

Project Report

Project Title : CUSTOMER CARE REGISTRY

Team ID: PNT2022TMID24365

Team Members:

Team Leader : KEVIN ROSHAN D

Team member : ABISHEK M

Team member : NAVEEN M

Team member : RAMESH L

Team member : JEGDESHVAR K

1. INTRODUCTION

1.1 PROJECT OVERVIEW:

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 a 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 role and responsibility of the admin are to take care of the whole process. Starting from Admin login followed by the agent creation and assigning the customer's complaints. Finally, He will be able to track the work assigned to the agent and a notification will be sent to the customer.

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

1.2 Purpose:

The main purpose of this Online Customer Care Registry software is to develop an information system to store, maintain, update and process data relating to the customer's complaints. It will prepare various reports to aid in smooth and speedy functioning of Customer Care functions.

2. LITERATURE SURVEY:

2.1 Existing problem:

Customer care employees (CCEs) are a great source of concepts for brand-new, improved customer services. CCEs can identify patterns in unmet and under met needs because they serve a large number of clients. CCEs are able to make recommendations that expand on already-existing skills because they are internal to the company as opposed to being external, which produces ideas that are simpler to put into practice. Employee suggestion gathering has been the subject of extensive research and practice, but very little of this work has explored how CCEs might be gathered into a transient online community to cocreate unique ideas. When using CCEs for internal collaborative crowdsourcing, a general framework, consequences, and future research paths are laid forth based on the findings.

2.2 References:

- 1.Arvind Malhotra, Ann Majehrzak, William Bonfield,Steve Myers "Engaging customer care employees in internal collaborative crowdsourcing".
- 2.Zain Raza, Syed M.,Raza "Customer Care Application"
- 3.Okello, Dickens, Driwaru, Winnyfred "Online clinical feedback system for tracking customer care issue"
- 4.Ahmed Elmorshidy "Applying The Technology Acceptance And Service Quality Models To Live Customer Support"

2.3 Problem Statement Definition:

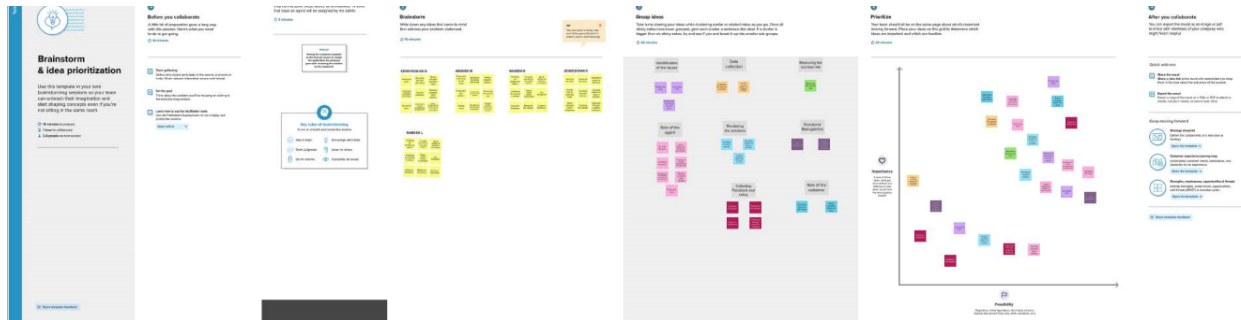
| Problem Statement (PS) | I am (Customer) | I'm trying to | But | Because | Which makes me feel |
|------------------------|-----------------|---|---|---|---|
| PS-1 | A Shopkeeper | Develop my shop with good profit. | My shop resides in a location where people rarely visits the department. | Insufficient knowledge about how to develop the shop and not knowing how to advertise about the shop. | Mentally low and unable to know how to attract the customers. |
| PS-2 | An Educator | Teach the students and gain them knowledge. | I don't know how to make the students to understand the subject. | Have no experience before. | Low self-esteem due to inability in effectively teaching the students. |
| PS-3 | A Farmer | Maximize the production of crops, protect it from pests and make profits. | My field is easily affected by the pests and agriculture department rarely visits my field. | Insufficient knowledge about how to treat crops. | Being unable to effectively treat the diseased plant makes me feel depressed. |

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming:



3.3 Proposed Solution:

Proposed Solution Template:

Project team shall fill the following information in proposed solution template.

| S.No. | Parameter | Description |
|-------|--|---|
| 1. | Problem Statement (Problem to be solved) | The Customer have many issue and don't have idea of how to solve the issue on their own. In this situation they need an external help. |
| 2. | Idea / Solution description | Application helps the customer in processing their complaints .An agent is assigned to solve the issue .The Admin will take care in assigning the agents to solve the issue of the customers. |
| 3. | Novelty / Uniqueness | Getting solution will be easier if the customer raised the same issue which was solved earlier. |
| 4. | Social Impact / Customer Satisfaction | The Admin will respond as soon as possible to solve the issue so that waiting time of the customer to solve the issue is very less. |
| 5. | Business Model (Revenue Model) | Customer satisfaction is the main motive of every business so by this application business will be developed and attracts many customers so revenue will be improved. |
| 6. | Scalability of the Solution | The Agent will be assigned as soon as the issue is raised by the customer and the issue will be solved in lesser time. |

3.4 Problem Solution fit:

| | | | | |
|--|--|---|--|---------------------------------------|
| Define CS, fit into CC | <p>1.CUSTOMER SEGEMENTS</p> <p>The people from any field, who needs guidance of the agent to solve their issue. Example: An Educator who had no previous experience need the guidance about how to solve the issue in while using online applications for their educating purpose.</p> | <p>6.CONSTRAINTS</p> <ul style="list-style-type: none"> Insufficient knowledge about how to use the application. The Customer needs to wait patiently till their issue is solved. | <p>5.AVAILABLE SOLUTIONS</p> <ul style="list-style-type: none"> The user can upload their issue in the portal. Admin will view the issue and assign an agent to the issue. Agent will solve the issue as soon as possible and provide the best solution. | Explore AS, differentiate |
| Focus on J&P, tap into BE, understand RC | <p>2.JOBS TO BE DONE/PROBLEMS</p> <ul style="list-style-type: none"> A Shopkeeper who needs to develop their shop has insufficient knowledge and advertise the shop needs the guidance. An Educator who had no previous experience wants the guidance from the experienced. A Farmer who is trying to maximize the production of crops and protects it from pests may have insufficient knowledge about how to treat it. | <p>9.PROBLEM ROOT CAUSE</p> <ul style="list-style-type: none"> One of the reasons that this problem exists is because of the poor knowledge as they cannot find a clear solution for the issue The users adapt to the application to find the solution for the issues in their daily life. | <p>7.BEHAVIOUR</p> <ul style="list-style-type: none"> Users can use the software application functionalities and get the mail when an Agent is assigned to solve the issue. Thereby finding the necessary features in the application is significant. | Focus on J&P, tap into BE, understand |

4. REQUIREMENT ANALYSIS

4.1 Functional requirement & 4.2 Non-Functional requirement

Functional Requirements:

Following are the functional requirements of the proposed solution.

| FR No. | Functional Requirement (Epic) | Sub Requirement (Story / Sub-Task) |
|--------|-------------------------------|--|
| FR-1 | User Registration | Registration through Form Registration through Gmail Registration through LinkedIn |
| FR-2 | User Confirmation | Confirmation via Email |
| FR-3 | Role of Customer | Customer will enter their problem in UI screen |
| FR-4 | Role of Admin | After receiving the problem from the customer the admin assigns the task to the agent. |
| FR-5 | Role of Agent | Agent will be notified after the task is assigned . Provide the solution as soon as possible. Saves the solution so that it can be used for the similar problem. |

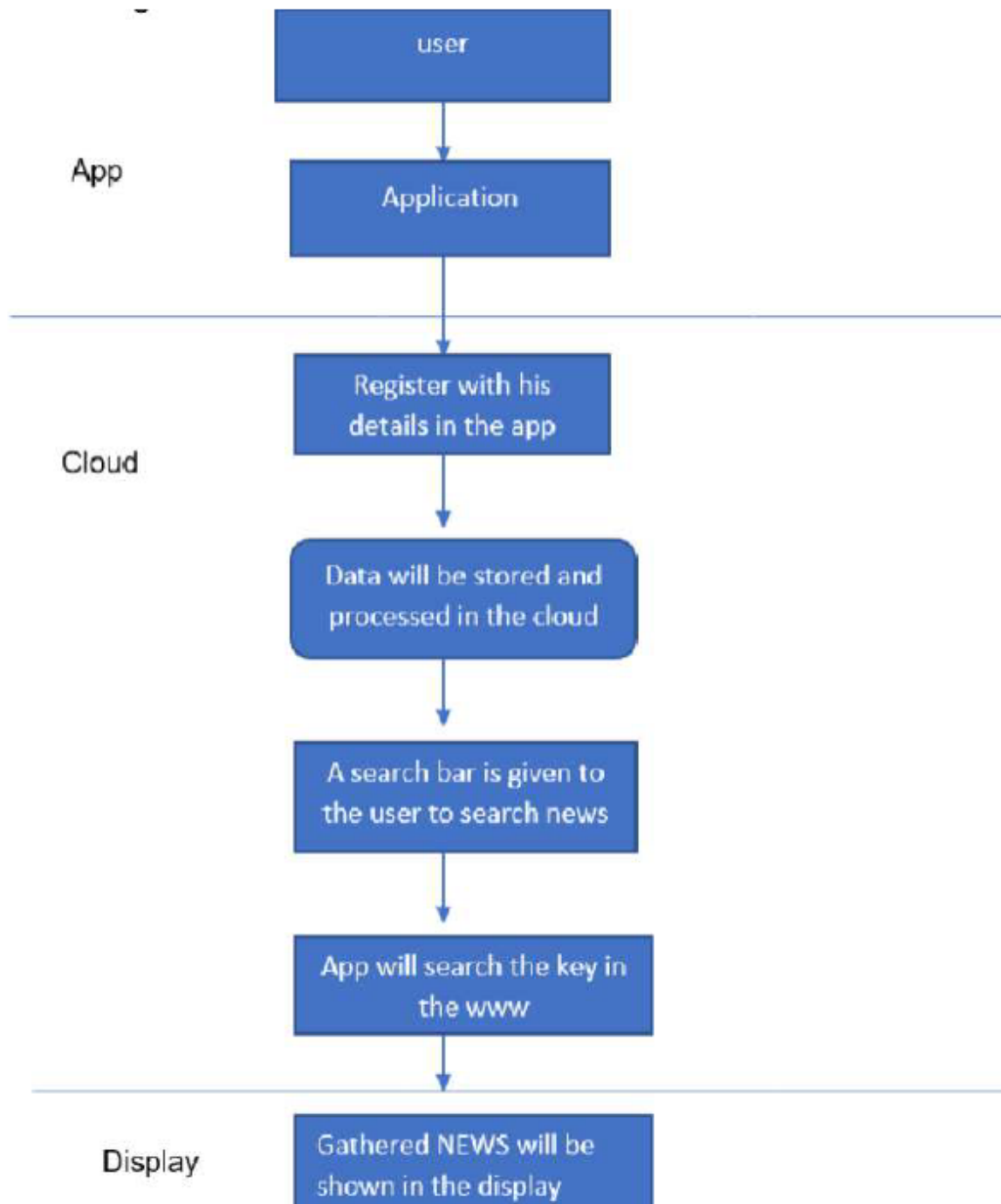
Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

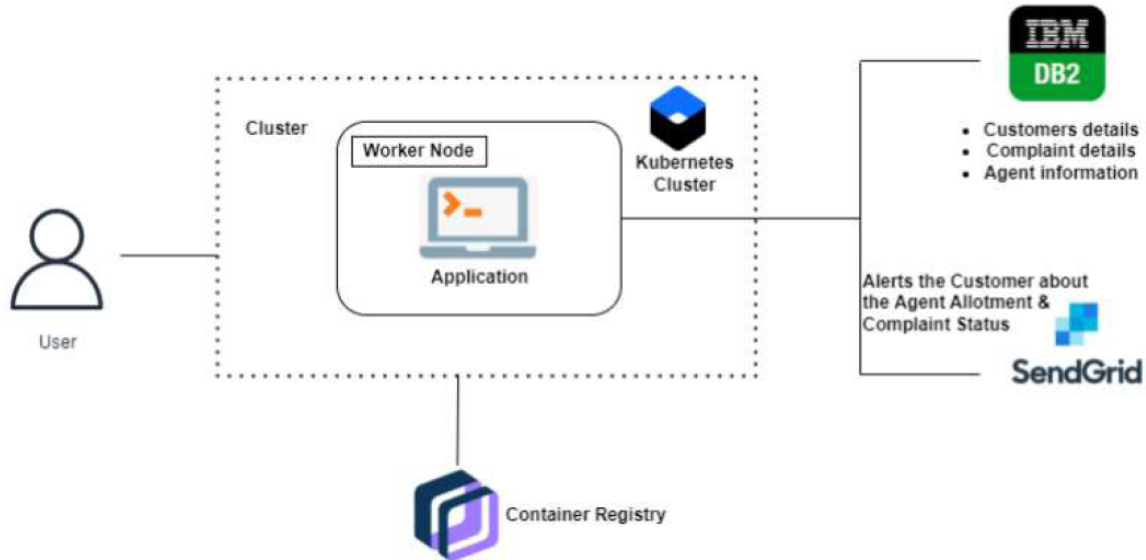
| FR No. | Non-Functional Requirement | Description |
|--------|----------------------------|---|
| NFR-1 | Usability | The Customer have many issue and don't have idea of how to solve the issue on their own . In this case they need an external help so they can use this application to solve the issue |
| NFR-2 | Security | Customer information will he held safe. The Registered user information will not be shared to any other user. |
| NFR-3 | Reliability | The solution of the issue is recorded so that it can be used further if the same problem occurs there by making the system more reliable. |
| NFR-4 | Performance | The Admin will respond as soon as possible to solve the issue so that waiting time of the customer to solve the issue is very less. |
| NFR-5 | Availability | The application is user-friendly and easy to use so that every user can use it .Admin will provide the solution as soon as possible so that every user can understand the solution. |
| NFR-6 | Scalability | The Agent will be assigned as soon as the issue is raised by the customer and the issue will be solved in lesser time. |

5. PROJECT DESIGN:

5.1 Data Flow Diagrams:



5.2 Solution & Technical Architecture:



5.3 User Stories:

| User Type | Functional Requirement (Epic) | User Story Number | User Story / Task | Acceptance criteria | Priority | Release |
|-------------------------|-------------------------------|-------------------|---|---|----------|----------|
| Customer (Mobile user) | Registration | USN-1 | As a user, I can register for the application by entering my email, password, and confirming my password. | I can access my account / dashboard | High | Sprint-1 |
| | | USN-2 | As a user, I will receive confirmation email once I have registered for the application | I can receive confirmation email & click confirm | High | Sprint-1 |
| | | USN-3 | As a user, I can register for the application through Facebook | I can register & access the dashboard with Facebook Login | Low | Sprint-2 |
| | | USN-4 | As a user, I can register for the application through Gmail | | High | Sprint-1 |
| | Login | USN-5 | As a user, I can log into the application by entering email & password | | High | Sprint-1 |
| | Dashboard | | | | | |
| Customer (Web user) | Log in | USN-6 | As a user, I can access the application through website | I can access my account / dashboard | High | |
| Customer Care Executive | 24x7 service | USN-7 | As a User, I may need help in using the app/website | Any help need | High | |
| Administrator | Owner/Chief executive | — | He/she will access the datas in the app | Product behaviour/improvement purposes | Low | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

6. PROJECT PLANNING & SCHEDULING:

6.1 Sprint Planning & Estimation & 6.2 Sprint Delivery Schedule:

| TITLE | DESCRIPTION | DATE |
|---|---|-------------------|
| Literature Survey & Information Gathering | Literature survey on the selected project & gathering information by referring the, technical papers, research publications etc. | 28 SEPTEMBER 2022 |
| Prepare Empathy Map | Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements | 24 SEPTEMBER 2022 |
| Ideation | List the by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance. | 25 SEPTEMBER 2022 |
| Proposed Solution | Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc. | 23 SEPTEMBER 2022 |
| Problem Solution Fit | Prepare problem - solution fit document. | 30 SEPTEMBER 2022 |
| Solution Architecture | Prepare solution architecture document. | 28 SEPTEMBER 2022 |

| | | |
|---|---|-----------------|
| Customer Journey | Prepare the customer journey maps to understand the user interactions & experiences with the application (entry to exit). | 20 OCTOBER 2022 |
| Functional Requirement | Prepare the functional requirement document. | 8 OCTOBER 2022 |
| Data Flow Diagrams | Draw the data flow diagrams and submit for review. | 9 OCTOBER 2022 |
| Technology Architecture | Prepare the technology architecture diagram. | 10 OCTOBER 2022 |
| Prepare Milestone & Activity List | Prepare the milestones & activity list of the project. | 22 OCTOBER 2022 |
| Project Development - Delivery of Sprint-1, 2, 3 & 4 | Develop & submit the developed code by testing it. | IN PROGRESS.. |

7. TESTING

7.1 Test Cases

LOGIN PAGE(CUSTOMER):

User has No Complaint

Steps To Execute :

1. Enter URL(http:// 169.51.206.150:32144/) and click go

2. To the User Login page and submit Your

Credentials Actual Result :No Complaint should

shown status : PASS

ADMIN PAGE(CUSTOMER_LIST)

Select the database for User

Steps To Execute :

1. Enter URL(http:// 169.51.206.150:32144/) and click go

2. To the User Login page and submit Your Credentials



2. Docker-Deployment:

```
3. apiVersion: apps/v1
4. kind: Deployment
5. metadata:
6.   name: flask-node-deployment
7. spec:
8.   replicas: 1
9.   selector:
10.    matchLabels:
11.      app: flasknode
12.   template:
13.     metadata:
14.       labels:
15.         app: flasknode
16.     spec:
17.       containers:
18.       - name: flasknode
19.         image: au.icr.io/customer-care-ibm/customer-care-ibm
20.         ports:
21.         - containerPort: 5000
```

```
File Edit Selection View Go Run Terminal Help Customer Care Registry.py - Visual Studio Code
Customer Care Registry.py X Customer Care Registry.html
C:\Users\> My PC > Desktop > src > Customer Care Registry.py
4 import reimport ibm_db
5 import pandas
6 import ibm_db_db1
7 from sqlalchemy import create_engine
8 engine = create_engine('sqlite:///')
9 echo = False
10 dsn_hostname = "125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud"
11 dsn_uid = "pkg90003"
12 dsn_pwd = "EkbUyegC0542s2dm"
13 dsn_driver = "{IBM DB2 ODBC DRIVER}"
14 dsn_database = "BLUD8"
15 dsn_port = "30426"
16 dsn_protocol = "TCPIP"
17 dsn_security = "SSL"
18 dsn = {
19   "DRIVER={0};"
20   "DATABASE={1};"
21   "HOSTNAME={2};"
22   "PORT={3};"
23   "PROTOCOL={4};"
24   "UID={5};"
25   "PWD={6};"
26   "SECURITY={7};".format(dsn_driver, dsn_database, dsn_hostname, dsn_port, dsn_protocol, dsn_uid, dsn_pwd,dsn_security)
27 try:
28   conn = ibm_db.connect(dsn, "", "")
29   print ("Connected to database: ", dsn_database, "as user: ", dsn_uid, "on host: ", dsn_hostname)
30 except:
31   print ("Unable to connect: ", ibm_db.conn_errormsg() )
32 app = Flask(__name__)
33 app.config.from_object(__name__)
34 app.config["SECRET_KEY"] = '7d441f27d441f27567d441f2b6176a'
35 @app.route("/")
```

Recorded with iTop Screen Recorder

Activate Windows
Go to Settings to activate Windows.

Ln 85, Col 1 Spaces: 4 UTF-8 CRLF Python 2:06 AM 17/11/2022

8. ADVANTAGES & DISADVANTAGES

Advantages:

Understanding Your Customer and Meeting Their Needs
Professionalism
Respect
Prompt services
Use of Technology
Prevents and Reduces Customer Churn

Disadvantages:

Resolution Delays
Outsourcing
Cost
Service Quality
Solutions and Resolutions

9. CONCLUSION

Companies today are modernizing customer care, using advanced AI to ensure a positive customer experience starting from the first interaction and throughout the buyer's journey. Customer care and customer service together help create a positive customer experience, or the overall impression a person has when interacting with your company. Both are vital, but there are subtle differences in how they are implemented.

High-quality customer care is proactive. The needs of customers throughout the buyer's journey are anticipated, making customers feel supported. That, in turn, helps create an emotional connection between the customer and the company.

Customer service is reactive. Here, the focus is on helping customers solve problems or answer questions before purchase, either in a self-serve fashion or via the customer support team.

10. FUTURE SCOPE

Consumer expectations are extremely high, putting increased pressure on companies to improve their customer relationships. According to Forrester , only 18% of customers said they would continue doing business with a brand that has disappointed them.

This can lead to lost information when the same person reaches out via multiple channels. When a customer service agent doesn't know the whole story and the customer has to repeatedly share the problem, it leaves both people frustrated. So we have to overcome these types of issues.

11. APPENDIX

Source Code

app.py:

```
from flask import Flask, render_template, request, redirect, url_for
from database import Database
from waitress import serve
```

```
app = Flask(__name__, static_url_path='')
@app.route('/')
def index():
    return render_template('index.html')
```

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

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

```
@app.route('/validateRegister', methods=['GET', 'POST'])
def validateRegister():
    name = request.form['name']
    email = request.form['email']
    password = request.form['password']
    db = Database()
    db.createUser(name, email, password)
    return render_template('login.html')
```

```
@app.route('/validateLogin', methods=['GET', 'POST'])
def validateLogin():
    email = request.form['email']
    password = request.form['password']
    db = Database()
    if email=='admin' and password==db.loginAdmin(email)[0]:
        return redirect(url_for('admin'))
```



```
else:
    if password == db.loginUser(email)[0]:
        return redirect(url_for('customercare'))
    else:
        return "Invalid Credentials"
```

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

```
@app.route('/admin')
def admin():
    db = Database()
    agents = db.getAgents()
    print(agents)
    return render_template('admin.html', agt=agents)
```

```
@app.route('/agentAssign')
def agentAssign():
    db = Database()
    complaints = db.getComplaints()
    print(complaints)
    return render_template('agentAssign.html', comp=complaints)
```

```
@app.route('/submit', methods=['POST'])
def submit():
    name = request.form['name']
    email = request.form['email']
    complaint = request.form['complaint']
    db = Database()
    db.registerComplaint(name, email, complaint)
    return redirect(url_for('customercare'))
```

```
if __name__ == "__main__":
    # serve(app, host="0.0.0.0", port=8080)
    app.run()
```

GitHub & Project Demo Link

GitHub Repo Link: <https://github.com/IBM-EPBL/IBM-Project-53987-1661586993>