

# Sprint3

**TeamID:PNT2022TMID53883**

**TeamMates:**

- 1.Ramakrishnan  
P(TeamLead)
- 2.Sundaram VA**
- 3.Sabeek Rahman**
- 4.Nambidurai**

**Sprint3:**

- 1.AdminDashboard
- 2.AgentDashboard
- 3.UserDashboard

**Output:**

- 1.Signup

- 2.SignIn

### **3.Userdashboard**

#### **4.Admindashboard**

**5.Agentdetailspage**

**6.Assignagentpage**

## 7.Agentdashboard

## Code:

### Admin.html

```
{%extends'base.html'%}
```

```
{%blockhead%}
```

```
<title>
AdminDashboard
</title>
```

```
{%endblock%}
```

```
{%blockbody%}
```

```
<!--things
```

```
div1
welcomej etson,signout
```

```
div2
yourcomp laintsstatus
```

```
add newc omplaint-->
<br>
!-- <br>
{% fori inrange(11)%}
```

```
{{i}}
{%endfor %}
```

```
<br>
{%f{oriincomplaints%}
{i<b['USERNAME']}}
r>{%
f orjin i.values()%}
{{j}}ndf
{%e<b or%}
r>
{%e ndfor%}-->
```

```
<divclass ="fordashboardtop">
<divc lass="fordashboardtopelements1">
WelcomeAdmin,
</div>
<divc lass="fordashboardtopelements2">
<a href="/login"><buttonclass="fo rbutton">Sign
out</button></a> </div>
```

```
</v>
i<br
> class="outerofdashdetails">
```

```
<divcla ss="fordashboarddetails">
<br><
!--tableleofcustomerscomp="faints-->
<tblas<thesortable">
d></
thead >
<tbody>
<tr>
<tdc lass="pad">a
<href="/ag ents">AgentDetails</a>
</td>
<tdc lass="pad">a
<href="/ti ckets">CustomerTicketDetails</a>
</td>
</tr>
</tbody>
```

```
</table>
```

```
<br>
```

```
</div>
```

```
</div>
```

```
{%endblock%}
```

## Agentdas.html

```
{%extends 'base.html'%}
```

{%blockhead%}

```
<title>  
AgentDashboard  
</title>
```

```
{%endblock%}
```

```
{%blockbody%}
```

```
<!--things
```

```
div1  
welcomej etson,signout
```

```
div2  
yourcomp laintsstatus
```

```
add<newc complaint-->
br><!
--      <br>o
{%f ri    in range(11)%}
    {{ i }}
{%e ndfor %}
```

```
<br>
{%f{oriincomplaints%}
{i<b['USERNAME']}}
r>{%
f    orjin    i.values()%}
    {{j}}ndf
{%e<b    or%}
    r>
{%e ndfor%}-->
```

```
<divclass="fordashboardtop">
  <div class="fordashboardtopleft">
    Welcome{{name}},
  </div>
  <div class="fordashboardtopright">
    <a href="/login"><button class="forbutton">Sign
  </div>
</div>
```



```

    </v>
di<br
    > class="outerofdashdetails">

```

```

<div class="fordashboarddetails">
    <br>
    <!-- table of customers complaints -->
    <table class="table">
        <thead>
            <th>Complaint ID</th>
            <th class="padding">Username</th>
            <th>Title</th>
            <th>Complaint</th>
            <th>Solution</th>
            <th>Status</th>
        </thead>
        <tbody>
            <tr>
                <td class="padding">{ i['C_ID'] }</td>
                <td class="padding">{ i['USERNAME'] }</td>
                <td>{ i['TITLE'] }</td>
                <td>{ i['COMPLAINT'] }</td>
                <td>{ i['SOLUTION'] }</td>
                <td>{ i['STATUS'] }</td>
            </tr>
        </tbody>
    </table>

```

```

</table>

```

```

<br>
<center>

```

```

<div class="fordashboarddetails">

```



```
</
title>{%eock%}
```

```
{%blockbody%}
```

```
<!--things
```

```
div1
welcomej etson,signout
```

```
div2
yourcomp laintsstatus
```

```
add<newc omplaint-->
br><!
-- <br>o
{%f ri in range(11)%}
{{ i}}
{%e ndfor %}
```

```
<br>
{%f{oriincomplaints%}
{i<b['USERNAME']}}
r>{%
f orjin i.values()%}
{{j}}ndf
{%e<b or%}
r>
{%e ndfor%}-->
```

```
<divclass ="fordashboardtop">
<divc lass="fordashboardtopelements1">
WelcomeAdmin,
</div>
<divc lass="fordashboardtopelements2">
<a href="/login"><buttonclass="fo rbutton">Sign
out</button></a> </div>
```

```
</v>
i<br
> class="outerofdashdetails">
```

```
<divcla ss="fordashboarddetails">
<br>
<!-- tablelefcustomerscomplaints="$-->
<tabclas fortable">
<thead>
<thclass="pad">Name</
<thth>>Username</th>
```

```

<th>Email</th>
<th>>Phone</th>
<th>>Domain</th>
</thead>
<tbody>
<tr>
<td class="pad">{
{ i['NAME'] }}
</td>
<td class="pad">{
{ i['USERNAME'] }}
</td>
<td>
{{i['EMAIL'] }}
</td>
<td>
{{i['PHN']}}
</td>
<td>
{{i['DOMAIN']}}
</td>
<td>
{%iAfi['g6ATUSj']==1%}
si{%edlifioBSTA
not{%vail[' TUS']==0%}
eAvaie able
{%e %labl%
ndif
%}
</td>
</tr>
</tbody>

```

```
</table>
```

```

<br>
<center>

```

```
<div class="fordashboarddetails">
```

```
<button type="button" class="collapsible">Add new agent </
```

```
button>
```

```

<div class="content">
<br>
<form action="/addnewagent" method="post">
<div class="form">
<div class="textinput formleft">
Username

```

```

</div>
<div class="textinformright">
    <input type="text" name="username">
</div>
</div>
<div class="form">1
    <div class="textinformleft">
        Na
    </div>
    <div class="textinformright">
        <input type="text" name="name">
    </div>
</div>
<div class="form">1
    <div class="textinformleft">
        Email
    </div>
    <div class="textinformright">
        <input type="text" name="email">
    </div>
</div>
<div class="form">1
    <div class="textinformleft">
        Phe
    </div>
    <div class="textinformright">
        <input type="text" name="phone">
    </div>
</div>
<div class="form">1
    <div class="textinformleft">
        Doin
    </div>
    <div class="textinformright">
        <input type="text" name="domain">
    </div>
</div>
<div class="form">1
    <div class="textinformleft">
        Paword
    </div>
    <div class="textinformright">
        <input type="text" name="password">
    </div>
</div>

<br>
<br>
<div>
    <button class="formbutton" type="submit">Submit

```

</button>

```

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

```

```

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

```

```
</div>
```

```
{%endblock%}
```

## Dashboard.html

```
{%extends'base.html'%}
```

```
{%blockhead%}
```

```

<title>
Dashboard
</title>

```

```
{%endblock%}
```

```
{%blockbody%}
```

```
<!--things
```

```

    div1
welcomej etson,signout

```

```

    div2
yourcomp laintsstatus

```

```

add new complaint
-- <br><
><!-- br>
- fori inrange(11)}}
%}{{ i %}
{% endfor

```

```

<br>
{%f oriincomplaints

```

```

%}{{i['USERNAME']}}
<br>
{% for jini. values()
%}{{j}}
{%e<bndfor%}
r>
{%e ndfor%}-->

```

```

<divclass ="fordashboardtop">
  <divc lass="fordashboardtopelements1">
    Welcome{{name}},
  </div>
  <divc lass="fordashboardtopelements2">
    <a href="/login"><buttonclass="fo rbutton">Sign
    out</button></a> </div>

```

```

</v>
i<br
> class="outerofdashdetails">

```

```

<divcla ss="fordashboarddetails">
  <br><
  !-- tablelefcustomerscompls=äints-->
  <tabclas fortable">
    <thead>
      <thComplaintIcID</
      <tclass="pad">Ass>Cth>oainthDetail</th>
      <thignedAg>Statuenmplt>
      <th> s</ </t
      <th>>Solution<h>
    </thead> /t
    <tbody>>
      {% fori incomplaints %}
      <tr>
        <td>
          {{i['C_ID'] }}
        </td>
        <tdclass="pad">{
          { i['TITLE']}}
        </td>
        <td>
          {{i['ASSIGNED_AGENT']}}
        </td>
        <td>
          {%ifi['CoSTATUS']==1%}
          mpleted{%
          elifiNotc['STATUS']==0%}ed
          ompl{%elset
          e }
          %Inprogress
          {%endif %}

```

```

        </td>
        <td>
            {{i ['SOLUTION']}}
        </td>
    </tr>
    {%-endfor%}
</tbody>

```

```
</table>
```

```

<br>
<center>

```

```
<divclass="fordashboarddetails">
```

```
<buttontype="button"class="collapsible">Addnewcomplaint</
```

```
button>
```

```

<divclass="content">
    <br>
    <formaction="" /thod="post">
        <divclassaddnew"me="f"
            <divclass="formleft">
                <divclass="textinput">
                    <inputtype="text">
                </div>
            </div>
            <divclass="textinformright">
                <inputtype="name"name="title">
            </div>
        </div>

```

```

        <divclass="form">
            <divclass="textinformleft">
                <inputtype="text">
            </div>
            <divclass="textinformright">
                <textareaname="description"
                    style="border-radius:1rem;width:100%;height:150%;background-color:black;color:white;"></textarea>
            </div>

```

```

        <br>
        <br>
        <div>
            <buttonclass="formbutton"type="submit">Submit

```

```
</button>
```

```

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

```



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

```
</div>
```

```
{%endblock%}
%}Login.html
{%extends'base.html'%}
```

```
{%blockhead%}
```

```
<title>
    Login
</title>
```

```
{%endblock%}
```

```
{%blockbody%}
```

```
<divclass="forpadding">
```

```
    <!--for boxofthesignupfs="sign-->
    <divclass="gn">
        <div>
            <pclass="fortitle">
                SignIn
            </p>
            <hr>
            <formaction="/"method="post">
                <divclass="login"met="get">
                    <divclass="form">
                        <divclass="textinformleft">
                            Username
                        </div>
                        <divclass="textinformright">
                            <inputtype="text"name="username">
                        </div>
                    </div>
                </div>
                <divclass="form">
                    <divclass="textinformleft">
                        Password
                    </div>
                    <divclass="textinformright">
                        <inputtype="password"name="password">
                    </div>
                </div>
            </form>
        </div>
    </div>
```

```

        </div>
    </div>

    <br>
    <button class="forbutton" type="submit">SignIn</button>
</div>
</div>
<br>
<div>
    Newuser?<a href="/signup">Signup</a>
</div>
<br>
</div>

```

```

</div>
</div>

```

```
{%endblock%}
```

## Signup.html

```
{%extends 'base.html'%}
```

```
{%blockhead%}
```

```

<title>
    Sign Up
</title>
{%endblock%}

```

```
{%blockbody%}
```

```
<div class="forpadding">
```

```

<!--for boxofthesignupfs="signup"-->
<div class="forpadding">
    <div>
        <p class="fortitle">
            RegisterNow!!
        </p>
        <hr>
        <form action="/signup" method="post">
            <div class="forform">
                <div class="textinput formleft">
                    Username
                </div>

```

```

        <divclass="textinformright">
            <inputtype="ame"name="username">
        </div>
    </div>
    <divclass="forform">
        <divclass="textinformleft">
            Name
        </div>
        <divclass="textinformright">
            <inputtype="ame"name="name">
        </div>
    </div>
    <divclass="forform">
        <divclass="textinformleft">
            E-mail
        </div>
        <divclass="textinformright">
            <inputtype="ame"name="email">
        </div>
    </div>
    <divclass="forform">
        <divclass="textinformleft">
            PhoneNumber
        </div>
        <divclass="textinformright">
            <inputtype="ame"name="phn">
        </div>
    </div>
    <divclass="forform">
        <divclass="textinformleft">
            Password
        </div>
        <divclass="textinformright">
            <inputtype="assword"name="pass">
        </div>
    </div>
    <divclass="forform">
        <divclass="textinformleft">
            Re-enterPasword
        </div>
        <divclass="textinformright">
            <inputtype="password"name="repass">
        </div>
    </div>
    <divclass="forbutton">
        <buttontype="submit">Signup</button>
    </div>
</div>
<br>
</div>

```

```
{{msg}}>
</v
di<t
r><c>
iv Alreadyhaveanaccount?<ahref="/login">Signin</a>>
</div
<br>
```

&lt;/div&gt;

```
</div>  
←+→  
div{%e_block%}
```

## Tickets.html

```
{%extends 'base.html'%}
```

{%blockhead%}

```
<title>
AgentDashboard
</title>
```

```
{%endblock%}
```

```
{%blockbody%}
```

```
<!--things
```

```
div1  
welcomej etson,signout
```

```
div2  
yourcomp laintsstatus
```

```
add<newc complaint-->
br><!
-- <br>o
{%f ri in range(11)%}
{{ i}}
{%e ndfor %}
```

```
<br>
{%f oriincomplaints%}
```

```

{{i['USERNAME']}}
br>{
%f orjin i.values()%)
{{j}}ndf
{%e<b or%}
r>
{%e ndfor%}-->

```

```

<divclass="fordashboardtop">
  <div
    class="fordashboardtopelements1">e
    WlcomeAdmin,
  </>
  <divdclass="fordashboardtopelements2">
    <a href="/login"><buttonclass="fo rbutton">Sign
    out</button></a> </div>

```

```

</v>
i<br
> class="outerofdashdetails">

```

```

<divcla ss="fordashboarddetails">
  <br><
  !-- tablelefcustomerscompls=äints-- >
  <tabclas forttable">
    <thead>
      <th>ComplaintID</th>e
      <th ass="pad">Us rname</th>
      <th>Title</
      <thth>>Complai/
      <thnt<>Solutioth>th
      <thn</>>
    </thead>Status</
    <tbody>th>
    {% foriincomplaints%}
    <tr>
      <td>{{i['C_ID'] }}</td>
      <tdclass="pad">
        {{i['USERNA ME'] }}
      </td>
      <td>
        {{i['TITLE']}}
      </td>
      <td>
        {{i['COMPLAINT']}} }
      </td>
      <td>
        {{i['SOLUTION']}}
      </td>
      <td>
        {%ifi['STATUS']==1%}

```

```

Completed
{%enlse }
ot %Completed}
{%e lndif%
</td>
</tr>
{%en dfor%}
</tbody>

</table>

<br>
<center>

<divclass="fordashboarddetails">
<button type="button" class="collapsible">Assignanagent
</button>

<divclass="content">
<br>
<form action="/" method="post">
<divclass="assignagen">
<div class="form"> nformleft">
Colass="textir
</div>plaintID
<divclass="textinformright">
<i puttype="ame" name="ccid">
</div>
</div>
<div class="form">l
<div class="textinformleft">g
<label for="agent">Choose anage nt:</label>
</div>
<div class="textinformright">
<select name="agent" id="agenst">
{%foriinfreeagent %}
<option value="{{i['USERNAME']}}">{{
i['USERNAME']}}</option>
{%endfor%}
</select>
</div>
</div>

<br>
<br>
<div>
<button class="forbutton" type="submit">Submit
</button>

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

```

```

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

```

```
</div>
```

```
{%endblock%}
```

**App.py:**

```

from flask import Flask, render_template, request, redirect, session,
url_for
from flask import Flask, render_template, request, redirect,
session
import ibm_db
import os

```

```
app = Flask(__name__)
```

```

import ibm_db
import os

```

```
app = Flask(__name__)
```

```

# for connection
conn = ""

```

```

app.secret_key = 'a'
print("Trying to connect...")
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=824dfd4d-99de-440d-9991-63832d.bs2io90108kqb1od8lcg.databases.appdomain.cloud;PORT=30119;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=qvk70423;PWD=sadlGasU4iQy1yvk;", '', '')
print("connected..")

```

```

@app.route('/signup', methods=['GET', 'POST'])
def signup():
    global user_id
    user_id = ''
    if request.method == 'POST':
        username = request.form['username']
        name = request.form['name']
        email = request.form['email']
        phone = request.form['phone']
        password = request.form['password']
        print("inside checking")
        print(name)
        print(password)

```

```

if len(username) == 0 or len(name) == 0 or len(email) == 0 or len(phone) == 0 or len(password) == 0 or len(repass) == 0:
    msg = "Form is not complete. Please fill all fields!"
    return render_template('signup.html', msg=msg)
elif password != repass:
    msg = "Passwords do not match!"
    return render_template('signup.html', msg=msg)
elif not re.match(r'[a-zA-Z0-9_@+]', username):
    msg = 'Username can contain only letters, numbers, and underscores'
    return render_template('signup.html', msg=msg)
elif not re.match(r'^[a-zA-Z0-9_@+]\.?[a-zA-Z0-9_@+]', email):
    msg = 'Invalid email address'
    return render_template('signup.html', msg=msg)
elif not re.match(r'^[a-zA-Z0-9_+]', name):
    msg = "Invalid name"
    return render_template('signup.html', msg=msg)
elif not re.match(r'^[0-9]{9}$', phone):
    msg = "Invalid phone number"
    return render_template('signup.html', msg=msg)

```

```

sql = "select * from users where username = ?"
cursor.execute(sql)
result = cursor.fetchall()
if result:
    msg = "Account already exists"
else:
    user_id = 1
    insert_sql = "insert into users (username, password, email, phone, name) values (?, ?, ?, ?, ?)"
    cursor.execute(insert_sql, (username, password, email, phone, name))
    commit()
    msg = "Signed up successfully"
    return render_template('dashboard.html', msg=msg, name=name)
else:
    return render_template('signup.html')

```



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

```
@app.route('/login', methods=["GET", "POST"])
def login():
    global user_id
    g = ''
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        if user_id == 'admin' and password == 'admin':
            print("Admin")
            return render_template('admin.html')
        else:
            sql = "select * from agents where username = ? and password = ?"
            stmt = ibm_db.prepare(conn, sql)
            ibm_db.bind_param(stmt, 1, username)
            ibm_db.bind_param(stmt, 2, password)
            ibm_db.execute(stmt)
            account = ibm_db.fetch_assoc(stmt)
            if account:
                session['loggedin'] = True
                session['user_id'] = account['USERID']
                session['username'] = account['USERNAME']
                session['password'] = account['PASSWORD']
                msg = 'logged in successfully'
            else:
                msg = 'login failed'
    return render_template('login.html', msg=msg)
```

```
# for getting complaints details
def get_complaints():
    sql = "select * from complaints where assigned_agent = ?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt, 1, username)
    ibm_db.execute(stmt)
    dictionary = ibm_db.fetch_assoc(stmt)
    while dictionary:
        complaints.append(dictionary)
        dictionary = ibm_db.fetch_assoc(stmt)
    return render_template('agentdash.html',
                           name=account['USERNAME'], complaints=complaints)
```

```
sql = "select * from users where username = ? and password = ?"
stmt = ibm_db.prepare(conn, sql)
ibm_db.bind_param(stmt, 1, username)
ibm_db.bind_param(stmt, 2, password)
```

```

ibm_db.execute(stmt)
account=ibm_db.fetch_assoc(stmt)
int(account)
if account:
    session['loggedin']=True
    session['userid']=account['USERID']
    session['username']=account['USERNAME']
    msg='logged in successfully'

```

```

#forgetting sql complaints details for
stmt="select * from complaints where username =?"
complaints=[]
ibm_db.prepare(conn,stmt)
ibm_db.execute(stmt,1,username)
complaints=ibm_db.fetch_assoc(stmt)
if complaint:
    #print("The ID is: ",complaint['ID'])
    print("The complaint is: ",complaint['complaint'])
    complaints.append(complaint)
complaints=ibm_db.fetch_assoc(stmt)

```

```

print(complaints)
return render_template('dashboard.html',name=account['USERNAME'],
complaints=complaints)
else:
    msg='Incorrect user credentials'
    return render_template('dashboard.html',msg=msg)
else:
    return render_template('login.html')

```

```

@app.route('/addnew',methods=["GET", "POST"])
def add():
    if request.method=='POST':
        title=request.form['title']
        desc=request.form['desc']
        try:
            sql="insert into complaints (username,title,complaint)
            values(?,?,?)"
            stmt=ibm_db.prepare(conn,sql)
            ibm_db.bind_param(stmt,1,userid)
            ibm_db.bind_param(stmt,2,title)
            ibm_db.bind_param(stmt,3,desc)
            ibm_db.execute(stmt)
        except:
            print(userid)
            print(title)
            print(desc)
            print("cant insert")

```

```

sql="select*com fromcomplaints where username =?"
plaints=[]stmt=
ibm_db.pyibm_db.prepare(conn,sql)mt
bind_param(ibm_db.BT_INTEGER,1,userid)
execute(stmt,db)
ary=ibm_db.fetch_assoc(stmt)
while dictionary:
    #print("Th#peaampes:"eidicti"onary["EtMPNO"]1
    print("Thcomp=nibm:d(dictionary["
    lains. s) ct_db.fetch_as
    dictionary em fe soc(stmt)
print(complaint
return render_t plate('dashboard.html',name=userid,
complaints=complaints)

```

```

@app.route('/agents')
defagents():
    sql= "select*fromagents"
    agents=[]ibm
    stmt=.execdb.prepare(conn,sql)
    ibm_db.fetch_assoc(stmt)=i
    while ts.cty! False:
        gon =append(dictionary)
        di rendary=ibm_db.fetch_assoc(stmt)
    return er template('agents.html',age nts=agents)

```

```

@app.route('/addnewagent',methods=["GET", "POST"])
defaddagent():
    ifrequest.method=='POST':e=r
        username=request.form['username']
        ame=emailrequest['name']
        l=phoneform['emrequest.fail']
        domainpam['ph=request.foone']
        sswor rm['dd=request.foromain']
        try:
            m['password']

            sql ="insertintoat=igentevaluesn(?,?,?,?,?,2)"
            stmtbm_db.prepare_db.b(stmt,sql)1,
            ibm_bind_param(stmt_db.BT_INTEGER,mt,msern2,name)
            ibm_bind_param(stmt_db.BT_INTEGER,mt,ame)3,em
            ibm_bind_param(stmt_db.BT_INTEGER,mt,ail4,pho)
            ibm_bind_param(stmt_db.BT_INTEGER,ne5,pass)
            ibm_bind_param(stmt_db.BT_INTEGER,w6,domaiond))
            ibm_bind_param(stmt_db.BT_INTEGER,n
            ibm_execute(stmt)
        except:
            print("cantinsert")
            sql=" select*fromagents"

```

```

agents=[]stmt
mt=ibm_ibm_db.prepare(conn,sql)
db.execute(stmt)=i
dictionarybm_db.onafetch_assoc(stmt)
whiledicti y! False:
    agents.=append(dictionary)
    dictionary=ibm db.fetch_assoc(stmt)

returnrender_template('agents.html',agents=agents)

```

```

@app.route('/updatecomplaint', methods=["GET",
"POST"])defupdatecomplaint():
    ifrequest.method=='POST':
        d=request.form['cid']solu'
        tion=request.form['solution']
        try:
            sql = "updatecomplaintssetsol ution=? wherecid=?and
assigned agent=?"
            stmt =ibm_bdb.prepare(conn,sql)mt
            ibm_d.bindb_param(stt,1,solunt, iont
            ibm_d.bindb_param(st2,cid)mt,3,
            ibm_d.bind _param(stuser id)
            ibmdb.execute(stmt)
            sqlnupdaameteagents setstatus=3wherebm_d
            use_d?"b.bi stmt=i b.prepare(conn,mt,1,sql)
            ibm ndb.executeparam(stuserid)
            ibm c te(stmt)
        except:
            print("cantinsert")o
            sql=" select*frrlai mcomplaints where assigned_agent =
            ?"compnts=[]ibm_d
            stmt=m_d.prepbind_ae(conn,sql)mt
            ibb.m_dparam(execute ,1,userid)
            ib.ctiote(stmary=it)
            diiled bm_dictionaryab.fetch_assoc(stmt)
            wh ry! = False:
                complaints.append(dictionary)t
                dictionary=i(cbm_db.fe ch_assoc(stmt)
            #print omplaintsre)
            returnnder_temp late('agentdash.html',name=userid,
complaints=complaints)

```

```

@app.route('/tickets')
deftickets():
    sql="select*=[ fromcomplaints"
    mplaintssq_db.p
    mt=ibmibm_cute(eparase(conn,sql)
    db.exe =ib tmt)
    dictionary m db.fetch_assoc(stmt)

```

```

while dictionary != False:
    complaints.append(dictionary)
    dictionary = ibm_db.fetch_assoc(stmt)

sql = "select username from agents where status <> 1"
freeagents = []
stmt = ibm_db.prepare(conn, sql)
ibm_db.execute(stmt)
dictionary = ibm_db.fetch_assoc(stmt)
while dictionary != False:
    freeagents.append(dictionary)
    dictionary = ibm_db.fetch_assoc(stmt)
print(freeagents)
return render_template('tickets.html', complaints=complaints,
freeagents=freeagents)

```

```

@app.route('/assignagent', methods=['GET',
'POST'])
def assign_agent():
    if request.method == 'POST':
        cid = request.form['ccid']
        agent = request.form['agent']
        print(agent)
        print(cid)
        try:
            sql = "update complaints set assigned_agent = ? where id = ?"
            stmt = ibm_db.prepare(conn, sql)
            ibm_db.bind_param(stmt, 1, agent)
            ibm_db.bind_param(stmt, 2, cid)
            ibm_db.execute(stmt)
            sql = "update agents set status = 1 where username = ?"
            stmt = ibm_db.prepare(conn, sql)
            ibm_db.bind_param(stmt, 1, agent)
            ibm_db.execute(stmt)
        except:
            print("cant update")
        return redirect(url_for('tickets'))

```

```

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

```

```

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

```

```

if __name__ == "__main__":
    app.run(debug=True)

```

