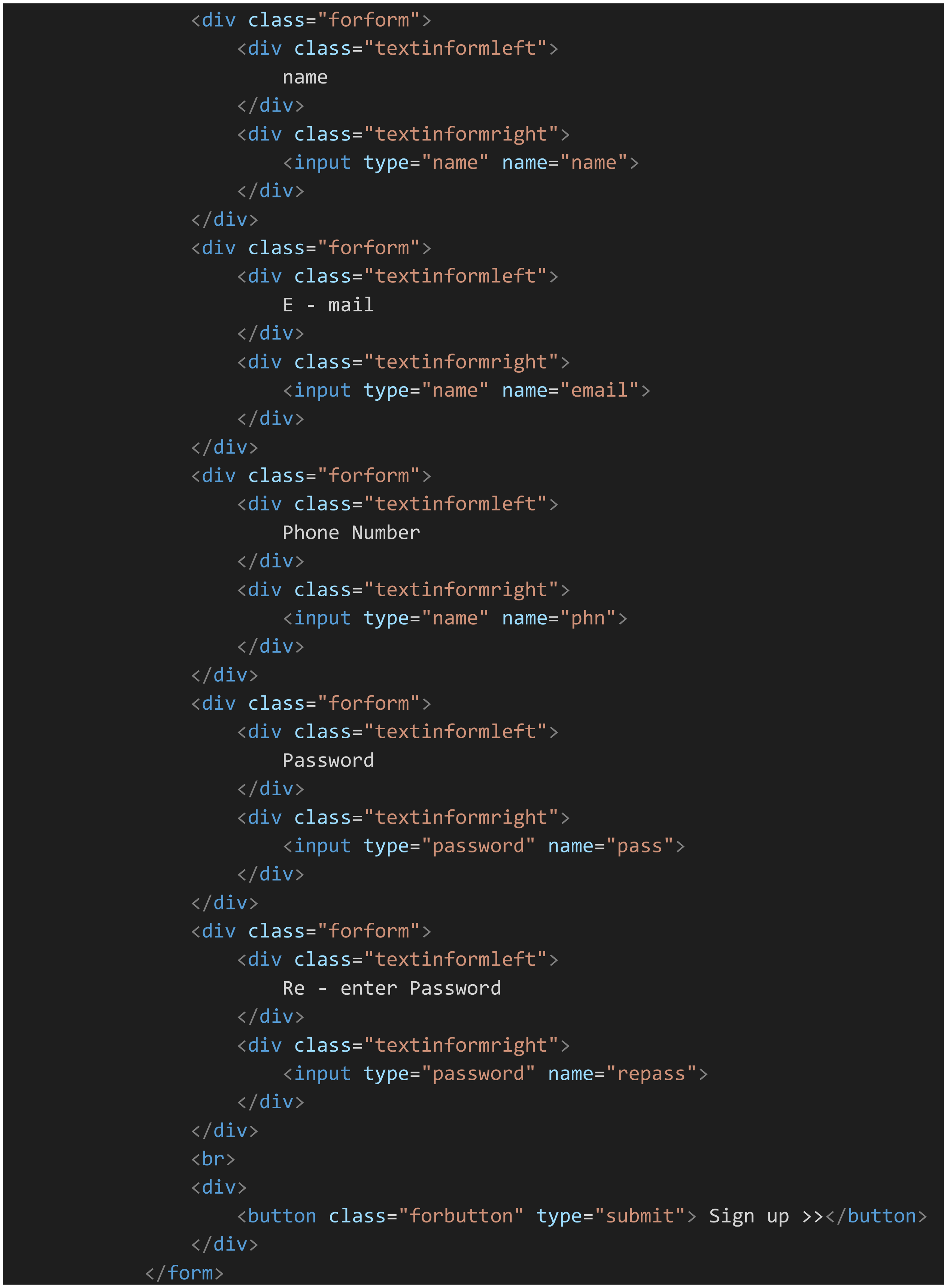
Date	18 November 2022
Team ID	PNT2022TMID01257
Project Name	Customer Care Registry

INTERACTIVE WEB APPLICATION UI

SIGNUP PAGE

```
{% 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>
```



```html
                <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 %}
```

LOGIN PAGE:

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



{% block head %}

<title>
    Login
</title>

{% endblock %}



{% block body %}
<div class="text">
<h1>CUSTOMER CARE REGISTRY</h1>
</div>
<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 %}
```

ADMIN LOGIN

```
{% 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 LOGIN

```
{% 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 %}
```

TICKETS.HTML

```
{% 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">
```

```html
        <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 %}
```

APP.PY

```

from flask import Flask, render_template, request, redirect, session, url_for
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=b1bc1829-6f45-4cd4-bef4-
10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32304;SECURITY=S
SL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=yjs69202;PWD=nCHXXVA1056i46k
y;", '', '')
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")
```

```python
            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)
```

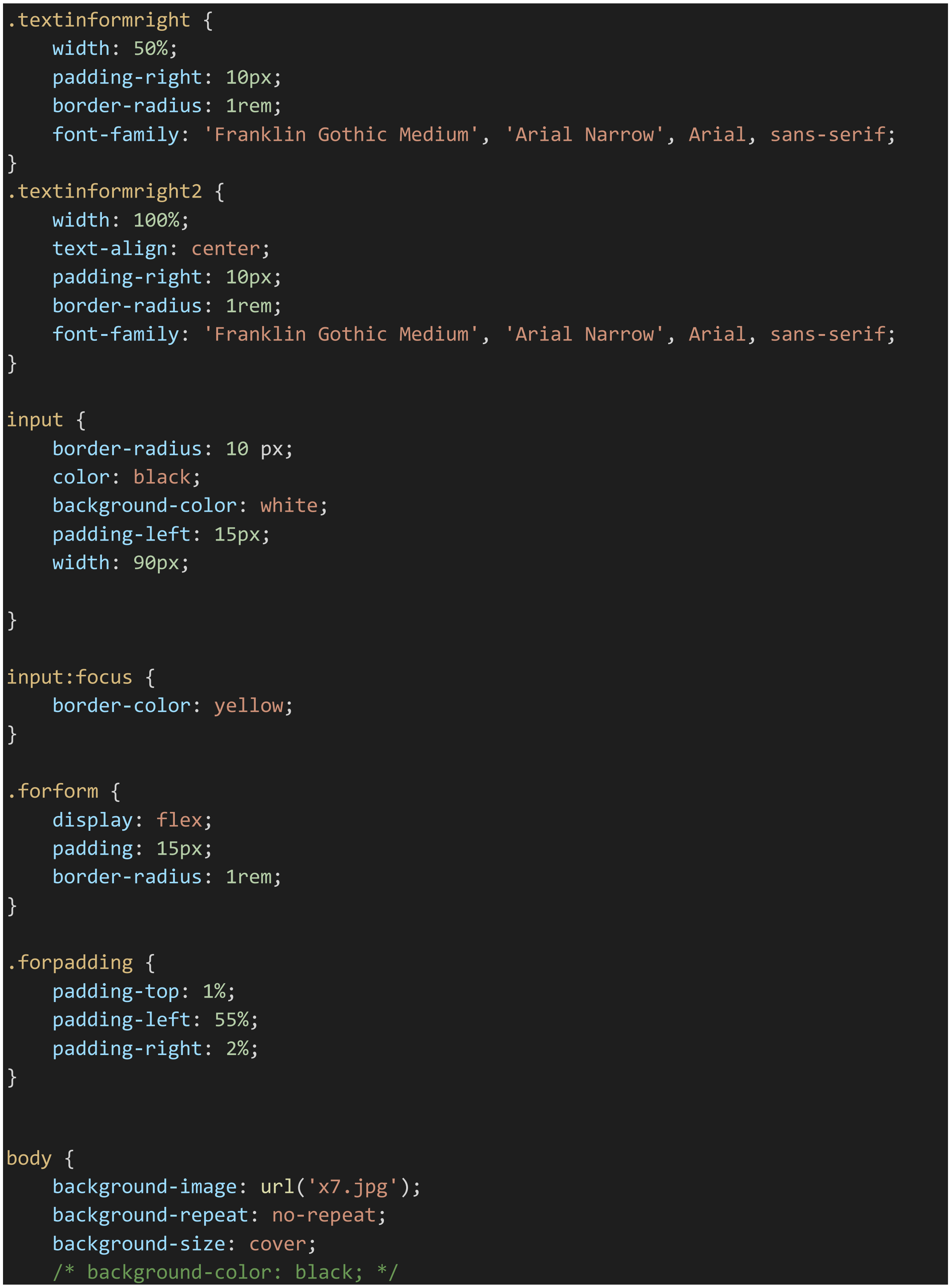
```python
@app.route('/assignagent', methods=['GET', 'POST'])
def assignagent():
    if request.method == "POST":
        ccid = request.form['ccid']
        agent = request.form['agent']
        print(ccid)
        print(agent)
        try:
            sql = "update complaints set assigned_agent =? 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)
```

DEPLOYMENT.YAML

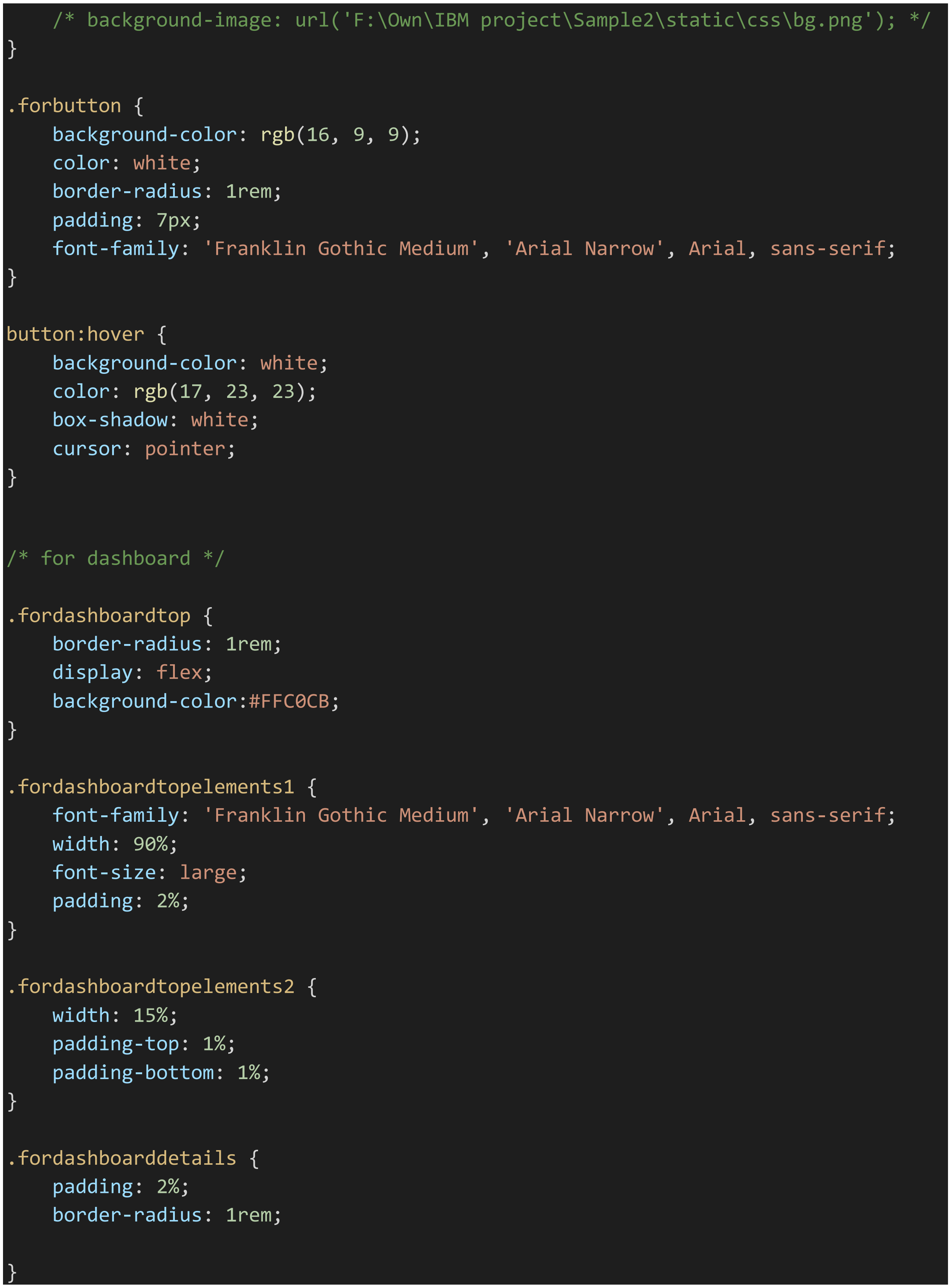
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: flask-node-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: flasknode
  template:
    metadata:
      labels:
        app: flasknode
    spec:
      containers:
      - name: flasknode
        image: au.icr.io/customer-care-ibm/customer-care-ibm
        ports:
        - containerPort: 5000
```

MAIN.CSS

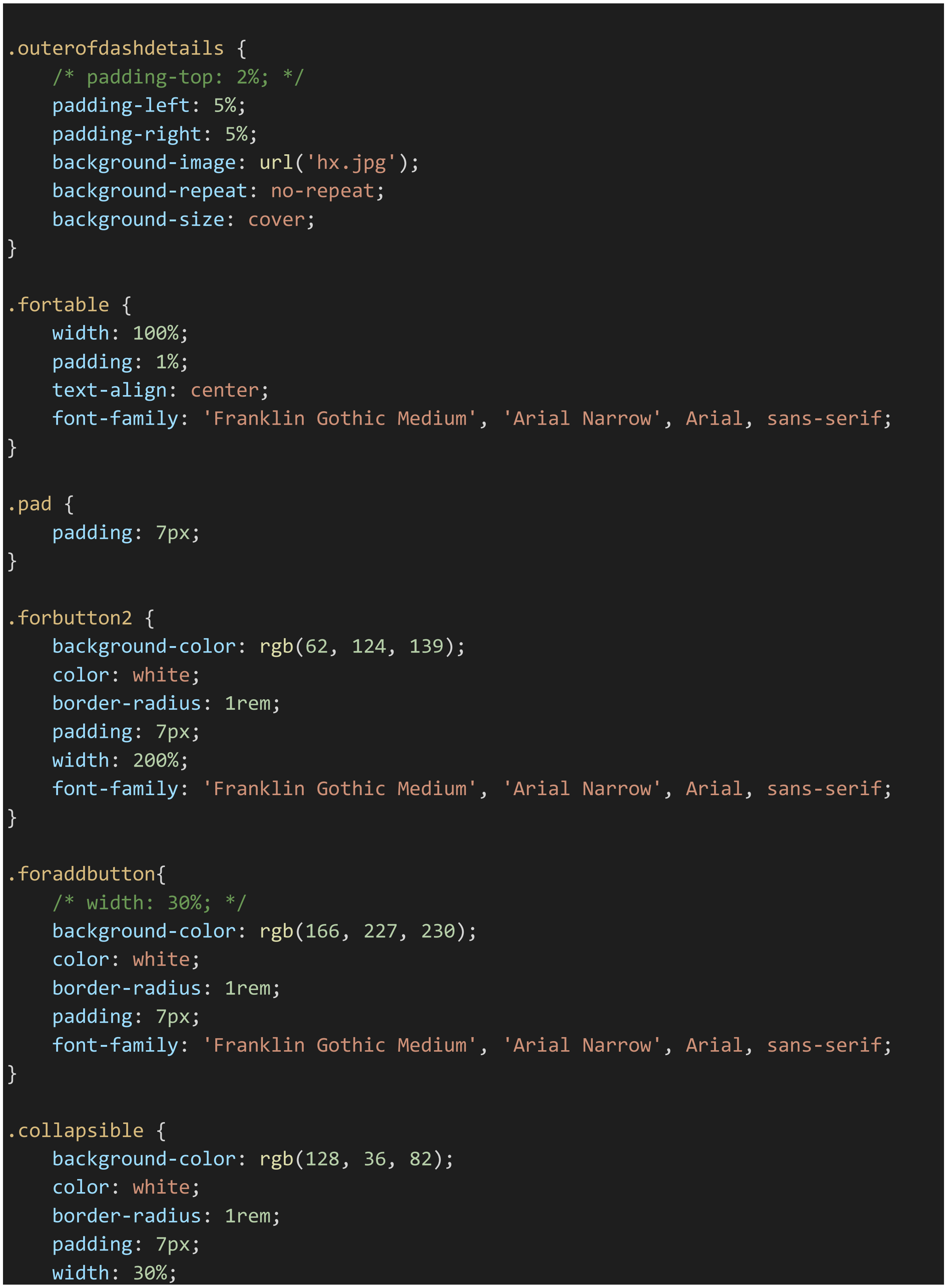
```
.sign {
    border-radius: 1rem;
    background-color: rgb(191, 191, 191);
    background-image: url('gr1.jpg');
    text-align: center;
    padding: 1%;
    padding-bottom: 3%;
    padding-top: 3%;
    margin-top: 5%;
    margin-left: 55%;
    border: 3px solid black;
}
.text{
    padding-left: 1025px;
    text-align: center;
}
.fortitle {
    font-size: medium;
    font-weight: 500;
    font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
    padding: 3px;
}

.forp {
    font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
}

.textinformleft {
    text-align: left;
    padding-left: -1%;
    padding-right: 3%;

    width: 50%;
    border-radius: 1rem;
    font-size: medium;
    font-weight: 500;
    font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
}
```



```css
.textinformright {
    width: 50%;
    padding-right: 10px;
    border-radius: 1rem;
    font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
}
.textinformright2 {
    width: 100%;
    text-align: center;
    padding-right: 10px;
    border-radius: 1rem;
    font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
}

input {
    border-radius: 10 px;
    color: black;
    background-color: white;
    padding-left: 15px;
    width: 90px;

}

input:focus {
    border-color: yellow;
}

.forform {
    display: flex;
    padding: 15px;
    border-radius: 1rem;
}

.forpadding {
    padding-top: 1%;
    padding-left: 55%;
    padding-right: 2%;
}


body {
    background-image: url('x7.jpg');
    background-repeat: no-repeat;
    background-size: cover;
    /* background-color: black; */
```



```css
    /* background-image: url('F:\Own\IBM project\Sample2\static\css\bg.png'); */
}

.forbutton {
    background-color: rgb(16, 9, 9);
    color: white;
    border-radius: 1rem;
    padding: 7px;
    font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
}

button:hover {
    background-color: white;
    color: rgb(17, 23, 23);
    box-shadow: white;
    cursor: pointer;
}


/* for dashboard */

.fordashboardtop {
    border-radius: 1rem;
    display: flex;
    background-color:#FFC0CB;
}

.fordashboardtopelements1 {
    font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
    width: 90%;
    font-size: large;
    padding: 2%;
}

.fordashboardtopelements2 {
    width: 15%;
    padding-top: 1%;
    padding-bottom: 1%;
}

.fordashboarddetails {
    padding: 2%;
    border-radius: 1rem;

}
```



```css
.outerofdashdetails {
    /* padding-top: 2%; */
    padding-left: 5%;
    padding-right: 5%;
    background-image: url('hx.jpg');
    background-repeat: no-repeat;
    background-size: cover;
}

.fortable {
    width: 100%;
    padding: 1%;
    text-align: center;
    font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
}

.pad {
    padding: 7px;
}

.forbutton2 {
    background-color: rgb(62, 124, 139);
    color: white;
    border-radius: 1rem;
    padding: 7px;
    width: 200%;
    font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
}

.foraddbutton{
    /* width: 30%; */
    background-color: rgb(166, 227, 230);
    color: white;
    border-radius: 1rem;
    padding: 7px;
    font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
}

.collapsible {
    background-color: rgb(128, 36, 82);
    color: white;
    border-radius: 1rem;
    padding: 7px;
    width: 30%;
    font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
    /* background-color: #777; */
    /* color: white; */
    cursor: pointer;
    /* padding: 18px; */
    /* width: 100%; */
    /* border: none;
    text-align: left; */
    /* outline: none;
    font-size: 15px; */
}

.collapsible:hover {
    background-color: white;
}

.content {
    /* padding: 0 18px; */
    display: none;
    border-radius: 1rem;
    background-color: #FFC0CB;
    width: 50%;
    /* overflow: hidden; */
    /* background-color: #f1f1f1; */
}
```