Project – Stray Cat Feeding System

If necessary, additional clarifications may be added to this document but no additional features will be added. Please check this document occasionally. Changes will be clearly indicated in this box.

*This project was inspired by Task8 from the Genius Group Winter 24 competition. Task 8 says “To do something towards kindness, only 1 time during the whole semester (Giving a needy, feeding a cat/dog, or even planting one tree)”.*

A local animal shelter would like to have a website constructed that allows them to install and monitor feeding stations around Qatar for stray cats. The purpose of the site is to ensure that the public is aware of the locations of where stray cats are gathered in larger numbers to help promote volunteers to ensure that cats are receiving food and possibly medical attention.

The shelter has not specified any specific details regarding the screens on the website but indicated that the bootstrap based Core-UI dashboard template must be used. A copy of the template to be used has been placed into D2L. Note that this template is just some sample pages; your application will have different pages (i.e. change the navigation menu to fit your application). Given that the customer has specifically asked for this template to be used, if you do not follow their wishes, your grade will be significantly reduced.

A screenshot of a computer

Description automatically generated

# Requirements

As with many programming projects in real-life, the requirements from the customer are not as detailed as you would hope. This is done to help provide flexibility for the project to allow you to make decisions about how the screens should look and what should be included in the screens. We provide descriptions only of the functionality that is expected.

# Public Viewers

Public viewers are people on the internet who have no credentials to log into the site. These users have a limited view of the system but will be able to do things like:

* Sign up for an account.
* View information about the locations of the feeding areas and posts that have been made about the feeding location.
* View information about the purpose of the website.

# Members

The term *member* refers to any person who has created an account on the site. When signed in, the member should be able to do the following:

* Create new text posts about any feeding site.
* Upload picture for any site. To keep this as simple as possible, you can implement support for just a single photo per post.
* Record details about a visit to the site: how much food and water was placed into the feeding station, how much food is currently there, how many cats were there.
* Report a health issue or other critical item. If a cat at the location appears to be sick or there is some damage to the area, the member can report this.

Although this may seem like a lot you, can actually put most of this together into a single form. Handling file/photo uploads will be covered in one of the lectures and a lab.

# Administrators

The administrators are the people at the shelter. They will be able to perform the following:

* View a dashboard showing the status of the feeding stations.
* View a report of the food situation each site (current level of food, date of last visit, etc.). This report must be in the form of a graph; see the template for examples of how these can be created.
* View urgent items (medical, feeding station problems, etc.).

# All Users with Accounts

* Manage credentials (i.e. change password).
* Reset password by email. For this feature, just print the email message to the console. We will implement password resets in one of the labs.

# General Comments

* Use a custom 404 error page. There is one in the core-ui template or you can add something else. We do not want to see any default express style messages such as “Cannot GET /url”.
* Do not use any JavaScript “alert” or “prompt” dialog boxes. If you need things to “pop up”, then use the Bootstrap **Modal** templates provided.

# Technical Requirements

* Software must be implemented using NodeJS.
* You are free to use client-side JavaScript but it is not required.
* Data must be stored in a cloud-based MongoDB instance. The design of the database is up to each group.
* No additional node packages are permitted beyond the ones introduced in the lectures/labs. If you are unsure, please ask your instructor.
* Protection against CSRF attacks must be implemented.

# GitHub

* You must use GitHub.com for this project to store the code.
* There is no need for GitHub Pro or GitHub Education for this project, just a regular free account is sufficient.
* You are free to use GitHub desktop, git bash, or any other application to interact with the repository.
* The use of features such as “Issues”, “Project”, “Pull Requests”, “Branches” are optional and up to the individual groups. We just need “pull”, “push”, “commit” for this project.
* You must sign up for an account at github.com using your UDST credentials. If you are having difficulty configuring or using the account, many people in the CCIT help center should be able to help with this in addition to the course instructors.
* One team member should create a private project and then add other team members and the instructor as a collaborator.

# Grading Rubric

A grading rubric will be provided in a separate document.

# Due Dates

## Team Formation and GitHub: March 5 (-2 points penalty for late)

Being able to work in a team environment is an important skill that is necessary to work in a real-life career outside of school. You are strongly encouraged to work with a group of people who are around your same academic level. Working with others who have similar skill levels will help everybody in the group to improve.

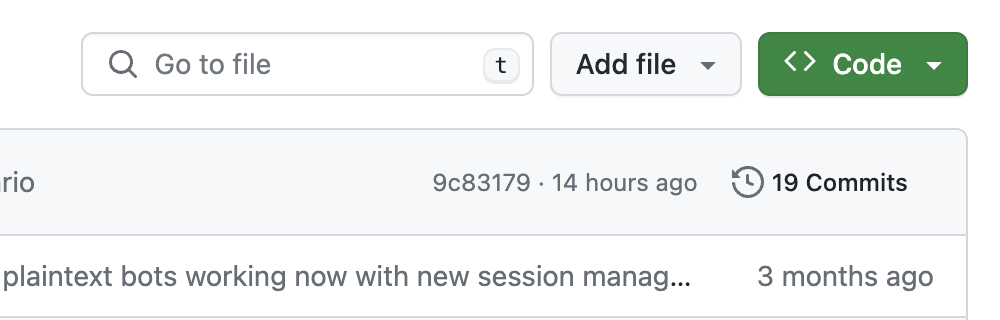
While it is important to develop skills of working together, we also understand that some situations might arise where collaborative work is not ideal for everyone, therefore it is allowed to work individually.

Teams must consist of between 1 and 3 students. Any requests for groups of 4 or larger will be denied. Once you have decided on your group, please email your LAB INSTRUCTOR so they can configure the group dropboxes.

Once you have formed the team, create the project on GitHub (keep it private) and add the members plus the instructor as collaborators. Please check with the instructor about the email address to use; it is expected they will use their UDST credentials however they might have an existing account they would prefer.

## Collaborative Work: April 18 (2 points)

Each team member is expected to be working on the project from the start until the end. Please submit screenshots of the commits for your project to the appropriate dropbox. You can find this history by clicking on the number of commits.



Put screenshot or screenshots of the page from GitHub.com into a word document and submit to the D2L dropbox. Although the instructor has an account, D2L is still the official place where things need to be submitted for archiving purposes.

The instructor will review to ensure that team members are contributing equally to the project through the project period (we will not be looking for commits during the Eid Holiday). Make sure that you do not have just one person doing all the commits, and make sure that you do not have just a lot of activity during the last day.

All commits must have reasonable comments explaining what has been changed in the committed work.

The image(s) of the history must be submitted on April 18. You can submit this early if the project is completed.

## Milestone Check: March 24 (8 points)

The milestone check is an early submission to encourage students to work sooner on the project and not leave everything until the last minute.

The following items will be evaluated at this time:

* The core-ui template does not need to be implemented at this time but you are encouraged to have it.
* The public viewers can see the list of feeding sites but you do not need to support the posts about the site yet. The data must be dynamic (i.e. it comes from the database).
* Session system is working. This means that you must be able to log in as a member or an administrator and see different screens. What you see when you get logged in is not important but it should be using the core-ui template.

You must submit a zip file of the code to the D2L dropbox. While the instructor is a collaborator on GitHub, we must have a copy of the project submitted to D2L.

## Final Delivery: April 18 (20 points)

All functionalities will be evaluated at this time.

Due to the Eid Holiday and Final Exam schedule, there will be no official interview required by all teams. However, instructors may request by email that some groups and/or students to appear for an interview. Failure to attend the interview may result in a reduced grade.

If there are special configuration instructions, you need to inform the instructor in the form of a README.md file. Without such a file, the instructor will run “npm install” to install any packages specified and then run “node web” to run the application called web.js.

You must submit a zip file of the code to the D2L dropbox. While the instructor is a collaborator on GitHub, we must have a copy of the project submitted to D2L.