

// TODO: Team Name

Nugget

Fall 2017

<https://github.com/mcculloughsco/CompSci326TermProject>

Note1: Github submission is on the 'Master' branch, not the 'dev' branch.

Note2: For better folder sorting, the project 1 directory is called `template_files` and the project 2 directory is called `django_application`. `Django_application` is where you will find the django app for our project.

*****LOGIN CREDENTIALS FOR A TEST USER*****

Username: testuser // Password: testing123

The Team:

Arwa Farrag (arwaGypt), Scott McCullough (mcculloughsco), Pinak Kapoor (pinakkapoor), Malachai Purgahn (mpurgahn), Emily Goroza (egoroza), Sangmin Yun (joeyun)

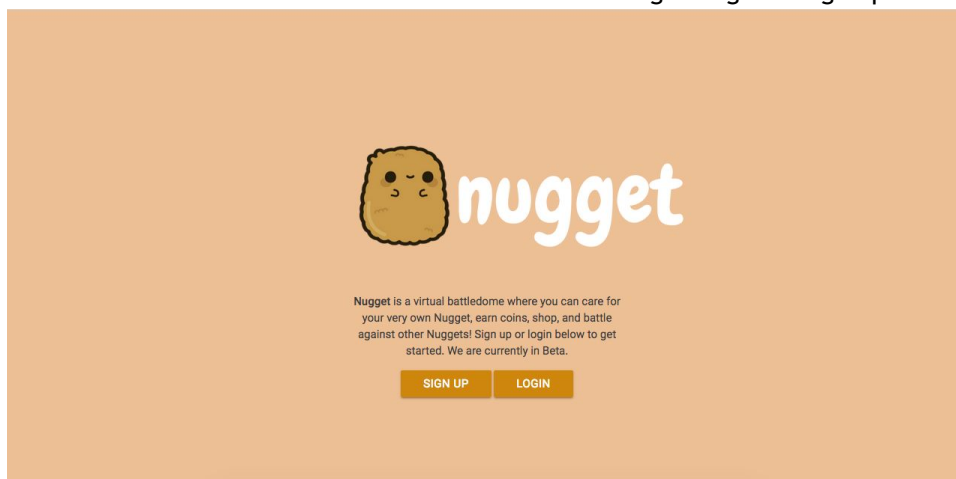
Overview:

Nugget is a virtual pet website in which individual users can login, create and customize their own pet, and interact with others. Users of Nugget have the ability to create their own '*Nugget*' and customize them to their hearts content. They can access the Store and purchase items to customize their Nugget, and interact with other users through the Chat feature. The site largely functions as an interactive social network of these Nuggets.

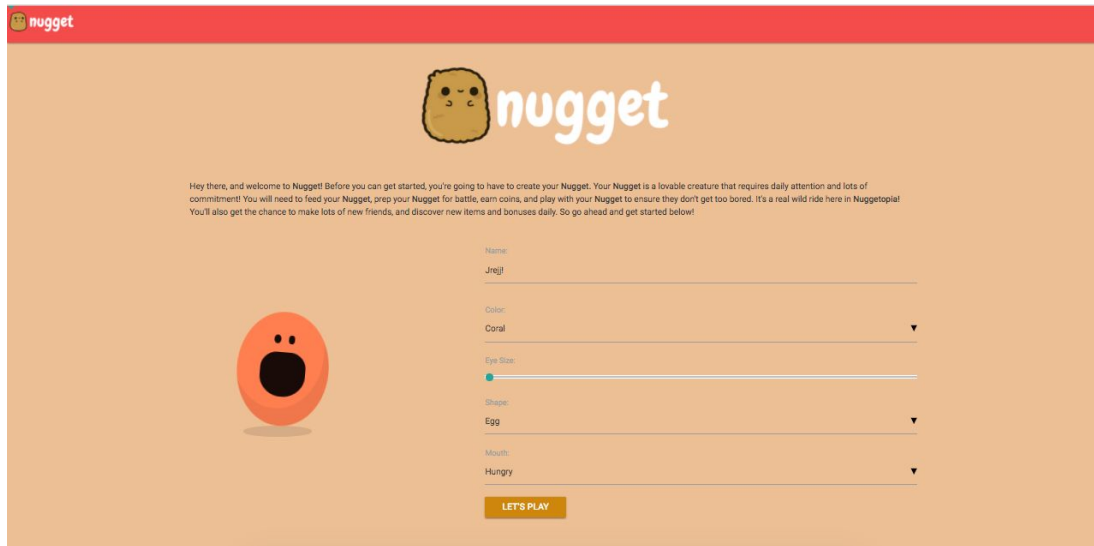
Nugget takes inspiration from 2000s based pet websites such as *Neopets*, *Webkinz* or *Club Penguin* in that users login and interact with each other. Our application brings these applications into the modern era, creating a new website that feels like a modern web application. Where our application differs is that it focuses less on the 'game' part of these sites and focuses on the customization/social aspect. Granted, our site still features a 'battle arena' in which users compete and wager their Coins.

User Interface:

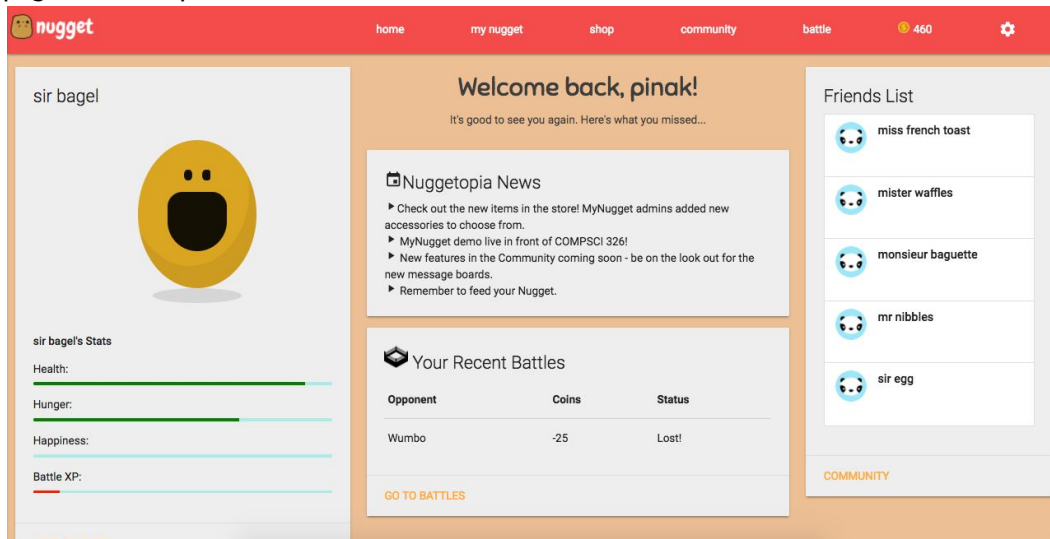
- **(index.html):** First is the landing page. This is the page people see when they first get to our website and aren't logged in. This is different from the login page in that it gives users information about what the website is without needing to login or sign up.



- **(create-a-nugget.html):** This page is the first page that users see after they sign up. Users will be redirected to this page until they create their nugget, and then are placed into the actual content of Nugget. This page will allow user to customize their Nugget, and change features about them such as the look, some of their attributes, and set up their Nugget to then interact with other users.

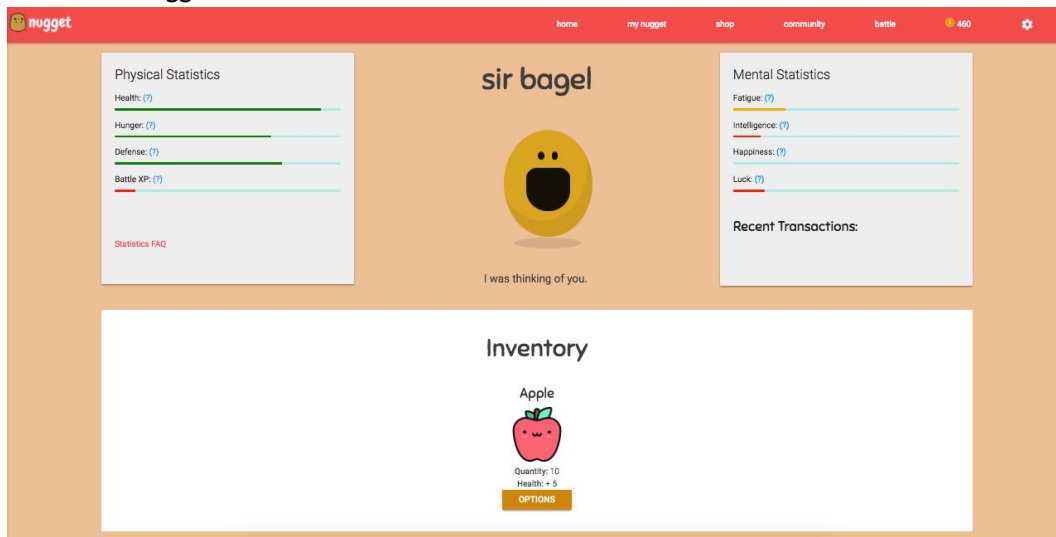


- **(home.html):** This is the page that users see when they log in (after they have created their Nugget). This is essentially the main dashboard of the site, sort of like a user's main feed on Facebook. From here, users can see quick information about their Nugget, as well as news about what is happening around the site (for example, there might be a special event for Halloween that will be advertised here). Most likely more information will be added to this page as development continues.

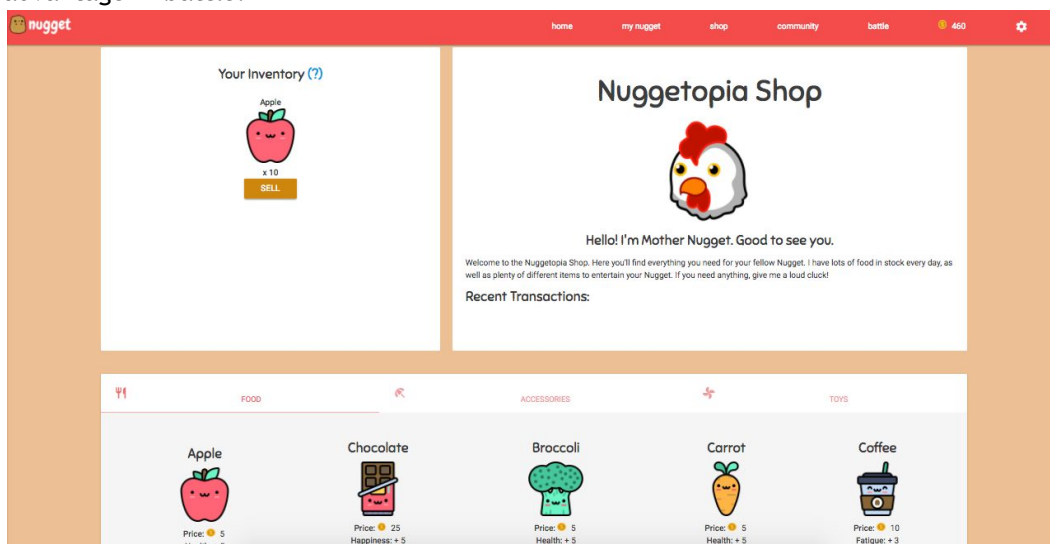


- **(nugget.html):** The My Nugget page is where users can interact with their Nugget, and find out information about their Nugget, and use items to either customize the look of their Nugget or

alter their Nugget's stats.

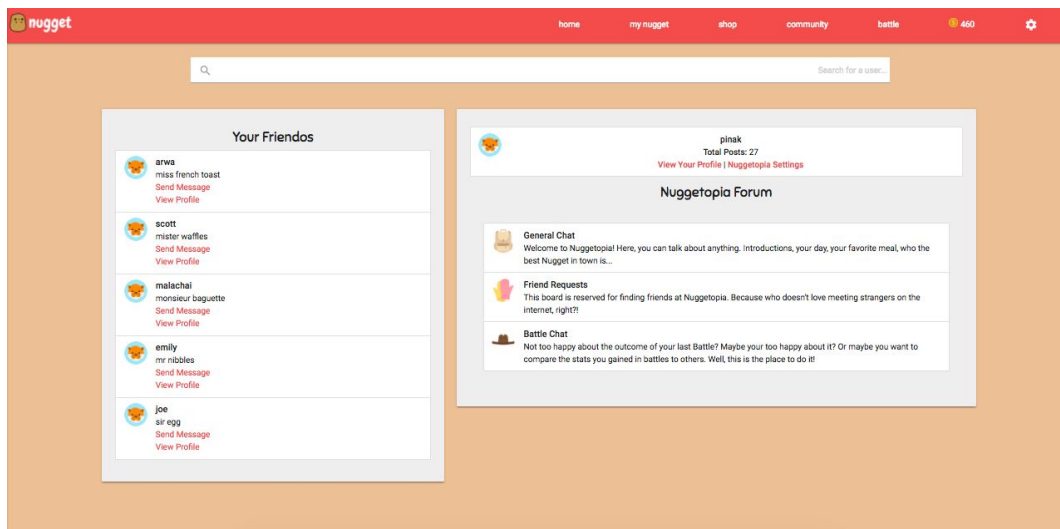


- **(shop.html):** The Shop is the place where users can purchase items using their accumulated Coins. From here they can buy items like hats that will be added to their inventory to customize their Nugget. Users can also buy items that may boost certain stats to give them an advantage in battle.

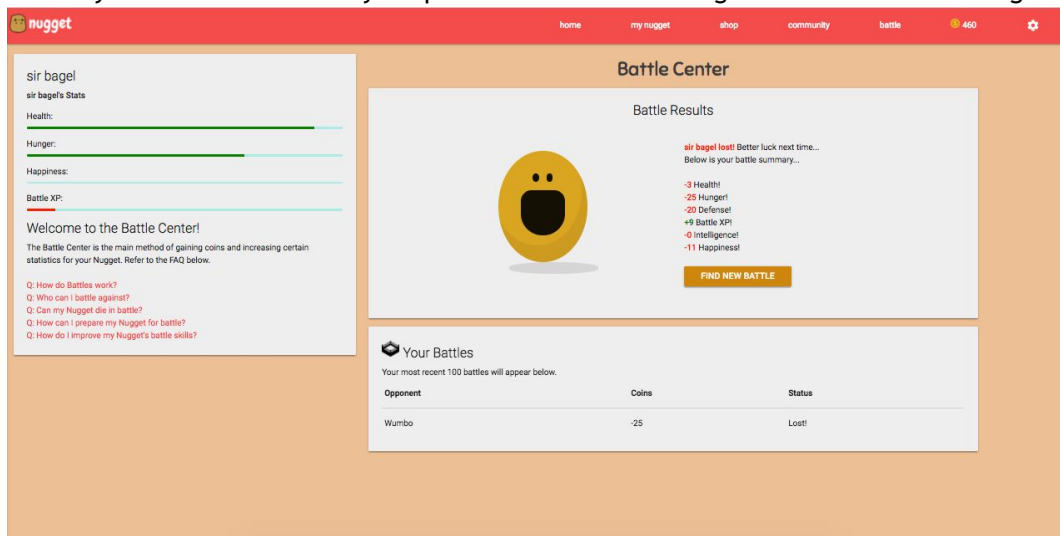


- **(community.html):** The community is a forum where users can directly message their friends or other members of the Nugget community. Here the user also has access to their friends list

to see with whom to connect with.



- **(battle.html):** Finally, the Battle page is the page where users use their Nugget for battle. The battle system will be relatively simplistic as to avoid making it feel too much like a game.

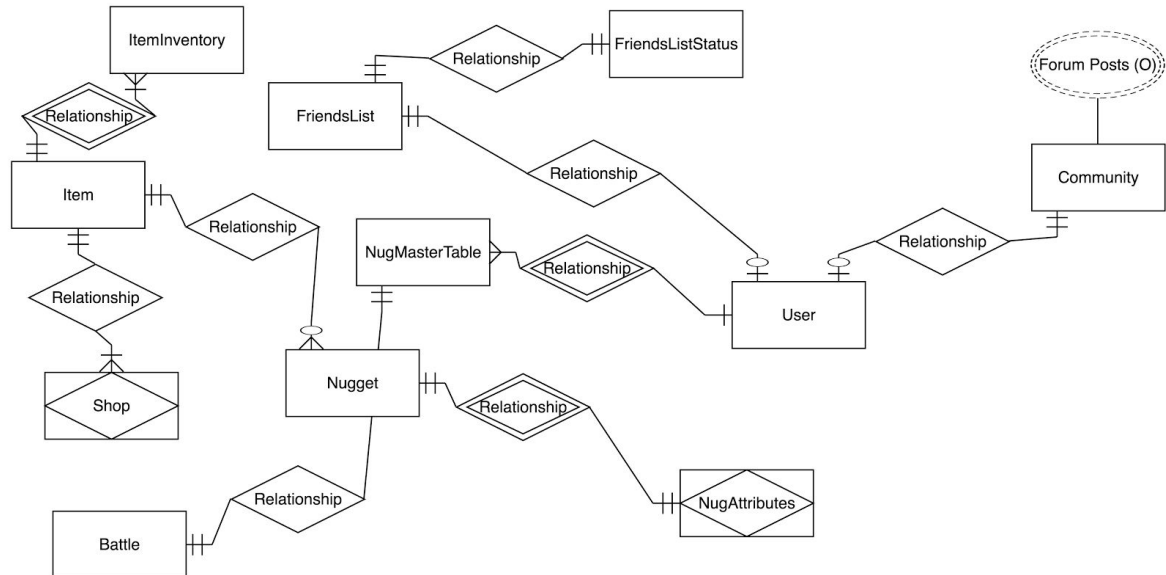


Data Model:

A final up-to-date diagram of your data model including a brief description of each of the entities in your model and their relationships.

- NugMasterTable contains a repository of all created NuggetUser IDs and is an identifying relationship, with a link to each user.
- The User entity contains a one to many relationship with the NugMasterTable and an optional one to one relationships with Friends List and Community. This being a Nugget user can not have any friends or posts in the community.
- The community entity contains a one to one optional relationship with each user, we pointed out the attribute containing posts to show that each post is an attribute.
- The FriendsList contains a one to one mandatory relationship with the FriendsListStatus entity as to maintain order in the friend world.
- The Item table contains a one to many relationship with ItemInventory as one item can belong to many inventories.
- The shop contains a one to many relationship with items as many items belong in the shop.

- The Nugget table contains a one to one relationship with NugAttributes as an identifier for unique characteristics of each Nugget. Also has a one to many optional relationship with Items, as one user can have many, one or none items.
- Battle contains one to one relationship with the Nugget as it further connects it to the FriendsList to integrate the social aspect.



URL Routes/Mappings:

URL Mappings (Currently Supported)	Description	Permission
r'^\$',	Maps to the index of our website	Login Not Required
r'^home\$'	Routes to the homepage of a logged in user	Login Required, Nugget Created
r'^nugget\$'	Routes to the Nugget page of a user	Login Required, Nugget Created
r'^shop\$'	Routes to the Shop page	Login Required, Nugget Created
r'^community\$'	Routes to the Community page for a user	Login Required, Nugget Created
r'^battle\$'	Routes to the Battle page for a user	Login Required, Nugget Created
r'^create\$'	Routes to the create a nugget page for new users	Login Required, Nugget Not Created
r'^help\$'	Routes to a default help page	Login Required, Nugget Created
r'^myaccount\$'	Routes to an account management page for a user	Login Required, Nugget Created

r'^profile/(?P<username>[a-zA-Z0-9]+)\$'	Routes to the profile of a given user by username	Login Required, Nugget Created
r'^hidden\$'	A hidden surprise!	Login Required, Nugget Created

Authentication/Authorization:

Our authentication model uses Django's built in User model. We then link a User to our Profile model which is how our application interacts with a user. The index page doesn't require authentication, but all other views require the user to be logged in and authorized to access it. There are two types of users: admins and standard users. The only admin account is the *compsci326* account which is used for admin console debugging. Otherwise, all other users share the same permissions, which aren't any different from any of the Django tutorials we did in class. The only weird permission is in the create a nugget page. If a user hasn't created a nugget yet, they will be redirected to this page when they login until they create a nugget. Likewise, if a user has created a nugget, they will always be redirected away from the create a nugget page as their nugget is already created.

Team Choice:

Our team choice component stemmed from the idea of creating a more social, up-to-date, online virtual pet game and a component that is quintessential of any game is a live chat component. After much research, we initially began implementation with high hopes and expectations as we found multiple Django resources such as channels and WebSockets; however our hopes came to a quick end with the realization that these applications only worked on a higher Django version. This didn't discourage us, as we quickly were able to resolve the issue with a new solution of creating our own web forum instead of implementing another's live chat. The forum would have been similar to a private message system as it would be one-to-one conversations with other users. We were able to implement the backend related information regarding models for the forum however ran into many issues with implementing its views. This leads us to our team choice being our myriad of exotic CSS elements amongst the others. We extensively worked on various CSS elements that allowed our pages to stand out, this is from the resizable and customizable nuggets, color schemes, carousel of friends customized nuggets and many displays of statistics.

Conclusion:

Our overall experience with this term project was enjoyable minus the many difficulties we had along the way throughout the various submissions. Through the design and implementation process we learned that there are many other css frameworks besides bootstrap that offers very beautiful and varying design perspectives that are very easy to implement. The first hurdle to overcome was having two User models, our User (now called Profile) and Django's user. This altered a lot of how we were accessing the User and retrieving User related data so changing all of that code over took a while. We also ran into a lot of problems with forms. We encountered problems with having multiple forms on one page (especially when it came to combining login and sign up on the same page), editing data in multiple models from a single form, and creating dynamic forms that generate based on dynamic data (i.e. since a user can have a variable number of items in their inventory, generating forms to accommodate the dynamic nature of the inventory). One page we had the most difficulty with was the battle page. We had difficulty dynamically loading the friends of the user on the page to select an opponent. Another difficulty was getting the other user to accept or decline the battle and having that show up in the views for that user's battle page. There were a few topics we wished we would have learned more about before tackling our models. This includes a more clearer understanding of how to work with and around the django application and more planning with our data models as we were not expecting to have so many models in our final application.