CUSTOMER CARE REGISTRY

(TEAM ID : PNT2022TMID34848)

PROJECT REPORT

Submitted by

KALAIVANI N. (962819104052)

ATHIRA R K. (962819104022

DHARSHNI C J. (962819104029)

JENUSHA R J. (962819104048)

in partial fulfillment for the award of degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

UNIVERSITY COLLEGE OF ENGINEERING ,NAGERCOIL

INDEX

CHAPTER 1
INTRODUCTION

Customer care is more than just providing great customer service. It’s a proactive approach to providing information, tools, and services to customers at each point they interact with a brand. 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.

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

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.

CHAPTER 2

LITERATURE SURVEY

1. In the era of globalization electronic marketing is a great revolution. Over the last decade maximum business organizations are running with technological change. Online shopping or marketing is the use of technology (i.e., computer) for better marketing performance. And retailers are devising strategies to meet the demand of online shoppers; they are busy in studying consumer behavior in the field of online shopping, to see the consumer attitudes towards online shopping. Therefore we have also decided to study consumer's attitudes towards online shopping and specifically studying the factors influencing consumers to shop online.

2. Fault management plays a major role in Telecommunication industry. An effective and efficient response to customer complaints is an essential index of organization's performance. The presented model for the CEMP has the ability to minimize customers' dissatisfaction and on the other hand it can encourage customers to participate in controlling the provided quality of the services. The customer may feel dissatisfied with the service if he or she receives a delayed response. Customers do not know where to fill the complaint, Current complaint handling in the organization still have these problems. Therefore, CEMP was proposed and implemented to solve the customer faults. CEMP was consisted both a mobile application and a web application linking the customer to technician in the field through a management portal. Proposed system has the functionalities of fault/technician tracking, maintain user profile, nearest technician acknowledgement and customer feedback which are beneficial to both customer and the company.

3. Customers are the essential factor in the organization. The business has to support the customers' preferences and demands for creating the customer loyalty, which make the customer still purchases with the particular company. The customer may feel dissatisfied with the service when he or she receives the delay of services and they do not know the channel for filing the complaint, and also the current complaint handling in the organizations still has the problems. Therefore, we, developers of this project implemented the Smart Complaint Management System (SCMS) consisting of the mobile application, chatbot and web application, for solving the customer's dissatisfaction issue. Furthermore, the SCMS has the service for classifying the complaint, then automatically direct to the responsible department, and the service for finding the similar complaint to avoid submitting the duplicate complaint. The test result shows that this system is able to reduce the time and procedures for complaint handling, increase the channel for filing the complaint, and increase the channel for progress reporting and tracking the status of the complaint.

4. In this research, we address the following questions that are becoming increasingly important to managers in automotive industries: is there a relationship between customer service and product quality with customer satisfaction and loyalty in the context of the Indian automotive industry? If yes, how is the relationship between these four variables? The automotive industry in India is one of the largest in the world and one of the fast growing globally. Customer satisfaction and loyalty are the most important factors that affect the automotive industry. On the other hand, Customer service can be considered as an innate element of industrial products. Customer service quality, product quality, customer satisfaction and loyalty can be measured at different stages, for example, at the beginning of the purchase, and one or two years after purchase. The population of the study is all of the Tata Indica car owners in Pune. Hypotheses of the study will be analyzed using regression and ANOVA. Results of the study show that there are high positive correlation between the constructs of costumer service and product quality with costumer satisfaction and loyalty.

5. This study endeavours to understand customer satisfaction in online shopping while investigating the major reasons that motivated customers' decision-making processes as well as inhibitions of online shopping. The Kotler and Killers (2009) Five Stage Buying Process Model was chosen as the basis of framework of this study to explain customer satisfaction through their motivations to buy products online. The existing literature was reviewed to discover reasons that would influence customers positively or negatively towards shopping online. Surveys were conducted by distributing questionnaires in the Wrexham area (North Wales) to gather data for this research. SPSS software package was used to present research data graphically and to test research hypothesis. From the findings, it was discovered that respondents use internet to purchase products through online because they believe it is convenience to them and the term convenient includes elements such as time saving, information availability, opening time, ease of use, websites navigation, less shopping stress, less expensive and shopping fun. In contrast, along with respondents' mind-sets, online payment security, personal privacy and trust, unclear warranties and returns policies and lack of personal customer service are the foremost barriers of online shopping. Furthermore, the result of hypotheses established that even though online shopping is convenient to all consumers, online payment system and privacy or security anxieties have significant impact on online shopping. Finally, some recommendations have been offered for online retailers to take initiatives for making online shopping more admired and trustworthy.

6. In the era of globalization electronic marketing is a great revolution. Over the last decade maximum business organizations are running with technological change. Online shopping or marketing is the use of technology (i.e., computer) for better marketing performance. And retailers are devising strategies to meet the demand of online shoppers; they are busy in studying consumer behavior in the field of online shopping, to see the consumer attitudes towards online shopping. Therefore we have also decided to study consumer's attitudes towards online shopping and specifically studying the factors influencing consumers to shop online.

2.2 References

1. A Study on Customer Satisfaction Towards Online ShoppingMrs. T. Shenbaga Vadivu*Assistant Professor, Department of Business Administration, Nehru Arts and Science College, Thirumalayampalayam, Coimbatore

2. Customer Experience Management PlatformM.P.M. Dias Department of Electronics Wayamba University of Sri Lanka Kuliyapitiya, Sri Lanka

3. Smart Complaint Management System Siripen Pongpaichet, Pattamaporn Kormpho, Panida Liawsomboon, Narut Phongoen Faculty of Information and Communication Technology Mahidol University Nakhon Pathom, Thailand

4. Study of the effects of customer service and product quality on customer satisfaction and loyalty Asghar Afshar Jahanshahi (Corresponding Author) PhD Scholar in Business Administration Department of Commerce & Research Center University of Pune, India

5. Customer Satisfaction in Online ShoppingRashed Al Karim (Assistant Professor, Business Administration, East Delta University, Bangladesh)

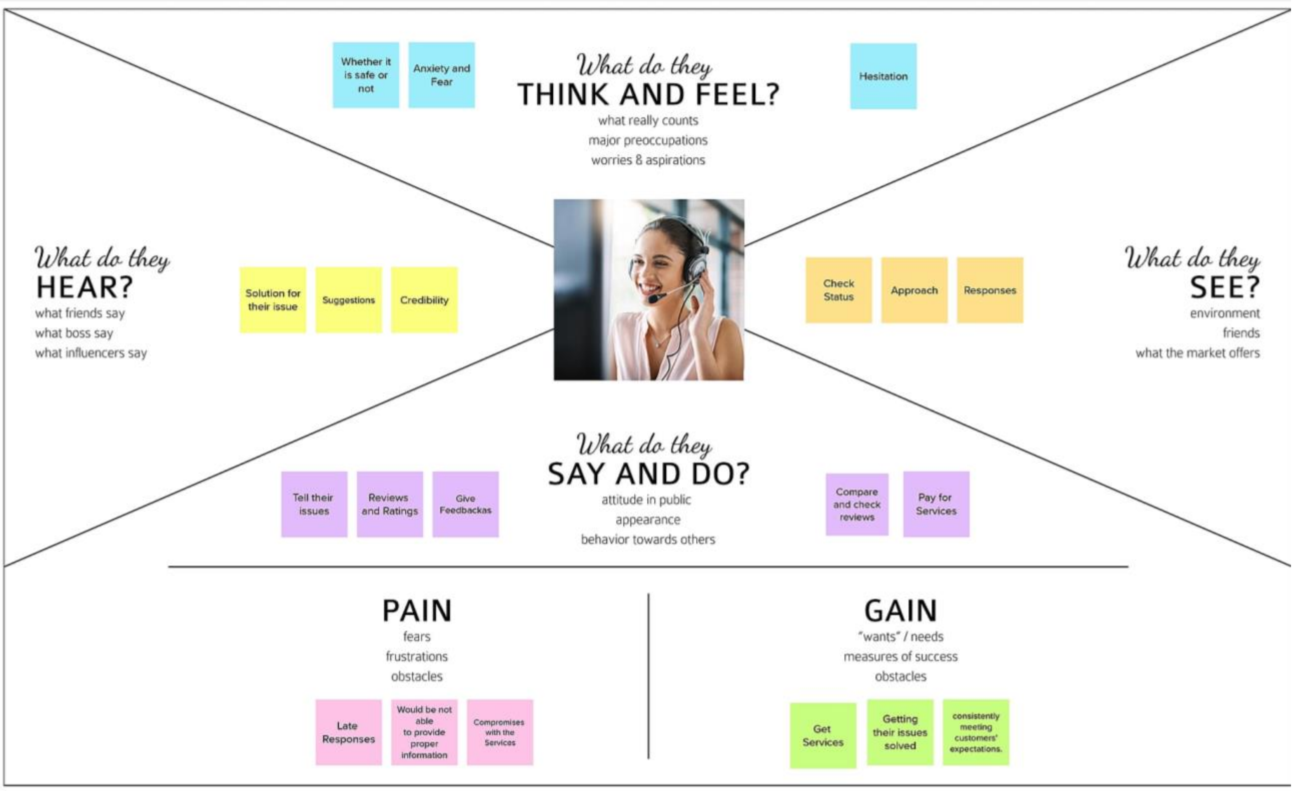
6. A Study on Customer Satisfaction Towards Online Shopping Mrs. T. Shenbaga Vadivu*Assistant Professor, Department of Business Administration, Nehru Arts and Science college, Thirumalayampalayam, Coimbatore

2.3 Problem Statement Definition

The project is designed in a way to help businesses manage to keep a digital record of customer interactions and build a connection with the customers. Admin, User & Agent, where Admin will have the main control over the system. Admin will be responsible for maintaining the system and also, keeping a check on customer information, complaints, inquiry, products details, etc

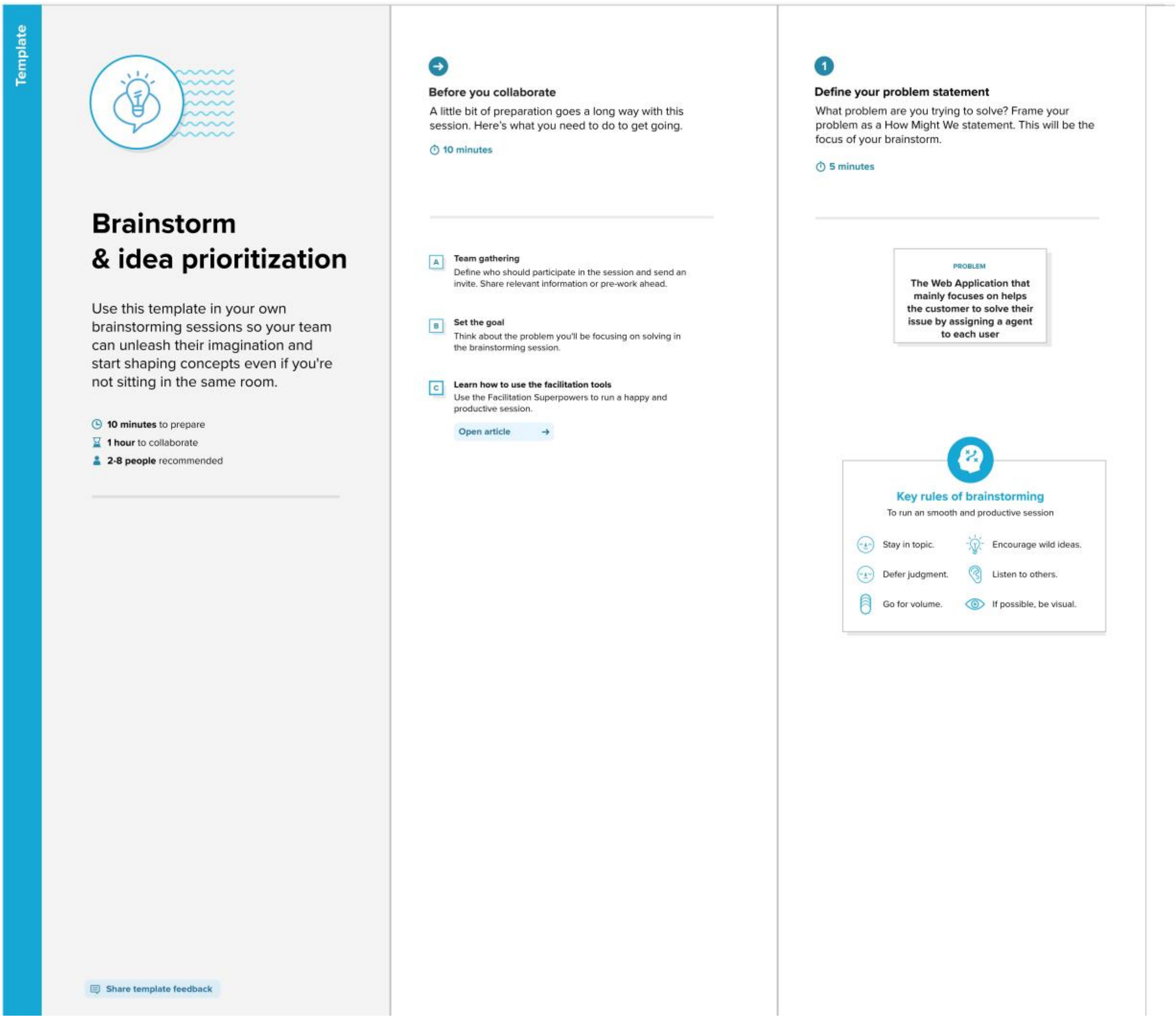
CHAPTER 3
IDEATION &PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming

Step-1: Team Gathering, Collaboration and Select the Problem Statement



Template

Brainstorm
& idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare

1 hour to collaborate

2-8 people recommended

Share template feedback

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

1 **Define your problem statement**

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

PROBLEM

The Web Application that mainly focuses on helps the customer to solve their issue by assigning a agent to each user

Key rules of brainstorming

To run an smooth and productive session

Stay in topic.

Encourage wild ideas.

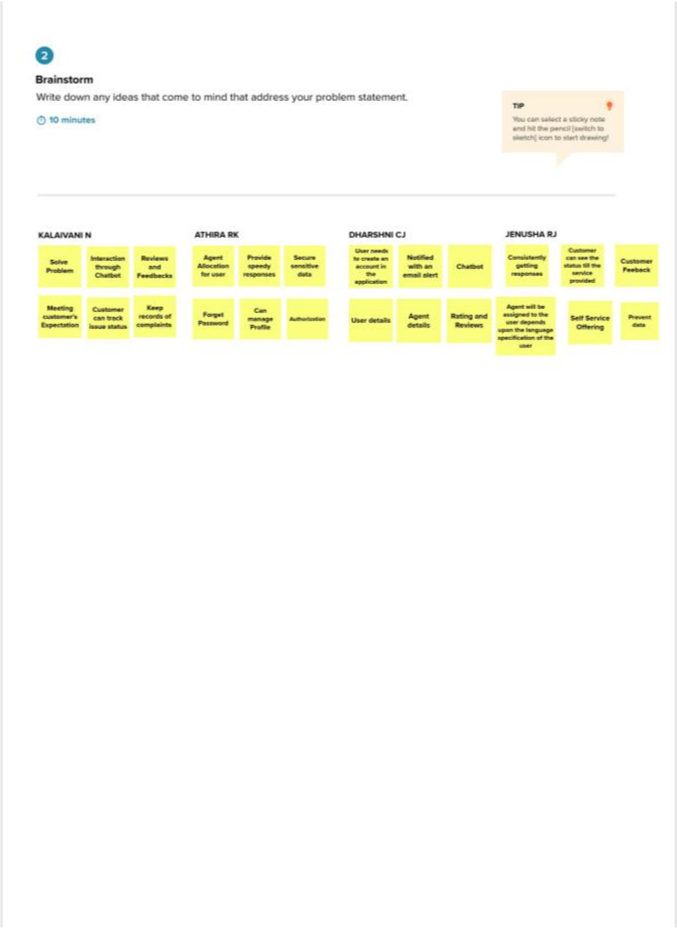
Defer judgment.

Listen to others.

Go for volume.

If possible, be visual.

Step-2: Brainstorm, Idea Listing and Grouping



2

Brainstorm

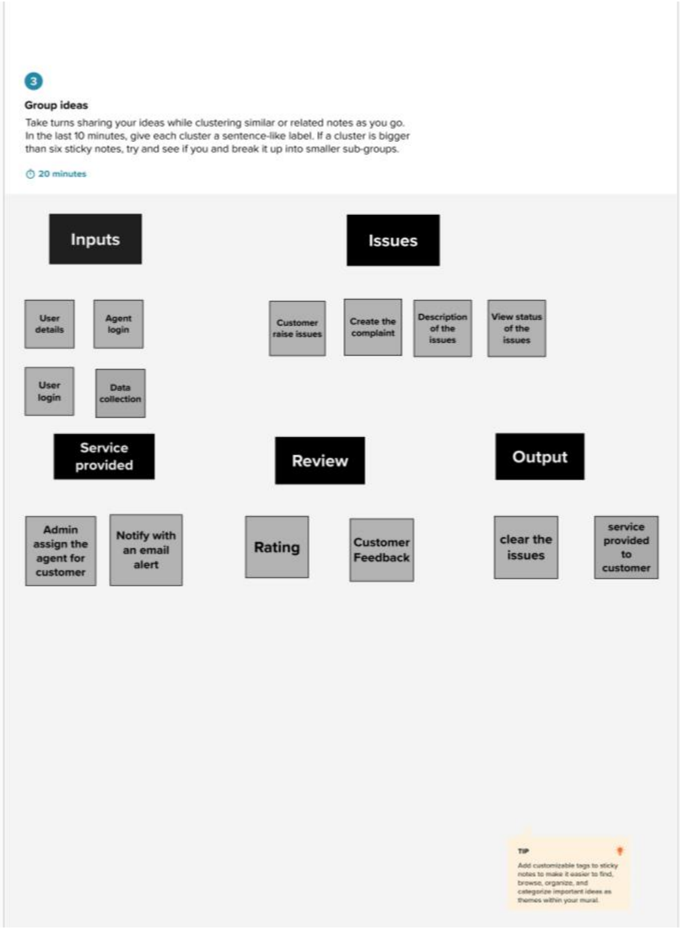
Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

KALAIVANI N
Solve Problem
Interaction through Chatbot
Reviews and Feedbacks
Meeting customer's Expectation
Customer can track issue status
Keep records of complaints
ATHIRA RK
Agent Allocation for user
Provide speedy responses
Secure sensitive data
Forget Password
Can manage Profile
Authorization
DHARSHNI CJ
User needs to create an account in the application
Notified with an email alert
Chatbot
User details
Agent details
Rating and Reviews
JENUSHA RJ
Consistently getting responses
Customer can see the status till the service provided
Customer Feeback
Agent will be assigned to the user depends upon the language specification of the user
Self Service Offering
Prevent data



3

Group ideas

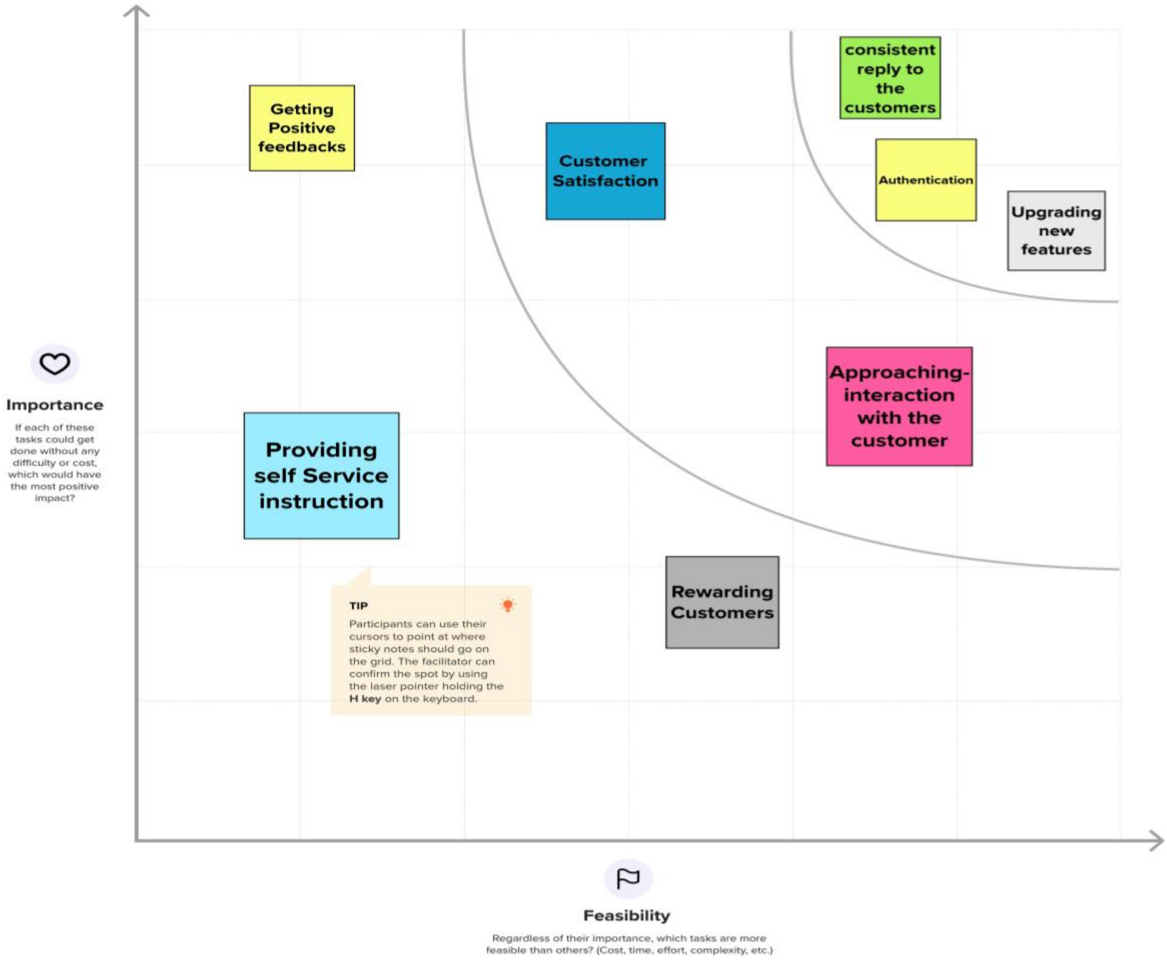
Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

Step-3: Idea Prioritization


Getting Positive feedbacks
Customer Satisfaction
consistent reply to the customers
Authentication
Upgrading new features
Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?
Approaching-interaction with the customer
Providing self Service instruction
Rewarding Customers
Feasibility
Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

TIP

Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H key** on the keyboard.

3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to besolved)	The project is designed in a way to help businesses manage to keep a digital record of customer interactions and build a connection with the customers. Admin, User & Agent, where Admin will have the main control over the system. Admin will be responsible for maintaining the system and also, keeping a check on customer information, complaints, inquiry, Products details, etc.
2.	Idea / Solution description	In order to solve the issues of the customer, we proposed a user-friendly, flexible and effective system. The user can simply register to the system. In order to file a complaint, a user has to fill up details such as their contact details, complaint domain. Admin has an important role here; the admin is the one who can update the status of each complaint. The user can also view the history of their complaints where they can see the update of the status of their complaints
3.	Novelty / Uniqueness	One of the major contributions of this project is the selection of different new features, which are capable enough to solve the customer's issues consistently.
4.	Social Impact/ Customer Satisfaction	Customers get the insights they need to make and informed purchase. Customer satisfaction can increase and customer loyalty can improve. Customer service agents spend less time on routine tasks and answering commonly asked questions, enabling agents to do more meaningful tasks.
5.	Business Model (Revenue Model)	A free web application system that can be used by many any person, E-commerce website or other websites in order to satisfy the customers. It will be used by most of the students and alsohave high market value. It's quick and trustworthy.
6.	Scalability of the Solution	As this is a web application and uses cloud storage, additional features can be added often. Any number of people can register for the system.

CHAPTER 4
REQUIREMENT ANALYSIS

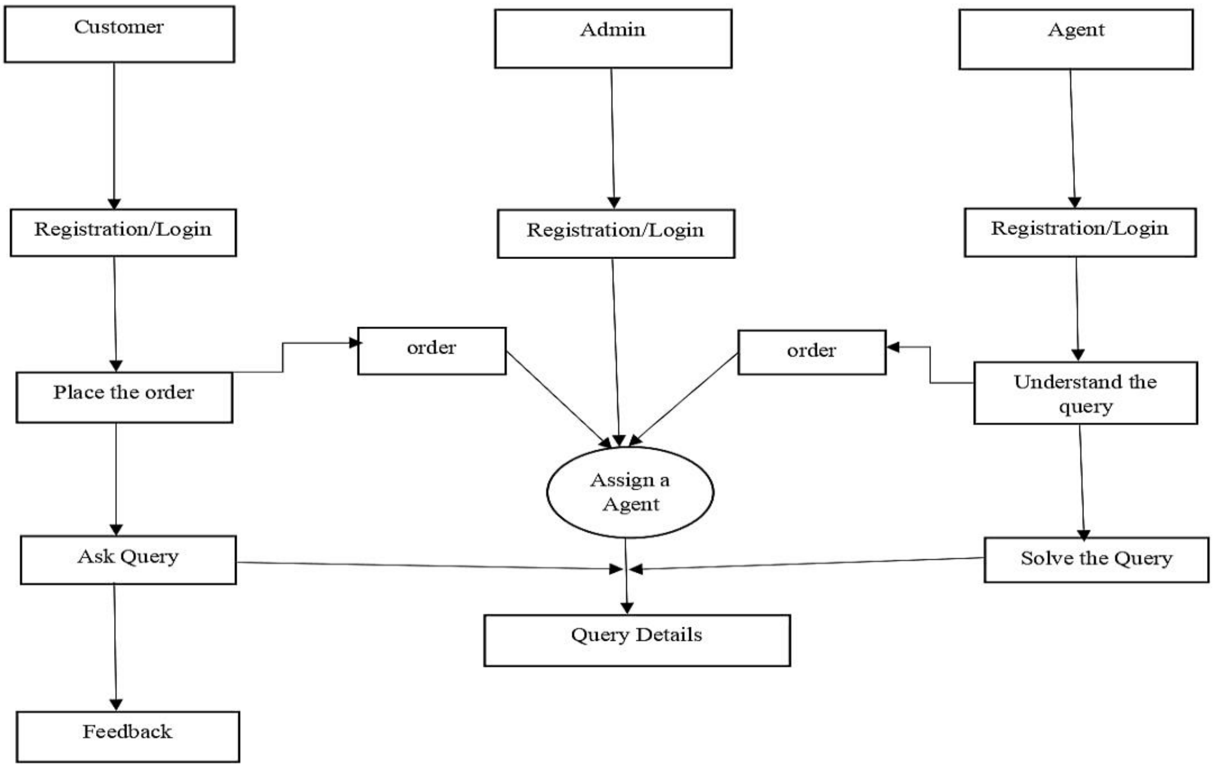
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through Google
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Login	Login via Google Login with Email id and Password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

4.2 Non-functional Requirements

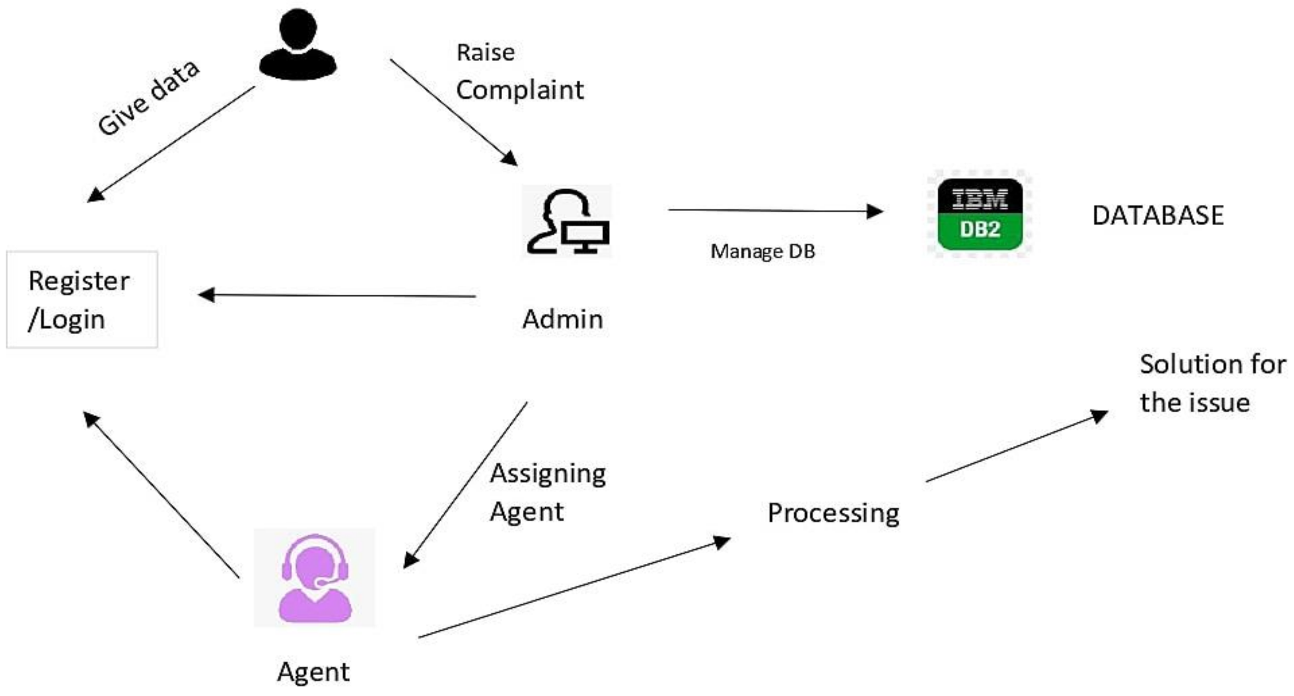
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	This system is really used to clear the quires of the user. To solve the quires or problems admin assign a agent to the user. Agent will give the solution for the quires .
NFR-2	Security	When dealing with your customers, you should ask for only the information you need to serve them. Track of login authentication .
NFR-3	Reliability	The performance of the system would be really good. Providing effective support to the user. Problems will be recognized and solved as early as possible.
	Perf	The performance mostly depends on fully problem solving skills and
NFR-5	Availability	System will provide consistent availability for customer support. The availability of the solution is effective and it should be helpful in the great way to the user for clear there problems. It provide 24/7 service.
NFR-6	Scalability	Customers get the insights they need to make an informed purchase. Customer satisfaction can increase and customer loyalty can improve.

CHAPTER 5

5.1 Data flow diagrams



5.2 Solution Architecture



3.3 Technical Architecture

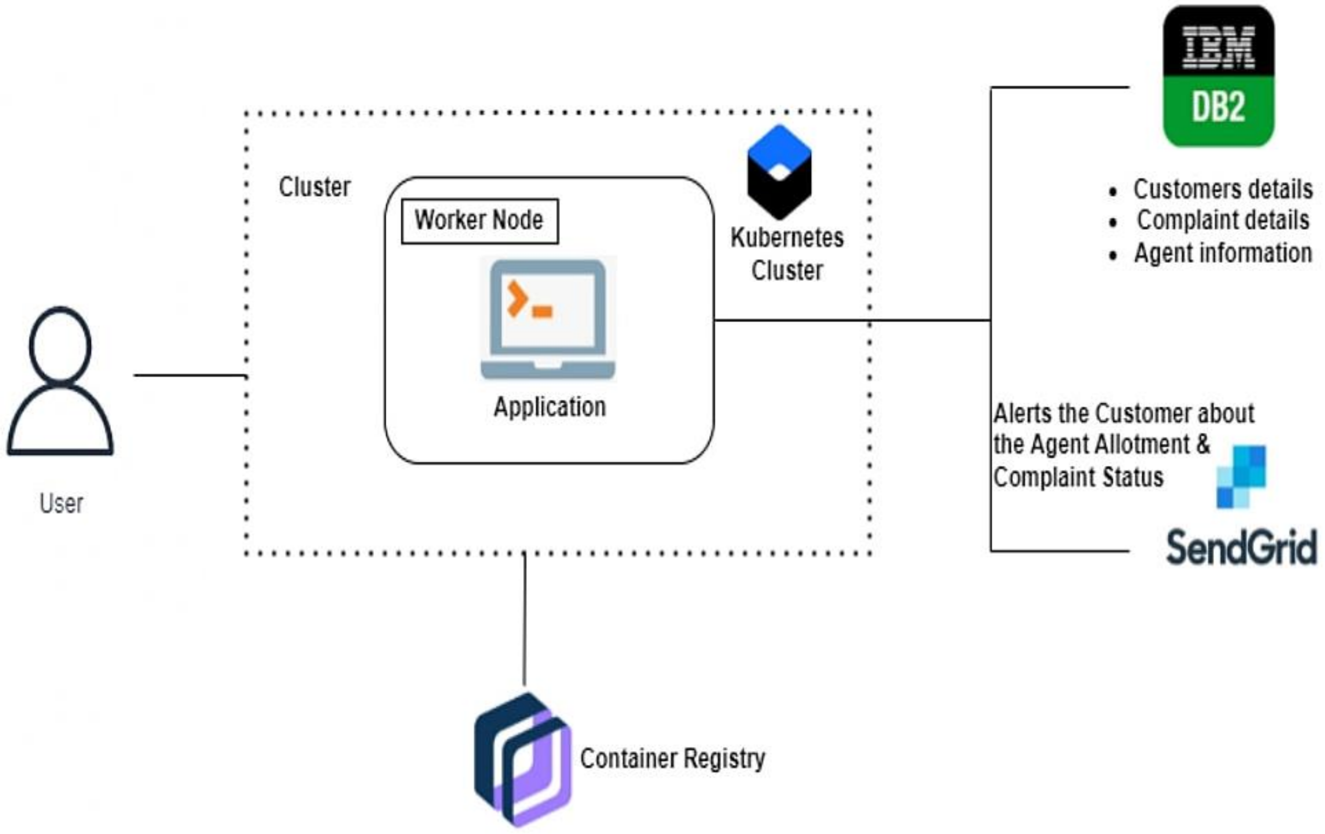


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript / Angular JS / React JS etc.
2.	Application Logic-1	Logic for a process in the application	Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Database	User data are stored in the MySQL database	MySQL, etc.
6.	Cloud Database	Database service on Cloud	IBM DB2, IBM Cloudant etc.
7.	File Storage	Used to store the data of the user	IBM Block Storage or Other Storage Service or Filesystem
8.	Cloud Deployment	Application Deployment on Local System/Cloud Local Server Configuration: Cloud Server Configuration:	Local, Cloud Foundry, Kubernetes, etc

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology

1.	Open-Source Frameworks	Application development, data pre-processing.	Visual studio code, Anaconda navigator, TensorFlow
2.	Security Implementations	It identifies the gesture only when thehand isin front of the camera.	OpenCV
3.	Scalable Architecture	It can be used in anyenvironment and is able toidentify the gesture	OpenCV
4.	Availability	It is used to reduce the possibility of spreading infections	AI
5.	Performance	Rapid response to the gesture.	CNN

User Stories

User Type	Functional Requirment (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer	Registration	USN-1	As a user, I can register for the web application by entering my email, password, and confirming my password.	I can access my account.	High	Sprint 1
	login	USN-2	As a user, ,I can login for the web application by entering my email, password.	I can access my account .	High	Sprint 1
	Dashboard	USN-3	As a user, I can see all the orders raised by me.	I can get all the information needed .	Medium	Sprint 2
	Order creation	USN-4	As a user, I can place my order with the detailed description of my query.	I can ask my query	High	Sprint 2
	Address Column	USN-5	As a user, 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 user, I can reset my password by this option incase I forgot my old password.	I can get access to my account again	High	Sprint 3

	Order details	USN-7	As a Customer ,I can see the current stats of order.	I can get a better understanding	High	Sprint 4
	Login	USN-8	As an agent I can login to the application by entering Correct email and password .	I can access my account / dashboard.	High	Sprint 4
	Dashboard	USN-9	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 4
	Address column		As an agent, I get to have conversations with the customer and clear his/er doubts	I can clarify the issues	High	
	Forgot password		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	
	Login		As a admin, I can login to the application by entering Correct email and password	I can access my account/dashboard	Medium	
	Dashboard		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.	Medium	
	Assignment agent		As an admin I can assign an agent for each order created by the customer.	Enable agent to clarify the queries.	High	
	Forgot password		As an admin I can reset my password by this option in case I	I get access to my account	High	

			forgot my old password.			

CHAPTER 6

PROJECT PLANNING AND SCHEDULING

6.1 Sprint Planning and Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story/ Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirmingmypassword.	5	High	TM – 1 TM – 4
Sprint-1	Login	USN-2	As a user, I will receiveconfirmation email onceIhave registered for the application	5	High	TM – 2 TM – 3
Sprint-2	Dashboard	USN-3	As a user, I can register for the application through facebook	10	Low	TM – 1 TM – 2
Sprint-1	Detailsabout	USN-4	As a user, I can register for the application through Gmail	5	Medium	TM – 3 TM – 1
Sprint-1	Login and repeated usage	USN-5	As a user, I canlog into the application by entering email& password	5	High	TM – 2 TM – 4
Sprint - 2	web page details	USN-6	As a user I mustcapture images of handand upload it into the web portal.	10	High	TM – 1 TM – 3
Sprint - 3	Upload the image in the web application	USN-7	As a user I must receivea correct handgesture	20	High	TM – 1 TM – 2

			as output			
Sprint - 4	Provide efficient customer support	USN-8	As a user,I need to getsupport from developers in case of queries andfailure of service provided	10	Medium	TM – 3 TM –4
Sprint - 4	Overview the entire process. Take all the responsibility andact bridge between users and developers	USN-9	We need to satisfy the customer needs in an efficient way and makesure any sort of errorsare fixed	10	High	TM – 2 TM – 1

6.2 Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story PointsCompleted (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	14 Oct 2022	20 Oct 2022	20	21 Oct 2022
Sprint-2	20	6 Days	22 Oct 2022	28 Oct 2022	20	29 Oct 2022
Sprint-3	20	6 Days	30 Oct 2022	05 Nov 2022	20	06 Nov 2022
Sprint-4	20	6 Days	07 Nov 2022	13 Nov 2022	20	14 Nov 2022

Velocity:

Imagine we have a 10-daysprint duration, and the velocity of the team is 20 (points per sprint). Let’scalculate the team’saverage velocity (AV) per iterationunit (story pointsper day)

$$\textbf{AV = sprint duration/ velocity = 20/6 = 3.33}$$

Burn down Chart:

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies suchas Scrum.However, burn down chartscan be applied to any project containing measurable progress over time.

CHAPTER 7
CODING AND SOLUTIONING

Style.css

```
.heading {
    text-align: center;
    display: inline-block;
}

body{
    /* background-image: url('{{ url_for(\'static\', filename=\'img/bg.jpg\')}}'); */
    background-size: cover;
}
```

Script.js

```
function form_create() {
    var CONTINUE = confirm("DOING THIS WILL DELETE YOUR PREVIOUS QUERY.Do you want to continue?")
    if (CONTINUE === true) {
        var elem = document.getElementById('add_elem')
        elem.innerHTML = `<div class="container pd-20"><form action="/success" method="post">
        <div class="form-group"><label class="col-md-5">Enter your query : </label>
        <input name="query" class=\"form-control col-md-5\"type="textarea" name="query" id="query" autocomplete="off">
        <input type="submit" value="Submit"></div></form></div>`
    }

}
```

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">
    <title>
        {%block title%}

        {%endblock%}
    </title>
    <!-- Bootstrap -->
    <link rel="stylesheet" href="{{url_for('static',filename='css/styles.css')}}">
    <link rel="stylesheet" href="{{ url_for('static',filename='css/bootstrap.min.css') }}">
    <script src="{{ url_for('static',filename='js/script.js') }}"></script>
</head>

<body style="background-image: url({{ url_for('static', filename='img/bg.jpg') }}) ">
    <nav class="navbar navbar-primary" style="font-family: cursive;">
```

```
        <div class="container-fluid">
            <!-- Brand and toggle get grouped for better mobile display -->
            <div class="navbar-header" style="margin-top: 20px;">
                <button type="button" class="navbar-toggle collapsed" data-toggle="collapse"
                    data-target="#bs-example-navbar-collapse-1" aria-expanded="false">
                    <span class="sr-only">Toggle navigation</span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>
                <a class="navbar-brand" href="/">HOME</a>
            </div>

            <ul class="nav navbar-nav navbar-right" style="margin-top: 20px; font-family: cursive;">
                <li><a href="{{url_for('login')}}">LOGIN</a></li>
                <li><a href="{{url_for('register')}}">REGISTER</a></li>

            </ul>
        </div>

    </nav>
    {%block content%}

    {%endblock%}
</body>

</html>
```

Register.html

```
{% extends 'base.html' %}
{%block title%}
Registration page
{%endblock%}

{%block content%}
<div class="container pd-20">
    <h1 style="margin-top:100px; font-family: cursive;">
        Welcome to Registration page
    </h1>
    <h2>{{ msg }}</h2>
    <form name="register" action="{{url_for('register')}}" method="POST" onsubmit="submitted()"
style="font-family: cursive;">
        <div class="form-group">
            <label class="col-md-5" style="font-family: cursive;">User Name</label><input class="form-
control col-md-5" name="name" type="text" autocomplete="off"
                required>
            <label class="col-md-5" style="font-family: cursive;">Email</label><input class="form-control col-
md-5" name="email" type="text" autocomplete="off" required>
            <label class="col-md-5" style="font-family: cursive;">New Password</label><input class="form-
control col-md-5" id="p1" name="password1"
                type="password" required>
            <label class="col-md-5" style="font-family: cursive;">confirm Password</label><input
class="form-control col-md-5" id="p2"
```

```
                name="password" type="password" required>
            <input name="submit" class="btn btn-success" type="submit" value="submit" style="margin-top: 19px;">
            <input class="btn btn-danger" type="reset" value="clear" style="margin-top: 19px;">
        </div>
        <div>
            <a class="btn btn-primary" href="{{ url_for('login') }}">Login instead?</a>
        </div>
    </form>
</div>
{%endblock%}
```

login.html

```
{%extends 'base.html'%}
{%block title%}
Login Page
{%endblock%}
{%block content%}
<div class="container pd-20" style="font-family: cursive;">
    <h1 style="margin-top:100px; font-family: cursive;">
        Welcome to Login page
    </h1>
    <h2>{{ msg }}</h2>
    <form action="{{url_for('login')}}" method="POST">
        <div class="form-group" style="font-family: cursive;">
        <label class="col-md-5">User Name</label><input class="form-control col-md-10" name="name" type="text" autocomplete="off">
        <label class="col-md-5">Password</label><input class="form-control col-md-10" name="password" type="password" autocomplete="off">
        <br><br><br>
        <input class="btn btn-success" type="submit" value="submit" style="margin-top: 19px;">
            <input class="btn btn-danger" type="reset" value="clear" style="margin-top: 19px;">
        </div>
        <div>
            <a href="{{url_for('home')}}" class="btn btn-primary">< Back to Home</a>

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

Done.html

```
{% extends 'base.html'%}
{% block title %}
Done
{%endblock%}
{%block content%}
{% if session['role'] == 'agent' %}
<center>
    <h1>{{msg}}</h1>
</center>
```

```
{% endif %}
{% if session['role'] == 'admin' %}
<center>
   <h1>{{msg}}</h1>
</center>
{% endif %}

{% endblock %}
```

index.html

```
{%extends 'base.html'%}
{%block title%}
HOME | Customer Care Registry
{%endblock%}
{%block content%}
<center>
   <div class="container">
      <h1 style="margin-top: 150px; font-family: cursive;" >Welcome to Customer Care Registry</h1>
      <br><br>
      <h2 style="font-family: cursive;">Good Service is Good Business</h2>
   </div>
</center>


{%endblock%}
```

dashboard.html

```
{%extends 'base.html'%}
{%block title%}
Dashboard
{%endblock%}
{%block content%}
<center>
   <div class="container container-pd-5">
      <h1>Welcome to Dashboard</h1>
      <div>You have successfully logged in as {{ session['role'] }}</div>
      <hr>
   </div>
</center>
{% if session['role'] =='user' %}
<center>
   {% set customer = session['customer'] %}
   <h1 class="heading">Hello {{ customer['USERNAME'] }}</h1>
   {% if customer['QUERY'] =='none' %}
   <h1>You Haven't made any Queries</h1>
   {% else %}
   <div>
      <h1>Your Previous Query :</h1>
      <div>{{ customer['QUERY'] }} </div>
      <h1>Your Ticket:</h1>
      <div>{{ customer['TICKET'] }}</div>
      {% if customer['REVIEW_STATUS'] == 1 %}
```

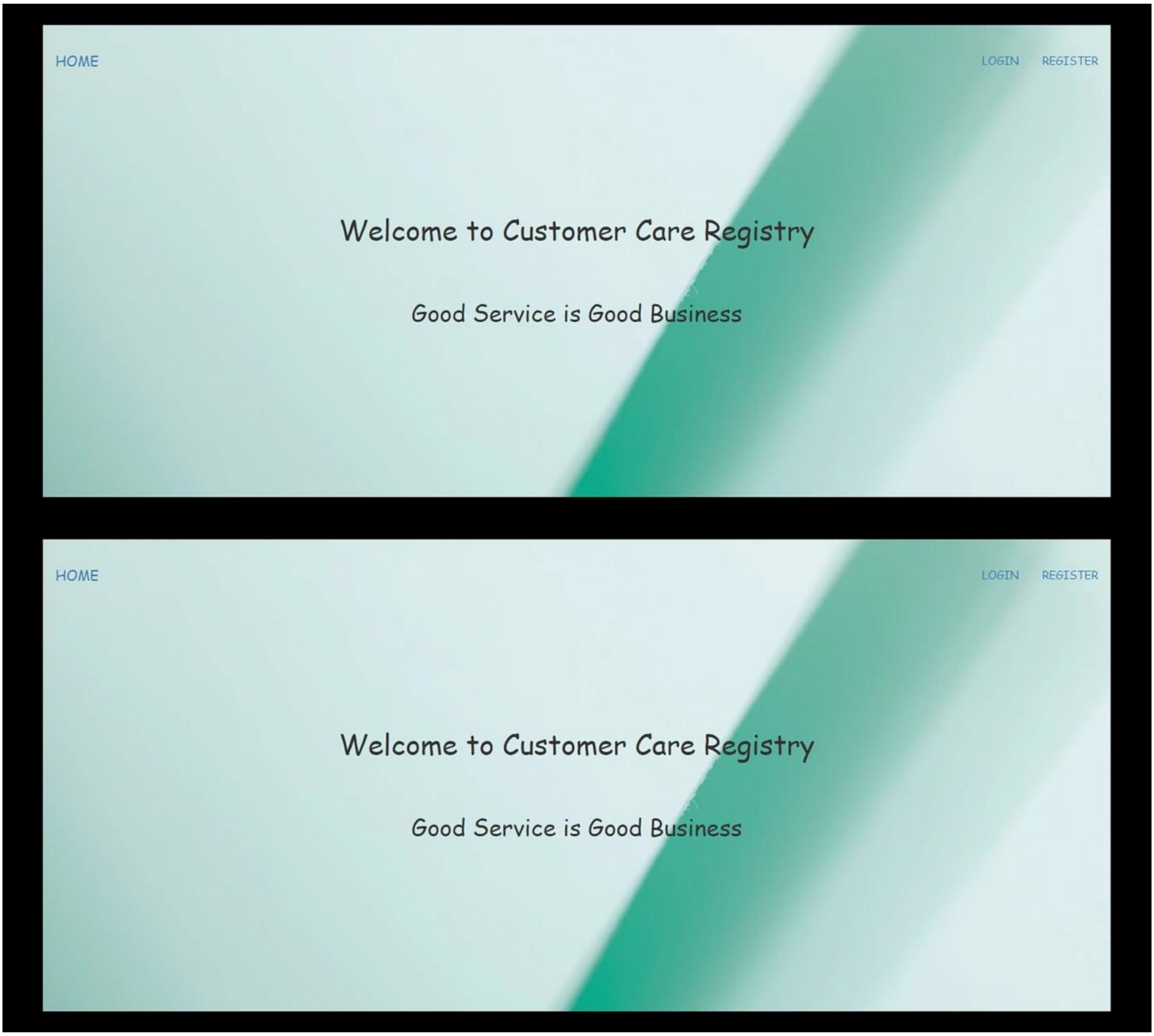
```
        <h1>Reply from the Agent:</h1>
        <div>{{ customer['REPLY'] }}</div>
        {% else %}
        <h1>Review Status:</h1>
        <div>Not Yet Reviewed</div>
        {% endif %}
    </div>
    <hr>
    {% endif %}
    <div>
        <div id="add_elem">
            <button id="form" class="btn btn-success" onclick="form_create()">Create Query</button>
            <a class="btn btn-primary" href="{{ url_for('redir') }}">Back to Home</a>
        </div>
    </div>
</center>

{% elif session['role'] == 'agent' %}
<hr>
<center>
    <form action="{{ url_for('agent_submit_reply') }}" method="POST">
        <table class="table table-hover">
            <tr>
                <th scope="col">Name</th>
                <th scope="col">Ticket No</th>
                <th scope="col">Query</th>
                <th scope="col">Reply</th>
            </tr>
            {% for cus in session['customer'] %}
            {% if cus['ASSIGNED_AGENT'] == session['name'] and cus['REPLY']!='none' and
cus['REVIEW_STATUS'] != 1 %}

            <tr>

                <td>
                    <select name="name">
                        <option value="{{ cus['USERNAME'] }}">{{ cus['USERNAME'] }}</option>
                    </select>
                </td>
                <td>
                    {{ cus['TICKET'] }}
                </td>
                <td>
                    {{ cus['QUERY'] }}
                </td>
                <td><textarea name="text" cols="50" rows="4"></textarea></td>
            </tr>
            {% endif %}
            {% endfor %}
            <tr>
                <td colspan="4">
                    <center><input type="submit" value="submit"></center>
                </td>
            </tr>

        </table>
    </form>





</center>
{% elif session['role']=='admin' %}
<center>
    <div class="container">
        <form action="{{ url_for('admin_query') }}" method="POST">
            <h1>All Customers</h1>
            <table class="table table-hover">
                <tr>
                    <th scope="col">Name</th>
                    <th scope="col">Ticket No</th>
                    <th scope="col">Email</th>
                    <th scope="col">Query</th>
                    <th scope="col">Query Status</th>
                    <th scope="col">Assigned to</th>
                    <th scope="col">Assign To</th>
                </tr>

                {% for cus in session['customer'] %}
                {% if cus['QUERY'] != 'none' and cus['REVIEW_STATUS'] != 1 %}

                <tr>

                    <td>
                        <select name="cus_name">
                            <option value="{{ cus['USERNAME'] }}">{{ cus['USERNAME'] }}</option>
                        </select>
                    </td>
                    <td>
                        {{ cus['TICKET'] }}
                    </td>
                    <td>
                        <select name="email">
                            <option value="{{ cus['EMAIL'] }}">{{ cus['EMAIL'] }}</option>
                        </select>
                    </td>

                    <td>
                        {{ cus['QUERY'] }}
                    </td>
                    <td>
                        {% if cus['REVIEW_STATUS'] == 1 %}
                        Reviewed
                        {% else %}
                        Not yet Reviewed
                        {% endif %}
                    </td>
                    <td>{{ cus['ASSIGNED_AGENT'] }}</td>
                    <td>
                        <select name="agent_name">
                            <option value="none">Select agent</option>
                            {% for agt in session['agent'] %}
                            <option value="{{ agt['USERNAME'] }}">{{ agt['USERNAME'] }}</option>
                            {% endfor %}

                        </select>
                    </td>
                </tr>
                {% endif %}
                {% endfor %}
                <tr>
                    <td colspan="7">
                        <center><input type="submit" value="submit"></center>
                    </td>

                </tr>

            </table>
        </form>
    </div>


    <h1>Available Agents</h1>

    <table>
        <tr>
            <th>Name</th>
        </tr>

        {% for agt in agent %}
        <tr>
            <td>
                {{ agt['USERNAME'] }}
            </td>
        </tr>
        {% endfor %}

    </table>
</center>
{% else %}
<h1>UnIdentified User</h1>
{% endif %}
{%endblock%}
```

success.html

```
{% extends 'base.html' %}
{% block title %}
Result
{% endblock %}
{% block content %}
<h1>{{ msg }}</h1>
<a class="btn btn-primary" href="{{ url_for('home') }}">Home</a>
{% endblock %
```

CHAPTER 9

RESULTS

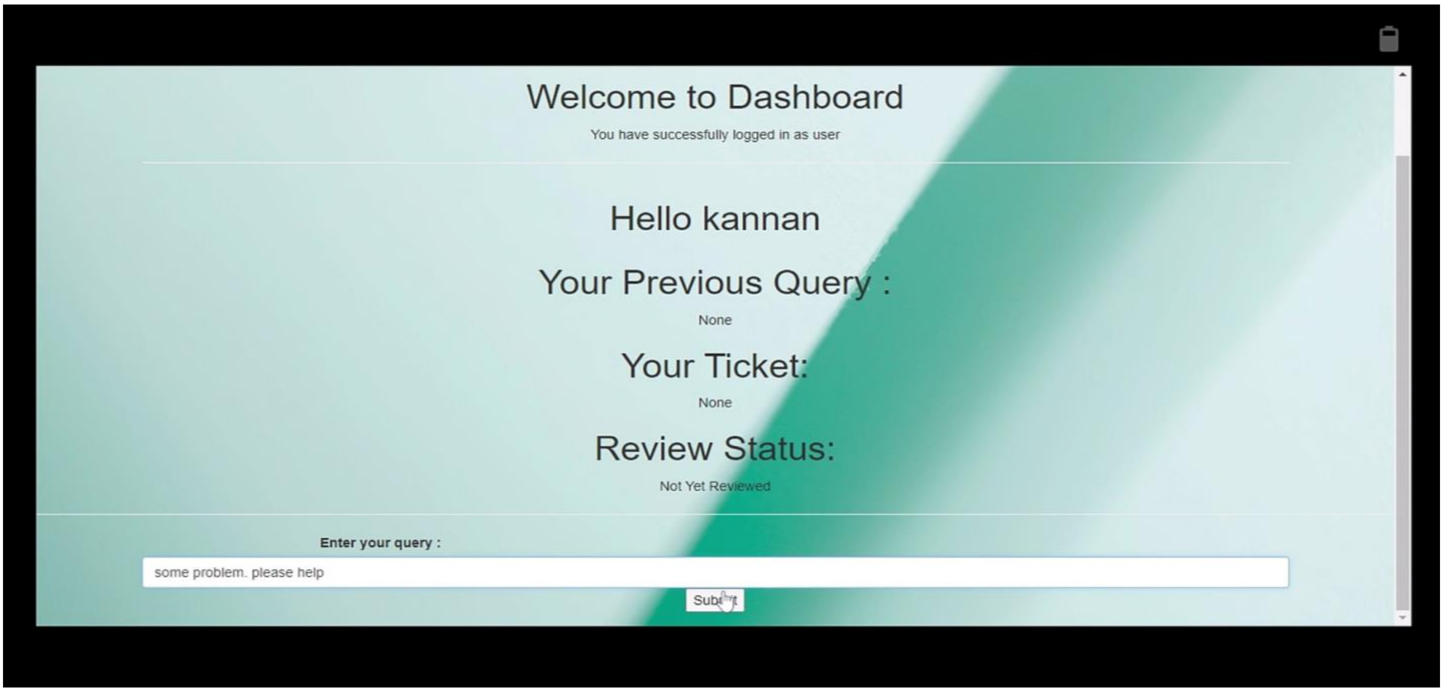


Welcome to Customer Care Registry

Good Service is Good Business

Welcome to Customer Care Registry

Good Service is Good Business



Welcome to Dashboard

You have successfully logged in as user

Hello kannan

Your Previous Query :

None

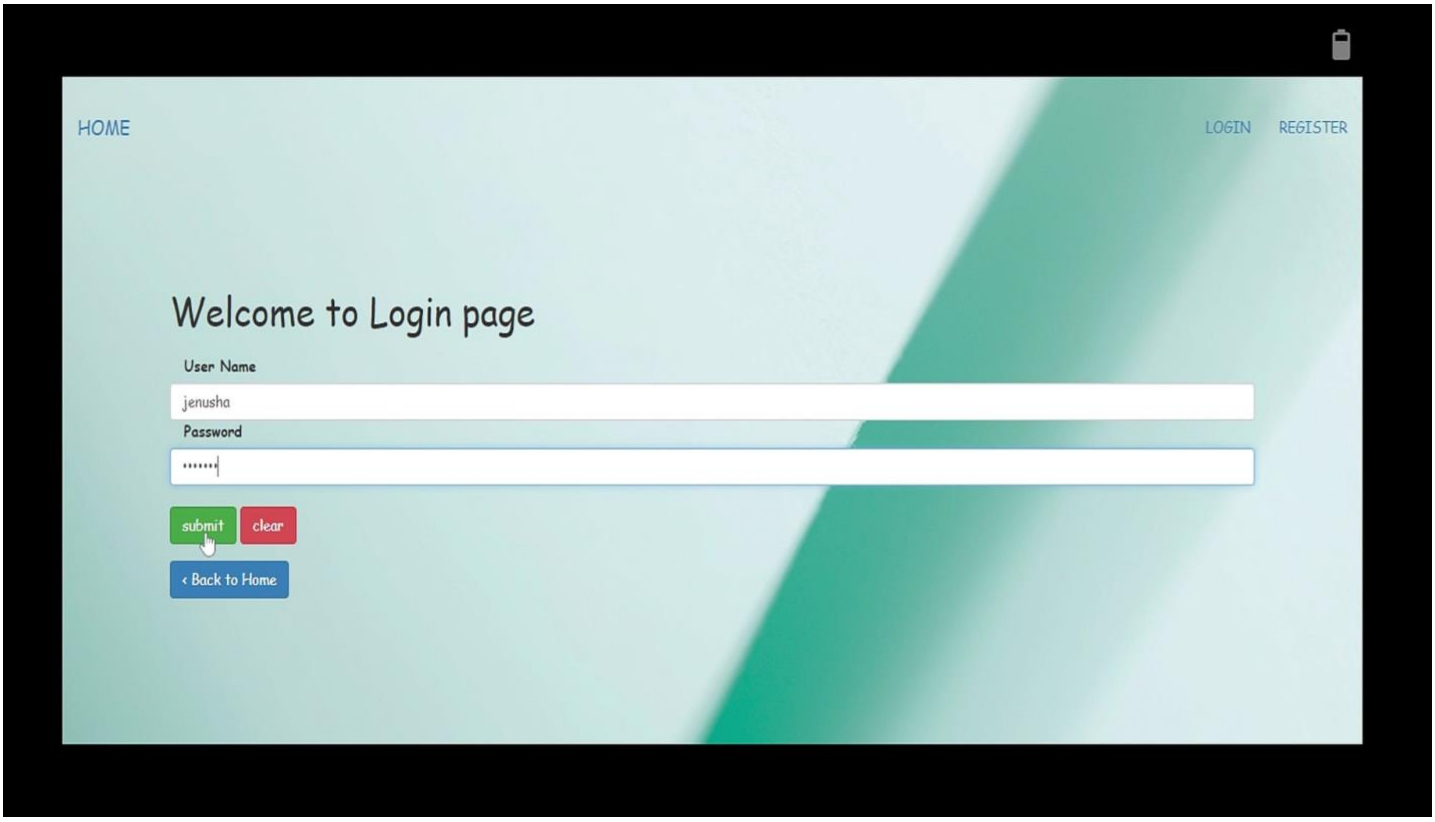
Your Ticket:

None

Review Status:

Not Yet Reviewed

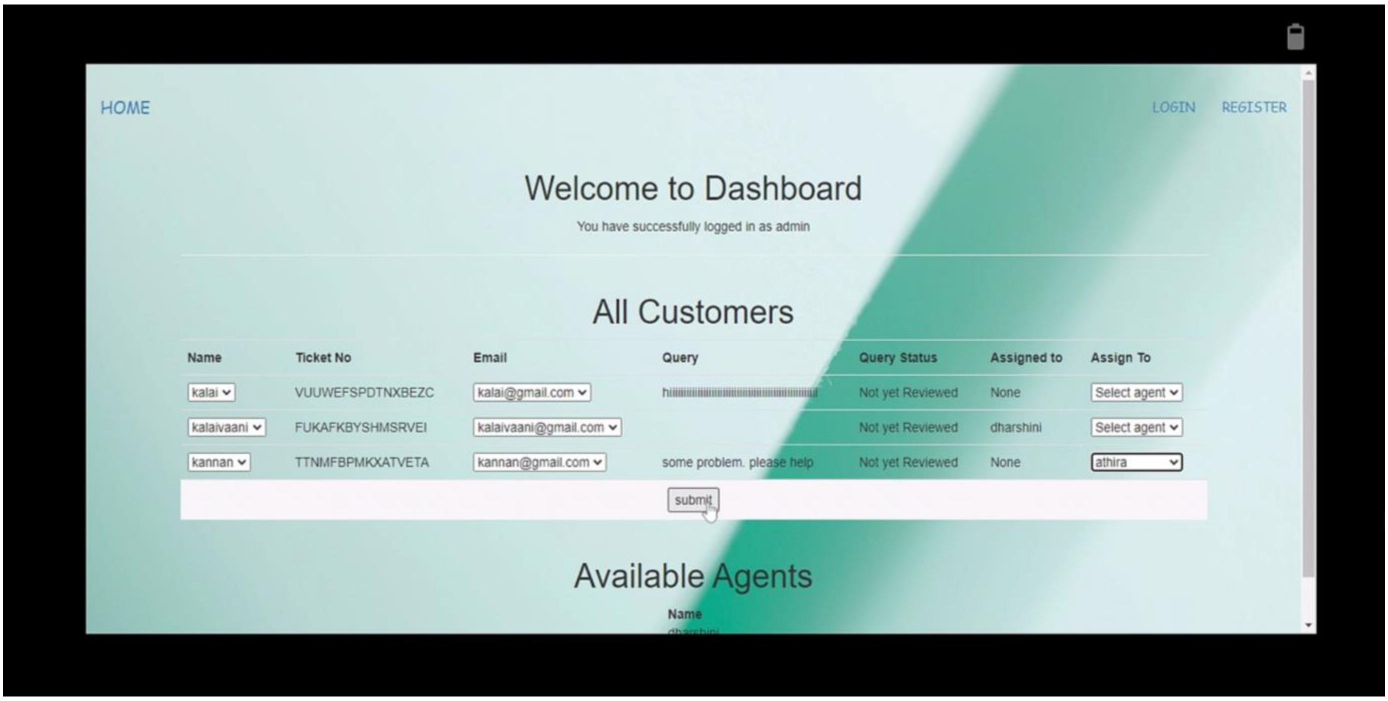
Enter your query : some problem. please help



Welcome to Login page

User Name: jenusha

Password: •••••••



Welcome to Dashboard

You have successfully logged in as admin

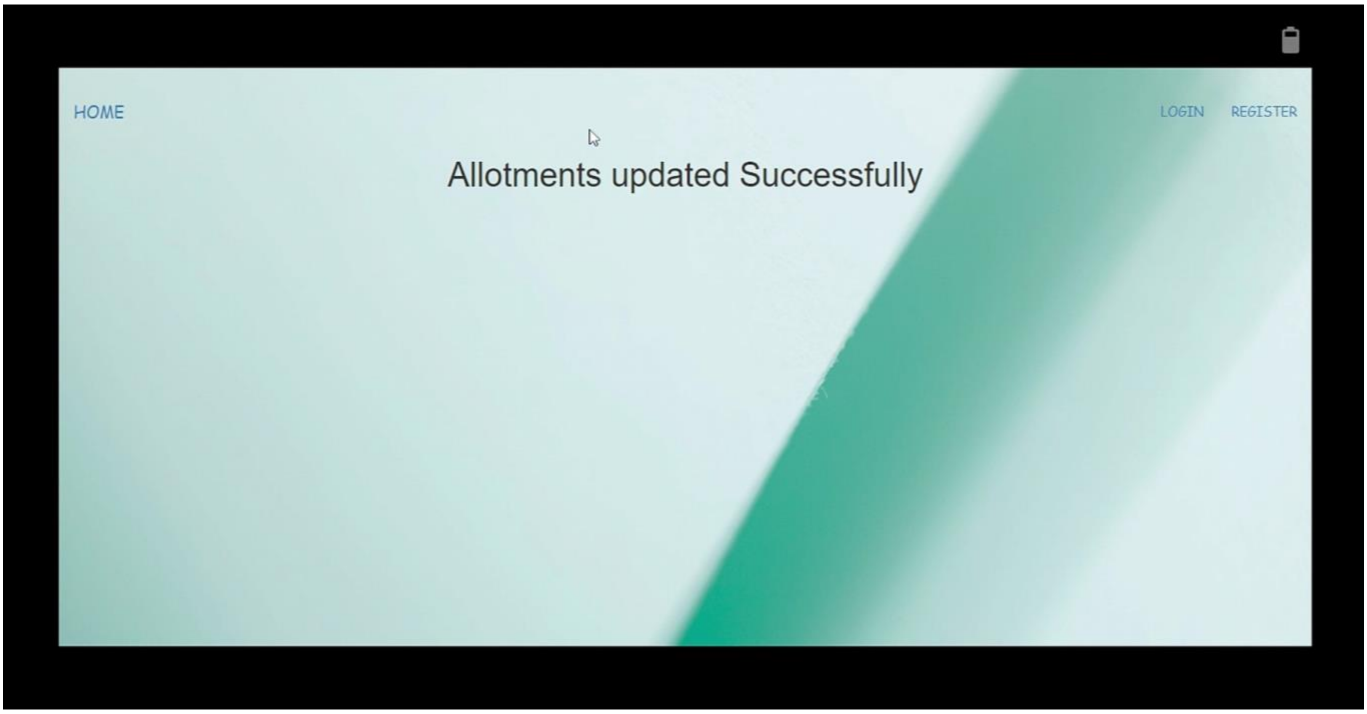
All Customers

Name	Ticket No	Email	Query	Query Status	Assigned to	Assign To
kalai	VUUWEFSPDTNXBEZC	kalai@gmail.com	hiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii	Not yet Reviewed	None	Select agent
kalaivaani	FUKAFKBYSHMSRVEI	kalaivaani@gmail.com		Not yet Reviewed	dharshini	Select agent
kannan	TTNMFBPMKXATVETA	kannan@gmail.com	some problem. please help	Not yet Reviewed	None	athira

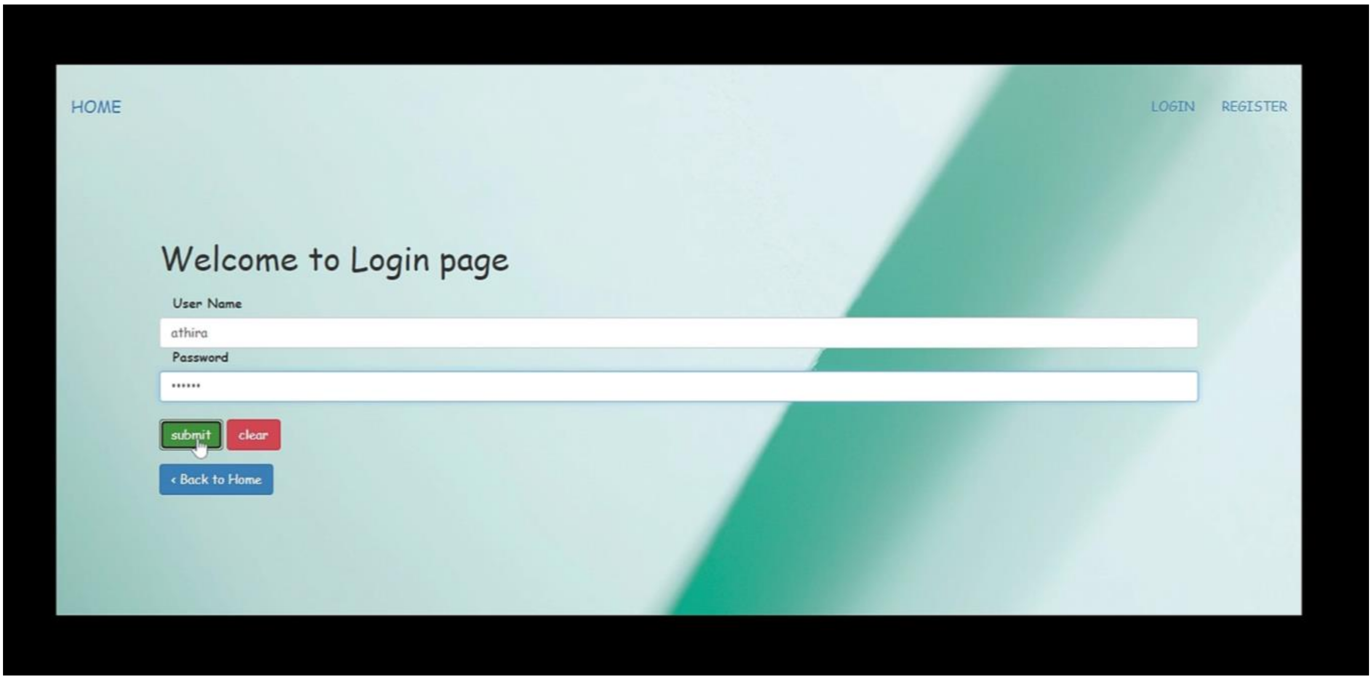
Available Agents

Name

dharshini



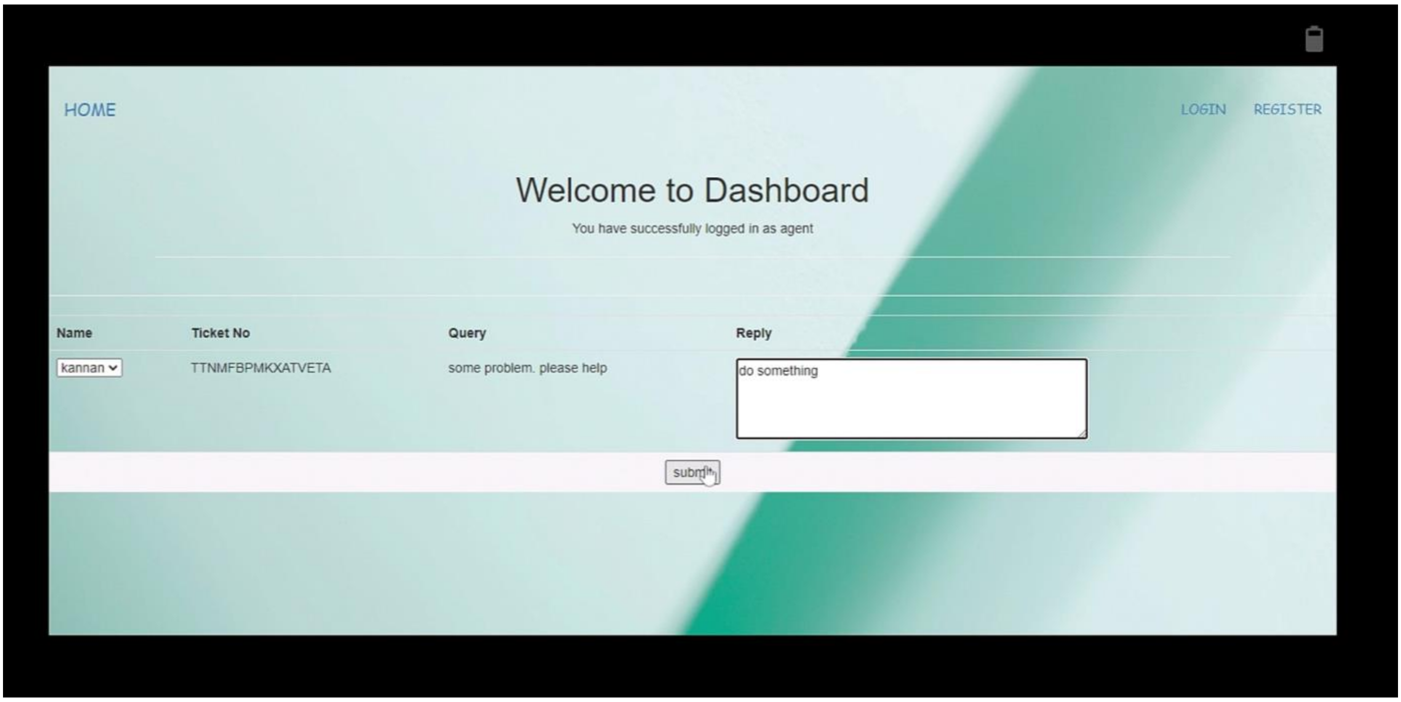
Allotments updated Successfully



Welcome to Login page

User Name: athira

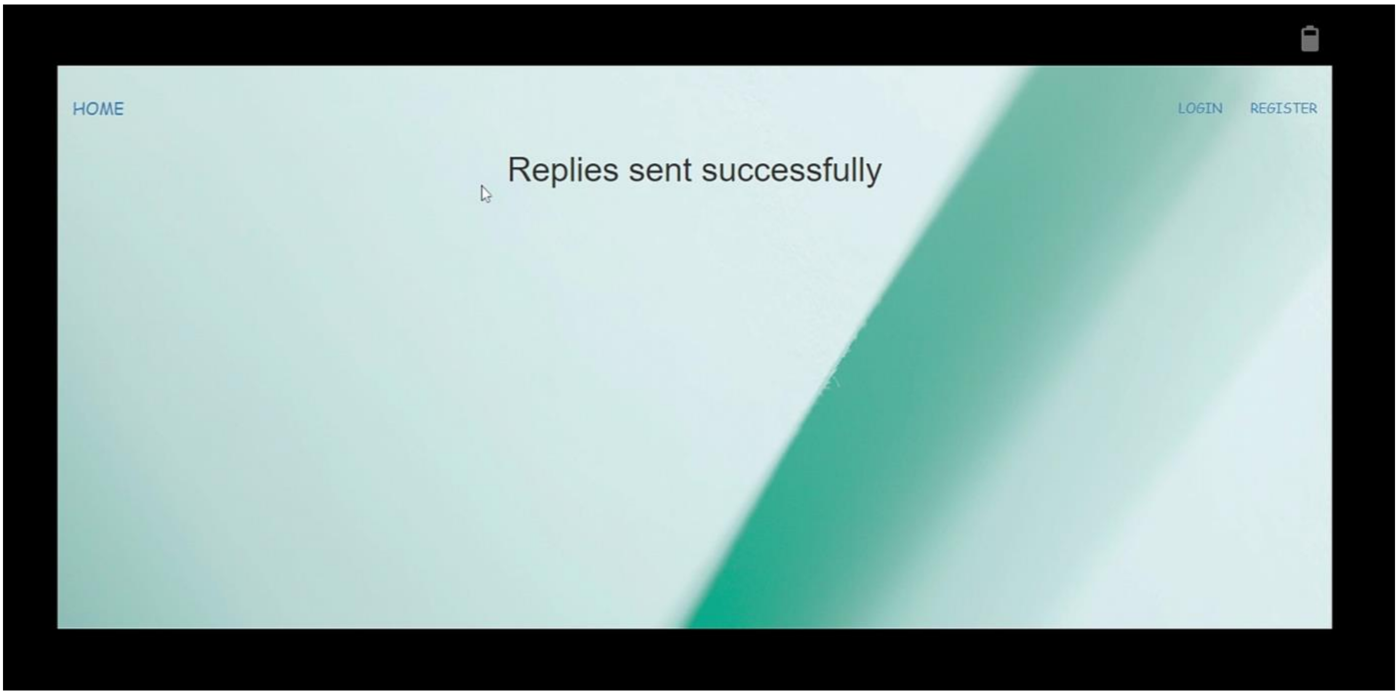
Password: ••••••



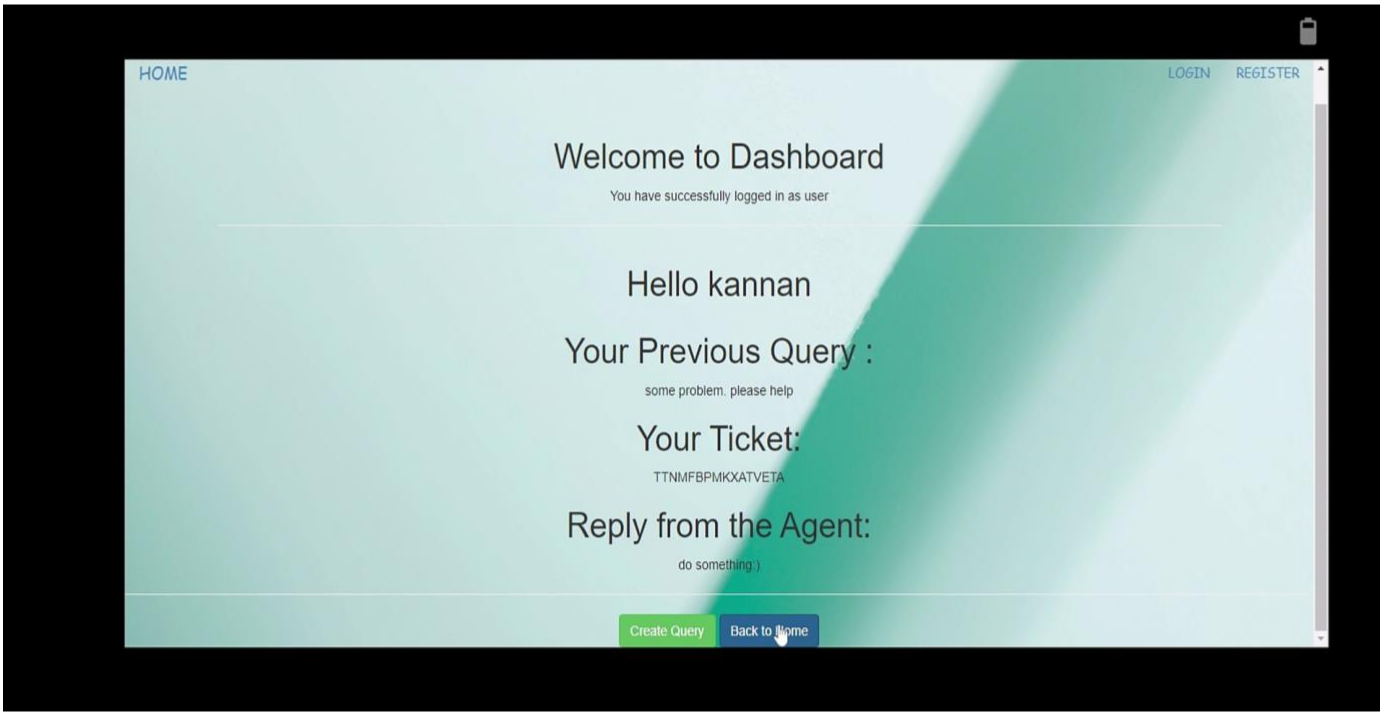
Welcome to Dashboard

You have successfully logged in as agent

Name	Ticket No	Query	Reply
kannan	TTNMFBPMKXATVETA	some problem. please help	do something



Replies sent successfully



Welcome to Dashboard

You have successfully logged in as user

Hello kannan

Your Previous Query :

some problem. please help

Your Ticket:

TTNMFBPMKXATVETA

Reply from the Agent:

do something:)

CHAPTER 10

ADVANTAGES AND DISADVANTAGES

Advantages

- Customer satisfaction enhancement.
- Product/service upgrade.
- Improvement of policies and procedures.
- Boost in customer communication.

Disadvantages

- required active internet connection

CHAPTER 10

CONCLUSION

In this project we developed a application that will help the customer to satisfy their complaints. This application provide service to handle the customer complaints without difficulty .This application is also easy to use and is simple than the regular method. 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 we live 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.

CHAPTER 12

FUTURE SCOPE

The future scope of this project is very broad . Few of them are:

- The total time for the customer satisfaction becomes faster.
- This helps the customer to clear the quires.
- Our system primarily focuses on building an efficient and user friendly Android mobile application.

CHAPTER 13

APPENDIX

Source Code

app.py

```
from flask_mail import Mail, Message
import random
import string
from flask import Flask, render_template, request, redirect, url_for, session
import ibm_db


def Upper_Lower_string(length):
    result = ".join((random.choice(string.ascii_uppercase)
                     for x in range(length)))
    return result


app = Flask(__name__)
app.config['SECRET_KEY'] = 'flaskisawesome'
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=824dfd4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=30119;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=hgh01434;PWD=a5YsjGuZxcdsXkkj", ", ")
app.config['MAIL_SERVER'] = 'smtp.gmail.com'
app.config['MAIL_PORT'] = 465
app.config['MAIL_USERNAME'] = 'hanayah231@gmail.com'
app.config['MAIL_PASSWORD'] = 'qgewrsmkutkvyszx'
app.config['MAIL_USE_TLS'] = False
app.config['MAIL_USE_SSL'] = True
mail = Mail(app)


@app.route('/')
def home():
    return render_template('index.html')


@app.route('/register', methods=['POST', 'GET'])
def register():
    if request.method == "POST":
        global rs
        name = request.form.get('name')
        email = request.form.get('email')
        password = request.form.get('password')
        stmt = ibm_db.prepare(conn, 'SELECT * FROM user WHERE username=?')
        ibm_db.bind_param(stmt, 1, name)
        ibm_db.execute(stmt)
        rs = ibm_db.fetch_assoc(stmt)
        print(rs)
        if rs:
            msg = 'Account already Exists'
            return render_template('register.html', msg=msg)
        else:
            reg_stmt = ibm_db.prepare(
                conn, 'INSERT INTO user ("USERNAME","EMAIL","PASSWORD") VALUES(?,?,?)')
            ibm_db.bind_param(reg_stmt, 1, name)
            ibm_db.bind_param(reg_stmt, 2, email)
            ibm_db.bind_param(reg_stmt, 3, password)
            ibm_db.execute(reg_stmt)
            msg = 'Successfully Registered'
            return render_template('register.html', msg=msg)
    else:
        return render_template('register.html')


@app.route('/login', methods=['POST', 'GET'])
def login():
    if request.method == "POST":
        customer = list()
        agent = list()
        name = request.form['name']
        password = request.form['password']
        log_stmt = ibm_db.prepare(
            conn, 'SELECT * FROM user WHERE username=? and password=?')
        ibm_db.bind_param(log_stmt, 1, name)
        ibm_db.bind_param(log_stmt, 2, password)
        ibm_db.execute(log_stmt)
        rs = ibm_db.fetch_assoc(log_stmt)
        if rs:
            session['role'] = 'user'
            session['customer'] = rs
            print(rs)
            return render_template('dashboard.html')
        log_stmt = ibm_db.prepare(
            conn, 'SELECT * FROM agent WHERE username=? and password=?')
        ibm_db.bind_param(log_stmt, 1, name)
        ibm_db.bind_param(log_stmt, 2, password)
        ibm_db.execute(log_stmt)
        rs = ibm_db.fetch_assoc(log_stmt)
        if rs:
            cms = ibm_db.exec_immediate(conn, 'SELECT * FROM user')
            agt = ibm_db.exec_immediate(conn, 'SELECT * FROM agent')
            customers = ibm_db.fetch_assoc(cms)
            agents = ibm_db.fetch_assoc(agt)
            while customers:
                customer.append(customers)
                customers = ibm_db.fetch_assoc(cms)
            while agents:
                agent.append(agents)
                agents = ibm_db.fetch_assoc(agt)
            print(customer)
            print(agent)
            session['role'] = 'agent'
            session['name'] = rs['USERNAME']
            session['customer'] = customer
            session['agent'] = agent
            return render_template('dashboard.html')
        log_stmt = ibm_db.prepare(
            conn, 'SELECT * FROM admin WHERE username=? and password=?')
        ibm_db.bind_param(log_stmt, 1, name)
        ibm_db.bind_param(log_stmt, 2, password)
        ibm_db.execute(log_stmt)
        rs = ibm_db.fetch_assoc(log_stmt)
        if rs:
            cms = ibm_db.exec_immediate(conn, 'SELECT * FROM user')
            agt = ibm_db.exec_immediate(conn, 'SELECT * FROM agent')
            customers = ibm_db.fetch_assoc(cms)
            agents = ibm_db.fetch_assoc(agt)
            while customers:
                customer.append(customers)
                customers = ibm_db.fetch_assoc(cms)
            while agents:
                agent.append(agents)
                agents = ibm_db.fetch_assoc(agt)
            print(customer)
            print(agent)
            session['role'] = 'admin'
            session['customer'] = customer
            session['agent'] = agent
```

```
            return render_template('dashboard.html', agent=agent, customer=customer)
        else:
            msg = 'UID/Password is incorrect'
            return render_template('login.html', msg=msg)
    else:
        return render_template('login.html')


@app.route('/dashboard', methods=['POST', 'GET'])
def dashboard():
    return render_template('dashboard.html')


@app.route('/success', methods=['POST', 'GET'])
def success():
    if request.method == "POST":
        ticket = session['ticket'] = Upper_Lower_string(16)
        print(ticket, session['ticket'])
        query = request.form['query']
        sql = "UPDATE user SET QUERY=?,TICKET=?,REVIEW_STATUS=0 WHERE USERNAME=?"
        out = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(out, 1, query)
        ibm_db.bind_param(out, 2, session['ticket'])
        ibm_db.bind_param(out, 3, session['customer']['USERNAME'])
        status = ibm_db.execute(out)
        if status:
            msg = 'Success ! Your Ticket Nno is :', ticket, 'You can now return to the home page'
            return render_template('success.html', msg=msg)
        else:
            msg = 'Error Submitting your Query'
            return render_template('success.html', msg=msg)


@app.route('/redirect')
def redir():
    return redirect(url_for('home'))


@app.route('/querying', methods=['POST'])
def admin_query():
    msg = ""
    agent = request.form.getlist('agent_name')
    usr_name = request.form.getlist('cus_name')
    emails = request.form.getlist('email')
    for i in range(0, len(agent)):
        if agent[i] != 'none':
            try:
                qr = ibm_db.prepare(
                    conn, "UPDATE USER SET ASSIGNED_AGENT=? WHERE USERNAME=?")
                ibm_db.bind_param(qr, 1, agent[i])
                ibm_db.bind_param(qr, 2, usr_name[i])
                result = ibm_db.execute(qr)
                print(agent[i], usr_name[i], emails[i])
                msg = Message(
                    f'Hello {usr_name[i]}',
                    sender='hanayah231@gmail.com',
                    recipients=[f'{emails[i]}']
                )
                msg.body = f'Agent named {agent[i]} alloted to your query.{agent[i]} will be responding you soon within 24 hrs.'
                mail.send(msg)
                msg = 'Allotments updated Successfully'
            except:
                msg = "Error saving allotments/sending emails"
    return render_template('done.html', msg=msg)


@app.route('/executing...', methods=['POST', 'GET'])
def agent_submit_reply():
    names = request.form.getlist('name')
    text = request.form.getlist('text')
    print(names)
    print(text)
    for i in range(0, len(names)):
        if not text[i] == '':
            try:
                sql = 'UPDATE USER SET REPLY=?,REVIEW_STATUS=1 WHERE USERNAME=?'
                query = ibm_db.prepare(conn, sql)
                ibm_db.bind_param(query, 1, text[i])
                ibm_db.bind_param(query, 2, names[i])
                ibm_db.execute(query)

                msg='Replies sent successfully'
            except:
                msg = 'Error Sending replies'
    return render_template('done.html', msg=msg)


if __name__ == '__main__':
    app.run(debug=True,host="0.0.0.0")
```

Github link: **https://github.com/IBM-EPBL/IBM-Project-34164-1660232097**

Demo link : **https://drive.google.com/drive/folders/12pvFIp4Z4GiV_gUwHsZUqYCMJZXL7sF7**

Source Code Link :
https://drive.google.com/drive/folders/19H_ErrHQ8DyZi6IMzaMGgmUv3iUnU-eS