

CUSTOMER CARE REGISTRY

CLOUD APP DEVELOPMENT

TEAM MENTOR: Ms. SINDUJA (Assistant Professor)

TEAM MEMBERS:

DEENATHAYALAN S – TEAM LEAD

CHUJEETHA R

DHIVESH D

CHANDRU S

| S.NO | TOPIC | PAGE.NO |
|-------------|---|----------------|
| 1 | INTRODUCTION | 5 |
| | 1.1 Project Overview | |
| | 1.2 Purpose of project | |
| 2 | LITERATURE SURVEY | 6 |
| | 2.1 Existing Problem | |
| | 2.2 References | |
| | 2.3 Problem Statement Definition | |
| 3 | IDEATION & PROPOSED SOLUTION | 7 |
| | 3.1 Empathy map canvas | |
| | 3.2 Proposed Solution | |
| | 3.3 Problem solution fit | |
| 4 | REQUIREMENT ANALYSIS | 8 |
| | 4.1 Functional Requirement | |
| | 4.2 Non-functional Requirements | |
| 5 | PROJECT DESIGN | 9 |
| | 5.1 Data flow diagrams | |
| | 5.2 Solution and Technical architecture | |
| | 5.3 User stories | |
| 6 | PROJECT PLANNING & SCHEDULING | 12 |
| | 6.1 Sprint planning & Estimation | |
| | 6.2 Sprint Delivery Schedule | |
| 7 | CODING & SOLUTIONING (Explain the features added in the project along with code) | 14 |
| | 7.1 Feature 1 | |
| | 7.2 Feature 2 | |

| | | |
|-----------|-------------------------------------|-----------|
| 8 | TESTING | 39 |
| | 8.1 Test cases | |
| | 8.2 User Acceptance Testing | |
| 9 | RESULTS | 40 |
| 10 | ADVANTAGES AND DISADVANTEGES | 42 |
| 11 | CONCLUSION | 43 |

CUSTOMER CARE REGISTRY

1. INTRODUCTION

1.1 Overview of the Project – CUSTOMER CARE REGISTRY MANAGEMENT SYSTEM

An effective customer care complaints management system is integral to providing quality customer service. It helps to measure customer satisfaction and is a useful source of information and feedback for improving services. Often customers are the first to identify when things are not working properly.

The customer care registry management system is a web based project. A virtual chat bot that can help you with immediate customer queries without needing to contact a person to deal with simple queries of the customer

Implementing effective complaints management systems within public sector agencies:

- Improves Companies internal complaints handling
- Reduces recurring complaints
- Improves standards of service to the community
- Raises standards of administrative decision-making

The scope of the system is quite wide. It can be implemented on a WAP-enabled mobile handset, thus providing the Customers and the Providers, the ease of accessing the projects and their status without any difficulty and within no time

2.2 Purpose of the project

An online comprehensive Customer Care Solution is to manage customer interaction and complaints with the Service Providers over a chat bot like query system. The system should have capability to integrate with any Service Provider from any domain or industry like Banking, Telecom, Insurance, etc.

Customer Service also known as Client Service is the provision of service to customers Its significance varies by product, industry and domain. In many cases customer services is more important if the information relates to a service as opposed to a Customer.

Customer Service may be provided by a Service Representatives Customer Service is normally an integral part of a company's customer value proposition.

2. LITERATURE SURVEY

2.1 PROBLEM IN EXISTING SYSTEM

The existing system is a semi-automated at where the information is stored in the form of excel sheets in disk drives. The information sharing to the Volunteers, Group members, etc. is through mailing feature only. The information storage and maintenance is more critical in this system. Tracking the member's activities and progress of the work is a tedious job here. This system cannot provide the information sharing by 24x7 days.

2.2 REFERENCES

- [1]. M. Baye, Managerial Economics & Business Strategy McGraw-Hill Education, London, Abacus:
The Undercover Economist, vol. 2013, pp. 12-23, 2017.
- [2]. J. Obliquity Kay, why our goals are best achieved indirectly, London:
Profile Book, pp. 15- 67, 2011.
- [3]. P. Keat and P.K. Young, Managerial Economics Global Edition, London:
Pearson, pp. 23- 46, 2014.
- [4]. Bai changhong and Liu Chi, "study on customer loyalty of service enterprises and its determinants [J]",
nankai business review, no. 06, pp. 64-69, 2002.
- [5]. Chip R. Bell, The service edge: 101 companies that profit from customer care by Ron Zemke with Dick Schaaf, New York:New American Library, pp. 584, 1989

2.3 PROBLEM STATEMENT & PROPOSED SYSTEM

The development of this new system objective is to provide the solution to the problems of existing system. By using this new system, we can fully automate the entire process of the current system.

The new system would like to make as web-enabled so that the information can be shared between the members at any time using the respective credentials. To track the status of an individual process, the status update can be centralized using the new system.

Being a web-enabled system, the process can be accessed across the world over internet.

This system also providing the features like Chatting, Mailing between the members, updating the process status in centralized location, generated reports can also be exporting to the application like MS-Excel, PDF etc.

The entire system is categorized as different modules like Admin module, Volunteer module, etc. at where we can classify the functionality as an individual process.

| S.NO \$ TITLE | PROPOSED WORK | TOOLS USED/ ALGORITHM | TECHNOLOGY | PROS/CONS |
|--|---|---|---|---|
| REAL WORLD SMART CHATBOT FOR CUSTOMER CARE USING A SOFTWARE AS A SERVICE(SAAS) ARCHITECTURE | The journal employ chatbot for customer care. This is done by providing a human way interaction using LUIS and cognitive services | <ul style="list-style-type: none"> • AWS public cloud • AWS Lambda • API Gateway • LUIS • IBM Watson chatbot | <ul style="list-style-type: none"> • Cloud computing • Machine Learning | <p>This proposes a robust, scalable and extensible architecture with a feasible tech stack</p> <p>The server creates the room functionality where the customer needs to be persistent over time</p> |

| S.NO \$ TITLE | PROPOSED WORK | TOOLS USED/ ALGORITHM | TECHNOLOGY | PROS/CONS |
|-------------------------------------|---|---|--|--|
| CHATBOT FOR CUSTOMER SERVICE | In this paper customer trust chatbots to provide the required support. Chatbots represent a potential means for automating customer service | <ul style="list-style-type: none"> • Chatbot • JavaScript | <ul style="list-style-type: none"> • Cloud computing • Artificial Intelligence • Machine Learning | This provides automated customer service with the use of cloud |

| S.NO \$ TITLE | PROPOSED WORK | TOOLS USED/ ALGORITHM | TECHNOLOGY | PROS/CONS |
|--|---|--|---|--|
| IMPLEMENTING CONTINUOUS CUSTOMER CARE | In this paper we employ SAAS model which introduces drastic improvement to the situation, as service provider can now have direct access to user data | <ul style="list-style-type: none"> • Javascript • HTML • Google Analytics | <ul style="list-style-type: none"> • Cloud computing • Machine Learning | <p>Feedback loops are used that allow the provider to capture feedback</p> <p>It is not always easy for SaaS providers to know what customers are experiencing</p> |

3. IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS:



3.2 PROPOSED SOLUTION:

- The development of this new system objective is to provide the solution to the problems of existing system. By using this new system, we can fully automate the entire process of the current system.
- Also the new addition of a virtual chat bot helps customers in solving immediate problems that are simple without having the need of a human help to solve the problem
- More over the chat bot helps in improving communication between the customer and the service provider efficiently.

3.3 PROBLEM SOLUTION FIT:

- Being a web-enabled system, the process can be accessed across the world over internet.
- This system also providing the features like Chatting, Mailing between the members, updating the process status in centralized location, generated reports can also be exporting to the application like MS-Excel, PDF etc.
- The entire system is categorized as different modules like Admin module, Volunteer module, etc. at where we can classify the functionality as an individual process.

4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS:

| FR No | Functional Requirement (Epic) | Sub Requirement (Story/ Sub-Task) |
|-------|-------------------------------|--|
| 1 | User Registration | Registration through Form Registration through Gmail Registration through Google |
| 2 | User Confirmation | Confirmation via Email Confirmation via OTP |
| 3 | User Login | Login via Google Login with Email id and Password |
| 4 | Admin Login | Login via Google Login with Email id and Password |
| 5 | Query Form | Description of the issues Contact information |
| 6 | E-mail | Login alertness |
| 7 | Feedback | Customer feedback |

4.2 NON-FUNCTIONAL REQUIREMENTS:

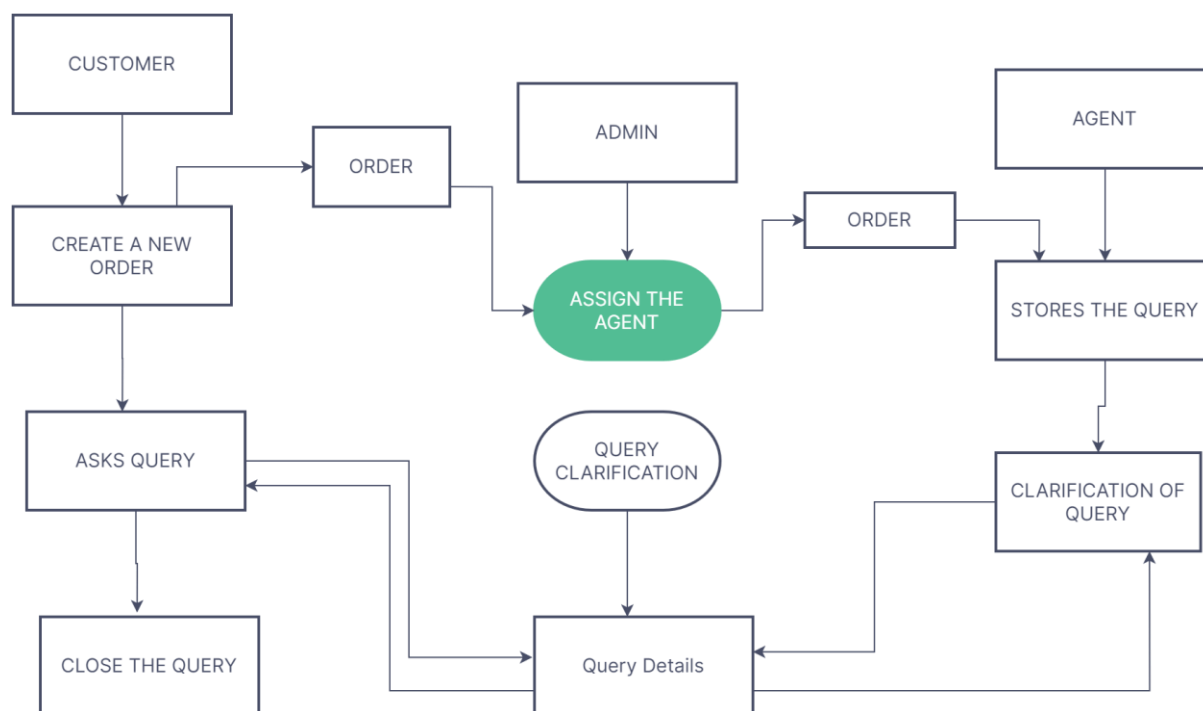
| FR No | Non-Functional Requirement | Description |
|-------|----------------------------|--|
| 1 | Usability | To provide the solution to the problem |
| 2 | Security | Track of login authentication |
| 3 | Reliability | Tracking of decade status through email |
| 4 | Performance | Effective development of web application |
| 5 | Availability | 24/7 service |
| 6 | Scalability | Agents' scalability as per the number of customers |

SYSTEM REQUIREMENTS:

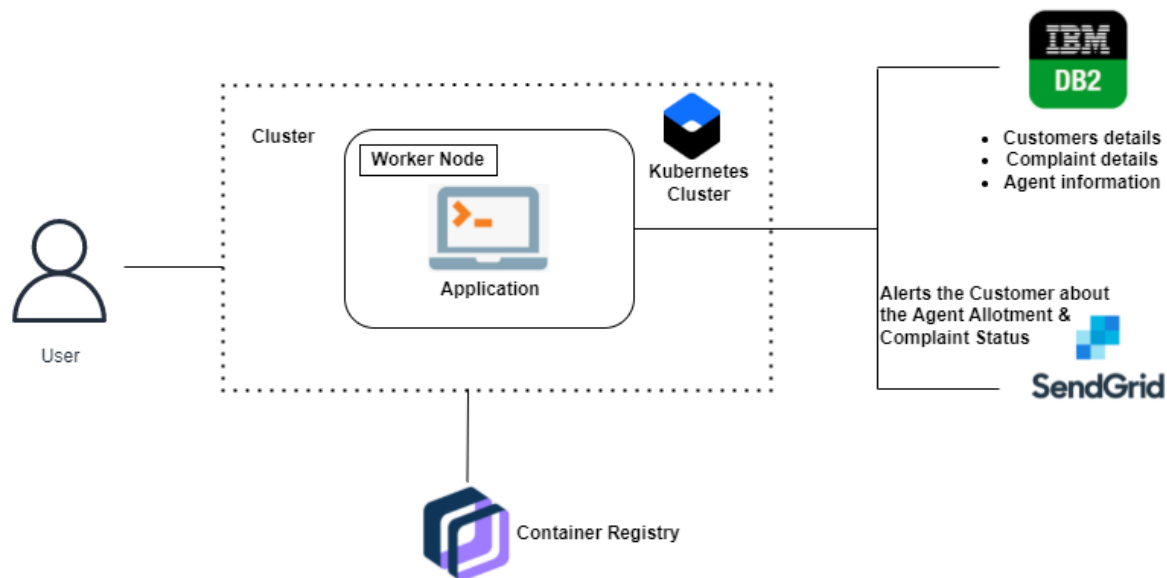
- Windows 7 or later versions
- 2 GB RAM
- Compatible web browser(CHROME)

5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS:



5.2 SOLUTION & TECHNICAL ARCHITECTURE:



5.3 USER STORIES:

- A user story is an informal, general explanation of a software feature written from the perspective of the end user. Its purpose is to articulate how a software feature will provide value to the customer.
- It's tempting to think that user stories are, simply put, software system requirements. But they're not.
- A key component of agile software development is putting people first, and a user story puts end users at the center of the conversation. These stories use non-technical language to provide context for the development team and their efforts. After reading a user story, the team knows why they are building, what they're building, and what value it creates.

| USER TYPE | FUNCTIONAL REQUIREMENT (EPIC) | USER STORY NUMBER | USER STORY / TASK | ACCEPTANCE CRITERIA | PRIORITY | RELEASE |
|------------------------|-------------------------------|-------------------|---|--|----------|----------|
| Customer (Mobile user) | Registration | USN-1 | As a customer, I can register for the application by entering my email, password, and confirming my password. | I can access my account / dashboard | High | Sprint-1 |
| | login | USN-2 | As a customer, I can login to the application by entering correct email and password. | I can access my account/dashboard. | High | Sprint-1 |
| | Dashboard | USN-3 | As a customer, I can see all the orders raised by me. | I get all the info needed in my dashboard. | Low | Sprint-2 |
| | Order creation | USN-4 | As a customer, I can place my order with the detailed description of my query | I can ask my query | Medium | Sprint-2 |
| | Address Column | USN-5 | As a customer, I can have conversations with the assigned agent and get my queries clarified | My queries are clarified. | High | Sprint-3 |
| | Forgot password | USN-6 | As a customer, I can reset my password by this option in case I forgot my old password. | I get access to my account again | Medium | Sprint-4 |
| | Order details | USN-7 | As a Customer I can see the current stats of order. | I get a better understanding | Medium | Sprint-4 |
| Agent (web user) | Login | USN-1 | As an agent I can login to the application by entering Correct email and password. | I can access my account / dashboard. | High | Sprint-3 |
| | Dashboard | USN-2 | As an agent, I can see the order details assigned to me by admin. | I can see the tickets to which I could answer. | High | Sprint-3 |
| | Address column | USN-3 | As an agent, I get to have conversations with the customer and clear his/her doubts | I can clarify the issues. | High | Sprint-3 |
| | Forgot password | USN-4 | As an agent I can reset my password by this option in case I forgot my old password. | I get access to my account again. | Medium | Sprint-4 |

| USER TYPE | FUNCTIONAL REQUIREMENT (EPIC) | USER STORY NUMBER | USER STORY / TASK | ACCEPTANCE CRITERIA | PRIORITY | RELEASE |
|---------------------|-------------------------------|-------------------|--|--|----------|----------|
| Admin (Mobile user) | Login | USN-1 | As an admin, I can login to the application by entering Correct email and password | I can access my account/dashboard | High | Sprint-1 |
| | Dashboard | USN-2 | As an admin I can see all the orders raised in the entire system and lot more | I can assign agents by seeing those order. | High | Sprint-1 |
| | Agent creation | USN-3 | As an admin I can create an agent for clarifying the customers queries | I can create agents. | High | Sprint-2 |
| | Assignment agent | USN-4 | As an admin I can assign an agent for each order created by the customer. | Enable agent to clarify the queries. | High | Sprint-1 |
| | Forgot password | USN-5 | As an admin I can reset my password by this option in case I forgot my old password. | I get access to my account. | High | Sprint-1 |

6. PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION:

| Sprint | Functional Requirement | User Story Number | User Story / Task | Story Points | Priority | Team Members |
|----------|------------------------|-------------------|--|--------------|----------|---|
| Sprint-1 | User Panel | CUS-1 | The user will login into the website and go through the services available on the webpage | 20 | High | DEENATHAYALAN S CHUJEETHA R DHIVESH D |
| Sprint-2 | Admin panel | CUS-2 | The role of the admin is to check out the database about the availability and have a track of all the things that the users are going to service | 20 | High | DEENATHAYALAN S CHANDRU S |
| Sprint-3 | Chat bot | CUS-3 | The user can directly talk to Chatbot regarding the services. Get the recommendations based on information provided by the user. | 20 | High | DHIVESH D CHANDRU S |
| Sprint-4 | Final Delivey | CUS-4 | Container of applications using docker Kubernetes and deployment the application. Create the documentation and final submit the application | 20 | High | CHUJEETHA R DHIVESH D CHANDRU S |

6.2 SPRINT DELIVERY SCHEDULE:

| Sprint | Total Story Points | Duration | Sprint Start Date | Sprint End Date | Story Points Completed (as on Planned End Date) | Sprint Release Date (Actual) |
|---------------|---------------------------|-----------------|--------------------------|------------------------|--|-------------------------------------|
| Sprint-1 | 20 | 7 days | 24-Sep-2022 | 30-Sep-2022 | | 30-Sep-2022 |
| Sprint-2 | 20 | 7 days | 30-Sep-2022 | 07-Oct-2022 | | 07-Oct-2022 |
| Sprint-3 | 20 | 7 days | 07-Oct-2022 | 14-Oct-2022 | | 14-Oct-2022 |
| Sprint-4 | 20 | 7 days | 14-Oct-2022 | 21-Oct-2022 | | 21-Oct-2022 |

7. CODING & SOLUTIONING

BACKEND:

App.py:

```
# importing the modules
from flask import Flask, render_template, request, redirect, session, url_for
from flask_mail import Mail, Message
from flask_mysql import MySQL
import MySQLdb.cursors
from passlib.hash import pbkdf2_sha256
import config

# app config
app = Flask(__name__)
app.config['MYSQL_HOST'] = "localhost"
app.config['MYSQL_USER'] = "root"
app.config['MYSQL_PASSWORD'] = ""
app.config['MYSQL_DB'] = "Deena"
app.config['MAIL_SERVER'] = 'smtp.hostinger.com'
app.config['MAIL_PORT'] = 465
app.config['MAIL_USERNAME'] = "customercare@experdsolutions.com"
app.config['MAIL_PASSWORD'] = "Chandru@123"
app.config['MAIL_USE_SSL'] = True
app.config['MAIL_USE_TLS'] = False

mysql = MySQL(app)
app.secret_key = 'returnzero'
mail = Mail(app)

# routes

# home
@app.route("/", methods=['GET', "POST"])
def home():

    return render_template("main.html")

@app.route("/complaint", methods=['GET', "POST"])
def loginHome():
    if ('user' not in session.keys()) or (session['user'] == None):
        return redirect(url_for('login'))
    else:
        cursor = mysql.connection.cursor()
```

```

        cursor.execute("SELECT * FROM User WHERE id = % s",[session['user']])
        userdetails = cursor.fetchone()
        print(userdetails)
        if userdetails[3] == 2:
            return render_template("home.html",user=userdetails)
        elif userdetails[3] == 1:
            cursor.execute("SELECT * FROM Tickets WHERE
agent=%s",[session['user']])
            tickets = cursor.fetchall()
            print(tickets, session['user'])
            return
render_template("home.html",user=userdetails,tickets=tickets)
        else:
            if request.method == "POST":
                title = request.form['title']
                description = request.form['description']
                cust_id = session['user']
                cursor = mysql.connection.cursor()
                cursor.execute("INSERT INTO
Tickets(customer,title,description)
VALUES(%s,%s,%s)",(cust_id,title,description))
                mysql.connection.commit()
                cursor.execute("SELECT * FROM User WHERE id = %
s",[session['user']])
                userdetails = cursor.fetchone()
                cursor.execute("SELECT * FROM Tickets WHERE customer =
%s",[session['user']])
                tickets = cursor.fetchall()
                return render_template("home.html",msg="Ticket
Filed",user=userdetails,tickets=tickets)
                cursor = mysql.connection.cursor()
                cursor.execute("SELECT * FROM User WHERE id = %
s",[session['user']])
                userdetails = cursor.fetchone()
                cursor.execute("SELECT * FROM Tickets WHERE customer =
%s",[session['user']])
                tickets = cursor.fetchall()

                return
render_template("home.html",user=userdetails,tickets=tickets)

# about
@app.route("/aboutus",methods=['GET',"POST"])
def about():
    return render_template("About.html")

# contact
@app.route("/contactus",methods=['GET',"POST"])

```

```

def contact():
    return render_template("Contact.html")

# user account registration
@app.route("/register",methods=["GET","POST"])
def register_account():
    if request.method == "POST":
        username = request.form['username']
        email = request.form['email']
        password = request.form['password']
        hashed_password = pbkdf2_sha256.hash(password)
        cursor = mysql.connection.cursor()
        cursor.execute("INSERT INTO User(username,email,password,role)
VALUES(%s,%s,%s,%s)",(username,email,hashed_password,0))
        mysql.connection.commit()
        msg = Message('registration customer
care',sender="customercare@experdsolutions.com",
            recipients=[email]
        )
        msg.body = '''
            Account creation in customer care registry was successful.
            for raising tickets, login with your email id and password.
            Thank You
        '''
        mail.send(msg)
        return redirect(url_for("login"))
    return render_template("register.html")

# login
@app.route('/login',methods=["GET","POST"])
def login():
    if request.method == "POST":
        email = request.form['email']
        password = request.form['password']
        cursor = mysql.connection.cursor()
        cursor.execute("SELECT * FROM User WHERE email = % s",[email])
        userdetails = cursor.fetchone()
        if userdetails:
            if pbkdf2_sha256.verify(password,userdetails[2]):
                session['user'] = userdetails[4]
                return redirect(url_for("loginHome"))

            else:
                msg = "Incorrect Password"
        else:
            msg = "User does not exist"
        return render_template("login.html",msg=msg)
    return render_template("login.html")

```



```

# logout
@app.route("/logout")
def logout():
    session['user'] = None
    return redirect(url_for("home"))

# ticket detail
@app.route("/ticket/<int:id>", methods=["GET", "POST"])
def ticket_detail(id):
    cursor = mysql.connection.cursor()
    cursor.execute("SELECT * FROM Tickets WHERE id=%s", [id])
    ticket = cursor.fetchone()
    print("cx", ticket)
    cursor.execute("SELECT * FROM User WHERE id=%s", [ticket[1]])
    customer = cursor.fetchone()
    cursor.execute("SELECT * FROM User WHERE id=%s", [session['user']])
    user = cursor.fetchone()
    cursor.execute("SELECT * FROM User WHERE role=1")
    all_users = cursor.fetchall()
    cursor.execute("SELECT * FROM User WHERE id=%s", [ticket[2]])
    agent = cursor.fetchone()
    print(agent, customer, ticket)
    if agent is None:
        agent = [None, None]
    if user is None:
        return redirect(url_for("login"))
    if request.method == "POST":
        agent = request.form['agent']
        print(agent)
        cursor.execute("UPDATE Tickets SET agent= %s WHERE id = %s", (agent, id))
        cursor.execute("UPDATE Tickets SET progress='assigned' WHERE id = %s", [id])
        mysql.connection.commit()
        cursor.execute("SELECT email FROM User WHERE id=%s", [agent])
        agent_mail = cursor.fetchone()[0]
        msg = Message('Assigned
Ticket', sender="customercare@experdsolutions.com",
                    recipients=[agent_mail]
                    )

        # send mail to agent
        msg = Message('Assigned
Ticket', sender="customercare@experdsolutions.com",
                    recipients=[agent_mail]
                    )
        cursor.execute("SELECT email FROM User WHERE id=%s", [ticket[1]])

```

```

        customer = cursor.fetchone()[0]
        msg.body = f'''
            You have been assigned a ticket.
            Ticket Title: {ticket[3]}
            posted by: {customer}
        '''
        mail.send(msg)

        # send mail to customer
        msg = Message('Ticked
Progress',sender="customercare@experdsolutions.com",
            recipients=[customer]
        )
        msg.body = f'''
            Dear Customer,
            Your Ticket progress has been Updated and
            Assigned to an Agent of ours.
            Agent : {agent_mail}
        '''
        mail.send(msg)
        return redirect(url_for("panel"))
    return
render_template("details.html",ticket=ticket,agent=agent,customer=customer,user=user,all_users=all_users)

# admin register
@app.route("/admin/register",methods=["GET","POST"])
def admin_register():
    if request.method == "POST":
        username = request.form['username']
        email = request.form['email']
        password = request.form['password']
        secret_key = request.form['secret']
        if secret_key == "12345":
            hashed_password = pbkdf2_sha256.hash(password)
            cursor = mysql.connection.cursor()
            cursor.execute("INSERT INTO User(username,email,password,role)
VALUES(%s,%s,%s,%s)",(username,email,hashed_password,2))
            mysql.connection.commit()
            return redirect(url_for("login"))
        else:
            return render_template("admin_register.html",msg="Invlaid Secret")

    return render_template("admin_register.html")

# promote agent
@app.route("/panel",methods=['GET','POST'])
def panel():

```

```

id = session['user']
if id is None:
    return redirect("login")
cursor = mysql.connection.cursor()
cursor.execute("SELECT * FROM User WHERE id=%s",[id])
user_details = cursor.fetchone()
if user_details[3] != 2:
    return "You do not have administrator privileges"
else:
    cursor.execute("SELECT * FROM User WHERE role=0")
    all_users = cursor.fetchall()
    cursor.execute("SELECT * FROM Tickets WHERE progress IS NULL")
    tickets = cursor.fetchall()
    print("bv",tickets)
    if request.method == "POST":
        user_id = request.form['admin-candidate']
        cursor = mysql.connection.cursor()
        cursor.execute("UPDATE User SET role=1 WHERE id = %s",[user_id])
        mysql.connection.commit()
        cursor.execute("SELECT * FROM User WHERE id = %s",[user_id])
        promoted_agent = cursor.fetchone()
        msg = Message('Promoted to
Agent',sender="customercare@experdsolutions.com",recipients=[promoted_agent[1]
])
        msg.body = """
        Dear User,
        You have been promoted to an Agent in the Customer-Care-
Registry.
        You will be able to handle tickets for the customer from now
on.
        Congratulations.
        """
        mail.send(msg)
        return redirect(url_for("panel"))
    return
render_template("panel.html",all_users=all_users,user=user_details,tickets=tic
kets)

# accept ticket
@app.route("/accept/<int:ticket_id>/<int:user_id>")
def accept(ticket_id,user_id):
    cursor = mysql.connection.cursor()
    cursor.execute("SELECT * FROM User WHERE id = %s",[user_id])
    agent = cursor.fetchone()
    cursor.execute("SELECT * FROM Tickets WHERE id=%s",[ticket_id])
    ticket = cursor.fetchone()
    cursor.execute("SELECT email FROM User WHERE id=%s",[ticket[1]])
    customer = cursor.fetchone()

```

```

        if agent[4] == ticket[2]:
            cursor.execute("UPDATE Tickets SET progress='accepted' WHERE
id=%s",[ticket_id])
            mysql.connection.commit()
            msg = Message('Ticket
Progress',sender="customercare@experdsolutions.com",recipients=[customer[0]])
            msg.body = f"""
                Dear User,
                Your Ticket has been accepted by {agent[1]}
            """
            mail.send(msg)
            return redirect(url_for("loginHome"))

# close ticket
@app.route("/delete/<int:ticket_id>/<int:user_id>")
def delete(ticket_id,user_id):
    cursor = mysql.connection.cursor()
    cursor.execute("SELECT * FROM User WHERE id = %s",[user_id])
    agent = cursor.fetchone()
    cursor.execute("SELECT * FROM Tickets WHERE id=%s",[ticket_id])
    ticket = cursor.fetchone()
    if agent[4] == ticket[2]:
        cursor.execute("DELETE FROM Tickets WHERE id=%s",[ticket_id])
        mysql.connection.commit()
        cursor.execute("SELECT * FROM User WHERE id=%s",[ticket[1]])
        customer = cursor.fetchone()
        msg = Message('Ticket
Progress',sender="customercare@experdsolutions.com",recipients=[customer[1]])
        msg.body = f"""
            Dear User,
            Your Ticket has been Closed by {agent[1]}
            Thanks For using Customer Care Registry.
        """
        mail.send(msg)
        return redirect(url_for("loginHome"))

# run server
if __name__ == "__main__":
    app.run(debug=True,host='0.0.0.0',port='8080')

```

DOCKER FILE:

```

FROM python:3.8.5-alpine

WORKDIR /app
ADD . /app

RUN set -e; \
    apk add --no-cache --virtual .build-deps \
        gcc \
        libc-dev \
        linux-headers \
        mariadb-dev \
        python3-dev \
        postgresql-dev \
    ;

COPY requirements.txt /app
RUN pip install -r requirements.txt
CMD ["python","app.py"]

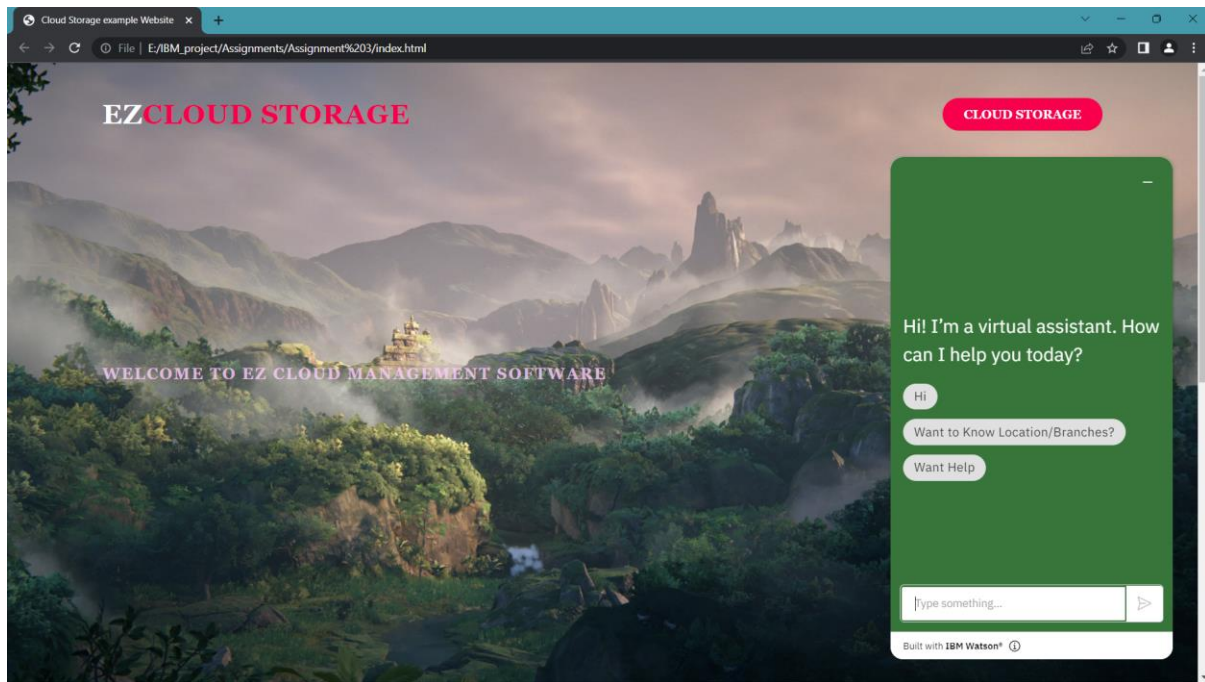
```

7.1 FEATURE 1: CHATBOT IMPLEMENTATION:

```

<script>
    window.watsonAssistantChatOptions = {
        integrationID: "66576f0c-5408-4edc-803b-d9de1f553e8b", // The ID of
this integration.
        region: "eu-gb", // The region your integration is hosted in.
        serviceInstanceID: "2607efc7-375b-465c-9e61-399a0f694519", // The ID
of your service instance.
        onLoad: function(instance) { instance.render(); }
    };
    setTimeout(function(){
        const t=document.createElement('script');
        t.src="https://web-
chat.global.assistant.watson.appdomain.cloud/versions/" +
(window.watsonAssistantChatOptions.clientVersion || 'latest') +
"/WatsonAssistantChatEntry.js";
        document.head.appendChild(t);
    });
</script>

```



7.2 FEATURE 2:

FRONTEND:

HTML:

Home.html

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

{% block body %}
<nav class="nav-items" style="top: 0;">
  <div class="logo" ><a href="/complaint">Customer Care</a></div>
  <input type="checkbox" id="click">
  <label for="click" class="menu-btn">
    <i class="fa fa-bars"></i>
  </label>
  <ul>
    <li><a href="{{url_for('logout')}}">Logout</a></li>
  </ul>
</nav>

<div class="container " style="margin-top: 100px;">
  <h2>Hi, {{ user[0] }}</h2>
  {% if user[3] == 0 %}
    <p>
      As a customer of our sevice, you can raise a ticket to
      bring you issue forward with a detailed description of
      the problem.
      Your issues will be assigned to an agent who will take
```

```

        care of it.
    </p>
    <div class="row">
        <div class="col">
            <h3>File a Complaint</h3>
            <form method="POST" >
                {% if msg %}
                    <div class="alert alert-success" role="alert">
                        {{ msg }}
                    </div>
                {% endif %}
                <input name="title" class="form-control form-
control-sm" type="text" placeholder="Ticket Header" aria-label=".form-control-
sm example" />

                <br>
                <div class="mb-3">
                    <textarea name="description"
placeholder="Problem Description..." class="form-control" id="problem-desc"
rows="3"></textarea>

                </div>
                <input type="submit" value="Raise" class="btn btn-
warning" />

            </form>
        </div>
        <div class="col">
            <h3>List of Pending Complaints</h3>
            <table class="table">
                <thead class="table-dark">
                    <tr>
                        <th>Title</th>
                        <th>Description</th>
                        <th>View</th>
                    </tr>
                </thead>
                <tbody>
                    {% for ticket in tickets %}
                        <tr>
                            <td>{{ ticket[3] }}</td>
                            <td>{{ ticket[4] }}</td>
                            <td><a
href="/ticket/{{ticket[0]}}"><button class="btn btn-
primary">View</button></a></td>

                        </tr>
                    {% endfor %}
                </tbody>
            </table>
        </div>
    </div>

```

```

        {% elif user[3] == 2 %}
            <div class="row">
                <a href="{{url_for('panel')}}"><button class="btn btn-
primary">Go To Admin Panel</button></a>
            </div>
        {% elif user[3] == 1%}
            <table class="table">
                <thead class="table-dark">
                    <tr>
                        <th>Title</th>
                        <th>Description</th>
                        <th>View</th>
                    </tr>
                </thead>
                <tbody>
                    {% for ticket in tickets %}
                        <tr>
                            <td>{{ ticket[3] }}</td>
                            <td>{{ ticket[4] }}</td>
                            <td><a href="/ticket/{{ticket[0]}}"><button
class="btn btn-primary">View</button></a></td>
                        </tr>
                    {% endfor %}
                </tbody>
            </table>
        {% endif %}
    <br>
</div>
{% endblock %}

```

login.html

```

{% extends 'base.html' %}

{% block body %}
<nav class="nav-items">
    <div class="logo"><a href="/">Customer Care</a></div>
    <input type="checkbox" id="click">
    <label for="click" class="menu-btn">
        <i class="fa fa-bars"></i>
    </label>
    <ul>
        <li><a href="/aboutus">About</a></li>
        <li><a href="/contactus">Contact</a></li>
        <li><a href="/login">Login</a></li>
        <li><a href="/register">Register</a></li>
    </ul>
</nav>

```



```

<div class="login">
  <div class="innerparent">
    <form method="POST" class="login-forms">
      <h2 style="text-align: center;">Login</h2>
      <div class="container">
        {% if msg == "Incorrect Password"%}
        <div class="alert alert-danger" role="alert">
          {{ msg }}
        </div>
        {% elif msg == "User does not exist" %}
        <div class="alert alert-primary" role="alert">
          {{ msg }}
        </div>
        {% endif %}
        <div class="mb-4">
          <label for="email-address" class="form-
label">Email</label>
          <input type="email" name="email" class="form-control"
id="email-address"
              placeholder="name@example.com">
        </div>
        <div class="mb-4">
          <label for="password" class="form-label">Password</label>
          <input type="password" name="password" class="form-
control" id="password" placeholder="Password">
        </div>
        <div class="row g-2">
          <div class="col-auto">
            <button type="submit" class="btn btn-primary mb-
3">Login</button>
          </div>
        </div>
        <p>Do not have an account ? <a href="{{
url_for('register_account') }}">Sign Up</a></p>
        </div>
      </form>
    </div>
  </div>
{% endblock %}

```

Main.html

```

{% extends 'base.html' %}

{% block body %}

```

```

    <!--main container-->
    <div class="main">
        <!--Navigation-->
        <nav class="nav-items">
            <div class="logo" ><a href="/">Customer Care Registry</a></div>
            <input type="checkbox" id="click">
            <label for="click" class="menu-btn">
            <i class="fa fa-bars"></i>
            </label>
            <ul>
                <li><a href="/aboutus">About</a></li>
                <li><a href="/contactus">Contact</a></li>
                <li><a href="/login">Login</a></li>
                <li><a href="/register">Register</a></li>
            </ul>
        </nav>
        <!--main section-->
        <main>
            <div class="sections">
                <!--main page-->
                <section id="home">
                    <div class="main-img-animation"><div class="bg-img"></div></div>
                    <div class="main-company">
                        <div class="box">
                            <h1>Customer Care Registry</h1>
                            <div class="box-row">
                                <div class="small-box"></div>
                                <div class="small-box"></div>
                                <div class="small-box"></div>
                            </div>
                            <p>Customer service shouldn't just be a department, it
should be the entire company
                            </p>
                            <button id="btn"><a href="#about">Read
More</a></button>
                        </div>
                    </div>
                </section>
                <!--about section-->
                <div class="about" id="about">
                    <div class="left-col">
                        <div class="about-box">
                            <h2>About</h2>
                            <div class="box-row">
                                <div class="small-box"></div>
                                <div class="small-box"></div>

```

```

        <div class="small-box"></div>
    </div>
    <p>We design and maintain our websites according
to the needs of our customers.While we worked for our clients, the only focus
that made us unique is the quality delivery being served at an affordable
cost. The design that we had crafted for customers stood unique and made us to
propel across the landscape.</p>
    <button id="btn"><a href="../about/">Read
More</a></button>

    </div>
</div>
<div class="right-col">
    
    </div>
</div>
<!--services page-->
<div class="services" id="services">
    <div class="sv">
        <div class="sv-
title"><h1><span>Services</span></h1></div>
        <div class="s-box-container">
            <div class="s-box">
                <div class="bar"></div>
                
                <h1>Web Development</h1>
                <div class="bar1"></div>
            </div>
            <div class="s-box">
                <div class="bar"></div>
                
                <h1>Web Hosting</h1>
                <div class="bar1"></div>
            </div>
            <div class="s-box">
                <div class="bar"></div>
                
                <h1>Digital marketing</h1>
                <div class="bar1"></div>
            </div>
        </div>
        <div class="sv-read-more"><span><a
href="../services/">Read More</a></span></div>
    </div>
</div>

```

```

    </div>
    <div class="footer-dark">
        <footer>
            <div class="container">
                <div class="row">
                    <div class="col-md-3 item">
                        <h3>Services</h3>
                        <ul>
                            <li><a href="../services/">Web
Development</a></li>
                            <li><a href="../services/">Web
Hosting</a></li>
                            <li><a href="../services/">Digital
Marketing</a></li>
                        </ul>
                    </div>
                    <div class="col-md-3 item">
                        <h3>About</h3>
                        <ul>
                            <li><a href="../about/">Company</a></li>
                            <li><a href="../team/">Team</a></li>
                        </ul>
                    </div>
                    <div class="col-md-3 item">
                        <h3>Contact</h3>
                        <ul>
                            <li><a
href="mailto:curatedsolution@gmail.com">curatedsolution@gmail.com</a></li>
                            <li><a
href="tel:6374520688">6374520688</a></li>
                        </ul>
                    </div>
                    <div class="col-md-3 item">
                        <h3>Image Credits</h3>
                        <ul>
                            <li><a href="https://www.freepik.com/"
target="_blank">Freepik</a></li>
                        </ul>
                    </div>
                    <div class="col item social">
                        <a
href="https://www.facebook.com/curatedsolution" target="_blank"><i class="fa
fa-facebook"></i></a>
                        <a href="https://twitter.com/SolutionCurated"
target="_blank"><i class="fa fa-twitter"></i></a>

```

```

        <a
href="https://www.instagram.com/curatedsolutions/" target="_blank"><i
class="fa fa-instagram"></i></a>
        <a href="https://www.linkedin.com/in/curated-
solutions-a09239216/" target="_blank"><i class="fa fa-linkedin"></i></a>
    </div>
</div>
<p class="copyright">Copyright © 2022.All rights
resevered</p>
</div>
</footer>
</div>
</main>
</div>

{% endblock %}

```

Admin-login.html

```

{% extends 'base.html' %}

{% block body %}
    <form method="POST" class="register-form">
        <div class="container">
            <h2>Administrator Sign-Up</h2>
            <div class="mb-3">
                <label for="email-address" class="form-label">Email
address</label>
                <input type="email" name="email" class="form-control"
id="email-address" placeholder="name@example.com">
            </div>
            <div class="mb-3">
                <label for="username" class="form-label">Username</label>
                <input type="text" name="username" class="form-control"
id="username" placeholder="name" />
            </div>
            <div class="row g-2">
                <div class="col-auto">
                    <label for="password" class="visually-
hidden">Password</label>
                    <input type="password" name="password" class="form-
control" id="password" placeholder="Password">
                </div>
                <div class="col-auto">
                    <label for="secret" class="visually-hidden">Secret
Key</label>
                    <input type="password" name="secret" class="form-
control" id="secret" placeholder="Secret-Key">
                </div>
            </div>
        </div>
    </form>

```

```

        </div>
        <div class="col-auto">
            <button type="submit" class="btn btn-primary mb-3">Create Account</button>
        </div>
    </div>
    <p>Already have an Account ? <a href="{{ url_for('login') }}">Login</a></p>
</div>
</form>
{% endblock %}

```

Base.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
rel="stylesheet">
    <link rel="stylesheet" href="{{ url_for('static',filename='css/main.css') }}" />
    <link rel="stylesheet" href="{{ url_for('static',filename='css/home.css') }}" />
    <link rel="stylesheet" href="{{ url_for('static',filename='css/about.css') }}" />
    <link rel="stylesheet" href="{{ url_for('static',filename='css/contact.css') }}" />

    <title>Customer-Care Registry</title>
    <script>
        window.watsonAssistantChatOptions = {
            integrationID: "66576f0c-5408-4edc-803b-d9de1f553e8b", // The ID of
this integration.
            region: "eu-gb", // The region your integration is hosted in.
            serviceInstanceID: "2607efc7-375b-465c-9e61-399a0f694519", // The ID
of your service instance.
            onLoad: function(instance) { instance.render(); }
        };
        setTimeout(function(){
            const t=document.createElement('script');
            t.src="https://web-
chat.global.assistant.watson.appdomain.cloud/versions/" +

```

```
(window.watsonAssistantChatOptions.clientVersion || 'latest') +
"/WatsonAssistantChatEntry.js";
    document.head.appendChild(t);
  });
</script>
</head>
<body>
  {% block body %}

  {% endblock %}
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta3/dist/js/bootstrap.bundle.min.js" ></script>
</body>
</html>
```

Contact.html

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

{% block body %}
<nav class="nav-items" style="top: 0;">
  <div class="logo" ><a href="/">Customer Care</a></div>
  <input type="checkbox" id="click">
  <label for="click" class="menu-btn">
    <i class="fa fa-bars"></i>
  </label>
  <ul>
    <li><a href="/aboutus">About</a></li>
    <li><a href="/contactus">Contact</a></li>
    <li><a href="/login">Login</a></li>
    <li><a href="/register">Register</a></li>
  </ul>
</nav>
<section class="contact" style="margin-top: 10em;">
  <div class="container">
    <div class="row contact-head mt-5">
      <div class="contact-title">
        <h1>Let's Start a Conversation </h1>
      </div>
      <p>Let's talk about your career and life. Send us a message
and we will be in touch within one business day</p>

    </div>
    <div class="row mt-4 mb-4">
      <div class="col-md-6">
        <h4 class="mb-4" style="color: #1f4788;"><b>Ask how we can
help you</b></h4>
        <p class="h5 mb-4">Having doubts in our services </p>
```

```

        <p class="h5 mb-4">Ask our experts and get a clear minded
solution to all your questions.</p>
        <p class="h5 mb-4">Connect With Us:</p>
        <div class="founder-social">
            <ul class="list-unstyled d-flex">
                <li class="mr-4 h3"><a
href="https://www.linkedin.com/in/experd-solutions-
467597220/" target="_blank"><i class="fa fa-linkedin"></i></a></li>
                <li class="mr-4 h3"><a
href="https://www.facebook.com/Experd-Solutions-
105826285184242" target="_blank"><i class="fa fa-facebook-
square"></i></a></li>
                <li class="mr-4 h3"><a
href="https://www.instagram.com/rajasekar.hr/" target="_blank"><i class="fa
fa-instagram"></i></a></li>
                <li class="mr-4 h3"><a
href="https://www.instagram.com/rajasekar.hr/" target="_blank"><i class="fa
fa-twitter"></i></a></li>
            </ul>
        </div>
    </div>
    <div class="col-md-6 contact-form">
        <form action="contact.php" method="post">
            <p>Please Note:All Fields are required</p>
            <input type="text" name="Name" id="name"
placeholder="Name" required>
            <input type="tel" name="mobile" id="mobile"
placeholder="Mobile No" required>
            <input type="email" name="Email" id="email"
placeholder="Email" required>
            <textarea name="message" id="message" cols="5"
rows="5" placeholder="Enter Message" required></textarea>
            <input type="submit" name="submit" value="Hear From An
Expert!!">
        </form>
    </div>
</div>
</div>
</section>
<div class="footer-dark">
    <footer>
        <div class="container">
            <div class="row">
                <div class="col-md-3 item">
                    <h3>Services</h3>
                    <ul>
                        <li><a href="../services/">Web
Development</a></li>

```



```

        <li><a href="../services/">Web
Hosting</a></li>
        <li><a href="../services/">Digital
Marketing</a></li>
    </ul>
</div>
<div class="col-md-3 item">
    <h3>About</h3>
    <ul>
        <li><a href="../about/">Company</a></li>
        <li><a href="../team/">Team</a></li>
    </ul>
</div>
<div class="col-md-3 item">
    <h3>Contact</h3>
    <ul>
        <li><a
href="mailto:curatedsolution@gmail.com">curatedsolution@gmail.com</a></li>
        <li><a
href="tel:6374520688">6374520688</a></li>
    </ul>
</div>
<div class="col-md-3 item">
    <h3>Image Credits</h3>
    <ul>
        <li><a href="https://www.freepik.com/"
target="_blank">Freepik</a></li>
    </ul>
</div>
<div class="col item social">
    <a href="https://www.facebook.com/curatedsolution"
target="_blank"><i class="fa fa-facebook"></i></a>
    <a href="https://twitter.com/SolutionCurated"
target="_blank"><i class="fa fa-twitter"></i></a>
    <a
href="https://www.instagram.com/curatedsolutions/" target="_blank"><i
class="fa fa-instagram"></i></a>
    <a href="https://www.linkedin.com/in/curated-
solutions-a09239216/" target="_blank"><i class="fa fa-linkedin"></i></a>
</div>
</div>
<p class="copyright">Copyright © 2021.All rights
resevered</p>
</div>
</footer>

```

```

    </div>
</div>
{% endblock %}

```

Panel.html

```

{% extends 'base.html' %}

{% block body %}
<nav class="navbar navbar-expand-lg navbar-dark bg-dark static-top">
  <div class="container">
    <a class="navbar-brand" href="#">
      Customer Care
    </a>
    <button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarSupportedContent" aria-
controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle
navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarSupportedContent">
      <ul class="navbar-nav ms-auto">
        <li class="nav-item">
          <a class="nav-link active" aria-current="page"
href="/">Home</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="/aboutus">About Us</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="/contactus">Contact Us</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="/login">Login</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="/register">Register</a>
        </li>
      </ul>
    </div>
  </div>
</nav>

  <div class="container">
    <div class="row">
      <div class="col">
        <h3>Promote Agents</h3>

```

```

        <div class="container">
            <form method="POST">
                <select name="admin-candidate">
                    {% for user in all_users %}
                        <option
value="{{user[4]}}">{{user[0]}}</option>
                    {% endfor %}
                </select>
                <input class="btn btn-danger btn-sm" type="submit"
value="Make Agent"/>
            </form>
        </div>
    </div>
    <div class="col">
        <h3>Assign Tasks</h3>
        <div class="container">
            <table class="table">
                <thead class="table-dark">
                    <tr>
                        <th>Title</th>
                        <th>Description</th>
                        <th>View</th>
                    </tr>
                </thead>
                <tbody>
                    {% for ticket in tickets %}
                        <tr>
                            <td>{{ ticket[3] }}</td>
                            <td>{{ ticket[4] }}</td>
                            <td><a
href="/ticket/{{ticket[0]}}"><button class="btn btn-
primary">View</button></a></td>
                        </tr>
                    {% endfor %}
                </tbody>
            </table>
        </div>
    </div>
</div>
{% endblock %}

```

Register.html

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

{% block body %}
<nav class="nav-items">
  <div class="logo"><a href="/">Customer Care</a></div>
  <input type="checkbox" id="click">
  <label for="click" class="menu-btn">
    <i class="fa fa-bars"></i>
  </label>
  <ul>
    <li><a href="/aboutus">About</a></li>
    <li><a href="/contactus">Contact</a></li>
    <li><a href="/login">Login</a></li>
    <li><a href="/register">Register</a></li>
  </ul>
</nav>
<div class="login">
  <div class="innerparent">
    <form method="POST" class="register-form">
      <h2 style="text-align: center;">Register Account</h2>
      <div class="container">
        <div class="mb-4">
          <label for="email-address" class="form-
label">Email</label>
          <input type="email" name="email" class="form-control"
id="email-address" placeholder="name@example.com">
        </div>
        <div class="mb-4">
          <label for="username" class="form-
label">Username</label>
          <input type="text" name="username" class="form-
control" id="username" placeholder="name" />
        </div>
        <div class="mb-4">
          <label for="password" class="form-
label">Password</label>
          <input type="password" name="password"
class="form-control" id="password" placeholder="Password">
        </div>
        <div class="row g-2">
          <div class="col-auto">
            <button type="submit" class="btn btn-primary mb-
3">Create Account</button>
          </div>
        </div>
        <p>Already have an Account ? <a href="{{ url_for('login')
}}">Login</a></p>
      </div>
    </form>
  </div>
</div>
{% endblock %}
```

```

        </div>
    </form>
</div>
</div>
{% endblock %}

```

CSS:

Main.css:

```

.detail-card{
    text-align: center;
    margin-top:0.5em !important;
    border-bottom: 1px black solid;
    border-radius: 5px;
    border: none;
    font-family: 'Lucida Sans', 'Lucida Sans Regular', 'Lucida Grande',
'Lucida Sans Unicode', Geneva, Verdana, sans-serif;
}

.bl{
    background-color: rgb(151, 151, 245) !important;
}

.yl{
    background-color: rgb(243, 240, 88);
}

.gr{
    background-color: rgb(95, 204, 91);
}

/* login form */
/* .login-forms{
    margin-top:10em;
} */

.login{
    display: flex;
    justify-content: center;
    align-items: center;
    width: 100%;
    height: 100vh;
}

.innerparent{
    border: 1px solid;
    padding: 20px;
    border-radius: 5px;
}

```

```
/* ticket detail */
.ticket-detail{
  margin-top:15em;
  border: 1px black solid;
  padding:1em;
  border-radius:0.2em;
}

/* .register-form{
  margin-top:10em;
} */

/* body{
  background-image: url('https://external-
content.duckduckgo.com/iu/?u=http%3A%2F%2Fwww.pixelstalk.net%2Fwp-
content%2Fuploads%2F2016%2F04%2FPhotos-download-abstract-minimalist-wallpaper-
HD.jpg&f=1&nofb=1');
  background-repeat: no-repeat;
  background-size: cover;
} */
```

8. TESTING

8.1 USER ACCEPTANCE TESTING:

User Acceptance Testing (UAT) is a type of testing performed by the end user or the client to verify/accept the software system before moving the software application to the production environment. UAT is done in the final phase of testing after functional, integration and system testing is done.

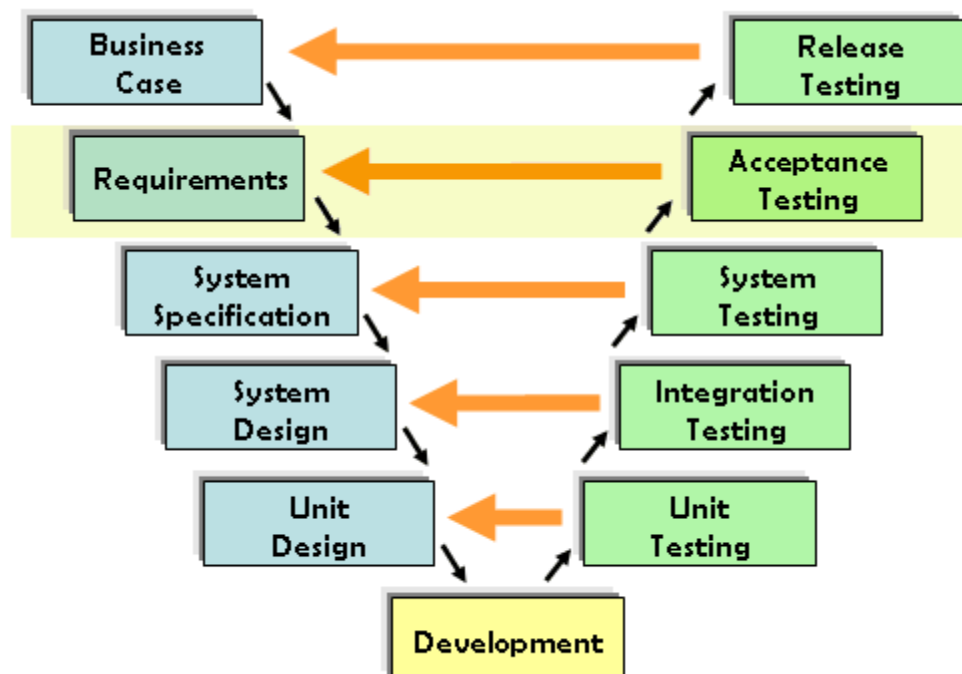
The main Purpose of UAT is to validate end to end business flow. It does not focus on cosmetic errors, spelling mistakes or system testing. User Acceptance Testing is carried out in a separate testing environment with production-like data setup. It is kind of black box testing where two or more end-users will be involved.

UAT is performed by –

Client

End users

In VModel, User acceptance testing corresponds to the requirement phase of the Software Development life cycle(SDLC).

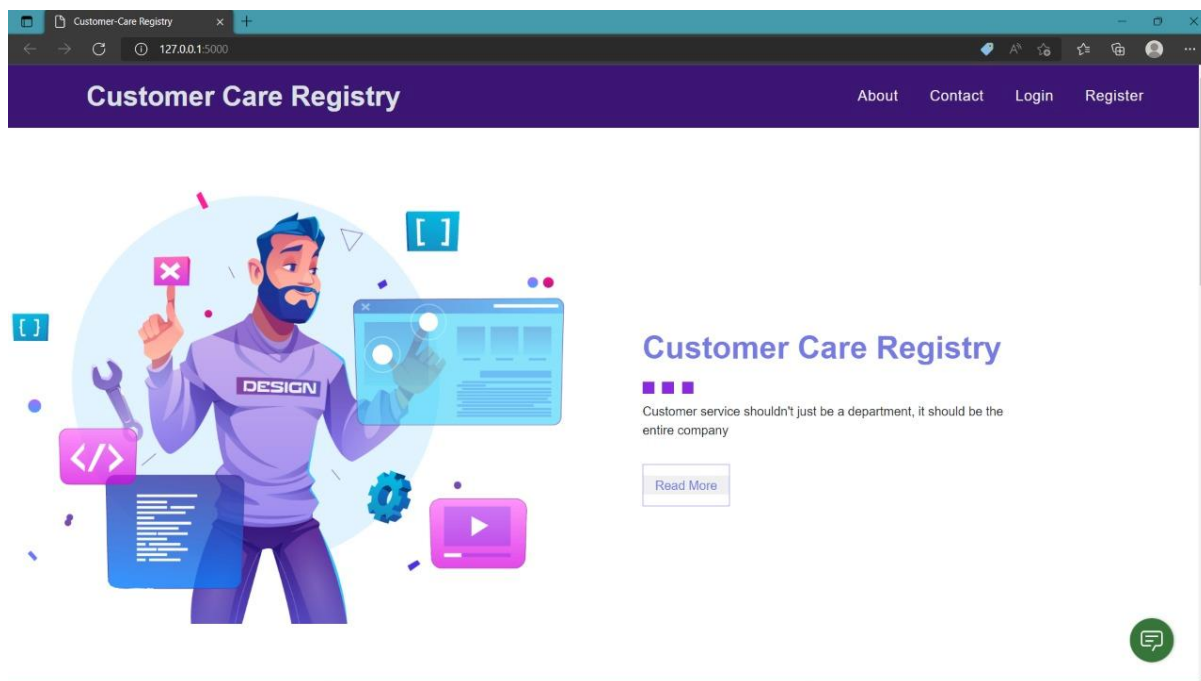


9. RESULTS

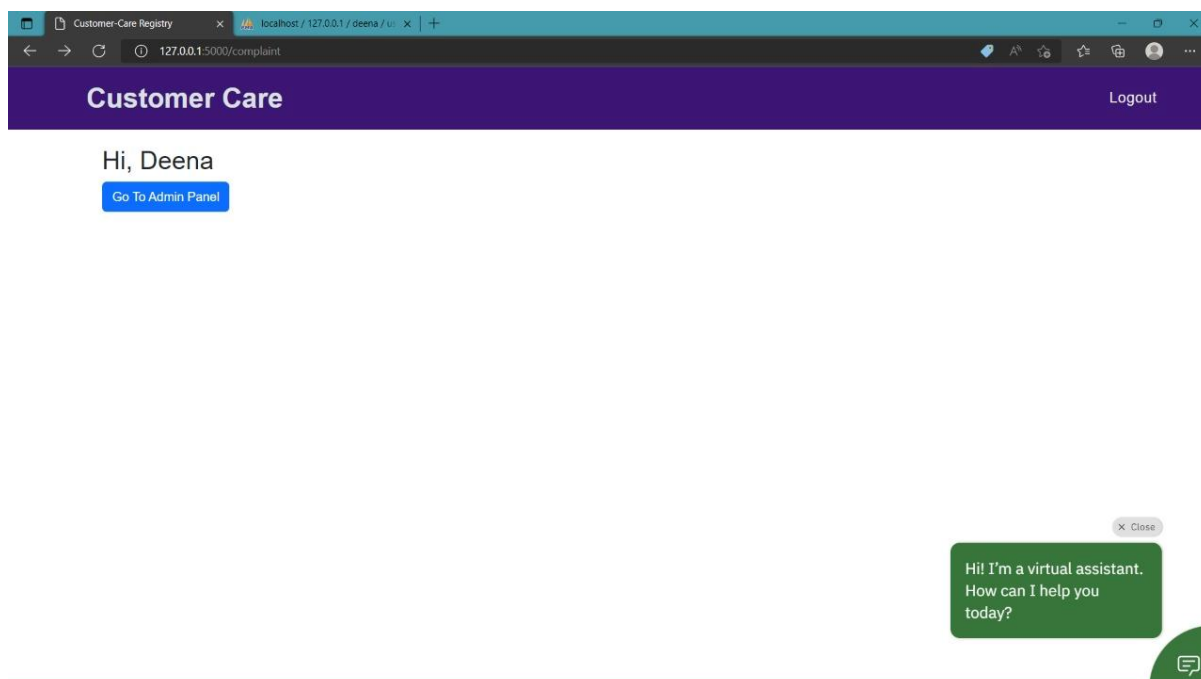
9.1 PERFORMANCE METRICS:

OUTPUTS:

HOMEPAGE:



ADMIN PAGE:



CUSTOMER PAGE:

Customer Care [Logout](#)

Hi, Deena

As a customer of our service, you can raise a ticket to bring your issue forward with a detailed description of the problem. Your issues will be assigned to an agent who will take care of it.

File a Complaint

Ticket Header

Problem Description...

[Raise](#)

List of Pending Complaints

| Title | Description | View |
|-------|-------------|------|
|-------|-------------|------|

Hi! I'm a virtual assistant. How can I help you today?

AGENT PAGE:

Customer Care [Logout](#)

Hi, Dhivesh

| Title | Description | View |
|-------|-------------|------|
|-------|-------------|------|

Hi! I'm a virtual assistant. How can I help you today?

10. ADVANTAGES & DISADVANTAGES

PROS:

- The system proposes a robust, scalable and extensible architecture with a feasible tech stack
- An automated customer service with the use of cloud.
- Feedback loops are used so that they allow the provider to capture feedback and provide continuous customer care.
- Implementation of virtual chat bot along with the website helps in solving simple problems without interference of human support

CONS:

- Not always easy for SaaS providers to know what customer are experiencing and might have discrepancies in updating the software frequently
- The server room creates the room functionality where the customer needs to be persistent over time
- The chat bot is limited to only specific problems and responds to that specification and might be unreliable at times
- Updating software is tricky

11. CONCLUSION

Looking ahead further development of this project will bring considerable changes in the customer care domain automating answering queries of customer.

Efficient improvement in relationship between customers and service providers

In conclusion, customer care, involves the use of basic ethics and any company who wants to have success and grow, needs to remember, that in order to do so, it must begin with establishing a code of ethics in regards to how each employee is to handle the dealing with customers.

Customers are at the heart of the company and its growth or decline. Customer care involves, the treatment, care, loyalty, trust the employee should extend to the consumer, as well in life.

This concept can be applied to so much more than just customer care. People need to treat others with respect and kindness, people should try to take others into consideration when making any decision.

If more people were to practice this policy, chances are the world would be a better, more understanding place for all to exist.

Overall this is a web enabled project and grants exposure to various web development technologies and framework used globally

12. FUTURE SCOPE

This web enabled project can be further developed to further enhance it's capabilities in serving the customer.

For instance the chat bot implemented can be further improved upon to accept more user queries and provide solutions.

The technologies used in the development can be further updated and replaced for efficient performance of the application

Further development of the application can help in enhancing and providing customer support in the future seamlessly.