

Capstone Project: Food Delivery System App

Author: Rogelio Lotho

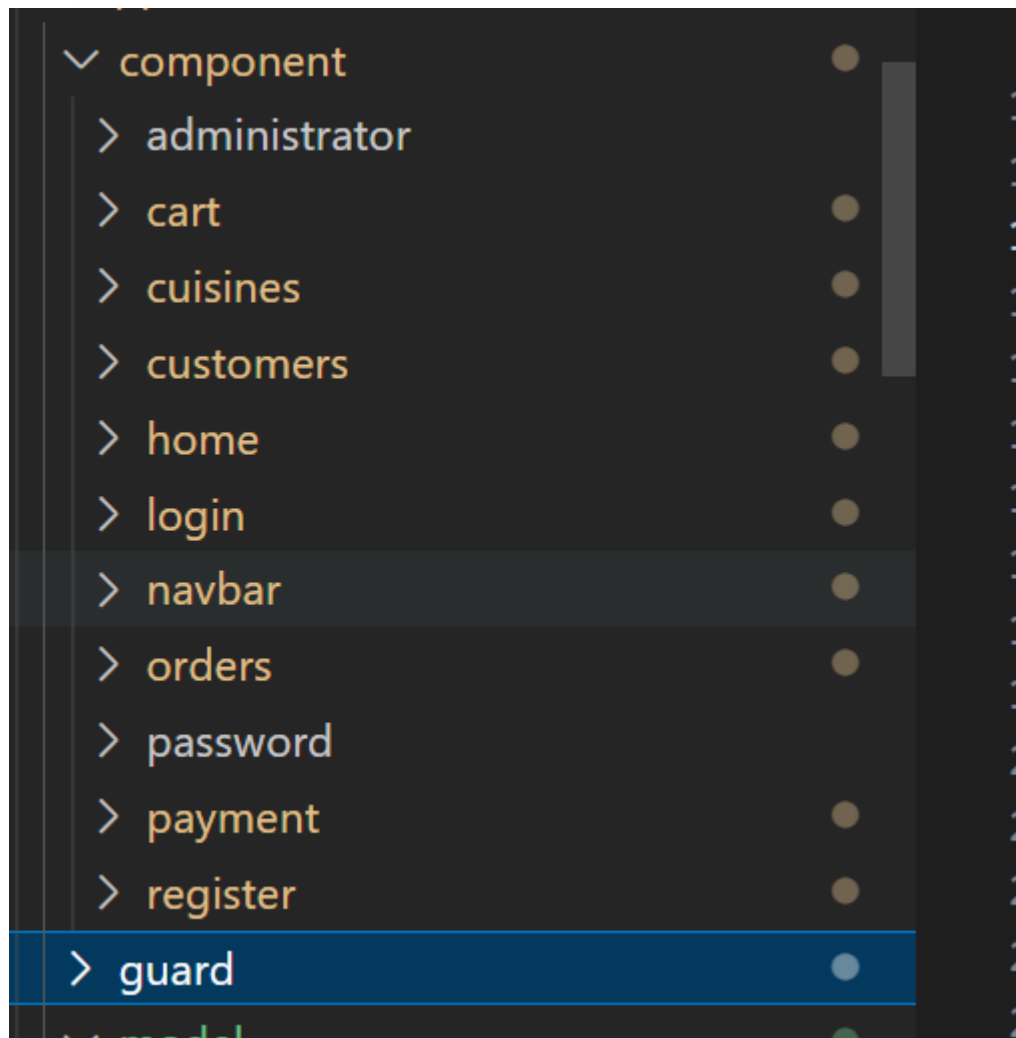
Goal: To Host the Application on AWS EC2 Instance, Automate the build and deployment using Jenkins on AWS EC2 and Containerize the application using Docker on AWS EC2

Objective: To build a dynamic and responsive food delivery app to display food items, filter based on user preferences, manage orders, user details.

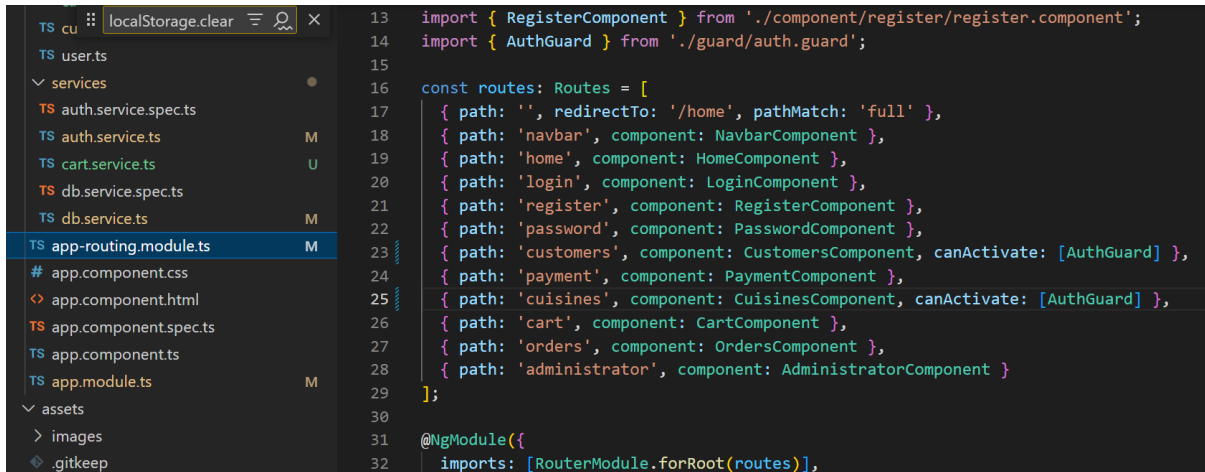
Tech Stack: Angular, Java, SpringBoot, MySQL, Jenkins, Docker, AWS, Git and GitHub

1.Task 1. Angular Compoments, Routing, Services and AuthGuard, Forms

1.1 Components: Cart, Administrator, Cuisines, Customers, Home, Login, Navbar, Orders, Password, Payment, Register



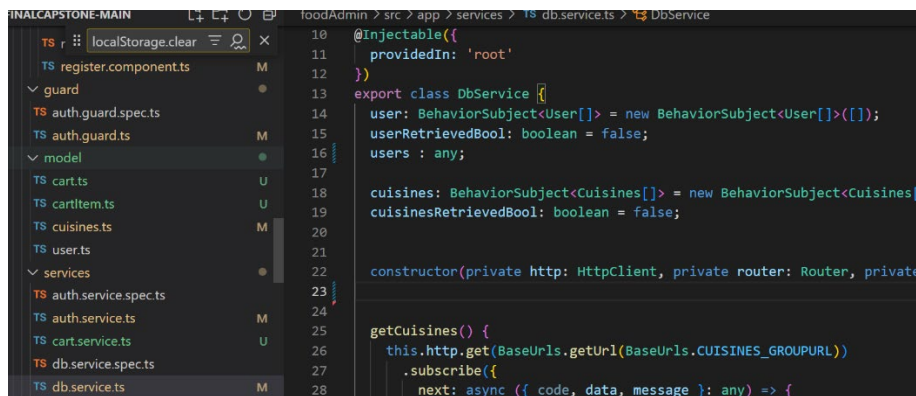
1.2 Routing



```
13 import { RegisterComponent } from './component/register/register.component';
14 import { AuthGuard } from './guard/auth.guard';
15
16 const routes: Routes = [
17   { path: '/', redirectTo: '/home', pathMatch: 'full' },
18   { path: 'navbar', component: NavbarComponent },
19   { path: 'home', component: HomeComponent },
20   { path: 'login', component: LoginComponent },
21   { path: 'register', component: RegisterComponent },
22   { path: 'password', component: PasswordComponent },
23   { path: 'customers', component: CustomersComponent, canActivate: [AuthGuard] },
24   { path: 'payment', component: PaymentComponent },
25   { path: 'cuisines', component: CuisinesComponent, canActivate: [AuthGuard] },
26   { path: 'cart', component: CartComponent },
27   { path: 'orders', component: OrdersComponent },
28   { path: 'administrator', component: AdministratorComponent }
29 ];
30
31 @NgModule({
32   imports: [RouterModule.forRoot(routes)],
```

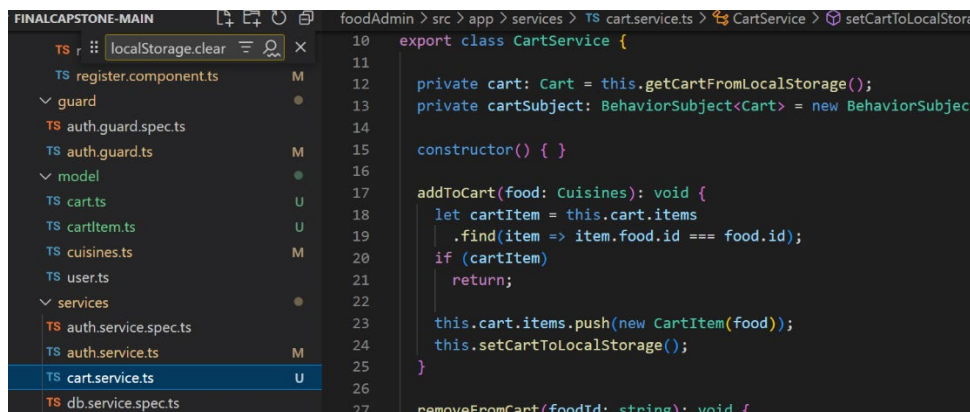
1.3 Services

1.3.1 dbService



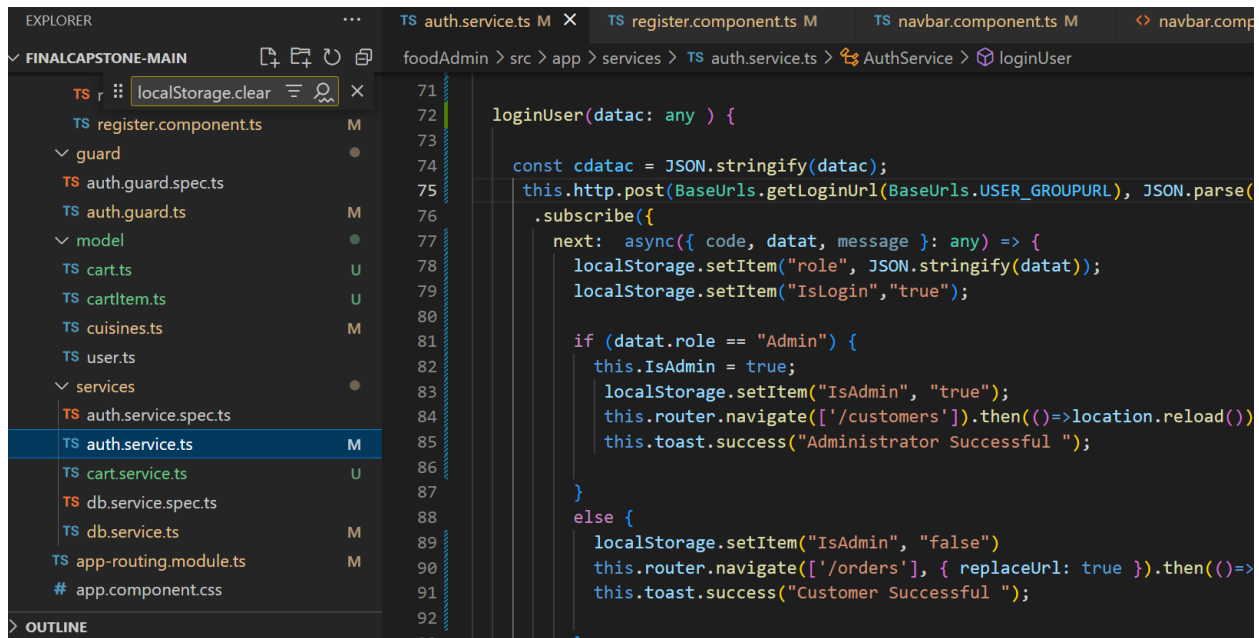
```
10 @Injectable({
11   providedIn: 'root'
12 })
13 export class DbService {
14   user: BehaviorSubject<User[]> = new BehaviorSubject<User[]>([]);
15   userRetrievedBool: boolean = false;
16   users : any;
17
18   cuisines: BehaviorSubject<Cuisines[]> = new BehaviorSubject<Cuisines[]>([]);
19   cuisinesRetrievedBool: boolean = false;
20
21   constructor(private http: HttpClient, private router: Router, private
22
23
24
25   getCuisines() {
26     this.http.get(BaseUrls.getUrl(BaseUrls.CUISINES_GROUPURL))
27       .subscribe({
28         next: async ({ code, data, message }: any) => {
```

1.3.2 Cart Service



```
10 export class CartService {
11
12   private cart: Cart = this.getCartFromLocalStorage();
13   private cartSubject: BehaviorSubject<Cart> = new BehaviorSubject<Cart>
14
15   constructor() { }
16
17   addToCart(food: Cuisines): void {
18     let cartItem = this.cart.items
19       .find(item => item.food.id === food.id);
20     if (cartItem)
21       return;
22
23     this.cart.items.push(new CartItem(food));
24     this.setCartToLocalStorage();
25   }
26
27   removeFromCart(foodId: string): void {
```

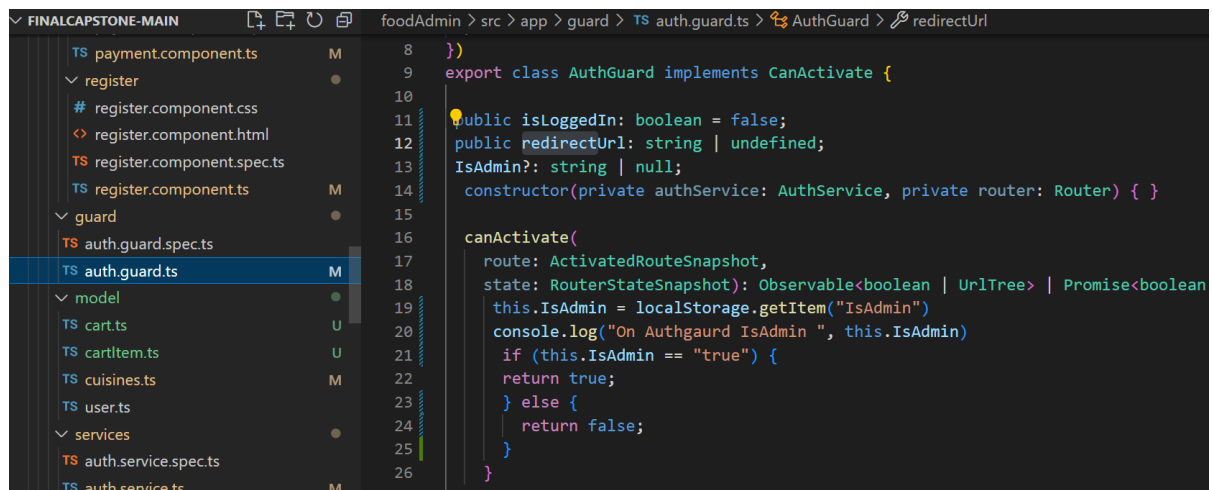
1.3.3 Auth Service



The screenshot shows the Visual Studio Code editor with the Explorer sidebar on the left and the Source Explorer on the right. The Explorer sidebar shows the project structure for 'FINALCAPSTONE-MAIN'. The Source Explorer shows the file 'auth.service.ts' in the 'services' folder. The code in 'auth.service.ts' defines a 'loginUser' function that takes 'data: any' as an argument. It uses 'JSON.stringify' to convert the data to a string and 'this.http.post' to send a POST request to the login endpoint. The response is handled with 'JSON.parse' and 'this.subscribe'. The 'next' function in the subscription checks the 'role' of the user. If the role is 'Admin', it sets 'this.IsAdmin' to 'true', updates 'localStorage' with the role and login status, navigates to '/customers', and shows a success toast. If the role is not 'Admin', it sets 'this.IsAdmin' to 'false', navigates to '/orders', and shows a success toast.

```
loginUser(data: any ) {  
    const cdatac = JSON.stringify(data);  
    this.http.post(BaseUrls.getLoginUrl(BaseUrls.USER_GROUPURL), JSON.parse(  
        .subscribe({  
            next: async({ code, datat, message }: any) => {  
                localStorage.setItem("role", JSON.stringify(datat));  
                localStorage.setItem("IsLogin","true");  
  
                if (datat.role == "Admin") {  
                    this.IsAdmin = true;  
                    localStorage.setItem("IsAdmin", "true");  
                    this.router.navigate(['/customers']).then(()=>location.reload())  
                    this.toast.success("Administrator Successful ");  
                }  
                else {  
                    localStorage.setItem("IsAdmin", "false")  
                    this.router.navigate(['/orders'], { replaceUrl: true }).then(()=>  
                    this.toast.success("Customer Successful ");  
                }  
            }  
        })  
    )  
}
```

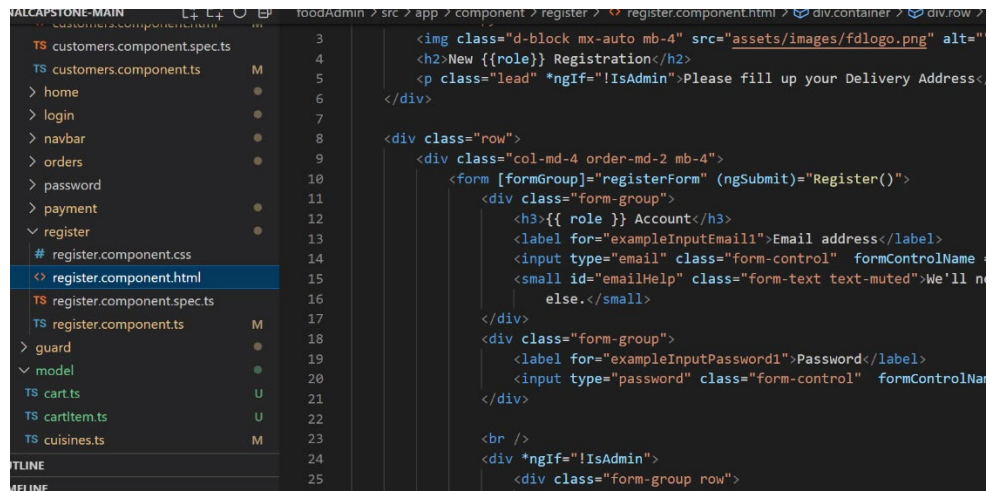
1.4 AuthGuard



The screenshot shows the Visual Studio Code editor with the Explorer sidebar on the left and the Source Explorer on the right. The Explorer sidebar shows the project structure for 'FINALCAPSTONE-MAIN'. The Source Explorer shows the file 'auth.guard.ts' in the 'guard' folder. The code in 'auth.guard.ts' defines an 'AuthGuard' class that implements the 'CanActivate' interface. It has two public properties: 'isLoggedInIn' (boolean) and 'redirectUrl' (string | undefined). It has a private property 'IsAdmin?' (string | null). The constructor takes 'authService: AuthService' and 'router: Router' as arguments. The 'canActivate' method takes 'route: ActivatedRouteSnapshot' and 'state: RouterStateSnapshot' as arguments. It checks 'this.IsAdmin' from 'localStorage'. If it is 'true', it returns 'true'. Otherwise, it returns 'false'.

```
export class AuthGuard implements CanActivate {  
    public isLoggedInIn: boolean = false;  
    public redirectUrl: string | undefined;  
    IsAdmin?: string | null;  
    constructor(private authService: AuthService, private router: Router) { }  
  
    canActivate(  
        route: ActivatedRouteSnapshot,  
        state: RouterStateSnapshot): Observable<boolean | UrlTree> | Promise<boolean>  
    {  
        this.IsAdmin = localStorage.getItem("IsAdmin")  
        console.log("On Authgaurd IsAdmin ", this.IsAdmin)  
        if (this.IsAdmin == "true") {  
            return true;  
        } else {  
            return false;  
        }  
    }  
}
```

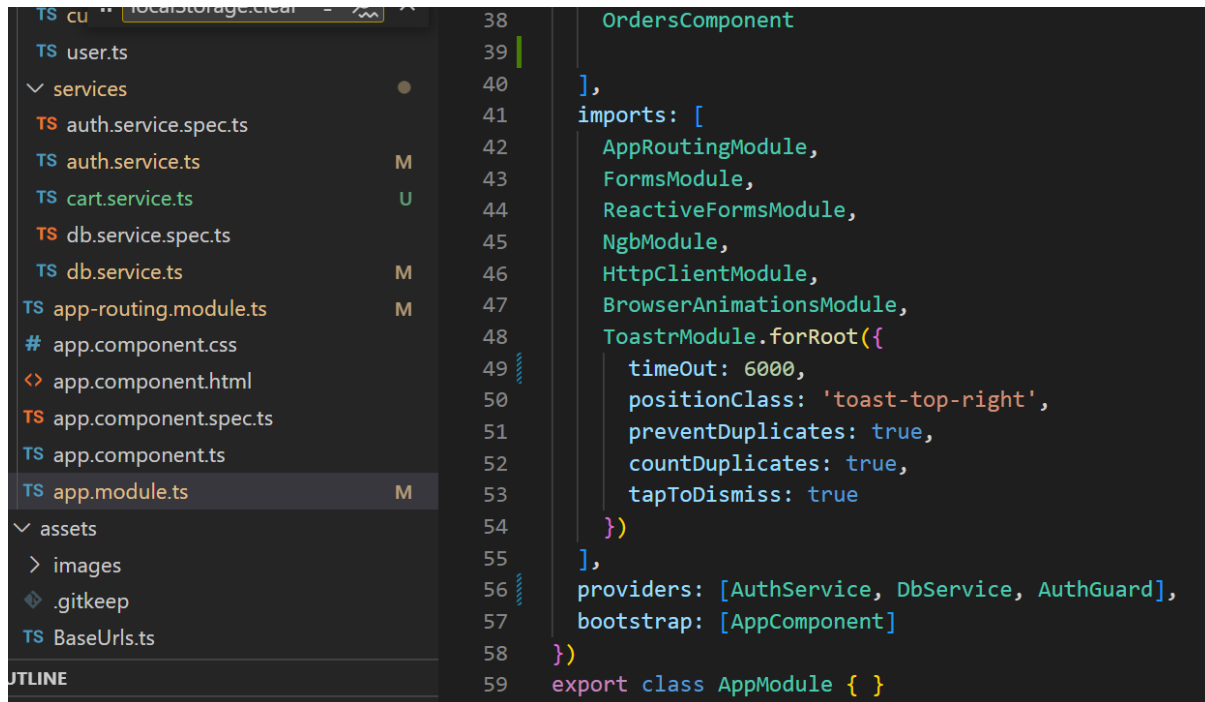
1.4.1 Form



```
3      
4      <h2>New {{role}} Registration</h2>
5      <p class="lead" *ngIf="!isAdmin">Please fill up your Delivery Address</p>
6
7      </div>
8
9      <div class="row">
10         <div class="col-md-4 order-md-2 mb-4">
11             <form [formGroup]="registerForm" (ngSubmit)="Register()">
12                 <div class="form-group">
13                     <h3>{{ role }} Account</h3>
14                     <label for="exampleInputEmail1">Email address</label>
15                     <input type="email" class="form-control" formControlName="email" />
16                     <small id="emailHelp" class="form-text text-muted">We'll never share your email with anyone else.</small>
17                 </div>
18                 <div class="form-group">
19                     <label for="exampleInputPassword1">Password</label>
20                     <input type="password" class="form-control" formControlName="password" />
21                 </div>
22             </form>
23         </div>
24         <div *ngIf="!isAdmin">
25             <div class="form-group row">
```

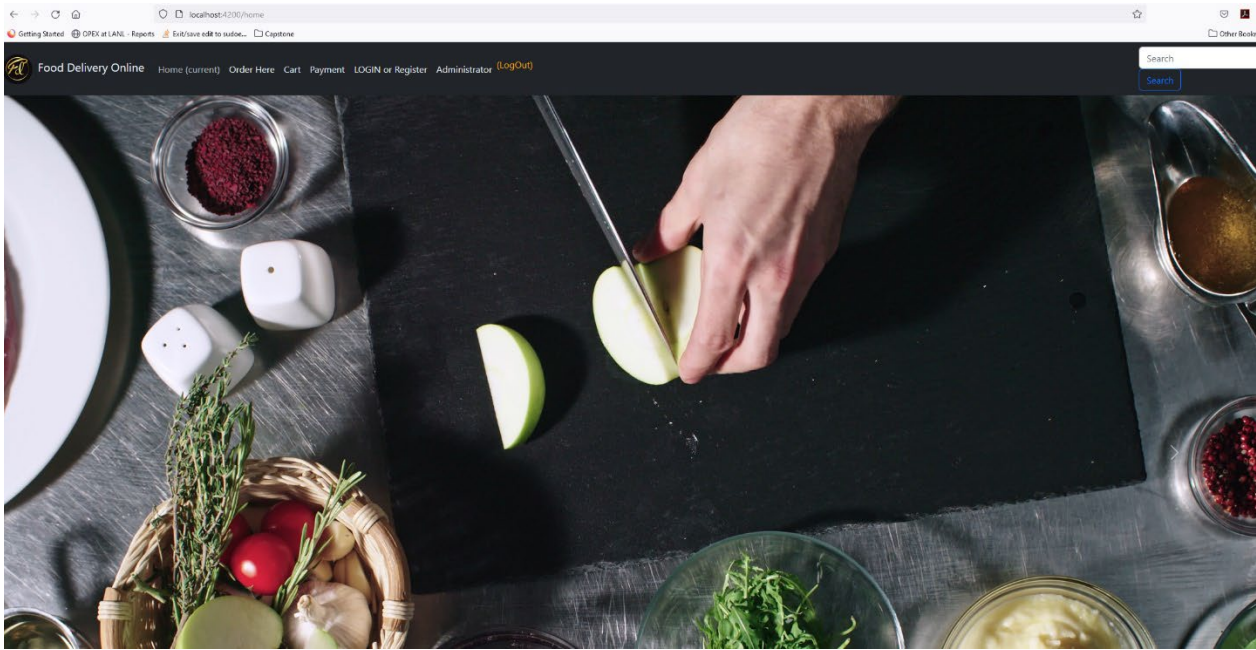
1.4 Task 2 Angular Components, Routing, Services, and Auth Guard, Forms

2.1 App Routing Module

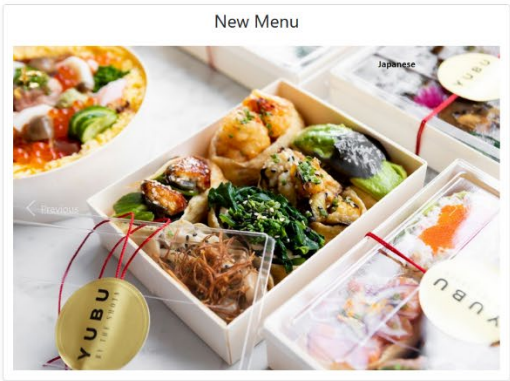
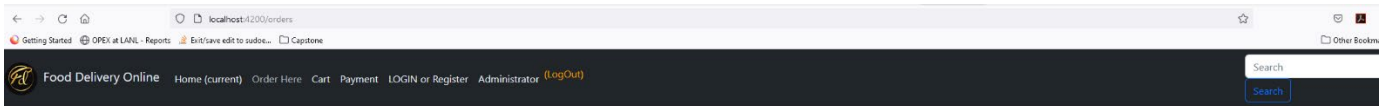


```
38
39
40
41
42 imports: [
43     AppRoutingModule,
44     FormsModule,
45     ReactiveFormsModule,
46     NgbModule,
47     HttpClientModule,
48     BrowserAnimationsModule,
49     ToastrModule.forRoot({
50         timeout: 6000,
51         positionClass: 'toast-top-right',
52         preventDuplicates: true,
53         countDuplicates: true,
54         tapToDismiss: true
55     })
56 ],
57 providers: [AuthService, DbService, AuthGuard],
58 bootstrap: [AppComponent]
59
60 export class AppModule { }
```








2.2. Home Component



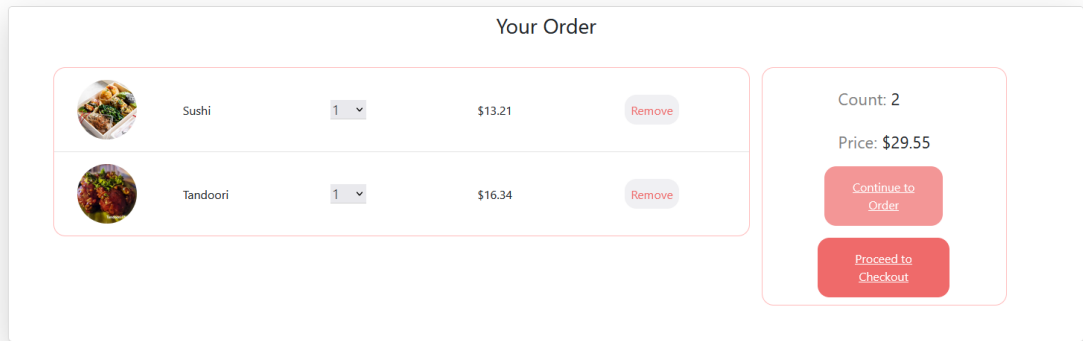
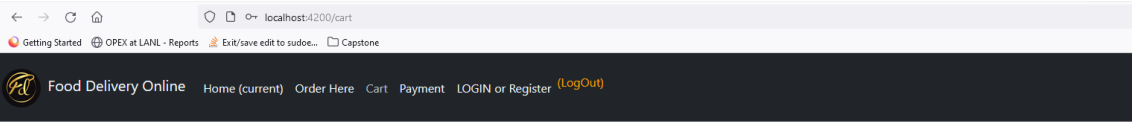
2.3 Order Component



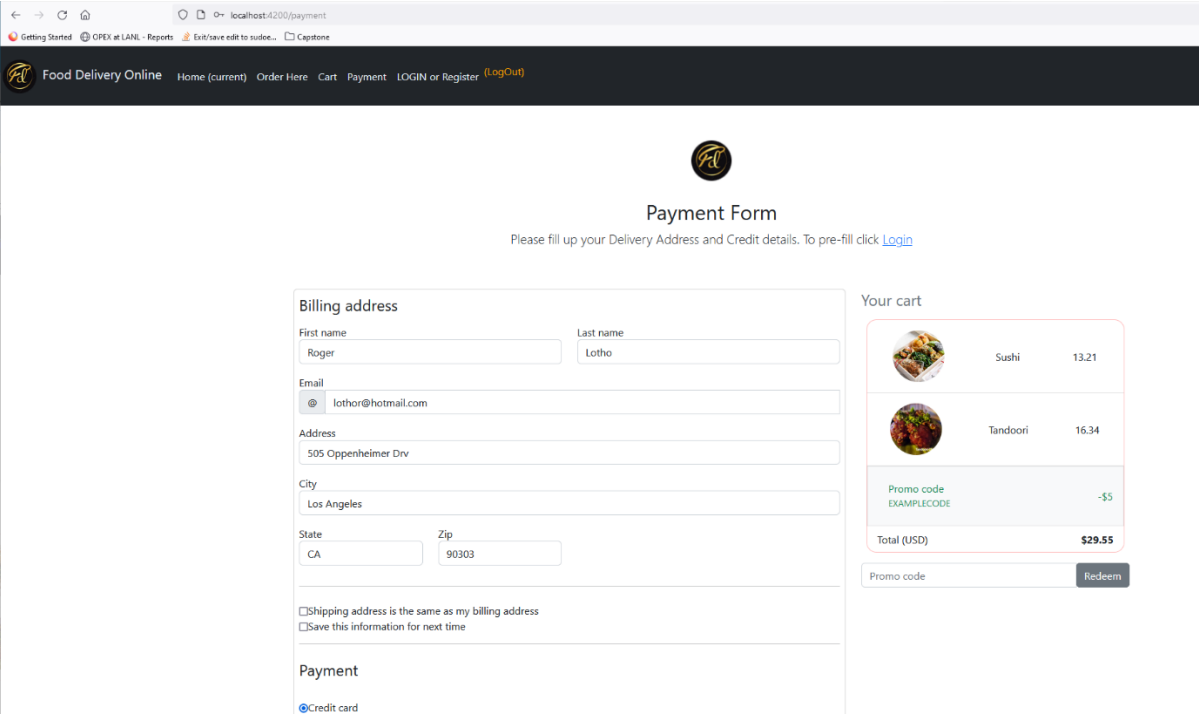
SELECT & ORDER:

 Sushi Japan ⌚ 10-45 mins. ORDER: \$13.21 Add To Cart	 Lumpia Asi ⌚ 10-45 mins. ORDER: \$21.21 Add To Cart	 Taco South America ⌚ 10-45 mins. ORDER: \$14.21 Add To Cart	 Hamburger American ⌚ 10-45 mins. ORDER: \$18.21 Add To Cart	 Spaghetti Italian ⌚ 10-45 mins. ORDER: \$6.3 Add To Cart
 Tandoori South East Asia ⌚ 10-45 mins. ORDER: \$16.34 Add To Cart	 Samosa South East Asia ⌚ 10-45 mins. ORDER: \$13.45 Add To Cart			

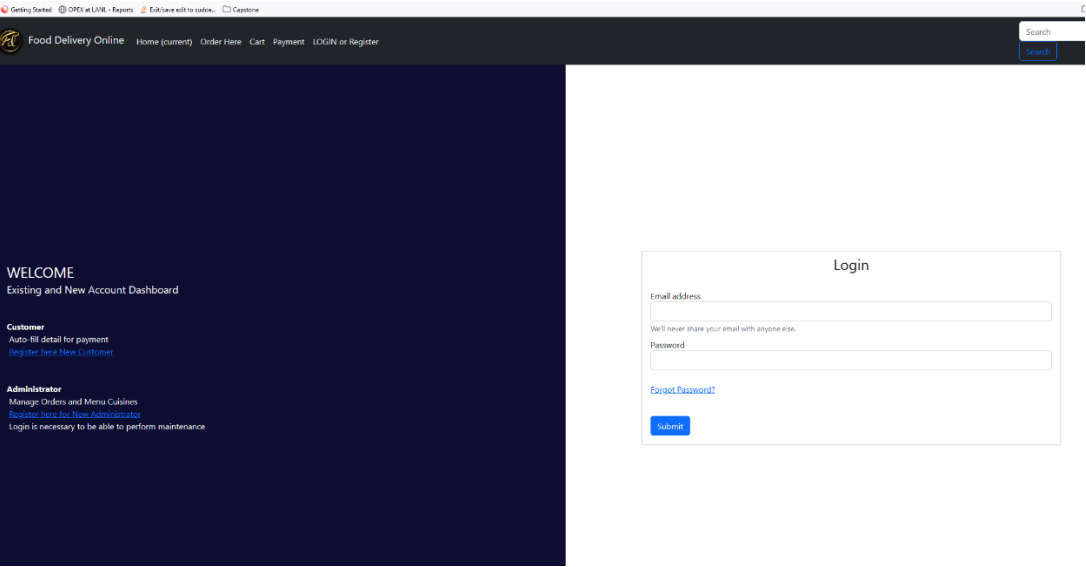
2.4. Cart Component



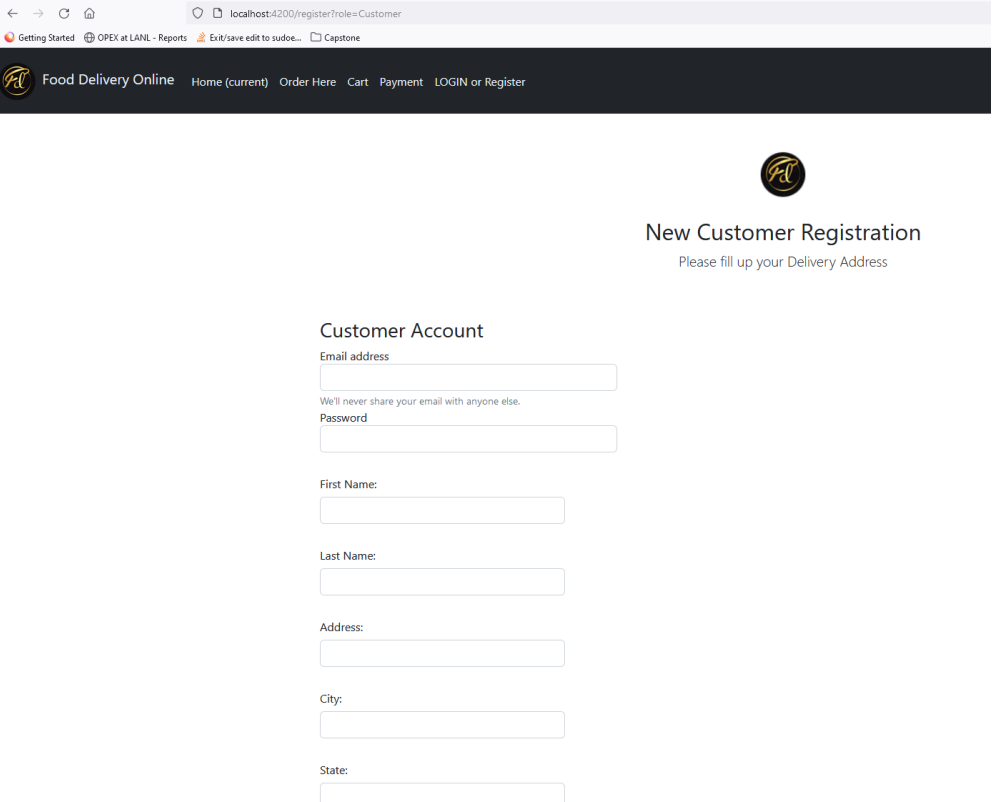
2.5 Payment Component



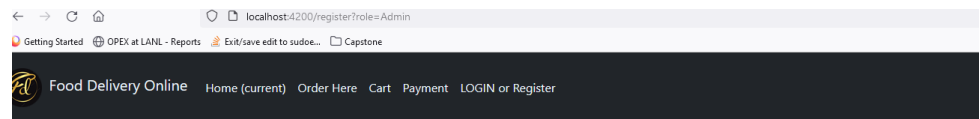
2.6. Login Component



2.7. Register Customer



2.8 Register Admin



New Admin Registration

Admin Account

Email address

We'll never share your email with anyone else.

Password






[Submit](#)

2.9. Admin Cuisine Maintenance

Food Delivery Online

Home (current)Order HereCartPaymentLOGIN or RegisterAdministrator (Logout)

Add New Cuisine

Sr. No	Name	Origin	Price	Available	As of	Image	Maintain
1	Sushi	Japan	13.21	false	2022-09-12		Update Cuisine Delete Cuisine Add Cuisine Cuisine
2	Lumpia	Asi	21.21	false	2022-04-24		Update Cuisine Delete Cuisine Add Cuisine Cuisine
4	Taco	South America	14.21	true	2022-09-21		Update Cuisine Delete Cuisine Add Cuisine Cuisine
5	Hamburger	American	18.21	true	2023-06-01		Update Cuisine Delete Cuisine Add Cuisine Cuisine
6	Spaghetti	Italian	6.3	false	2022-12-24		Update Cuisine Delete Cuisine Add Cuisine Cuisine

2.10 Admin Update Cuisine

The screenshot shows a web application interface for updating a cuisine. On the left, a table lists existing cuisines. A modal titled 'Cuisine Update' is open, allowing for editing a specific cuisine's details.

Cuisines Table:

Sr. No	Name	Origin
1	Sushi	Japan
2	Lumpia	Asi
4	Taco	South America
5	Hamburger	American

Cuisine Update Form:

- Name: Sushi
- Origin: Japan
- Price: 13.21
- Available (true or false): false
- Added On: 2022-09-12
- Image filename: sushi
- Select Image File: Browse... (No file selected.)

Buttons: Cancel, Save

2.11. Admin Update Account Customer/Admin

The screenshot shows a web application interface for updating a customer's account. On the left, a table lists existing customers. A modal titled 'Customer Update' is open, allowing for editing a specific customer's details.

Food Delivery Customers Table:

#	First	Last	Email
1	Roger	Lotho	lothor@hotmail.com
2	Marth	Care	mc@gmail.com
3	Divo	Thur	dthru@hotmail.com
4	Carlos	Mi	cm@gmail.gov

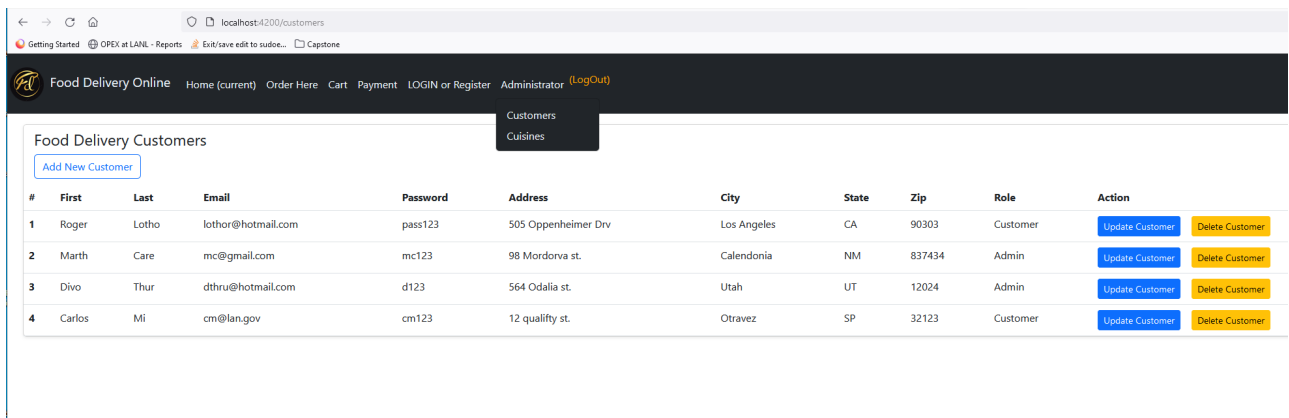
Customer Update Form:

- First Name: Roger
- Last Name: Lotho
- eMail: lothor@hotmail.com
- Password: pass123
- Phone: (505)303-303-3434
- Delivery Address:
 - Address: 505 Oppenheimer Drv
 - City: Los Angeles
 - State: CA
 - ZipCode: 90303

Buttons: Cancel, Save

2.12.AuthGuard

The screenshot shows a VS Code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like 'model' and 'services', and files like 'auth.guard.ts' which is selected. The code editor shows the implementation of the 'AuthGuard' class, including the 'canActivate' method which checks if the user is an admin using 'localStorage.getItem'.



3. Task 3: SQL CRUD Commands, Primary and Foreign Key Relationship

[illegible]

```

10 • select * from user;
11
12 • describe cuisine;
13
14
15
16
17
18

```

id	address	city	email	firstname	lastname	password	phone	role	state	zip
1	505 Oppenheimer Drv	Los Angeles	lothor@hotmail.com	Roger	Lotho	pass123	(505) 303-303-3434	Customer	CA	90303
2	98 Mordorva st.	Calendonia	mc@gmail.com	Marth	Care	mc123	(310) 550-3434	Admin	NM	837434
3	564 Odalia st.	Utah	dthru@hotmail.com	Divro	Thur	d123	(210) 534-3434	Admin	UT	12024
4	12 qualify st.	Otravez	cm@lan.gov	Carlos	Mi	cm123	(250) 443-34343	Customer	SP	32123
7	NULL	NULL	a	NULL	NULL	a	NULL	Admin	NULL	NULL

4.Task 4: SpringBoot Web Dependency, RestController, RequestMapping, Post and Get request

4.1 SpringBoot Web Dependency

Project Explorer

foodBackend [boot]

src/main/java

com.fds.FoodDelivery

FoodDeliveryApplication.java

com.fds.FoodDelivery.controller

CuisinesController.java

UserController.java

com.fds.FoodDelivery.model

Cuisines.java

Response.java

User.java

com.fds.FoodDelivery.repository

CuisinesRepository.java

UserRepository.java

src/main/resources

static

templates

application.properties

src/test/java

JRE System Library [JavaSE-17]

Maven Dependencies

target/generated-sources/annotations

target/generated-test-sources/test-annotations

src

target

HELP.md

mvnw

mvnw.cmd

pom.xml

Response.java

UserReposit...

FoodDelivery...

User.java

CuisinesRep...

foodBackend/...

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
4   <modelVersion>4.0.0</modelVersion>
5   <parent>
6     <groupId>org.springframework.boot</groupId>
7     <artifactId>spring-boot-starter-parent</artifactId>
8     <version>3.0.2</version>
9     <relativePath><!-- lookup parent from repository -->
10   </parent>
11   <groupId>com.fds</groupId>
12   <artifactId>FoodDelivery</artifactId>
13   <version>0.0.1-SNAPSHOT</version>
14   <name>FoodDelivery</name>
15   <description>Final Capstone project</description>
16   <properties>
17     <java.version>17</java.version>
18   </properties>
19   <dependencies>
20     <dependency>
21       <groupId>org.springframework.boot</groupId>
22       <artifactId>spring-boot-starter-data-jpa</artifactId>
23     </dependency>
24     <dependency>
25       <groupId>org.springframework.boot</groupId>
26       <artifactId>spring-boot-starter-web</artifactId>
27     </dependency>
28   </dependencies>
29   <dependency>
30     <groupId>com.mysql</groupId>
31     <artifactId>mysql-connector-j</artifactId>
32     <scope>runtime</scope>
33   </dependency>
34   <dependency>
35     <groupId>org.springframework.boot</groupId>
36     <artifactId>spring-boot-starter-test</artifactId>
37     <scope>test</scope>
38   </dependency>
39 </dependencies>
40
41   <build>
42     <plugins>
43       <plugin>
44         <groupId>org.springframework.boot</groupId>
45         <artifactId>spring-boot-maven-plugin</artifactId>
46       </plugin>
47     </plugins>
48   </build>
49
50 </project>
51

```

4.2 Rest Controller

finalcapstone-main - foodBackend/src/main/java/com/fds/FoodDelivery/controller/UserController.java - Eclipse IDE

```
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer X
  foodBackend [boot]
    src/main/java
      com.fds.FoodDelivery
        com.fds.FoodDelivery.controller
          CuisinesController.java
          UserController.java
        com.fds.FoodDelivery.model
          Cuisines.java
          Response.java
          User.java
        com.fds.FoodDelivery.repository
      src/main/resources
        static
        templates
        application.properties
      src/test/java
      JRE System Library [JavaSE-17]
      Maven Dependencies
      target/generated-sources/annotations
      target/generated-test-sources/test-annotations
      src
      target
        HELP.md
        mvnw
        mvnw.cmd
        pom.xml

application... index.html Response.java UserReposit... UserController... Cuisines.java CuisinesCon...
1 package com.fds.FoodDelivery.controller;
2
3 import java.util.ArrayList;
4
5 @RestController
6 @CrossOrigin(origins = "http://localhost:4200")
7 @RequestMapping("/user")
8 public class UserController {
9
10     public static String TAG = "[User]";
11
12     @Autowired
13     UserRepository userRepo;
14
15     Date date = new Date();
16
17     @PostMapping(path="/add")
18     //public Response addUser( @RequestParam String email, @RequestParam String password, @RequestParam(required=false) Str
19     public Response<User> addUser(@RequestBody User data) {
20         //User data = new User(null, email, password, firstName, lastName, phone, address, city, state, zip, role);
21         userRepo.save(data);
22         return new Response<User>(101, TAG+" "+data+" Added Successful on "+date, data);
23     }
24
25     @DeleteMapping(path="/delete/{id}")
26     public Response<User> deleteUser(@PathVariable("id") Integer id) {
27
28         User auser = userRepo.findById(id).get();
29         userRepo.deleteById(id);
30         userRepo.delete(auser);
31         System.out.print(auser);
32         return new Response<User> (101, auser+" "+TAG+"s DELETED Successful on "+date, auser);
33     }
34
35     @GetMapping(path="/get")
36     public Response<User> getUser() {
37         ArrayList<User> data = new ArrayList<User>();
38         userRepo.findAll().forEach((auser)->data.add(auser));
39         System.out.print(data);
40         return new Response<User>(101, " success ", (List<User>) data);
41     }
42
43 }
```

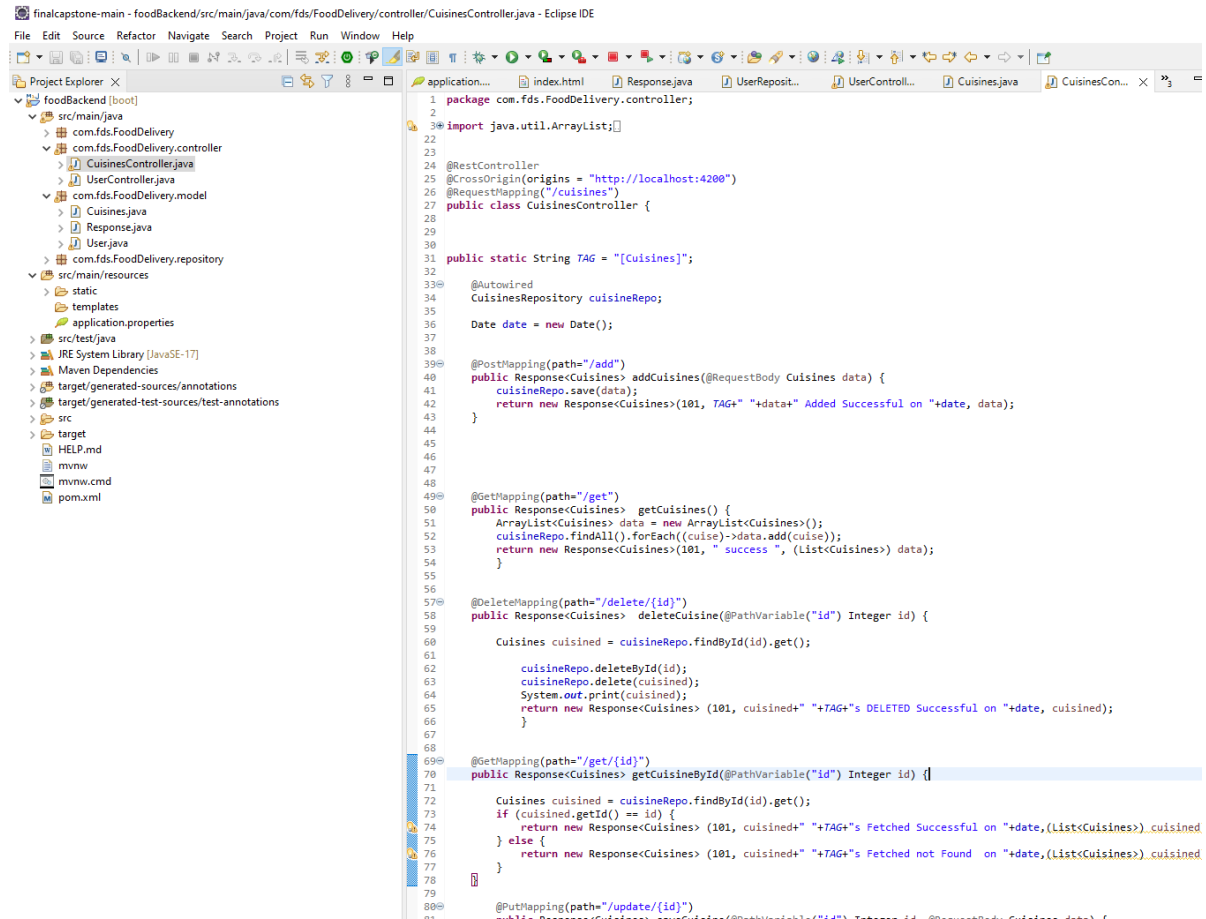
4.3 Request Mapping

finalcapstone-main - foodBackend/src/main/java/com/fds/FoodDelivery/model/Cuisines.java - Eclipse IDE

```
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer X
  foodBackend [boot]
    src/main/java
      com.fds.FoodDelivery
        com.fds.FoodDelivery.controller
          CuisinesController.java
          UserController.java
        com.fds.FoodDelivery.model
          Cuisines.java
          Response.java
          User.java
        com.fds.FoodDelivery.repository
      src/main/resources
        static
        templates
        application.properties
      src/test/java
      JRE System Library [JavaSE-17]
      Maven Dependencies
      target/generated-sources/annotations
      target/generated-test-sources/test-annotations
      src
      target
        HELP.md
        mvnw
        mvnw.cmd
        pom.xml

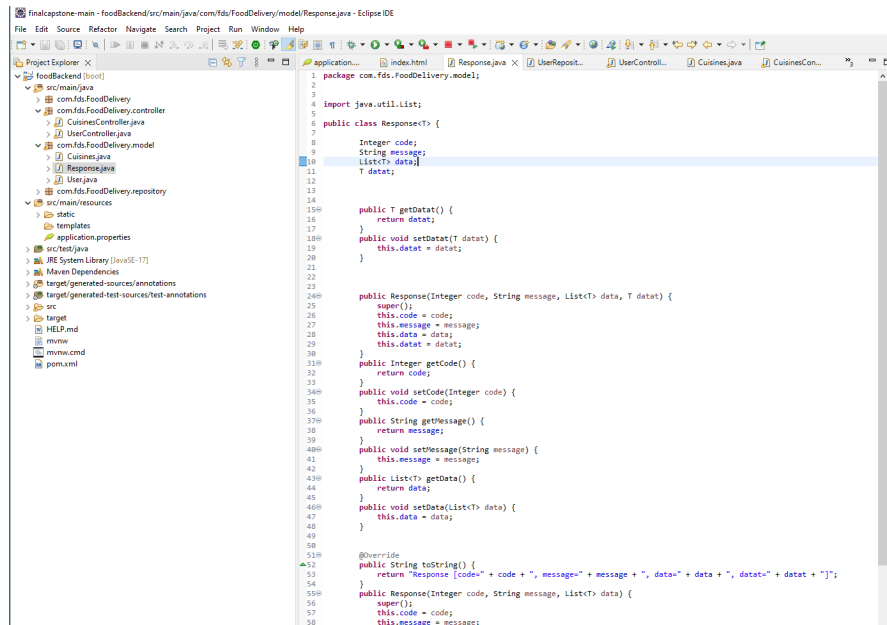
application... index.html Response.java UserReposit... UserController... Cuisines.java CuisinesCon...
16 public String origin;
17 public Double price;
18 public Boolean available;
19 public String addidion;
20 public String image;
21 public String fimage;
22
23 public Cuisines(Integer id, String name, String origin, Double price, Boolean available, String addidion,
24                 String image, String fimage) {
25     super();
26     this.id = id;
27     this.name = name;
28     this.origin = origin;
29     this.price = price;
30     this.available = available;
31     this.addidion = addidion;
32     this.image = image;
33     this.fimage = fimage;
34 }
35 public String getImage() {
36     return fimage;
37 }
38 public void setImage(String fimage) {
39     this.fimage = fimage;
40 }
41 public Integer getId() {
42     return id;
43 }
44 public void setId(Integer id) {
45     this.id = id;
46 }
47 public String getName() {
48     return name;
49 }
50 public void setName(String name) {
51     this.name = name;
52 }
53 public String getOrigin() {
54     return origin;
55 }
56 public void setOrigin(String origin) {
57     this.origin = origin;
58 }
59
60 }
```

4.4. Post and Get Request



```
1 package com.fds.FoodDelivery.controller;
2
3 import java.util.ArrayList;
4
5 @RestController
6 @CrossOrigin(origins = "http://localhost:4200")
7 @RequestMapping("/cuisines")
8 public class CuisinesController {
9
10     public static String TAG = "[Cuisines]";
11
12     @Autowired
13     CuisinesRepository cuisinesRepo;
14
15     Date date = new Date();
16
17     @PostMapping(path="/add")
18     public Response<Cuisines> addCuisines(@RequestBody Cuisines data) {
19         cuisinesRepo.save(data);
20         return new Response<Cuisines>(101, TAG+" "+date+" Added Successful on "+date, data);
21     }
22
23     @GetMapping(path="/get")
24     public Response<Cuisines> getCuisines() {
25         ArrayList<Cuisines> data = new ArrayList<Cuisines>();
26         cuisinesRepo.findAll().forEach((cuisine)->data.add(cuisine));
27         return new Response<Cuisines>(101, " success ", (List<Cuisines>) data);
28     }
29
30     @DeleteMapping(path="/delete/{id}")
31     public Response<Cuisines> deleteCuisine(@PathVariable("id") Integer id) {
32
33         Cuisines cuisined = cuisinesRepo.findById(id).get();
34
35         cuisinesRepo.deleteById(id);
36         cuisinesRepo.delete(cuisined);
37         System.out.print(cuisined);
38         return new Response<Cuisines> (101, cuisined+" "+TAG+"s DELETED Successful on "+date, cuisined);
39     }
40
41     @GetMapping(path="/get/{id}")
42     public Response<Cuisines> getCuisineById(@PathVariable("id") Integer id) {
43
44         Cuisines cuisined = cuisinesRepo.findById(id).get();
45         if (cuisined.getId() == id) {
46             return new Response<Cuisines> (101, cuisined+" "+TAG+"s Fetched Successful on "+date,(List<Cuisines>) cuisined)
47         } else {
48             return new Response<Cuisines> (101, cuisined+" "+TAG+"s Fetched not Found on "+date,(List<Cuisines>) cuisined)
49         }
50     }
51
52     @PutMapping(path="/update/{id}")
53     public Response<Cuisines> updateCuisine(@PathVariable("id") Integer id, @RequestBody Cuisines data) {
54
55         Cuisines cuisined = cuisinesRepo.findById(id).get();
56         cuisinesRepo.save(data);
57         return new Response<Cuisines> (101, cuisined+" "+TAG+"s Updated Successful on "+date, cuisined);
58     }
59 }
```

4.5 Response model



```
1 package com.fds.FoodDelivery.model;
2
3 import java.util.List;
4
5 public class Response<T> {
6     Integer code;
7     String message;
8     List<T> data;
9     T datat;
10
11     public T getDatat() {
12         return datat;
13     }
14
15     public void setDatat(T datat) {
16         this.datat = datat;
17     }
18
19     public Response(Integer code, String message, List<T> data, T datat) {
20         super();
21         this.code = code;
22         this.message = message;
23         this.data = data;
24         this.datat = datat;
25     }
26
27     public Integer getCode() {
28         return code;
29     }
30
31     public void setCode(Integer code) {
32         this.code = code;
33     }
34
35     public String getMessage() {
36         return message;
37     }
38
39     public void setMessage(String message) {
40         this.message = message;
41     }
42
43     public List<T> getData() {
44         return data;
45     }
46
47     public void setData(List<T> data) {
48         this.data = data;
49     }
50
51     @Override
52     public String toString() {
53         return "Response {code=" + code + ", message=" + message + ", data=" + data + ", datat=" + datat + "}";
54     }
55
56     public Response(Integer code, String message, List<T> data) {
57         super();
58         this.code = code;
59         this.message = message;
60     }
61 }
```

5.Task 5: Angular HTTP Client Library, HTTP Request Response, JSON

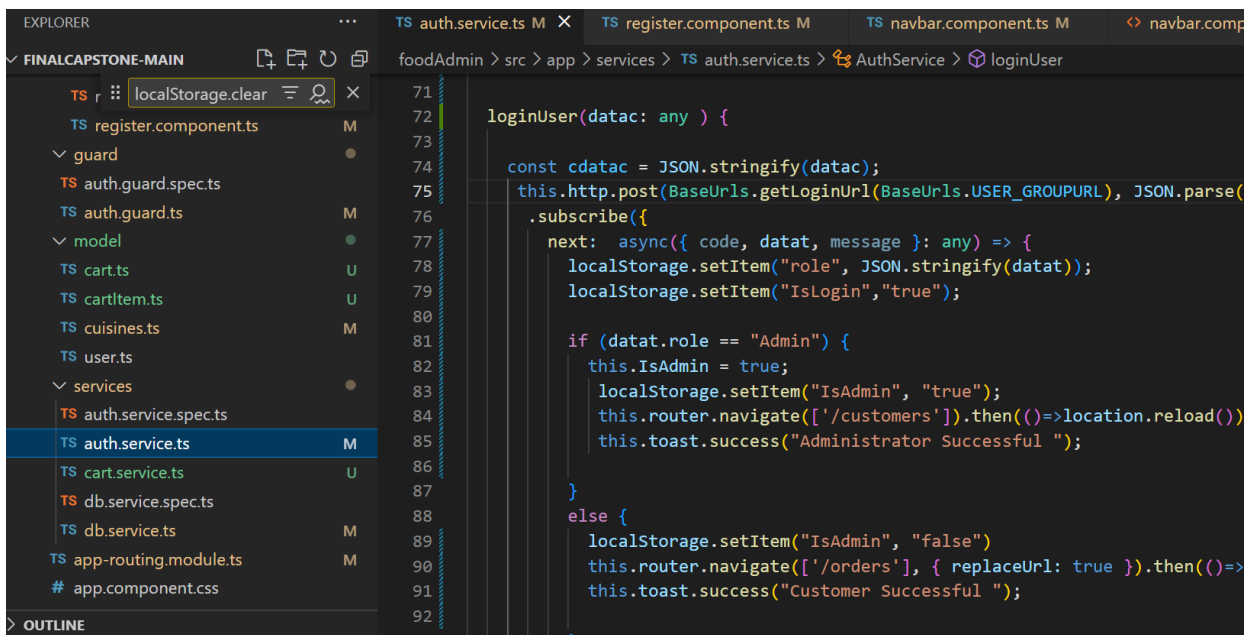
5.1. Base URL Cross Origin



The screenshot shows the VS Code editor with the Explorer on the left and the Editor on the right. The Explorer shows the project structure for 'FINALCAPSTONE-MAIN' with files like 'db.service.spec.ts', 'db.service.ts', 'app-routing.module.ts', 'app.component.css', 'app.component.html', 'app.component.spec.ts', 'app.component.ts', 'app.module.ts', 'assets', 'images', '.gitkeep', and 'BaseUrls.ts'. The Editor shows the 'BaseUrls.ts' file with the following code:

```
1  export class BaseUrls {
2      public static readonly BASE_HREF: string = "http://localhost:8082";
3
4      public static readonly USER_GROUPURL: string = "user";
5      public static readonly CUISINES_GROUPURL: string = "cuisines";
6
7
8      public static getLoginUrl(key: string): string { return `${this.BASE_HREF}/${key}`; }
9      public static getUrl(key: string): string { return `${this.BASE_HREF}/${key}/get`; }
10     public static addUrl(key: string): string { return `${this.BASE_HREF}/${key}/add`; }
11     public static deleteUrl(key: string): string { return `${this.BASE_HREF}/${key}/delete`; }
12     public static updateUrl(key: string): string { return `${this.BASE_HREF}/${key}/update`; }
13
14 }
```

5.1 HTTP Request Response



The screenshot shows the VS Code editor with the Explorer on the left and the Editor on the right. The Explorer shows the project structure for 'FINALCAPSTONE-MAIN' with files like 'register.component.ts', 'guard', 'auth.guard.spec.ts', 'auth.guard.ts', 'model', 'cart.ts', 'cartItem.ts', 'cuisines.ts', 'user.ts', 'services', 'auth.service.spec.ts', 'auth.service.ts', 'cart.service.ts', 'db.service.spec.ts', 'db.service.ts', 'app-routing.module.ts', and 'app.component.css'. The Editor shows the 'auth.service.ts' file with the following code:

```
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```


5.2. JSON

```
loginUser(datac: any ) {  
    const cdatac = JSON.stringify(datac);  
    this.http.post(BaseUrls.getLoginUrl(BaseUrls.USER_GROUPEURL), JSON.parse(cdatac)).subscribe({  
        next: async({ code, datat, message }: any) => {  
            localStorage.setItem("role", JSON.stringify(datat));  
            localStorage.setItem("IsLogin","true");  
  
            if (datat.role == "Admin") {  
                this.IsAdmin = true;  
                localStorage.setItem("IsAdmin", "true");  
                this.router.navigate(['/customers']).then(()=>location.reload())  
                this.toast.success("Administrator Successful ");  
            }  
            else {  
                localStorage.setItem("IsAdmin", "false")  
                this.router.navigate(['/orders'], { replaceUrl: true }).then(()=>window.location.reload())  
                this.toast.success("Customer Successful ");  
            }  
        }  
    })  
}
```

6.Task 6: Jenkinsfile stages and step declarations

6.1 Jenkins Stages

```
pipeline {
    agent any

    stages {
        stage('Build') {
            steps {
                //Get code from GitHub repository
                git (branch: 'main', url: 'https://github.com/lothoroger/capstone2023fdsfood.git')

                //Run maven wrapper
                bat "mvn compile"

                echo 'Building the Food Delivery Project with Maven compiler'
            }
        }

        stage('Test') {
            steps {
                bat 'mvn test'
                echo 'Testing the Food Delivery project with Maven test'
            }
        }

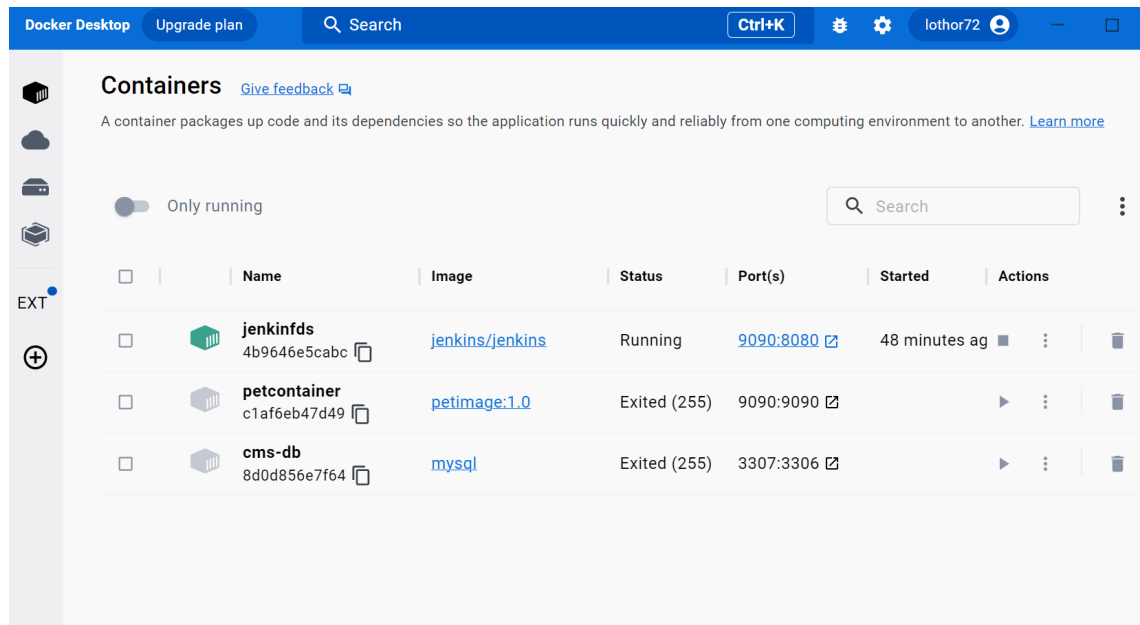
        stage('Deploy') {
            steps {
                bat 'mvn package'
                echo 'Deploy the Food Delivery project with Maven package'
            }
        }
    }
}
```

6.2 Build Success

```
[INFO] --- maven-resources-plugin:3.2.0:resources (default-resources) @ FoodDelivery ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Using 'UTF-8' encoding to copy filtered properties files.
[INFO] skip non existing resourceDirectory C:\ProgramData\Jenkins\workspace\Capstone2023FoodDelivery\src\main\resources
[INFO] skip non existing resourceDirectory C:\ProgramData\Jenkins\workspace\Capstone2023FoodDelivery\src\main\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.10.1:compile (default-compile) @ FoodDelivery ---
[INFO] No sources to compile
[INFO]
[INFO] --- maven-resources-plugin:3.2.0:testResources (default-testResources) @ FoodDelivery ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Using 'UTF-8' encoding to copy filtered properties files.
[INFO] skip non existing resourceDirectory C:\ProgramData\Jenkins\workspace\Capstone2023FoodDelivery\src\test\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.10.1:testCompile (default-testCompile) @ FoodDelivery ---
[INFO] No sources to compile
[INFO]
[INFO] --- maven-surefire-plugin:2.22.2:test (default-test) @ FoodDelivery ---
[INFO] No tests to run.
[INFO]
[INFO] --- maven-jar-plugin:3.2.2:jar (default-jar) @ FoodDelivery ---
[WARNING] JAR will be empty - no content was marked for inclusion!
[INFO] Building jar: C:\ProgramData\Jenkins\workspace\Capstone2023FoodDelivery\target\FoodDelivery-0.0.1-SNAPSHOT.jar
[INFO]
[INFO] --- spring-boot-maven-plugin:2.7.5:repackage (repackage) @ FoodDelivery ---
[INFO] Replacing main artifact with repackaged archive
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 3.318 s
[INFO] Finished at: 2023-02-20T10:08:42-07:00
[INFO]
[Pipeline] echo
Deploy the Food Delivery project with Maven package
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] // pipeline
[Pipeline] end of pipeline
Finished: SUCCESS
```

7.Task 7: Dockerfile and commands to assemble an image

7.1 Docker Jenkins 9090:8080



7.2 Docker image

```
336536@pn2036725 MINGW64 ~/Desktop/caltech/finalcapstone-main (main)
$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
jenkins/jenkins     latest             f16216f97fcb       12 days ago        467MB
jenkins/jenkins     (none)             74f22659b4b6       4 months ago        462MB

336536@pn2036725 MINGW64 ~/Desktop/caltech/finalcapstone-main (main)
$ docker ps
CONTAINER ID   IMAGE                  COMMAND                  CREATED             STATUS
PORTS         NAMES
f321da42909d   74f22659b4b6          "/usr/bin/tini -- /u... 4 months ago       Up 22 minutes
50000/tcp, 0.0.0.0:9090->8080/tcp   jenkins

336536@pn2036725 MINGW64 ~/Desktop/caltech/finalcapstone-main (main)
$ docker run --name=jenkinfds -d -p 9090:8080 jenkins/jenkins
4b9646e5cabc6eff1d1df6af1451fa144efa67f7fba70a243440adf37e306355
docker: Error response from daemon: driver failed programming external connecti
n endpoint jenkins (1f5829cf5561ae82b538c996c4efc778683da2e723cf687c84a996b1e4
4): Bind for 0.0.0.0:9090 failed: port is already allocated.
```

7.3 Docker file script

Configure the project with Docker File
FROM openjdk:11
RUN mkdir /app
COPY target/ /app
WORKDIR /app
CMD java -jar FoodDelivery-0.0.1-SNAPSHOT.jar
--spring.config.name=application.properties

8. Task 8: Jenkins PipeLine Project Creating with git SCM

8.1 Jenkins pipeline script

```
pipeline {  
    agent any  
    stages {  
        stage('Build') {  
            steps {  
                sh 'chmod +x ./mvnw'  
                sh './mvnw compile'  
                echo 'Building the Food Delivery Project with Maven compiler'  
            }  
        }  
  
        stage('Test') {  
            steps {  
                sh './mvnw test'  
                echo 'Testing the Food Delivery project with Maven test'  
            }  
        }  
  
        stage('Deploy') {  
            steps {  
                sh './mvnw package'  
                echo 'Deploy the Food Delivery project with Maven package'  
            }  
        }  
    }  
}
```

8.2 Docker Jenkins 9090:8080 Git SCM

FoodDeliveryJenkins Config [Jenkins] x New Tab

localhost:9090/job/FoodDeliveryJenkins/configure

Getting Started OPEX at LANL - Reports Exit/save edit to sudo... Capstone

Dashboard > FoodDeliveryJenkins > Configuration Pipeline

Configure

- General
- Advanced Project Options
- Pipeline

Definition

Pipeline script from SCM

SCM ?

Git

Repositories ?

Repository URL ?

https://github.com/lothoroger/fdsFinalCapstone.git

Credentials ?

- none -

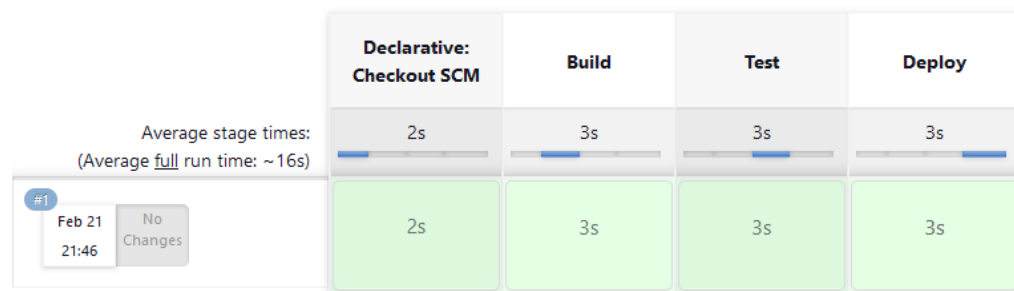
+ Add

8.3 Build Stage

Pipeline FoodDeliveryJenkins

This is the Food Delivery Jenkins Pipeline CI/CD

Stage View

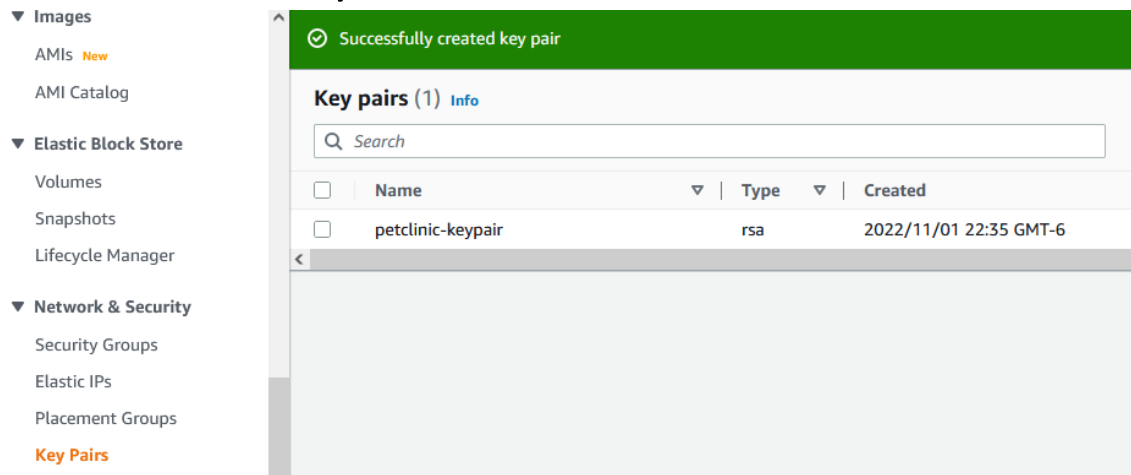


Permalinks

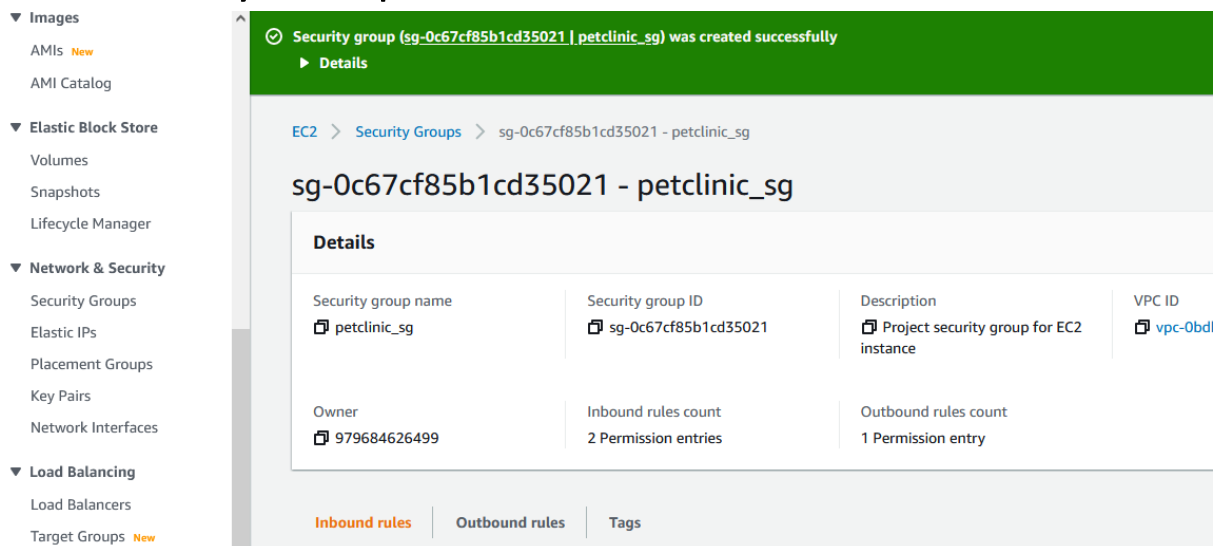
9.Task 9: AWS, EC2, SSH/Cloud Shell Connection, Tool Configuration

9.1 Create and Launch AWS EC2 Instance

9.1.1 Create Key Pair



3.b Security Group



3.c EC2 Instance – PetClinic_Instance

EC2 > Instances > i-0b54333710d140e54	
Instance summary for i-0b54333710d140e54 (PetClinic_instance) info <small>Updated less than a minute ago</small>	
Instance ID i-0b54333710d140e54 (PetClinic_instance)	Public IPv4 address 52.53.182.38 open address
IPv6 address –	Instance state Running
Hostname type IP name: ip-172-31-21-64.us-west-1.compute.internal	Private IP DNS name (IPv4 only) ip-172-31-21-64.us-west-1.compute.internal
Answer private resource DNS name IPv4 (A)	Instance type t2.micro
Auto-assigned IP address 52.53.182.38 [Public IP]	VPC ID vpc-0bdba75e97ca10d70
IAM Role –	Subnet ID subnet-00aa328fc8a29f5fc
Details Security Networking Storage Status checks Monitoring Tags	
▼ Instance details info	
Platform Amazon Linux (Inferred)	AMI ID ami-017c001a88dd93847

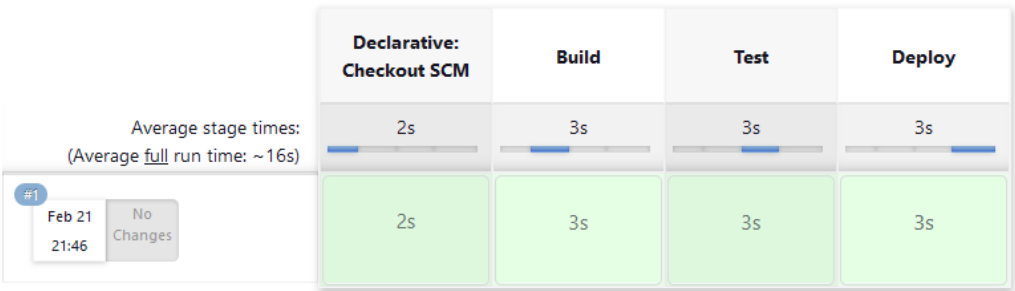
10. Configure EC2 Instance with JDK 11
 - Install jdk 8 and Install jdk11
 - a. Create an Identity andAccessManagement (IAM)
11. Configure EC2 Instance with Docker
 - 5.1 Install the docker on EC2
12. Configure EC2 Instance with Jenkins
 - 6.1 Install the Jenkins on EC2
 - 6.2 Create the Admin User
13. Upload the given code to git repo
14. Create Jenkins Pipeline on EC2 with SCM as git
15. Build the Pipeline to dockerize the application

10. Task 10 Jenkins Pipeline Build

Pipeline FoodDeliveryJenkins

This is the Food Delivery Jenkins Pipeline CI/CD

Stage View



Permalinks