

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-KJ3o2DKtIkY1K3UENzmM7KCKr/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.min.js"
    integrity="sha384-JZR6Spejh4U02d8j0t6vLEHfe/JQGiRRSQxSfFWpi1MquVdAyjUar5+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">
                
                <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>
            <div class="card bg-success">
                
    <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">  
          
        <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">  
              
            <div class="card-body">
```

```
        <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]="registrationForm.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>
```



```

        [(ngModel)]="complaint.username"
ngModel readonly>
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>
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>
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"
```



```
<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>
</p><b>{{complaint.id}}</b>
            <p><b>Complaint:</b>
{{complaint.complaint}}</p>
            <p><b>Status:</b><span
style="color: red">{{complaint.status}}</span>
            </p>
            <p><b>Customer Name:</b>
</p><b>{{complaint.firstName}} {{complaint.lastName}}</b>
            </p>
            <p><b>Username:</b>
</p><b>{{complaint.username}}</b>
            </p>
            <p><b>Current Address:</b>
</p><b>{{complaint.address}}, {{complaint.state}}</b>
            </p>
            <p><b>Pincode:</b>
</p><b>{{complaint.pinCode}}</b>
            </p>
            <p><b>Contact Number:</b>
</p><b>{{complaint.contact}}</b>
            </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"
```



```

        <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 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>

package 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.UsernamePasswordAuthenticationToken;
import org.springframework.security.core.context.SecurityContextHolder;
import org.springframework.security.core.userdetails.UserDetails;
import
org.springframework.security.web.authentication.WebAuthenticationDetailsSource;
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 userDetailsService;

    @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.getAut
horities());
            usernamePasswordAuthenticationToken.setDetails(new
WebAuthenticationDetailsSource().buildDetails(request));
        }
    }
}
```

```
        SecurityContextHolder.getContext().setAuthentication(usernamePasswordAuthenticationToken);
    }
} 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}").buildAnd
Expand(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:

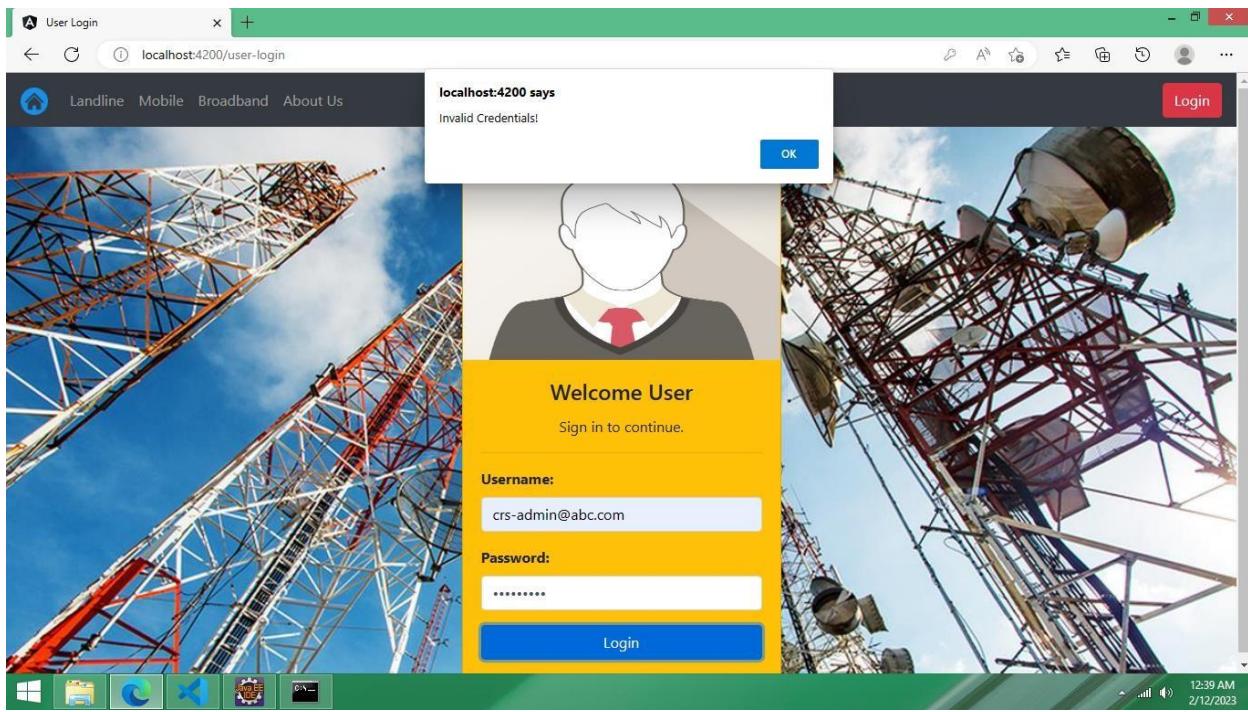
<https://github.com/shubham-pawar-08/Complaint-Redressal-System.git>

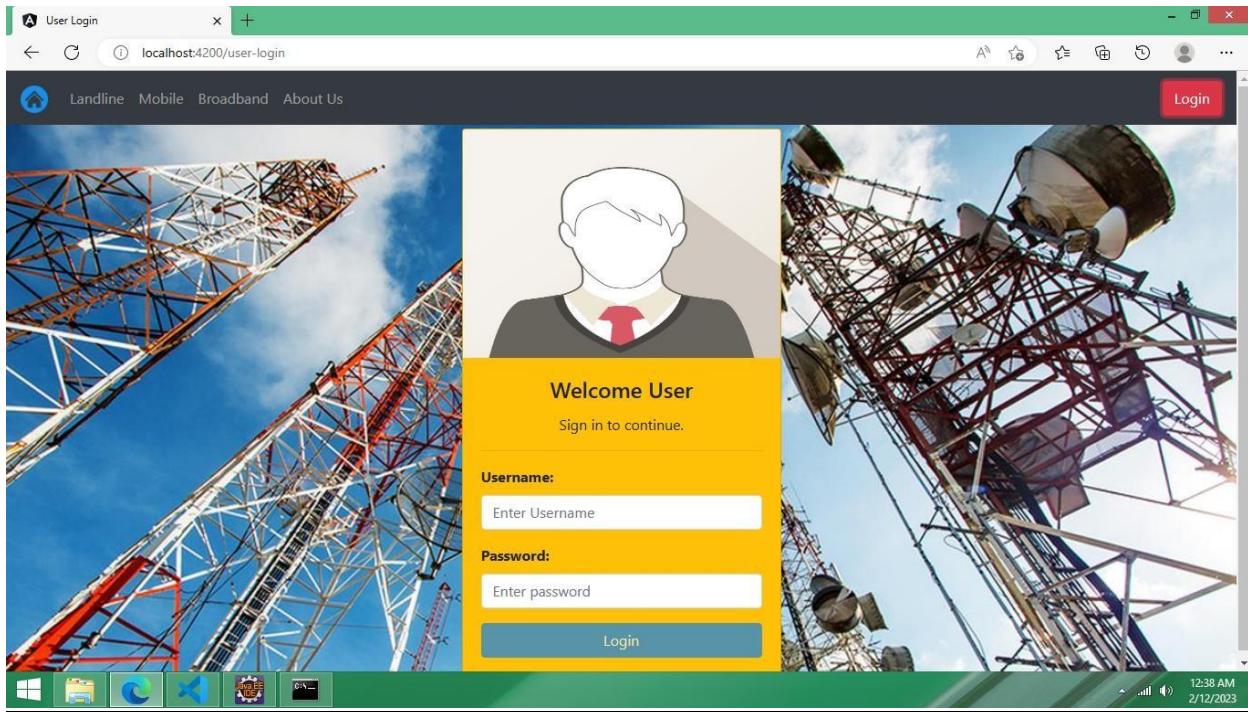
1. Complaint Redressal System HomePage:





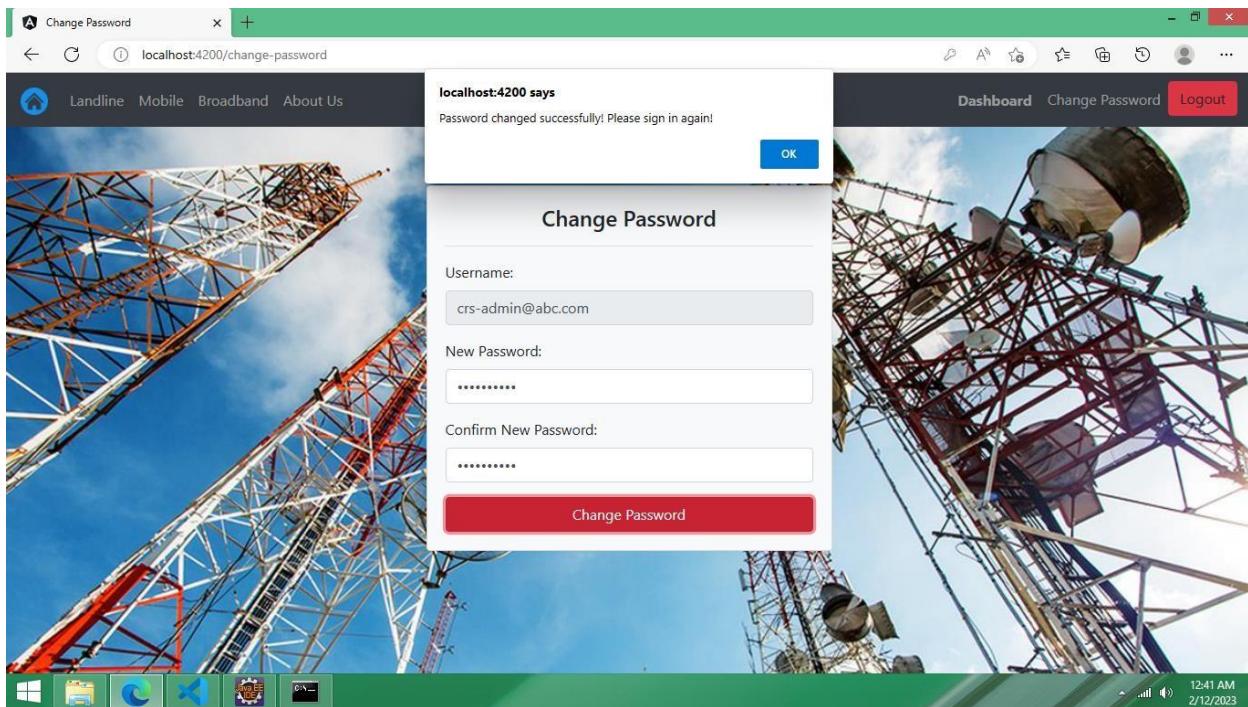
2. On Clicking Login:

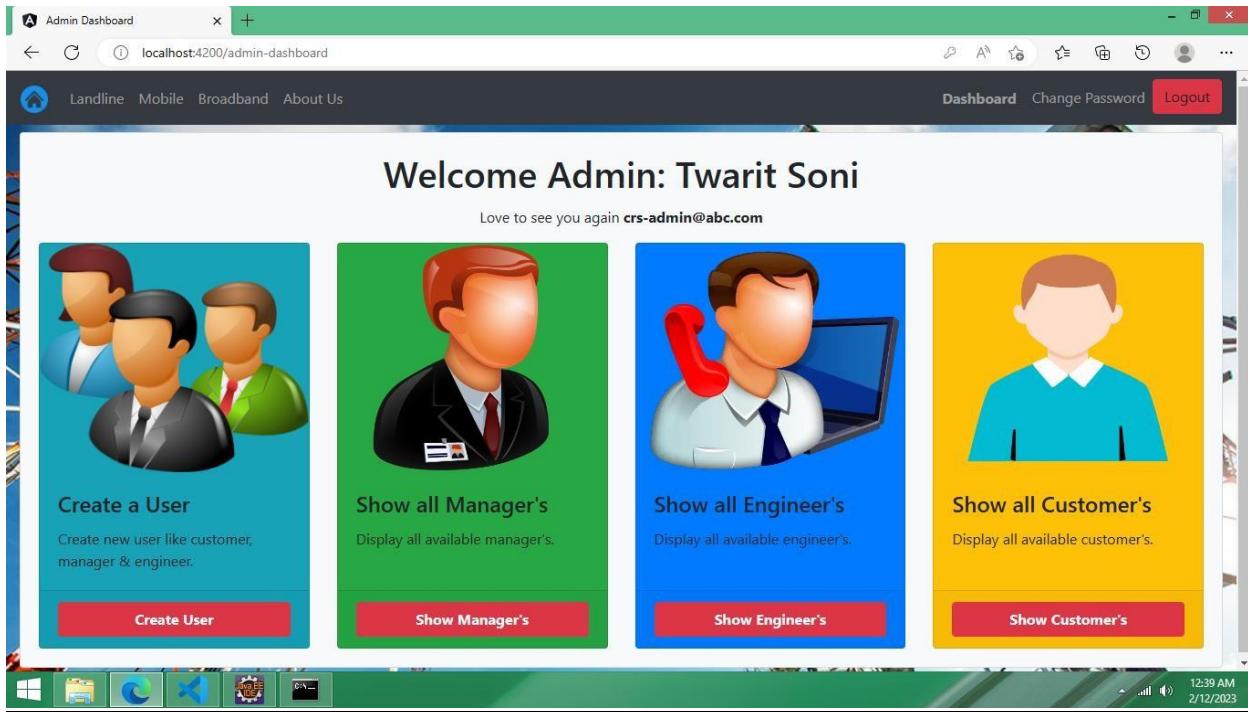




3.On giving wrong admin credentials:

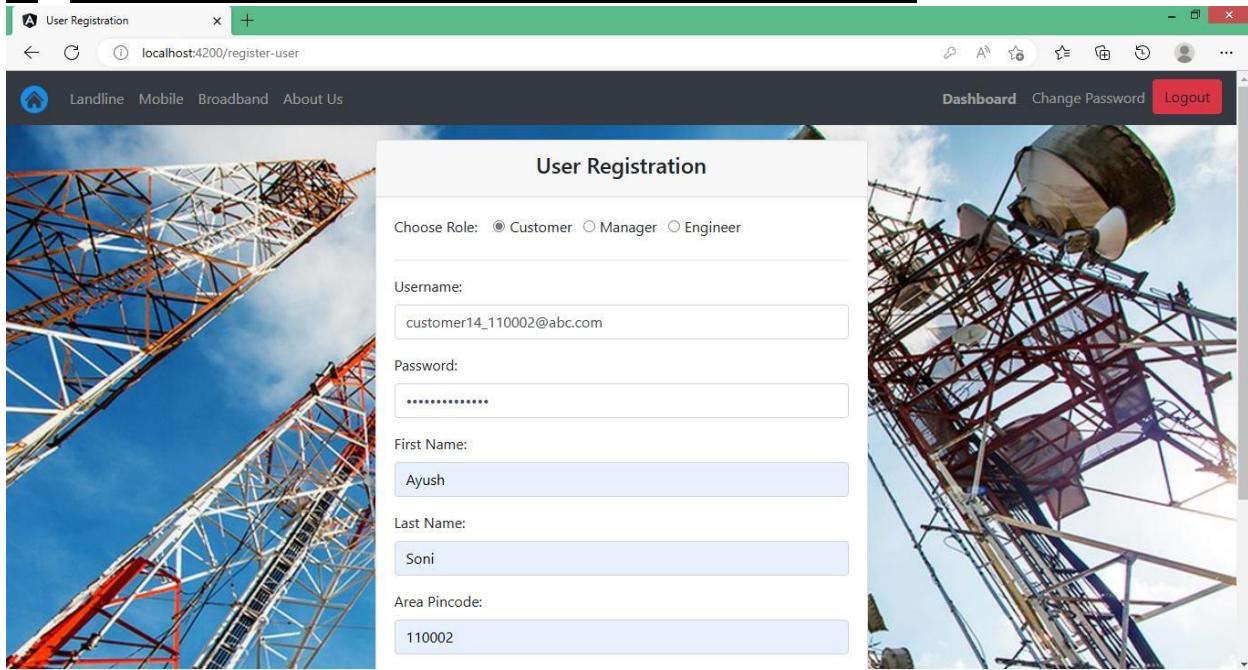
4.Admin Dashboard after giving right credentials:

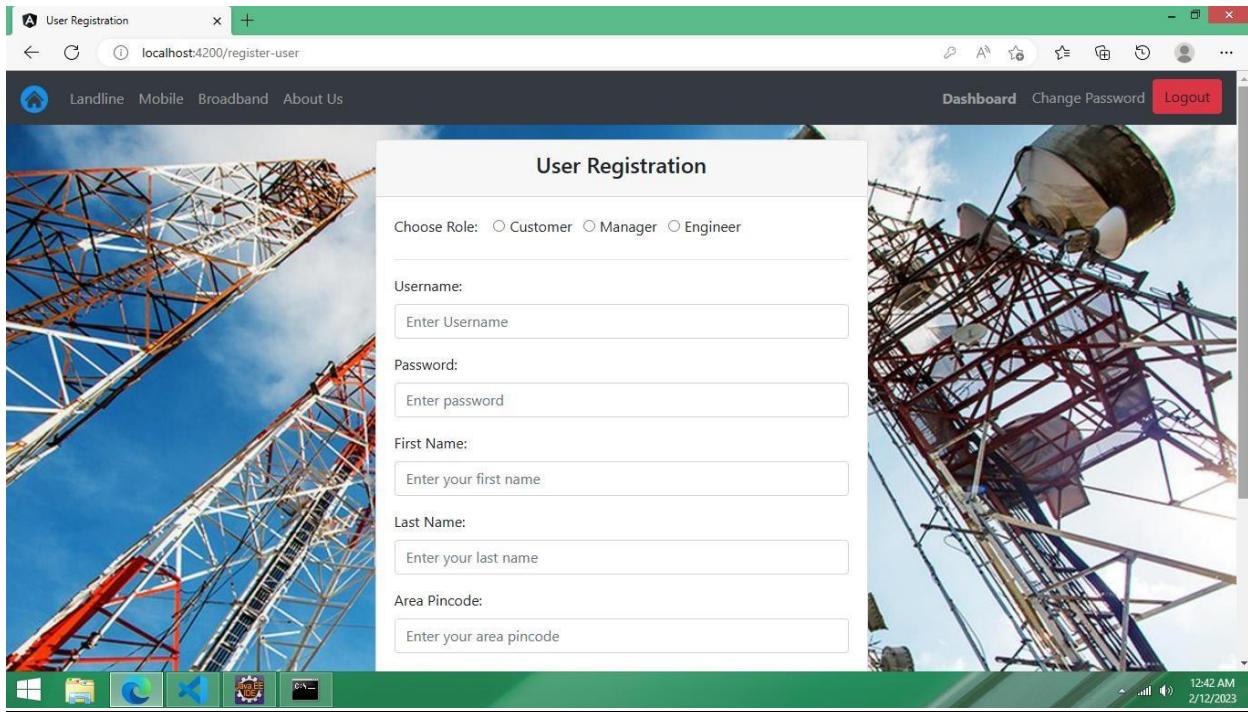




5. On clicking ChangePassword and changing the password:

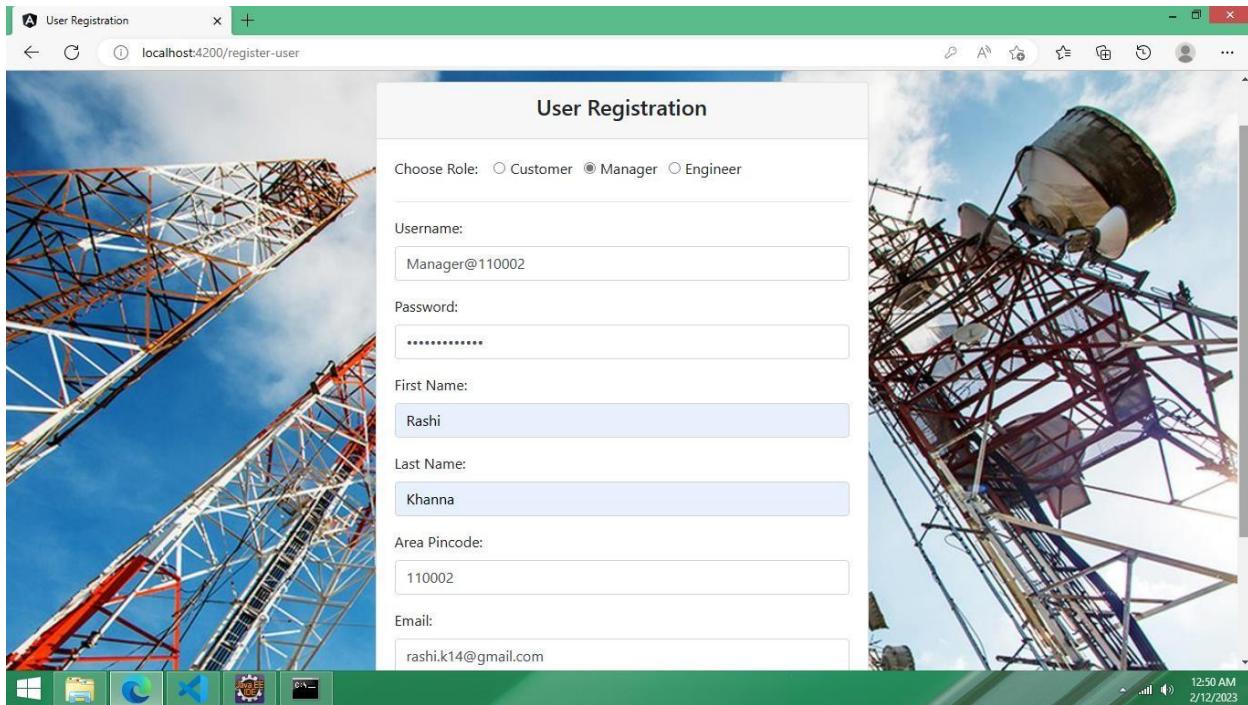
6. From Admin Dashboard on click Create User:

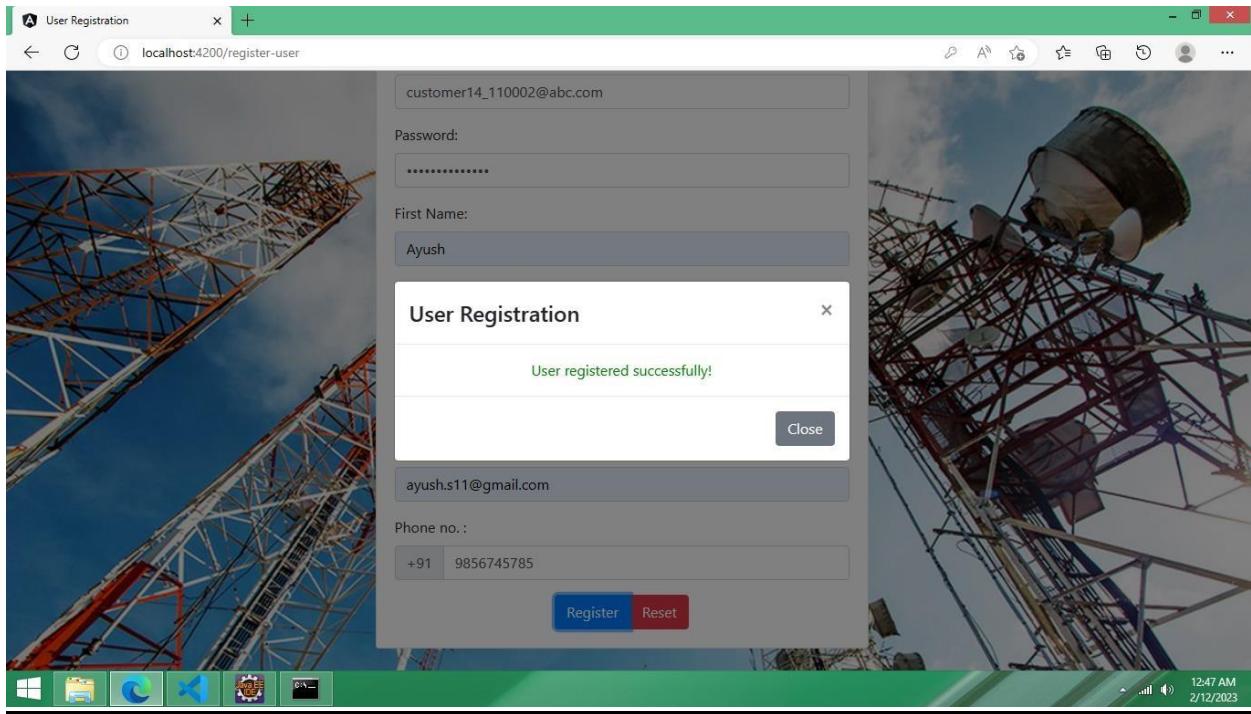




7. Creating a Customer User:

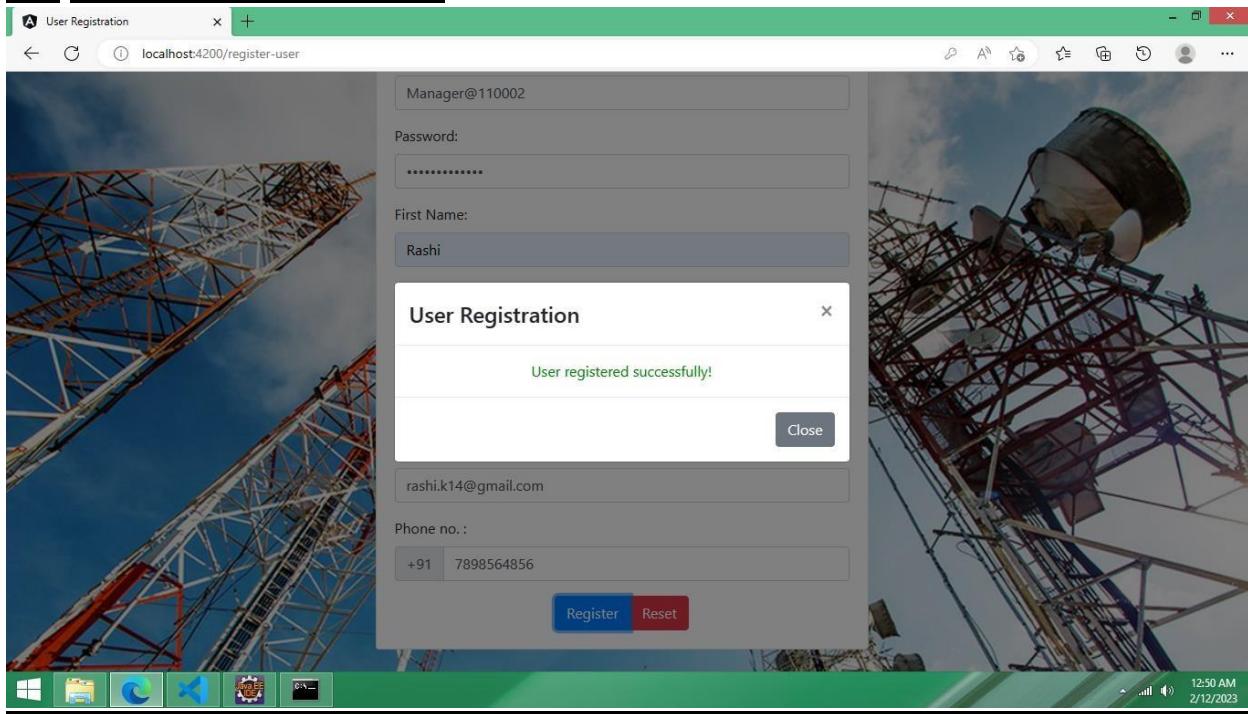
8. On clicking Register after entering details:



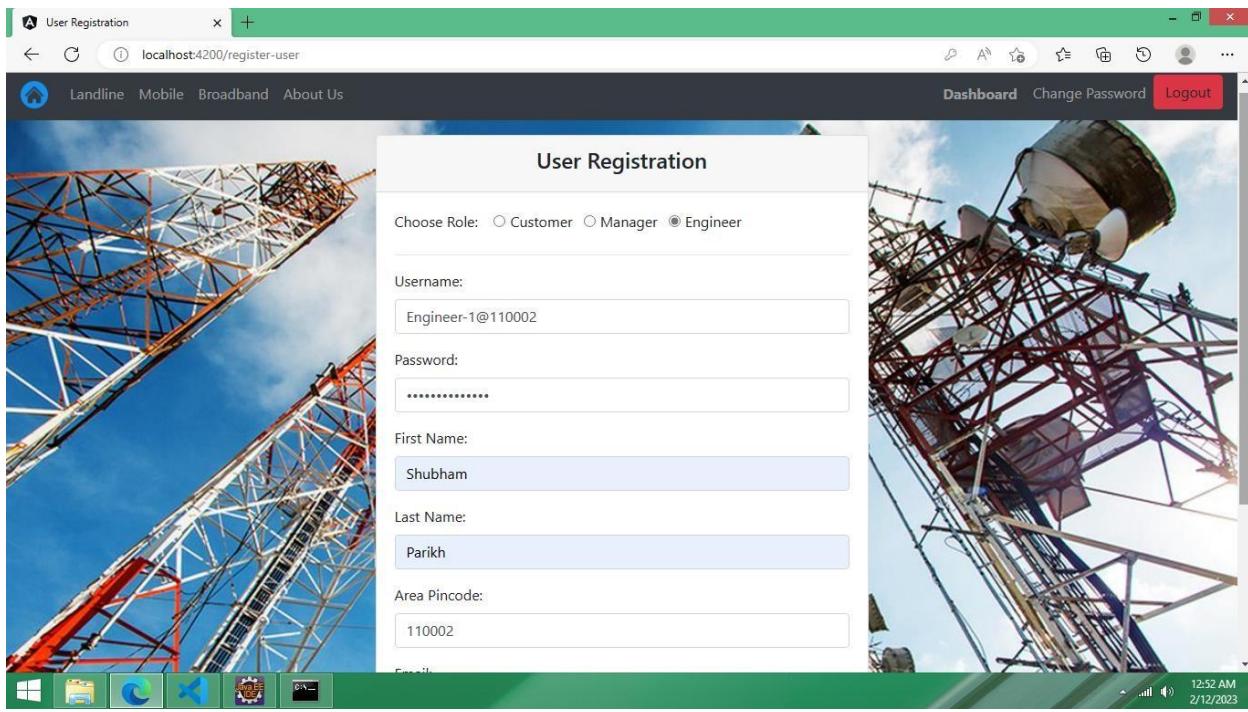


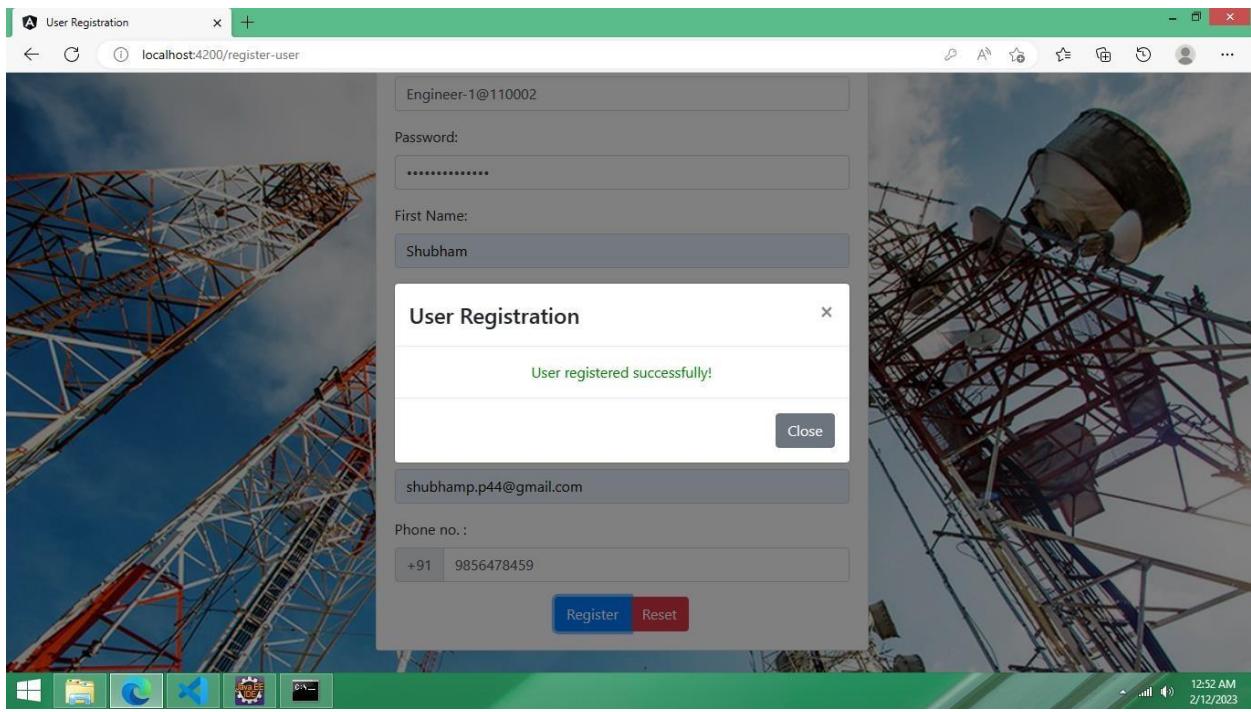
9. Creating a Manager User:

10. On clicking register:



11. Creating a Engineer User and clicking register:





A screenshot of a web browser window titled "Managers". The URL is "localhost:4200/managers". The page shows a table titled "All Available Manager's" with the following data:

S.No.	Manager Id	User Name	First Name	Last Name	Pincode	E-mail	Contact	Action
1	7	Manager@110005	Rajesh	Soni	110005	rajesh.s88@gmail.com	7454956812	<button>Delete</button> <button>Update</button>
2	51	Manager@110002	Rashi	Khanna	110002	rashi.k14@gmail.com	7898564856	<button>Delete</button> <button>Update</button>
3	55	Manager@110004	Prateek	Soni	110004	prateek.s98@gmail.com	6598745689	<button>Delete</button> <button>Update</button>
4	57	Manager@110006	Sumit	Jain	110006	sumits78@gmail.com	7895684856	<button>Delete</button> <button>Update</button>
5	59	Manager@110008	Ananya	Joshi	110008	ananya.j44@gmail.com	9856745685	<button>Delete</button> <button>Update</button>

The background of the page shows a photograph of a tall metal lattice tower against a blue sky. The top navigation bar includes links for "Managers", "Engineers", "Landline", "Mobile", "Broadband", "About Us", "Dashboard", "Change Password", and "Logout". The system tray at the bottom shows standard Windows icons and the date/time "12/12/2023 12:52 AM".

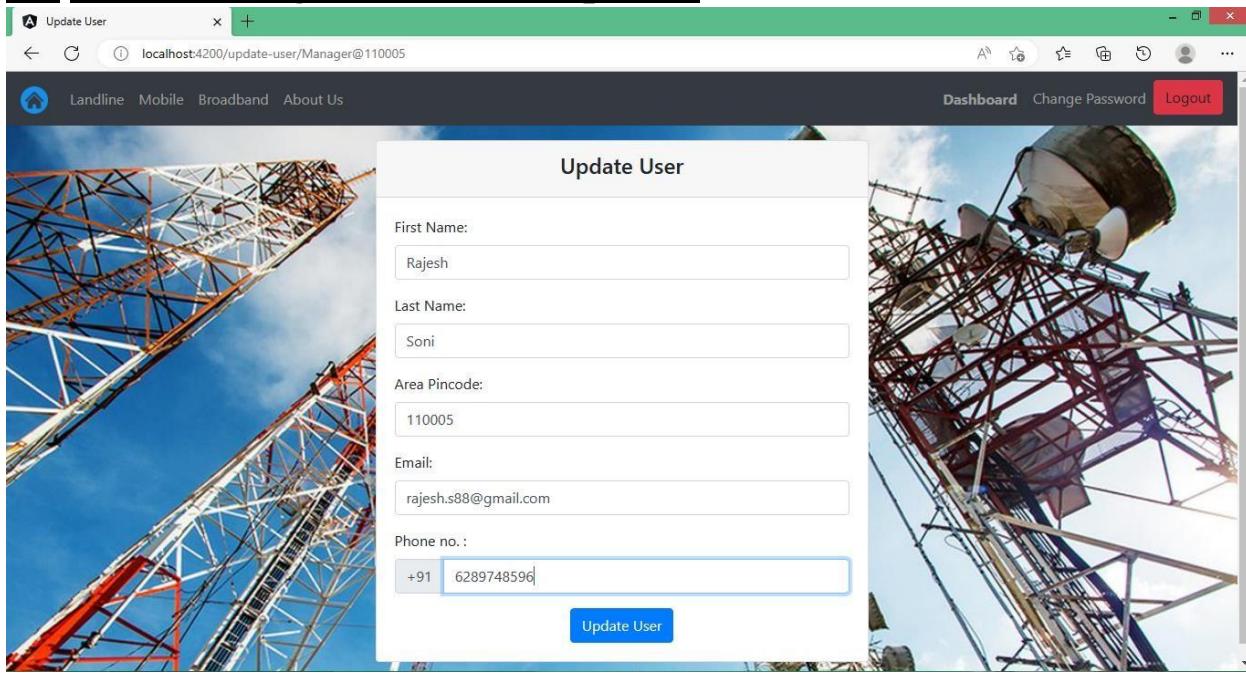
12. From dashboard on clicking Show All Manager's:

13. From dashboard on clicking Show All Customer's:

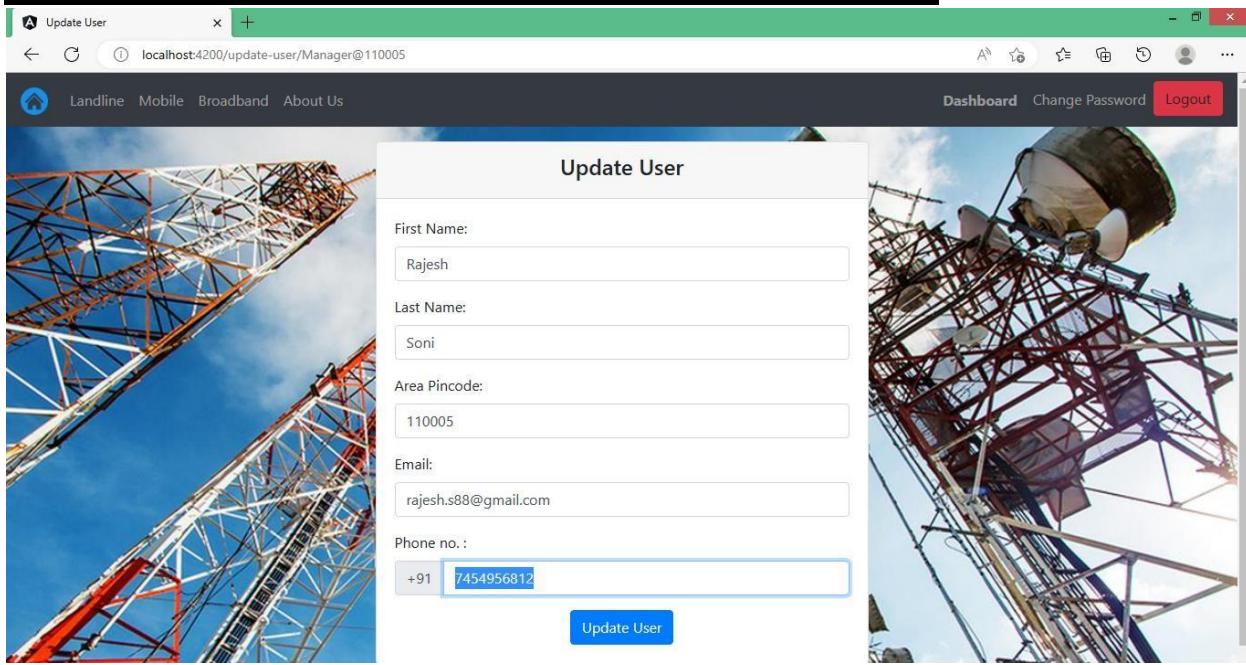
S.No.	Customer Id	User Name	First Name	Last Name	Pincode	E-mail	Contact	Action
1	11	customer14_110005@abc.com	Anupam	Jain	110005	anupam.m02@gmail.com	7895456858	<button>Delete</button> <button>Update</button>
2	15	customer16_110005@abc.com	Animesh	Chatterjee	110005	animesh.c04@gmail.com	6265859905	<button>Delete</button> <button>Update</button>
3	49	customer14_110002@abc.com	Ayush	Soni	110002	ayush.s11@gmail.com	9856745785	<button>Delete</button> <button>Update</button>
4	67	customer8_110004@abc.com	Aditi	Trivedi	110004	aditi.t79@gmail.com	9857484564	<button>Delete</button> <button>Update</button>

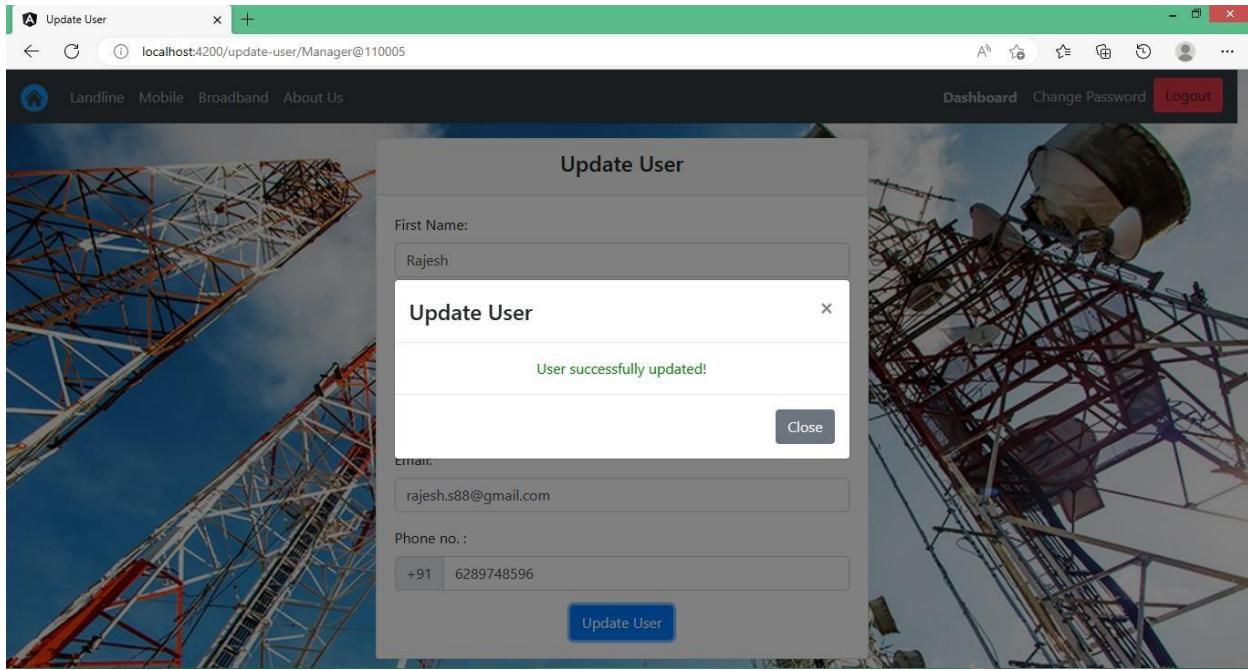
S.No.	Engineer Id	User Name	First Name	Last Name	Pincode	E-mail	Contact	Action
1	9	Engineer-1@110005	Prashant	Jain	110005	prashant.j14@gmail.com	9854679856	<button>Delete</button> <button>Update</button>
2	53	Engineer-1@110002	Shubham	Parikh	110002	shubhamp.p44@gmail.com	9856478459	<button>Delete</button> <button>Update</button>
3	61	Engineer-1@110004	Aditya	Dwivedi	110004	aditya.d96@gmail.com	7485956847	<button>Delete</button> <button>Update</button>
4	63	Engineer-1@110006	Mitali	Dasgupta	110006	mitali.d76@gmail.com	6598745865	<button>Delete</button> <button>Update</button>
5	65	Engineer-1@110008	Vineeta	Singh	110008	vini.s88@gmail.com	7895684859	<button>Delete</button> <button>Update</button>

14. From Manager's, on click Update:



Updating phone number and clicking Update User:

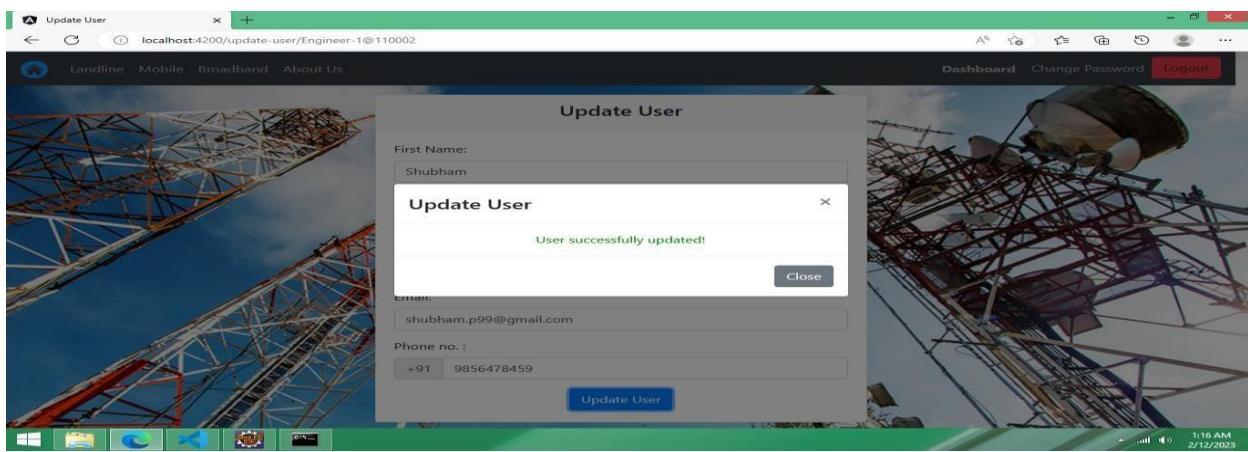
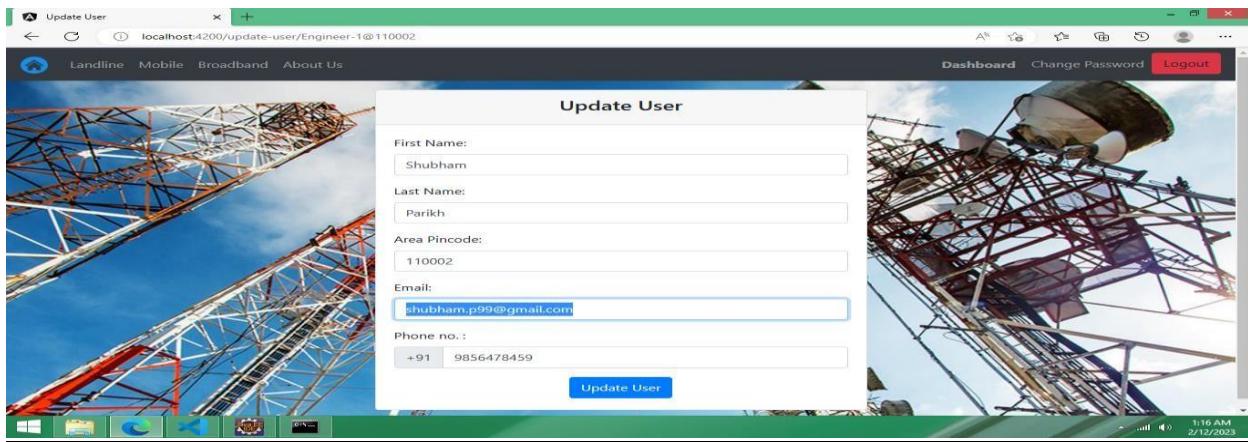




15. On Show All Manager's, Contact is updated :

S.No.	Manager Id	User Name	First Name	Last Name	Pincode	E-mail	Contact	Action
1	7	Manager@110005	Rajesh	Soni	110005	rajesh.s88@gmail.com	6289748596	<button>Delete</button> <button>Update</button>
2	51	Manager@110002	Rashi	Khanna	110002	rashi.k14@gmail.com	7898564856	<button>Delete</button> <button>Update</button>
3	55	Manager@110004	Prateek	Soni	110004	prateek.s98@gmail.com	6598745689	<button>Delete</button> <button>Update</button>
4	57	Manager@110006	Sumit	Jain	110006	sumit.s78@gmail.com	7895684856	<button>Delete</button> <button>Update</button>
5	59	Manager@110008	Ananya	Joshi	110008	ananya.j44@gmail.com	9856745685	<button>Delete</button> <button>Update</button>

16. Similarly updating other user's:



All Available Engineer's								
S.No.	Engineer Id	User Name	First Name	Last Name	Pincode	E-mail	Contact	Action
1	9	Engineer-1@110005	Prashant	Jain	110005	prashant.j14@gmail.com	9854679856	<button>Delete</button> <button>Update</button>
2	53	Engineer-1@110002	Shubham	Parikh	110002	shubham.p99@gmail.com	9856478459	<button>Delete</button> <button>Update</button>
3	61	Engineer-1@110004	Aditya	Dwivedi	110004	aditya.d96@gmail.com	7485956847	<button>Delete</button> <button>Update</button>
4	63	Engineer-1@110006	Mitali	Dasgupta	110006	mitali.d76@gmail.com	6598745865	<button>Delete</button> <button>Update</button>
5	65	Engineer-1@110008	Vineeta	Singh	110008	vini.s88@gmail.com	7895684859	<button>Delete</button> <button>Update</button>

17. From Show All Customer's Deleting a customer:

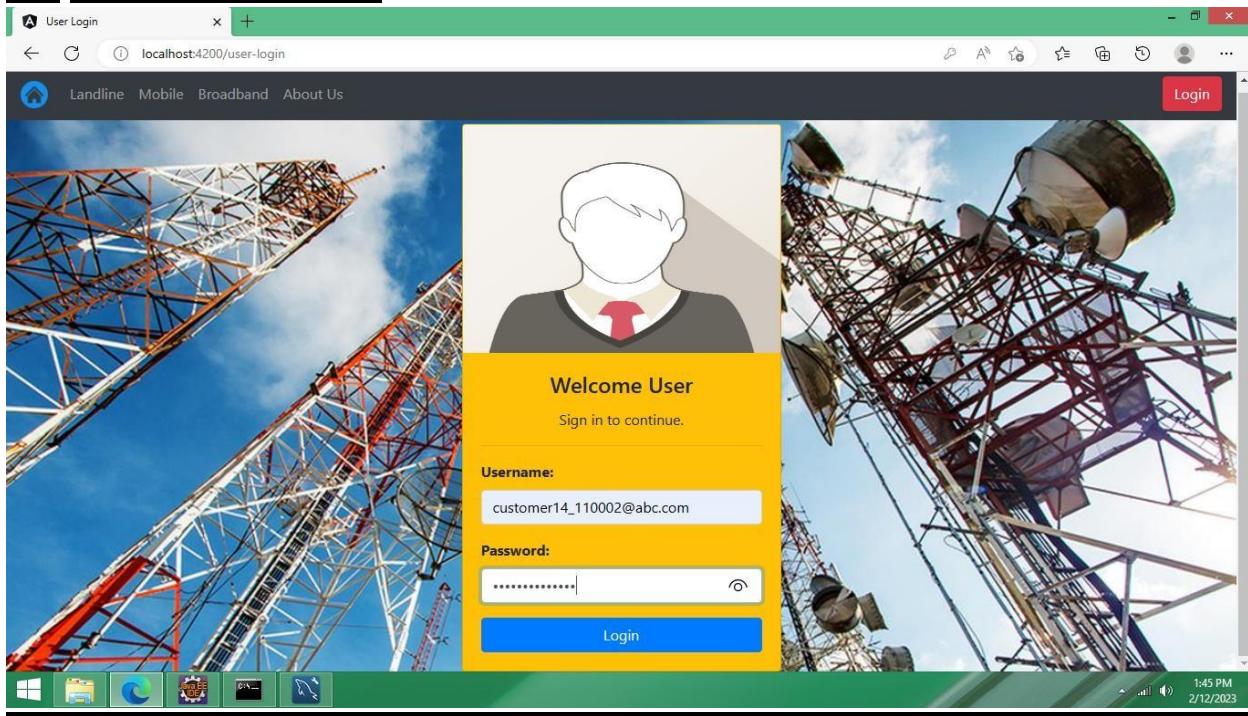
A screenshot of a web browser window titled "Customers" at the URL "localhost:4200/customers". The page header includes links for "Landline", "Mobile", "Broadband", "About Us", "Dashboard", "Change Password", and "Logout". The main content area has a title "All Available Customer's" over a background image of a telecommunications tower. A table lists five customer entries with columns: S.No., Customer Id, User Name, First Name, Last Name, Pincode, E-mail, Contact, and Action (with "Delete" and "Update" buttons). The fifth row's "Action" column is highlighted with a red border.

S.No.	Customer Id	User Name	First Name	Last Name	Pincode	E-mail	Contact	Action
1	11	customer14_110005@abc.com	Anupam	Jain	110005	anupam.m02@gmail.com	7895456858	<button>Delete</button> <button>Update</button>
2	15	customer16_110005@abc.com	Animesh	Chatterjee	110005	animesh.c04@gmail.com	6265859905	<button>Delete</button> <button>Update</button>
3	49	customer14_110002@abc.com	Ayush	Soni	110002	ayush.s11@gmail.com	7898564585	<button>Delete</button> <button>Update</button>
4	67	customer8_110004@abc.com	Aditi	Trivedi	110004	aditi.t79@gmail.com	9857484564	<button>Delete</button> <button>Update</button>
5	69	customer88_110008@abc.com	Shreya	Soni	110008	shreya.s99@gmail.com	9854785954	<button>Delete</button> <button>Update</button>

A screenshot of a web browser window titled "Customers" at the URL "localhost:4200/customers". The page header and main content area are identical to the first screenshot, but the table now only contains four rows of customer data, indicating that the fifth row has been successfully deleted.

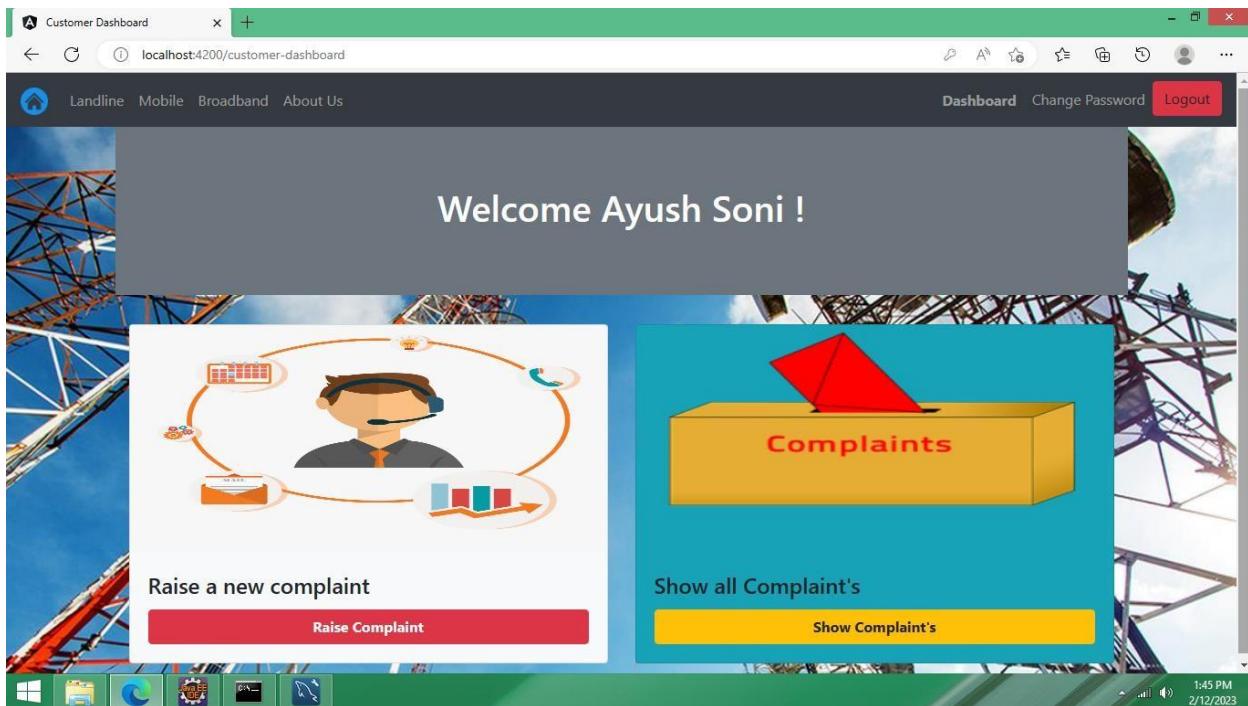
S.No.	Customer Id	User Name	First Name	Last Name	Pincode	E-mail	Contact	Action
1	11	customer14_110005@abc.com	Anupam	Jain	110005	anupam.m02@gmail.com	7895456858	<button>Delete</button> <button>Update</button>
2	15	customer16_110005@abc.com	Animesh	Chatterjee	110005	animesh.c04@gmail.com	6265859905	<button>Delete</button> <button>Update</button>
3	49	customer14_110002@abc.com	Ayush	Soni	110002	ayush.s11@gmail.com	7898564585	<button>Delete</button> <button>Update</button>
4	67	customer8_110004@abc.com	Aditi	Trivedi	110004	aditi.t79@gmail.com	9857484564	<button>Delete</button> <button>Update</button>

18. Customer Login:

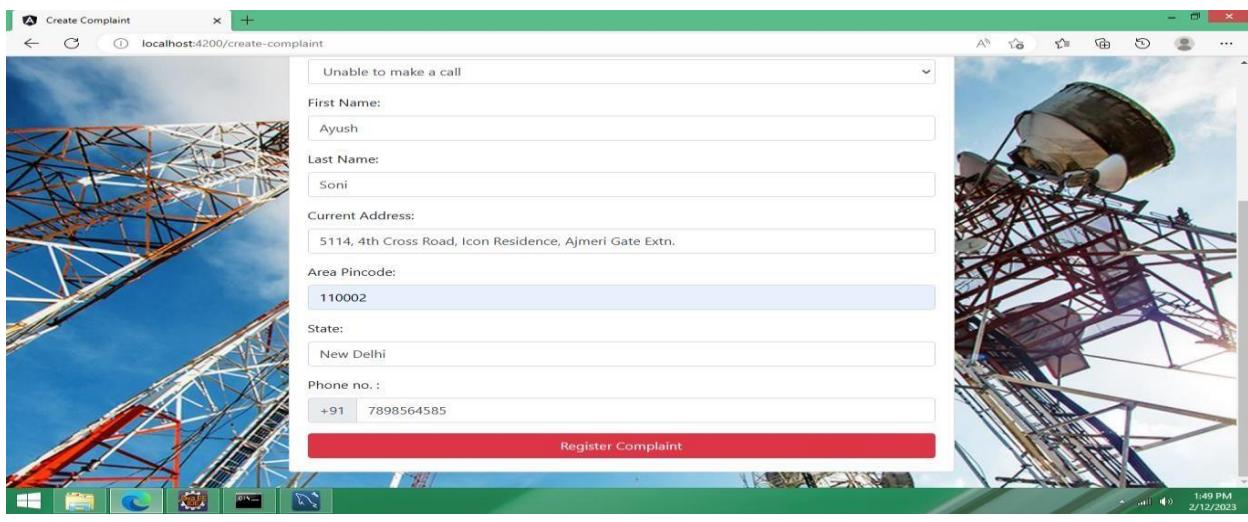


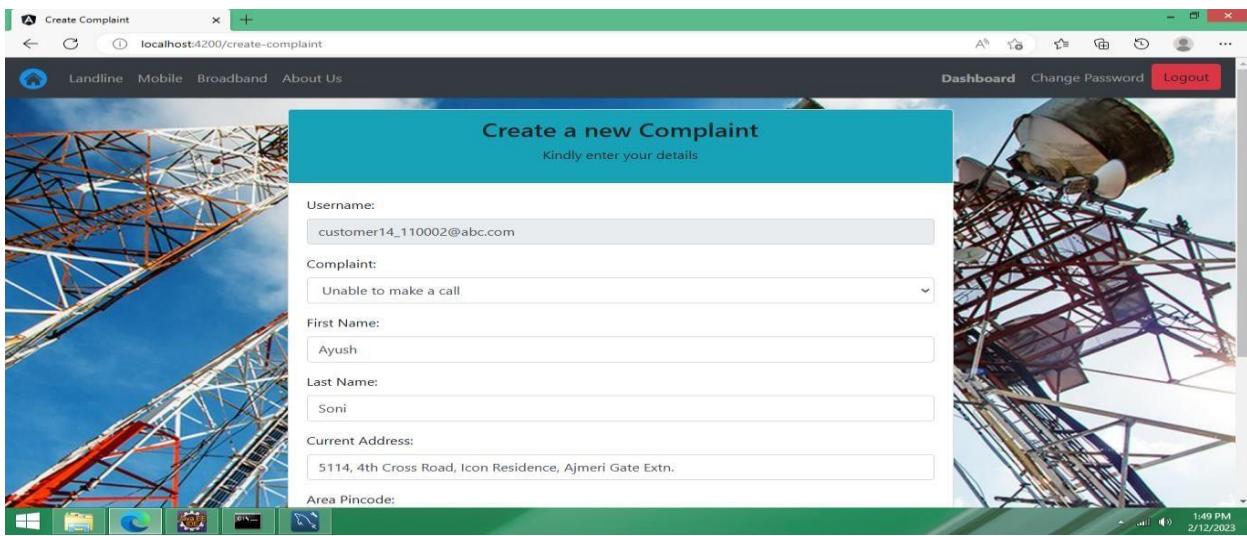
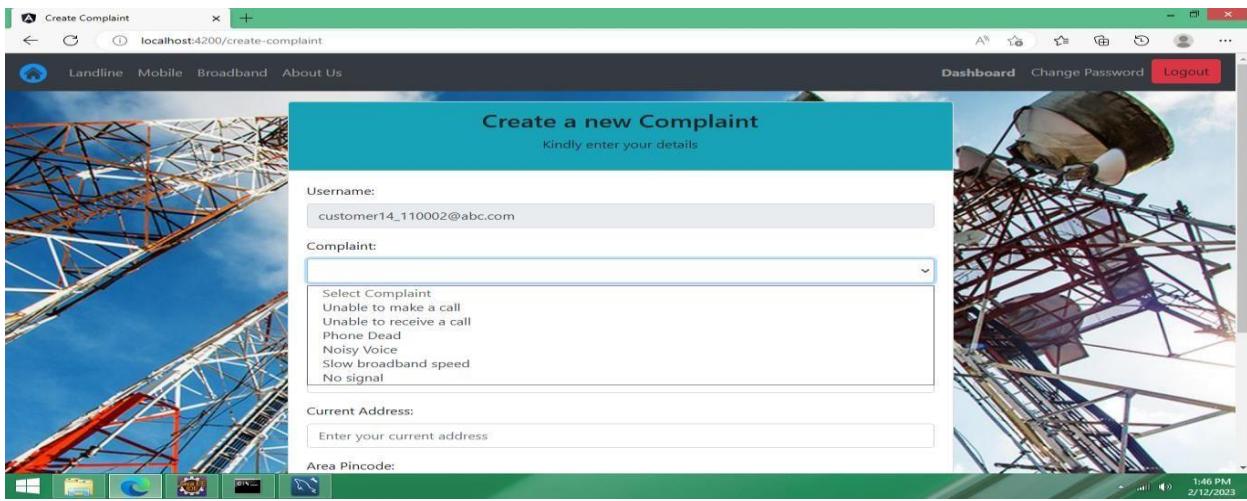
19. Customer Dashboard:



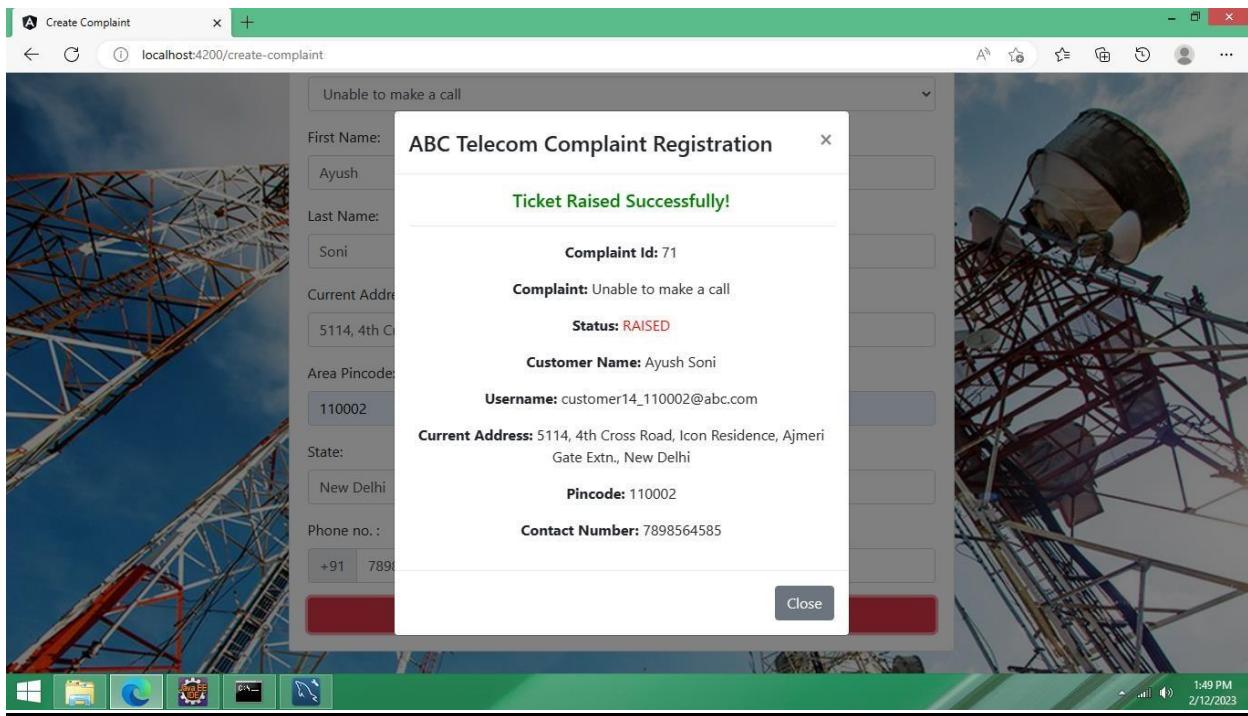


20. On clicking Raise Complaint:





21. On clicking Register Complaint:

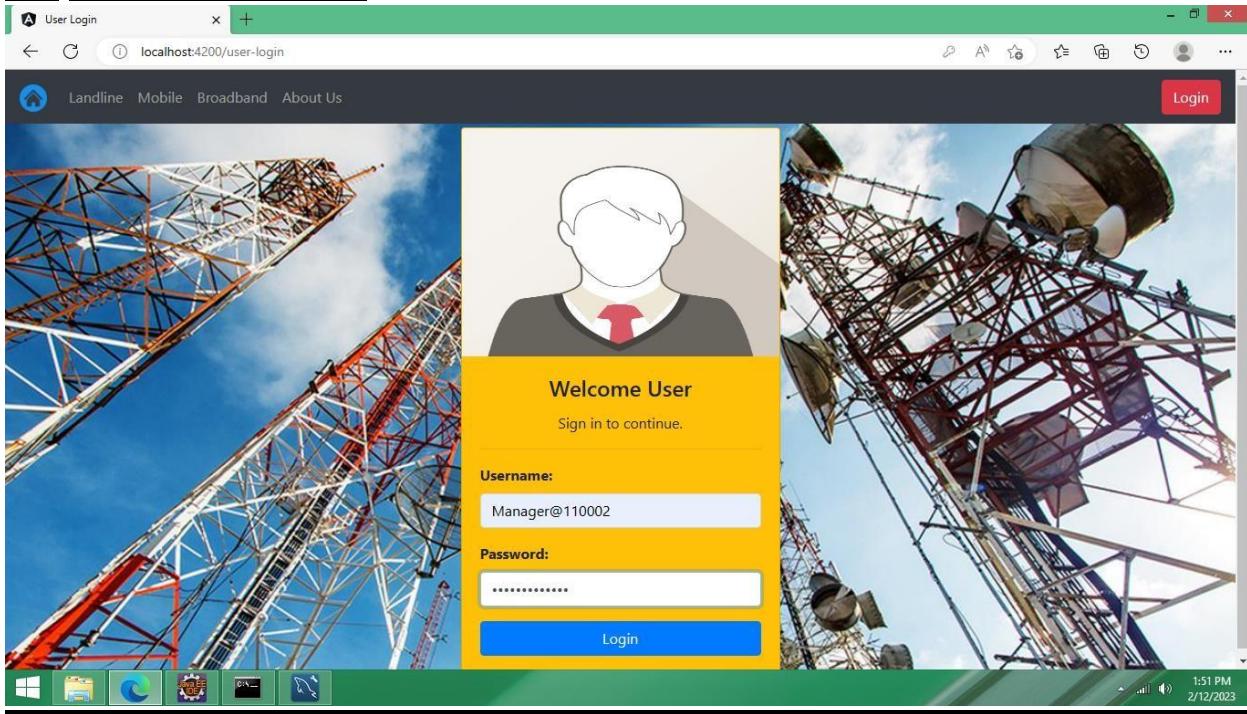


22. From customer dashboard, click on Show Complaint's:

A screenshot of a web browser window titled "All Complaints" with the URL "localhost:4200/customer/get-complaints". The page header includes links for "Landline", "Mobile", "Broadband", "About Us", "Dashboard", "Change Password", and "Logout". The main content area is titled "All Complaints" and displays a table of complaints. The table has the following columns: S.No., Complaint Id, Date, Complaint, Customer Name, Username, Current Address, Pincode, Contact, Status, and Remark. There is one entry in the table:

S.No.	Complaint Id	Date	Complaint	Customer Name	Username	Current Address	Pincode	Contact	Status	Remark
1	71	Feb 12, 2023	Unable to make a call	Ayush Soni	customer14_110002@abc.com	5114, 4th Cross Road, Icon Residence, Ajmeri Gate Extn., New Delhi.	110002	7898564585	RAISED	Ticket Raised.

23. Manager Login:



24. Manager Dashboard:



25. On clicking Assign Engineer:

Unassigned Complaints by PinCode-110002

Date	Complaint	Customer Name	Username	Current Address	Pincode	Contact	Status	Remark	Action
Feb 12, 2023	Unable to make a call	Ayush Soni	customer14_110002@abc.com	5114, 4th Cross Road, Icon Residence, Ajmeri Gate Extn., New Delhi.	110002	7898564585	RAISED	Ticket Raised.	Assign Engineer

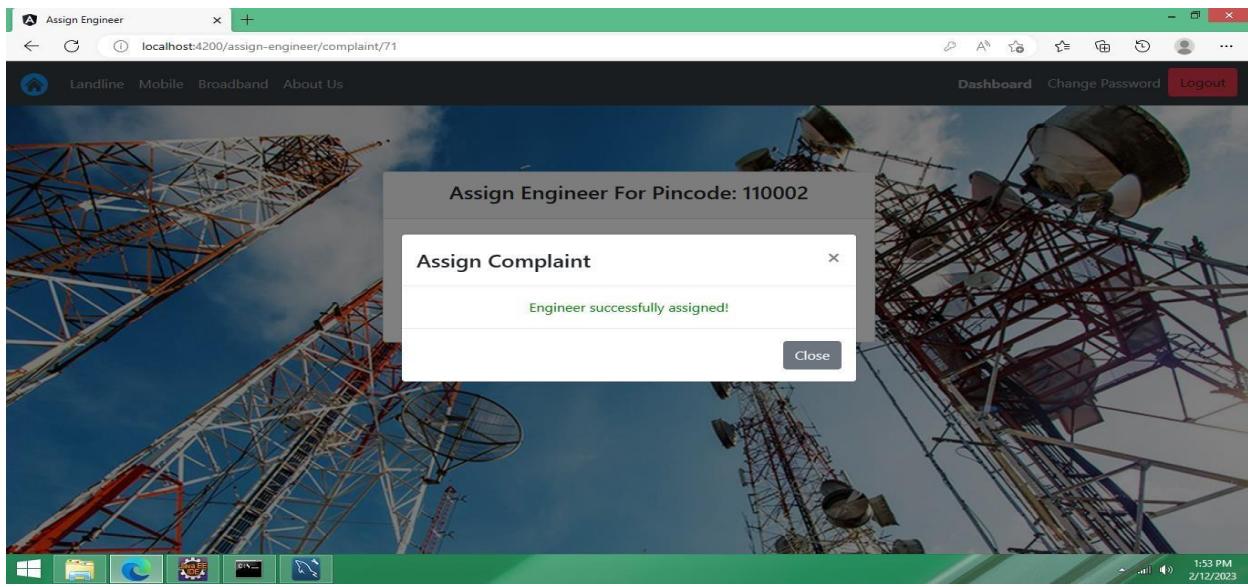
Assigning Engineer:

Assign Engineer For Pincode: 110002

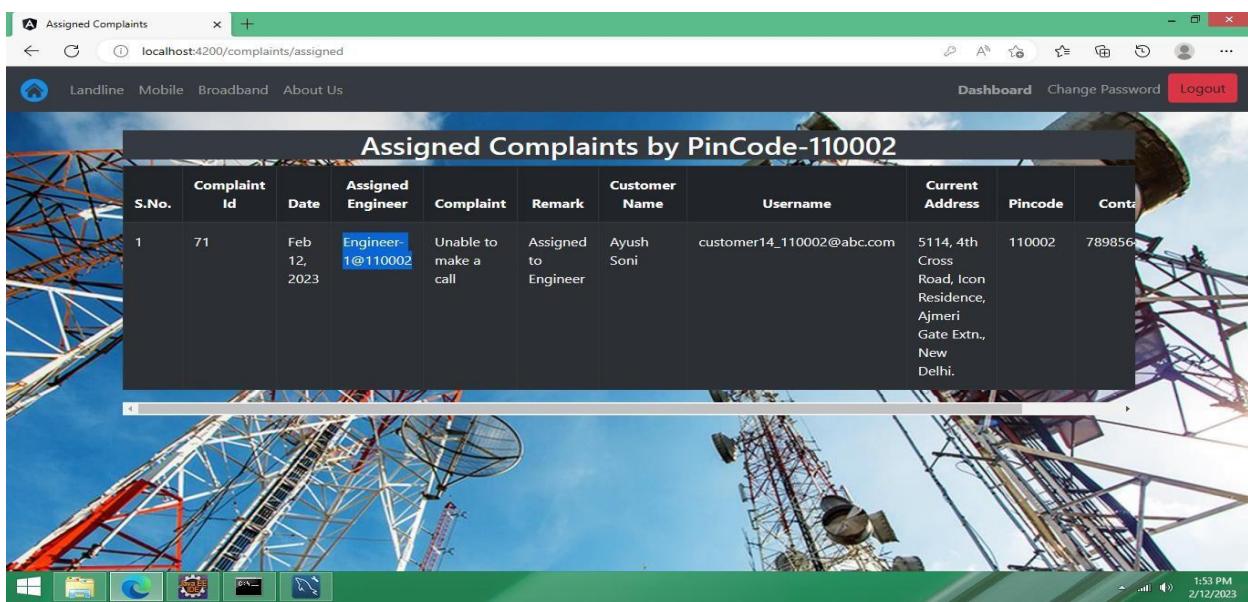
Select Engineer

- Engineer-1@110005
- Engineer-1@110002
- Engineer-1@110004
- Engineer-1@110006
- Engineer-1@110008

Assign Engineer



26. On clicking Show Assigned Complaints from dashboard:

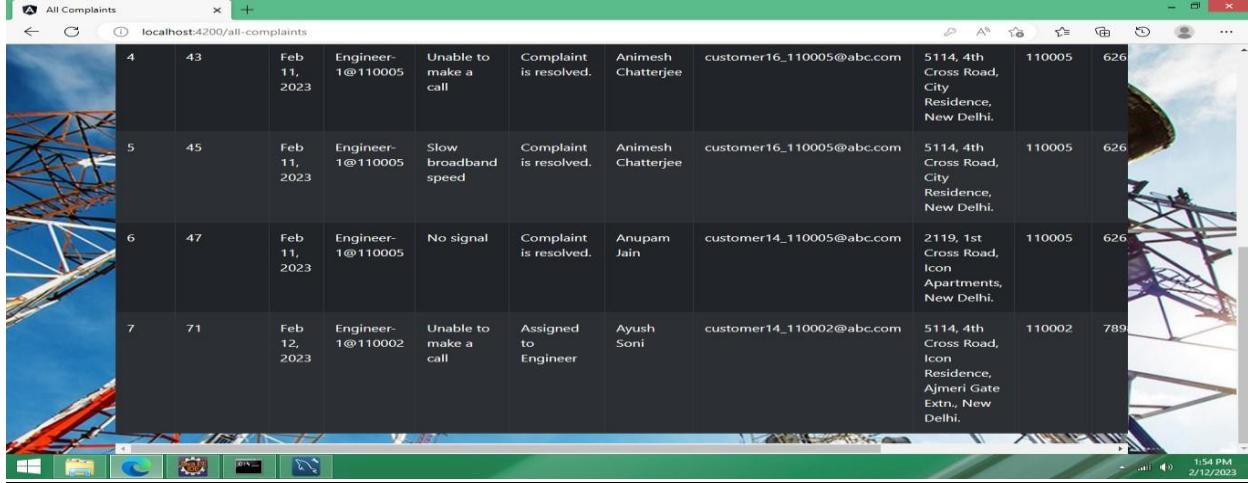


27. On clicking Show All Complaints from dashboard:



The screenshot shows a table titled "All Complaints" with the following data:

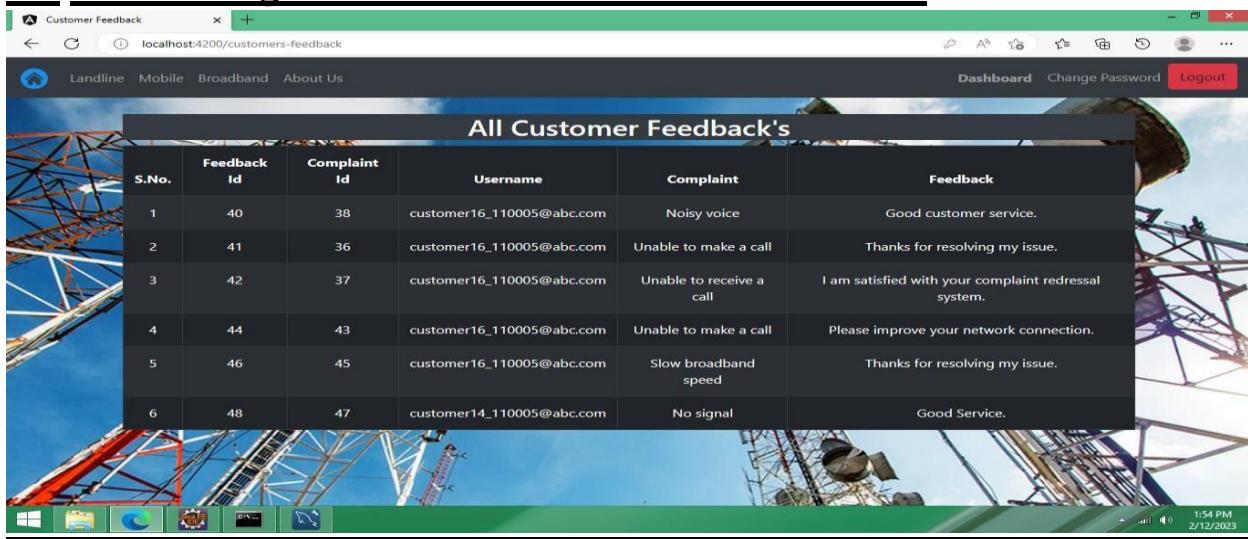
S.No.	Complaint Id	Date	Assigned Engineer	Complaint	Remark	Customer Name	Username	Current Address	Pincode	Category
1	36	Feb 8, 2023	Engineer-1@110005	Unable to make a call	Reassigned to field workers.	Animesh Chatterjee	customer16_110005@abc.com	5114, 4th Cross Road, City Residence, New Delhi.	110005	626
2	37	Feb 8, 2023	Engineer-1@110005	Unable to receive a call	Complaint is resolved.	Animesh Chatterjee	customer16_110005@abc.com	5114, 4th Cross Road, City Residence, New Delhi.	110005	626
3	38	Feb 8, 2023	Engineer-1@110005	Noisy voice	Complaint is closed	Animesh Chatterjee	customer16_110005@abc.com	5114, 4th Cross Road, City Residence, New Delhi.	110005	626



The screenshot shows a table titled "All Complaints" with the following data:

S.No.	Complaint Id	Date	Assigned Engineer	Complaint	Remark	Customer Name	Username	Current Address	Pincode	Category
4	43	Feb 11, 2023	Engineer-1@110005	Unable to make a call	Complaint is resolved.	Animesh Chatterjee	customer16_110005@abc.com	5114, 4th Cross Road, City Residence, New Delhi.	110005	626
5	45	Feb 11, 2023	Engineer-1@110005	Slow broadband speed	Complaint is resolved.	Animesh Chatterjee	customer16_110005@abc.com	5114, 4th Cross Road, City Residence, New Delhi.	110005	626
6	47	Feb 11, 2023	Engineer-1@110005	No signal	Complaint is resolved.	Anupam Jain	customer14_110005@abc.com	2119, 1st Cross Road, Icon Apartments, New Delhi.	110005	626
7	71	Feb 12, 2023	Engineer-1@110002	Unable to make a call	Assigned to Engineer	Ayush Soni	customer14_110002@abc.com	5114, 4th Cross Road, Icon Residence, Ajmeri Gate Extn, New Delhi.	110002	789

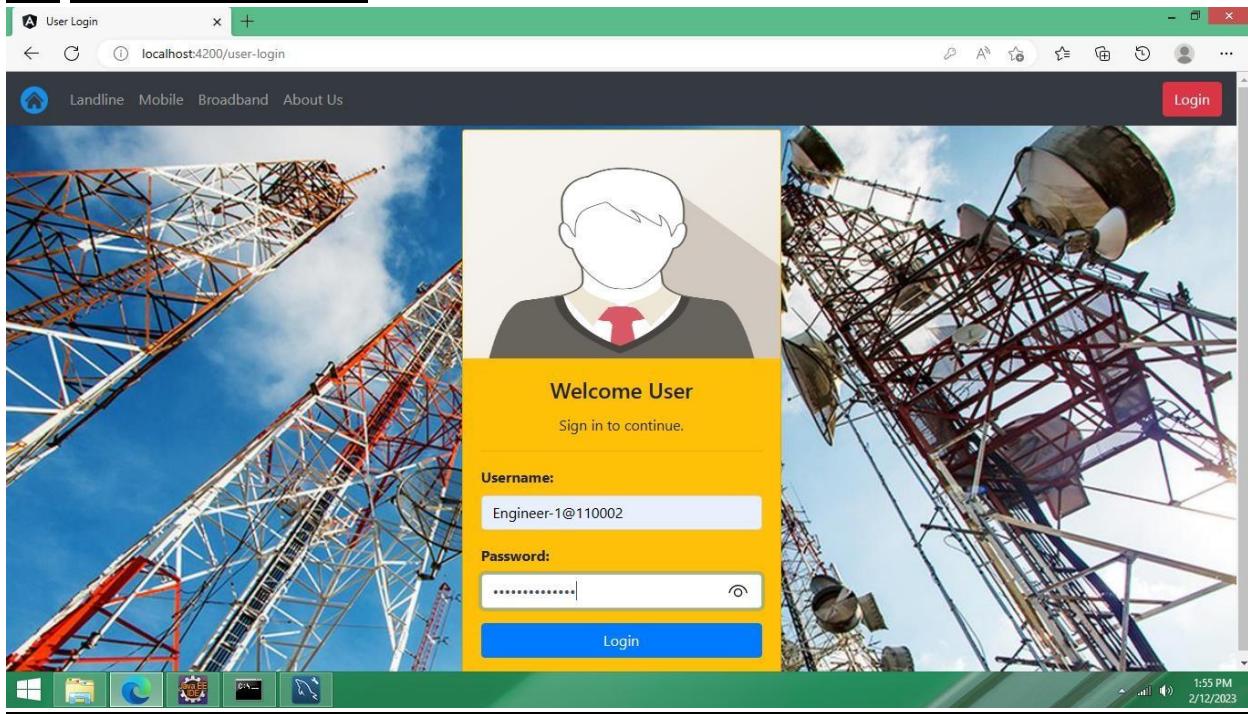
28. On clicking Show Feedback's from dashboard:



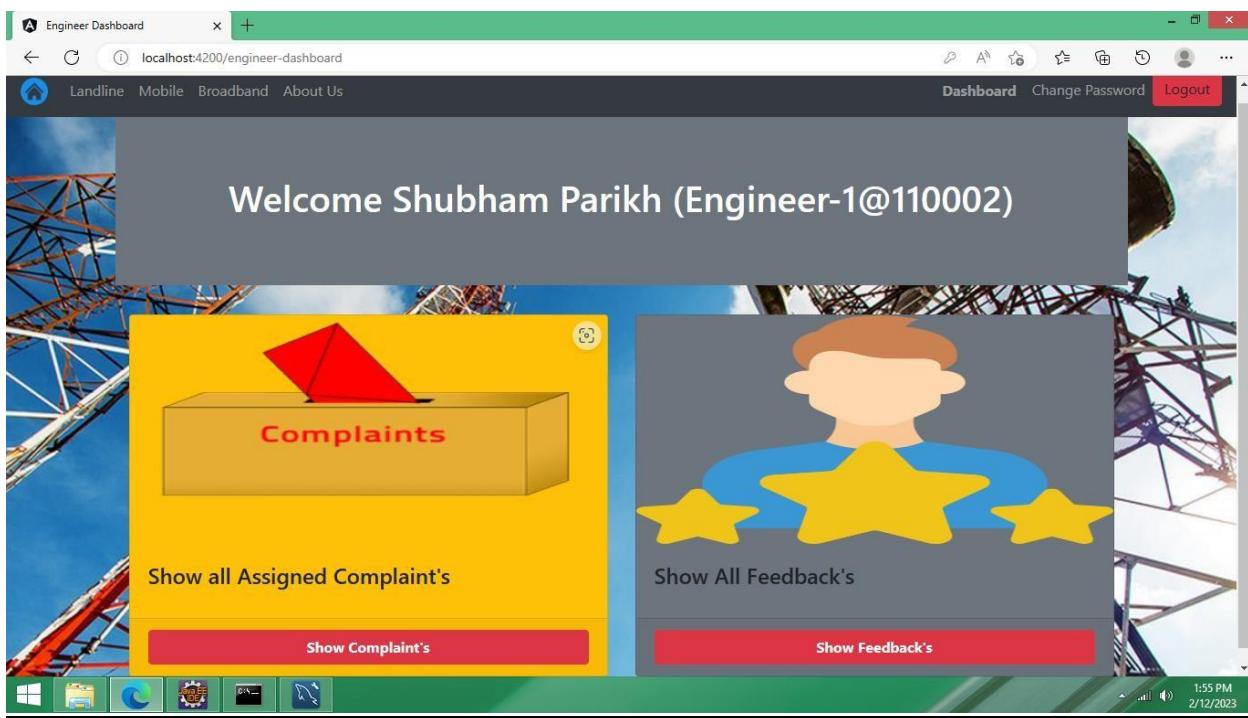
The screenshot shows a table titled "All Customer Feedback's" with the following data:

S.No.	Feedback Id	Complaint Id	Username	Complaint	Feedback
1	40	38	customer16_110005@abc.com	Noisy voice	Good customer service.
2	41	36	customer16_110005@abc.com	Unable to make a call	Thanks for resolving my issue.
3	42	37	customer16_110005@abc.com	Unable to receive a call	I am satisfied with your complaint redressal system.
4	44	43	customer16_110005@abc.com	Unable to make a call	Please improve your network connection.
5	46	45	customer16_110005@abc.com	Slow broadband speed	Thanks for resolving my issue.
6	48	47	customer14_110005@abc.com	No signal	Good Service.

29. Engineer Login:



30. Engineer Dashboard:



31. On clicking Show Assigned Complaints and then Assign Status:

A screenshot of a web browser window titled "Complaints Assigned". The URL is "localhost:4200/engineer/assigned-complaints". The page header includes "Landline", "Mobile", "Broadband", "About Us", "Dashboard", "Change Password", and "Logout". The main content area has a title "All Assigned Complaints" and a table with the following data:

S.No.	Complaint Id	Date	Assigned Engineer	Complaint	Remark	Customer Name	Username	Current Address	Pincode	Contact
1	71	Feb 12, 2023	Engineer-1@110002	Unable to make a call	Assigned to Engineer	Ayush Soni	customer14_110002@abc.com	5114, 4th Cross Road, Icon Residence, Ajmeri Gate Extn., New Delhi.	110002	7898564585

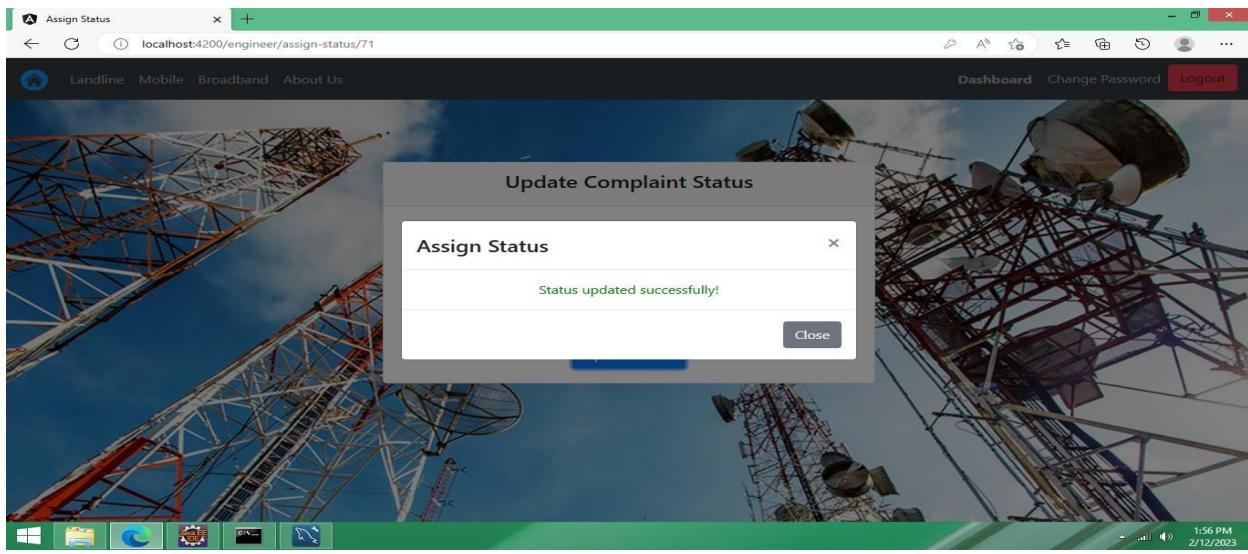
A screenshot of a web browser window titled "Complaints Assigned". The URL is "localhost:4200/engineer/assigned-complaints". The page header includes "Landline", "Mobile", "Broadband", "About Us", "Dashboard", "Change Password", and "Logout". The main content area has a title "All Assigned Complaints" and a table with the following data. A blue button labeled "Assign Status" is visible in the last column of the table.

Assigned Engineer	Complaint	Remark	Customer Name	Username	Current Address	Pincode	Contact	Status	Action
Engineer-1@110002	Unable to make a call	Assigned to Engineer	Ayush Soni	customer14_110002@abc.com	5114, 4th Cross Road, Icon Residence, Ajmeri Gate Extn., New Delhi.	110002	7898564585	RAISED	<button>Assign Status</button>

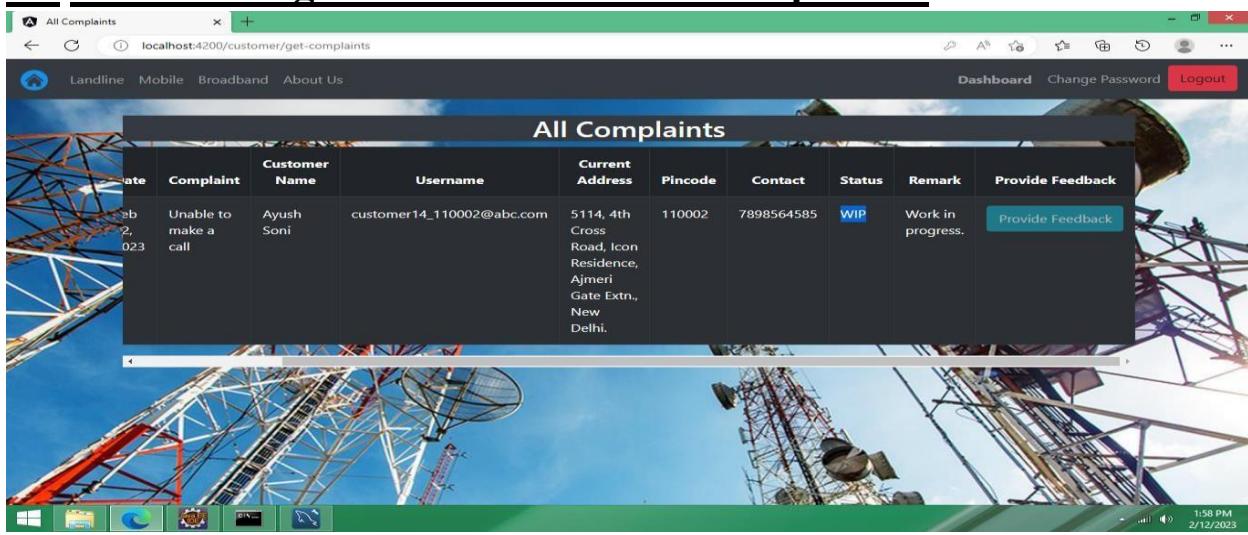
A screenshot of a web browser window titled "Assign Status". The URL is "localhost:4200/engineer/assign-status/71". The page header includes "Landline", "Mobile", "Broadband", "About Us", "Dashboard", "Change Password", and "Logout". The main content area has a title "Update Complaint Status" and a form with the following fields:

Select Status:

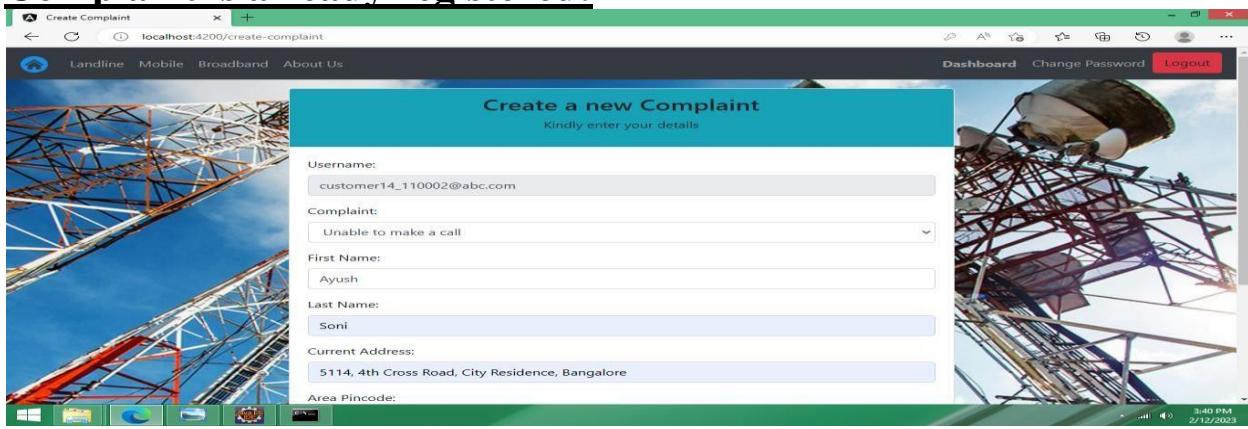
Remark:

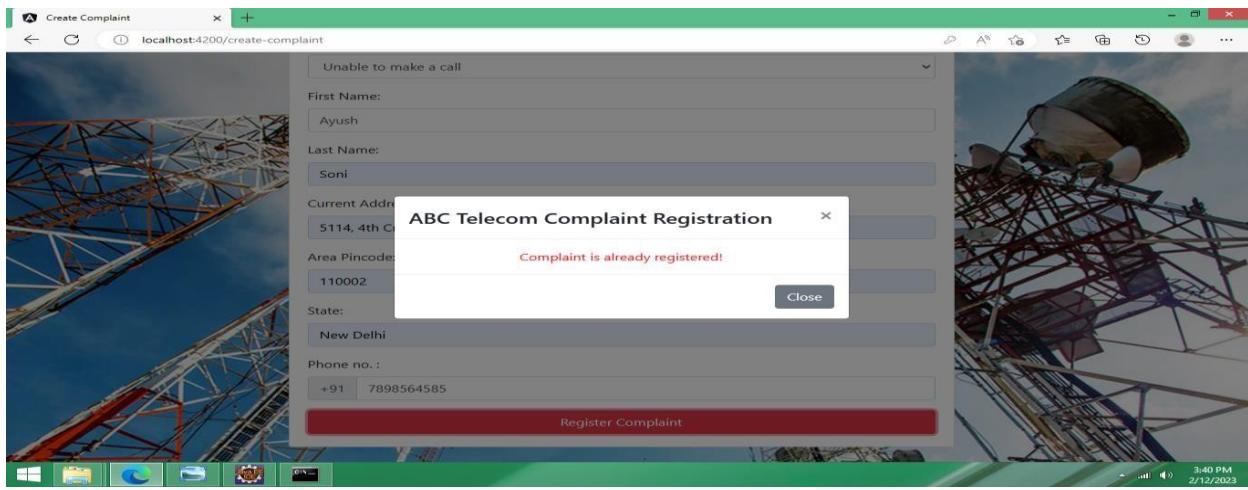


32. Customer login and click on Show Complaints:

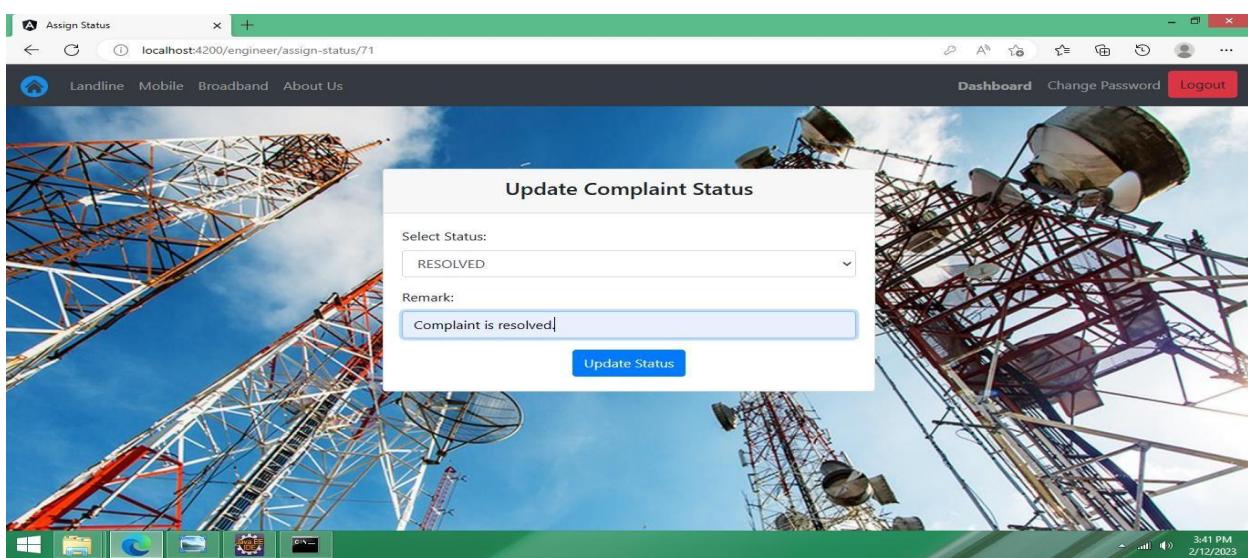


If Customer again Register a complaint on same issue if not resolved or work in progress:
Complaint is already registered:

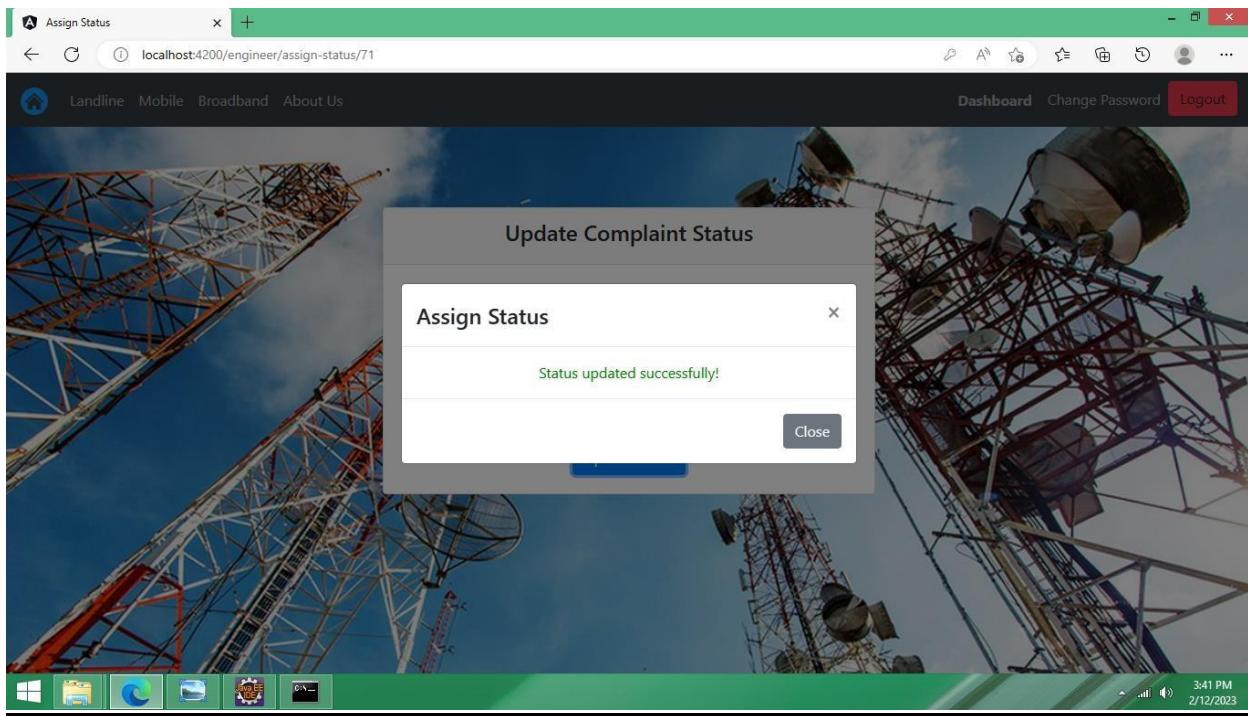




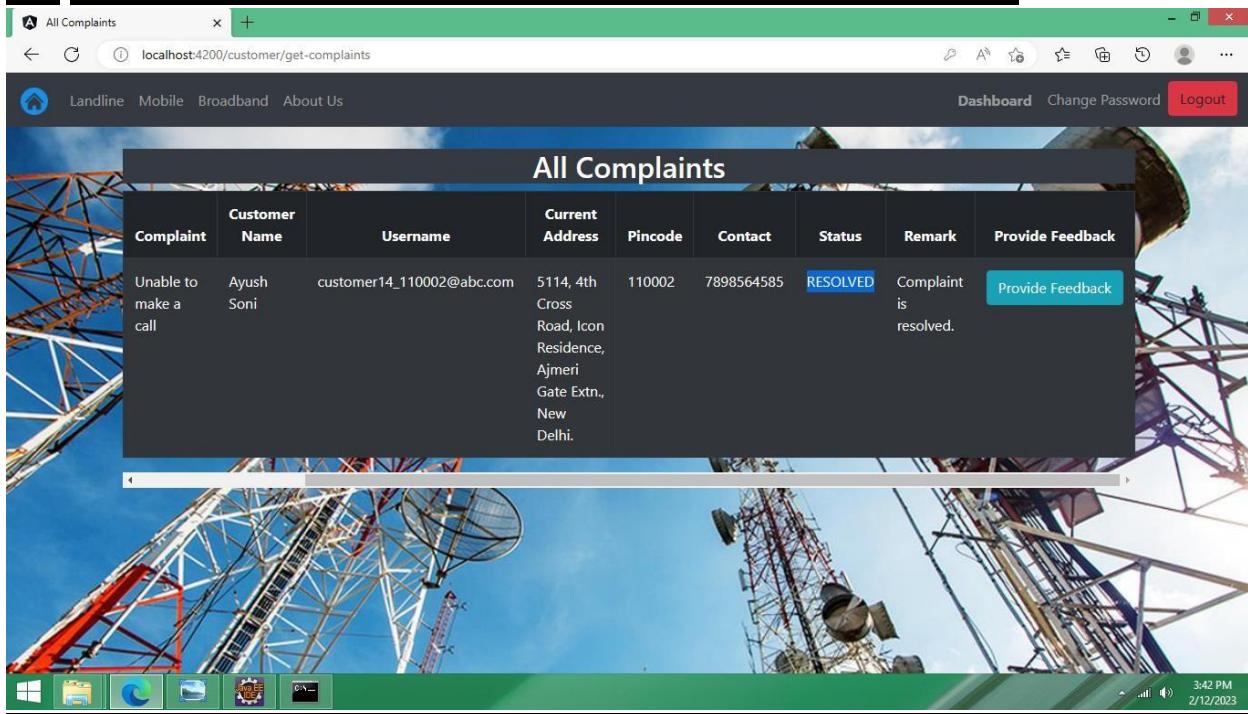
33. Again Engineer Login and Assign a new status:



Status assigned:



34. Customer login and see the new status of complaint:



35. Provide feedback for Resolved or Escalated Complaints:

