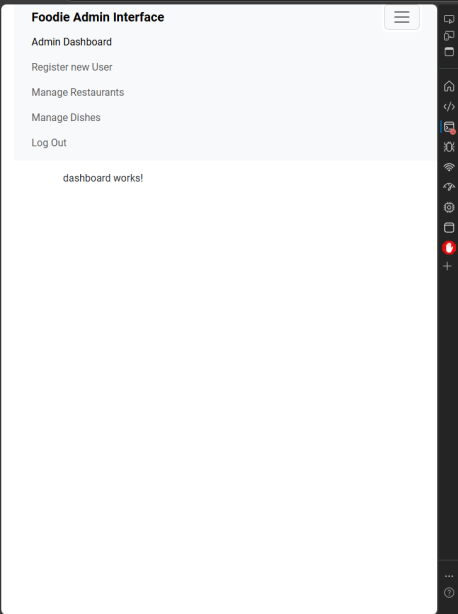
Capstone Project  
  
1) Angular Admin + Client Side  
Doing a login system that allows for different use roles allows for users and admins to go through the same system and receive different endpoints, including a different navbar and features.

When logging in as an admin, you will get more features such as the ability to register new users and add restaurants (and dishes)



Add Dish

A screenshot of a computer

Description automatically generated

Footer Shows Successful Creation

A screenshot of a computer

Description automatically generated

Successfully written into backend

A screenshot of a computer screen

Description automatically generated

Add Restaurant

A screenshot of a computer

Description automatically generated

Footer confirms successful creation

A screenshot of a computer

Description automatically generated

Appears in the mySQL backend

A screenshot of a computer

Description automatically generated

Generating auth guard for admin

A screen shot of a computer screen

Description automatically generated

Auth-guard.ts (Will add more authentication when backend is made)

Login as Admin:

A screen shot of a computer program

Description automatically generated

A computer screen shot of text

Description automatically generated

A screenshot of a computer

Description automatically generated

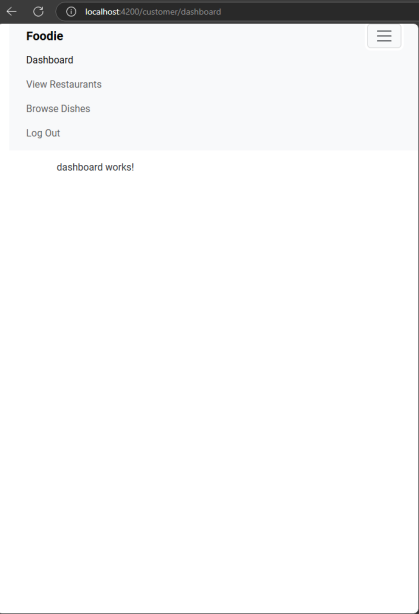
Admin View Dishes/Restaurants  
  
A screenshot of a computer

Description automatically generated

2) Angular User Side

User side has no management system or option to create additional users. This is accessed by logging in via an account that has UserRole = customer instead of admin.

Admin accounts cannot be created in registration.



View Dishes + Restaurants

3) Database SQL:  
  
CREATE DATABASE foodie;

USE foodie;

CREATE TABLE Restaurants (

id BIGINT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(255) NOT NULL,

location VARCHAR(255) NOT NULL

);

CREATE TABLE Dishes (

id BIGINT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(255) NOT NULL,

price FLOAT NOT NULL,

restaurant\_id BIGINT,

FOREIGN KEY (restaurant\_id) REFERENCES Restaurants(id) ON DELETE CASCADE

);  
  
Database Test Data  
  
INSERT INTO Restaurants (name, location) VALUES

('Greasy Grove', 'Manchester'),

('Pizza Place', 'Manchester'),

('Sushi Shop’, 'Leeds'),

('Tomato Town', 'Leeds’);  
  
INSERT INTO Dishes (name, price, restaurant\_id) VALUES

('Beef Burger', 2.99, 1),

('Cheeseburger', 3.99, 1),

('Pepperoni Pizza', 9.99, 2),

('Veggie Pizza', 8.99, 2),

(‘California Roll’, 4.99, 3),

('Edamame', 1.99, 3),

('Penne Pasta', 1.99, 4),

('Spaghetti Bolognaise', 3.99,4);

Test Queries

A screenshot of a menu

Description automatically generated

A screenshot of a menu

Description automatically generated

A screenshot of a computer

Description automatically generated

4) Spring Boot Java Backend

Initialise spring boot

A screenshot of a computer

Description automatically generated

A computer screen shot of a program

Description automatically generated

When ran, users generated in MYSQL

A screen shot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

Dishes + Restaurants created on springboot backend rather than SQL File:

A screen shot of a computer program

Description automatically generated

A screen shot of a computer

Description automatically generated