PET SHOP WEB APPLICATION

USING ANGULAR 4 AND TYPESCRIPT

Submitted by:

SRIRAM KRISHNAN

(skrishn5@binghamton.edu)

Under the Guidance of:

Prof. Leslie Lander

Department of Computer Science

Thomas J. Watson School of Engineering and Applied Science

Binghamton University

Table Contents:

1. INTRODUCTION:

A pet shop web application is a place where user can add pets, edit the details of the pets and select the pet food appropriate for the pet. It allows the user to send the pet food to the shopping list by clicking on the to shopping list from the drop down. There is a shopping list where the user can add pet food to the page, delete them and clear the text entered. It is a single page web application which is achieved through angular taking care of the routing and component aspects and typescript facilitating business logic.

Bootstrap CSS has been used for styling of the web page and angular CLI to ease out the creation of the web application.

2. PROJECT OVERVIEW:

The main purpose of the application is to add pets to the list, edit and manage the pets.

Main Features:

1. Add Pets.
2. Edit Pets.
3. Add Images.
4. edit pet food.
5. Add pet food to the shopping list.
6. Edit shopping list.
7. Delete Pet food from shopping list.
8. Clear the text in shopping list.
9. Save and Fetch Data.

3. Run the Application:

I had done this project using the new version of Angular and typescript. The reason I chose this JavaScript framework was it's very recently introduced into the market and it's an updated version of angular 2 thereby providing certain cool features as compared to angular 2. The project has been done using the webstorm IDE. It can also be done using sublime and other editors. I had used angular cli to ease out the coding as it helps in creating templates for components and directives. The code has also been written in such a way that user understands it and if any bugs in the future he can easily debug it.

The following steps are required for running the application:

1. Install npm in your computer.
2. In the terminal type npm install –g @angular/cli
3. Install bootstrap npm install –save bootstrap
4. Add bootstrap path to styles in angular.cli. json file
5. Then type ng serve
6. Type the localhost url localhost:4200
7. The application should be up and running.

4. TECHNOLOGIES USED:

4.1 Hardware Requirements:

1. Computer with processor (at least 1.5 Ghz)
2. Ram – 512 MB (at least)

4.2 Software Requirements:

1. operating system (Windows/Macos) any version
2. Angular CLI
3. NPM
4. Bootstrap
5. Webstorm
6. Firebase account

5. IMPLEMENTATION:

For implementation I used Angular and Typescript.
The Angular project comprises of the following:

1. Components
2. Directives
3. Routing
4. Services
5. Modules
6. Model

Angular startup:

1. When angular application loads main.ts is the first one to start in which the root module is bootstrapped which is similar to ng-app in angular 1.x.
2. Next it goes to the root module typescript and we have a bootstrap array which tells the root component we are looking for.
3. Angular analyses the root component typescript file and checks the setup. It knows the selector and template url through the component decorator through which it knows which html file should be displayed.
4. Now with content of the root component angular finds index.html and replaces the <app-root> section with the root html file.

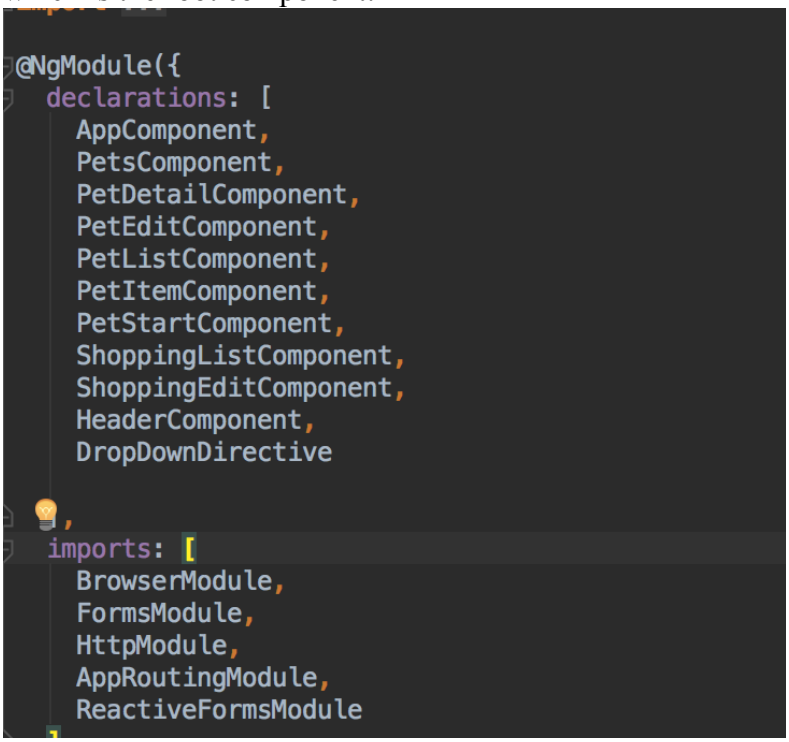
The root content for this application is named app.

Components:

To add a component, the following must be done:

1. Create a typescript class.
2. Import component decorators and add selector, template url and styling url to the component decorator.

3. Then provide the component in the NgModule decorator of the appcomponent.ts file which is the root component.



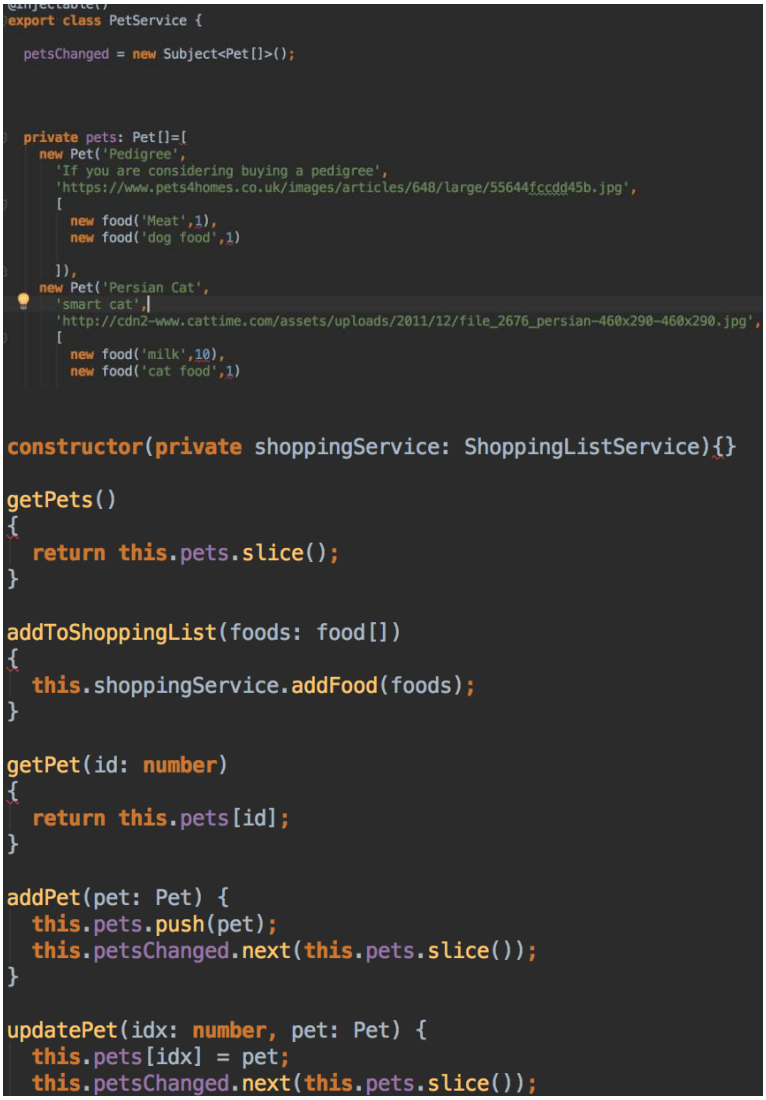
The hierarchy of the components is as follows:

1. App component is the root component.
2. It has three children header, Pet, Shopping List.
3. Pet has 3 children Pet detail, pet edit, pet list.
4. Pet list has a component within it called pet item.
5. Similarly, for shopping list it has one child shopping edit.

Services:

There are two services which takes care of the communication between different components.

petServices.ts:



```
export class PetService {

  petsChanged = new Subject<Pet[]>();



  private pets: Pet[]=[
    new Pet('Pedigree',
      'If you are considering buying a pedigree',
      'https://www.pets4homes.co.uk/images/articles/648/large/55644fccdd45b.jpg',
      [
        new food('Meat',1),
        new food('dog food',1)

      ]),
    new Pet('Persian Cat',
      'smart cat',
      'http://cdn2-www.cattime.com/assets/uploads/2011/12/file_2676_persian-460x290-460x290.jpg',
      [
        new food('milk',10),
        new food('cat food',1)


constructor(private shoppingService: ShoppingListService){}

getPets()
{
  return this.pets.slice();
}

addToShoppingList(foods: food[])
{
  this.shoppingService.addFood(foods);
}

getPet(id: number)
{
  return this.pets[id];
}

addPet(pet: Pet) {
  this.pets.push(pet);
  this.petsChanged.next(this.pets.slice());
}

updatePet(idx: number, pet: Pet) {
  this.pets[idx] = pet;
  this.petsChanged.next(this.pets.slice());
```

This service class is called through the component typecript class in case of any events for eg click events.We have a subject observable which helps in subscribing in case of any changes in the application content asynchronously.

Another service that had been created is for shopping list which serves almost the same purpose. We can create event emitters in case we have to propogate the data from a child to parent and provide an @Output() decorator with variable and list to the event emitted in the parent html files and add the business logic to the parents typescript file.Similary we can also provide input from the parent to the child by using @input() in the child which is the receiving end and providing the values in the parent and binding it to the property of the child.

```
<div class="row">
  <div class="col-xs-12">
    <app-pet-item
      *ngFor="let pet of pets;let i=inde
      [petItem]="pet"
      [index]="i"></app-pet-item>
  </div>
```

In the above code you can see the attribute [pet-item] which is the input to the app-pet item component.

You can find the input properties in the below typescript code of the pet-item component.

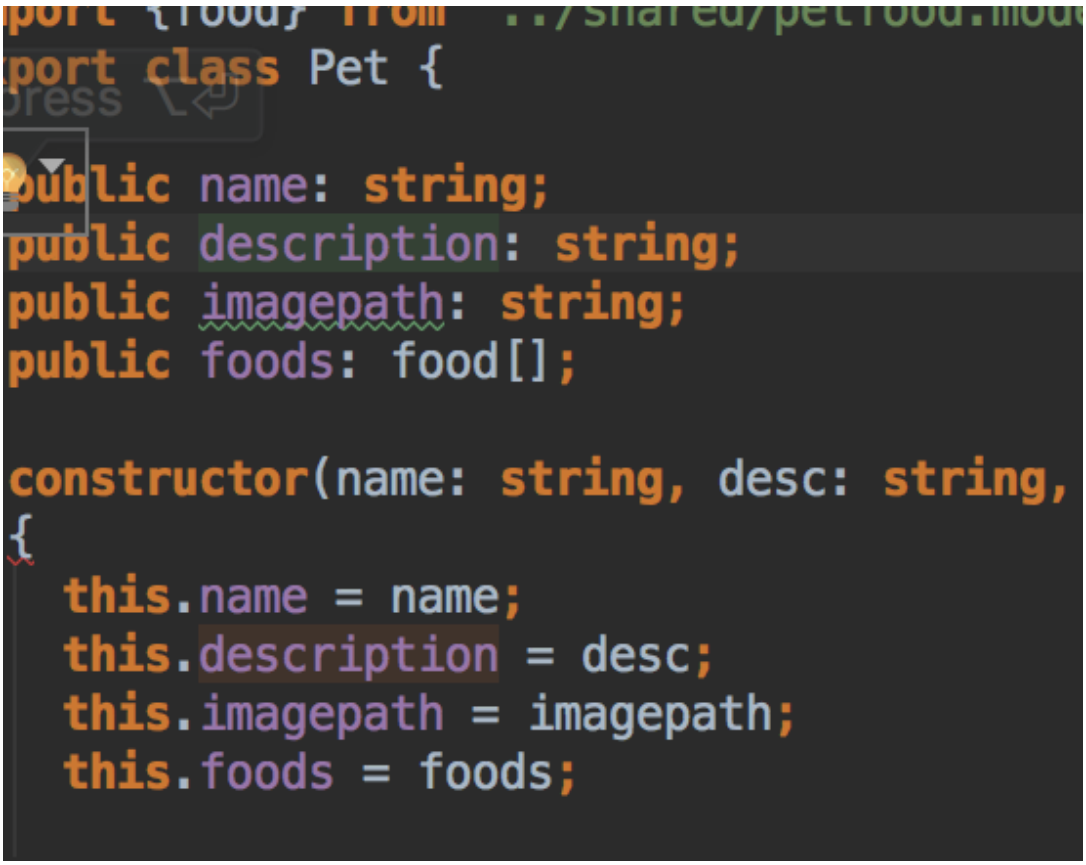
Pet-item.component.ts

```
  selector: 'app-pet-item',
  templateUrl: './pet-item.component.htm
  styleUrls: ['./pet-item.component.css'
})
export class PetItemComponent implements

  @Input() petItem: Pet;
  @Input()  index: number;

  constructor() { }

  ngOnInit() {
```

Model:

There were 2 model classes created for loading the data this can be extended when there are more functionalites introduced and large amount of data has to be used so instead of providing the data in the same ts file we can create a model class which segregates the data from the business logic.

The below model class is petmodel.ts which has a class Pet which can be exported such that it can be imported within other typescript classes.



Routing:

Routing is done by providing a separate module class called app-routing module.ts and registering the routing through RouterModule.forRoot("you routes variable").

The router module is imported in the app-module.ts so that angular knows where to load the routes from.If you look at the above code you can find we can provide parameters and children routes too.
The route is loaded into the component using <router-outlet> and angular decides based on the link clicked what compoenent should be loaded.
We can provide router-link in the html either relative or absoluted based on the location of the path.We can also provide programatically using router.navigate and pass the path which gets mapped to appRoutes loaded in the router module.

Form Submission:

There are two ways to submit form Template driven approach or Reactive form approach.Shopping list components are implemented using TD approach and Pet component is implemented using reactive forms.The difference between these two is the TD approach is taken care by angular.
We can provide a local reference in the html element like #f and then get the entire details of the form using the local reference.we can access the local reference in the typescript file using @Viewchild('local ref') var:type is ngForm.Now using this variable we can access the entire form content and manipulate the data.we should also not forget to provide ngModel attribute to the html elements which registers all the elements to Model attribute making the angular lnow we are using TD approach.Be sure to import FormsModule in the app.module.ts.

In case of reactive approach the programmer has to handle the form this is done by providing property binding [formGroup]="any name" in the form element.This makes us use the entire form group and declare the variable in the TS file whose type is Formgroup .Now we can provide default values and validations on the html elements in the ngOnInit lifecycle of angular.The reactive approach solely depends on the hands of the programmer unlike TD approach which the angular takes care of.Be sure to import reactiveFormsModule in the app.module.ts.

Firebase:

I used Firebase as a dummy backend.It is a google product which allows you to create three projects after which you are priced.I created the database for the pet shop and changed the read and write to true.such that any body can access/modify it.

Thus we can click on manage dropdown and click on save and fetch data to store the data in the firebase databasae.

HTTP

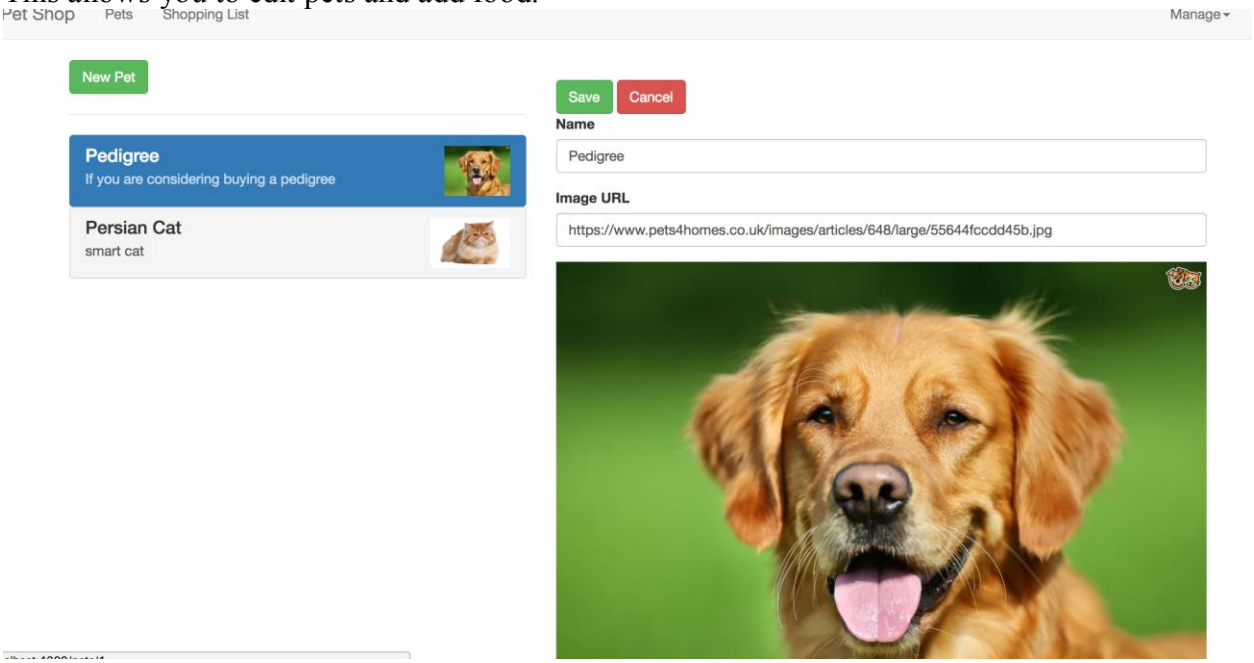
HTTPModule has to be first imported in the import part in app.module.ts.Then a service file is created where we can inject the http service by providing @Injectable().Then we can use get,put,post requestes depending on the requireement.
We can also transform the JSON data received by using the map method and then providing the subscribe method to listen to the response.Unless you provide the subscribe method we cannot see the repsonse as the output of http is an observable so we have to listen to it.

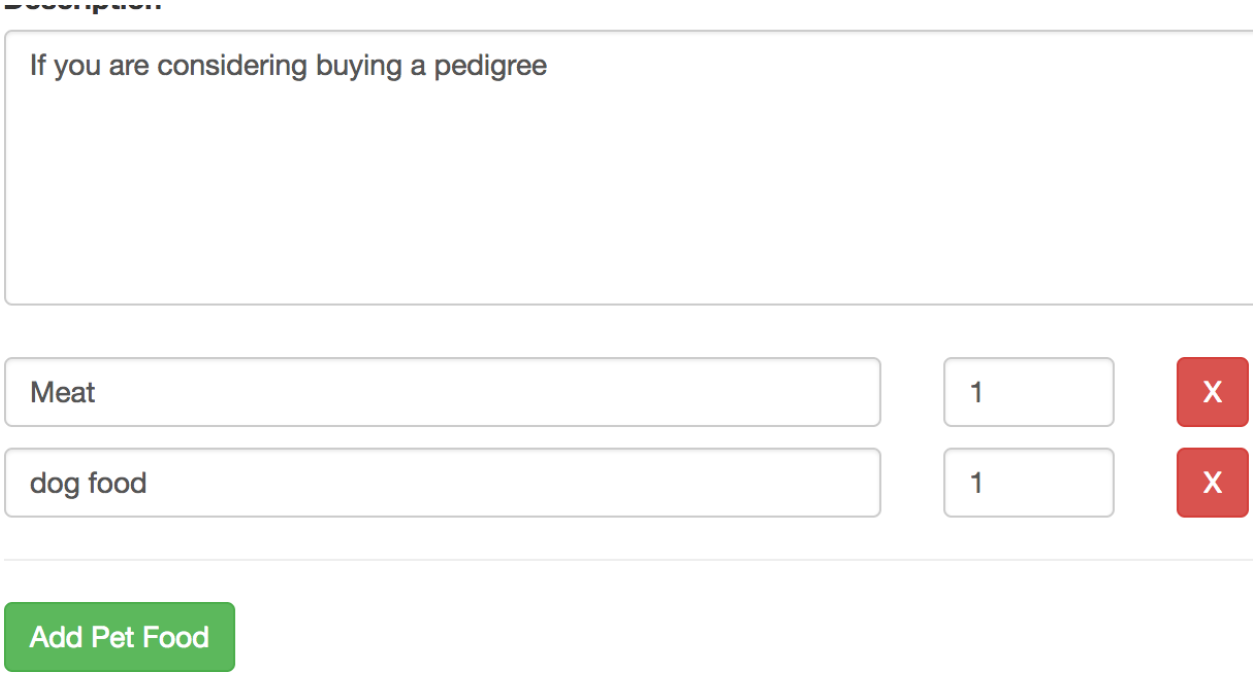
6.Project Summary:

The following are the screen shots of the website

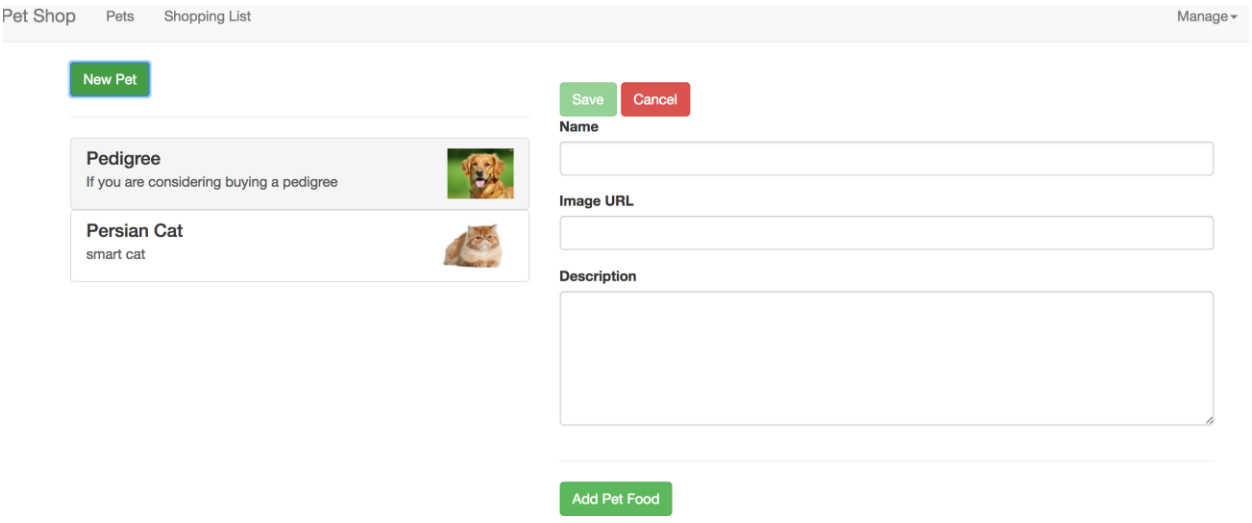
Pet Edit:

This allows you to edit pets and add food.

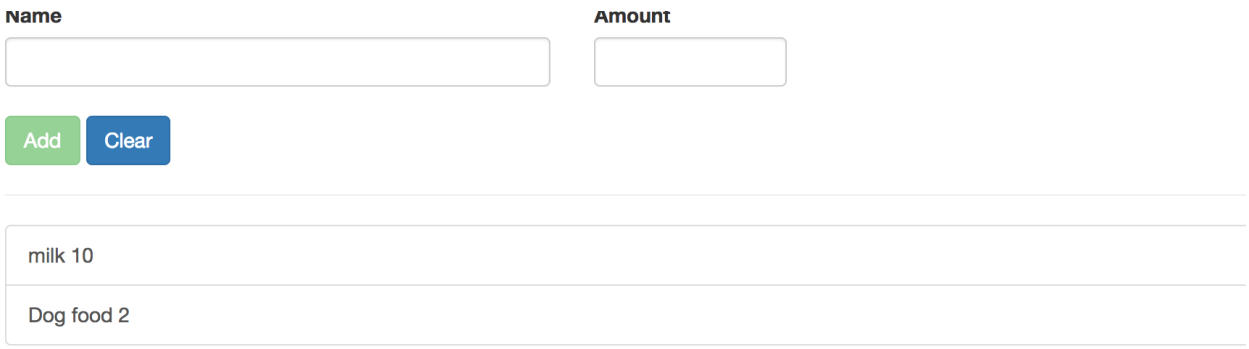




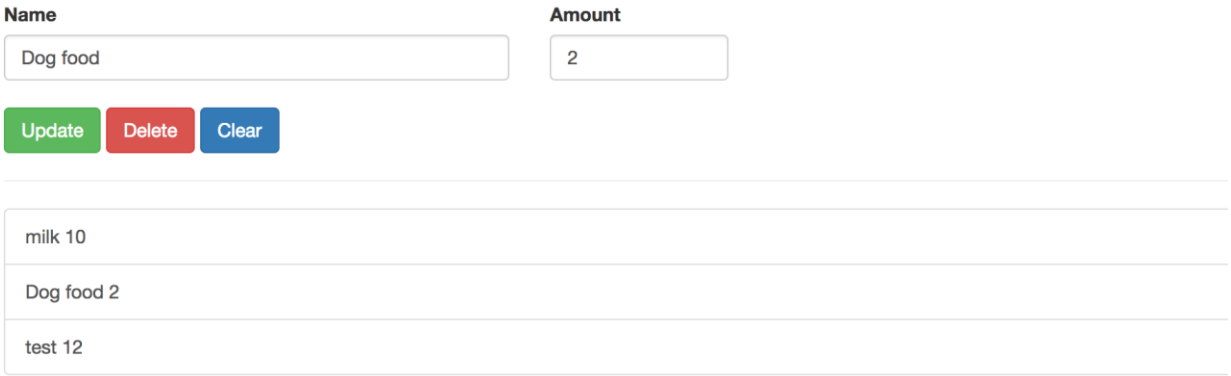
Pet Add New:
This allows you to add new pets.



Shopping list:



ShoppingList edit/delete:



You can launch the application through the npm install followed by ng serve command in the angular CLI.Then the application gets launched in the default angular port 4200.

7:Bibliography

References:

http://stackoverflow.com/

https://www.udemy.com/the-complete-guide-to-angular-2/learn/v4

https://angular.io/docs/ts/latest/cookbook/ajs-quick-reference.html