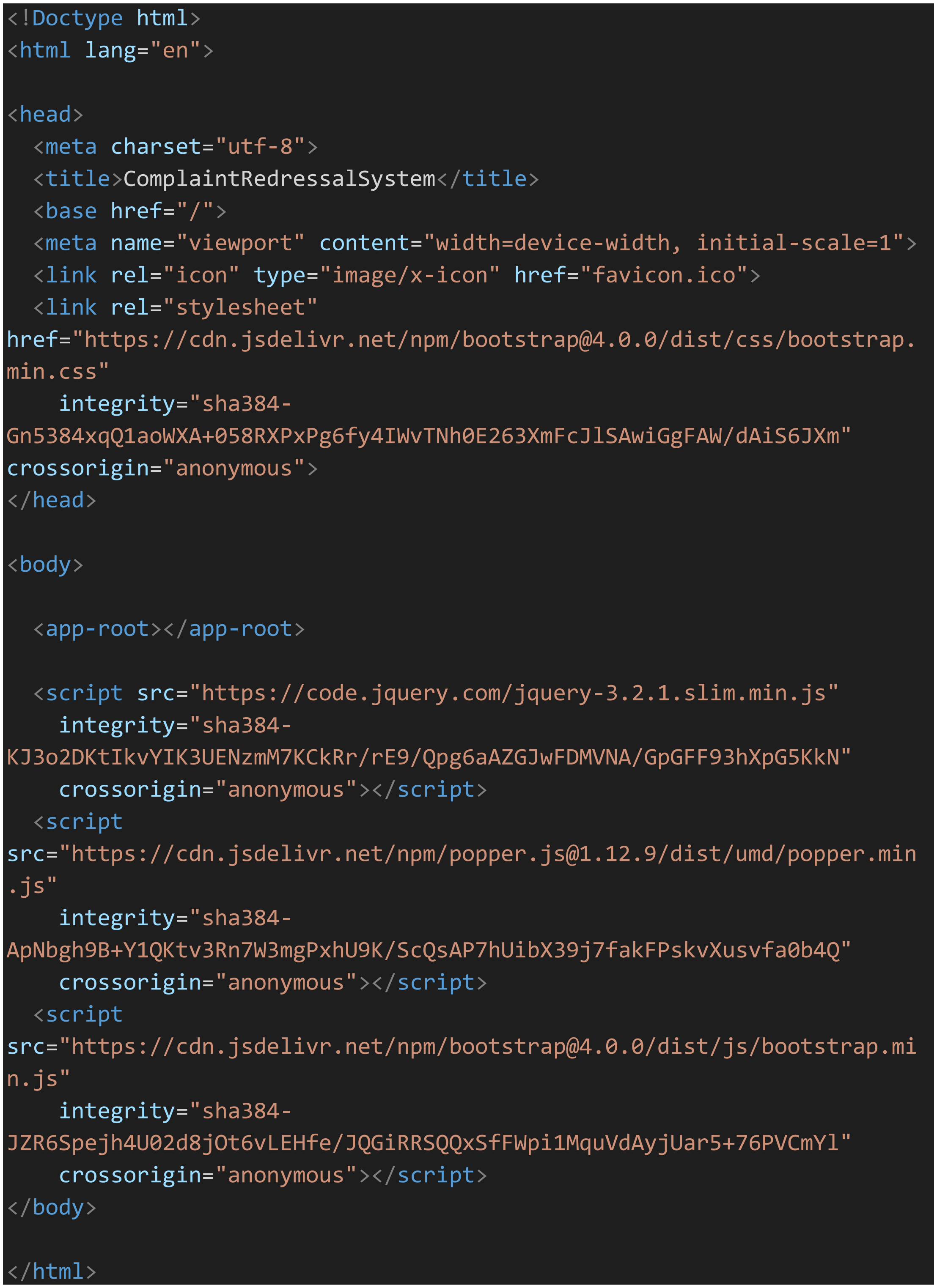
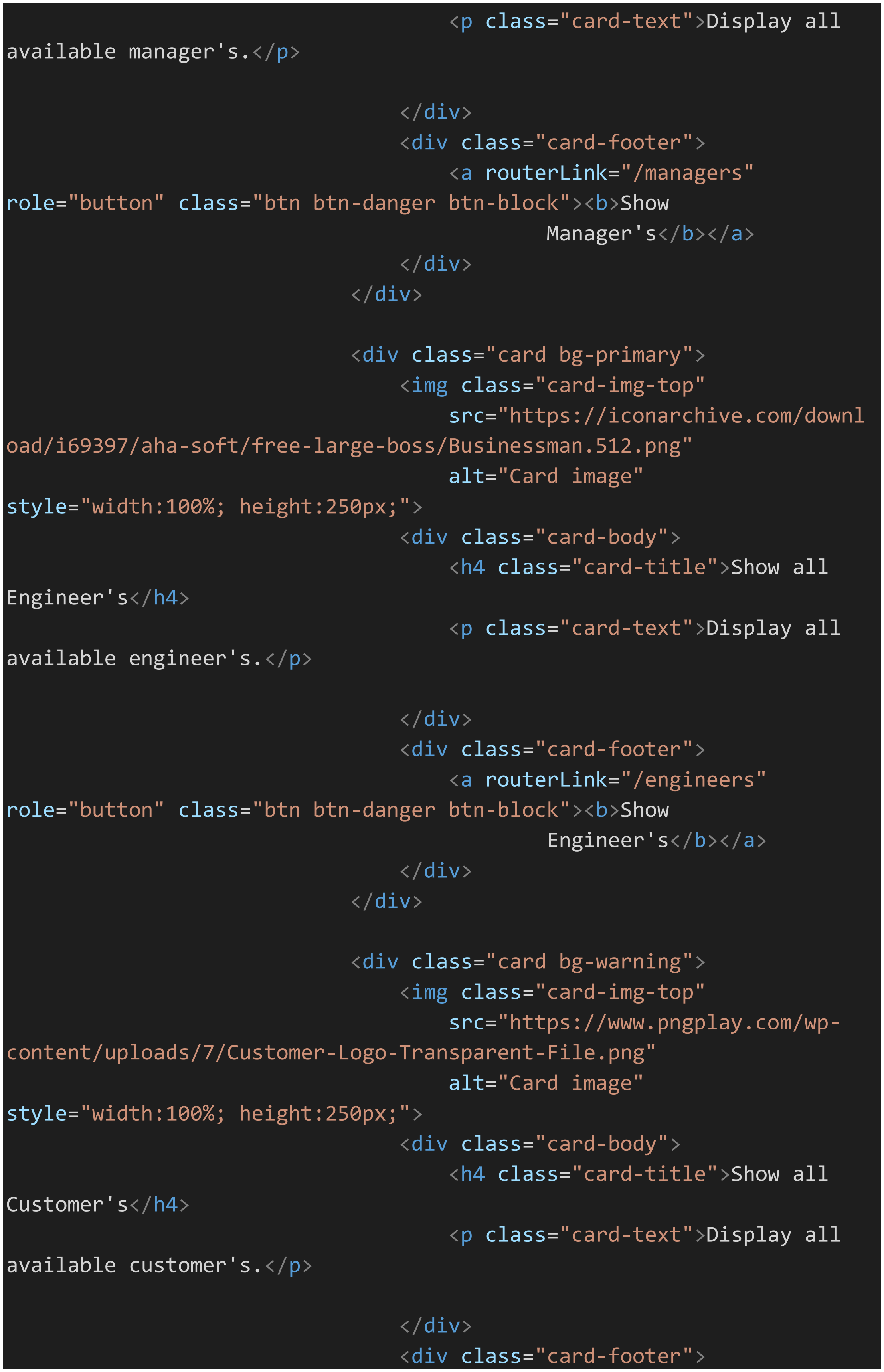
Front-end



```
<!Doctype html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <title>ComplaintRedressalSystem</title>
  <base href="/">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="favicon.ico">
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/css/bootstrap.
min.css"
    integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">
</head>

<body>

  <app-root></app-root>

  <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
    integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
    crossorigin="anonymous"></script>
  <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/dist/umd/popper.min
.js"
    integrity="sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
    crossorigin="anonymous"></script>
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/js/bootstrap.mi
n.js"
    integrity="sha384-
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"
    crossorigin="anonymous"></script>
</body>

</html>
```

Admin

```
<main>
    <div class="container-fluid">
        <div class="row">
            <div class="col">
                <div class="card mx-auto bg-light mb-2 mt-2">
                    <div class="card-body">
                        <h1 class="card-title text-center">Welcome
Admin: {{name}}
                        </h1>
                        <p class="card-text text-center">Love to see
you again <b>{{username}}</b></p>
                        <div class="card-deck">
                            <div class="card bg-info">
                                <img class="card-img-top"
                                    src="https://www.freeiconspng.com/
thumbs/customers-icon/customers-icon-12.png"
                                    alt="Card image"
style="width:100%; height:250px;">
                                <div class="card-body">
                                    <h4 class="card-title">Create a
User</h4>
                                    <p class="card-text">Create new
user like customer, manager & engineer. </p>

                                </div>
                                <div class="card-footer">
                                    <a routerLink="/register-user"
role="button"
                                        class="btn btn-danger btn-
block"><b>Create
                                            User</b>
                                    </a>
                                </div>
                            </div>

                            <div class="card bg-success">
                                <img class="card-img-top"
                                    src="https://icons.iconarchive.com
/icons/aha-soft/free-large-boss/512/Manager-icon.png"
                                    alt="Card image"
style="width:100%; height:250px;">
                                <div class="card-body">
                                    <h4 class="card-title">Show all
Manager's</h4>
```



```
                                    <p class="card-text">Display all
available manager's.</p>

                                </div>
                                <div class="card-footer">
                                    <a routerLink="/managers"
role="button" class="btn btn-danger btn-block"><b>Show
                                            Manager's</b></a>
                                </div>
                            </div>

                            <div class="card bg-primary">
                                <img class="card-img-top"
                                    src="https://iconarchive.com/downl
oad/i69397/aha-soft/free-large-boss/Businessman.512.png"
                                    alt="Card image"
style="width:100%; height:250px;">
                                <div class="card-body">
                                    <h4 class="card-title">Show all
Engineer's</h4>
                                    <p class="card-text">Display all
available engineer's.</p>

                                </div>
                                <div class="card-footer">
                                    <a routerLink="/engineers"
role="button" class="btn btn-danger btn-block"><b>Show
                                            Engineer's</b></a>
                                </div>
                            </div>

                            <div class="card bg-warning">
                                <img class="card-img-top"
                                    src="https://www.pngplay.com/wp-
content/uploads/7/Customer-Logo-Transparent-File.png"
                                    alt="Card image"
style="width:100%; height:250px;">
                                <div class="card-body">
                                    <h4 class="card-title">Show all
Customer's</h4>
                                    <p class="card-text">Display all
available customer's.</p>

                                </div>
                                <div class="card-footer">
```

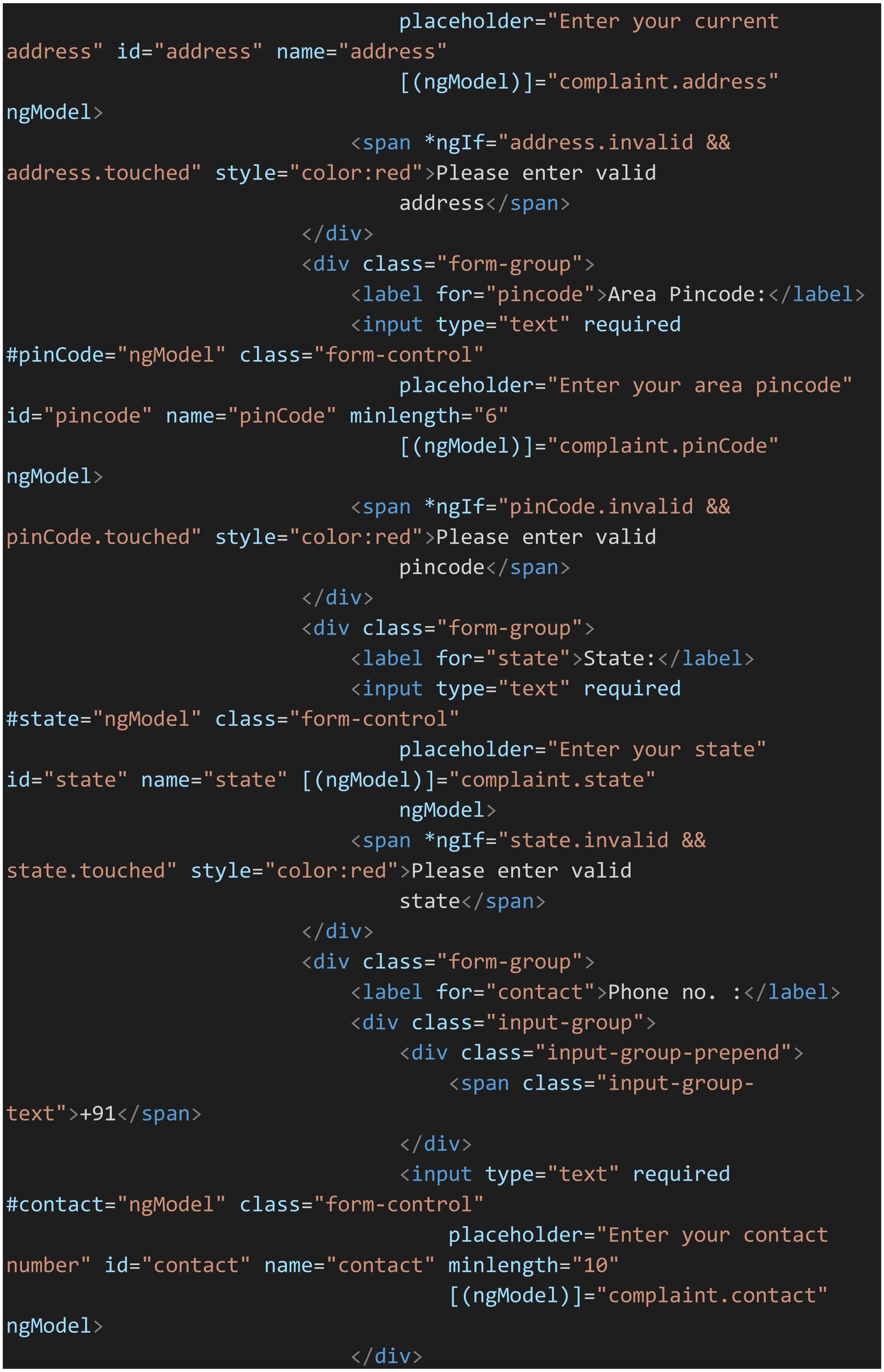
```
                                    <a routerLink="/customers"
role="button" class="btn btn-danger btn-block"><b>Show
                                            Customer's</b></a>
                                </div>
                            </div>

                        </div>
                    </div>
                </div>
            </div>
        </div>

    </div>
</main>
<div class="container">
    <div class="row">
        <div class="col-6 mx-auto mt-3 mb-4">
            <div class="card">
                <div class="card-header">
                    <h4 class="text-center">User Registration</h4>
                </div>
                <div class="card-body">
                    <form #registerationForm="ngForm"
(ngSubmit)="onSubmit()">

                        <label>Choose Role: </label>&nbsp;&nbsp;&nbsp;

                        <div class="form-check-inline">
                            <label class="form-check-label">
                                <input type="radio" class="form-check-
input" name="optradio" [(ngModel)]="user.roleName"
                                    value="CUSTOMER">Customer
                            </label>
                        </div>
                        <div class="form-check-inline">
                            <label class="form-check-label">
                                <input type="radio" class="form-check-
input" name="optradio" [(ngModel)]="user.roleName"
                                    value="MANAGER">Manager
                            </label>
                        </div>
                        <div class="form-check-inline">
                            <label class="form-check-label">
                                <input type="radio" class="form-check-
input" name="optradio" [(ngModel)]="user.roleName"
```

```
                                    value="ENGINEER">Engineer
                            </label>
                        </div>
                        <hr>
                        <div class="form-group">
                            <label for="username">Username:</label>
                            <input type="text" class="form-control"
required #username="ngModel"
                                placeholder="Enter Username"
id="username" name="username" [(ngModel)]="user.username"
                                ngModel>
                            <span *ngIf="username.invalid &&
username.touched" style="color: red">Please enter a valid
                                username.</span>

                        </div>
                        <div class="form-group">
                            <label for="pwd">Password:</label>
                            <input type="password" required
#password="ngModel" class="form-control"
                                placeholder="Enter password" id="pwd"
minlength="8" name="password"
                                [(ngModel)]="user.password" ngModel>
                            <span *ngIf="password.invalid &&
password.touched" style="color: red">Please enter valid
                                password of minimum 8
characters.</span>
                        </div>
                        <div class="form-group">
                            <label for="fname">First Name:</label>
                            <input type="text" class="form-control"
required #firstName="ngModel"
                                placeholder="Enter your first name"
id="fname" name="firstName"
                                [(ngModel)]="user.firstName" ngModel>
                            <span *ngIf="firstName.invalid &&
firstName.touched" style="color:red">Please enter valid
                                first name</span>
                        </div>
                        <div class="form-group">
                            <label for="lname">Last Name:</label>
                            <input type="text" required
#lastName="ngModel" class="form-control"
                                placeholder="Enter your last name"
id="lname" name="lastName"
```

```
                                [(ngModel)]="user.lastName" ngModel>
                            <span *ngIf="lastName.invalid &&
lastName.touched" style="color: red">Please enter valid
                                last name</span>
                        </div>
                        <div class="form-group">
                            <label for="pincode">Area Pincode:</label>
                            <input type="text" required
#pinCode="ngModel" class="form-control"
                                placeholder="Enter your area pincode"
id="pincode" name="pinCode" minlength="6"
                                [(ngModel)]="user.pinCode" ngModel>
                            <span *ngIf="pinCode.invalid &&
pinCode.touched" style="color:red">Please enter valid
                                pincode</span>
                        </div>
                        <div class="form-group">
                            <label for="email">Email:</label>
                            <input type="email" required
#email="ngModel" class="form-control"
                                placeholder="Enter your email address"
id="email" name="email" [(ngModel)]="user.email"
                                ngModel>
                            <span *ngIf="email.invalid &&
email.touched" style="color:red">Please enter valid email
                                address</span>
                        </div>
                        <div class="form-group">
                            <label for="phn">Phone no. :</label>
                            <div class="input-group">
                                <div class="input-group-prepend">
                                    <span class="input-group-
text">+91</span>
                                </div>
                                <input type="text" required
#phone="ngModel" class="form-control"
                                    placeholder="Enter your phone
number" id="phn" name="phone" minlength="10"
                                    [(ngModel)]="user.phone" ngModel>
                            </div>
                            <span *ngIf="phone.invalid &&
phone.touched" style="color:red">Please enter valid phone
                                number</span>
                        </div>
                        <div class="text-center">
                            <div class="btn-group">
                                <button type="submit" class="btn btn-
primary" [disabled]="registerationForm.invalid"
                                    data-toggle="modal" data-
target="#myModal">Register</button>
                                <!-- The Modal -->
                                <div class="modal fade" id="myModal">
                                    <div class="modal-dialog modal-
dialog-centered">
                                        <div class="modal-content">

                                            <!-- Modal Header -->
                                            <div class="modal-header">
                                                <h4 class="modal-
title">User Registration</h4>
                                                <button type="button"
class="close"
                                                    data-
dismiss="modal">&times;</button>
                                            </div>

                                            <!-- Modal body -->
                                            <div class="modal-body">
                                                <span *ngIf="isValid"
style="color:green">{{message}}</span>
                                                <span *ngIf="!isValid"
style="color: red">{{message}}</span>
                                            </div>

                                            <!-- Modal footer -->
                                            <div class="modal-footer">
                                                <button type="button"
class="btn btn-secondary"
                                                    data-
dismiss="modal">Close</button>
                                            </div>

                                        </div>
                                    </div>
                                </div>
                                <button type="reset" class="btn btn-
danger">Reset</button>
                            </div>
                        </div>
```

```
                    </form>
                </div>
            </div>


        </div>
    </div>
</div>
<div class="container">
    <div class="row">
        <div class="col-8 mx-auto mt-3 mb-4">
            <div class="card">
                <div class="card-header bg-info">
                    <h3 class="text-center">Create a new
Complaint</h3>
                    <p class="text-center">Kindly enter your
details</p>
                </div>
                <div class="card-body">
                    <form #registerationForm="ngForm"
(ngSubmit)="onSubmit()">
                        <div class="form-group">
                            <label for="username">Username:</label>
                            <input type="text" class="form-control"
required #username="ngModel"
                                placeholder="Enter Username"
id="username" name="username"
                                [(ngModel)]="complaint.username"
ngModel readonly>
                            <span *ngIf="username.invalid &&
username.touched" style="color: red">Please enter a valid
                                username.</span>
                        </div>
                        <div class="form-group">
                            <label
for="newcomplaint">Complaint:</label>
                            <select class="form-control"
id="newcomplaint" name="complaint"
                                [(ngModel)]="complaint.complaint"
ngModel required #newcomplaint="ngModel">
                                <option selected disabled>Select
Complaint</option>
                                <option value="Unable to make a
call">Unable to make a call</option>
```

```
                                <option value="Unable to receive a
call">Unable to receive a call</option>
                                <option value="Phone dead">Phone
Dead</option>
                                <option value="Noisy voice">Noisy
Voice</option>
                                <option value="Slow broadband
speed">Slow broadband speed</option>
                                <option value="No signal">No
signal</option>
                            </select>
                            <span *ngIf="newcomplaint.invalid &&
newcomplaint.touched" style="color: red">Please select
                                a
                                valid complaint.</span>
                        </div>
                        <div class="form-group">
                            <label for="fname">First Name:</label>
                            <input type="text" class="form-control"
required #firstName="ngModel"
                                placeholder="Enter your first name"
id="fname" name="firstName"
                                [(ngModel)]="complaint.firstName"
ngModel>
                            <span *ngIf="firstName.invalid &&
firstName.touched" style="color:red">Please enter valid
                                first name</span>
                        </div>
                        <div class="form-group">
                            <label for="lname">Last Name:</label>
                            <input type="text" required
#lastName="ngModel" class="form-control"
                                placeholder="Enter your last name"
id="lname" name="lastName"
                                [(ngModel)]="complaint.lastName"
ngModel>
                            <span *ngIf="lastName.invalid &&
lastName.touched" style="color: red">Please enter valid
                                last name</span>
                        </div>
                        <div class="form-group">
                            <label for="address">Current
Address:</label>
                            <input type="text" required
#address="ngModel" class="form-control"
```



```
                                placeholder="Enter your current
address" id="address" name="address"
                                [(ngModel)]="complaint.address"
ngModel>
                            <span *ngIf="address.invalid &&
address.touched" style="color:red">Please enter valid
                                address</span>
                        </div>
                        <div class="form-group">
                            <label for="pincode">Area Pincode:</label>
                            <input type="text" required
#pinCode="ngModel" class="form-control"
                                placeholder="Enter your area pincode"
id="pincode" name="pinCode" minlength="6"
                                [(ngModel)]="complaint.pinCode"
ngModel>
                            <span *ngIf="pinCode.invalid &&
pinCode.touched" style="color:red">Please enter valid
                                pincode</span>
                        </div>
                        <div class="form-group">
                            <label for="state">State:</label>
                            <input type="text" required
#state="ngModel" class="form-control"
                                placeholder="Enter your state"
id="state" name="state" [(ngModel)]="complaint.state"
                                ngModel>
                            <span *ngIf="state.invalid &&
state.touched" style="color:red">Please enter valid
                                state</span>
                        </div>
                        <div class="form-group">
                            <label for="contact">Phone no. :</label>
                            <div class="input-group">
                                <div class="input-group-prepend">
                                    <span class="input-group-
text">+91</span>
                                </div>
                                <input type="text" required
#contact="ngModel" class="form-control"
                                    placeholder="Enter your contact
number" id="contact" name="contact" minlength="10"
                                    [(ngModel)]="complaint.contact"
ngModel>
                            </div>
```

```
              <span *ngIf="contact.invalid &&
contact.touched" style="color:red">Please enter valid phone
                number</span>
            </div>

            <div class="text-center">

              <button type="submit" class="btn btn-
danger btn-block"
                [disabled]="registerationForm.invalid"
data-toggle="modal"
                data-target="#myModal">Register
Complaint</button>
              <!-- The Modal -->
              <div class="modal fade" id="myModal">
                <div class="modal-dialog modal-dialog-
centered">
                  <div class="modal-content">

                    <!-- Modal Header -->
                    <div class="modal-header">
                      <h4 class="modal-
title">ABC Telecom Complaint Registration</h4>
                      <button type="button"
class="close" data-dismiss="modal">&times;</button>
                    </div>

                    <!-- Modal body -->
                    <div class="modal-body">
                      <h5 *ngIf="isValid"
style="color:green">{{message}}</h5>

                      <div *ngIf="isValid"
class="text-center">
                        <hr>
                        <p><b>Complaint Id:
</b>{{complaint.id}}</p>
                        <p><b>Complaint: </b>
{{complaint.complaint}}</p>
                        <p><b>Status:
</b><span style="color: red">{{complaint.status}}</span>
                        </p>
                        <p><b>Customer Name:
</b>{{complaint.firstName}} {{complaint.lastName}}
                        </p>
```

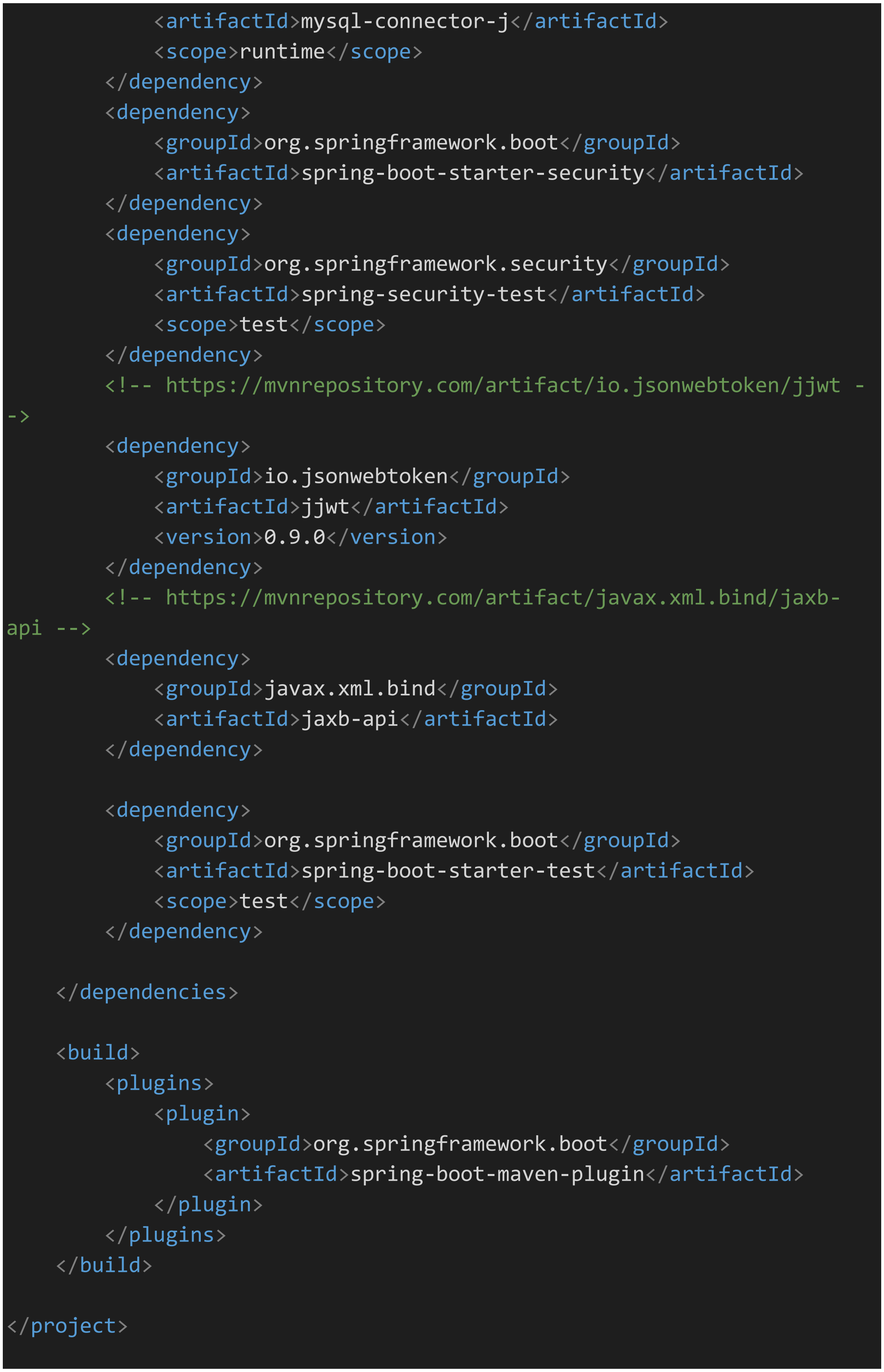
```
                                                <p><b>Username:
</b>{{complaint.username}}</p>
                                                <p><b>Current Address:
</b>{{complaint.address}}, {{complaint.state}}
                                                </p>
                                                <p><b>Pincode:
</b>{{complaint.pinCode}}</p>
                                                <p><b>Contact Number:
</b>{{complaint.contact}}</p>
                                            </div>
                                            <span *ngIf="!isValid"
style="color: red">{{message}}</span>
                                        </div>

                                        <!-- Modal footer -->
                                        <div class="modal-footer">
                                            <button type="button"
class="btn btn-secondary" data-dismiss="modal"
                                                (click)="onClick()">Cl
ose</button>
                                        </div>

                                    </div>
                                </div>
                            </div>

                        </div>
                    </form>
                </div>
            </div>


        </div>
    </div>
</div>
<div class="container">
    <div class="row">
        <div class="col-lg-4 col-md-8 mx-auto">
            <div class="card mt-1 bg-warning">
                <img class="card-img-top"
src="https://www.w3schools.com/bootstrap4/img_avatar1.png" alt="Card
image"
                    style="width:100%; height:250px;">
                <div class="card-body">
                    <div class="text-center">
```

```
                        <h4 class="card-title">Welcome User</h4>
                        <p>Sign in to continue.</p>
                    </div>
                    <hr>
                    <form #login="ngForm" (ngSubmit)="onSubmit()">
                        <div class="form-group">
                            <label
for="email"><b>Username:</b></label> <input type="text"
name="username"
                                #username="ngModel" class="form-
control" placeholder="Enter Username" id="email"
                                [(ngModel)]="credentials.username"
required ngModel>
                            <span *ngIf="username.invalid &&
username.touched" style="color:red">Please enter a valid
                                username.</span>
                        </div>
                        <div class="form-group">
                            <label for="pwd"><b>Password:</b></label>
<input type="password" name="password"
                                #password="ngModel" minlength="8"
class="form-control" placeholder="Enter password"
                                id="pwd"
[(ngModel)]="credentials.password" required ngModel>
                            <span *ngIf="password.invalid &&
password.touched" style="color: red">Please enter a valid
                                password.</span>
                        </div>
                        <button type="submit" class="btn btn-primary
btn-block"
                            [disabled]="login.invalid">Login</button>
                    </form>
                </div>
            </div>

        </div>
    </div>
</div>
```

Refer github link fr more source code:

Backend

```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.7.7</version>
        <relativePath /> <!-- lookup parent from repository -->
    </parent>
    <groupId>com.abc.telecom.ltd</groupId>
    <artifactId>Phase5_Complaint_Redressal_System</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>Phase5_Complaint_Redressal_System</name>
    <description>Phase5 CRS backend project</description>
    <properties>
        <java.version>17</java.version>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-data-jpa</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-validation</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-devtools</artifactId>
            <scope>runtime</scope>
            <optional>true</optional>
        </dependency>
        <dependency>
            <groupId>com.mysql</groupId>
```

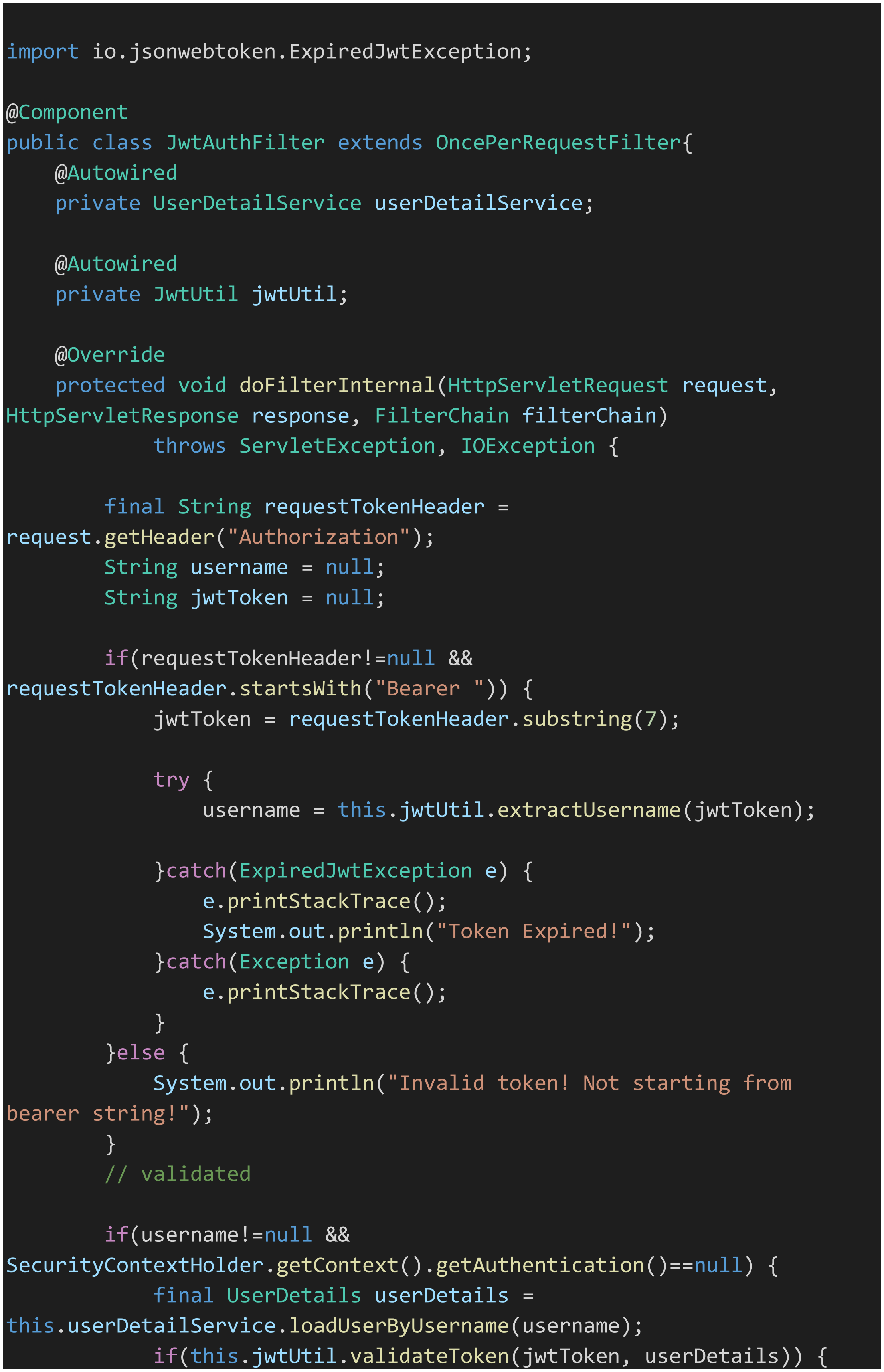


```
            <artifactId>mysql-connector-j</artifactId>
            <scope>runtime</scope>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-security</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.security</groupId>
            <artifactId>spring-security-test</artifactId>
            <scope>test</scope>
        </dependency>
        <!-- https://mvnrepository.com/artifact/io.jsonwebtoken/jjwt -
->
        <dependency>
            <groupId>io.jsonwebtoken</groupId>
            <artifactId>jjwt</artifactId>
            <version>0.9.0</version>
        </dependency>
        <!-- https://mvnrepository.com/artifact/javax.xml.bind/jaxb-
api -->
        <dependency>
            <groupId>javax.xml.bind</groupId>
            <artifactId>jaxb-api</artifactId>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>

    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
    </build>

</project>
```

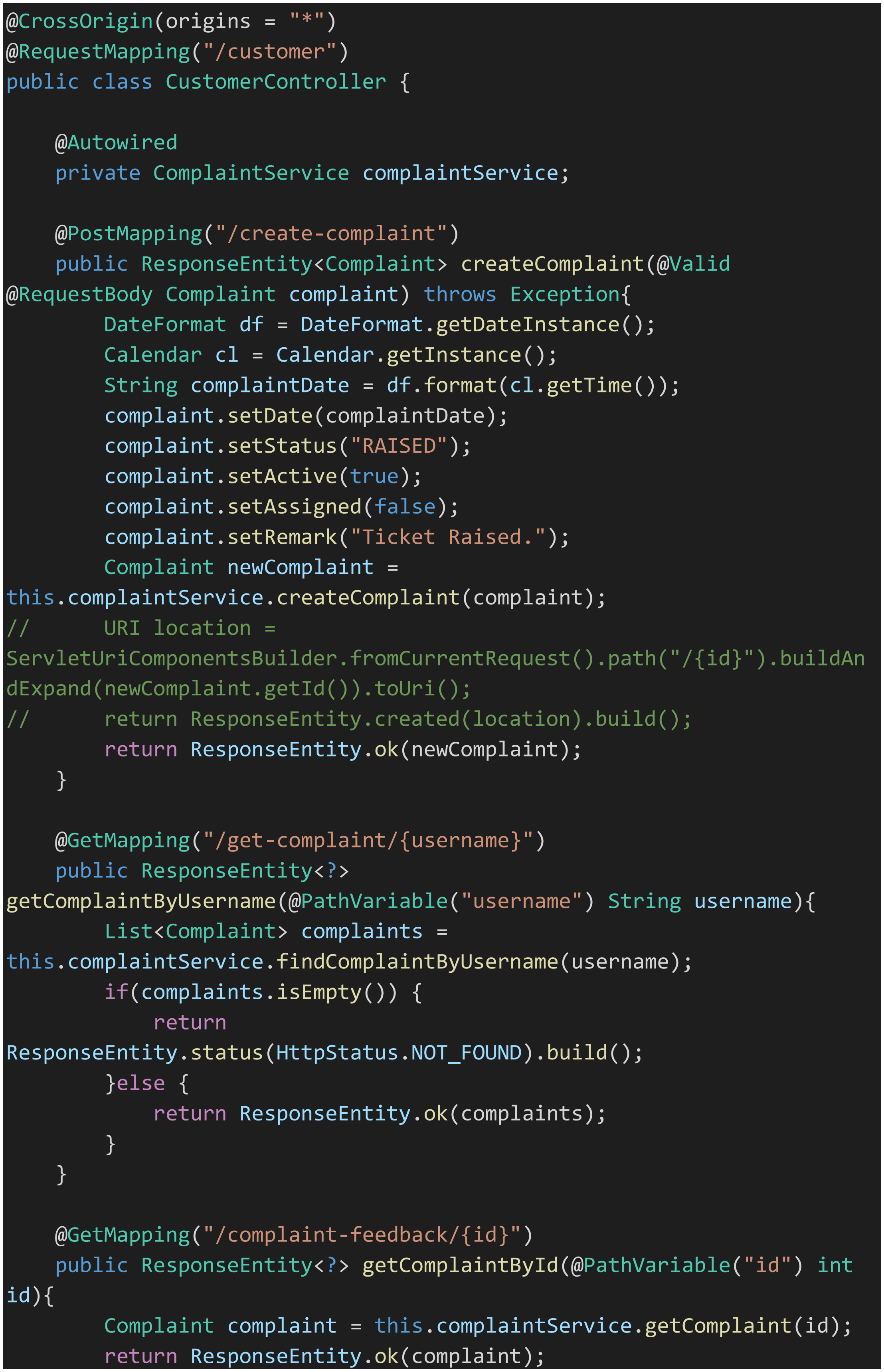
```
ackage com.crs.config;

import java.io.IOException;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.springframework.security.core.AuthenticationException;
import org.springframework.security.web.AuthenticationEntryPoint;
import org.springframework.stereotype.Component;

@Component
public class AuthEntryPoint implements AuthenticationEntryPoint{

    @Override
    public void commence(HttpServletRequest request,
HttpServletResponse response,
            AuthenticationException authException) throws IOException,
ServletException {
        response.sendError(HttpServletResponse.SC_UNAUTHORIZED,
"Unauthorized");
    }

}
```

```
package com.crs.config;

import java.io.IOException;
import java.util.Iterator;
import java.util.LinkedList;
import java.util.List;

import org.springframework.security.core.GrantedAuthority;
import
org.springframework.security.core.authority.SimpleGrantedAuthority;

import com.fasterxml.jackson.core.JacksonException;
import com.fasterxml.jackson.core.JsonParser;
import com.fasterxml.jackson.databind.DeserializationContext;
import com.fasterxml.jackson.databind.JsonDeserializer;
import com.fasterxml.jackson.databind.JsonNode;
import com.fasterxml.jackson.databind.ObjectMapper;
```

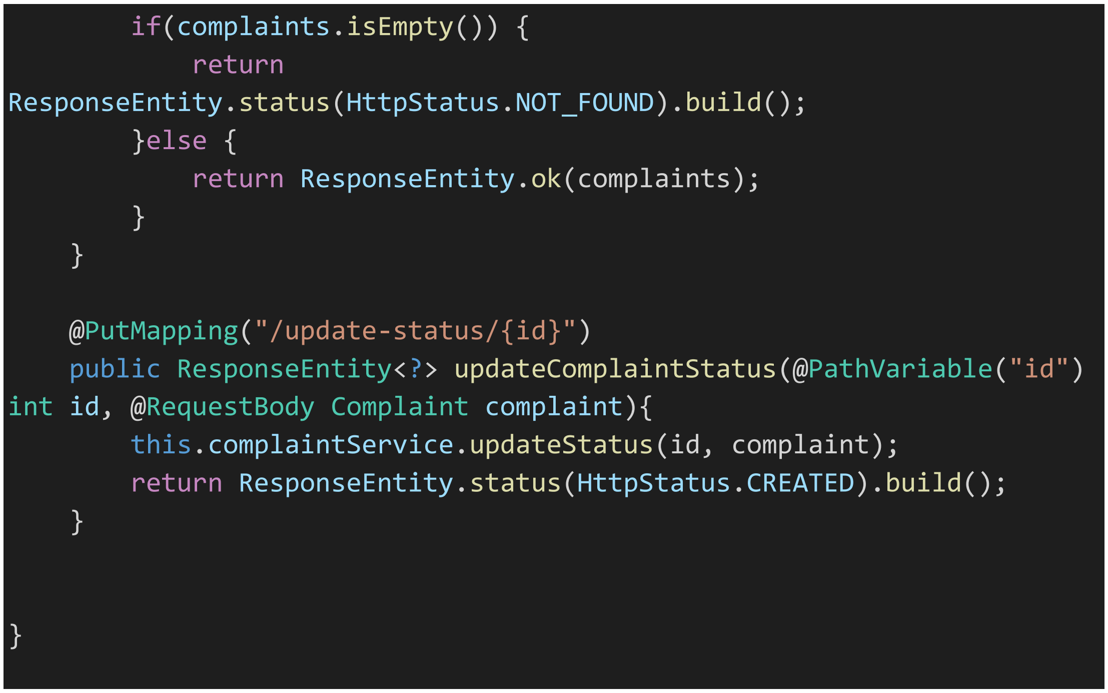
```
public class CustomAuthorityDeserializer extends
JsonDeserializer<Object>{

    @Override
    public Object deserialize(JsonParser jp, DeserializationContext
ctxt) throws IOException, JacksonException {
        ObjectMapper mapper = (ObjectMapper) jp.getCodec();
        JsonNode jsonNode = mapper.readTree(jp);
        List<GrantedAuthority> grantedAuthorities = new
LinkedList<>();

        Iterator<JsonNode> elements = jsonNode.elements();
        while (elements.hasNext()) {
            JsonNode next = elements.next();
            JsonNode authority = next.get("authority");
            grantedAuthorities.add(new
SimpleGrantedAuthority(authority.asText()));
        }
        return grantedAuthorities;
    }

}

package com.crs.config;

import java.io.IOException;

import javax.servlet.FilterChain;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.springframework.beans.factory.annotation.Autowired;
import
org.springframework.security.authentication.UsernamePasswordAuthentica
tionToken;
import
org.springframework.security.core.context.SecurityContextHolder;
import org.springframework.security.core.userdetails.UserDetails;
import
org.springframework.security.web.authentication.WebAuthenticationDetai
lsSource;
import org.springframework.stereotype.Component;
import org.springframework.web.filter.OncePerRequestFilter;

import com.crs.service.UserDetailService;
```



```
import io.jsonwebtoken.ExpiredJwtException;

@Component
public class JwtAuthFilter extends OncePerRequestFilter{
    @Autowired
    private UserDetailService userDetailService;

    @Autowired
    private JwtUtil jwtUtil;

    @Override
    protected void doFilterInternal(HttpServletRequest request,
HttpServletResponse response, FilterChain filterChain)
            throws ServletException, IOException {

        final String requestTokenHeader =
request.getHeader("Authorization");
        String username = null;
        String jwtToken = null;

        if(requestTokenHeader!=null &&
requestTokenHeader.startsWith("Bearer ")) {
            jwtToken = requestTokenHeader.substring(7);

            try {
                username = this.jwtUtil.extractUsername(jwtToken);

            }catch(ExpiredJwtException e) {
                e.printStackTrace();
                System.out.println("Token Expired!");
            }catch(Exception e) {
                e.printStackTrace();
            }
        }else {
            System.out.println("Invalid token! Not starting from
bearer string!");
        }
        // validated

        if(username!=null &&
SecurityContextHolder.getContext().getAuthentication()==null) {
            final UserDetails userDetails =
this.userDetailService.loadUserByUsername(username);
            if(this.jwtUtil.validateToken(jwtToken, userDetails)) {
                //token is valid
                UsernamePasswordAuthenticationToken
usernamePasswordAuthenticationToken = new
UsernamePasswordAuthenticationToken(userDetails,null,userDetails.getAu
thorities());
                usernamePasswordAuthenticationToken.setDetails(new
WebAuthenticationDetailsSource().buildDetails(request));
                SecurityContextHolder.getContext().setAuthentication(u
sernamePasswordAuthenticationToken);
            }
        }else {
            System.out.println("Token is not valid! Please generate a
new token!");
        }

        filterChain.doFilter(request, response);

    }

}

package com.crs.controller;

import java.text.DateFormat;
import java.util.Calendar;
import java.util.List;

import javax.validation.Valid;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.CrossOrigin;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import com.crs.entities.Complaint;
import com.crs.entities.Feedback;
import com.crs.service.ComplaintService;

@RestController
```



```
@CrossOrigin(origins = "*")
@RequestMapping("/customer")
public class CustomerController {

    @Autowired
    private ComplaintService complaintService;

    @PostMapping("/create-complaint")
    public ResponseEntity<Complaint> createComplaint(@Valid
@RequestBody Complaint complaint) throws Exception{
        DateFormat df = DateFormat.getDateInstance();
        Calendar cl = Calendar.getInstance();
        String complaintDate = df.format(cl.getTime());
        complaint.setDate(complaintDate);
        complaint.setStatus("RAISED");
        complaint.setActive(true);
        complaint.setAssigned(false);
        complaint.setRemark("Ticket Raised.");
        Complaint newComplaint =
this.complaintService.createComplaint(complaint);
//      URI location =
ServletUriComponentsBuilder.fromCurrentRequest().path("/{id}").buildAn
dExpand(newComplaint.getId()).toUri();
//      return ResponseEntity.created(location).build();
        return ResponseEntity.ok(newComplaint);
    }

    @GetMapping("/get-complaint/{username}")
    public ResponseEntity<?>
getComplaintByUsername(@PathVariable("username") String username){
        List<Complaint> complaints =
this.complaintService.findComplaintByUsername(username);
        if(complaints.isEmpty()) {
            return
ResponseEntity.status(HttpStatus.NOT_FOUND).build();
        }else {
            return ResponseEntity.ok(complaints);
        }
    }

    @GetMapping("/complaint-feedback/{id}")
    public ResponseEntity<?> getComplaintById(@PathVariable("id") int
id){
        Complaint complaint = this.complaintService.getComplaint(id);
        return ResponseEntity.ok(complaint);
    }

    @PostMapping("/save-feedback")
    public ResponseEntity<?> saveFeedback(@RequestBody Feedback
feedback) throws Exception{
        Feedback savedFeedback =
this.complaintService.saveFeedback(feedback);
        return ResponseEntity.ok(savedFeedback);
    }


}

package com.crs.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.CrossOrigin;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PutMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import com.crs.entities.Complaint;
import com.crs.service.ComplaintService;

@RestController
@CrossOrigin(origins = "*")
@RequestMapping("/engineer")
public class EngineerController {

    @Autowired
    private ComplaintService complaintService;

    @GetMapping("/get-all-complaints/{assignedEngineer}")
    public ResponseEntity<?>
getAssignedComplaints(@PathVariable("assignedEngineer") String
assignedEngineer){
        List<Complaint> complaints =
this.complaintService.assignedComplaints(assignedEngineer);
```



```
        if(complaints.isEmpty()) {
            return
ResponseEntity.status(HttpStatus.NOT_FOUND).build();
        }else {
            return ResponseEntity.ok(complaints);
        }
    }

    @PutMapping("/update-status/{id}")
    public ResponseEntity<?> updateComplaintStatus(@PathVariable("id")
int id, @RequestBody Complaint complaint){
        this.complaintService.updateStatus(id, complaint);
        return ResponseEntity.status(HttpStatus.CREATED).build();
    }


}
```

Refer the full source code on github: https://github.com/shubham-pawar-08/Complaint-Redressal-System.git