

# **PROJECT REPORT**

## **Customer Care Registry**

Name : Prabhanjan Kumar

Project : Customer Care Registry

Domain : Python Flask

## **CONTENTS:**

### **1.INTRODUCTION**

#### **1.1 Overview**

### **2.LITERATURE SURVEY**

#### **2.1 Existing problem**

#### **2.2 Proposed solution**

### **3.THEORETICAL ANALYSIS**

#### **3.1 Block diagram**

#### **3.2 Hardware/Software designing**

### **4.EXPERIMENTAL INVESTIGATIONS**

### **5.RESULT**

### **6.ADVANTAGES & DISADVANTAGES**

### **7.APPLICATIONS**

### **8. FUTURE SCOPE**

### **9.CONCLUSION**

### **10.BIBLIOGRAPHY**

## **1.INTRODUCTION**

### **1.1 OVERVIEW**

Online customer care and service center is a web-based application Developed using java programming language. With a platform of a typical “service center”, this system provides online technical services to its customers on a 24×7 basis. The whole process involves writing large volume of data in registers and preparing several reports daily.the basic services include hardware and software of a computer. It also maintains database of their employ details of their customers, and many more. Online customer care and service center application is developed to automate all the office activities of a typical service center.The main objective of this Online Customer Care and Service Center software is to develop an information system to store, maintain, update and process data relating to the shop. It will prepare various reports to aid in smooth and speedy functioning of ‘Service Center’ activities.

## **2.LITERATURE SURVEY:**

### **2.1EXISTING PROBLEM:**

the present computer service centers generally keep the details of the customers and products in word documents, spreadsheets or paper register, and the management of all records is illegal to some extent.There

are problems relating redundancy of data like customer name and address, details of their account and also allocation of duties to the employees.

When a customer takes some kind of services, the charge is calculated manually, and this process is time consuming. Also, regular and overtime duties are not maintained properly. This leads to improper calculation of employees becomes quite complicated for every employee. Another problem usually faced by the organization which has been solved in the proposed Online Customer Care and Service Center Project is the frequent complaints by the customers for not getting timely services.

## **2.2 PROPOSED SOLUTION:**

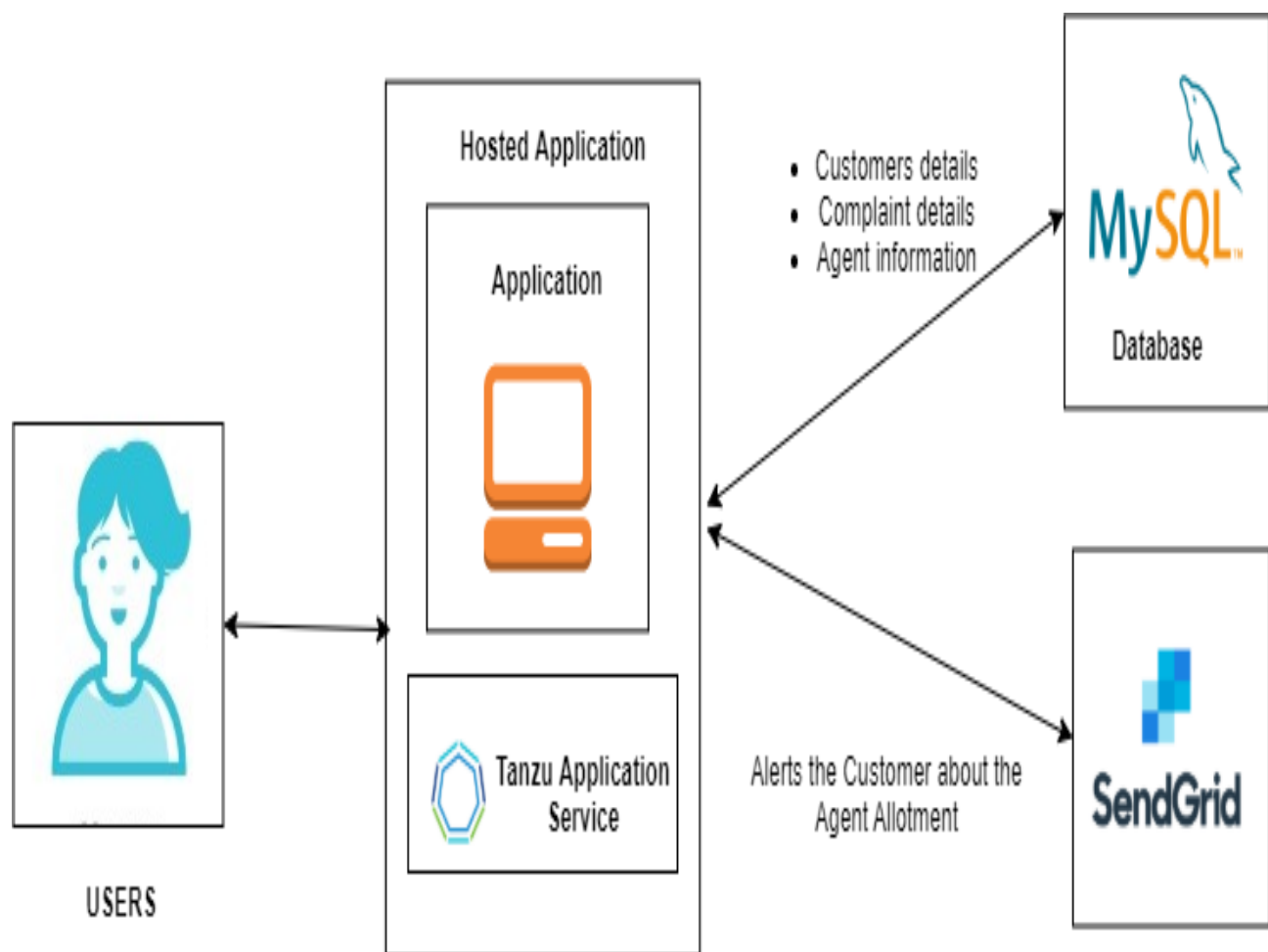
This Application has been developed to help the customer in processing their complaints. The customers can raise the ticket with a detailed description of the issue. An Agent will be assigned to the Customer to solve the problem. Whenever the agent is assigned to the customer they will be notified with an email alert. Customers can view the status of the ticket till the service is provided.

Admin: The main roles and responsibilities of the admin is to take care of the whole process. Starting from Admin login followed by the agent creation and assigning the customers complaints. Finally, He will be able to track the work assigned to the agent and notification will be sent to the customer.

User: They can register for an account. After the login, they can create a complaint with a description of the problem they are facing. Each user will be assigned an agent. They can view the status of their complaint.

### 3.THEORITICAL ANALYSIS:

#### 3.1. BLOCK DIAGRAM:



#### 3.2. SOFTWARE DESIGNING:

1. Python3
2. Flask Framework

3. Send Grid
4. Vmware Tanzu
5. Docker
6. Remote SQL

#### **4.EXPERIMENTAL INVESTIGATIONS:**

##### **Vmware Tanzu:**

VMware Tanzu is the suite or portfolio of products and solutions that allow its customers to Build, Run, and Manage Kubernetes controlled container-based applications.

##### **Send Grid:**

SendGrid is a cloud-based email service that provides reliable transactional email delivery, scalability, and real-time analytics along with flexible APIs that make custom integration easy.

##### **Docker:**

Docker is a tool designed to make it easier to create, deploy, and run applications by using containers. Containers allow a developer to package up an application with all of the parts it needs, such as libraries and other dependencies, and deploy it as one package.

##### **RemoteSql:**

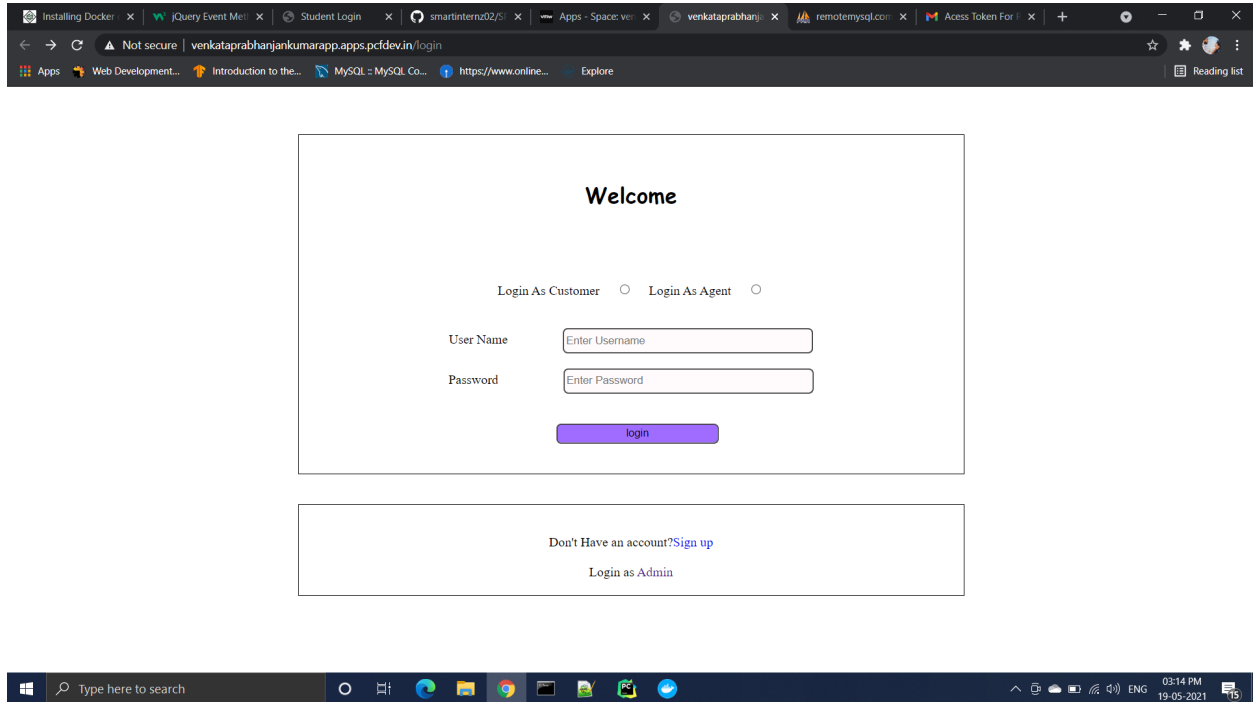
Remote access is the ability to get access to a SQL Server from a remote distance in order to manipulate data which are located on that SQL Server. The SQL Server instance to allow the protocol being requested.

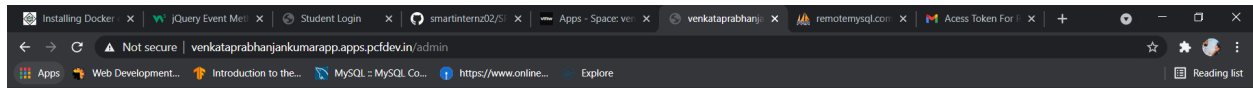
Allow access through the Window Firewall.

## 5.RESULT:

applink:

<https://venkataprabhanjankumarapp.apps.pcfdev.in/>

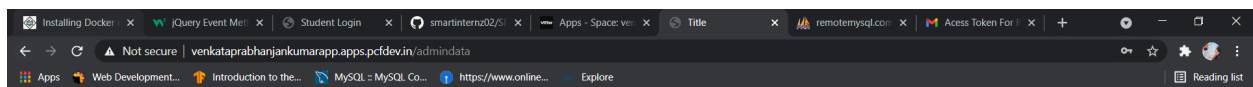




## Administrator Login

User Name

Password



[Agent Registration](#)[Agent Customer Details](#)[Admin Details](#)[Logout](#)

## Welcome Admin

**Username: 9JtF7WHjWK**

**Tables Present In DataBase**

Users

agentdetails

agentnumbers

agentquestions

questions

userquestions



[Agent Registration](#) [Agent Customer Details](#) [Admin Details](#) [Logout](#)

## Agent Registration

Enter Email Id

Generate Token

[Agent Registration](#) [Agent Customer Details](#) [Admin Details](#) [Logout](#)

## Agent Registration

Enter Email Id

Generate Token

## Agent Registration

Created Successfully

Enter Email Id

**Gmail** Search mail

Compose

**Inbox** 17

- Starred
- Snoozed
- Sent
- Drafts 6
- More

**Meet**

- New meeting
- Join a meeting

**Hangouts**

- Sampath +

No recent chats  
Start a new one

1 of 76

appmanaging84@gmail.com  
Access Token For Registration LCOT36 9:12 AM (6 hours ago) ☆

appmanaging84@gmail.com  
Access Token For Registration 99KNYV 9:46 AM (5 hours ago) ☆

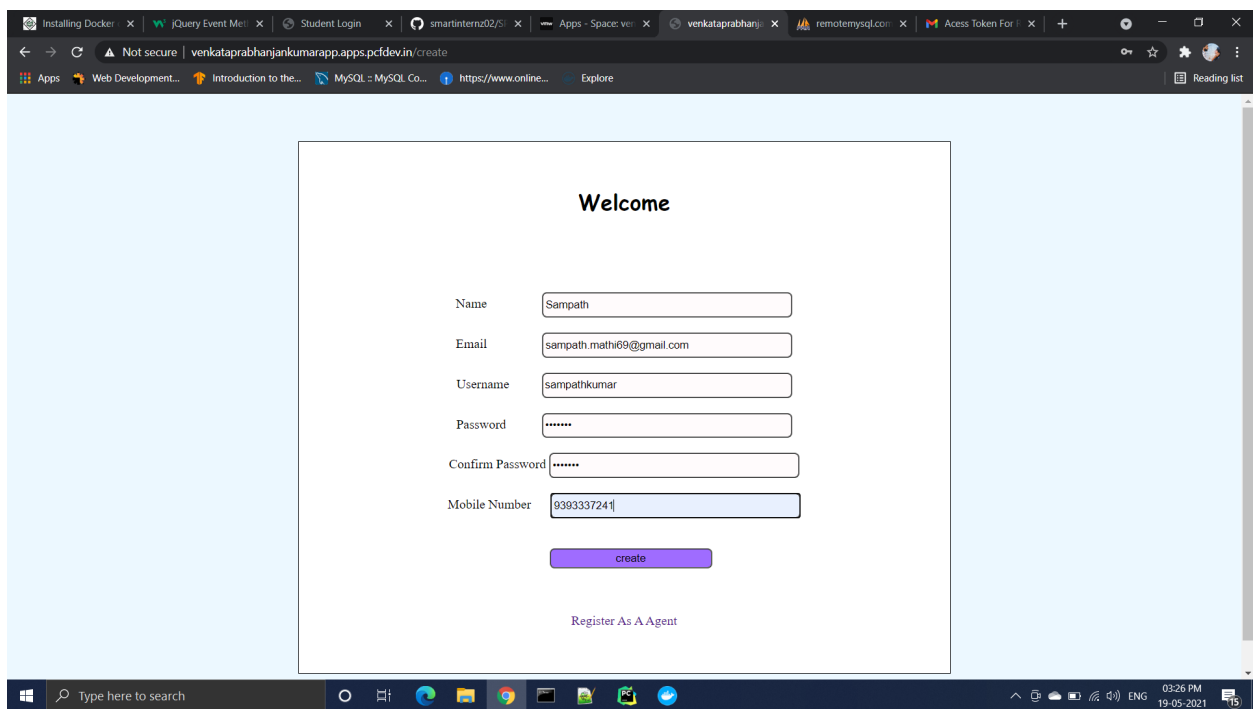
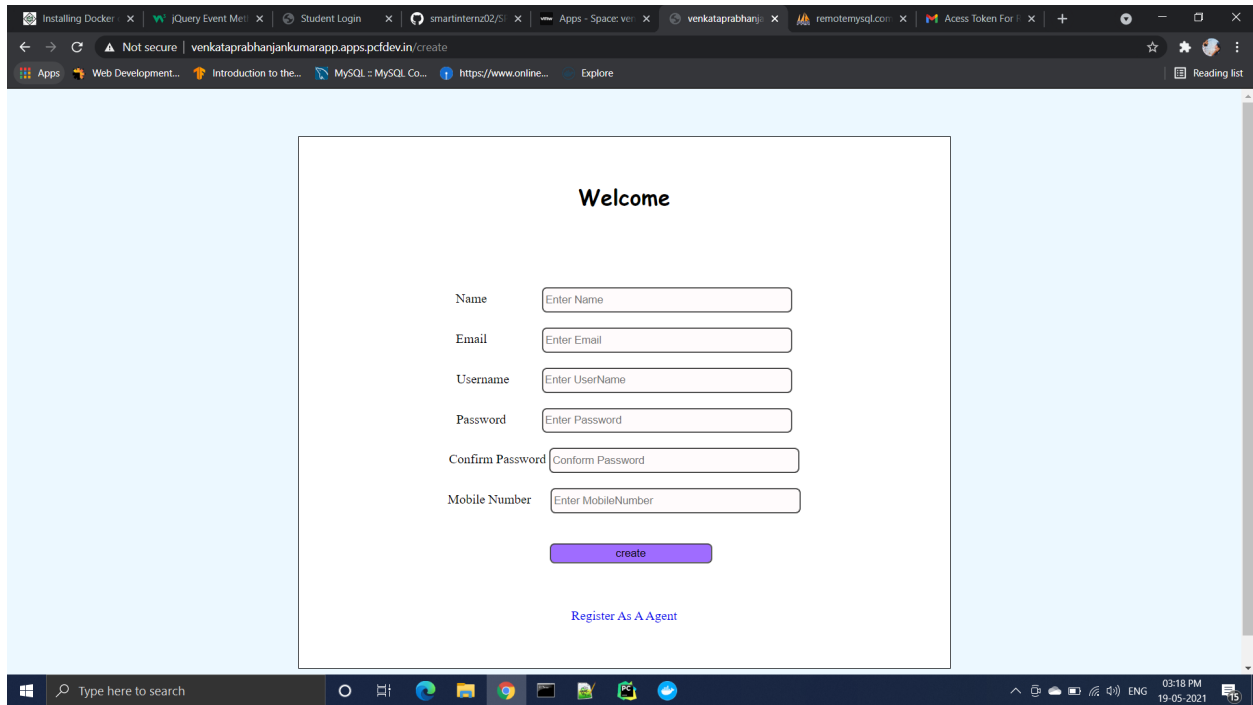
appmanaging84@gmail.com  
Access Token For Registration Y64J00 3:12 PM (7 minutes ago) ☆

appmanaging84@gmail.com via sendgrid.net  
to me 3:17 PM (2 minutes ago) ☆ ↶ ⋮

### Access Token For Registration

49HK22

03:20 PM 19-05-2021



Installing Docker x jQuery Event Me... x Student Login x smartintern202/ x Apps - Space: ve... x Title x remotemysql.com x Access Token For... x +

Not secure | venkataprabhanjankumarapp.apps.pcfdev.in/logindata

Apps Web Development... Introduction to the... MySQL : MySQL Co... https://www.online... Explore

Reading list

My Profile Ask a Question Questions Review Logout

## Welcome To Customer Care Registry

### Hello Customer

Name: Sampath

UserName: sampathkumar

Email: sampath.mathi69@gmail.com

Mobile Number: 9393337241

Type here to search

03:27 PM 19-05-2021

Installing Docker x jQuery Event Me... x Student Login x smartintern202/ x Apps - Space: ve... x Title x remotemysql.com x Access Token For... x +

Not secure | venkataprabhanjankumarapp.apps.pcfdev.in/askquestion

Apps Web Development... Introduction to the... MySQL : MySQL Co... https://www.online... Explore

Reading list

My Profile Ask a Question Questions Review Logout


### Ask Question

Ask a Question

Ask Question

Type here to search

03:28 PM 19-05-2021



[My Profile](#)
[Ask a Question](#)
[Questions Review](#)
[Logout](#)

Ask Question

Good Afternoon Sir I have Issue With My Payment

Ask Question

Windows taskbar showing search bar, taskbar icons (Edge, File Explorer, Chrome, etc.), and system tray (clock, date, network, volume).

[My Profile](#)
[Ask a Question](#)
[Questions Review](#)
[Logout](#)

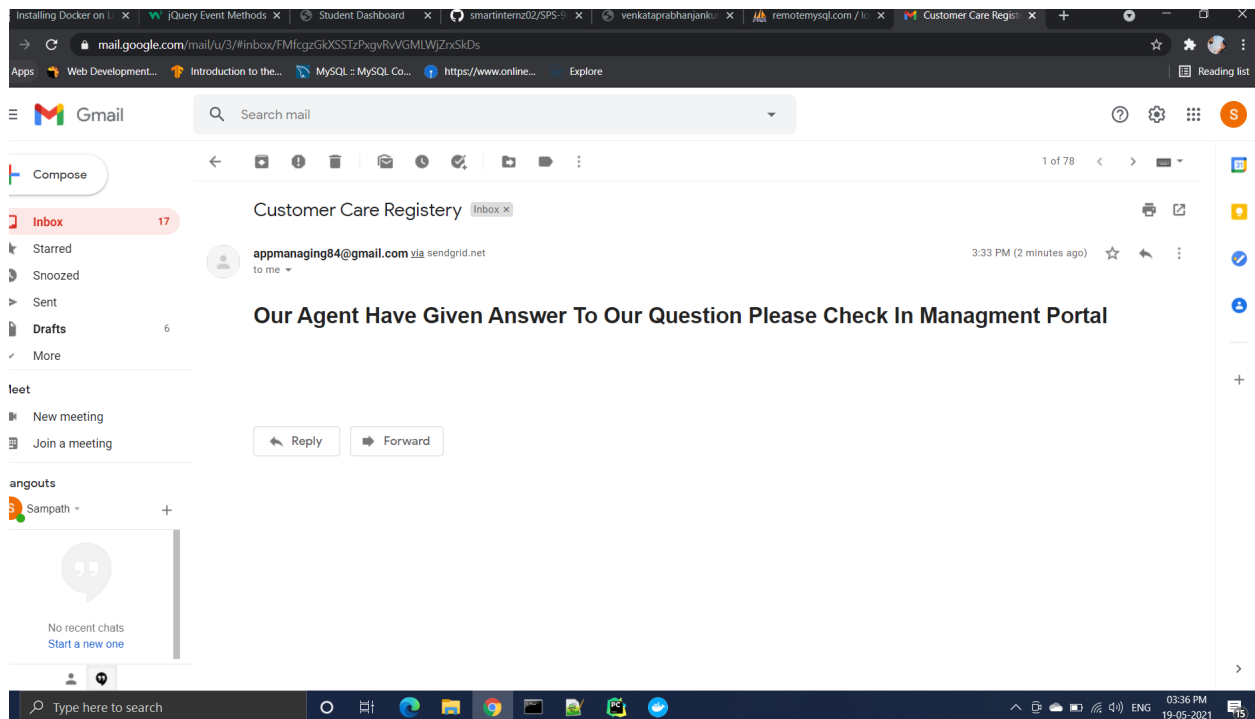
Question	Answer	Status
Good Afternoon Sir I have Issue With My Payment	We Will Check And Intimate You	Answered

Windows taskbar showing search bar, taskbar icons (Edge, File Explorer, Chrome, etc.), and system tray (clock, date, network, volume).

	Question Id	User	Question	Answer	Update
14		sampathkumar	Good AfterNoon Sir I have Issue With My Payment	We Will Check And Intimate You	Update



Question	Answer	Status
Good Afternoon Sir I have Issue With My Payment	We Will Check And Intimate You	Answered



## 6.ADVANTAGES:

1. Boost in customer communication
2. Results in Positive Reviews and brand Image
3. Easy For Customers to solve their doubts

## 7.APPLICATIONS:

1. Integrate with Health Care System,Business System to intract with the Customers
2. Integrate with web based applications

## 8.FUTURE SCOPE:

Future application of this Online Customer Care and Service Center Project includes integration with online marketing, customer and human resource management, all by using web-based technology.

## 9.CONCLUSION:

About security, this system allows the users to use their username and password for a comprehensive and multi-level security. It keeps data secure from unauthorized access, modification or reporting by allowing you to create as many 'users' as you wish.

## 10.BIBILOGRAPHY:

applink:

<https://venkataprabhanjankumarapp.apps.pcfdev.in/>

app.py

```
1 from flask import Flask
2 from flask import render_template,session,request,redirect
3 from flask_mysqlldb import MySQL
4 import random
5 from sendgrid import SendGridAPIClient
6 from sendgrid.helpers.mail import Mail
7
8 app = Flask(__name__)
9
```



```
10mysql=MySQL(app)
11
12api_sendgrid="SG.8VoERMRSjG9gvZATH2ldg.90gaNaxu3AQ
    y5ZenDoAUTj1WBy4-8HQsxBfeKheOGBw"
13
14app.config['MYSQL_HOST']='remotemysql.com'
15app.config['MYSQL_USER']='9JtF7WHjWK'
16app.config['MYSQL_PASSWORD']='nNAXVhfFtY'
17app.config['MYSQL_DB']='9JtF7WHjWK'
18app.secret_key='a'
19@app.route('/')
20def startapp():
21    return redirect('/login')
22@app.route('/adminagentdetails')
23def admin_agent_details():
24    is_admin_agent = False
25    is_admin_user_details = True
26    is_admin_details = False
27    auname = session.get('adminusername')
28    apassword = session.get('adminpassword')
29
30    if auname == app.config['MYSQL_USER']:
31        if apassword ==
            app.config['MYSQL_PASSWORD']:
32            cursor = mysql.connection.cursor()
33            cursor.execute("show tables;")
34            record = cursor.fetchall()
```

```
35         cursor.execute("select * from Users;")
36         table1=cursor.fetchall()
37         cursor.execute("select * from
agentdetails;")
38         table2=cursor.fetchall()
39         cursor.execute("select * from
agentnumbers;")
40         table3=cursor.fetchall()
41         cursor.execute("select * from
userquestions;")
42         table4=cursor.fetchall()
43         cursor.execute("select * from
agentquestions;")
44         table5=cursor.fetchall()
45         cursor.execute("select * from
questions;")
46         table6=cursor.fetchall()
47         print("Hello")
48                                     return
render_template('adminpanel.html',
adminusername=auname,                record=record,
admin_agent=is_admin_agent,
49 admin_user_details=is_admin_user_details,
50 admin_details=is_admin_details,table4=table4,table3
=table3,table2=table2,table1=table1,table5=table5,t
```

```

        able6=table6)
51     else:
52         return render_template('admin.html')
53 @app.route('/adminagentreg')
54 def adminagentregister():
55     auname = session.get('adminusername')
56     apassword = session.get('adminpassword')
57
58     if auname == app.config['MYSQL_USER']:
59         if apassword ==
        app.config['MYSQL_PASSWORD']:
60             is_admin_agent = True
61             is_admin_user_details = False
62             is_admin_details = False
63             return
        render_template('adminpanel.html',admin_agent=is_a
        dmin_agent,admin_user_details=is_admin_user_details,a
        dmin_details=is_admin_details)
64     else:
65         return render_template('admin.html')
66
67 @app.route('/adminagentdatain',methods=['POST'])
68 def adminagentdetailsintake():
69     is_admin_agent=True
70     is_admin_user_details=False
71     is_admin_details=False
72     if request.method=='POST':

```

```

73
    otp_str="ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890"
74
    agent_acces_token=""
75
    for _ in range(6):
76
        agent_acces_token+=otp_str[random.randint(1,35)]
77
        print(agent_acces_token)
78
        agentemailreq=str(request.form['agentaccgen'])
79
        curser = mysql.connection.cursor()
80
        curser.execute("select * from agentnumbers
            where email='"+ " '"+\"'+agentemailreq+\"'+\"';")
81
        records=curser.fetchall()
82
        if(len(records) != 0):
83
                                                    return
        render_template('adminpanel.html',
            admin_agent=is_admin_agent,
84
            admin_user_details=is_admin_user_details,
            admin_details=is_admin_details,emailerr="Email
            Already Exists")
85
        try:
86
            message = Mail(
87
                from_email='appmanaging84@gmail.com',
88
                to_emails=agentemailreq,
89
                subject='Acess Token For

```

```

    Registration',
90         html_content='<h1>Access Token For
    Registration</h1><br/><br/>' + agent_acces_token
91     )
92     sg = SendGridAPIClient(api_sendgrid)
93     response = sg.send(message)
94     print(response.status_code)
95     print("INSERT INTO `agentnumbers`
    (`agentid`, `email`, `activation`, `accstatus`)
    VALUES                                     (NULL,
    "+"\"'+agentemailreq+\"'+\",'+\"'+agent_acces_toke
    n+\"'+\",'+\"'+\"0\"+\"'+");")
96     curser.execute("INSERT INTO
    `agentnumbers` (`agentid`, `email`, `activation`,
    `accstatus`)                               VALUES          (NULL,
    "+"\"'+agentemailreq+\"'+\",'+\"'+agent_acces_toke
    n+\"'+\",'+\"'+\"0\"+\"'+");")
97     mysql.connection.commit()
98                                     return
    render_template('adminpanel.html',
    admin_agent=is_admin_agent,admin_user_details=is_admin
    in_user_details,
99
    admin_details=is_admin_details,upmsg="Created
    Sucessfully")
100
101     except:

```

```
102                                                                 return
    render_template('adminpanel.html',
        admin_agent=is_admin_agent,admin_user_details=is_admin_user_details,
103
        admin_details=is_admin_details,    email_err="Please
        Check Email")
104
105@app.route('/agentregistration')
106def agent_registration():
107     return render_template('agentlogin.html')
108@app.route('/agentregistrationdata',methods=['POST
    '])
109def agent_register_data():
110     if request.method== "POST":
111         acctoken = str(request.form['acctoken'])
112         regname = str(request.form['regname'])
113         regemail= str(request.form['regemail'])
114                                     regusername=
            str(request.form['regusername'])
115                                     regpassword=
            str(request.form['regpassword'])
116                                     regconformpassword=
            str(request.form['regconformpassword'])
117                                     regphonenumber=
            str(request.form['regphonenumber'])
118         curser=mysql.connection.cursor()
```

```
119         curser.execute("select * from agentdetails
    where ausername='"+'\''+regusername+'\'";")
120         userrecord=curser.fetchall()
121         if(len(userrecord)!=0):
122                                     return
        render_template('agentlogin.html',usererr="Username
        Already Exists")
123         print(regemail)
124         print("select * from agentnumbers where
        email='"+'\''+regemail+'\'";")
125         curser.execute("select * from agentnumbers
        where email='"+'\''+regemail+'\'";")
126         records = curser.fetchall()
127         print("he")
128         print(records)
129         if(len(records)==0):
130                                     return
        render_template('agentlogin.html',err_email_msg="Pl
        ease Use Email Id Having Access Id")
131         else:
132                 if(records[0][3]==1):
133                                     return
        render_template('agentlogin.html',errmsg="Account
        Created Already With Given Accces Id")
134         else:
135                                     if regpassword!=
        regconformpassword:
```

```

136                                                                 return
    render_template('agentlogin.html', passmsg='Plesse
    Re enter Password Correctly')
137                     if acctoken==records[0][2]:
138                             print("INSERT INTO
    `agentquestions` (`aqid`, `auser`, `nquestions`)
    VALUES (NULL,"+"\'"+regusername+"\'"+", '0');")
139                             curser.execute("INSERT INTO
    `agentquestions` (`aqid`, `auser`, `nquestions`)
    VALUES (NULL,"+"\'"+regusername+"\'"+", '0');")
140                             curser.execute("INSERT INTO
    `agentdetails` (`agid`, `ausername`, `apassword`,
    `aemail`, `aname`, `amobilenum`) VALUES (NULL,
    "+"\'"+regusername+"\'"+', '+'\'"+regpassword+"\'"+'
    , '+'\'"+regemail+"\'"+', '+'\'"+regname+"\'"+', '+'\'
    "+regphonenumber+"\'"+");")
141                             curser.execute("UPDATE
    `agentnumbers` SET `accstatus` = "+"\'"+"1"+"\'" +
    "
    WHERE          `agentnumbers`.`email`          =
    "+"\'"+regemail+"\'";")
142                             mysql.connection.commit()
143                                                                 return
    render_template('agentlogin.html',sucessmsg='Sucess
    fully Regestered ')
144                     else:
145                                                                 return
    render_template('agentlogin.html',accmsg="Enter

```



```

Valid Access Token")
146@app.route('/admin')
147def admin():
148    is_admin_agent = False
149    is_admin_user_details = False
150    is_admin_agent_details = False
151    is_admin_details = True
152    auname = session.get('adminusername')
153    apassword = session.get('adminpassword')
154
155    if auname == app.config['MYSQL_USER']:
156        if apassword ==
            app.config['MYSQL_PASSWORD']:
157            session['adminusername'] = auname
158            session['adminpassword'] = apassword
159            cursor = mysql.connection.cursor()
160            cursor.execute("show tables;")
161            record = cursor.fetchall()
162            print("Hello")
163            return
    render_template('adminpanel.html',
        adminusername=auname, record=record,
        admin_agent=is_admin_agent,
164        admin_user_details=is_admin_user_details,
        admin_agent_details=is_admin_agent_details,
165

```

```

        admin_details=is_admin_details)
166     else:
167         return render_template('admin.html')
168 @app.route('/adminlogout')
169 def adminlogout():
170     session.pop('adminusername')
171     session.pop('adminpassword')
172     return redirect('/admin')
173 @app.route('/admindata',methods=['POST'])
174 def admin_data():
175     is_admin_agent = False
176     is_admin_user_details = False
177     is_admin_agent_details = False
178     is_admin_details = True
179     if request.method=='POST':
180         auname=str(request.form['auname'])
181         apassword=str(request.form['apassword'])
182         if auname==app.config['MYSQL_USER']:
183                                                     if
184             apassword==app.config['MYSQL_PASSWORD']:
185                 session['adminusername'] = auname
186                 session['adminpassword'] =
187                 apassword
188                 cursor=mysql.connection.cursor()
189                 cursor.execute("show tables;")
190                 record=cursor.fetchall()
191                 print("Hello")

```

```

190                                                                 return
    render_template('adminpanel.html',adminusername=au
name,record=record,admin_agent=is_admin_agent,admin
_user_detals=is_admin_user_detals,admin_agent_detai
ls=is_admin_agent_details,admin_details=is_admin_de
tails)
191                     else:
192                                                                 return
    render_template('admin.html',msg='Login Failed')
193                     else:
194                                                                 return
    render_template('admin.html',msg='Login Failed')
195@app.route('/login')
196def login():
197     uname = session.get('sessusername')
198     passwo = session.get('sesspassword')
199     print(uname)
200     print(passwo)
201     typeofuser = session.get('usertype')
202     if typeofuser == 'rcustomer':
203         try:
204             curser = mysql.connection.cursor()
205             curser.execute(
206                 "select * from Users where
    username=" + "\"" + uname + "\"" + "and passw=" +
    "\"" + passwo + "\"";")
207             records = curser.fetchall()

```

```

208         print(records)
209         print("hell cursor")
210         if len(records) == 0:
211             print("Hii")
212             return render_template('log.html')
213         else:
214             is_question = False
215             is_profile = True
216             is_review = False
217             return
218         render_template('customerpanel.html',    msg="Hello
Customer",
219         record=records,is_profile=is_profile,is_question=is
_question,is_review=is_review)
220     except:
221         print("Exception")
222         return render_template('log.html')
223 if typeofuser == 'ragent':
224     try:
225         curser = mysql.connection.cursor()
226         curser.execute(
227             "select * from agentdetails where
228             ausername=" + "\"" + uname + "\"" + "and
229             apassword=" + "\"" + passwo + "\"';")
230         records = curser.fetchall()
231         print(records)
232         print("hell cursor")

```

```
229         if len(records) == 0:
230             print("Hii")
231             return render_template('log.html')
232         else:
233             is_answer = False
234             is_agent_profile = True
235             return
236         render_template('agentpanel.html', msg="Hello
237         Customer",
238         record=records,is_answer=is_answer,is_agent_profile
239         =is_agent_profile)
240
241
242     except:
243         return render_template('log.html')
244
245     else:
246         return render_template('log.html')
247
248
249 @app.route('/logout')
250 def logout():
251     try:
252         session.pop('sessusername')
253         session.pop('sesspassword')
254         session.pop('usertype')
255         return redirect('/login')
256     except:
257         return render_template('log.html')
258
259 @app.route('/logindata',methods=['POST'])
```

```

252 def logindata():
253     if request.method=='POST':
254
255         rcusorag=request.form['agentcustomerradio']
256         if rcusorag=='rcustomer':
257             uname=str(request.form['uname'])
258             passwo=str(request.form['passwo'])
259             try:
260                 curser=mysql.connection.cursor()
261                 curser.execute("select * from
Users      where      username='"+'\''+uname+'\''+"and
passw='"+'\''+passwo+'\'';")
262                 records=curser.fetchall()
263                 print("hell cursor")
264                 if len(records) == 0:
265                     return
266                 render_template('log.html',msg="Login Failed")
267                 else:
268                     print(records[0][1])
269                     session['sessusername']=records[0][2]
270                     session['sesspassword']=records[0][3]
271                     session['usertype']=rcusorag
272                     is_question = False
273                     is_profile = True
274                     is_review = False

```

```

273                                     return
    render_template('customerpanel.html',msg="Hello
    Customer",record=records,is_profile=is_profile,is_q
    uestion=is_question,is_review=is_review)
274         print(records)
275     except:
276         return render_template('log.html')
277     else:
278         uname = str(request.form['uname'])
279         passwo = str(request.form['passwo'])
280         try:
281             curser = mysql.connection.cursor()
282             curser.execute(
283                 "select * from agentdetails
    where ausername=" + "\"" + uname + "\"" + "and
    apassword=" + "\"" + passwo + "\";")
284             records = curser.fetchall()
285             print("hell cursor")
286             if len(records) == 0:
287                                     return
    render_template('log.html', msg="Login Failed")
288         else:
289             print(records[0][2])
290             print(records[0][1])
291             session['sessusername'] =
    records[0][1]
292             session['sesspassword'] =

```

```

    records[0][2]
293         session['usertype'] = rcusorag
294         is_answer = False
295         is_agent_profile = True
296         print("True")
297         return
    render_template('agentpanel.html', msg="Hello
    Customer", record=records,
298     is_answer=is_answer,
    is_agent_profile=is_agent_profile)
299     except:
300         return render_template('log.html')
301 @app.route('/answerquestions')
302 def answerquestions():
303     agentusername = session.get("sessusername")
304     passwo = session.get('sesspassword')
305     cursor = mysql.connection.cursor()
306     try:
307         cursor.execute(
308             "select * from agentdetails where
    ausername=" + "\"" + agentusername + "\"" + "and
    apassword=" + "\"" + passwo + "\";")
309     except:
310         return render_template('log.html')
311     records = cursor.fetchall()
312     print(records)

```



```
313     print("hell cursor")
314     if len(records) == 0:
315         print("Hii")
316         return render_template('log.html')
317     else:
318         is_answer = True
319         is_agent_profile = False
320         cursor.execute("select * from agentdetails
    where ausername=" + "\"" + agentusername + "\";")
321         agent_records = cursor.fetchall()
322         aemail = agent_records[0][3]
323         print(aemail)
324         cursor.execute("SELECT * FROM `questions`
    WHERE status='0' AND aemail=" + "\"" + aemail +
    "\";")
325         question_records = cursor.fetchall()
326         return render_template('agentpanel.html',
    msg="Hello Customer",
    question_records=question_records,
327     is_answer=is_answer,
    is_agent_profile=is_agent_profile,
    record=agent_records)
328 @app.route('/update', methods=['POST'])
329 def updateanswers():
330     if request.method == 'POST':
331         is_answer = True
```

```

332         is_agent_profile = False
333         uname=request.form['userquestion']
334         answer=request.form['answer']
335         qid=request.form['qid']
336         print(uname)
337         print(answer)
338         cursor = mysql.connection.cursor()
339         agentusername =
            session.get("sessusername")
340         cursor = mysql.connection.cursor()
341         cursor.execute("select * from agentdetails
            where ausername=" + "\"" + agentusername + "\";")
342         agent_records = cursor.fetchall()
343         aemail = agent_records[0][3]
344         print(aemail)
345         print(answer)
346         cursor.execute("UPDATE questions SET
            answer=" + "\"" + answer + "\"" + " WHERE
            aemail=" + "\"" + aemail + "\"" + " AND
            qid=" + "\"" + qid + "\";")
347         cursor.execute(
348             "UPDATE questions SET status=" + "\""
            + '1' + "\"" + " WHERE aemail=" + "\"" + aemail +
            "\"" + " AND qid=" + "\"" + qid + "\";")
349         mysql.connection.commit()
350         cursor.execute("SELECT * FROM `questions`
            WHERE status='0' AND aemail=" + "\"" + aemail +

```

```

    "\';")
351         question_records = cursor.fetchall()
352         cursor.execute("select * from Users where
    username="+"\ '"+uname+"\';")
353         user_records=cursor.fetchall()
354         print(user_records[0][4])
355         user_message = Mail(
356             from_email='appmanaging84@gmail.com',
357             to_emails=user_records[0][4],
358             subject='Customer Care Registry',
359             html_content='<h1>Our Agent Have Given
    Answer To Our Question Please Check In Managment
    Portal</h1><br/><br/>'
360         )
361         sg = SendGridAPIClient(api_sendgrid)
362         response = sg.send(user_message)
363         print(response.status_code)
364         return render_template('agentpanel.html',
    msg="Hello Customer",
    question_records=question_records,
365     is_answer=is_answer,
    is_agent_profile=is_agent_profile,
    record=agent_records)
366 @app.route('/askquestion')
367 def askquestion():
368     uname = session.get('sessusername')

```

```

369     passwo = session.get('sesspassword')
370     curser = mysql.connection.cursor()
371     try:
372         curser.execute(
373             "select * from Users where username="
374             + "\"" + uname + "\"" + "and passw=" + "\"" +
375             passwo + "\"';")
376     except:
377         return render_template('log.html')
378     records = curser.fetchall()
379     print(records)
380     print("hell cursor")
381     if len(records) == 0:
382         print("Hi")
383         return render_template('log.html')
384     else:
385         is_question = True
386         is_profile = False
387         is_review = False
388         return
389         render_template('customerpanel.html',
390             is_profile=is_profile, is_question=is_question,
391             is_review=is_review)
392 @app.route('/recieve', methods=['POST'])
393 def recievequestion():
394     if request.method == 'POST':

```

```

391         question = request.form['question']
392         uname=session.get('sessusername')
393         print(uname)
394         print(question)
395         cursor = mysql.connection.cursor()
396         cursor.execute("select * from
        agentquestions;")
397         record = cursor.fetchall()
398         question_count=record[0][2]
399         print(question_count)
400         cursor.execute("select min(nquestions)
        from agentquestions;")
401         min_count = cursor.fetchall()
402         print(min_count[0][0])
403         cursor.execute("select * from
        agentquestions where
        nquestions="+"\ '"+str(min_count[0][0])+"\ '";")
404         qrecords = cursor.fetchall()
405         cursor.execute("select * from
        userquestions where quser="+"\ '"+uname+"\ '";")
406         user_record=cursor.fetchall()
407         agents_list = []
408         for agent_username in qrecords:
409             agents_list.append(agent_username[1])
410         print(agents_list)
411         print(qrecords)
412         selected_agent =

```

```

    random.choice(agents_list)
413         print(selected_agent)
414             cursor.execute("SELECT * FROM
    agentquestions,agentdetails WHERE
    agentquestions.auser=agentdetails.ausername AND
    auser="+"\ '"+selected_agent+"\';")
415         final_agent = cursor.fetchall()
416         print(final_agent[0][6])
417         print(user_record[0][2])
418         print(min_count[0][0])
419         print(final_agent)
420             print(str(final_agent[0][6]) + "Agent
    Mail")
421         agent_message = Mail(
422             from_email='appmanaging84@gmail.com',
423             to_emails=final_agent[0][6],
424             subject='Customer Tocken',
425             html_content='<h4>You Have Reviewed A
    Question From a Customer </h4><br/><br/>' +
    question
426         )
427         agentsg = SendGridAPIClient(api_sendgrid)
428         response = agentsg.send(agent_message)
429         print(response.status_code)
430         cursor.execute(
431             "SELECT * FROM Users,userquestions
    WHERE Users.username=userquestions.quser AND

```

```

    username=" + "\" + uname + "\";")
432         customer_record = cursor.fetchall()
433         print(customer_record[0][4])
434         customer_message = Mail(
435             from_email='appmanaging84@gmail.com',
436             to_emails=customer_record[0][4],
437             subject='Customer Care',
438             html_content='<h4>Thanks For Using
Customer Care Service You will get answer Soon From
Our Agent </h4><br/><br/>'
439         )
440         costomersg =
SendGridAPIClient(api_sendgrid)
441         response = costomersg.send(agent_message)
442         print(response.status_code)
443         cursor.execute("INSERT INTO `questions`
(`qid`, `cuname`, `question`, `answer`, `status`,
`aemail`)
VALUES
(NULL, '"+ "\" + uname + "\" + ", '"+ "\" + question + "\" + ",
', '0', '"+ "\" + (final_agent[0][6]) + "\" + ");")
444         cursor.execute("UPDATE `userquestions` SET
`nquestions`
=
 '"+ "\" + str((user_record[0][2]+1)) + "\" + " WHERE
quser = '"+ "\" + uname + "\";")
445         cursor.execute("UPDATE `agentquestions`
SET
`nquestions`
=
 '"+ "\" + str((min_count[0][0]+1)) + "\" + " WHERE

```

```

    auser = "+"\'"+(final_agent[0][4])+"\'";"
446         mysql.connection.commit()
447         is_question = True
448         is_profile = False
449         is_review = False
450                                     return
    render_template('customerpanel.html',
    is_profile=is_profile, is_question=is_question,
451
    is_review=is_review,sucessmsg="Your    Question    Is
    Sent To Our Agent You will get answer Soon")
452@app.route('/reviewquestion')
453def review_questions():
454     uname = session.get('sessusername')
455     passwo = session.get('sesspassword')
456     print(uname)
457     print(passwo)
458     typeofuser = session.get('usertype')
459     if typeofuser == 'rcustomer':
460         curser = mysql.connection.cursor()
461         try:
462             curser.execute(
463                 "select * from Users where
    username=" + "\'" + uname + "\'" + "and passw=" +
    "\'" + passwo + "\';")
464         except:
465             return render_template('log.html')

```



```
466         records = curser.fetchall()
467         print(records)
468         print("hell cursor")
469         if len(records) == 0:
470             print("Hii")
471             return render_template('log.html')
472         else:
473             is_question = False
474             is_profile = False
475             is_review = True
476             cursor = mysql.connection.cursor()
477             cursor.execute("select * from
questions where cuname=" + "\"" + uname + "\";")
478             record_questions = cursor.fetchall()
479             return
render_template('customerpanel.html',
is_profile=is_profile, is_question=is_question,
480 is_review=is_review,
record_questions=record_questions)
481@app.route('/create')
482def create():
483     return render_template('login.html')
484@app.route('/createdata',methods=['POST'])
485def create_data():
486     if request.method=='POST':
487         try:
```

```

488         fname=request.form['name']
489         username=str(request.form['username'])
490         password=str(request.form['password'])
491
492         cpassword=str(request.form['conformpassword'])
493         email=str(request.form['email'])
494
495         mnumber=str(request.form['phonenumner'])
496         curs=mysql.connection.cursor()
497         curs.execute("select * from Users;")
498         recorddetails=curs.fetchall()
499         print(recorddetails)
500         if password !=cpassword:
501             return
502             render_template('login.html',msg='Plesse Re enter
503             Password Correctlr')
504             for erecord in recorddetails:
505                 print(erecord)
506                 if str(erecord[2])==username:
507                     return
508                     render_template('login.html',msg='User Already
509                     Exists')
510                 if str(erecord[4])==email:
511                     return
512                     render_template('login.html',msg='Email Already
513                     Exists')
514
515         curs.execute("INSERT INTO

```

```

        `userquestions` (`uqid`, `quser`, `nquestions`)
VALUES (NULL,"+"\`"+username+"\`"+",'+0');")
507         print("INSERT INTO `Users`
(`userid`,`name`, `username`, `passw`, `email`,
`mobilenum`)
VALUES
(NULL,"+"\`"+fname+"\`"+','++"\`"+username+"\`"+','+
+"\`"+password+"\`"+','++"\`"+email+"\`"+','++"\`"+mnu
mber+"\`"+");")
508         curs.execute("INSERT INTO `Users`
(`userid`,`name`, `username`, `passw`, `email`,
`mobilenum`)
VALUES
(NULL,"+"\`"+fname+"\`"+','++"\`"+username+"\`"+','+
+"\`"+password+"\`"+','++"\`"+email+"\`"+','++"\`"+mnu
mber+"\`"+");")
509         mysql.connection.commit()
510         print("commit")
511         return
        render_template('login.html',sucess_msg="Sucessfull
y Created ")
512     except:
513         return render_template('login.html')
514     else:
515         return render_template('login.html')
516
517 if __name__ == '__main__':
518     app.run(debug=True,port=8080,host='0.0.0.0')
519

```

## admin.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <style>
6         .field
7         {
8             width:300px;
9             height:25px;
10            background-color:#ffffbf;
11            border-style: ridge;
12            border-color:gray;
13            border-radius:6px
14        }
15        #log1
16        {
17            padding-left:350px;
18            padding-top:50px;
19            padding-right:350px;
20        }
21        #sub
22        {
23            width:200px;
24            height:25px;
25            background-color:#9f6cff;
26            border-style: ridge;
```

[illegible]

[illegible]

## adminagent.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <style>
6         .field
7         {
8             width:300px;
9             height:25px;
10            background-color:#ffffbf;
11            border-style: ridge;
12            border-color:gray;
13            border-radius:6px
14        }
15    #log1
```

```

16      {
17          padding-left:350px;
18          padding-top:50px;
19          padding-right:350px;
20      }
21      #sub
22      {
23          width:200px;
24          height:25px;
25          background-color:#9f6cff;
26          border-style: ridge;
27          border-color:gray;
28          border-radius:6px;
29      }
30  </style>
31</head>
32<body>
33    <center>
34        <div id="log1">
35            <div style="border:1px solid
black;background-color:#ffffff">
36                <br/><br/>
37                <h1
style="font-size:20pt;font-family:cursive;">Adminis
trator Login</h1><br/><br/>
38
39                {{ msg }}
40                <br/>
41                <form method="post"
action="/adminagentdatain">
42                    Enter Email Id

```

```

43         <input type="text"
            name="agentaccgen" placeholder="Enter UserName "
            class="field" required/> <br/><br/>
44         <input type="submit" value="Generate Token" id="sub">
45         <br/><br/>
46     </form>
47 </div>
48 </div>
49 </center>
50</body>
51</html>

```

**adminpanel.html**

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Title</title>
6     <style>
7         *
8         {
9             padding:0;
10            box-sizing:border-box;
11        }
12        #log1
13        {
14            padding-left:150px;
```



```
15         padding-top:50px;
16         padding-right:150px;
17     }
18     .menu-bar
19     {
20         background:rgb(100,100,0);
21         text-align:center;
22     }
23     .menu-bar ul
24     {
25         display:inline-flex;
26         list-style:none;
27         color:#fff;
28     }
29     .menu-bar ul li
30     {
31         width:200px;
32         margin:0px;
33         padding:5px;
34     }
35     .field
36     {
37         width:300px;
38         height:25px;
39         background-color:#ffffbf;
40         border-style: ridge;
41         border-color:gray;
42         border-radius:6px
43     }
44     #sub
45     {
```

```
46         width:200px;
47         height:25px;
48         background-color:#9f6cff;
49         border-style: ridge;
50         border-color:gray;
51         border-radius:6px;
52     }
53     table, th, td {
54         border: 1px solid black;
55     }
56 </style>
57</head>
58<body style="background-color: antiquewhite">
59     <div class="menu-bar">
60         <ul>
61             <li> <a href="/adminagentreg"
62 style="text-decoration:none ;font-size:
63 large">Agent Registration</a> </li>
64             <li><a href="/adminagentdetails"
65 style="text-decoration:none;">Agent Customer
66 Details</a></li>
67             <li><a href="/admin"
68 style="text-decoration:none;">Admin
69 Details</a></li>
70             <li><a href="/adminlogout"
71 style="text-decoration:none;">Logout</a></li>
72         </ul>
73     </div>
74
75     <div id="log1">
76         <div style="border:1px solid
```

[illegible]

```

93         {% endif %}
94     </form>
95     {% if admin_user_details %}
96         <br/><br/>
97         <h4>Users</h4>
98         <table style="width: 800px;height:
auto;border-style: inset">
99             <th>User Id</th>
100             <th>Name</th>
101             <th>Username</th>
102             <th>Password</th>
103             <th>Email</th>
104             <th>Mobile Number</th>
105             {% for row in table1 %}
106                 <tr>
107                     {% for i in range(6)
108                         <td>
109                             {{ row[i] }}
110                         </td>
111                     {% endfor %}
112                 </tr>
113             {% endfor %}
114         </table>
115         <br/><br/>
116         <h4>Agent Details</h4>
117         <table style="width:
800px;height: auto;border-style: inset">
118             <th>Agent Id</th>
119             <th>Username</th>
120             <th>Password</th>

```

```

121         <th>Email</th>
122         <th>Name</th>
123         <th>Mobile Number</th>
124         {% for row in table2 %}
125             <tr>
126                 {% for i in range(6)
127                     %}
128                     <td>
129                         {{ row[i] }}
130                     </td>
131                 {% endfor %}
132             </tr>
133         {% endfor %}
134     </table>
135     <br/><br/>
136     <h4>Agent Numbers</h4>
137     <table style="width:
138         800px;height: auto;border-style: inset">
139         <th>Agent Id</th>
140         <th>Email</th>
141         <th>Activation</th>
142         <th>Status</th>
143         {% for row in table3 %}
144             <tr>
145                 {% for i in range(4)
146                     %}
147                     <td>
148                         {{ row[i] }}

```

```

149             {% endfor %}
150         </table>
151         <br/><br/>
152         <h4>User Questions</h4>
153         <table style="width:
154             800px;height: auto;border-style: inset">
155             <th>Qid</th>
156             <th>Username</th>
157             <th>No Of Questions</th>
158             {% for row in table4 %}
159                 <tr>
160                     {% for i in range(3)
161                         %}
162                         <td>
163                             {{ row[i] }}
164                         </td>
165                     {% endfor %}
166                 </tr>
167             {% endfor %}
168         </table>
169         <br/><br/>
170         <h4>Agent Questions</h4>
171         <table style="width:
172             800px;height: auto;border-style: inset">
173             <th>Qid</th>
174             <th>Username</th>
175             <th>No Of Questions</th>
176             {% for row in table5 %}
177                 <tr>
178                     {% for i in range(3)
179                         %}

```

```

176             <td>
177                 {{ row[i] }}
178             </td>
179         {% endfor %}
180     </tr>
181     {% endfor %}
182 </table>
183 <br/><br/>
184 <h4>Questions</h4>
185 <table style="width:
    800px;height: auto;border-style: inset">
186     <th>qid</th>
187     <th>Customer
    Username</th>
188     <th>Question</th>
189     <th>Answer</th>
190     <th>Status</th>
191     <th>Agent Email</th>
192     {% for row in table6 %}
193         <tr>
194             {% for i in range(6)
    %}
195                 <td>
196                     {{ row[i] }}
197                 </td>
198             {% endfor %}
199         </tr>
200     {% endfor %}
201 </table>
202 <br/><br/>
203 <br/>

```

```
204             {% endif %}
205         </center>
206     </div>
207 </div>
208 </body>
209 </html>
```

#### agentlogin.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <style>
6         .rfield
7         {
8             width:300px;
9             height:25px;
10            background-color:#ffbfbc;
11            border-style: ridge;
12            border-color:gray;
13            border-radius:6px
14        }
15        #log1
16        {
17            padding-left:350px;
18            padding-top:50px;
19            padding-right:350px;
20        }
21        #sub
22        {
```



```

23         width: 200px;
24         height: 25px;
25         background-color:#9f6cff;
26         border-style: ridge;
27         border-color: gray;
28         border-radius: 6px;
29     }
30 </style>
31</head>
32<body style="background-color:#ecf8ff">
33    <center>
34        <div id="log1">
35            <div style="border:1px solid
black;background-color:#ffffff;padding-left:30px;pa
dding-right:30px;">
36                <br/><br/>
37                <h1
style="font-size:20pt;font-family:cursive;">Agent
Registration </h1><br/><br/>
38                <p style="font-size:15pt;color:red">
{{accmsg}}</p>
39                {{msg}}
40                {{ err_email_msg}}
41                {{ errmsg }}
42                {{ accmsg }}
43                {{ usererr }}
44                {{ passmsg }}
45                {% if sucessmsg %}
46                    Back To <a href="/login">Login</a>
47                {% endif %}
48            <br/>

```

[illegible]



```
6      .rfield
7      {
8          width:300px;
9          height:25px;
10         background-color:#ffffbf;
11         border-style: ridge;
12         border-color:gray;
13         border-radius:6px
14     }
15     #log1
16     {
17         padding-left:350px;
18         padding-top:50px;
19         padding-right:350px;
20     }
21     #sub
22     {
23         width: 200px;
24         height: 25px;
25         background-color:#9f6cff;
26         border-style: ridge;
27         border-color: gray;
28         border-radius: 6px;
29     }
30     </style>
31</head>
32<body style="background-color:#ecf8ff">
33     <center>
34         <div id="log1">
35         <div style="border:1px solid
black;background-color:#ffffff;padding-left:30px;pa
```

[illegible]

[illegible]

69

70</html>

### log.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <style>
6         .field
7         {
8             width:300px;
9             height:25px;
10            background-color:#ffbfbc;
11            border-style: ridge;
12            border-color:gray;
13            border-radius:6px
14        }
15        #log1
16        {
17            padding-left:350px;
18            padding-top:50px;
19            padding-right:350px;
20        }
21        #sub
22        {
23            width:200px;
24            height:25px;
25            background-color:#9f6cff;
26            border-style: ridge;
```

[illegible]



[illegible]

67

68</html>

**agentpanel.html:**

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Title</title>
6     <style>
7         *
8         {
9             padding:0;
10            box-sizing:border-box;
11        }
12        #log1
13        {
14            padding-left:100px;
15            padding-top:50px;
16            padding-right:100px;
17        }
18        .menu-bar
19        {
20            background:rgb(100,100,0);
21            text-align:center;
22        }
23        .menu-bar ul
24        {
25            display:inline-flex;
26            list-style:none;
```

```
27         color:#fff;
28     }
29     .menu-bar ul li
30     {
31         width:200px;
32         margin:0px;
33         padding:5px;
34     }
35     #sub
36     {
37         width:200px;
38         height:25px;
39         background-color:#9f6cff;
40         border-style: ridge;
41         border-color:gray;
42         border-radius:6px;
43     }
44     </style>
45</head>
46<body style="background-color: antiquewhite">
47     <div class="menu-bar">
48         <ul>
49             <li> <a href="/login"
50                 style="text-decoration:none;">My Profile</a> </li>
51             <li><a href="/answerquestions"
52                 style="text-decoration:none;">Answer
53                 Questions</a></li>
54             <li><a href="/logout"
55                 style="text-decoration:none;">Logout</a></li>
56         </ul>
57     </div>
```

```

54
55     <div id="log1">
56         <div style="border:1px solid
57             black;background-color:#ffffffa">
58             <center>
59                 {% if is_agent_profile %}
60                     <h1> Welcome To Customer Care
61                     Registry </h1>
62                     <h1>{{ msg }}</h1><br/>
63                     <h2>Name:    {{
64                     record[0][4]}}</h2><br/>
65                     <h2>UserName:    {{
66                     record[0][4]}}</h2><br/>
67                     <h2>Email:    {{
68                     record[0][3]}}</h2><br/>
69                     <h2>Mobile Number:    {{
70                     record[0][5]}}</h2><br/>
71                 {% endif %}
72                 {% if is_answer %}
73                     <br/><br/>
74                     <form method="post"
75                     action="/update" id="answerform">
76                         <table style="text-align:
77                             center;height: auto;width: auto">
78                             <th>Question Id</th>
79                             <th>User</th>
80                             <th>Question</th>
81                             <th>Answer</th>
82                             <th>Update</th>
83                             <tr>

```

```

77                                     <td><input
    type="text" value="{{ question_records[0][0] }}"
    name="qid" readonly style="border-style:
    hidden"></td>
78                                     <td> <input
    type="text" value="{{ question_records[0][1] }}"
    name="userquestion" readonly style="border-style:
    hidden"></td>
79                                     <td>{{
    question_records[0][2] }}</td>
80                                     <td><textarea
    form="answerform" rows="5" cols="80"
    name="answer"></textarea></td>
81                                     <td><input
    type="submit" value="Update" id="sub"></td>
82
83                                     </tr>
84
85                                 </table>
86                            </form>
87                    {% endif %}
88                </center>
89            </div>
90        </div>
91</body>
92</html>

```

**customerpanel:**

```

1 <!DOCTYPE html>
2 <html lang="en">

```

```
3 <head>
4   <meta charset="UTF-8">
5   <title>Title</title>
6   <style>
7       *
8       {
9           padding:0;
10          box-sizing:border-box;
11      }
12      #log1
13      {
14          padding-left:150px;
15          padding-top:50px;
16          padding-right:150px;
17      }
18      .menu-bar
19      {
20          background:rgb(100,100,0);
21          text-align:center;
22      }
23      .menu-bar ul
24      {
25          display:inline-flex;
26          list-style:none;
27          color:#fff;
28      }
29      .menu-bar ul li
30      {
31          width:200px;
32          margin:0px;
33          padding:5px;
```

```
34     }
35     #sub
36     {
37         width:200px;
38         height:25px;
39         background-color:#9f6cff;
40         border-style: ridge;
41         border-color:gray;
42         border-radius:6px;
43     }
44     table, th, td {
45         border: 1px solid black;
46     }
47 </style>
48</head>
49<body style="background-color: antiquewhite">
50     <div class="menu-bar">
51         <ul>
52             <li> <a href="/login"
53 style="text-decoration:none;">My Profile</a> </li>
54             <li><a href="/askquestion"
55 style="text-decoration:none;">Ask a
56 Question</a></li>
57             <li><a href="/reviewquestion"
58 style="text-decoration:none;">Questions
59 Review</a></li>
60             <li><a href="/logout"
61 style="text-decoration:none;">Logout</a></li>
62         </ul>
63     </div>
64 </body>
65 </html>
```

```

59     <div id="log1">
60         <div style="border:1px solid
        black;background-color:#ffffffa">
61             <center>
62                 {% if is_profile %}
63                     <h1> Welcome To Customer Care
        Registry </h1>
64                     <h1>{{ msg }}</h1><br/>
65                     <h2>Name:  {{
        record[0][1]}}</h2><br/>
66                     <h2>UserName:  {{
        record[0][2]}}</h2><br/>
67                     <h2>Email:  {{
        record[0][4]}}</h2><br/>
68                     <h2>Mobile Number:  {{
        record[0][5]}}</h2><br/>
69                 {% endif %}
70                 {% if is_question %}
71                     <form method="post"
        action="/recieve" id="qform">
72                         {% if sucessmsg %}
73                             <p>{{ sucessmsg }}</p>
74                         {% endif %}
75                         <br/><br/>
76                         <label for="ques">Ask
        Question</label><br/>
77                         <textarea form="qform"
        name="question" placeholder="Ask a Question"
        rows="7" cols="100" id="ques" required></textarea>
78                         <br/><br/>
79                         <input type="submit"

```



```

value="Ask Question" id="sub">
80         </form>
81         <br/><br/>
82         {% endif %}
83         {% if is_review %}
84         <br/><br/>
85         <table style="width:
800px;text-align: center;height: auto">
86             <th>Question</th>
87             <th>Answer</th>
88             <th>Status</th>
89             {% for question in
record_questions %}
90                 <tr>
91                     <td>{{
question[2]}}</td>
92                     <td>{{
question[3]}}</td>
93                     <td>
94                         {% if
question[4]==1 %}
95                             Answered
96                         {% else %}
97                             Not Answered
98                         {% endif %}
99                     </td>
100                 </tr>
101             {% endfor %}
102         </table>
103         <br/><br/>
104         {% endif %}

```

```
105                 </center>
106             </div>
107         </div>
108     </body>
109 </html>
```