Anusha Nelluri – anhx2@umsystem.edu

GitHub Link - https://github.com/NelluriAnusha/Demo_Remote/tree/main/Webpart/ICP6

Achyuth Kumar Valeti – avgh3@umsystem.edu

GitHub Link - https://github.com/AchyuthValeti/Demo_Remote/tree/main/Webpart/ICP6

ICP6 (REST APIs with Angular)

Introduction:

In Angular, the HTTP requests are handled by the HttpClient API. HTTP protocol provides a high level of flexibility when communicating with a server. We use the HttpClient API to handle the data on the remote server by performing GET, POST, PUT, and Delete operations. In order to consume REST APIs, we must import and configure the HttpClient service in our Angular project.

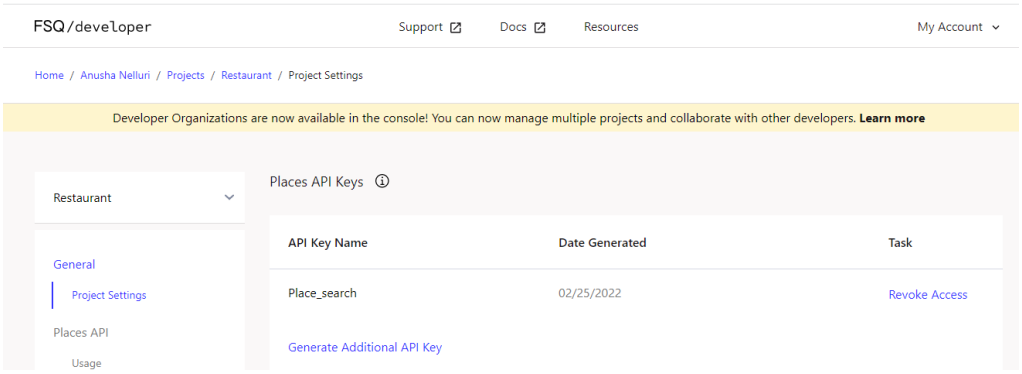
Tasks:

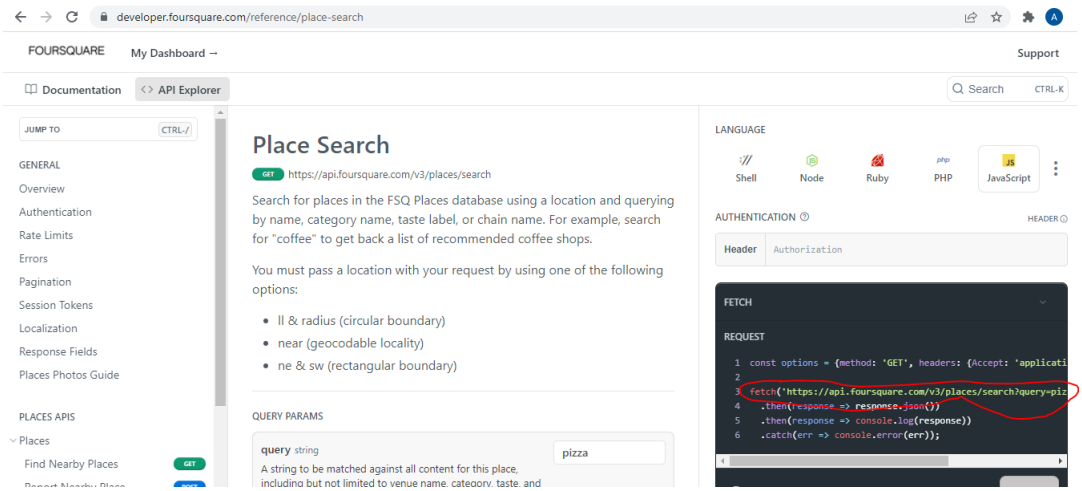
- In this ICP, we are going to develop the following two applications using routers, services, HTTP and RESTful APIs in Angular.
  1. Create an application in Angular which displays nearby restaurants
  2. Create an application in Angular which displays recipes
- Firstly, we have created applications on Four Square API & EDAMAM API to get the Recipe & location data.

Steps to create an account in above mentioned APIs:

1. **Created an account using: https://foursquare.com/**

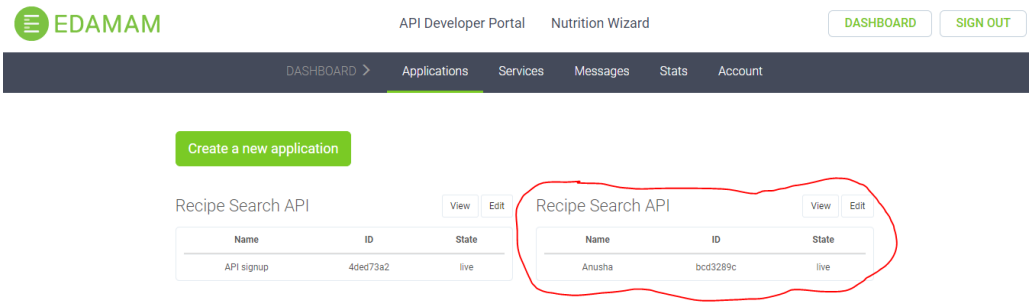
- After creating an account, logged into the account and created a new project.
- Once the project creation done, generated API Key for place search.
- When making API request call, we used this generated API Key value in HTTPHeaders field.

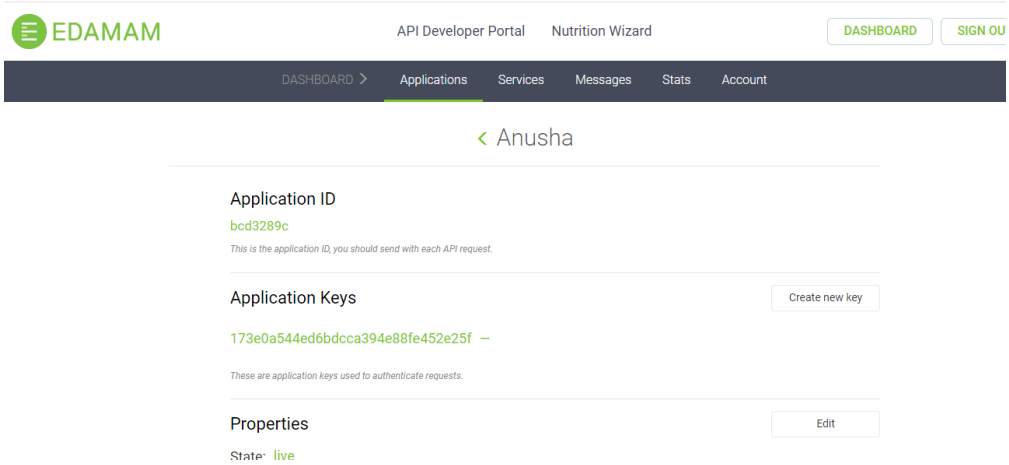




2. Created an account using https://developer.edamam.com/

- After creating an account, logged into the account and created a new application to get the recipe details.
- Once the application creation is done, APP ID & APP Keys will be generated.
- The APP ID & APP keys are used when making API request call to get the recipe data.





Implementation:

- I have implemented this application with the help of provided source code.
- Four components were created in this application as 'home', 'header' & 'search-recipe' and 'contact'.
- In '**home**' component, just created a user interface using html file and user can enter an email id for subscription.The content in this home page is static.
- In **'header'** component, used the 'nav-bar' which mainly contains header name or application name.

```
<a routerLink="/home" class="navbar-brand bg-primary" style="color: white">
  Find your Recipe!!!
</a>
```

- Used routing functionality and created 3 menu options and the page gets routed based on the user selection.
- The 'nav-bar' class has been used for 3 links, 1st is for 'home', 2nd is for 'search-recipe' and 3rd is for 'contact-us'.

```
<div class="collapse navbar-collapse tx">
  <ul class="nav navbar-nav">
    <li><a routerLink="/home">Home</a></li>
    <li><a routerLink="/search-recipe">Search Recipe</a></li>
    <li><a routerLink="/contact">Contact Us</a></li>
  </ul>
</div>
```

- In 'app.module.ts' file, all the created components information was included along with paths.
- Import the 'HttpClientModule' also in 'app.module.ts' file to use the HttpClient.
- In '**contact**' component, just created a user interface using html & css file and user can enter their contact details.
- Coming to the **search-recipe** component, this html file has two forms.
- The form contains two input fields which allows us to give 'Recipe name' and 'Place'.
- Once we enter these values, there is a submit button named it as **'Go'.**
- When we click on this button, **getVenues()** method will be called.

```
<div class="form-group col-lg-12 input-group">
  <input #recipe class="form-control" placeholder="Recipe name" type="text">
  <input #place class="form-control" placeholder="Place" type="text">
</div>
<button (click)="getVenues()" class="form-button " type="button">Go</button>
```

- The above mentioned two API calls (foursquare & EDAMAM) will be happening by calling the **getVenues()** method.

- As we have already registered for both the accounts, we need to mention the unique ID information in URLS.
- The httpClient, which is mentioned in the constructor is already initiated in load of the page.
- In 'ngOnInit()' method, used the geolocation function of getCurrentLocation to retrieve the current position's longitude and latitude coordinates.
- Using **nativeElement.value**, we can take the values of user entered Recipe name & Place name.
- We have two if conditions to check whether if the user entered recipe value or place value should not be null.
- If the values are not null, we are making the API calls.
- **The first API call is EDAMAM**, we are directly entering APP ID & APP Key in URL and passing 'recipe value'.
- After then, we call the http **get** method followed by subscribe callback which receives the response data in 'result' variable.
- Then, I have assigned the 'hits' in the result object to the 'output_recipeData' variable. And, pushing all these values to the 'recipeList', which is used to display the values in html.

```
if (this.recipeValue !== null) {
  // To get the recipe details by using get http
  this.recipeList = [];
  this._http.get(this.recipe_url + '&q=' + this.recipeValue ).subscribe( options: result => {
    const output_recipeData = result['hits'];
    console.log('Recipe Data', output_recipeData);
    output_recipeData.map(each => {
      this.recipeList.push({
        name: each.recipe.label,
        url: each.recipe.url,
        icon: each.recipe.image
      });
    });
    console.log(this.recipeList);
  });
}
```

- The second API call is FourSquare API, we are passing 'Recipe name' & 'Place value' and unique API Key with headers.
- After then, we call the http **get** method followed by subscribe callback which receives the response data in 'result' variable.
- Then, I have assigned the 'results' in the result object to the 'output_recipeData' variable. And, pushing all these values to the 'venueList', which is used to display the values in html.

```
// Code to get the Restaurant details by using get http
this.venueList = [];
const HTTP_headers = new HttpHeaders({
  Accept: "application/json",
  Authorization: "fsq3yqOCssKTh2VGw61HN6IiSO3JHd1RJrIuHBtEyFAULLs=",
});
this._http.get(this.location_url + this.recipeValue + '&near=' + this.placeValue + "&v=20220222",
  {headers: HTTP_headers}).subscribe( options: (result: any) => {
  const output_locationData = result['results'];
  console.log('Location:', output_locationData);
  output_locationData.map(each => {
    this.venueList.push({
      name : each.name,
      location : {
        formattedAddress : [
          each.location.formatted_address,
          each.location.locality]
      }
    });
  });
```

- Coming to the **search-recipe.component.html,** we were using ng directive '*ngFor' which is used to check the value in 'recipeList' or 'venueList' .
- If the 'recipeList' is not null, we print name of the recipe, URL of the recipe which has all the information (like preparation & ingredients list) related to the recipe.
- We are also displaying the image of the recipe.
- If the 'venueList' is not null, we print the name of the venue and formatted address.
- The anchor tag will help us to make a call to maps of google by passing formatted address as parameter. So that when we click on the link or image, user will be redirected to map of the venue Location.
- We were also displaying google map icon using tag.

CSS Changes:

- **navbar-dark:** This class changes the text's color to light. When choosing a dark background color, this is applied.

```
<nav class="navbar navbar-dark bg-primary">
```

- **.bg-primary:** This will set the color to the primary color.
- Background image of the app has been changed using below code in **app.component.css**

```
.image{
  height: 100vh;
  background-size:cover;
  width:100%;
  background-image:url('https://www.synopsys.com/blogs/software-security/wp-content/uploads/2019/07/open-source-ma-h
  background-position:50% 50%;
}
```

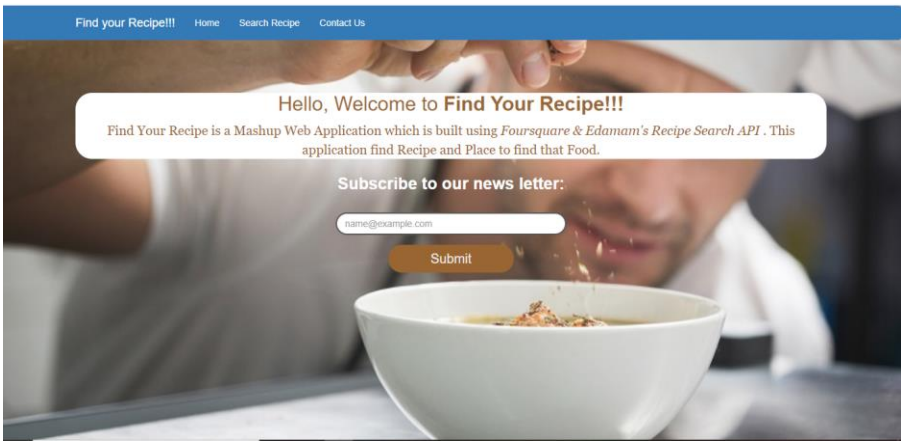
- Various form-groups in CSS are modified using the below code:

```css
.form-control {
  border-radius: 5px;
  align-items: center;
  padding: 5px;
  margin: 5px;
}

.form-group{
  border-radius: 50px;
  background-color: #2b669a;
}

.form-button {
  width: 18%;
  padding: 10px 20px;
  display: flex;
  justify-content: center;
  align-items: center;
  border-radius: 20px;
  background-color: #4caf50;
  color: black;
  font-size: 15px;
  margin: 5px 2px;
  cursor: pointer;
  margin-left: 40%;
}
```

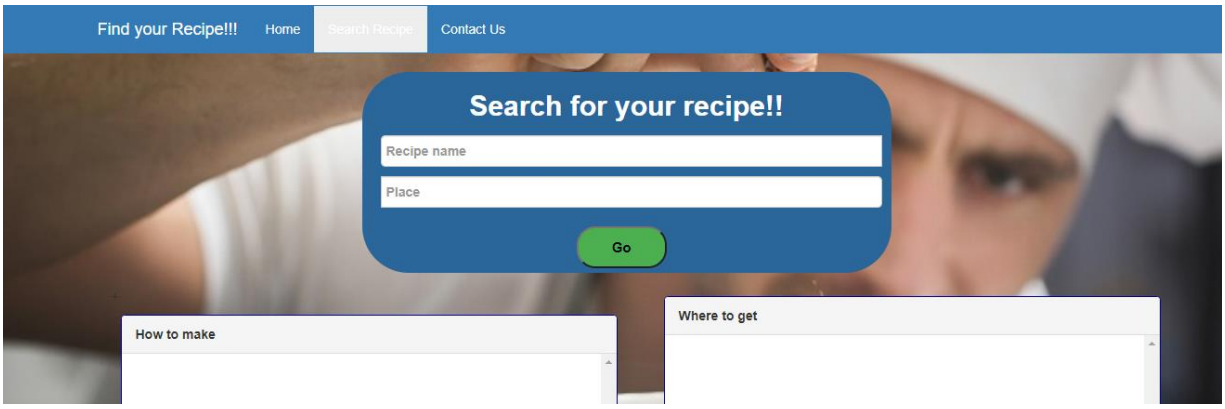
```css
.form-group {
  padding-top: 20px;
  position: center;
  text-align: center;
}

.form-group label {
  color: white;
  font-size: 25px;
}

.form-control {
  border: 2px solid ;
  border-radius: 20px;
}
.btn {
  border-radius: 20px;
  background-color: #996633;
  color: #eeeeee;
  font-size: 20px;
}
```

Output Pages:

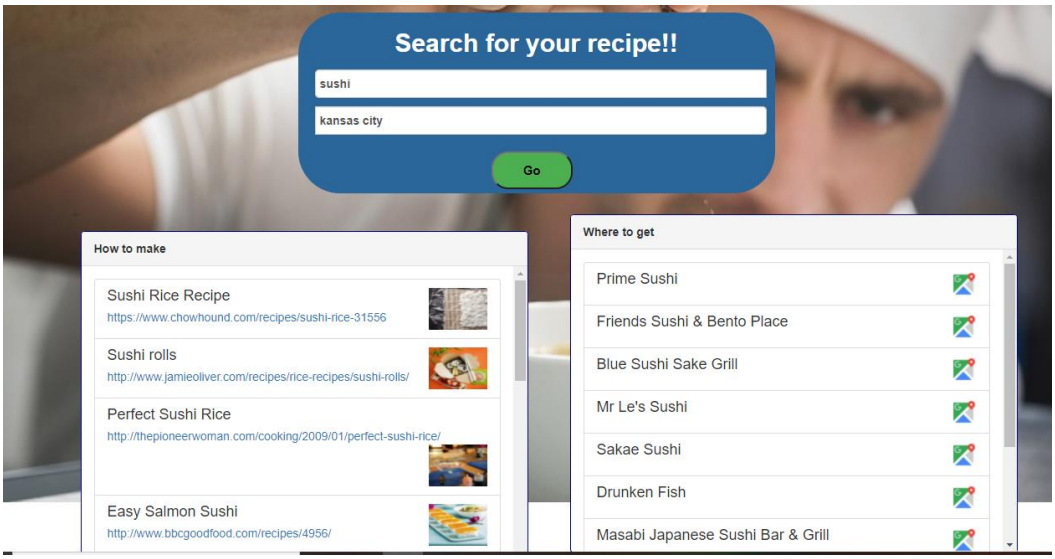
- Here is the home-screen, user can enter an email id for subscription.
- The content in this home page is static and the home page looks like below.



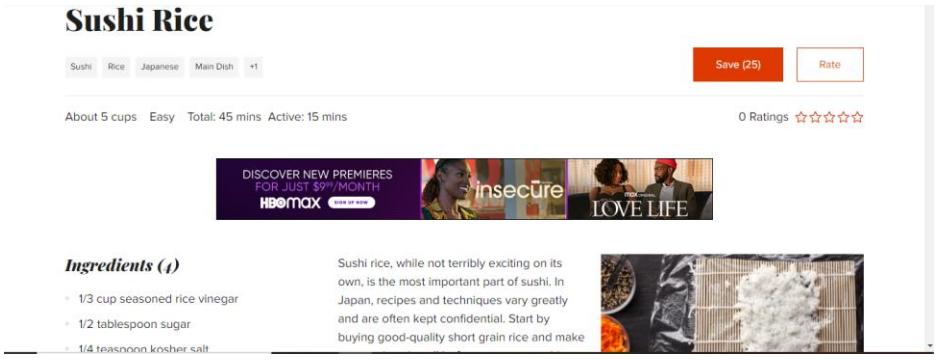
- Here is the search-recipe screen. When we click on ‘Search Recipe’ button from the home screen, user gets navigated to the below page.



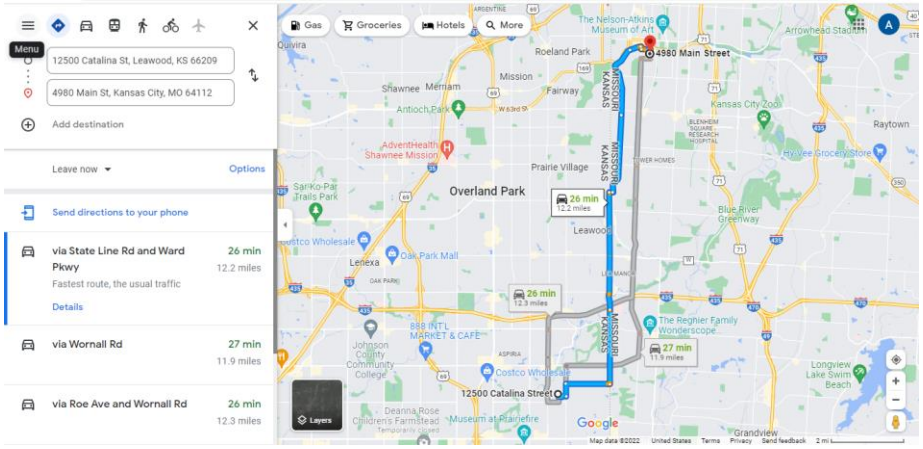
- When we enter recipe name & place, click on the ’Go’ button, recipe lists and venue lists get displayed as below.



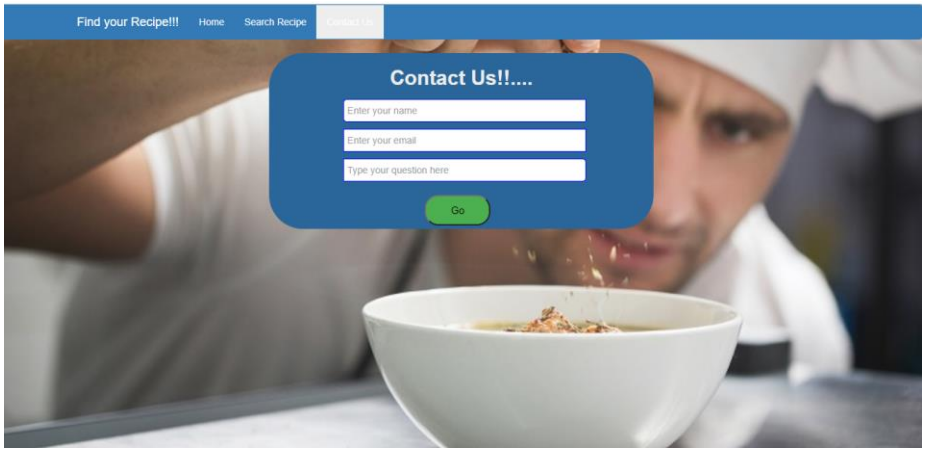
- When we click on the link that is displayed below the recipe name, user will be redirected to that particular recipe site which has all the ingredients information & preparation steps as well.



- When we click on google map icon, the page will be redirected to google maps page and shows us the destination map from the current location.



- Here is the contact-us screen, the content in this page is static.



Contribution:

We have contributed equally.

Conclusion:

- In this ICP, we have learned routers, services, HTTP and RESTful APIs in Angular and developed a web application using the same.

Challenges:

We faced some challenges while retrieving the information from FourSquare API. We could not be able to get the Venues values when we make the API request call using HTTP get method. Earlier it was working fine but it is not working at present. Since Venues is not working, we used places APIs and keys to get the values for location.

Apart from this, we have not faced any major challenges.