

# ICP6 Assignment

**Name :** Vari ManiRaj

**ID :** 16321350

**GitHub Source Code :** [https://github.com/manivari1/web\\_mobile/tree/main/icp06/Source](https://github.com/manivari1/web_mobile/tree/main/icp06/Source)

## **Goal of Assignment :**

The main aspect of the assignment is to learn and understand Angular (routers, services, HTTP, and RESTful APIs). And also Wikipedia viewer use case and Recipe search with location finder application task.

Need to Create an application in Angular which displays nearby restaurants (Using Foursquare API) and to create an application in Angular which displays recipes (Hint: Use EDAMAM API)

**Challenges:** While running angular scripts, were getting many errors. But it was corrected by trial and errors.

## **Tasks :**

### **Foursquare API**

Create an application in Angular which displays nearby restaurants (Hint: Use Foursquare API)

### **EDAMAM API**

Create an application in Angular which displays recipes (Hint: Use EDAMAM API)

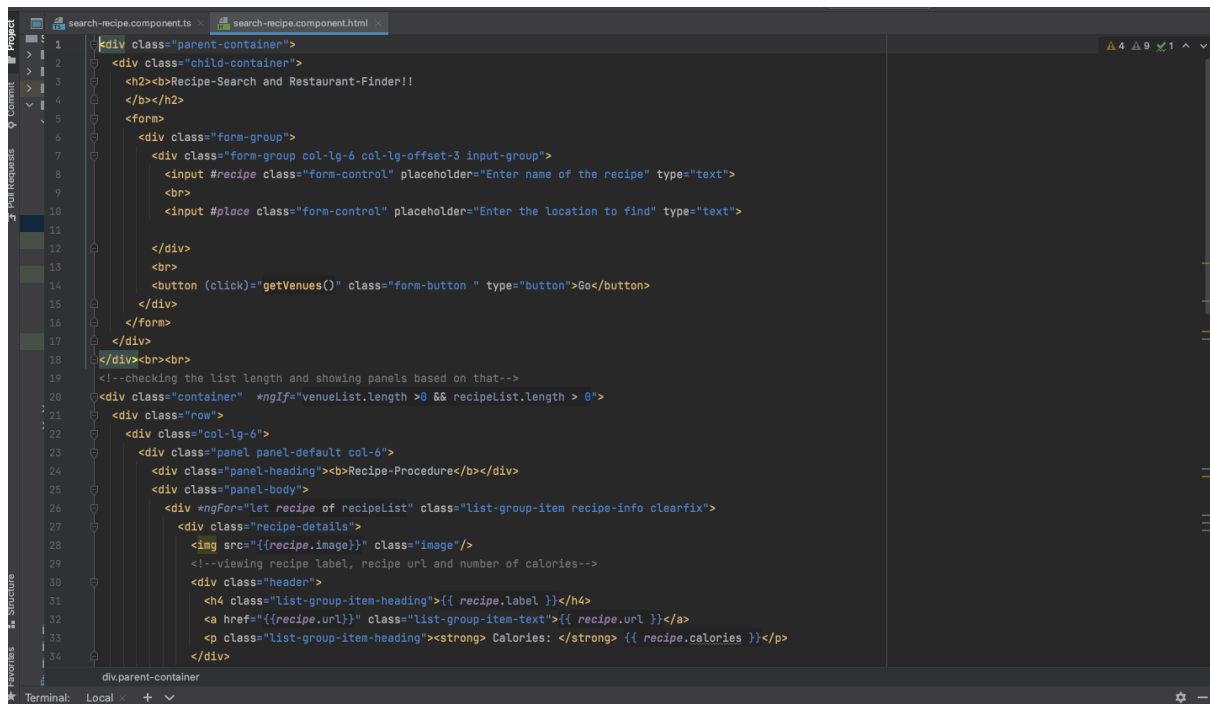
Can also, combine both the APIs in a single application.

## **Source Code :**

- 1) Firstly, created accounts in both Foursquare API and EDAMAM API.
- 2) Then client ID and client secret ID for foursquare API and Application ID and Application Keys for EDAMAM API are copied from each API's.
- 2) Foursquare API is used for getting location of restaurants near to the present location for the specified search recipe.
- 3) EDAMAM API recipes are integrated according to customer specified dish into the application
- 4) Need to paste keys and ID's from the API's into the environment.ts file. Import this file into the script file which lets to know the behaviour of application.
- 5) Using http called and got the response
- 6) Parsed JSON and appended to HTML using angular directives

## **HTML CODE:**

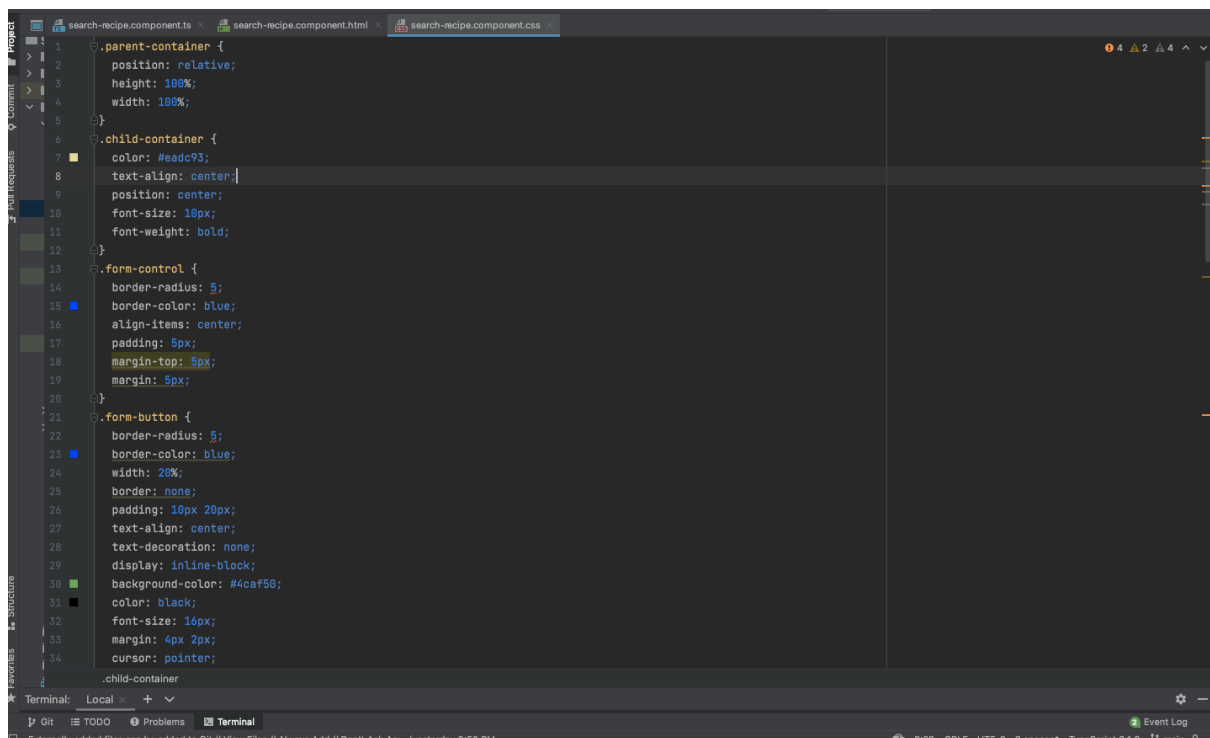
This is the screenshot of the HTML file that represents the search-recipe bar that shows us the two options that are recipe and the location as we have used the Foursquare and Edamam for those search.



```
1 <div class="parent-container">
2   <div class="child-container">
3     <h2><b>Recipe-Search and Restaurant-Finder!!</b></h2>
4   </div>
5   <form>
6     <div class="form-group">
7       <div class="form-group col-lg-6 col-lg-offset-3 input-group">
8         <input #recipe class="form-control" placeholder="Enter name of the recipe" type="text">
9         <br>
10        <input #place class="form-control" placeholder="Enter the location to find" type="text">
11      </div>
12      <br>
13      <button (click)="getVenues()" class="form-button" type="button">Go</button>
14    </div>
15  </form>
16 </div>
17 </div>
18 </div><br><br>
19 <!--checking the list length and showing panels based on that-->
20 <div class="container" *ngIf="venueList.length > 0 && recipeList.length > 0">
21   <div class="row">
22     <div class="col-lg-6">
23       <div class="panel panel-default col-6">
24         <div class="panel-heading"><b>Recipe-Procedure</b></div>
25         <div class="panel-body">
26           <div *ngFor="let recipe of recipeList" class="list-group-item recipe-info clearfix">
27             <div class="recipe-details">
28               
29               <!--viewing recipe label, recipe url and number of calories-->
30               <div class="header">
31                 <h4 class="list-group-item-heading">{{ recipe.label }}</h4>
32                 <a href="{{recipe.url}}" class="list-group-item-text">{{ recipe.url }}</a>
33                 <p class="list-group-item-heading"><strong> Calories: </strong> {{ recipe.calories }}</p>
34             </div>
35           </div>
36         </div>
37       </div>
38     </div>
39   </div>
40 </div>
41 </div>
```

## CSS code:

This below is the screenshot of the CSS file that we have used to style and represent my web page in a beautiful outlook which attracts the users of my web page.



```
1 .parent-container {
2   position: relative;
3   height: 100%;
4   width: 100%;
5 }
6 .child-container {
7   color: #eadc93;
8   text-align: center;
9   position: center;
10  font-size: 18px;
11  font-weight: bold;
12 }
13 .form-control {
14   border-radius: 5;
15   border-color: blue;
16   align-items: center;
17   padding: 5px;
18   margin-top: 5px;
19   margin: 5px;
20 }
21 .form-button {
22   border-radius: 5;
23   border-color: blue;
24   width: 20%;
25   border: none;
26   padding: 10px 20px;
27   text-align: center;
28   text-decoration: none;
29   display: inline-block;
30   background-color: #4caf50;
31   color: black;
32   font-size: 16px;
33   margin: 4px 2px;
34   cursor: pointer;
35 }
36 .child-container
```

## JavaScript :

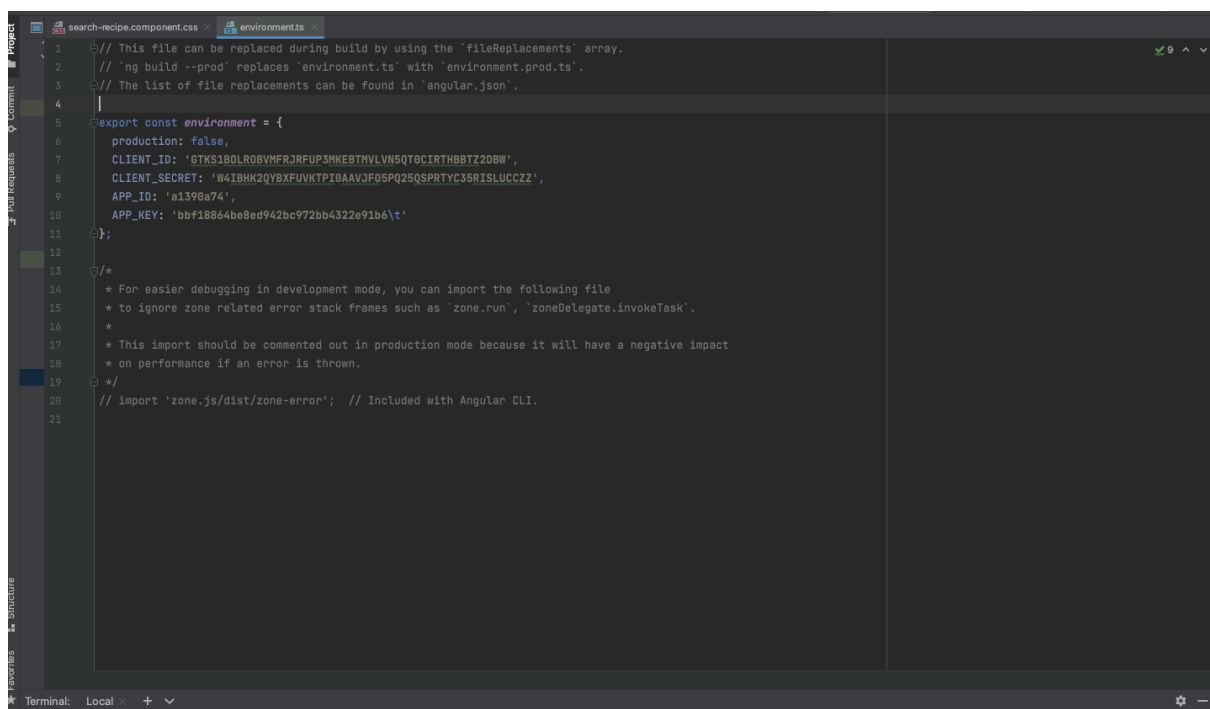
These are the javascript files that are coded for finding recipes and the location with the help of the Edamam and Foursquare. These are the files that are worked on the back end they will display the output to the user as their request.

```
environment.ts search-recipe.service.ts app.component.html app.component.css
1 import { Injectable } from '@angular/core';
2 import { Environment as env } from '../environments/environment';
3 import { HttpClient } from '@angular/common/http';
4 import { map } from 'rxjs/operators';
5 import { Observable } from 'rxjs';
6
7 @Injectable({
8   providedIn: 'root'
9 })
10 export class SearchRecipeService {
11   //giving api url with our client id and client secret keys and adding endpoints
12   public FOURSQUARE_API_URL = 'https://api.foursquare.com/v2/venues/explore?client_id=${env.CLIENT_ID}&client_secret=${env.CLIENT_SECRET}&v=20180323&near=';
13
14   constructor(private http: HttpClient) {}
15
16   getVenueByItem(placeName: String, recipeName: String): Observable<any[]> {
17     const url = this.FOURSQUARE_API_URL + placeName + '&query=${recipeName}';
18     return this.http.get(url).pipe(map( project: venues => {
19       return venues['response'].groups[0].items;
20     }));
21   }
22
23   getRecipes(recipeName:String): Observable<any> {
24     //giving api url with our app id and app key along with giving recipe name and adding endpoints and getting data from that
25     const EDAMAM_API_URL = 'https://api.edamam.com/search?q=${recipeName}&app_id=${env.APP_ID}&app_key=${env.APP_KEY}';
26     return this.http.get(EDAMAM_API_URL).pipe(map( project: e => e['hits']));
27   }
28 }
29
Terminal: Local +
Git TODO Problems Terminal
Externally added files can be added to Git // View Files // Always Add // Don't Ask An... (yesterday 9:50 PM)
11 1F UTE-B 2 spaces TypeScript 3.1.6 main
```

```
search-recipe.component.ts
1 import { Component, ElementRef, OnInit, ViewChild } from '@angular/core';
2 import { HttpClient } from '@angular/common/http';
3 import { SearchRecipeService } from './search-recipe.service';
4
5 @Component({
6   selector: 'app-search-recipe',
7   templateUrl: './search-recipe.component.html',
8   styleUrls: ['./search-recipe.component.css']
9 })
10 export class SearchRecipeComponent implements OnInit {
11   @ViewChild('recipe') recipes: ElementRef;
12   @ViewChild('place') places: ElementRef;
13   recipeValue: any;
14   placeValue: any;
15   venueList = [];
16   recipeList = [];
17
18   currentLat: any;
19   currentLong: any;
20   geolocationPosition: any;
21
22   constructor(private searchService: SearchRecipeService) {}
23
24
25
26   ngOnInit() {
27     window.navigator.geolocation.getCurrentPosition(
28       successCallback: position => {
29         this.geolocationPosition = position;
30         this.currentLat = position.coords.latitude;
31         this.currentLong = position.coords.longitude;
32       });
33   }
34
35   SearchRecipeComponent
Terminal: Local +
Git TODO Problems Terminal
Externally added files can be added to Git // View Files // Always Add // Don't Ask An... (yesterday 9:50 PM)
10:10 1F UTE-B 2 spaces TypeScript 3.1.6 main
```

## Environment.ts:

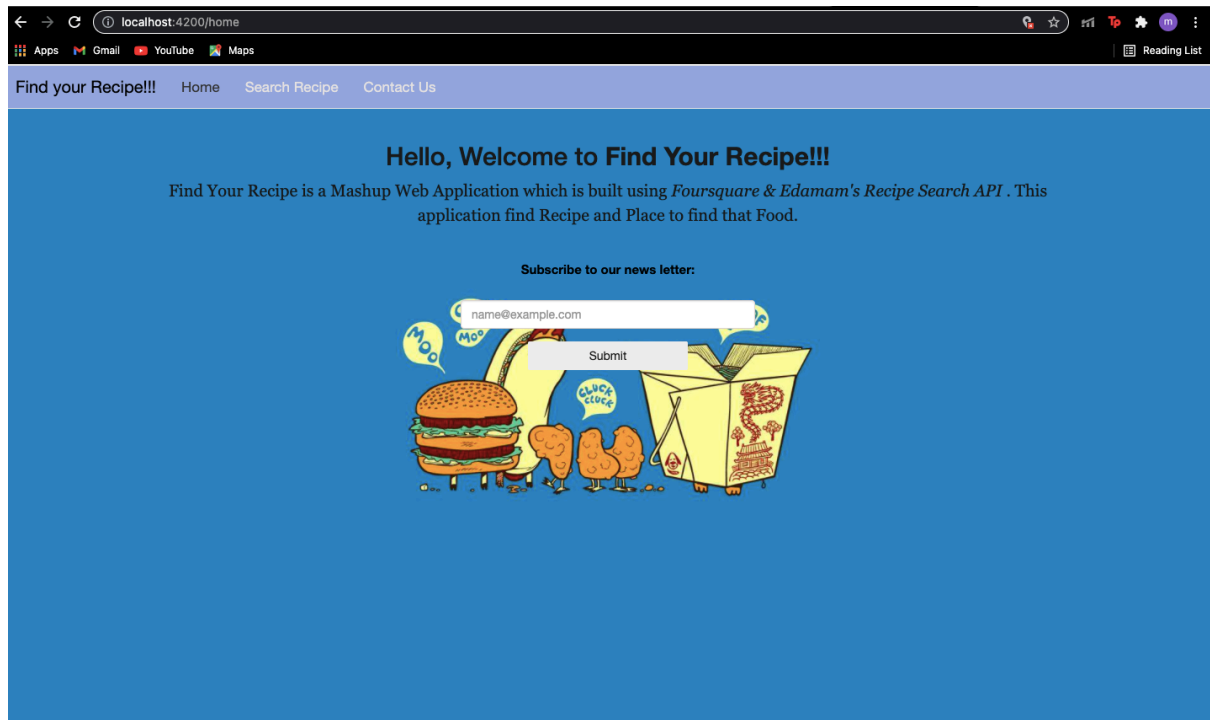
This below is the screenshot of the environment.ts file that maintains the whole climate in the code and we mention out CLINT\_ID, CLIENT\_SECRET, APP\_ID, APP\_KEY from the Edamam and foursquare as they are used to fetch the data.



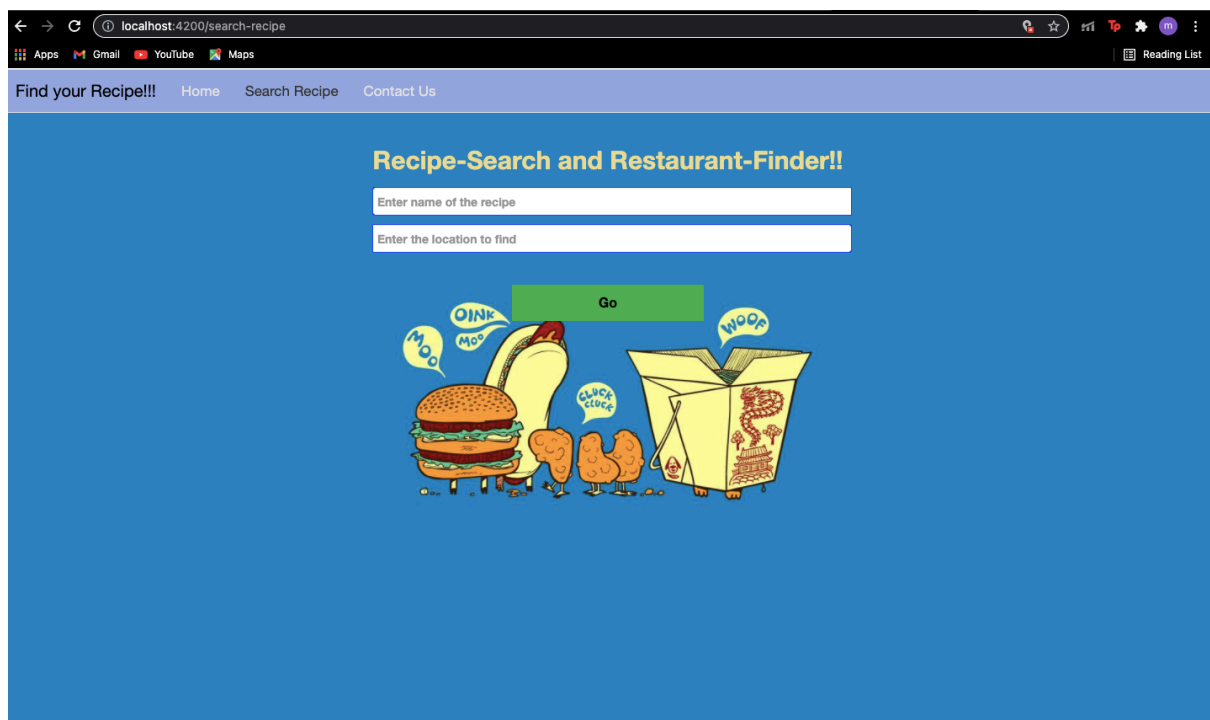
```
1  // This file can be replaced during build by using the 'fileReplacements' array.
2  // 'ng build --prod' replaces 'environment.ts' with 'environment.prod.ts'.
3  // The list of file replacements can be found in 'angular.json'.
4
5  export const environment = {
6    production: false,
7    CLIENT_ID: 'GTKS1B0LROBVMFRJRFUP3MKEBTHVLVN5QTGCIETHB8TZ2DBW',
8    CLIENT_SECRET: 'W4IBHK2QYBXFUVKTP10AAVJF05PQ25QSPRTYC35RISLUCCZZ',
9    APP_ID: 'a1390a74',
10   APP_KEY: 'bbf18864be8ed942bc972bb4322e91b6\t'
11 };
12
13 /**
14  * For easier debugging in development mode, you can import the following file
15  * to ignore zone related error stack frames such as 'zone.run', 'zoneDelegate.invokeTask'.
16  *
17  * This import should be commented out in production mode because it will have a negative impact
18  * on performance if an error is thrown.
19  */
20 // import 'zone.js/dist/zone-error'; // Included with Angular CLI.
21
```

## OUTPUT:

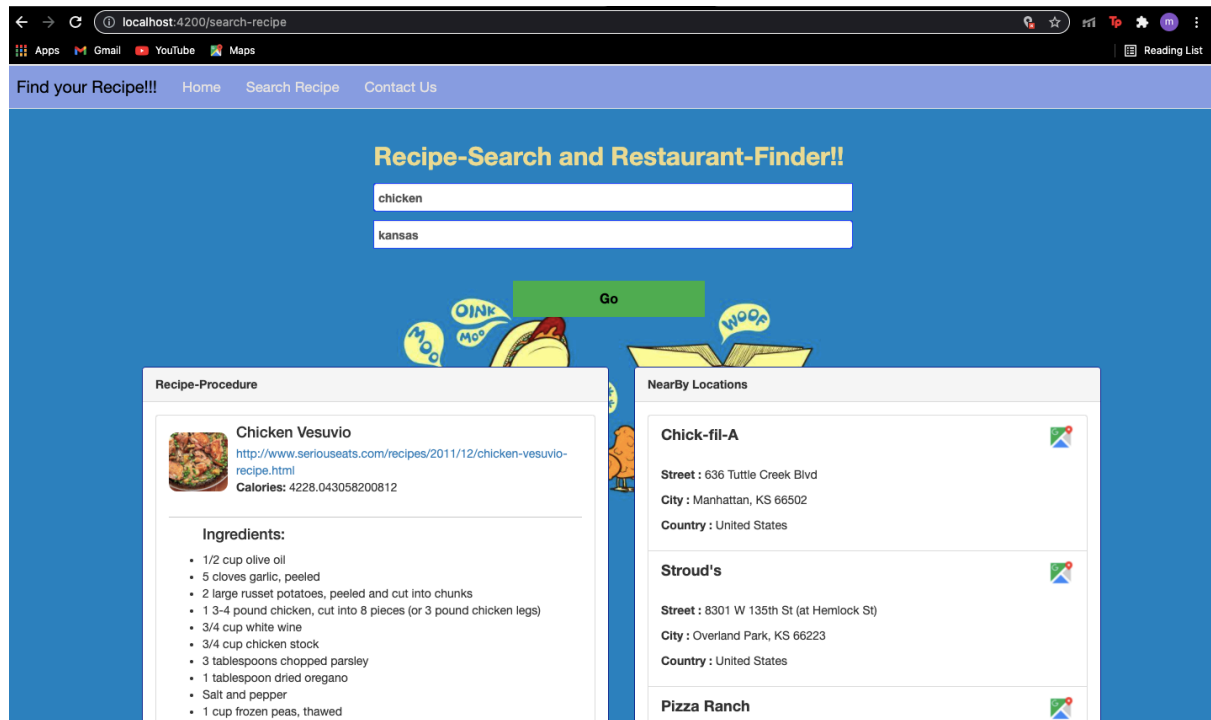
1. Below screen short is the intial outlook of my output which shows the FIND THE RECIPE, HOME, SEARCH RECIPE, CONTACT US in the navbar. And below that we can find the welcome para with some info, their we can also find the a box with submit button for entering the user email, so that they can get notifications from out web page.



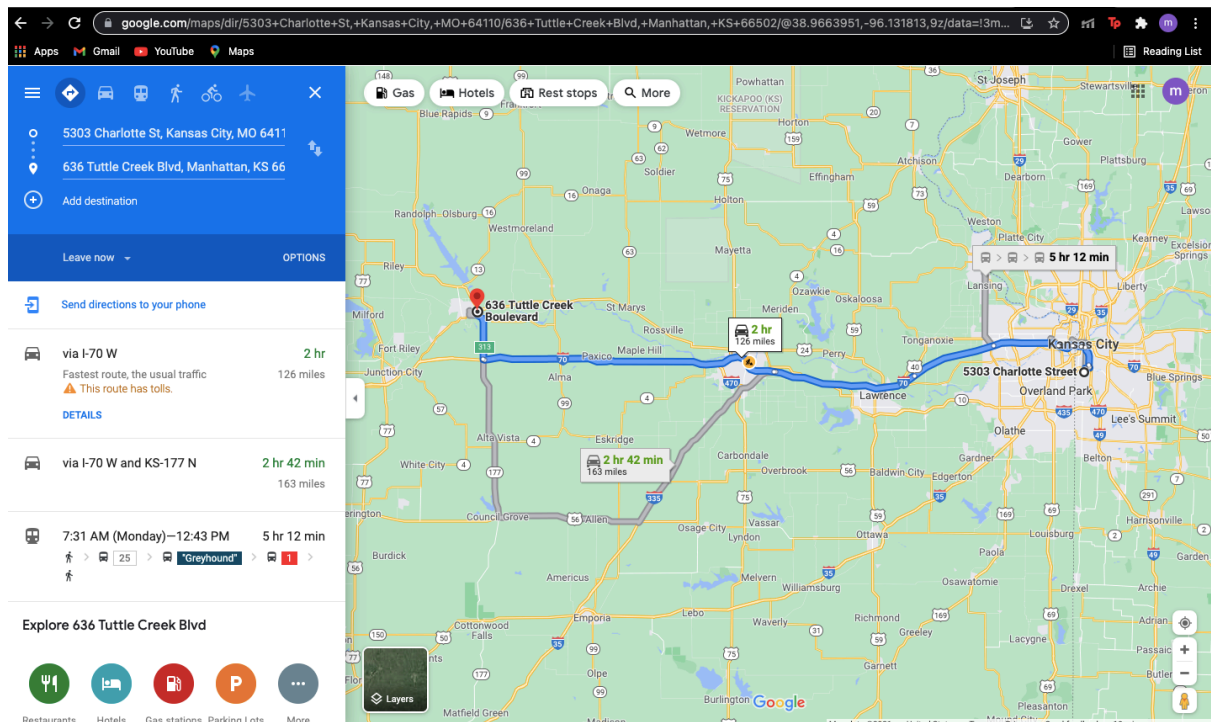
2. In the below output we have selected the “search recipe” from the navbar then we can see the RECIPE-SEARCH AND RESTAURANT-FINDER pop’s up their we can see two boxes for entering the recipe and location and then we need to press GO button at below.



3. When we enter the recipe and location as we can see that I have entered the recipe as chicken and the location as the Kansas for test case. Then when we submit the GO button we will find all the RECIPE-PROCEDURE and the NEAR-BY-LOCATIONS at the bottom side by side as shown below. And we can also find the small map icon at the side of the locations that are listed below. When we click on that map icon the 4 starts



4. When we click on that map icon then we are directly directed to the google maps and we can find the location of that particular location and also we can find the route from our location as we can see below.



- Then the final part that was remained in the navbar is CONTACT-US, when we click on the contact us then we will see the three boxes that are for as follows NAME, EMAIL, THEIR QUERY...

And by pressing go that details are saved...

