# ICP6

Keenan Flynn – <a href="mailto:kpfxn8@umsystem.edu">kpfxn8@umsystem.edu</a> – https://github.com/kfly2fly/Web-Mobile-Spring-2022 Jasmine Naraine- <a href="mailto:jnytc@umstyem.edu">jnytc@umstyem.edu</a> - https://github.com/JNaraine/Web

This ICP focuses on the REST APIs. RESTful APIs grant applications access to resources across the internet. These APIs allow us to use algorithms from 3rd parties such as FourSquare and Edamam in our own projects. REST uses HTTP protocols. These protocols include GET, PUT, POST, and DELETE. In this ICP, we used the get() method provided with Angular. This method retrieves the current state of the FourSquare and Edamam database and returns that information in the form of a JSON object. We can then query this object to get useful information.

To query the FourSquare and Edamam APIs, we needed to pass some values. Each API has a standard way of passing values in the URL, which we looked up on the developer part of each respective website. There were some challenges related to getting the desired information. For example, the FourSquare api was recently updated and needed HttpHeaders. We had to figure out how to pass these headers in the get() request so that we could get the right response from the website.

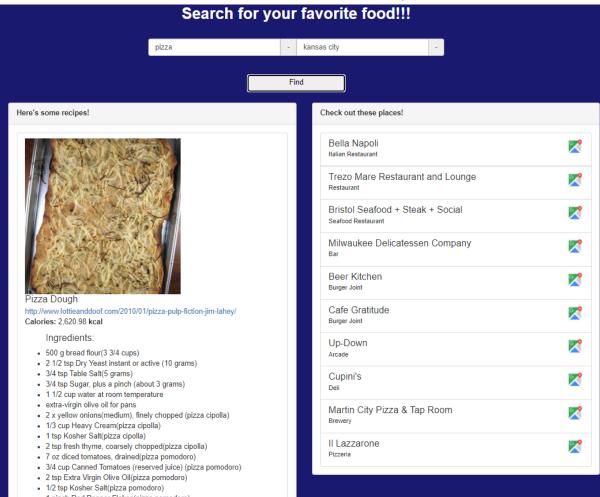
Another aspect of web design is routing. Routing allows you to navigate to different pages based on the url. You can also set up aliases so that one url will redirect you to another url or component. We have basic routing set up in this ICP so that when the user is directed to the recipe-search module with the url <a href="http://localhost:4200/search-recipe">http://localhost:4200/search-recipe</a> and the url localhost:4200/.

The UI consists of 2 form fields where the user can enter a food and a place or city. This information is passed to the APIs and then displayed so that the user can select between different recipes or different restaurants to get that food.

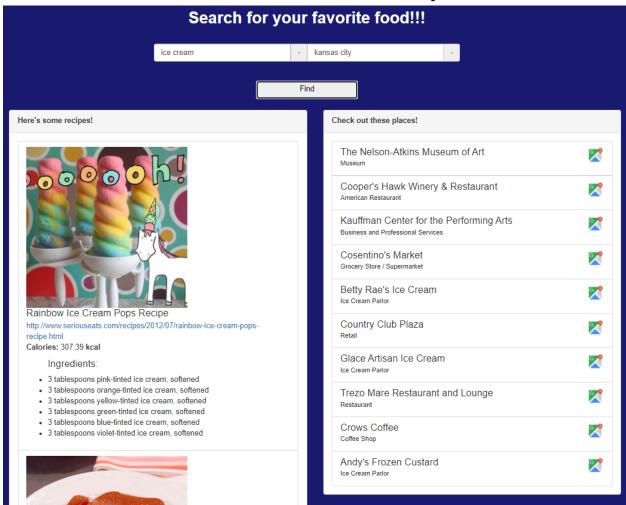
#### **Default Screen**



# Search for Pizza in Kansas City



# Search for Ice Cream in Kansas City



#### environment.ts

```
export const environment = {{
    production: false,
    //Edaman application ID and KEy
    APP_ID: "e2f0f7f3",
    APP_KEY:"42de8fcd337d2185069fdacc8c6e9adf"
};
```

## search-recipe-component.ts

```
getVenues() {
  this.recipeValue = this.recipes.nativeElement.value;
  this.placeValue = this.places.nativeElement.value;
  if (this.recipeValue !== null) {
   this.searchService.getRecipes(this.recipeValue).subscribe(data => {
   this.recipeList = data.map(e => e.recipe)
if (this.placeValue !== null && this.placeValue !== '' && this.recipeValue !== null && this.recipeValue !== '') {
   const headers= new HttpHeaders()
     .set('Accept', 'application/json')
     .set('Authorization', 'fsq3lNgWyv9sczLrigyn5/WI4acjplEA0JXYzR7DWFdEXmA=');
   var url = 'https://api.foursquare.com/v3/places/search?exclude_all_chains=true&sort=RATING&limit=50&query=' + this.recipeVa
   this._http.get(url, { 'headers': headers })
     .subscribe((data:any) => {
     this.isLoading=false;
      this.venueList = Object.keys(data.results).map(function (k){
       var i = data.results[k];
       console.log(i);
       return {name: i.name, location: i.location, category: i.categories[0].name}
      //Log the results to the console for testing
     console.log(this.venueList);
```

# search-recipe.service.ts

```
app / search-recipe > is search-recipes.etvice.is / is searchRecipeservice
import { Injectable } from '@angular/common/http';
import { HttpClient} from '@angular/common/http';
import { map } from 'rxjs/operators';
import { Observable } from 'rxjs';

@Injectable({
    providedIn: 'root'
})
export class SearchRecipeService {

constructor(private http:HttpClient) { }
//api url with the app id and app key to search for any recipe

getRecipes(recipeName:String): Observable<any> {
    const EDAMAM_API_URL = `https://api.edamam.com/search?q=${recipeName}&app_id=${env.APP_ID}&app_key=${env.APP_KEY}`;
    return this.http.get(EDAMAM_API_URL).pipe(map(e => e['hits']))
}
```

## search-recipe.html

```
iv class="container">
<div class="row">
 <div class="col-lg-6">
   <div class="panel panel-default col-6">
     <div class="panel-heading"><b>Here's some recipes!</b></div>
     <div class="panel-body">
       <div *ngFor="let recipe of recipeList" class="list-group-item clearfix">
        <div class="detail">
          <img src="{{recipe.image}}" class="image mt" />
          <div class="header">
          <h4 class="list-group-item-heading mt">{{ recipe.label }}</h4>
          <a href="{{recipe.url}}" class="list-group-item-text">{{ recipe.url }}</a>
          <strong> Calories: </strong> {{recipe.calories | number: '1.2-2' }}<strong> kcal</strong</pre>
         <!--Recipe Ingredients-->
          <h4>Ingredients: </h4>
          {{ingredientLine}}
        <span class="pull-right">
 <div class="col-lg-6">
   <div class="panel panel-default col-6">
     <div class="panel-heading"><b>Check out these places!</b></div>
     <div class="panel-body">
       <div *ngFor="let venue of venueList" class="list-group-item clearfix">
        <div class="pull-left">
          <h4 class="list-group-item-heading">{{ venue.name }}</h4>
          <h6 class="list-group-item-heading">{{ venue.category}} </h6>
         <a href="http://maps.google.com/maps?saddr={{currentLat}},{{currentLong}}</pre>
         &daddr={{venue.location.address}}, {{venue.location.locality}}, {{venue.location.region}}">
           <img src="../../assets/google.png" style="width: 30px; height: 30px;"></a>
```