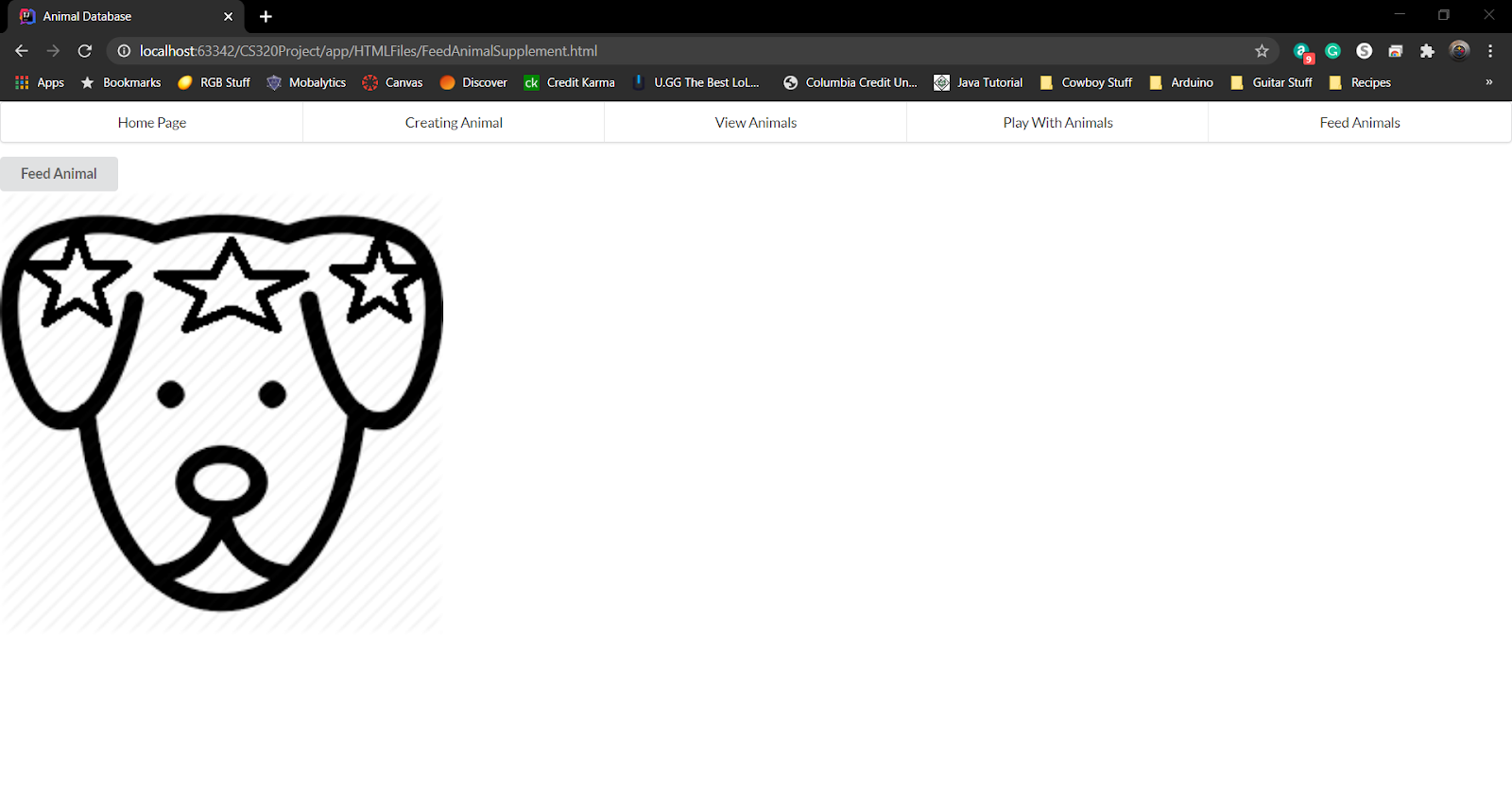
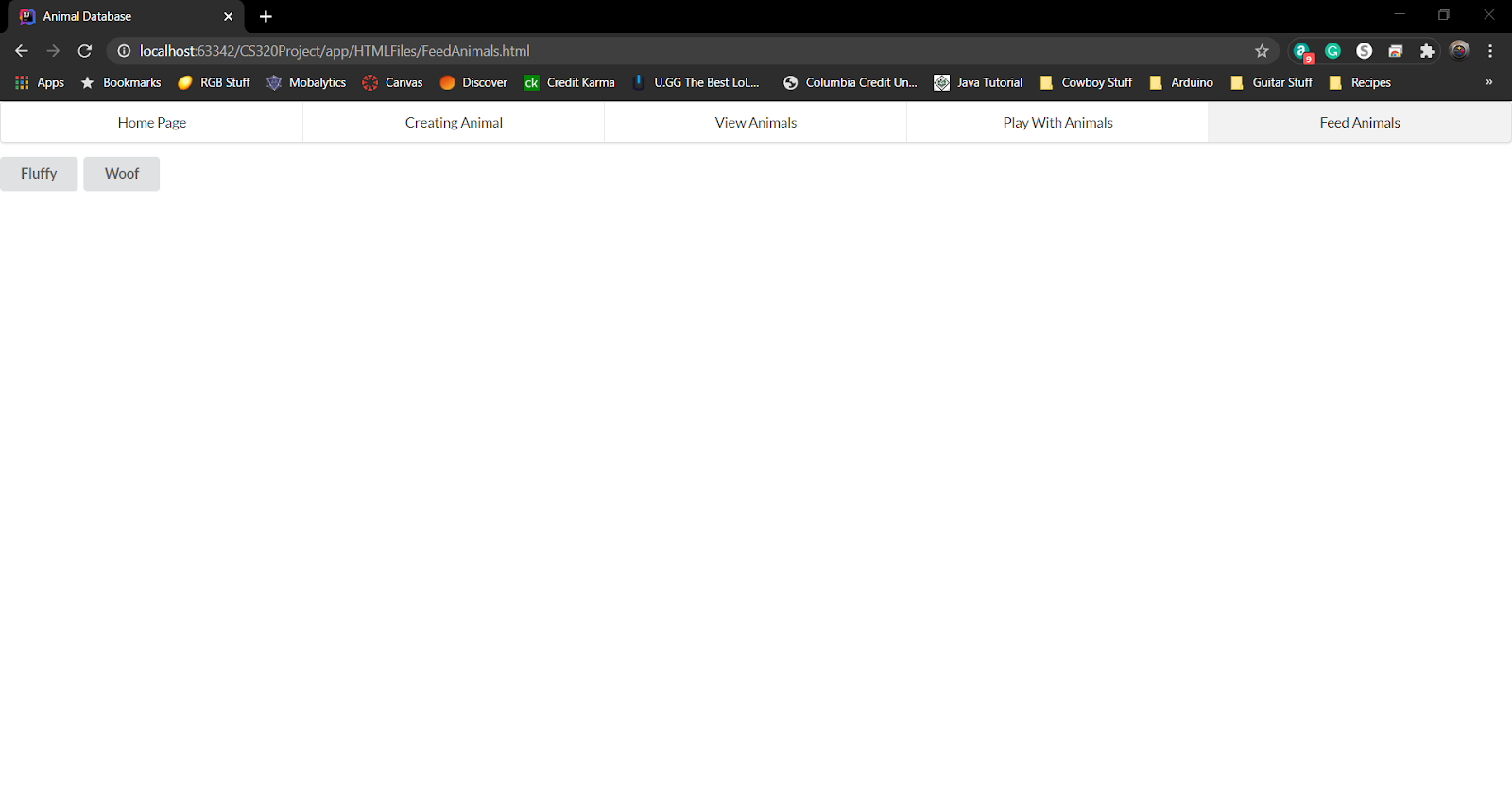
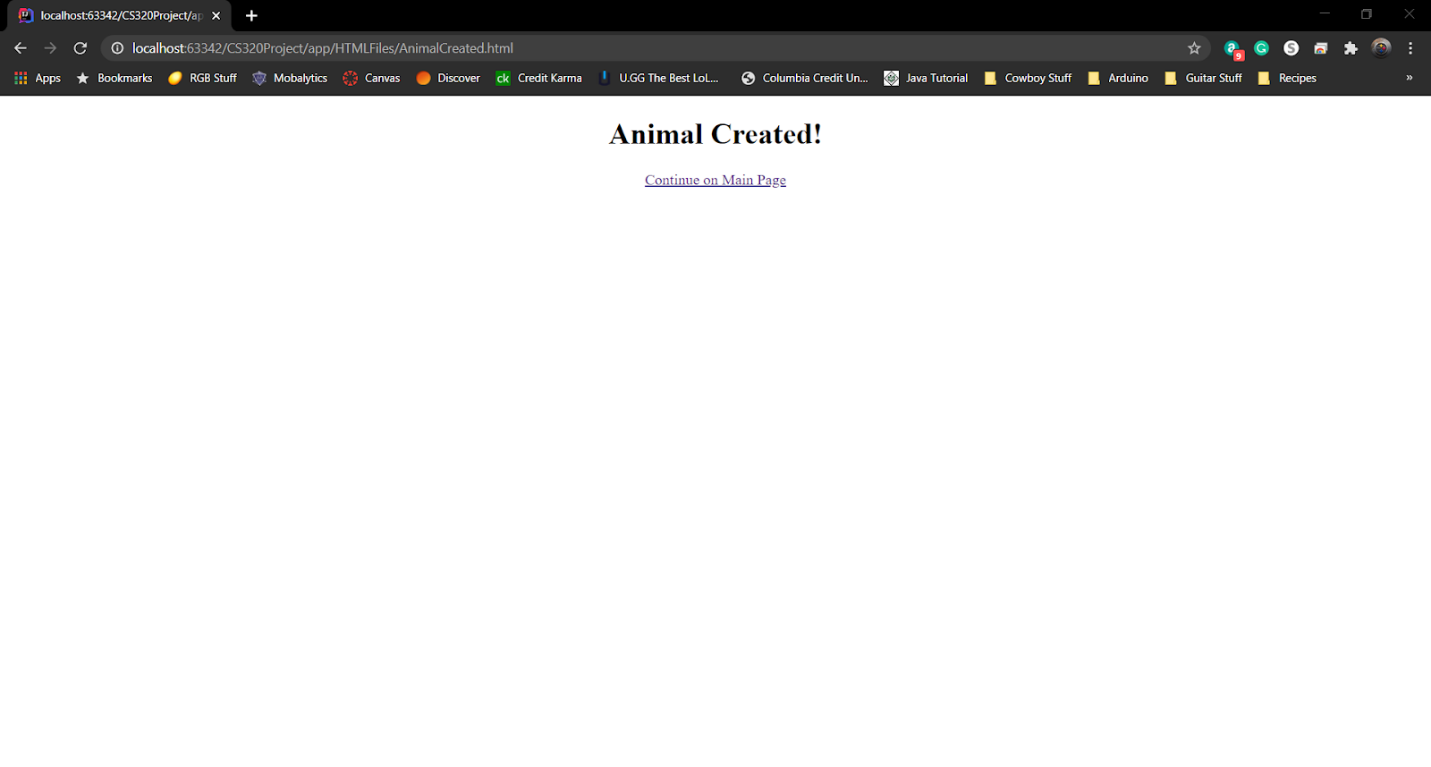
< Update your UML diagrams in milestone 2, to reflect the real implementation of this software.

TO DO: Provide an updated version of the UML diagrams, including use case diagrams, sequence (or state) diagrams, activities diagrams, and class diagrams. If you don’t have an updated version, just mention: “our implementation strictly follows the design document (milestone 2)”. >

## Interface Design



sad



# Implementation

## Development Environment

During development, the majority of coding was done in IntelliJ IDEA and Notepad+. We used Github as our version control system as well as ESLint as a syntax checker.

## Task Distribution

*Design and implementation of Create Animal Page -> John Stevenson*

*Implementation of animal database and temporary storage -> John Stevenson*

*Design of activity diagrams -> John Stevenson*

*Design and implementation of Login page -> Vinh Duong*

*Design and implementation of html headers -> Vinh Duong*

*Group Lead -> Vinh Duong*

*Design and implementation of Main page and Play with Animal page -> Seth Lanante*

*Initial layout of software testing -> Seth Lanante*

*Co-design of class diagrams -> Seth Lanante*

*Design and implantation of View Animal page -> Yevgeniy Diriyenko*

*Design of sequence diagrams -> Yevgeniy Diriyenko*

*Co-design of class diagrams -> Yevgeniy Diriyenko*

## Challenges

Two of our group members were not able to view the final versions of the play with animal and feed animal pages. One of them was able to fix this by switching from Google Chrome to Microsoft Edge when running the html code. We think this may be linked to how cookies are handled in the browser, but we aren’t certain.

Finding a way to store animals took a bit of researching. Most methods we found were either difficult to implement or didn’t quite fit our requirements. For instance, using a file to store long-term information proved to be impossible when running in a browser. We settled on localStorage and sessionStorage as long-term and temporary storage for the app. This allowed us to not only keep track of information across sessions, but we could load session-specific data into temporary storage to easily interface with our stored data (load a user’s animals into temporary storage).

# Testing

## <*This section is a summary of your testing report>*

## Testing Plan

<Describe your testing plan for the project.

TODO: Give a list of items or functions you want to test, and also a schedule for performing the testing. >

## Tests for Functional Requirements

<Describe your test results for the functional requirements.

TODO: Provide a list of use cases or functions you have tested, as well as the testing results (whether or not the system passed the tests).>

## Tests for Non-functional Requirements

<Similar to the Section 4.2, but this section is for the non-functional requirements. >

## Hardware and Software Requirements

<Describe the hardware and software requirements for performing the tests. >

# Analysis

<In this Section you need to analyze the effort that has been put on this project.

TODO: Describe how many hours (approximately) each team member spent on the project, for each milestone, which milestone took the most effort and why. >

# Conclusion

<Conclude the document with what you have learned through working on the project.>

Appendix A - Group Log

< Describe how frequently the group members meet during the semester, and how effective the communication is. This is optional for one-person projects.>