SPRINT-2

TEAM ID	PNT2022TMID49560
PROJECT NAME	CUSTOMER CARE REGISTRY

CODE:

Userprofile.html:

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Dashboard</title>
    <link rel="stylesheet" href="../static/profile.css">
    <link rel="shortcut icon" type="image/jpg" href="../static/img3.jpg">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css"/>
  </head>
    <body>
     <div class="navigation">
          <ul>
            <li>
              <a href="#">
                <span class="icon"><ion-icon name="home-outline"></ion-icon></span>
                <span class="title">Customer care Registry</span>
              </a>
            </li>
            <li>
              <a href="/profile">
                <span class="icon"><ion-icon name="person-circle-outline"></ion-icon></span>
                <span class="title">Profile</span>
              </a>
            </li>
            <li>
              <a href="#">
                <span class="icon"><ion-icon name="layers-outline"></ion-icon></span>
                <span class="title">Dashboard</span>
              </a>
            </li>
            <li>
              <a href="/addcomplaint">
                <span class="icon"><ion-icon name="chatbox-ellipses-outline"></ion-
icon></span>
                <span class="title">Lodge Complaint</span>
              </a>
            </li>
            <li>
              <a href="/logout">
                <span class="icon"><ion-icon name="log-out-outline"></ion-icon>
                </span>
                <span class="title">Logout</span>
              </a>
            </li>
          </ul>
          <div class="toggle"></div>
        </div>
        <div class="float-container">
          <div class="float-child1">
         </div>
          <div class="float-child2">
            <div class="card">
                <div class="lines"></div>
                <div class="imgBx">
                    <img src="{{url_for('static',filename='avatar.jpg')}}">
                </div>
        <div class="content">
          <div class="details">
            <h2>{{ fullname }}<br><span>{{ email }}</span></h2>
            <div class="data">
              <h3>5<br><span>Complaints</span></h3>
              <h3>4<br><span>Solved</span></h3>
              <h3>1<br><span>Pending</span></h3>
            </div>
          </div>
        </div>
      </div>
     </div>
     </div>
<script type="module" src="https://unpkg.com/ionicons@5.5.2/dist/ionicons/ionicons.esm.js"></script>
<script nomodule src="https://unpkg.com/ionicons@5.5.2/dist/ionicons/ionicons.js"></script>
<script>
 let navigation=document.querySelector('.navigation');
 let toggle=document.querySelector('.toggle');
 toggle.onclick=function(){
  navigation.classList.toggle('active')
 }

</script>
  </body>
</html>
```

Userprofile.css:

```
@import url('https://fonts.googleapis.com/css?family=Ubuntu:wght@300;400;500;700&display=swap');
*{
 padding: 0;
```

```
  margin: 0;
  box-sizing: border-box;
  font-family: 'Ubuntu',sans-serif;
}
body {
  min-height: 100vh;
  background:black;
}
.navigation{
  position: fixed;
  inset: 20px;
  background: #ed4828;
  width:80px;

  border-left: 10px solid #ed4828;
  border-radius: 50px;
  overflow: hidden;
  box-shadow: 15px 15px 25px rgba(0,0,0,.05);
  transition: 0.5s;
}
.navigation.active{
  width: 300px;
  border-radius: 20px;

}
.toggle{
  position: absolute;
  bottom: 15px;
  right: 15px;
  width: 50px;
  height:50px;
  background: #fff;
  border-radius: 50%;
  box-shadow: 5px 5px 10px rgba(0,0,0,.15);
  cursor: pointer;
  display: flex;
  justify-content: center;
  align-items: center;
}
.toggle::before{
  content: '';
  position: absolute;
  width: 26px;
  height: 3px;
  border-radius: 3px;
  background: #ed4828;
  transform: translateY(-5px);
  transition: 1s;
}
.toggle::after{
  content: '';
  position: absolute;
  width: 26px;
  height: 3px;
  border-radius: 3px;
  background: #ed4828;
  transform: translateY(5px);
  transition: 1s;
}
.navigation.active .toggle::before{
  transform:translateY(0px) rotate(-405deg);
}
.navigation.active .toggle::after{
  transform:translateY(0px) rotate(255deg);
```

```
}
.navigation ul{
  position: absolute;
  top:0;
  left:0;
  width:100%;
}
.navigation ul li{
  position: relative;
  list-style:none ;
  width: 100%;
  border-top-left-radius: 30px;
  border-bottom-left-radius: 30px;
  /*background:#fff;*/
}
.navigation ul li:hover{
  background: #fff;
  color: #222;
}
.navigation ul li:nth-child(1){
  top:20px;
  margin-bottom: 40px;
  background: none;
}
.navigation ul li:not(:first-child):hover::before{
  content: '';
  position: absolute;
  top: -20px;
  right: 0;
  width:20px;
  height:20px;
  background: transparent;
  border-bottom-right-radius: 20px;
  box-shadow: 7.5px 7.5px 0 7.5px #fff;
}
.navigation ul li:not(:first-child):hover::after{
  content: '';
  position: absolute;
  bottom: -20px;
  right: 0;
  width:20px;
  height:20px;
  background: transparent;
  border-top-right-radius: 20px;
  box-shadow: 7.5px -7.5px 0 7.5px #fff;
}


.navigation ul li a{
  position: relative;
  display: block;
  width:100%;
  display: flex;
  text-decoration: none;
  color: #fff;
}
.navigation ul li:hover:not(:first-child) a {
  color: #ed4828;
}

.navigation ul li a .icon{
  position: relative;
  display: block;
  min-width: 60px;
  height: 60px;
  line-height: 70px;
  text-align: center;
}
.navigation ul li a .icon ion-icon{
  font-size: 1.75em;
}
.navigation ul li a .title{
  position: relative;
  display: block;
  padding: 0 10px;
  height: 60px;
  line-height: 60px;
  text-align: start;
  white-space: nowrap;
}
.float-container{
            height: 700px;
            width: 1000px;
            display: inline;
  margin-top: 100px;
  margin-left: 400px;


}

.float-child2{
  float: left;
  margin-left: 600px;
  width: 250px;
 margin-top: 250px;
}
```

```
.float-child2 h1{
            font-size: 50px;
  text-align: center;
   font-weight: bold;
  -webkit-animation: glow 2s ease-in-out infinite alternate;
  -moz-animation: glow 2s ease-in-out infinite alternate;
  animation: glow 2s ease-in-out infinite alternate;
}

@keyframes glow {
     from {
      color: #fff;
    text-shadow: 0 0 10px #ed4828, 0 0 20px #ed4828, 0 0 30px #ed4828, 0 0 40px #ed4828, 
0 0 50px #ed4828, 0 0 60px #ed4828, 0 0 70px #ed4828, 0 0 90px #ed4828;
  }

  to {
     color: gray;
    text-shadow: 0 0 20px #e92424, 0 0 30px #ed1e1e, 0 0 40px #ed2424, 0 0 50px #f72e2e, 0 
0 60px #f02b2b, 0 0 70px #ef2a2a, 0 0 80px #f43535, 0 1 90px #ed2020;
  }
}



.card
{
    position: relative;
    width: 500px;
    height: 190px;
    background: black;

    transition: 0.5s;
```

```
}
.card:hover
{
   height: 400px;
}
.card .lines
{
   position: absolute;
   inset: 0;
   background: #000;
   overflow: hidden;

}
.card .lines::before
{
   content: '';
   position: absolute;
   top: 50%;
   left: 50%;
   transform:  translate( -50%, -50% );
   width: 600px;
   height: 120px;
   background: linear-gradient(transparent,#d5321d,#d5321d,#d5321d,transparent);
   animation: animate 4s linear infinite;
}
@keyframes animate
{
   0%
   {
      transform: translate(-50%, -50%) rotate(0deg);
   }
   100%
```

```
    {
        transform: translate(-50%, -50%) rotate(360deg);
    }

}
.card .lines::after
{
    content: '';
    position: absolute;
    inset: 3px;
    background: #292929;
}
.card .imgBx
{
    position: absolute;
    top: -30%;
    left: 50%;
    transform: translateX(-50%);
    width: 150px;
    height: 150px;
    background: #000;
    transition: 0.5s;
    z-index: 10;
    overflow: hidden;
}
.card:hover .imgBx
{
    width: 200px;
    height: 200px;
}
.card .imgBx::before
{
    content: '';
    position: absolute;
    top: 50%;
    left: 50%;
    width: 500px;
    height: 150px;
    transform: translate(-50%, -50%);
    background: linear-gradient(transparent,white,white,white,transparent);
    animation: animate2 6s linear infinite;
}
@keyframes animate2
{
    0%
    {
        transform: translate(-50%, -50%) rotate(360deg);
    }
    100%
    {
        transform: translate(-50%, -50%) rotate(0deg);
    }
}
.card .imgBx::after
{
    content: '';
    position: absolute;
    inset: 3px;
    background: #292929;
}
.card .imgBx img
{
    position: absolute;
    top: 10px;
    left: 10px;
    z-index: 1;
    width: calc(100% - 20px);
    height: calc(100% - 20px);
    /*filter: grayscale(1);*/
}
.card .content
{
    position: absolute;
    width: 100%;
    height: 100%;
    display: flex;
    justify-content: center;
    align-items: flex-end;
    overflow: hidden;
}
.card .content .details
{
    padding: 40px;
    text-align: center;
    width: 100%;
    transition: 0.5s;
    transform: translateY(145px);
}
.card:hover .content .details
{
    transform: translateY(0px);

}
.card .content .details h2
{
    font-size: 1.25em;
    font-weight: 600;
    color: #f64f32;
    line-height: 1.2em;
    margin-bottom: 70px;
}
.card .content .details h2 span
{
    font-size: 0.75em;
    font-weight: 500;
    color: #fff;
}
.card .content .details .data
{
    display: flex;
    justify-content: space-between;
    margin: 20px 0;
}
.card .content .details .data h3
{
    font-size: 1em;
    color: #f64f32;
    line-height: 1.2em;
    font-weight: 600;
}
.card .content .details .data h3 span
{
    font-size: 0.85em;
    font-weight: 400;
    color: #fff;
}
.card .content .details .actionBtn
{
    display: flex;
    justify-content: space-between;
    gap: 20px;
}
.card .content .details .actionBtn button
{
    padding: 10px 30px;
    border: none;
    outline: none;
    border-radius: 5px;
    font-size: 1em;
    font-weight: 500;
    background: #45f3ff;
    color: #222;
    cursor: pointer;
    opacity: 0.9;
}
.card .content .details .actionBtn button:nth-child(2)
{
    background: #fff;
}
.card .content .details .actionBtn button:hover
{
    opacity: 1;
}
```

ComplaintRegister.html:

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Complaint registration form</title>
    <link rel="stylesheet" href="../static/bootstrap.min.css">
  <link rel="stylesheet" href="../static/addcomplaint.css">
  <link rel="shortcut icon" type="image/jpg" href="../static/img3.jpg">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css"/>
 </head>
  <body>


    <div class="float-container">
     <div class="float-child1">

        <form class="signup-form" action="/addcomplaint" method="post">

          <!-- form header -->
          <div class="form-header">
            <div class="form-header">
              <h1>Create Account</h1>
             </div>
          </div>

          <!-- form body -->
          <div class="form-body">
            <div class="form-body">

              <!-- Firstname and Lastname -->
              <div class="horizontal-group">
                <div class="form-group left">
                  <label for="firstname" class="label-title">Select Category</label>
                  <select name="selectcategory" required>
                    <option  value="category"></option>
                    <option  value="E-commerse">E-commerse</option>
                    <option  value="general">General</option>
```

```
                              </select>
                            </div>
                            <div class="form-group right">
                              <label for="lastname" class="label-title">Date</label>
                              <input type="date" name="dt">
                            </div>
                          </div>
                          <!-- Email -->
<div class="form-group">
   <label for="email" class="label-title">Email*</label>
   <input type="email" id="email"name="email" class="form-input" placeholder="enter your
email" required="required">
  </div>

  <!-- Password and confirm password -->
  <div class="horizontal-group">

   <div class="form-group left">
     <label for="password" class="label-title">Phone number*</label>
     <input  id="password" class="form-input" placeholder="enter your phone number"
name="phone" required="required">
   </div>

   <div class="form-group right">
     <label for="confirm-password" class="label-title">additional *</label>
     <input  class="form-input" id="confirm-password" placeholder="enter your opinion"
name="additional" required="required">
   </div>
   <!-- Gender and Hobbies -->
<div class="horizontal-group">

   <div class="form-group left">
       <label class="label-title">Status:</label>
```

```
        <div class="input-group">
            <label for="male">
                <input type="radio" name="status" value="male" id="male"> pending</label>
            <label for="female">
                <input type="radio" name="status" value="female" id="female">solved</label>
        </div>
    </div>




</div>
 <!-- Source of Income and Income Amount -->
 <div class="horizontal-group">

    <div class="form-group left" >
      <label class="label-title">State</label>
      <select class="form-input" name="state"id="level" >
        <option value="Tamil nadu">Tamil nadu</option>
        <option value="Kerala">Kerala</option>
        <option value="others">others</option>
      </select>
    </div>




  </div>
  <!-- Bio -->
<div class="form-group">
    <label for="choose-file" class="label-title">complaint</label>
    <textarea class="form-input"name="complaint" rows="4" cols="50"
style="height:auto"></textarea>
  </div>
  </div>
                </div>

                <!-- form footer -->
                <div class="form-footer">
                    <!-- form footer -->
<div class="form-footer">
    <span>* required</span>

    <button type="submit" onclick=display() class="btn">Create</button>
  </div>
                </div>

             </form>

          </div>




        </div>




      </div>


      <script type="module"
src="https://unpkg.com/ionicons@5.5.2/dist/ionicons/ionicons.esm.js"></script>
<script nomodule
src="https://unpkg.com/ionicons@5.5.2/dist/ionicons/ionicons.js"></script>
```

```
<!-- Script for range input label -->
<script>
    var rangeLabel = document.getElementById("range-label");
    var experience = document.getElementById("experience");
    function change() {
    rangeLabel.innerText = experience.value + "%pending";
    }
  </script>
<script>
  let navigation=document.querySelector('.navigation');
  let toggle=document.querySelector('.toggle');
  toggle.onclick=function(){
    navigation.classList.toggle('active')
  }
  function display(){
    alert("you successfully registered the complaints")
  }

</script>
    </body>
</html>
```

ComplaintRegister.css:

```
@import url('httpss://fonts.googleapis.com/css?family=Roboto');

body {
  background:black
}
.float-container{
            height: 700px;
            width: 1000px;
            display: inline;
  margin-top: 100px;
  margin-left: 400px;



}
.float-child1{
  float: right;


  margin-top: 40px;
  margin-right:300px ;


  width: 800px;




}




@keyframes glow {
     from {
      color: #fff;
    text-shadow: 0 0 10px #ed4828, 0 0 20px #ed4828, 0 0 30px #ed4828, 0 0 40px #ed4828,
0 0 50px #ed4828, 0 0 60px #ed4828, 0 0 70px #ed4828, 0 0 90px #ed4828;
  }


  to {
     color: gray;
    text-shadow: 0 0 20px #e92424, 0 0 30px #ed1e1e, 0 0 40px #ed2424, 0 0 50px #f72e2e, 0
0 60px #f02b2b, 0 0 70px #ef2a2a, 0 0 80px #f43535, 0 1 90px #ed2020;
  }
}
```

```
.signup-form {
  font-family: "Roboto", sans-serif;
  width:650px;
  margin-right: 600px;
  background:linear-gradient(to right, #050505 0%, #090909 50%, #060606 99%);
  border-radius: 10px;
  border-style: solid;
  border-color: #ed1e1e;
  border-width: 1px;
}

.form-header  {
  background-color: #060606;
  border-top-left-radius: 10px;
  border-top-right-radius: 10px;
}

.form-header h1 {
  font-size: 30px;
  text-align:center;
  color:#ed1e1e;
  padding:20px 0;
  border-bottom:1px solid #ed1e1e;
}
/*----------------------------------------*/
/* Form Body */
/*----------------------------------------*/
.form-body {
    padding:10px 40px;
    color:#ed1e1e;
  }
```

```
.form-group{
  margin-bottom:20px;
}

.form-body .label-title {
  color:#ed1e1e;
  font-size: 17px;
  font-weight: bold;
}

.form-body .form-input {
    font-size: 17px;
    box-sizing: border-box;
    width: 100%;
    height: 34px;
    padding-left: 10px;
    padding-right: 10px;
    color: #ed1e1e;
    text-align: left;
    border: 1px solid #ed1e1e;
    border-radius: 4px;
    background: black;
    outline: none;
}



.horizontal-group .left{
  float:left;
  width:49%;
}
```

```
.horizontal-group .right{
  float:right;
  width:49%;
}

input[type="file"] {
 outline: none;
 cursor:pointer;
 font-size: 17px;
}

#range-label {
 width:15%;
 padding:5px;
 background-color: #ed1e1e;
 color:white;
 border-radius: 5px;
 font-size: 17px;
 position: relative;
 top:-8px;
}


::-webkit-input-placeholder {
 color:white;
}

/*---------------------------------------*/
/* Form Footer */
/*---------------------------------------*/
.form-footer {
   clear:both;
  }
  /*--------------------------------------*/
/* Form Footer */
/*--------------------------------------*/
.signup-form .form-footer  {
    background-color: black;
    border-bottom-left-radius: 10px;
    border-bottom-right-radius: 10px;
    padding:10px 40px;
    text-align: right;
    border-top: 1px solid #ed1e1e;
  }

  .form-footer span {
    float:left;
    margin-top: 10px;
    color:#ed1e1e;
    font-style: italic;
    font-weight: thin;
  }

  .btn {
     display:inline-block;
     padding:10px 20px;
     background-color: #ed1e1e;
     font-size:17px;
     border:none;
     border-radius:5px;
     color:#bcf5e7;
     cursor:pointer;
  }
  .btn:hover {
    background-color: red;
    color:white;
  }
```

AgentCreation.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
        <meta charset="UTF-8">
        <title>Agent Creation</title>

    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="../static/login.css">

</head>
<body>

        <div class="container" >
                <div class="row">
                        <div class="column1">



                        </div>
                        <div class="column2">
                                <div class="contact-form">
                                        <img  src="{{url_for('static',filename='a1.jpg')}}"
alt="" class="avatar">
                                        <h2> Agent Creation</h2>
                                        <form action="/agentdata" method="POST">
                                                <p>email</p><input placeholder="Enter your
name" type="text" name='username'>
```

```
                              <p>Password</p><input placeholder="Enter password"
type="password" name='password'>
                              <input type="submit" value="Create">


                                                  </form>
                                                  <span class="alert {{indicator}}">{{a}}</span>


                                                  <div class="links">
                                                            <p style="margin-top:8px;">click here for
dashboard <a href="/adminpage"> Click here</a></p>
                                                  </div>
                                        </div>


                              </div>
                    </div>
          </div>
AgentCreation.css:
body {
          margin: 0;
          padding: 0;
          font-family: 'Poppins', sans-serif;
          background-color: black;


}
.row{


          height: 700px;
          width: 1300px;
          margin-top: 10px;


          border-radius: 20px;
}
.column1{
```

```
            float: left;
            width: 900px;



            height: 650PX;
            background-image:url(../static/login.svg);


            background-repeat: no-repeat;
            background-attachment: fixed;
    object-fit: cover;
            background-position: left;




}
.column2{
            float: right;
            width: 50%;

}

.hello {
            position: fixed;
            width: 100vw;
            height: 100vh;
            background-position: center center;
            background-repeat: no-repeat;
            background-attachment: fixed;
            -webkit-background-size: cover;
            background-size: cover;
            -webkit-filter: blur(10px);
            -moz-filter: blur(10px);
```

```
        z-index: -1;

}
.contact-form {
        position: absolute;
        top: 50%;
        left: 50%;
        transform: translate(-50%, -50%);
        width: 300px;
        height: 350px;
        padding: 80px 40px;
        background: rgba(0, 0, 0, 0.3);
        border-width: 1px;
        border-color: #ed4828;
        border-style: solid;
        border-radius: 20px;
        margin-left: 400px;
        box-shadow: 0 0 8px 0 rgb(234, 13, 13);
}
.avatar {
        position: absolute;
        width: 90px;
        height: 80px;
        border-radius: 50%;
        overflow: hidden;
        top: calc(-80px/2);
        left: 150px;
}
.contact-form h2 {
        margin: 0;
        padding: 0 0 20px;
        color: #fff;
        text-align: center;
        text-transform: uppercase;
}
.contact-form p {
        margin: 0;
        padding: 0;
        font-weight: bold;
        color: #fff;
}
.contact-form input {
        width: 100%;
        margin-bottom: 20px;
}
 .contact-form input[type="password"] {
        border: none;
        border-bottom: 1px solid #fff;
        background: transparent;
        outline: none;
        height: 40px;
        color: #fff;
        font-size: 16px;
}
.contact-form input[type="text"]{
    border: none;
        border-bottom: 1px solid #fff;
        background: transparent;
        outline: none;
        height: 40px;
        color: #fff;
        font-size: 16px;
}
.contact-form input[type="password"]:focus{
```

```
            border-color: black;

}
.contact-form input[type="text"]:focus{
            border-color: black;

}

.contact-form input[type=submit] {
            height: 30px;
            color: #fff;
            font-size: 15px;
            background:#ed4828;
            cursor: pointer;
            border-radius: 25px;
            border: none;
            outline: none;
            margin-top: 15%;
}
.contact-form input[type=submit]:hover{
    background-color:rgb(251, 116, 116) ;
            transition: 1s ease;

}
.alert {
    width:100%;
    height:40px;
    color: white;
    padding: 5px 10px;
    border-radius: 10px;
    font-size:14px;
```

```
}
.success {
    color:green;
    font-weight:bold;
    border: 2px solid green;
}

.failure {
    color:red;
    font-weight:bold;
    border: 2px solid red;
}
.links a{
            color: #ed4828;

}
```

AgentLogin.html:

```
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>Agent Login Form</title>
        <link rel="stylesheet" href="../static/agentlogin.css">
    </head>
<body>
        <div class="box">
            <div class="form">
                <form action="/logindata" method="POST">
                <h2>Agent Login</h2>
               <div class="inputBox">
                <input type="text" name="username"required="required">
                <span>Email</span>
                <i></i>
        </div>
        <div class="inputBox">
          <input type="password" name="password"required="required">
          <span>Password</span>
          <i></i>
        </div>

        <input type="submit" value="Login">
        </form>
     </div>
    </div>
</body>
</html>
```

AgentLogin.css:

```
@import url('https://fonts.googleapis.com/css2?
family=Poppins:wght@300;400;500;600;700;800;900& display=swap');
*
{
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: 'Poppins', sans-serif;
}

body
{
    display: flex;
    justify-content: center;
    align-items: center;
    min-height: 100vh;
    background: black;
}
```

```
.box
{
   position: relative;
   width: 380px;
   height: 420px;
   background: #2e2d2d;
   border-radius: 8px;
   overflow: hidden;
}

.box::before
{
   content: '';
   position: absolute;
   top: -50%;
   left: -50%;
   width: 380px;
   height: 420px;
   background: linear-gradient(0deg,transparent,
   transparent,#ed4828);
   animation: animate 6s linear infinite;
   transform-origin: bottom right;


}
.box::after
{
   content: '';
   position: absolute;
   top: -50%;
   left: -80%;
   width: 380px;
   height: 420px;
```

```
    background: linear-gradient(0deg,transparent,
    transparent,#ed4828);
    animation: animate 6s linear infinite;
    animation-delay: -3s;
    transform-origin: bottom right;

}

@keyframes animate
{
    0%
    {
        transform: rotate(0deg);

    }
    100%
    {
        transform: rotate(360deg);
    }
}

.form
{
    position: absolute;
    inset: 2px;
    border-radius: 8px;
    background: black;
    z-index: 10;
    padding: 50px 40px;
    display: flex;
    flex-direction: column;
}
```

```
.form h2
{
   color: #ed4828;
   font-weight: 500;
   text-align: center;
   letter-spacing: 0.1em;
}

.inputBox
{
   position: relative;
   width: 300px;
   margin-top: 35px;
}

.inputBox input
{
   position: relative;
   width: 100%;
   padding: 20px 10px 10px;
   background: transparent;
   border: none;
   outline: none;
   color: #ed4828;
   font-size: 1em;
   letter-spacing: 0.05em;
   z-index: 10;
}

.inputBox span
{
   position: relative;
   left: 0;
   padding: 20px 0px 10px;
   font-size: 1em;
   color: #ed4828;
   pointer-events: none;
   letter-spacing: 0.05em;
   transition: 0.5s;
}

.inputBox input:valid ~ span,
.inputBox input:focus ~ span
{
   color: #ed4828;
   transform: translateX(0px) translateY(-34px);
   font-size: 0.75em;
}

.inputBox i
{
   position: absolute;
   left: 0;
   bottom: 0;
   width: 100%;
   height: 2px;
   background: #ed4828;
   border-radius: 4px;
   transition: 0.5s;
   pointer-events: none;
   z-index: 9;
}
```

```
.inputBox input:valid ~ i,
.inputBox input:focus ~ i
{
    height: 44px;
}

.links
{
    display: flex;
    justify-content: space-between;
}

.links a
{
    margin: 10px 0;
    font-size: 0.75em;
    color: #ed4828;
    text-decoration: none;
}

.links a:hover,
.links a:nth-child(2)
{
    color: #ed4828;
}

input[type = "submit"]
{
    border: none;
    outline: none;
    background:#ed4828;
    padding: 11px 25px;
    width: 100px;
    margin-top: 40px;
    border-radius: 4px;
    font-weight: 600;
    cursor: pointer;
}

input[type = "submit"]:active
{
    opacity: 0.8;

}
```

App.py:

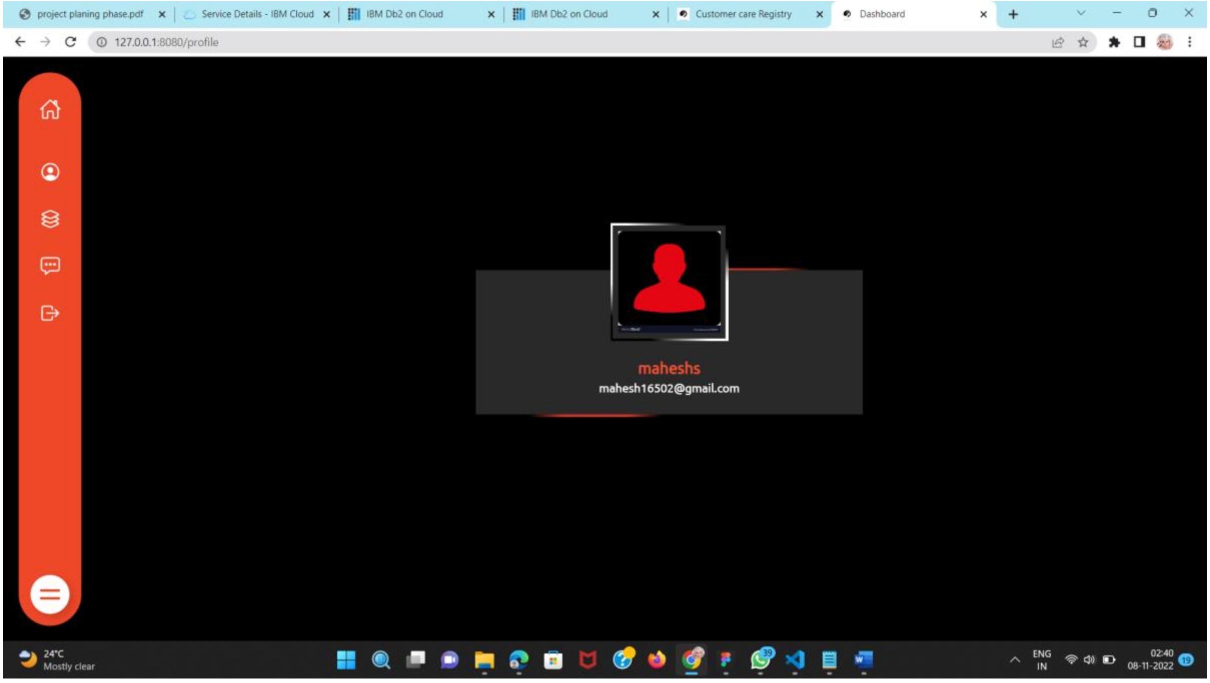
```
from flask import Flask, render_template, request, redirect, url_for, session
import ibm_db
import re
app = Flask(__name__)

app.secret_key='a'

conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud; PORT=32459; Security=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=dmz49400; PWD=gyRzSBXn1Biv7bk5;",",")



@app.route('/')
def home():
    return render_template('home.html')
@app.route('/registertemp',methods=["POST","GET"])
def registertemp():
    return render_template("register.html")
@app.route('/uploaddata',methods =['GET','POST'])
```

```
def register():
    msg = ''
    if request.method == 'POST':
        firstname = request.form['firstname']
        lastname = request.form['lastname']
        username = request.form['username']
        email = request.form['email']
        password = request.form['password']
        address = request.form['address']
        stmt = ibm_db.prepare(conn, 'SELECT * FROM user WHERE username = ?')
        ibm_db.bind_param(stmt, 1, username)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        if account:
            msg = 'Account already exists !'
        elif not re.match(r'[^@]+@[^@]+\.[^@]+', email):
            msg = 'Invalid email address !'
        elif not re.match(r'^[A-Za-z0-9_.-]*$', username):
            msg = 'name must contain only characters and numbers !'
        else:
            prep_stmt = ibm_db.prepare(conn,'INSERT INTO user(firstname, lastname,
username, email, password, address) VALUES(?, ?, ?, ?, ?, ?)')
            ibm_db.bind_param(prep_stmt, 1, firstname)
            ibm_db.bind_param(prep_stmt, 2, lastname)
            ibm_db.bind_param(prep_stmt, 3, username)
            ibm_db.bind_param(prep_stmt, 4, email)
            ibm_db.bind_param(prep_stmt, 5, password)
            ibm_db.bind_param(prep_stmt, 6, address)
            ibm_db.execute(prep_stmt)
            msg = 'Dear % s You have successfully registered!'%(username)
        return render_template('register.html',a = msg,indicator="success")
    return render_template('register.html')
```

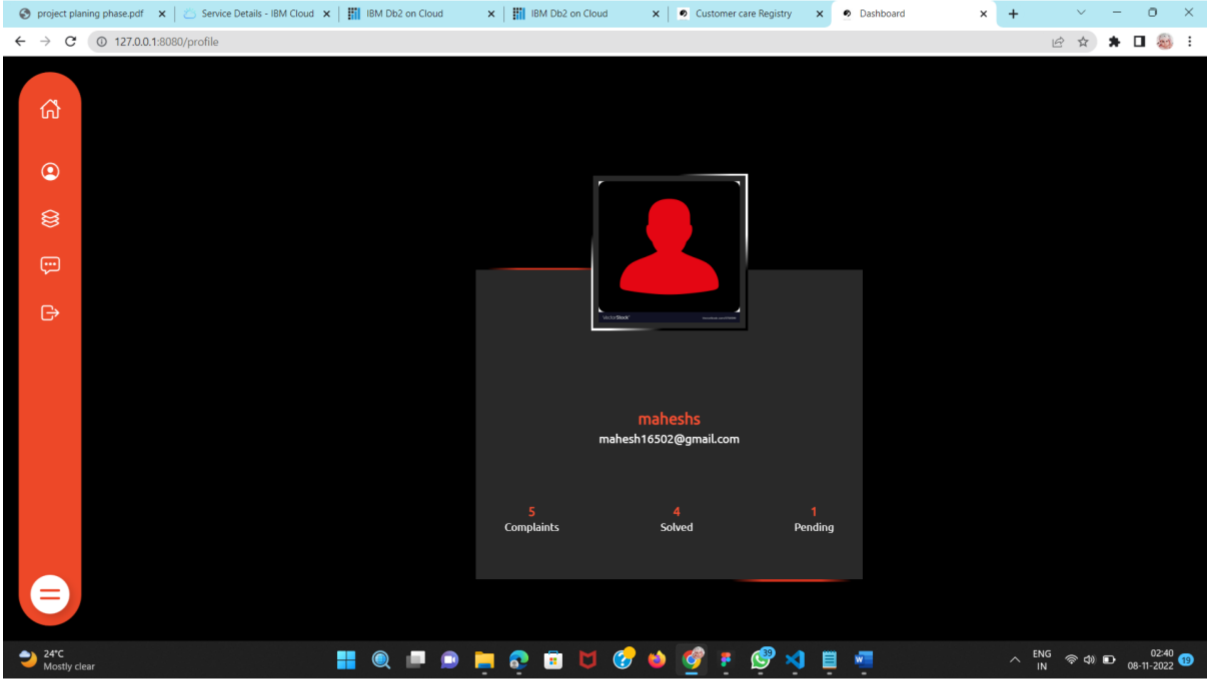
```
@app.route('/login',methods=["POST","GET"])
def login():
    return render_template("login.html")
@app.route('/logindata',methods=["POST","GET"])
def logindata():
    global userid
    msg = ''
    if request.method == 'POST' and 'username' in request.form and 'password' in
request.form:
        username = request.form['username']
        password = request.form['password']
        stmt = ibm_db.prepare(conn,'SELECT * FROM user WHERE username = ? AND
password = ?')
        ibm_db.bind_param(stmt,1,username)
        ibm_db.bind_param(stmt,2,password)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_tuple(stmt)
        if account:
            session['id'] = account[0]
            userid =  account[0]
            session['username'] = account[1]
            return redirect(url_for('dashboard'))
        else:
            msg = 'Incorrect username / password !'
            return render_template('login.html', b = msg, indicator="failure")


@app.route('/home')
def dashboard():
    if 'id' in session:
        uid = session['id']
        stmt = ibm_db.prepare(conn, 'SELECT * FROM user WHERE id = ?')
        ibm_db.bind_param(stmt, 1, uid)
        ibm_db.execute(stmt)
        ibm_db.fetch_tuple(stmt)
        username = session['username']
        return render_template('user dashboard.html', name = username)



@app.route('/profile',methods=["POST","GET"])
def profile():
    if 'id' in session:
        uid = session['id']
        stmt = ibm_db.prepare(conn, 'SELECT * FROM user WHERE id = ?')
        ibm_db.bind_param(stmt, 1, uid)
        ibm_db.execute(stmt)
        acc = ibm_db.fetch_tuple(stmt)
        return
render_template('userprofile.html',fullname=acc[1]+acc[2],email=acc[4],address=acc[6])
    return render_template('userprofile.html')

@app.route('/admin')
def admin():
    return render_template('admin.html')

@app.route('/adminpage')
def adminpage():
    return render_template('admin dashboard.html')

@app.route('/adminlog',methods=["POST","GET"])
def adminlog():
    msg = ''
    email = request.form['email']
    password = request.form['password']
    stmt = ibm_db.prepare(conn, 'SELECT * FROM agentinfo  WHERE email = ?  and
password = ?')
    ibm_db.bind_param(stmt,1, email)
    ibm_db.bind_param(stmt,2, password)
    ibm_db.execute(stmt)
    logged = ibm_db.fetch_assoc(stmt)
    if(logged):
        msg = 'successfully loggedin'
        return render_template("admin dashboard.html",a=msg)
    else:
        return render_template("admin.html",a="Incorrect email/password")
@app.route('/addcomplaint',methods=["POST","GET"])
def complaint():
    if request.method == "POST":
        if 'id' in session:
            msg = ''
            uid=session['id']
            selectcategory = request.form['selectcategory']
            date = request.form['dt']
            phone = request.form['phone']
            additional = request.form['additional']
            state = request.form['state']
            email = request.form['email']
            complaint = request.form.get('complaint')
            stmt = ibm_db.prepare(conn,"SELECT * FROM user WHERE id = ?")
            ibm_db.bind_param(stmt, 1, uid)
            ibm_db.execute(stmt)
            ibm_db.fetch_assoc(stmt)
            prep_stmt = ibm_db.prepare(conn,'INSERT INTO complaintdetails(uid,
selectcategory, dt , phone, additional, state, email, complaint, status) VALUES (?, ?, ?, ?, ?, ?,
?, ?, ?)')
            ibm_db.bind_param(prep_stmt, 1, uid)
            ibm_db.bind_param(prep_stmt, 2, selectcategory)
            ibm_db.bind_param(prep_stmt, 3, date)
            ibm_db.bind_param(prep_stmt, 4, phone)
            ibm_db.bind_param(prep_stmt, 5, additional)
            ibm_db.bind_param(prep_stmt, 6, state)
            ibm_db.bind_param(prep_stmt, 7, email)
            ibm_db.bind_param(prep_stmt, 8, complaint)
            ibm_db.bind_param(prep_stmt, 9, 'pending')
            ibm_db.execute(prep_stmt)
            msg = 'You have successfully registered your complaint'
            return render_template('userlodgecomp.html',a = msg)
    return render_template('userlodgecomp.html')
@app.route('/agent',methods=["POST","GET"])
def agent():
    return render_template('agent.html')
@app.route('/loggout')
def loggout():
    if 'id ' in session:
        session.pop('id',None)
        session.pop('email',None)
        session.pop('password',None)
    return redirect(url_for('home'))
@app.route('/logout')
def logout():
    session.pop('loggedin', None)
    session.pop('id', None)
    session.pop('username', None)
    return redirect(url_for('login'))

@app.route('/agentdata',methods=["POST","GET"])
def agentdata():
    msg = ''
    username = request.form['username']
    password = request.form['password']
    stmt = ibm_db.prepare(conn,'INSERT INTO agent(username, password) VALUES (?, ?)')
    ibm_db.bind_param(stmt, 1, username)
    ibm_db.bind_param(stmt, 2, password)
    ibm_db.execute(stmt)
    msg = 'Agent has been created successfully'
    return render_template('agent.html',a = msg,indicator="success")

@app.route('/agentlogin',methods=["POST","GET"])
def agentlogin():
    return render_template('agentlogin.html')

if __name__ == '__main__':
    app.debug=True
    app.run(host='0.0.0.0',port=8080)
```

Output:

Profile Page:



mahesh
mahesh16502@gmail.com



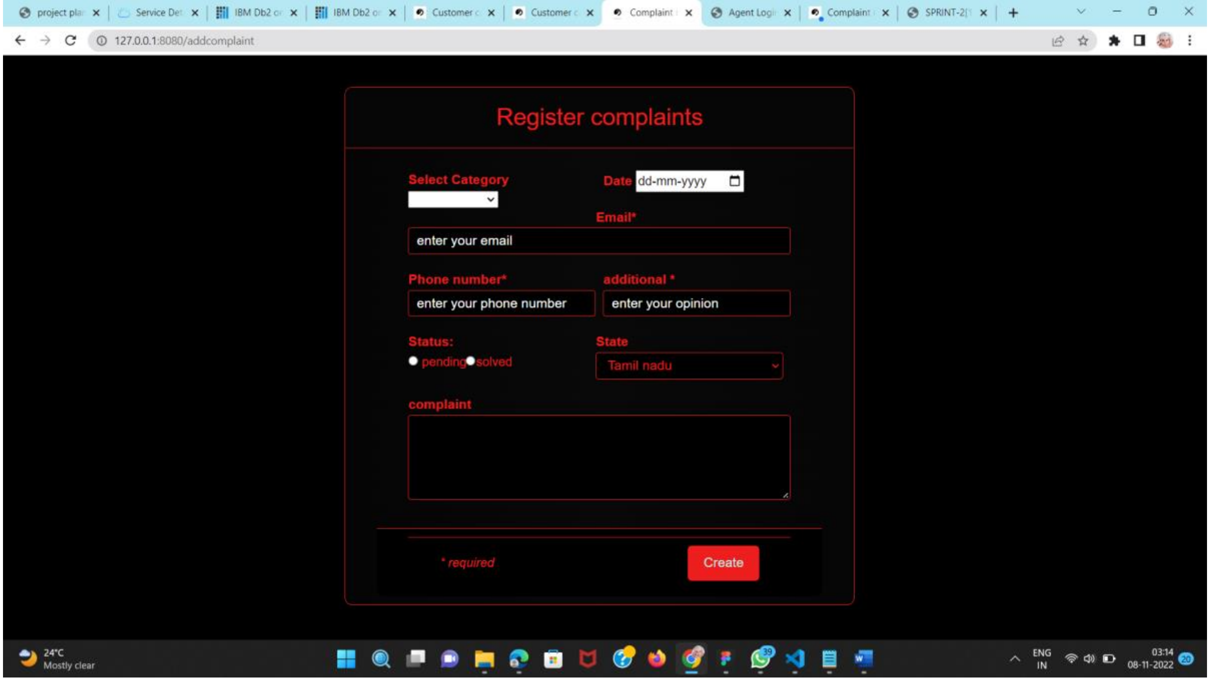
mahesh
mahesh16502@gmail.com

5
Complaints

4
Solved

1
Pending

Complaint registered Page:



Register complaints

Select Category

Date dd-mm-yyyy

Email*

enter your email

Phone number*

enter your phone number

additional *

enter your opinion

Status:

● pending ● solved

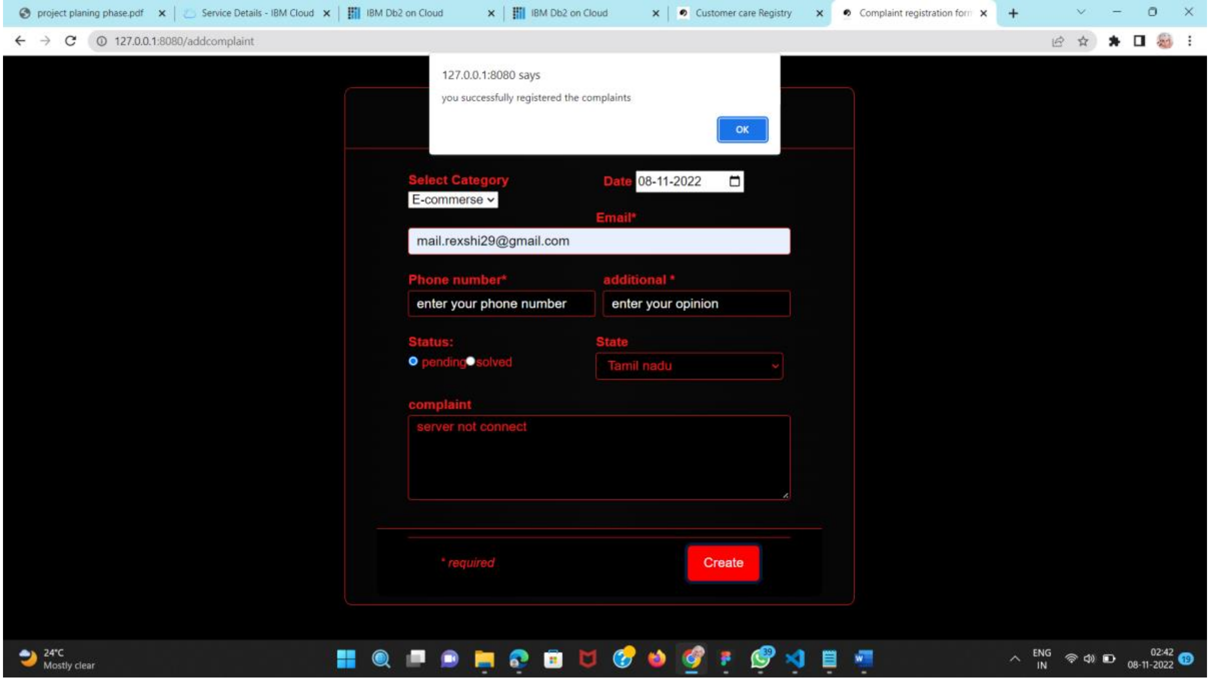
State

Tamil nadu

complaint

* required

Create


project planing phase.pdf
Service Details - IBM Cloud
IBM Db2 on Cloud
IBM Db2 on Cloud
Customer care Registry
Complaint registration forr
127.0.0.1:8080/addcomplaint
24°C
Mostly clear
ENG
IN
02:42
08-11-2022

127.0.0.1:8080 says

you successfully registered the complaints

OK

Select Category

E-commerse

Date 08-11-2022

Email*

mail.rexshi29@gmail.com

Phone number*

enter your phone number

additional *

enter your opinion

Status:

● pending ● solved

State

Tamil nadu

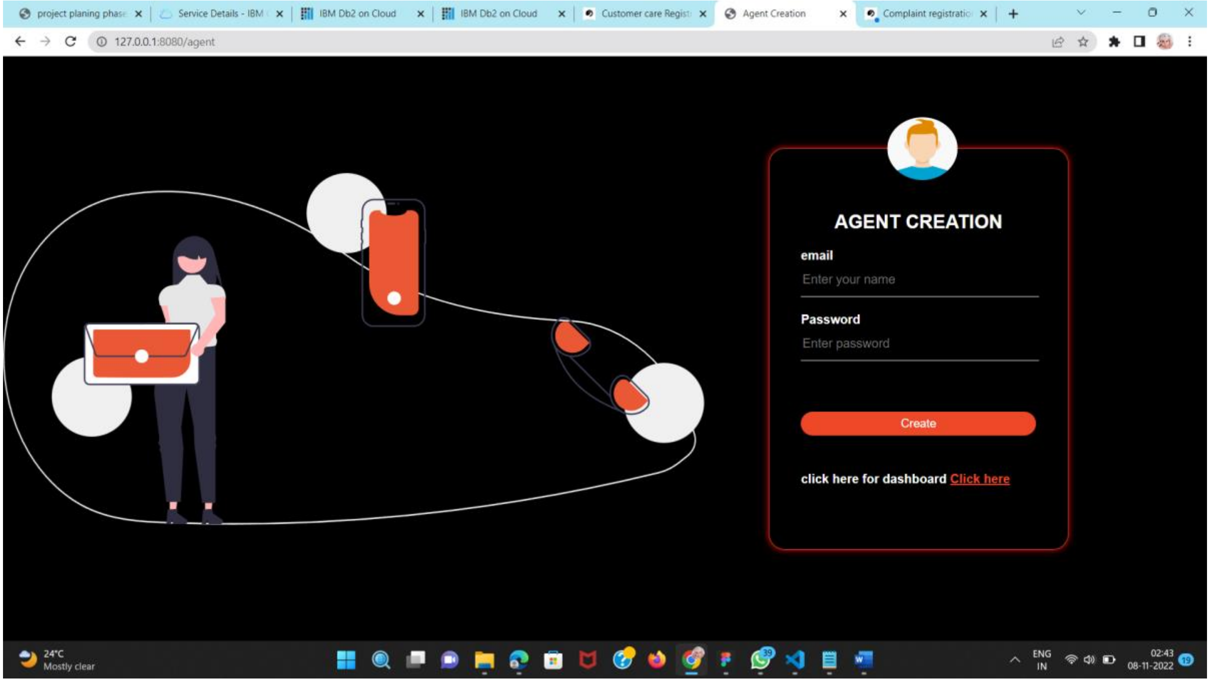
complaint

server not connect

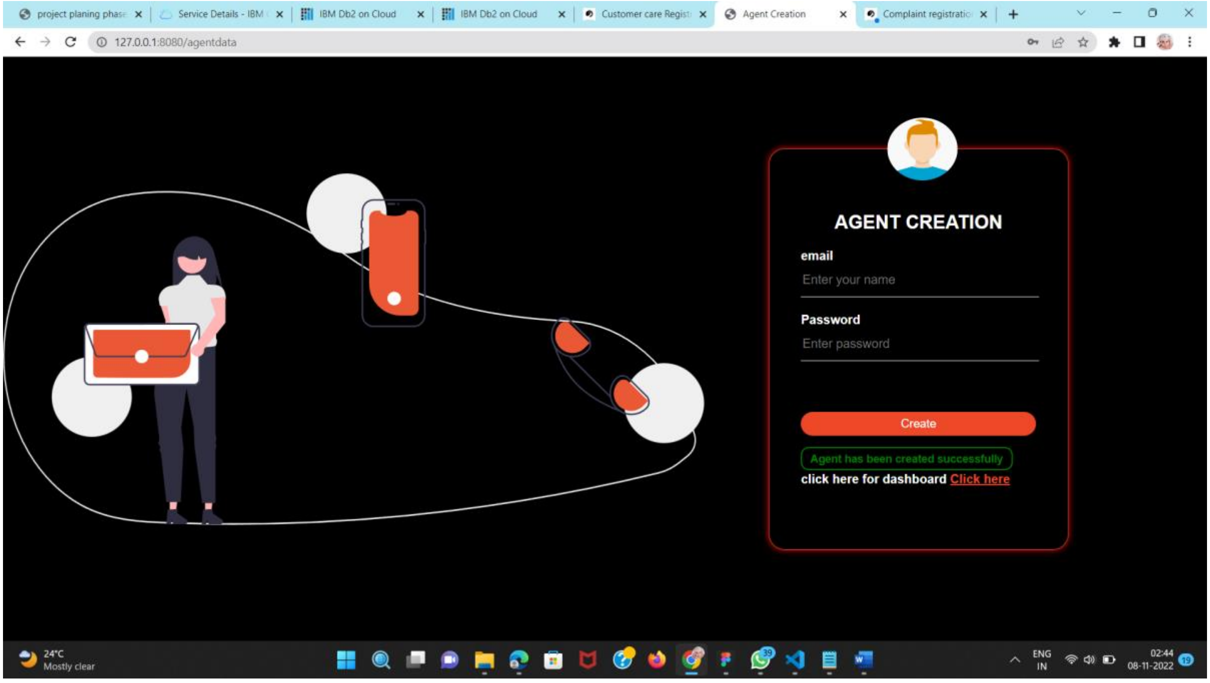
* required

Create

Agent creation page:



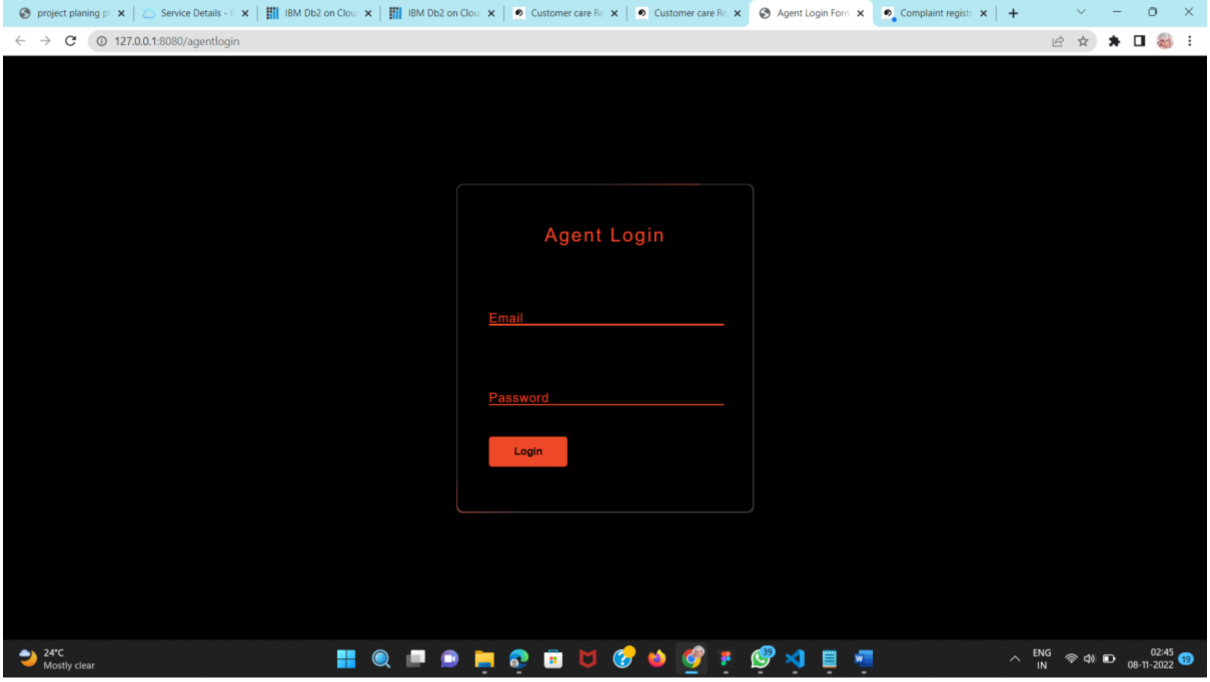
click here for dashboard Click here



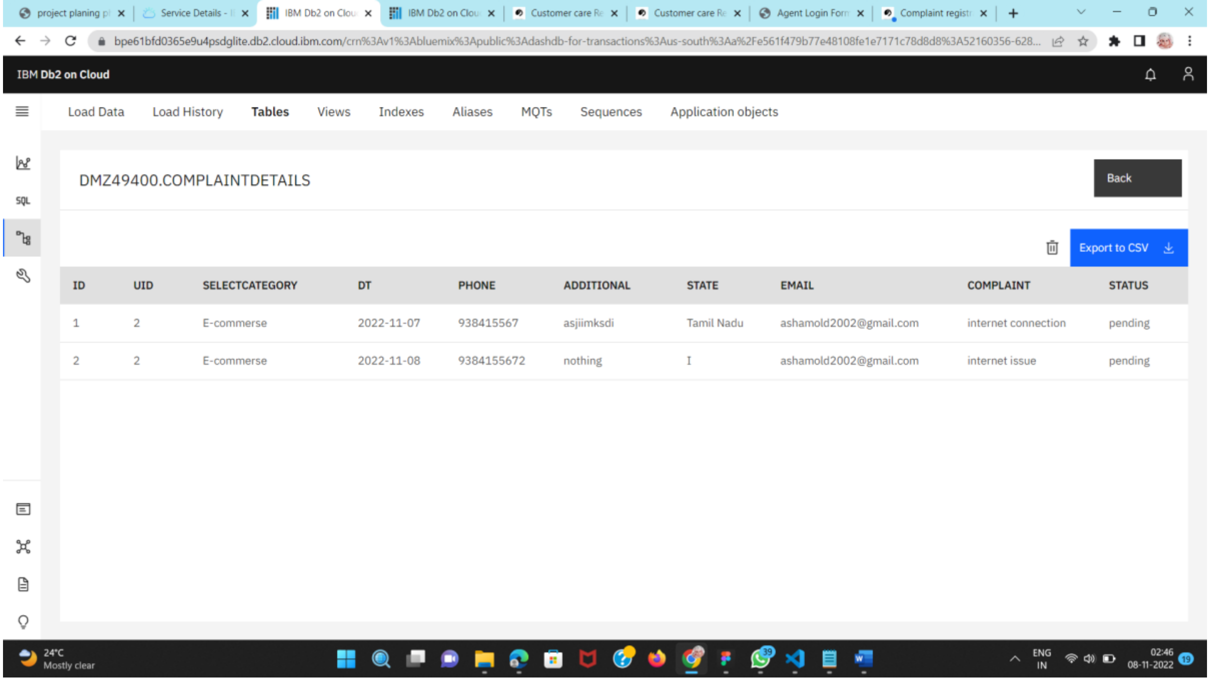
Agent has been created successfully

click here for dashboard Click here

Agent login page:



Complaint database:



DMZ49400.COMPLAINTDETAILS

ID	UID	SELECTCATEGORY	DT	PHONE	ADDITIONAL	STATE	EMAIL	COMPLAINT	STATUS
1	2	E-commerse	2022-11-07	938415567	asjiimksdi	Tamil Nadu	ashamold2002@gmail.com	internet connection	pending
2	2	E-commerse	2022-11-08	9384155672	nothing	I	ashamold2002@gmail.com	internet issue	pending