CUSTOMER CARE REGISTRY

NALAIYA THIRAN PROJECT BASED LEARNING

ON

PROFESSIONAL READINESS FOR INNOVATION,

EMPLOYABILITY AND ENTREPRENEURSHIP

A PROJECT REPORT

TEAM ID:PNT2022TMID37726

T. Dhanalakshmi	410119106012
R. Manoja	410119106031
I. Priyadharshini	410119106049
L. Sangeetha	410119106054

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

ADHI COLLEGE OF ENGINEERING AND TECHNOLOGY

KANCHEEPURAM – 631 605

ADHI COLLEGE OF ENGINEERING AND TECHNOLOGY

KANCHEEPURAM – 631 605

INTERNAL MENTOR

Ms. Vidhya ME.,

(Assistant Professor)

INDUSTRY MENTOR

Mr. Vasudeva Hanush

(IBM Team)

TABLE OF CONTENT

ABSTRACT

The main objective of this Online Customer Care and Service Center software is to develop an information system to store, maintain, update and process data relating to the shop. It will prepare various reports to aid in smooth and speedy functioning of ‘Service Center’ activities. Below are the objectives and goals of this project/software.

1. INTRODUCTION

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.

2. OBJECTIVE

- Besides these, the software we have to monitor the records of every customer, employee, attendance of the every employee and duty allocation of employee.
- Support the duty allocation, salary allocation, attendance and payroll for both day &night shifts and Maintain accounts.
- Well customize stock handling with advance feature of controlling and updating stocks, Prepare bills for different customers.
- Generate different kind of necessary reports and queries.

3. IDEATION PHASE

3.1 Literature Survey

The existing system is a semi-automated at where the information is stored in the form ofexcel 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. When the company pushes the wrong product or service to customer this can severely impact to company's profit, growth and brand reputation. The customer cannot track the status of the Queries that are posted by them. Some queries will be left Unanswered. To overcome this issues a good customer care should be provided to solve the customer's queries.

REFERENCE

PAPER 1:

Author name: Merlin Stone

Year of publishing: 1 st June 2011

Description: Merlin Stone is Head of Research at The Customer Framework. He is a leading expert in customer management, including strategies and tactics for customer recruitment, retention and development and has been a leading contributor to the development of the customer management assessment methodologies for which The Customer Framework is best known. He is author or co-author of many articles and 30 books on customer management. The UK' s Chartered Institute of Marketing listed him in 2003 as one of the world's top 50 marketing thinkers, he was nominated as one of the 20 most influential people in the direct marketing industry in a Precision Marketing readership poll in 2003.

PAPER 2:

Author name: Wangenheim & Bayon

Year of publishing: September 2019

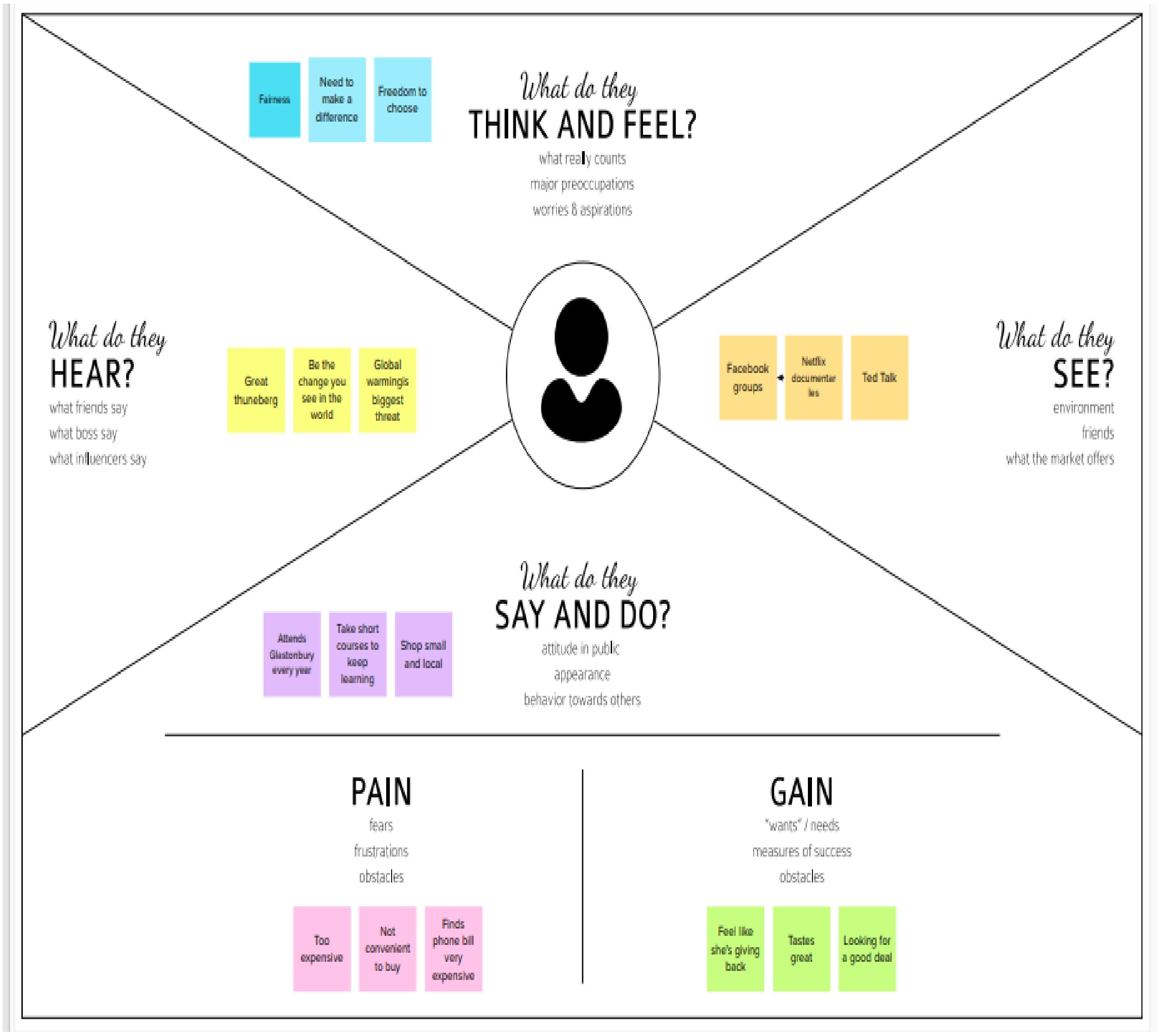
Description: Customer satisfaction is now for all companies the primary criterion for the assessment of their relationship with the market, a permanent object of their operating policies and an important element for the reinforcement of company reputation, as well as a fundamental guide to direct operational processes. So this paper is done in order to have a deeper understanding of the customers satisfaction but especially to help the students, the managers and also all people who can use it.

We will be going to see some definitions of the customer satisfaction, factors affecting customer satisfaction, and also measuring the customer satisfaction.

PAPER 3:

Author name: Arnaud
Year of publishing: 1987
Description: Offered four dimensions of service that form a system in which the elements can reinforce one another or weaken them. The Technical dimension is the heart of the service offer, the technical solution, in which some variables are more visible than others. The Relational dimension impacts on the maintenance of credibility over time, and the Functional dimension concerns how service is delivered and the added value. The Institutional dimension is a result of the other three dimensions.

3.2 Prepare Empathy Map:



3.3 Ideation:

Problem Statement

Who does the problem affect?	Frequent online users.
What are the boundaries of the problem?	IT sector, e-Commerce, Online Booking System.
What is the issue?	Stay in touch with the customer until theissue is resolved An Agent will be assigned to the Customerto solve the problem. Based on complaint type related authoritytake action and try to solve complaint as early as possible.
When does the issue occurs?	Not knowing answer to a questions, When the customer needs does no satisfied, Transferring customer calls, Not having right tools, Custome service workflowsaren't aligned with customer journey.
Where is the issue occurring?	The issue occur in several department like IT sectors, e-Commerce etc.,
Why is it important that we fix the problem?	By solving this issue, Customer can get thesolution for their raised complaints.

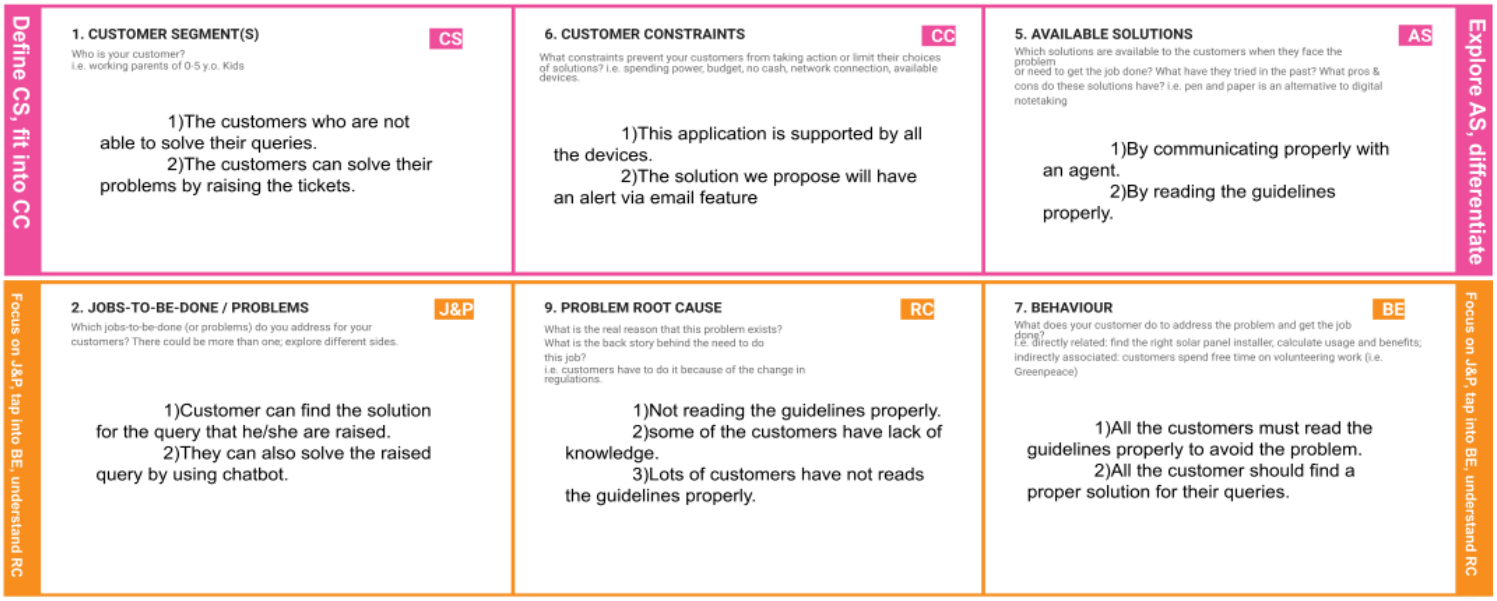
4. Project Design Phase – I

4.1 Proposed Solution:

S. No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To solve customer issues using Cloud Application Development.
2.	Idea / Solution description	Assigned Agent routing can be solved by directly routing to the specific agent about the issue using the specific email. Automated Ticket closure by using daily sync of the daily database. Status Shown to the Customer can display the status of the ticket to the customer. Regular data retrieval in the form of retrieving lost data.
3.	Novelty / Uniqueness	Assigned Agent Routing, Automated Ticket Closure, Status Shown to the Customer, and Backup data in case of failures.
4.	Social Impact / Customer Satisfaction	Customer Satisfaction, Customer can track their status and Easy agent communication.

5.	Business Model (Revenue Model)	● Key Partners - Third-party applications, agents, and customers. ● Activities - Customer Service, System Maintenance. ● Key Resources - Engineers, Multi-channel. ● Customer Relationship - 24/7 Email Support, Knowledge-based channel. ● Cost Structure - Cloud Platform, Offices.
6.	Scalability of the Solution	All customers are prioritized based on SLA (Service Level Agreement) - Urgent, Moderate, Low.

4.2 Problem Solution Fit:



Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Who is your customer?
i.e. working parents of 0-5 y.o. Kids

1)The customers who are not able to solve their queries.
2)The customers can solve their problems by raising the tickets.

6. CUSTOMER CONSTRAINTS CC

What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

1)This application is supported by all the devices.
2)The solution we propose will have an alert via email feature

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customers when they face the problem
or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking

1)By communicating properly with an agent.
2)By reading the guidelines properly.

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.

1)Customer can find the solution for the query that he/she are raised.
2)They can also solve the raised query by using chatbot.

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists?
What is the back story behind the need to do this job?
i.e. customers have to do it because of the change in regulations.

1)Not reading the guidelines properly.
2)some of the customers have lack of knowledge.
3)Lots of customers have not reads the guidelines properly.

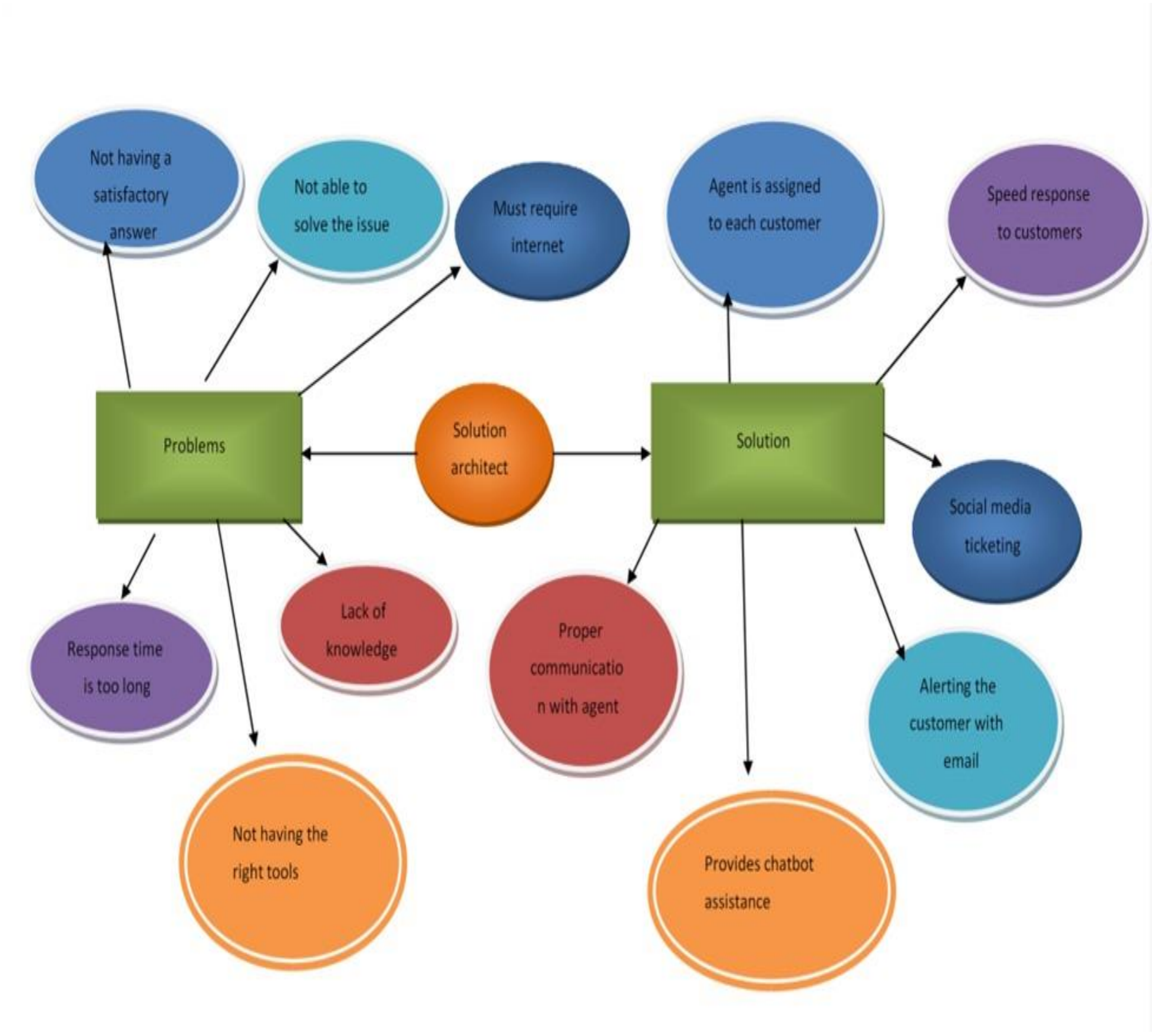
7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done?
i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

1)All the customers must read the guidelines properly to avoid the problem.
2)All the customer should find a proper solution for their queries.

Focus on J&P, tap into BE, understand RC

4.3 Solution Architecture:

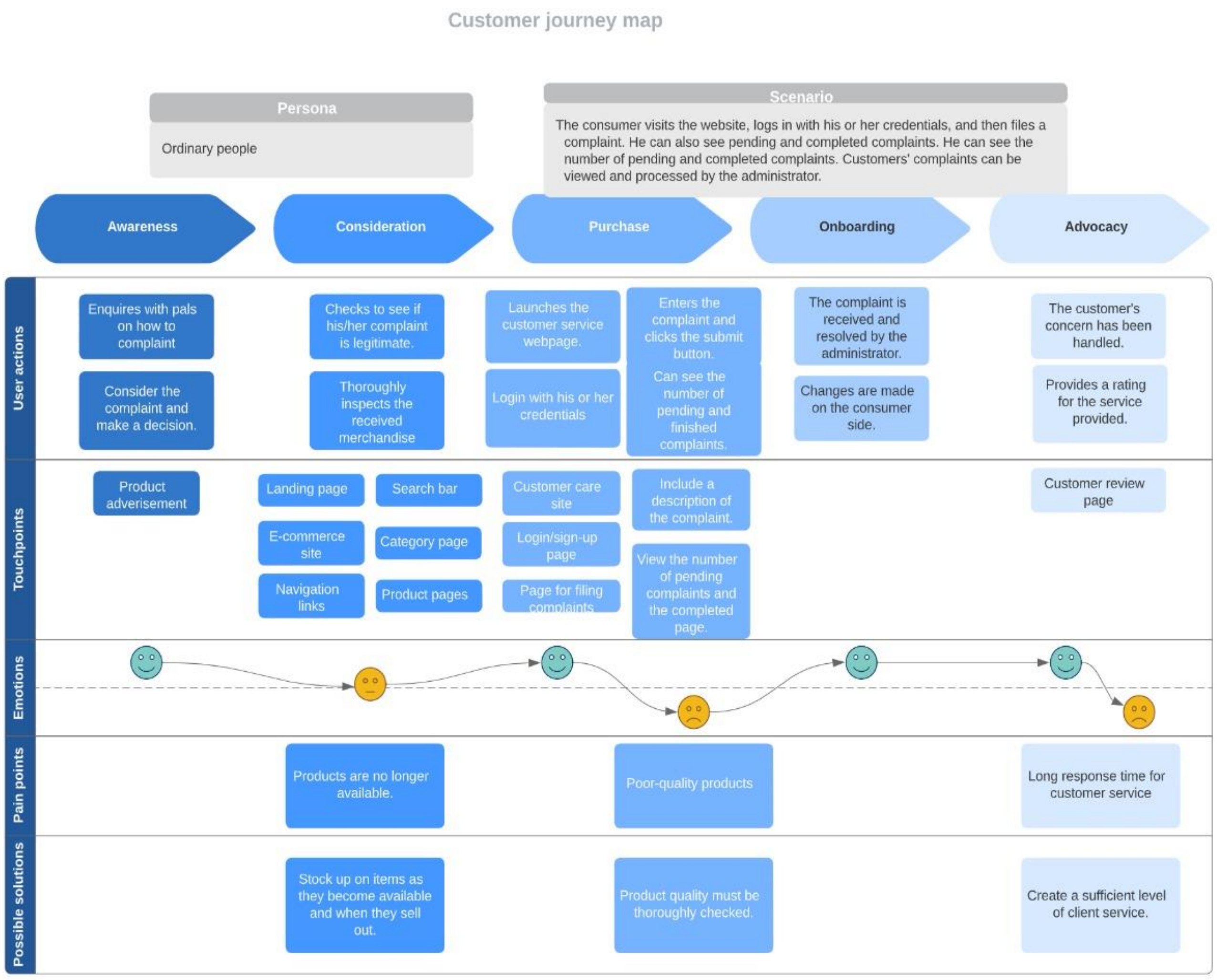

Not having a satisfactory answer
Not able to solve the issue
Must require internet
Agent is assigned to each customer
Speed response to customers
Problems
Solution architect
Solution
Social media ticketing
Response time is too long
Lack of knowledge
Proper communicatio n with agent
Alerting the customer with email
Not having the right tools
Provides chatbot assistance

4.4 Problem statement:

Who does the problem affect?	Frequent online users.
What are the boundaries of the problem?	IT sector, e-Commerce, Online Booking System.
What is the issue?	Stay in touch with the customer until theissue is resolved An Agent will be assigned to the Customerto solve the problem. Based on complaint type related authoritytake action and try to solve complaint as early as possible.
When does the issue occurs?	Not knowing answer to a questions, When the customer needs does not satisfied, Transferring customer calls, Not having right tools, Customer service workflows aren't aligned with customer journey.
Where is the issue occurring?	The issue occur in several department like IT sectors, e-Commerce etc.,
Why is it important that we fix the problem?	By solving this issue, Customer can get thesolution for their raised complaints.

5.Project Design Phase –II

5.1 Customer Journey map:


Customer journey map
Persona
Ordinary people
Scenario
Awareness
Consideration
Purchase
Onboarding
Advocacy
User actions
Enquires with pals on how to complaint
Consider the complaint and make a decision.
Checks to see if his/her complaint is legitimate.
Thoroughly inspects the received merchandise
Launches the customer service webpage.
Login with his or her credentials
Enters the complaint and clicks the submit button.
Can see the number of pending and finished complaints.
The complaint is received and resolved by the administrator.
Changes are made on the consumer side.
The customer's concern has been handled.
Provides a rating for the service provided.
Touchpoints
Product adverisement
Landing page
Search bar
E-commerce site
Category page
Navigation links
Product pages
Customer care site
Login/sign-up page
Page for filing complaints
Include a description of the complaint.
View the number of pending complaints and the completed page.
Customer review page
Emotions
Pain points
Products are no longer available.
Poor-quality products
Long response time for customer service
Possible solutions
Stock up on items as they become available and when they sell out.
Product quality must be thoroughly checked.
Create a sufficient level of client service.

The consumer visits the website, logs in with his or her credentials, and then files a complaint. He can also see pending and completed complaints. He can see the number of pending and completed complaints. Customers' complaints can be viewed and processed by the administrator.

5.2 Functional Requirement:

Following are the functional requirements of the proposed solution.

FR.NO.	Functional Requirement (Epic)	Sub Requirement (Story/Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIn Register with valid mobile number
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP Two step verification for new device login
FR-3	Agent Registration	Registration through Form Registration through Gmail Registration through LinkedIn Register with valid mobile number
FR-4	Agent Confirmation	Confirmation via Email Confirmation via OTP Two step verification for new device login
FR-5	Admin	Admin have both user details and agent detail Admin maintain agent allotment to the user based on problem's category

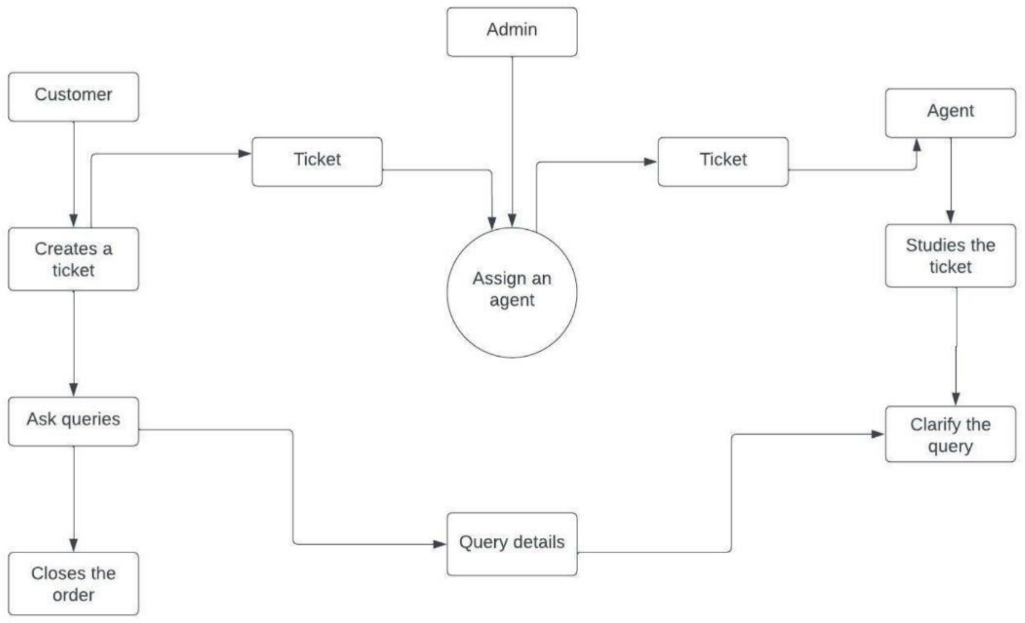
Non-functional Requirements:

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

FR.NO.	Non-Functional Requirement	Description
NFR-1	Usability	To provide optimal usability for our proposed solution we have mainly concentrated on easier navigation throughout our website. For user, they can easily login with their credentials and also they can register by themselves either with unique valid email id or wit their mobile number if they don't have any prior account. After good navigation we have concentrated on visual clarity and developed web application which looks pleasant and simple thus making easier accessible to any aged person. For the first time users, Guide tour will also be available in order to provide better user satisfaction. Also,

		made our web application flexible to all type of devices such as android, mac and desktops.
NFR-2	Security	Before any user trying to login their account to any new device, verification code will be sent either to their registered email id or to their registered mobile number. Only after entering their code, they will be allowed to login. That code will also made expire within particular time limit. Also notification will be sent for each and every user activity. Thus everyone will have a secured account and also their details will be maintained securely in the admin side.
NFR-3	Reliability	Since we had split the agents into categories, system's response time for each and every individual will be lesser. Thus making our web application more reliable.
NFR-4	Performance	In order to bring best performance, we have concentrated on overload of user requests. To minimize the overloads and to minimize the system's response time we have created more agent service. Agents will be separated and categorized according to user's needs. For example to resolve product missing category some agents will be assigned and ti resolve damaged products category some agents will be assigned. So every individual user will be allotted with individual agents.
NFR-5	Availability	Customer care registry will be made available even in the weekends and our agents will also be allotted at anytime to any individual user. User can interact with their respective agents 24*7 by following proper user-agent guidelines.
NFR-6	Scalability	With respect to increase in user's requests, allotment will be increased. Data storage will increase accordingly. Rescaling is always adaptable.

5.3 Data Flow Diagrams:

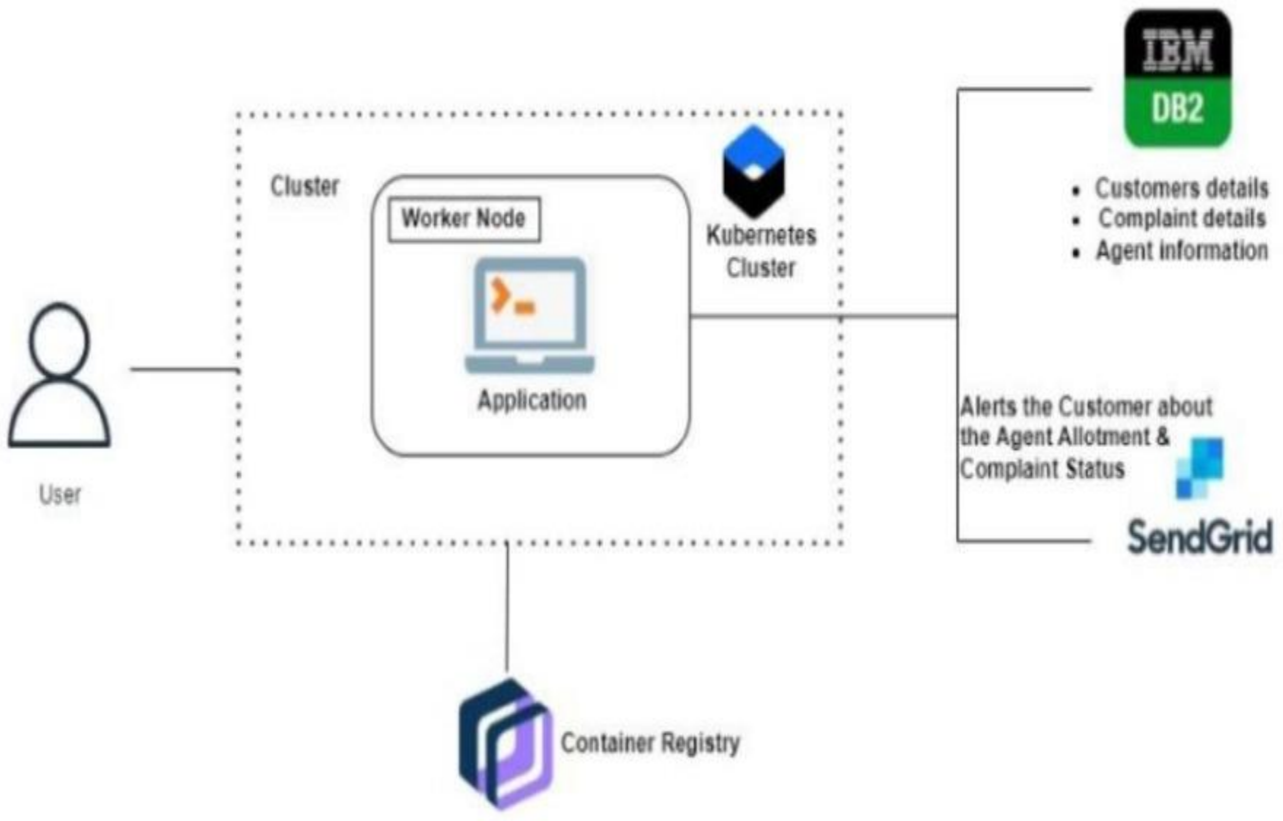


User Stories

Use the below template to list all the user stories for the product.

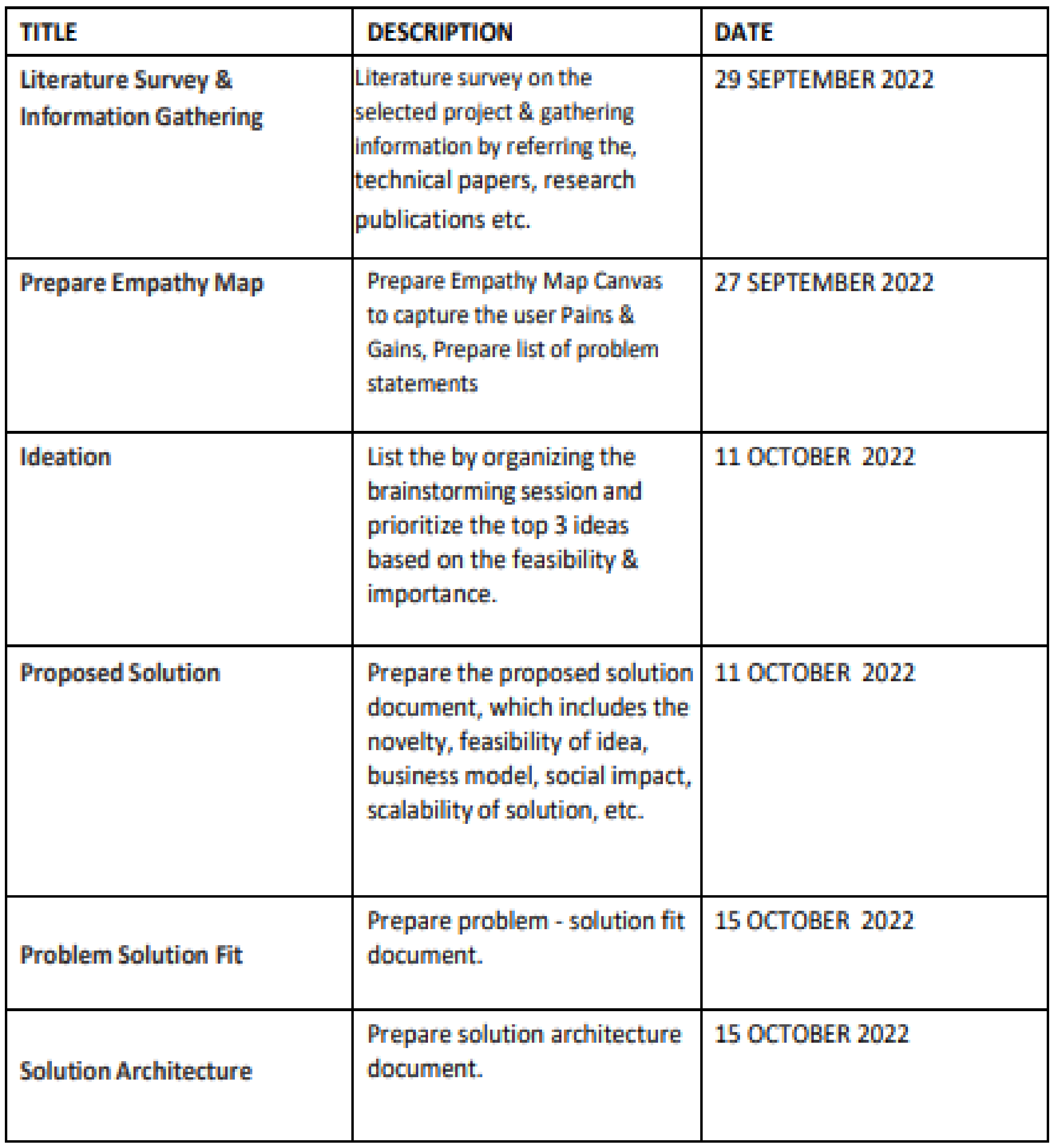
User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
	Acknowledgement	USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
	Ticket creation	USN-3	As a user, I can create new tickets with descriptions of my query.	I can create a ticket and ask my query.	Medium	Sprint-2
	Forget password	USN-4	As a user, I can reset my password by this option in case I forgot my password.	I can change the password	Medium	Sprint-1
	Login	USN-5	As a user, I can login into the application by entering email & password	I can access my account	High	Sprint-1
	Dashboard	USN-6	As a user, I am able to see all the tickets raised by me.	I get all information in the dashboard	Low	Sprint-1
Agent	Login	USN-1	As an agent, I can login to the application by entering the email id and password.	I can access my account	High	Sprint-2
	Forget password	USN-2	As an agent, I can reset my password in case I forget my password	I can change my password	High	Sprint-2
	Dashboard	USN-3	As an agent, I can able to see all the tickets raised by the customers	I can see all the tickets and clarify the queries	High	Sprint-2
Admin	Login	USN-1	As a admin, I can login to the application by entering email id and password	I can access my account	High	Sprint-3
	Agent creation	USN-2	As a admin, I can able to create agent for the customers to solve the queries	I can create agents	High	Sprint-3
	Forget password	USN-3	As a admin, I can reset my password by this option in case I forgot my password	I can change password	Medium	Sprint-3
	Assigning Agent	USN-4	As a admin, I can assign agents to the customers who raised the tickets.	I can assign agents to the customers	High	Sprint-3

5.4 Technology Architecture:



6.Project Planning Phase

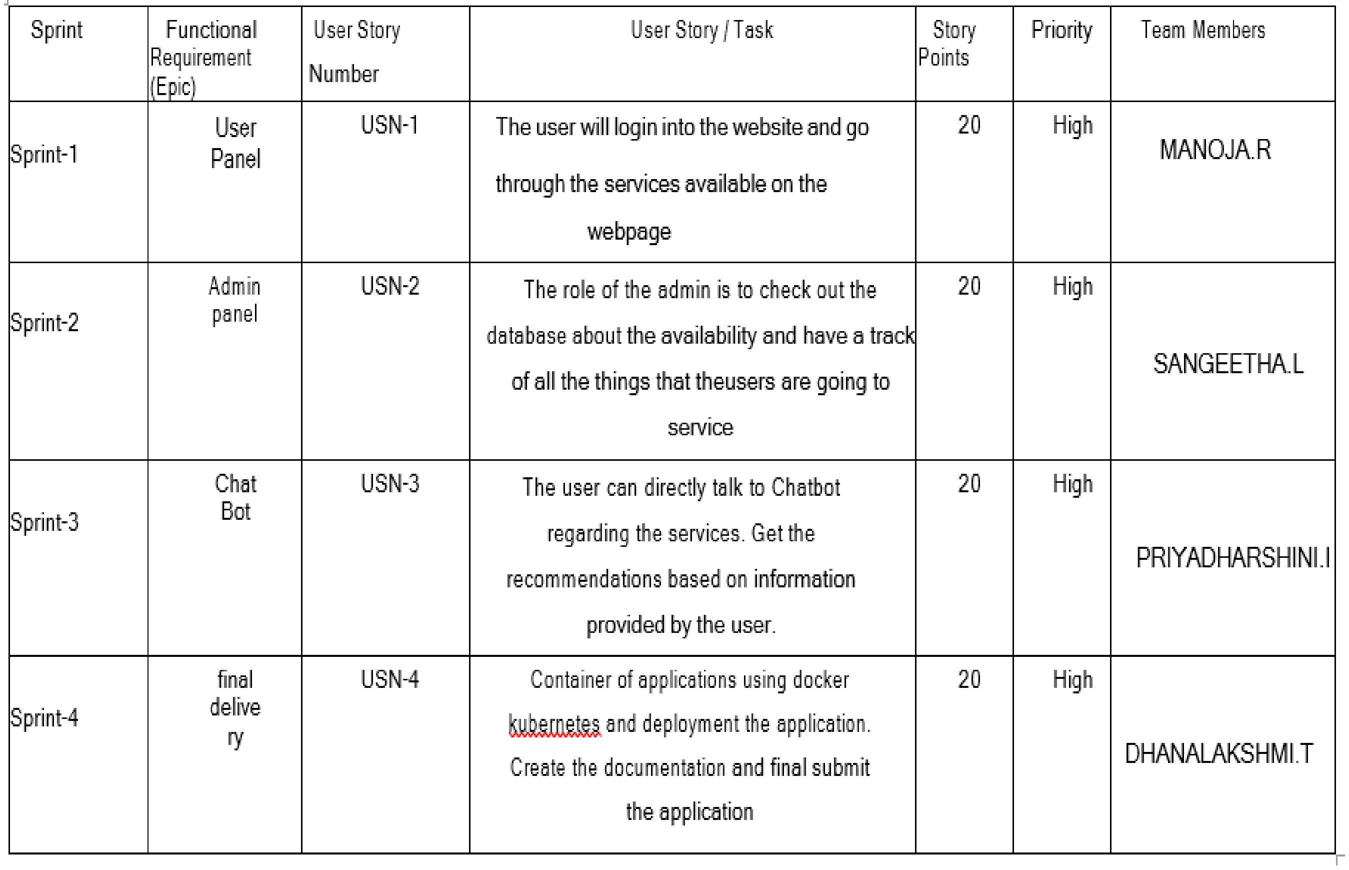
6.1 Prepare Milestone & Activity List:



TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Literature survey on the selected project & gathering information by referring the, technical papers, research publications etc.	29 SEPTEMBER 2022
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements	27 SEPTEMBER 2022
Ideation	List the by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	11 OCTOBER 2022
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	11 OCTOBER 2022
Problem Solution Fit	Prepare problem - solution fit document.	15 OCTOBER 2022
Solution Architecture	Prepare solution architecture document.	15 OCTOBER 2022

Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application (entry to exit).	15 OCTOBER 2022
Functional Requirement	Prepare the functional requirement document.	15 OCTOBER 2022
Data Flow Diagrams	Draw the data flow diagrams and submit for review.	19 OCTOBER 2022
Technology Architecture	Prepare the technology architecture diagram.	15 OCTOBER 2022
Prepare Milestone & Activity List	Prepare the milestones & activity list of the project.	28 OCTOBER 2022
Project Development - Delivery of Sprint-1, 2, 3 & 4	Develop & submit the developed code by testing it.	IN PROGRESS..

6.2 Sprint Delivery Plan:



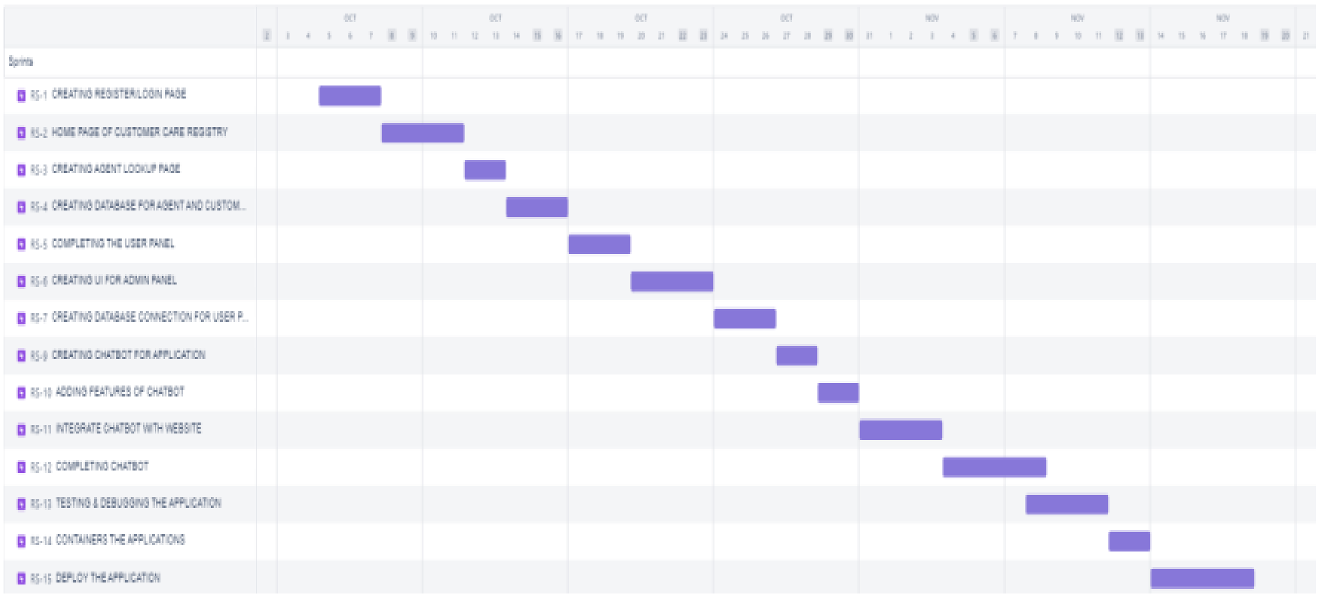
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	User Panel	USN-1	The user will login into the website and go through the services available on the webpage	20	High	MANOJA.R
Sprint-2	Admin panel	USN-2	The role of the admin is to check out the database about the availability and have a track of all the things that theusers are going to service	20	High	SANGEETHA.L
Sprint-3	Chat Bot	USN-3	The user can directly talk to Chatbot regarding the services. Get the recommendations based on information provided by the user.	20	High	PRIYADHARSHINI.I
Sprint-4	final delive ry	USN-4	Container of applications using docker kubernetes and deployment the application. Create the documentation and final submit the application	20	High	DHANALAKSHMI.T

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	28 Oct 2022	03 Nov 2022		03 Nov 2022
Sprint-2	20	6 Days	03 Nov 2022	08 Nov 2022		08 Nov 2022
Sprint-3	20	6 Days	08 Nov 2022	13 Nov 2022		13 Nov 2022
Sprint-4	20	6 Days	13 Nov 2022	19 Nov 2022		19 Nov 2022

Velocity:
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

BURNDOWN CHART

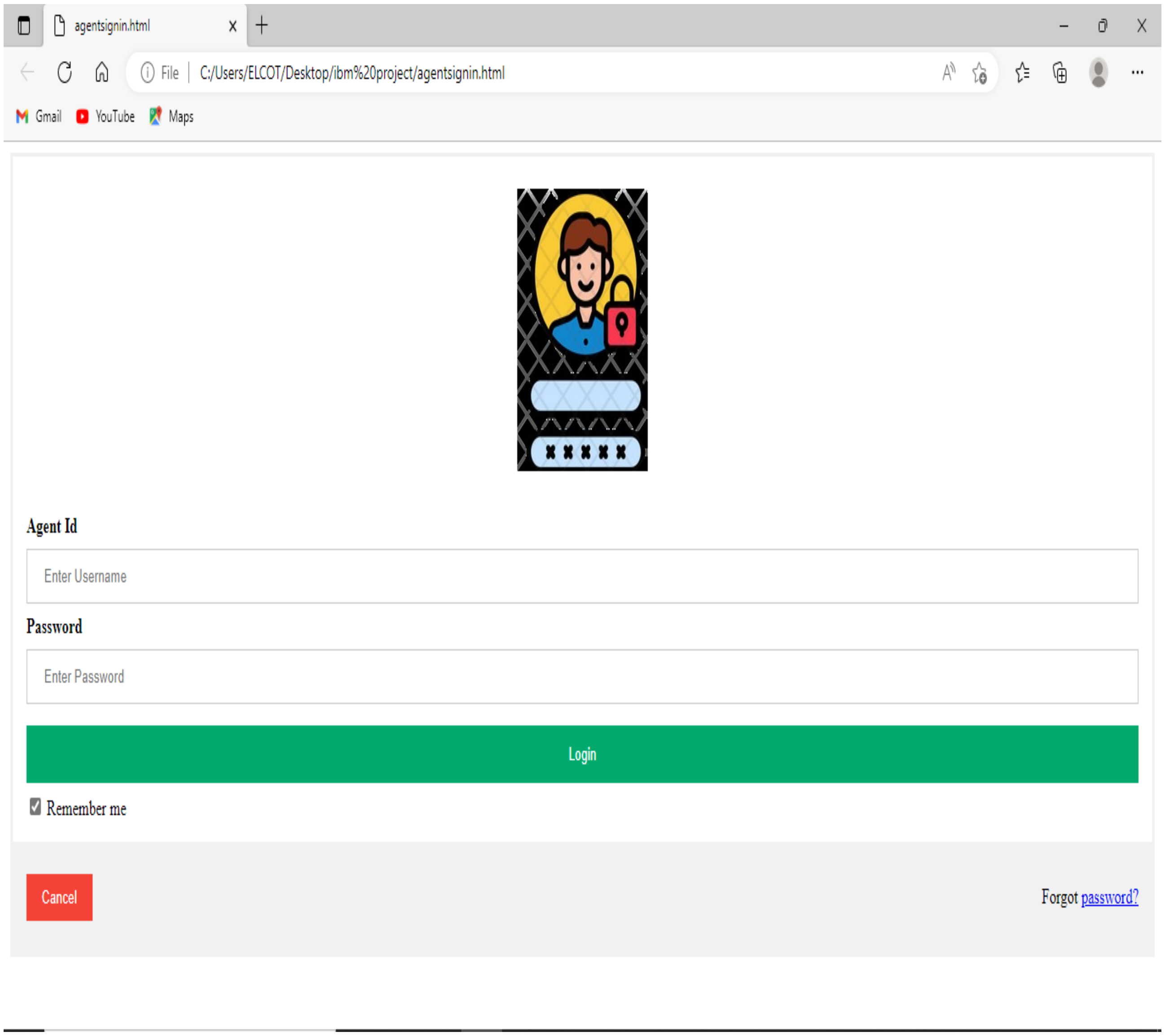


Sprints

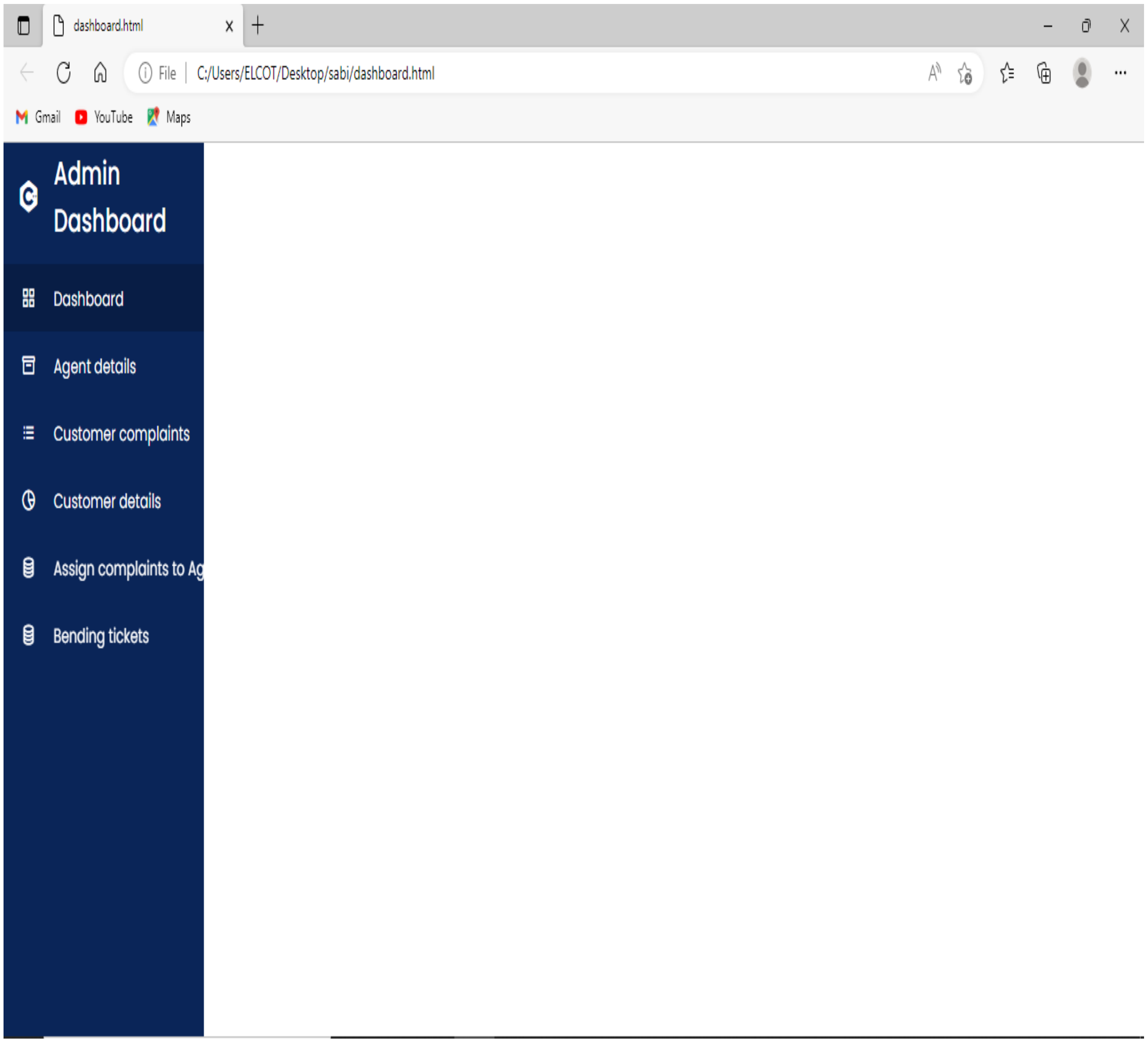
RS-1 CREATING REGISTER/LOGIN PAGE
RS-2 HOME PAGE OF CUSTOMER CARE REGISTRY
RS-3 CREATING AGENT LOOKUP PAGE
RS-4 CREATING DATABASE FOR AGENT AND CUSTOM...
RS-5 COMPLETING THE USER PANEL
RS-6 CREATING UI FOR ADMIN PANEL
RS-7 CREATING DATABASE CONNECTION FOR USER P...
RS-9 CREATING CHATBOT FOR APPLICATION
RS-10 ADDING FEATURES OF CHATBOT
RS-11 INTEGRATE CHATBOT WITH WEBSITE
RS-12 COMPLETING CHATBOT
RS-13 TESTING & DEBUGGING THE APPLICATION
RS-14 CONTAINERS THE APPLICATIONS
RS-15 DEPLOY THE APPLICATION

7. Project Development Phase

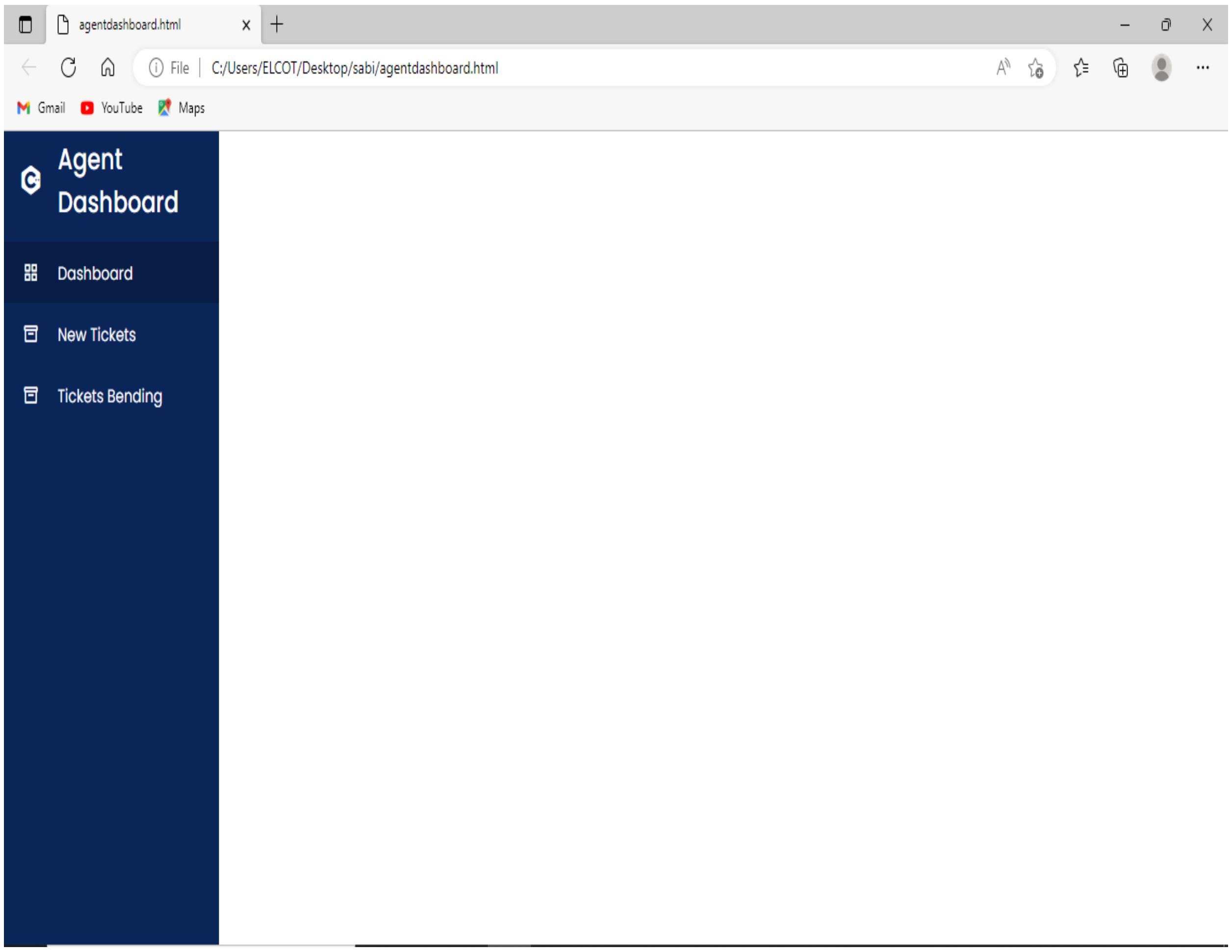
7.1 Project Development - Delivery of Sprint-1:



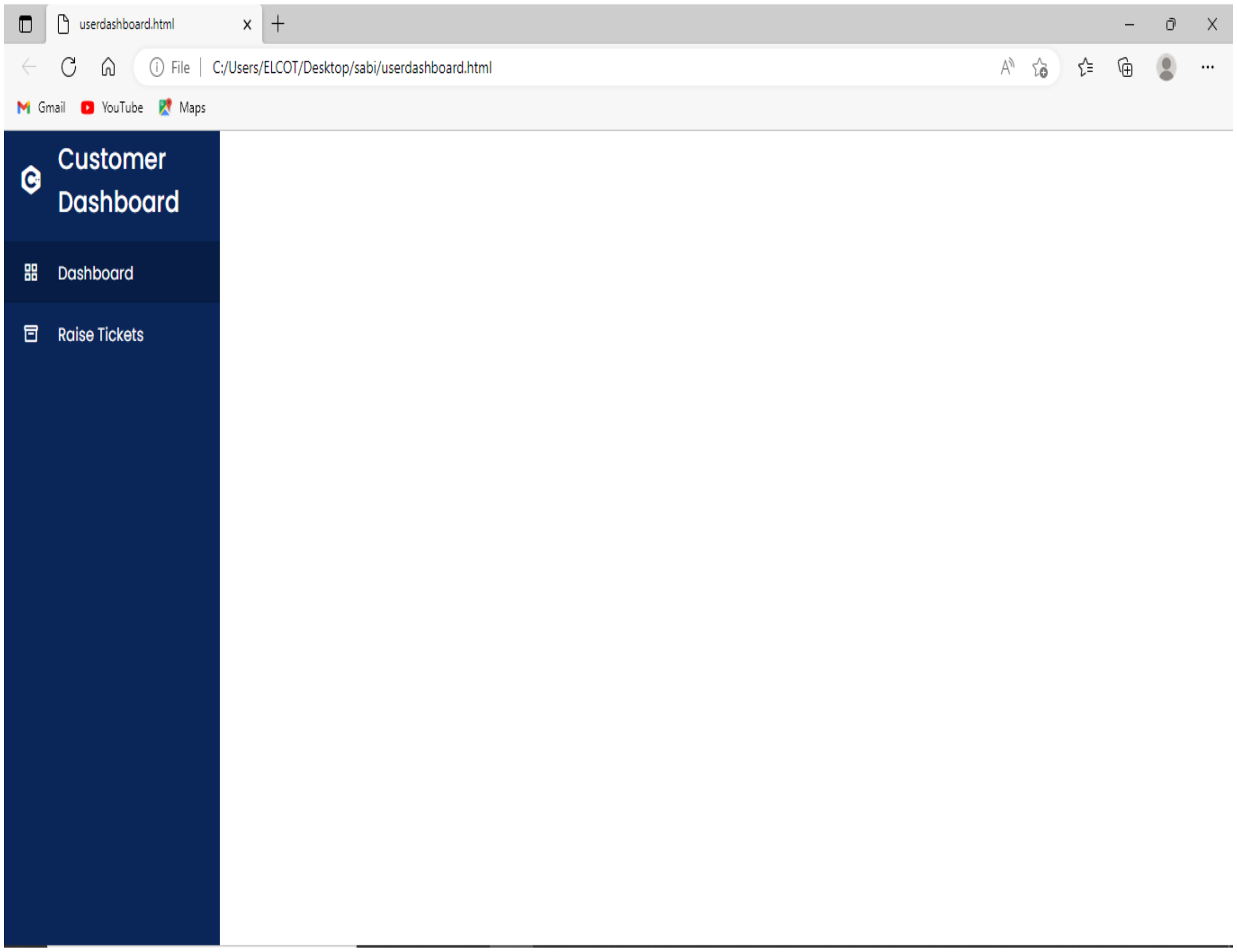
7.2 Project Development - Delivery of Sprint-2:



7.3 Project Development - Delivery of Sprint-3:



7.4 Project Development - Delivery of Sprint-4:



8.ADVANTAGES & DISADVANTAGES

ADVANTAGES:

- It retains the customer
- Gets you more references
- Increases profitability
- Gives you and your employees confidence
- Creates a holistic marketing scenario
- Competitive advantage
- Boost Customer Loyalty
- Enhance Brand Reputation
- Improve Products, Services, Procedures and Staff

DISADVANTAGES:

- Higher staff wages from hiring employees who are experts in customer service.
- Paying for staff training
- The extra services offered, such as refreshments
- Higher wage costs from the extra time staff take to provide post-sales service.

9.CONCLUSION

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.

10.FUTURE SCOPE

Machine learning (ML), emerging customer service trends 2022 can help businesses in improving overall CX. Chat applications powered by AI are trending. Large companies, as well as startups, are leveraging this to reduce costs and improve service for customers.

Predictive analytics has particularly proved to be very useful.

11.APPENDIX

Source Code

```
from __future__ import print_function
from audioop import add
import datetime
from unicodedata import name
from sib_api_v3_sdk.rest import ApiException
from pprint import pprint
from flask import Flask, render_template, request, redirect, url_for, session, flash
from markupsafe import escape
from flask import *
import ibm_db
import sib_api_v3_sdk
from init import randomnumber
from init import id
from init import hello

import datetime



conn =
ibm_db.connect("DATABASE=bludb;HOSTNAME="";PORT="";SECURITY=SSL;SSLServerCertifi
cate="";UID="";PWD=""", '', '')
print(conn)
print("connection successful...")

app = Flask(__name__)
app.secret_key = 'your secret key'


@app.route('/')
def home():
   message = "TEAM ID : PNT2022TMID37544" +" "+ "BATCH ID : B1-1M3E "
   return render_template('index.html',mes=message)
```

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


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

   return render_template('signinpageagent.html')


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


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



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


@app.route('/newissue/<name>', methods=['POST', 'GET'])
def newissue(name):
   name = name
   return render_template('complaint.html',msg=name)


@app.route('/forgot', methods=['POST', 'GET'])
def forgot():
    try:
        global randomnumber
        ida = request.form['custid']
        print(ida)
        global id
        id = ida
        sql = "SELECT EMAIL,NAME FROM Customer WHERE id=?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, ida)
        ibm_db.execute(stmt)
        emailf = ibm_db.fetch_both(stmt)
        while emailf != False:
            e = emailf[0]
            n = emailf[1]
            break

        configuration = sib_api_v3_sdk.Configuration()
        configuration.api_key['api-key'] = ""

        api_instance = sib_api_v3_sdk.TransactionalEmailsApi(
            sib_api_v3_sdk.ApiClient(configuration))
        subject = "Verification for Password"
        html_content = "<html><body><h1>Your verification Code is : <h2>" + \
            str(randomnumber)+"</h2> </h1> </body></html>"
        sender = {"name": "IBM CUSTOMER CARE REGISTRY",
                  "email": "ibmdemo6@yahoo.com"}
        to = [{"email": e, "name": n}]
        reply_to = {"email": "ibmdemo6@yahoo.com", "name": "IBM"}
        headers = {"Some-Custom-Name": "unique-id-1234"}
        params = {"parameter": "My param value",
                  "subject": "Email Verification"}
        send_smtp_email = sib_api_v3_sdk.SendSmtpEmail(
            to=to, reply_to=reply_to, headers=headers, html_content=html_content,
params=params, sender=sender, subject=subject)

        api_response = api_instance.send_transac_email(send_smtp_email)
        pprint(api_response)
        message = "Email send to:"+e+" for password"
        flash(message, "success")

    except ApiException as e:
        print("Exception when calling SMTPApi->send_transac_email: %s\n" % e)
        flash("Error in sending mail")
    except:
        flash("Your didn't Signin with this account")
    finally:
        return render_template('forgot.html')


@app.route('/verifyemail', methods=['POST', 'GET'])
def verifyemail():
    try:
        email = request.form['verifyemail']
        sql = "SELECT ID,NAME FROM Customer WHERE email=?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, email)
        ibm_db.execute(stmt)
        emailf = ibm_db.fetch_both(stmt)
        while emailf != False:
            id = emailf[0]
            name = emailf[1]
            break
        configuration = sib_api_v3_sdk.Configuration()
        configuration.api_key['api-key'] = ""

        api_instance = sib_api_v3_sdk.TransactionalEmailsApi(
            sib_api_v3_sdk.ApiClient(configuration))
        subject = "Regarding of your Customer Id"
        html_content = "<html><body><h1>Your Customer Id  is : <h2>" + \
            str(id)+"</h2> </h1> </body></html>"
        sender = {"name": "IBM CUSTOMER CARE REGISTRY",
                  "email": "ibmdemo6@yahoo.com"}
        to = [{"email": email, "name": name}]
        reply_to = {"email": "ibmdemo6@yahoo.com", "name": "IBM"}
        headers = {"Some-Custom-Name": "unique-id-1234"}
        params = {"parameter": "My param value",
                "subject": "Email Verification"}
        send_smtp_email = sib_api_v3_sdk.SendSmtpEmail(
            to=to, reply_to=reply_to, headers=headers, html_content=html_content,
params=params, sender=sender, subject=subject)

        api_response = api_instance.send_transac_email(send_smtp_email)

        pprint(api_response)
        message = "Email send to:"+email+" for password"
        flash(message, "success")

    except ApiException as e:
        print("Exception when calling SMTPApi->send_transac_email: %s\n" % e)
        flash("Error in sending mail.")
    except:
        flash("Database not found in mail! Please Register Your account.", "danger")
    finally:
        return render_template('signinpage.html')


@app.route('/otp', methods=['POST', 'GET'])
def otp():
    try:
        otp = request.form['otp']
        cusid = id
        print(id)
        sql = "SELECT PASSWORD FROM Customer WHERE id=?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, cusid)
        ibm_db.execute(stmt)
        otpf = ibm_db.fetch_both(stmt)
        while otpf != False:
            verify = otpf[0]
            break
        if otp == str(randomnumber):
            msg = "Your Password is "+verify+""
            flash(msg, "success")
            return render_template('forgot.html')
        else:
            flash("Wrong Otp", "danger")
    finally:
        return render_template('forgot.html')


@app.route('/admin', methods=['POST', 'GET'])
def admin():
    userdatabase = []
    sql = "SELECT * FROM customer"
    stmt = ibm_db.exec_immediate(conn, sql)
    dictionary = ibm_db.fetch_both(stmt)
    while dictionary != False:
        userdatabase.append(dictionary)
        dictionary = ibm_db.fetch_both(stmt)
    if userdatabase:
        sql = "SELECT COUNT(*) FROM customer;"
        stmt = ibm_db.exec_immediate(conn, sql)
        user = ibm_db.fetch_both(stmt)

    users = []
    sql = "select * from ISSUE"
    stmt = ibm_db.exec_immediate(conn, sql)
    dict = ibm_db.fetch_both(stmt)
    while dict != False:
        users.append(dict)
        dict = ibm_db.fetch_both(stmt)
    if users:
        sql = "SELECT COUNT(*) FROM ISSUE;"
        stmt = ibm_db.exec_immediate(conn, sql)
        count = ibm_db.fetch_both(stmt)
    agent = []
    sql = "SELECT * FROM AGENT"
    stmt = ibm_db.exec_immediate(conn, sql)
    dictionary = ibm_db.fetch_both(stmt)
    while dictionary != False:
        agent.append(dictionary)
        dictionary = ibm_db.fetch_both(stmt)

    if agent:
        sql = "SELECT COUNT(*) FROM AGENT;"
        stmt = ibm_db.exec_immediate(conn, sql)
        cot = ibm_db.fetch_both(stmt)

    return
render_template("admin.html",complaint=users,users=userdatabase,agents=agent,messag
e=user[0],issue=count[0],msgagent = cot[0])


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

    otp = request.form['otpv']
    if otp == 'C':
        try:
            insert_sql = f"delete from customer"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.execute(prep_stmt)
            flash("delected successfully the Customer", "success")
        except:
            flash("No data found in Customer", "danger")
        finally:
            return redirect(url_for('signuppage'))
    if otp == 'A':
        try:
            insert_sql = f"delete from AGENT"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.execute(prep_stmt)
            flash("delected successfully the Agents", "success")
        except:
            flash("No data found in Agents", "danger")
        finally:
           return redirect(url_for('signuppage'))

    if otp == 'C':
        try:
            insert_sql = f"delete from AGENT"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.execute(prep_stmt)
            flash("delected successfully the Complaints", "success")
        except:
            flash("No data found in Complaints", "danger")
        finally:
            return redirect(url_for('signuppage'))

@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        try:

            id = request.form['idn']
            global hello
            hello = id
            password = request.form['password']
            print(id, password)
            if id == '1111' and password == '1111':
                return redirect(url_for('admin'))

            sql = f"select * from customer where id='{escape(id)}' and
password='{escape(password)}'"
            stmt = ibm_db.exec_immediate(conn, sql)
            data = ibm_db.fetch_both(stmt)

            if data:
                session["name"] = escape(id)
                session["password"] = escape(password)
                return redirect(url_for("welcome"))

            else:
                flash("Mismatch in credetials", "danger")
        except:
            flash("Error in Insertion operation", "danger")

    return render_template('signinpage.html')

@app.route('/welcome', methods=['POST', 'GET'])
def welcome():
    try:
        id = hello
        sql = "SELECT ID,DATE,TOPIC,SERVICE_TYPE,SERVICE_AGENT,DESCRIPTION,STATUS
FROM ISSUE WHERE CUSTOMER_ID =?"
        agent = []
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, id)
        ibm_db.execute(stmt)
        otpf = ibm_db.fetch_both(stmt)
        while otpf != False:
            agent.append(otpf)
            otpf = ibm_db.fetch_both(stmt)
        sql = "SELECT COUNT(*) FROM ISSUE WHERE CUSTOMER_ID = ?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, id)
        ibm_db.execute(stmt)
        t = ibm_db.fetch_both(stmt)
        return render_template("welcome.html",agent=agent,message=t[0])
    except:

        return render_template("welcome.html")

@app.route('/loginagent', methods=['GET', 'POST'])
def loginagent():
    if request.method == 'POST':
        try:
            global loginagent
            id = request.form['idn']
            loginagent = id
            password = request.form['password']

            sql = f"select * from AGENT where id='{escape(id)}' and
password='{escape(password)}'"
            stmt = ibm_db.exec_immediate(conn, sql)
            data = ibm_db.fetch_both(stmt)

            if data:
                session["name"] = escape(id)
                session["password"] = escape(password)
                return redirect(url_for("agentwelcome"))

            else:
                flash("Mismatch in credetials", "danger")
        except:
            flash("Error in Insertion operation", "danger")

    return render_template("signinpageagent.html")

@app.route('/delete/<ID>')
def delete(ID):
    sql = f"select * from customer where Id='{escape(ID)}'"
    print(sql)
    stmt = ibm_db.exec_immediate(conn, sql)
    student = ibm_db.fetch_row(stmt)
    if student:
        sql = f"delete from customer where id='{escape(ID)}'"
        stmt = ibm_db.exec_immediate(conn, sql)

        flash("Delected Successfully", "success")
        return redirect(url_for("admin"))
```