

## **PROJECT DEVELOPMENT PHASE**

### **CLOUD & APPLICATION REFINING**

#### **SPRINT - 4**

<b>Date:</b>	<b>15 NOV 2022</b>
<b>Team ID:</b>	<b>PNT2022TMID24347</b>
<b>Project name:</b>	<b>Customer Care Registry</b>

## Agents & Tickets:

The screenshot displays the Visual Studio Code interface with a project named "Project Development Phase". The Explorer sidebar on the left shows the file structure, including a "templates" folder containing "customercare.html". The main editor area shows the content of "customercare.html", which is an HTML form with the following structure:

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <link rel="stylesheet" href="/css/styles.css">
8   <title>Customer Care</title>
9 </head>
10 <body>
11   <form action="/submit" method="post" class="form">
12     <h1>Customer Care</h1>
13     <div class="form-data">
14       <label for="name">Name:</label>
15       <input name="name" placeholder="Name" type="text" required>
16     </div>
17     <div class="form-data">
18       <label for="email">Email:</label>
19       <input name="email" placeholder="Customer Email" type="email" required>
20     </div>
21     <div class="form-data">
22       <label for="complaint">Complaint:</label>
23       <textarea name="complaint" cols="30" rows="10" placeholder="Customer Complaint" required></textarea>
24     </div>
25     <button type="submit" class="btns">Submit</button>
26     <a href="/login"><button type="button" class="btns">Logout</button></a>
27   </form>
28 </body>
29 </html>

```

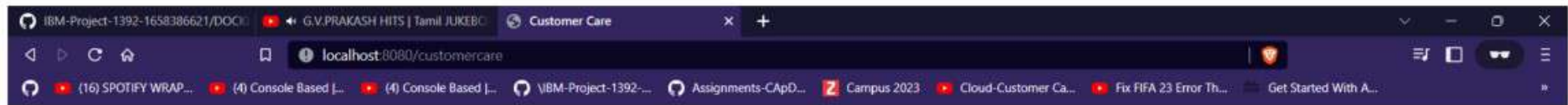
The terminal at the bottom shows the command to run the application:

```

PS C:\Users\sanja\OneDrive\Desktop\Kevin IBM\Kevin IBM\Project Development Phase> & "c:/Program Files (x86)/Python37-32/python.exe" "c:/Users/sanja/OneDrive/Desktop/
Kevin IBM/Kevin IBM/Project Development Phase/Sprint 3/main.py"
PS C:\Users\sanja\OneDrive\Desktop\Kevin IBM\Kevin IBM\Project Development Phase>

```

## Customer Dashboard:



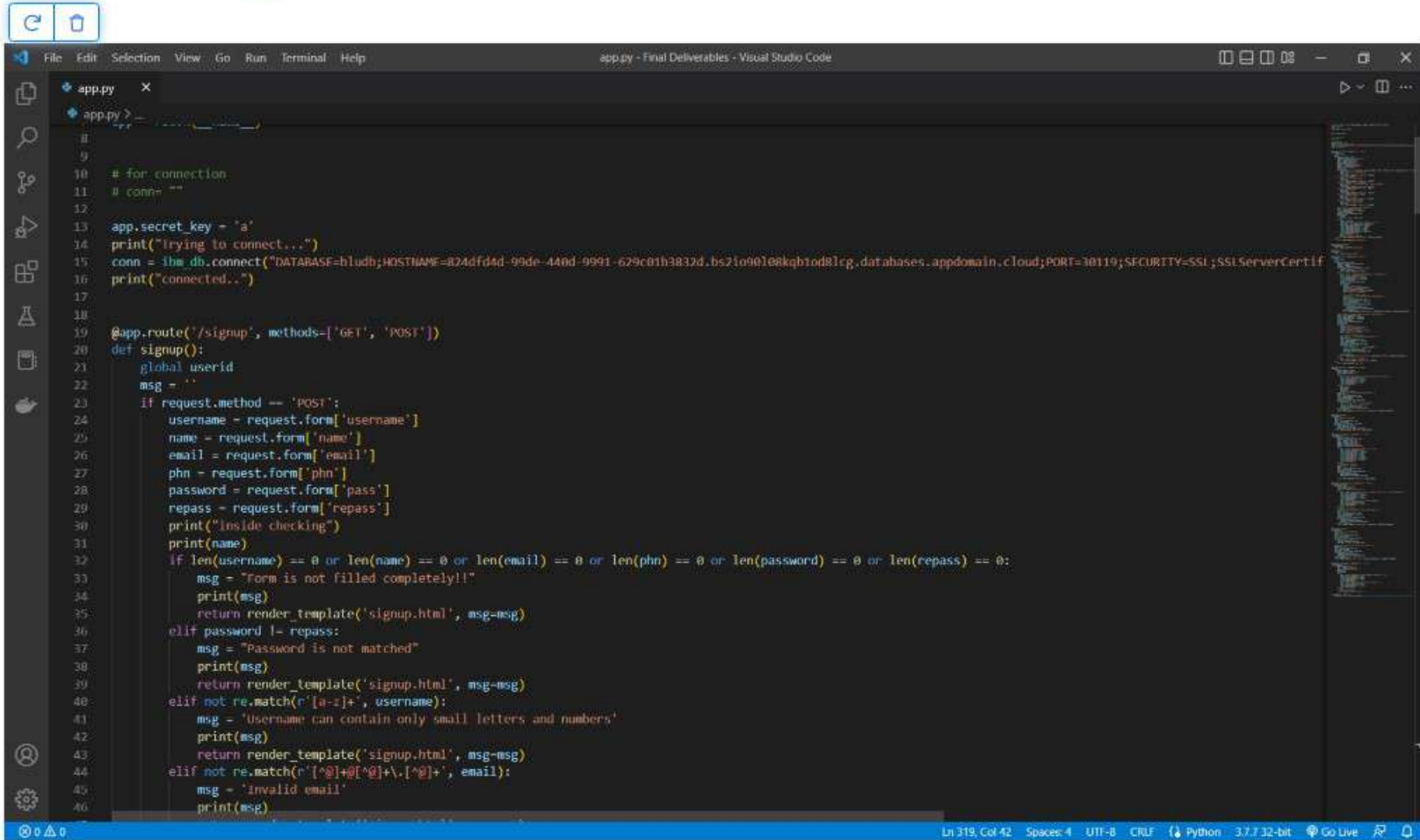
### Customer Care

Name:

Email:

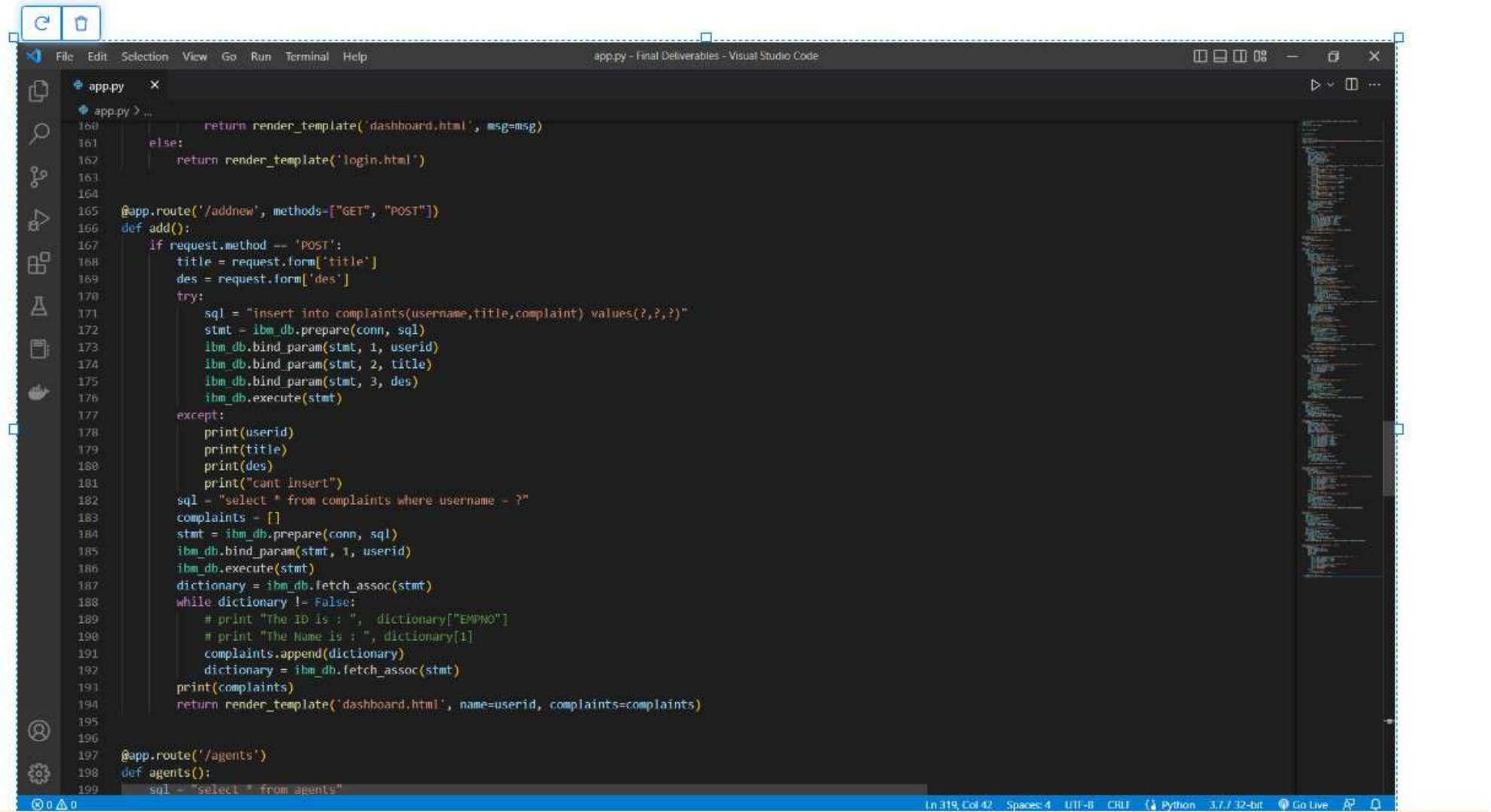
Complaint:

## Ibm\_db and integration with CLI



```
10 # for connection
11 # conn= ""
12
13 app.secret_key = 'a'
14 print("trying to connect...")
15 conn = ibm_db.connect("DATABASE=hludb;HOSTNAME=824dfd4d-99de-44nd-9991-629c01h3832d.bs2io90168kqbiod8lcp.databases.appdomain.cloud;PORT=30110;SECURITY=SSL;SSLServerCertif
16 print("connected..")
17
18
19 @app.route('/signup', methods=['GET', 'POST'])
20 def signup():
21     global userid
22     msg = ''
23     if request.method == 'POST':
24         username = request.form['username']
25         name = request.form['name']
26         email = request.form['email']
27         phn = request.form['phn']
28         password = request.form['pass']
29         repass = request.form['repass']
30         print("inside checking")
31         print(name)
32         if len(username) == 0 or len(name) == 0 or len(email) == 0 or len(phn) == 0 or len(password) == 0 or len(repass) == 0:
33             msg = "form is not filled completely!!"
34             print(msg)
35             return render_template('signup.html', msg=msg)
36         elif password != repass:
37             msg = "Password is not matched"
38             print(msg)
39             return render_template('signup.html', msg=msg)
40         elif not re.match(r'[a-z]+', username):
41             msg = 'Username can contain only small letters and numbers'
42             print(msg)
43             return render_template('signup.html', msg=msg)
44         elif not re.match(r'^[a-zA-Z0-9]+@[a-zA-Z0-9]+\.[a-zA-Z]+$', email):
45             msg = 'invalid email'
46             print(msg)
```

## Db Validation & Instance Storage in DB2



```
app.py
app.py > ...
160         return render_template('dashboard.html', msg=msg)
161     else:
162         return render_template('login.html')
163
164
165 @app.route('/addnew', methods=["GET", "POST"])
166 def add():
167     if request.method == 'POST':
168         title = request.form['title']
169         des = request.form['des']
170         try:
171             sql = "insert into complaints(username,title,complaint) values(?,?,?)"
172             stmt = ibm_db.prepare(conn, sql)
173             ibm_db.bind_param(stmt, 1, userid)
174             ibm_db.bind_param(stmt, 2, title)
175             ibm_db.bind_param(stmt, 3, des)
176             ibm_db.execute(stmt)
177         except:
178             print(userid)
179             print(title)
180             print(des)
181             print("cant insert")
182         sql = "select * from complaints where username = ?"
183         complaints = []
184         stmt = ibm_db.prepare(conn, sql)
185         ibm_db.bind_param(stmt, 1, userid)
186         ibm_db.execute(stmt)
187         dictionary = ibm_db.fetch_assoc(stmt)
188         while dictionary != False:
189             # print "The ID is : ", dictionary["EMPNO"]
190             # print "The Name is : ", dictionary[1]
191             complaints.append(dictionary)
192             dictionary = ibm_db.fetch_assoc(stmt)
193         print(complaints)
194         return render_template('dashboard.html', name=userid, complaints=complaints)
195
196
197 @app.route('/agents')
198 def agents():
199     sql = "select * from agents"
```

Ln 319, Col 42 Spaces: 4 UTF-8 CRLF Python 3.7.7 32-bit Go Live