

## ICP-6

**Name:** Yaswanth Paruchuri

**Email:** ypfm7@umsystem.edu

**GitHub Repo link:**

<https://github.com/Yaswanthypfm7/WebDevCourse/tree/main/WebPart/ICP6>

**Name:** Tej Deep Parvatha Reddy

**Email:** tpgkd@umsystem.edu

**GitHub Repo Link:**

<https://github.com/TejdeepP/WebProgrammingSpring2022/tree/main/WebPart/ICP6>

## Introduction

Angular routing is the key factor of any Web application. An application recommends can be based on whether it has good routing capabilities or not. It mostly helps user in single and multipage navigations. Angular has good routing capabilities that can accommodate simple and complex web applications. Routing can be defined as the navigating via a view. Angular has a specific routing technique called Routing Module that helps applications to navigate in a page.

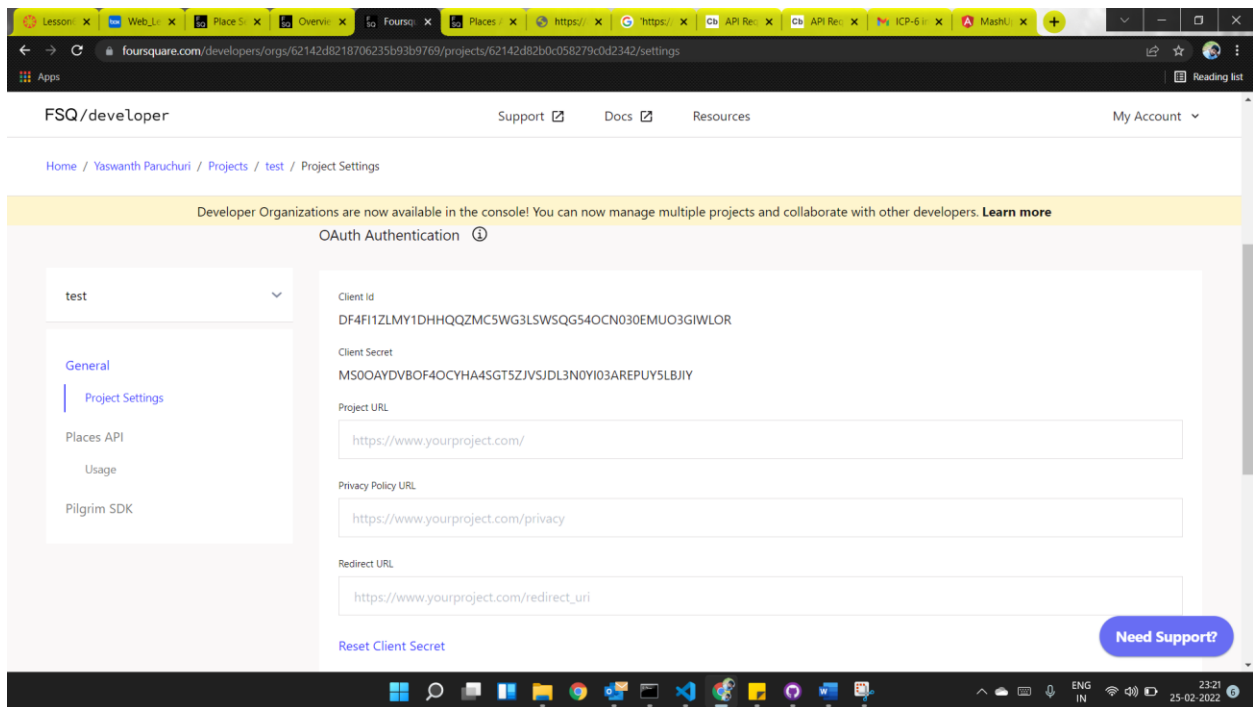
Dependency Injection in angular can be more powerful in many ways. A dependency injection can be explained as if there is a client and the service the feature of a client need to forward to a service, this can be done via dependency injection that helps to transfer and help the client to make useful to the provided service to the service class.

Rest API , Angular http requests – Rest API acts as a bridge between the client and the webmodule. It uses the Hyper text markup language protocol and makes the request to the web. It takes the helps of resources and transfer or gets the data from the client and server like GET , POST etc..

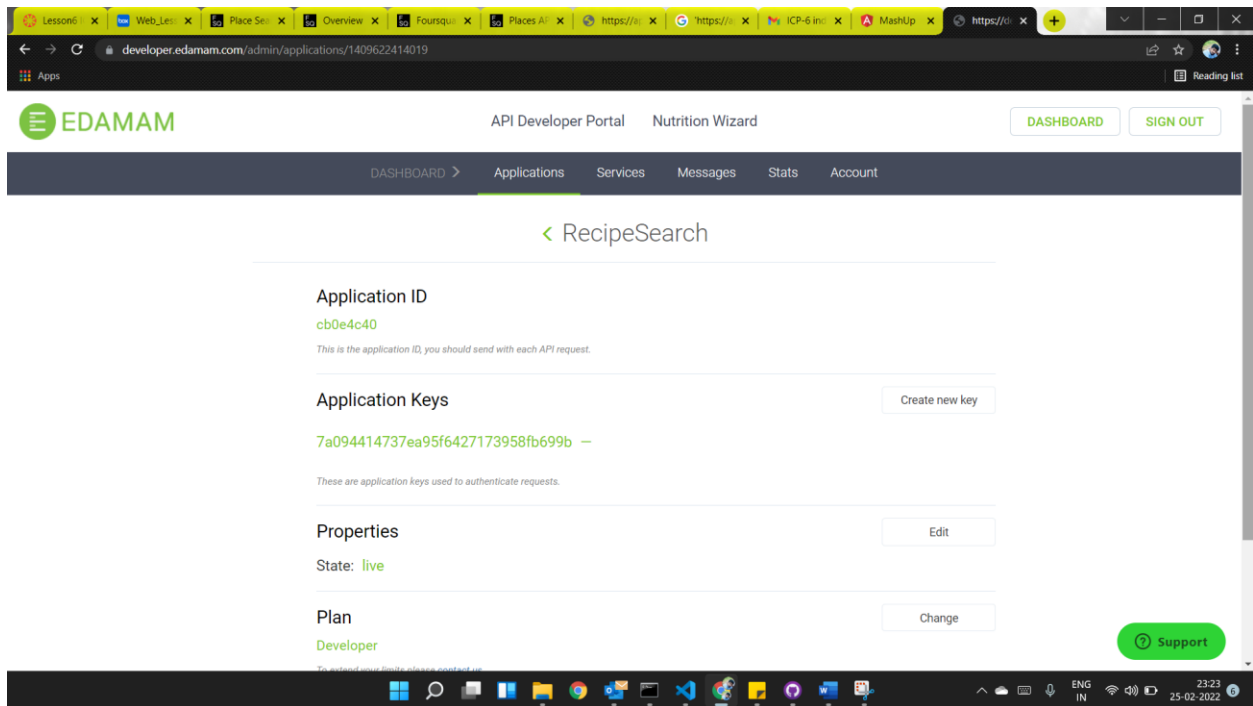
## TASKS

In this ICP we have worked on the Angular routing components , Rest API , http client requests and Angular dependency injection. We have created a web application for finding the recipes for a geo-location with the filtering of dishes.

Below are the client ID and client secret for four square API

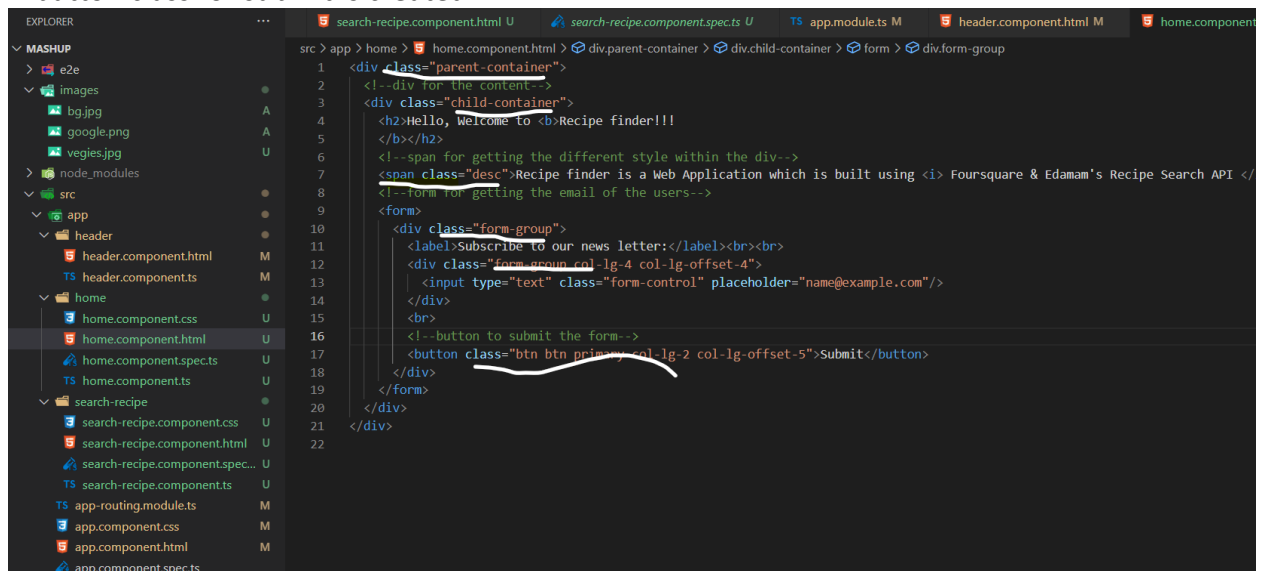


Below are the API details for EDAMAM API



The HTML for the home page : Here we have created a class named desc for the recipe finder webpage. A parent container and child container class are created.

- A button class for submit is created.



- CSS for Home Page: CSS for parent container with height and width.

- Color green is given for child container and aligned to center.
- The text is given in Georgia format and color is black.
- Submit button is given with Green background color and on click given white color.

```

src > app > home > home.component.css > button
1  .parent-container {
2      position: relative;
3      height: 100%;
4      width: 100%;
5  }
6
7  .child-container {
8      color: green;
9      text-align: center;
10     position: center;
11 }
12
13 .desc {
14     font-family: Georgia;
15     font-size: 17px;
16     color: black;
17 }
18
19 button {
20     background-color: green;
21     color: white;
22 }
23

```

Home Component script:

Here we have given the routing components like for the home component.html and home component.css are imported here with template and the style urls.

```

src > app > home > TS home.component.ts > HomeComponent
1  import { Component, OnInit } from '@angular/core';
2
3  @Component({
4      selector: 'app-home',
5      templateUrl: './home.component.html',
6      styleUrls: ['./home.component.css']
7  })
8  export class HomeComponent implements OnInit {
9
10     constructor() { }
11
12     ngOnInit() {
13     }
14
15 }
16

```

## Search Recipe:

Here we have given title as Search for your recipe and with inputs as receipe and place and a button. We have given a container and divided into left and right parts by using the “row” class. Left part is used for displaying the recipes list which ever the user selected and the right part is used to display the venues where the recipe is located near the place which the user has given in the input.

```
TS search-recipe.component.ts U X search-recipe.component.html U X search-recipe.component.spec.ts U TS app.module.ts M header.co
src > app > search-recipe > search-recipe.component.html > br
1 <div class="parent-container">
2   <div class="child-container">
3     <h2><b>Search for your recipe!!
4     </b></h2><br>
5     <!--creating an input for both the recipe and the place to find-->
6     <form>
7       <div class="form-group">
8         <div class="form-group col-lg-6 col-lg-offset-3 input-group">
9           <input #recipe class="form-control" placeholder="Enter Recipe to search" type="text">
10          <br> <br> <br>
11          <input #place class="form-control" placeholder="Enter the Place" type="text">
12        </div>
13        <br>
14        <!--Invoking the getvenues method on button click-->
15        <button (click)="getVenues()" class="btn btn primary col-lg-2 col-lg-offset-5 " type="button">Go</button>
16      </div>
17    </form>
18  </div>
19 </div>
20 </div><br><br>
21 <!--Creating new container to get the output to display-->
22 <div class="container">
23   <div class="row">
24     <!--Creating two divisions within the row to display in two parts-->
25     <div class="col-lg-6">
26       <!--display the left part-->
27       <div class="panel panel-default col-6">
28         <div class="panel-heading"><b>How to make</b></div>
29         <div class="panel-body">
30           <!--for loop to display all the values of the recepies-->
31           <div *ngFor="let recipe of recipelist" class="list-group-item clearfix">
32             <div class="pull-left">
```

In the left component we have given the recipe “Name”, “URL” and “ICON”.

In the right component we have given the “Place name”, “Address”, “Geo location with the current address and the destination address to find the path of the restaurant.

```

36         <a href="{{recipe.url}}" class="list-group-item-text">{{ recipe.url }}</a>
37     </div>
38     <!--display url to right corner using span-->
39     <span class="pull-right">
40         
41     </span>
42 </div>
43 </div>
44 </div>
45 </div>
46 </div>
47 </div>
48 </div>
49 </div>
50 <!--right side panel of the row-->
51 <div class="col-lg-6">
52     <div class="panel panel-default col-6">
53         <div class="panel-heading"><b>Where to get</b></div>
54         <div class="panel-body">
55             <!--for loop to display all the values of the venues-->
56             <div *ngFor="let venue of venueList" class="list-group-item clearfix">
57                 <div class="pull-left">
58                     <!--display name ,address-->
59                     <h4 class="list-group-item-heading">{{ venue.name }}</h4>
60                     <p>{{venue.location.address}},{{venue.location.country}}</p>
61                 </div>
62                 <!--display map as a link at right corner-->
63                 <span class="pull-right">
64                     <a href="http://maps.google.com/maps?saddr={{currentLat}},{{currentLong}}
65                     &sdaddr={{venue.location.address}},{{venue.location.address_extended}}">
66                         </a>
67                 </span>

```

Recipe Logic: In the class first we have initialized all the values

```

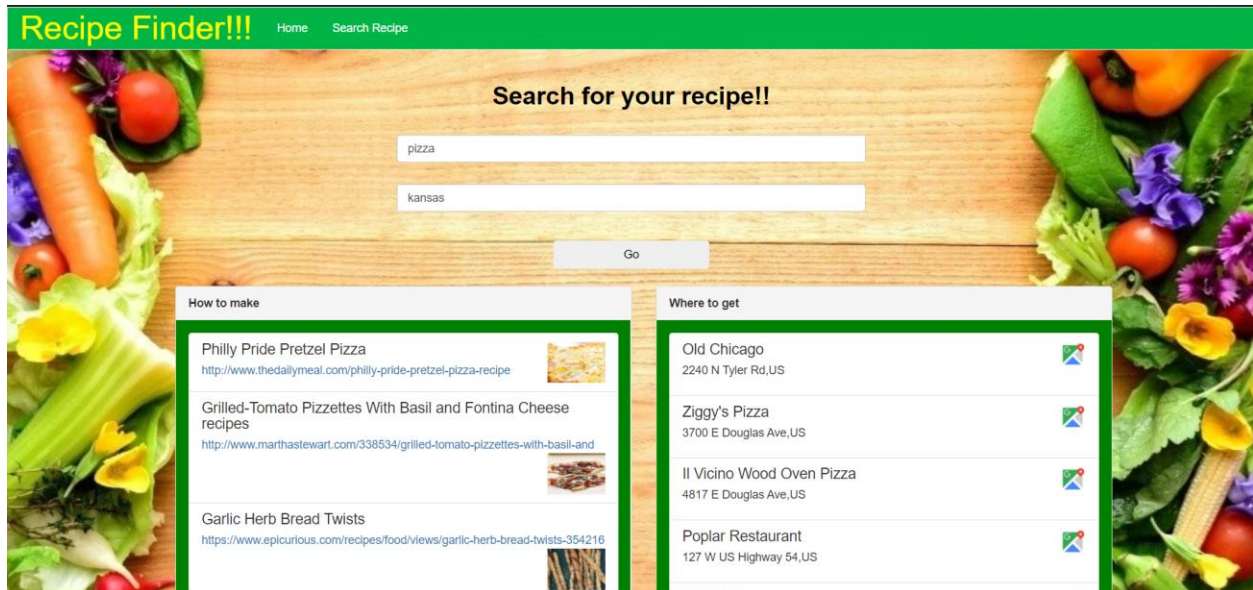
TS search-recipe.component.ts U X search-recipe.component.html U search-recipe.component.spec.ts U TS app.module.ts M
src > app > search-recipe > TS search-recipe.component.ts > SearchRecipeComponent > getVenues
1  import {Component, ElementRef, OnInit, ViewChild} from '@angular/core';
2  import {HttpClient, HttpHeaders} from '@angular/common/http';
3  import { getLocalePluralCase } from '@angular/common';
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     //getting the values from theinput and assigning to the recipies and places
12     @ViewChild('recipe') recipes: ElementRef;
13     @ViewChild('place') places: ElementRef;
14     //initializing all the values
15     recipeValue: any;
16     placeValue: any;
17     venueList = [];
18     recipeList = [];
19
20     currentLat: any;
21     currentLong: any;
22     geolocationPosition: any;
23
24     constructor(private _http: HttpClient) {
25     }
26
27     ngOnInit() {
28
29         window.navigator.geolocation.getCurrentPosition(
30             position => {
31                 this.geolocationPosition = position;
32                 this.currentLat = position.coords.latitude;

```

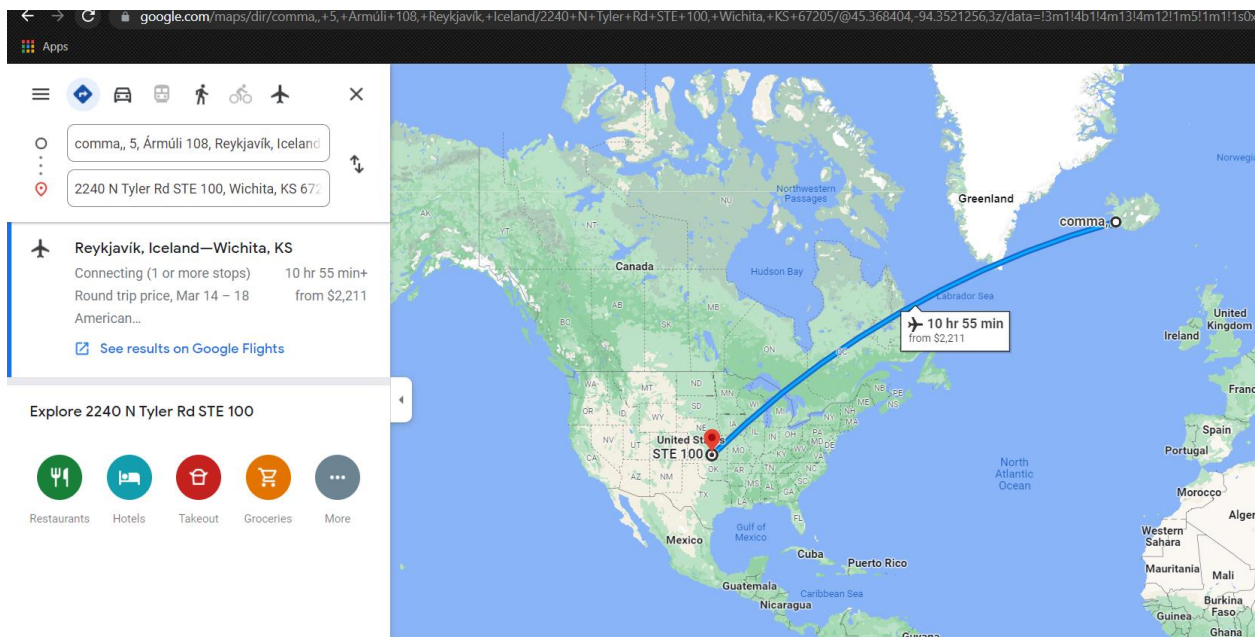
Here we have written the get venues method where we have taken the input from the user and assigned it to the local variables and also used the FOURSQUARE API and EDMAM API for getting the venues and the recipes respectively .We used our API keys , Client ID , Client Secret to get the values and assign them to the list and passed to the HTML to display them.

```
    });
  }
  //method to returnt the result
  getVenues() {
    //adding the values to the variables
    this.recipeValue = this.recipes.nativeElement.value;
    this.placeValue = this.places.nativeElement.value;
    // to ge thte recepies list
    if (this.recipeValue !== null) {
      //using the httpget method to get the recepies using the string concatenations for the api
      //used the edmam url to fet recepies added the api key and appid and few parameters
      this._http.get('https://api.edamam.com/api/recipes/v2?type=public&q=' + this.recipeValue +
        '&app_id=cb0e4c40&app_key=7a094414737ea95f6427173958fb699b&diet=balanced&cuisineType=American').subscribe((recipes: any) => {
        //assigning the list result to the recipelist
        this.recipeList = Object.keys(recipes.hits).map((index) => {
          const recipe = recipes.hits[index].recipe;
          return { name: recipe.label, icon: recipe.image, add: recipe.address, url: recipe.url }
        });
      });
    }
    // to ge thte places list
    if (this.placeValue != null && this.placeValue != '' && this.recipeValue != null && this.recipeValue != '') {
      //create the basic url with api,cliend id,secret and the recipe value from the input
      let configUrl='https://api.foursquare.com/v3/places/search?client_id=1M3QBJOFDJK30GGVNH20FTZKZQROT5MPTOTBVSAPFVGYDRLP&client
      // added the authorisation to ge the response form the url in the headers
      let headers= new HttpHeaders().set('Authorization', 'fsq3Ejh3P6V8A0qacog6Y//AztuqABFL3hAZzTesbnkVcR4=');
      const httpOptions={
        headers:headers
      }
      //performing the http get operation by giving the headers and the url also added the place value from the input
      this._http.get(configUrl+this.placeValue,httpOptions).subscribe((places: any) => {
        //assigning the resultant places list to eh venues list that is initialised earlier
        this.venueList=places.results
      });
    }
  }
}
```

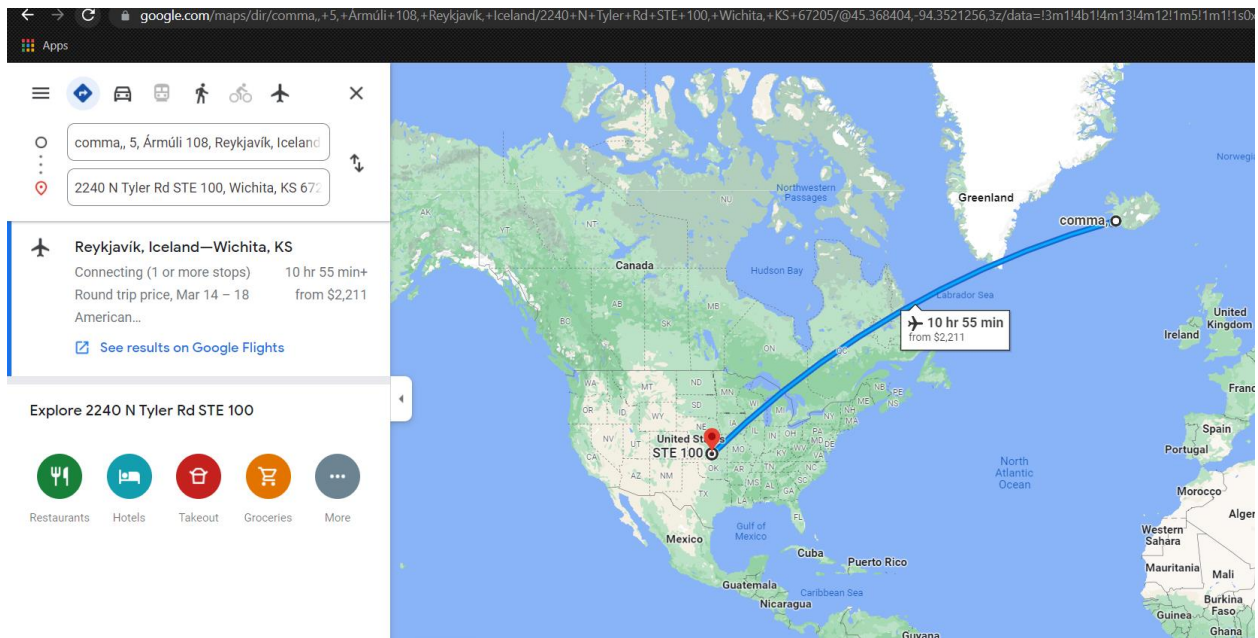
Output:



When clicked on the Google maps it is redirecting to the geo location on MAPS







When clicked on the recipe link it is redirecting to the recipe

