Sprint 4:

Final delivery: Deploy of app

Admin:

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

<title>
  Admin Dashboard
</title>
{% endblock %}
{% block body %}
<!-- things

  div 1
welcome jetson,   sign out

  div 2
your complaints status

add new complaint -->
<br>
<!-- <br>
{% for i in range(11) %}
 {{ i }}
{% endfor %}

<br>
{% for i in complaints %}
{{ i['USERNAME'] }}
<br>
{% for j in i.values() %}
  {{ j }}
{% endfor %}
<br>
{% endfor %} -->

<div class="fordashboardtop">
  <div class="fordashboardtopelements1">
    Welcome Admin,
  </div>
  <div class="fordashboardtopelements2">
    <a href="/login"><button class="forbutton">Sign out</button></a>
  </div>

</div>
<br>
```

```
<div class="outerofdashdetails">

    <div class="fordashboarddetails">
        <br>
        <!-- table of customers complaints -->
        <table class="fortable">
            <thead>
            </thead>
            <tbody>
                <tr>
                    <td class="pad">
                        <a href="/agents">Agent Details</a>
                    </td>
                    <td class="pad">
                        <a href="/tickets">Customer Ticket Details</a>
                    </td>
                </tr>
            </tbody>

        </table>

        <br>

    </div>

</div>

{% endblock %}
```

Agent:

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




{% block head %}

<title>
   Dashboard
</title>





{% endblock %}
{% block body %}



<!-- things


    div 1
welcome jetson,   sign out


    div 2
your complaints status


add new complaint -->
<br>
<!-- <br>
{% for i in range(11) %}
  {{ i }}
{% endfor %}

<br>
{% for i in complaints %}
{{ i['USERNAME'] }}
<br>
{% for j in i.values() %}
    {{ j }}
{% endfor %}
<br>
{% endfor %} -->

<div class="fordashboardtop">
    <div class="fordashboardtopelements1">
    Welcome Admin,
  </div>
  <div class="fordashboardtopelements2">
    <a href="/login"><button class="forbutton">Sign out</button></a>
  </div>

</div>
<br>
<div class="outerofdashdetails">

  <div class="fordashboarddetails">
    <br>
    <!-- table of customers complaints -->
    <table class="fortable">
      <thead>
        <th class="pad">Name</th>
        <th>Username</th>
        <th>Email</th>
        <th>Phone</th>
        <th>Domain</th>
        <th>Status</th>
      </thead>
      <tbody>
        {% for i in agents %}
        <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>
            {% if i['STATUS'] == 1 %}
            Assigned to job
            {% elif i['STATUS'] == 0 %}
            not Available
            {% else %}
            Available
            {% endif %}
          </td>
        </tr>
        {% endfor %}
      </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="forform">
              <div class="textinformleft">
                  Username
              </div>
              <div class="textinformright">
                  <input type="name" name="username">
              </div>
            </div>
            <div class="forform">
              <div class="textinformleft">
                  Name
              </div>
              <div class="textinformright">
                  <input type="name" name="name">
              </div>
            </div>
            <div class="forform">
              <div class="textinformleft">
                  Email
              </div>
              <div class="textinformright">
                  <input type="name" name="email">
              </div>
            </div>
            <div class="forform">
              <div class="textinformleft">
                  Phone
              </div>
              <div class="textinformright">
                  <input type="name" name="phone">
              </div>
            </div>
            <div class="forform">
              <div class="textinformleft">
                  Domain
```

```
                     </div>
                     <div class="textinformright">
                        <input type="name" name="domain">
                     </div>
                  </div>
                  <div class="forform">
                     <div class="textinformleft">
                        Password
                     </div>
                     <div class="textinformright">
                        <input type="password" name="password">
                     </div>
                  </div>

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


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

</div>

{% endblock %}
```

Login:

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



{% block head %}

<title>
    Login
</title>

{% endblock %}



{% block body %}


<div class="forpadding">

    <!-- for box of the signup form -->
    <div class="sign">
        <div>
            <p class="fortitle">
                Sign In
            </p>
            <hr>
            <form action="/login" method="post">
                <div class="forform">
                    <div class="textinformleft">
                        Username
                    </div>
                    <div class="textinformright">
                        <input type="name" name="username">
```

```
          </div>
        </div>

        <div class="forform">
          <div class="textinformleft">
            Password
          </div>
          <div class="textinformright">
            <input type="password" name="pass">
          </div>
        </div>

        <br>
        <div>
          <button class="forbutton" type="submit"> Sign In >></button>
        </div>
      </form>
      <br>

      <div>
        New user? <a href="/signup">Sign up</a>
      </div>
      <br>
    </div>

  </div>
</div>

{% endblock %}
```

Signup:

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

```
{% block head %}

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



{% block body %}

<div class="forpadding">

    <!-- for box of the signup form -->
    <div class="sign">
        <div>
            <p class="fortitle">
                Register Now!!
            </p>
            <hr>
            <form action="/signup" method="post">
                <div class="forform">
                    <div class="textinformleft">
                        Username
                    </div>
                    <div class="textinformright">
                        <input type="name" name="username">
                    </div>
                </div>
                <div class="forform">
```

```
  <div class="textinformleft">
    Name
  </div>
  <div class="textinformright">
    <input type="name" name="name">
  </div>
</div>
<div class="forform">
  <div class="textinformleft">
    E - mail
  </div>
  <div class="textinformright">
    <input type="name" name="email">
  </div>
</div>
<div class="forform">
  <div class="textinformleft">
    Phone Number
  </div>
  <div class="textinformright">
    <input type="name" name="phn">
  </div>
</div>
<div class="forform">
  <div class="textinformleft">
    Password
  </div>
  <div class="textinformright">
    <input type="password" name="pass">
  </div>
</div>
<div class="forform">
  <div class="textinformleft">
```

```
                        Re - enter Password
                    </div>
                    <div class="textinformright">
                        <input type="password" name="repass">
                    </div>
                </div>
                <br>
                <div>
                    <button class="forbutton" type="submit"> Sign up >></button>
                </div>
            </form>
            <br>
            <div>
                {{msg}}
            </div>
            <br>
            <div>
                Already have an account? <a href="/login">Sign in</a>
            </div>
            <br>

        </div>

    </div>
</div>

{% endblock %}
```

Ticket:

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

```
{% block head %}

<title>
   Agent Dashboard
</title>




{% endblock %}




{% block body %}



<!-- things


   div 1
welcome jetson,   sign out


   div 2
your complaints status


add new complaint -->
<br>
<!-- <br>
{% for i in range(11) %}
  {{ i }}
{% endfor %}

<br>
{% for i in complaints %}
```

```
{{ i['USERNAME'] }}
<br>
{% for j in i.values() %}
  {{ j }}
{% endfor %}
<br>
{% endfor %} -->

<div class="fordashboardtop">
  <div class="fordashboardtopelements1">
    Welcome Admin,
  </div>
  <div class="fordashboardtopelements2">
    <a href="/login"><button class="forbutton">Sign out</button></a>
  </div>

</div>
<br>
<div class="outerofdashdetails">

  <div class="fordashboarddetails">
    <br>
    <!-- table of customers complaints -->
    <table class="fortable">
      <thead>
        <th>Complaint ID</th>
        <th class="pad">Username</th>
        <th>Title</th>
        <th>Complaint</th>
        <th>Solution</th>
        <th>Status</th>
      </thead>
      <tbody>
    {% for i in complaints %}
    <tr>
      <td>{{ i['C_ID'] }}</td>
      <td class="pad">
        {{ i['USERNAME'] }}
      </td>
      <td>
        {{ i['TITLE'] }}
      </td>
      <td>
        {{ i['COMPLAINT'] }}
      </td>
      <td>
        {{ i['SOLUTION'] }}
      </td>
      <td>
        {% if i['STATUS'] == 1 %}
        Completed
        {% else %}
        Not Completed
        {% endif %}
      </td>
    </tr>
    {% endfor %}
  </tbody>

</table>

<br>
<center>

  <div class="fordashboarddetails">
```

```
            <button type="button" class="collapsible">Assign an agent ⚡</button>
            <div class="content">
               <br>
               <form action="/assignagent" method="post">
                  <div class="forform">
                     <div class="textinformleft">
                        Complaint ID
                     </div>
                     <div class="textinformright">
                        <input type="name" name="ccid">
                     </div>
                  </div>
                  <div class="forform">
                     <div class="textinformleft">
                        <label for="agent">Choose an agent:</label>
                     </div>
                     <div class="textinformright">
                        <select name="agent" id="agent">
                           {% for i in freeagents %}
                           <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 %}
```

Base:

```
<!DOCTYPE html>

<head>

    <link rel="stylesheet" href="static/main.css"/>



    {% block head %}
    {% endblock %}

</head>

<body>
    {% block body %}

    {% endblock %}
    <script>
        var coll = document.getElementsByClassName("collapsible");
        var i;
    for (i = 0; i < coll.length; i++) {
      coll[i].addEventListener("click", function () {
        this.classList.toggle("active");
        var content = this.nextElementSibling;
        if (content.style.display === "block") {
          content.style.display = "none";
        } else {
          content.style.display = "block";
        }
      });
    }
  </script>
  <footer style="text-align: right ;">
    <a href="/about">Wanna know more about us? Click here</a>
  </footer>
</body>

</html>
```

Chatbot:

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

{% block head %}

<title>

   Dashboard

</title>

{% endblock %}

{% block body %}

<!—things
```

```
  div 1
 welcome jetson,   sign out
div 2
 your complaints status
add new complaint -->
 <br>
 <!-- <br>
 {% for i in range(11) %}
   {{ i }}
 {% endfor %}


 <br>
 {% for i in complaints %}
 {{ i['USERNAME'] }}
 <br>
 {% for j in i.values() %}
     {{ j }}
 {% endfor %}
 <br>
 {% endfor %} -->
 <script>
   window.watsonAssistantChatOptions = {
```

```
   integrationID: "9e791132-c22e-405e-a872-2b2757aff525", // The ID of this integration.
   region: "eu-gb", // The region your integration is hosted in.
   serviceInstanceID: "1b39367e-68a6-4d60-8040-c2f08814d426", // The ID of your service instance.
   onLoad: function(instance) { instance.render(); }
  };
  setTimeout(function(){
   const t=document.createElement('script');
   t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" +
 (window.watsonAssistantChatOptions.clientVersion || 'latest') + "/WatsonAssistantChatEntry.js";
   document.head.appendChild(t);
  });
 </script>
 <div class="fordashboardtop">
   <div class="fordashboardtopelements1">
     Welcome {{ name }},
   </div>
   <div class="fordashboardtopelements2">
     <a href="/login"><button class="forbutton">Sign out</button></a>
   </div>
</div>
 <br>
 <div class="outerofdashdetails">
   <div class="fordashboarddetails">
```

```
<br>
<!-- table of customers complaints -->
<table class="fortable">
    <thead>
        <th>Complaint ID</th>
        <th class="pad">Complaint Detail</th>
        <th>Assigned Agent</th>
        <th>Status</th>
        <th>Solution</th>
    </thead>
    <tbody>
        {% for i in complaints %}
        <tr>
            <td>
                {{ i['C_ID'] }}
            </td>
            <td class="pad">
                {{ i['TITLE'] }}
            </td>
            <td>
                {{ i['ASSIGNED_AGENT'] }}
            </td>
```

```
          <td>
            {% if i['STATUS'] == 1 %}
            Completed
            {% elif i['STATUS'] == 0 %}
            Not completed
            {% else %}
            In progress
            {% endif %}
          </td>
          <td>
            {{ i['SOLUTION'] }}
          </td>
        </tr>
        {% endfor %}
      </tbody>
    </table>
    <br>
    <center>
      <div class="fordashboarddetails">
        <button type="button" class="collapsible">Add new complaint ✚</button>
        <div class="content">
          <br>
          <form action="/addnew" method="post">
            <div class="forform">
              <div class="textinformleft">
                Title
              </div>
              <div class="textinformright">
                <input type="name" name="title">
              </div>
            </div>
            <div class="forform">
              <div class="textinformleft">
                Complaint
              </div>
              <div class="textinformright">
                <textarea name="des"
                  style="border-radius: 1rem;width: 90%;height: 150%;background-color: black;color:
white;"></textarea>
              </div>
            </div>
            <br>
            <br>
            <div>
              <button class="forbutton" type="submit"> Submit </button>
```

```
                        </div>

                    </form>

                    <br>

                </div>

          </div>

       </center>

    </div>

</div>

{% endblock %}
```

App:

```
from flask import Flask, render_template, request, redirect, url_for, session

import ibm_db

import re




app = Flask(__name__)




# for connection

# conn= ""
```

```
app.secret_key = 'a'

print("Trying to connect...")

conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=824dfd4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.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 userid

    msg = ''

    if request.method == 'POST':

        username = request.form['username']

        name = request.form['name']

        email = request.form['email']

        phn = request.form['phn']

        password = request.form['pass']

        repass = request.form['repass']

        print("inside checking")

        print(name)

        if len(username) == 0 or len(name) == 0 or len(email) == 0 or len(phn) == 0 or len(password) == 0 or len(repass) == 0:

            msg = "Form is not filled completely!!"
            print(msg)

            return render_template('signup.html', msg=msg)

        elif password != repass:

            msg = "Password is not matched"

            print(msg)

            return render_template('signup.html', msg=msg)

        elif not re.match(r'[a-z]+', username):

            msg = 'Username can contain only small letters and numbers'

            print(msg)

            return render_template('signup.html', msg=msg)

        elif not re.match(r'[^@]+@[^@]+\.[^@]+', email):

            msg = 'Invalid email'

            print(msg)

            return render_template('signup.html', msg=msg)

        elif not re.match(r'[A-Za-z]+', name):

            msg = "Enter valid name"

            print(msg)

            return render_template('signup.html', msg=msg)

        elif not re.match(r'[0-9]+', phn):

            msg = "Enter valid phone number"

            print(msg)

            return render_template('signup.html', msg=msg)
```

```
sql = "select * from users where username = ?"
stmt = ibm_db.prepare(conn, sql)
ibm_db.bind_param(stmt, 1, username)
ibm_db.execute(stmt)
account = ibm_db.fetch_assoc(stmt)
print(account)
if account:
    msg = 'Acccount already exists'
else:
    userid = username
    insert_sql = "insert into users values(?,?,?,?,?)"
    prep_stmt = ibm_db.prepare(conn, insert_sql)
    ibm_db.bind_param(prep_stmt, 1, username)
    ibm_db.bind_param(prep_stmt, 2, name)
    ibm_db.bind_param(prep_stmt, 3, email)
    ibm_db.bind_param(prep_stmt, 4, phn)
    ibm_db.bind_param(prep_stmt, 5, password)
    ibm_db.execute(prep_stmt)
    print("successs")
    msg = "succesfully signed up"
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('/')
def base():
    return redirect(url_for('login'))


@app.route('/login', methods=["GET", "POST"])
def login():
    global userid
    msg = ''
    if request.method == 'POST':
        username = request.form['username']
        userid = username
        password = request.form['pass']
        if userid == 'admin' and password == 'admin':
            print("its 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)
    print(account)
    if account:
        session['Loggedin'] = True
        session['id'] = account['USERNAME']
        userid = account['USERNAME']
        session['username'] = account['USERNAME']
        msg = 'logged in successfully'

        # for getting complaints details
        sql = "select * from complaints where assigned_agent = ?"
        complaints = []
        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 != False:
                complaints.append(dictionary)
                dictionary = ibm_db.fetch_assoc(stmt)
            print(complaints)
            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)
    print(account)
    if account:
        session['Loggedin'] = True
        session['id'] = account['USERNAME']
        userid = account['USERNAME']
        session['username'] = account['USERNAME']
        msg = 'logged in successfully'

        # for getting complaints details
```

```
            sql = "select * from complaints where username = ?"
            complaints = []
            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 != False:
                # print "The ID is : ",  dictionary["EMPNO"]
                # print "The Name is : ", dictionary[1]
                complaints.append(dictionary)
                dictionary = 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']
        des = request.form['des']
        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, des)
            ibm_db.execute(stmt)
        except:
            print(userid)
            print(title)
            print(des)
            print("cant insert")
        sql = "select * from complaints where username = ?"
        complaints = []
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, userid)
        ibm_db.execute(stmt)
        dictionary = ibm_db.fetch_assoc(stmt)
        while dictionary != False:
            # print "The ID is : ",  dictionary["EMPNO"]
            # print "The Name is : ", dictionary[1]
            complaints.append(dictionary)
            dictionary = ibm_db.fetch_assoc(stmt)
        print(complaints)
        return render_template('dashboard.html', name=userid, complaints=complaints)



@app.route('/agents')
def agents():
    sql = "select * from agents"
    agents = []
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.execute(stmt)
    dictionary = ibm_db.fetch_assoc(stmt)
    while dictionary != False:
        agents.append(dictionary)
        dictionary = ibm_db.fetch_assoc(stmt)
    return render_template('agents.html', agents=agents)
```

```
@app.route('/addnewagent', methods=["GET", "POST"])

def addagent():

    if request.method == 'POST':

        username = request.form['username']

        name = request.form['name']

        email = request.form['email']

        phone = request.form['phone']

        domain = request.form['domain']

        password = request.form['password']

        try:

            sql = "insert into agents values(?,?,?,?,?,?,2)"

            stmt = ibm_db.prepare(conn, sql)

            ibm_db.bind_param(stmt, 1, username)

            ibm_db.bind_param(stmt, 2, name)

            ibm_db.bind_param(stmt, 3, email)

            ibm_db.bind_param(stmt, 4, phone)

            ibm_db.bind_param(stmt, 5, password)

            ibm_db.bind_param(stmt, 6, domain)

            ibm_db.execute(stmt)

        except:

            print("cant insert")

        sql = "select * from agents"
        agents = []
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.execute(stmt)
        dictionary = ibm_db.fetch_assoc(stmt)
        while dictionary != False:
            agents.append(dictionary)
            dictionary = ibm_db.fetch_assoc(stmt)

        return render_template('agents.html', agents=agents)


@app.route('/updatecomplaint', methods=["GET", "POST"])
def updatecomplaint():
    if request.method == 'POST':
        cid = request.form['cid']
        solution = request.form['solution']
        try:
            sql = "update complaints set solution =?,status=1 where c_id = ? and assigned_agent=?"
            stmt = ibm_db.prepare(conn, sql)
            ibm_db.bind_param(stmt, 1, solution)
            ibm_db.bind_param(stmt, 2, cid)
            ibm_db.bind_param(stmt, 3, userid)
            ibm_db.execute(stmt)

            sql = "update agents set status =3 where username=?"

            stmt = ibm_db.prepare(conn, sql)

            ibm_db.bind_param(stmt, 1, userid)

            ibm_db.execute(stmt)

        except:

            print("cant insert")

        sql = "select * from complaints where assigned_agent = ?"

        complaints = []

        stmt = ibm_db.prepare(conn, sql)

        ibm_db.bind_param(stmt, 1, userid)

        ibm_db.execute(stmt)

        dictionary = ibm_db.fetch_assoc(stmt)

        while dictionary != False:

            complaints.append(dictionary)

            dictionary = ibm_db.fetch_assoc(stmt)

        # print(complaints)

        return render_template('agentdash.html', name=userid, complaints=complaints)



@app.route('/tickets')

def tickets():
    sql = "select * from complaints"
    complaints = []
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.execute(stmt)
    dictionary = ibm_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 assignagent():
    if request.method == "POST":
        ccid = request.form['ccid']
        agent = request.form['agent']
        print(ccid)
        print(agent)
        try:
            sql = "update complaints set assigned_agent =? where c_id = ?"
            stmt = ibm_db.prepare(conn, sql)
            ibm_db.bind_param(stmt, 1, agent)
            ibm_db.bind_param(stmt, 2, ccid)
            ibm_db.execute(stmt)
            sql = "update agents set status =1 where username = ?"
            stmt = ibm_db.prepare(conn, sql)
            ibm_db.bind_param(stmt, 1, userid)
            ibm_db.execute(stmt)
        except:
            print("cant update")
        return redirect(url_for('tickets'))


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