CUSTOMER CARE REGISTRY

NALAIYA THIRAN PROJECT REPORT 2022

Team ID: PNT2022TMID16191

Submitted by

HARIPRASATH K	**927619BIT4033**
HAJI ALI MEERAN M	**927619BIT4031**
BHARATHKUMAR V	**927619BIT4009**
JIJENDIRAN R	**927619BIT4042**

Content

1. INTRODUCTION

1.1 PROJECT OVERVIEW

Internet is gaining more importance day after day in all life aspects, especially in business and marketing due to the amazing increase in internet users around the world with an estimate of 2.4 billion users in 2012, when comparing this number to the number of internet users in 2000, a growth of 566% can be noticed. This is making internet the fastest media of all time in both growth rate and number of users (internet world stats, 2012). Based on a comprehensive study in 2011, the number of advertisements circulated over the net was more than 3.5 million daily. Internet became one of the most efficient ways to conduct business. In developed and well-developed countries, internet proved to be of much help for local enterprises where it provides great potential for such enterprises to compete worldwide. The main idea is to implement an online system for managing the internet customers and complaint system for customers for raising complaints on the issues related to ISP provider and provide best customer care service for users using this application. There are many Internet security providers in a country that will provide internet services for users on different packages. Basically ISP works on three connections, Dial Up using telephone service, Broad band and wireless connections.

1.2 PURPOSE

The main idea is to implement an online system for managing the internet customers and complaint system for customers for raising complaints on the issues related to ISP provider and provide best customer care service for users using this application. By adding more entries to the data base store, the application can respond to more number of queries from the customers. The importance is given on giving correct reply to the input queries.

2.LITERATURE REVIEW

2.1 EXISTING PROBLEM

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. 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

2.2 References

PAPER 1

TITLE: Automated Ticket Routing System

AUTHOR: Muhammad Zikri bin Zulkifli

In the existing helpdesk system, the tickets were created and assigned to the end user manually. When the ticket is created, it is assigned to the agent manually before they attend that specific ticket. This manual process of ticket creation needs more manpower and takes more time. Instead of putting the effort and time into this task, the ticket creation and assigning can be done automatically when we create an Automated Ticket Routing system. The automated ticket creation and assignment process reduce the time and then the manpower can be used for other purposes. Then, by using the manual ticket creation and assignment process, the distribution of good skill sets, and workload balancing will be missed out. Finding a good skill set and assigning the tickets to the specific skilled agent automatically is considered a good job distribution. Here, the wrong agent represents the sense that the agent doesn’t know well about that particular problem or issue. If the tickets are mistakenly routed, then the resources may get wasted and a lot of time will be spent unnecessarily. Using the location, skill sets, work schedule, and workload balancing, the tickets can be routed automatically to that particular agent perfectly. We can execute the above process perfectly by categorizing the tickets based on the issues.

PAPER 2

TITLE: The impact of the magnitude of service failure and complaint handling on satisfaction and brand credibility in the banking industry

AUTHOR: Chalazal Shams1 · Mohsin Abdur Rahman

The present research aims to investigate the effects of service failure and complaint handling on customer satisfaction with complaint handling which consequently impacts overall satisfaction and brand credibility. To examine the objectives of the present research, the authors deployed a sample of 384 respondents in Persian banks within Iran. Structural equation modelling has been used to analyse the data. The findings suggest that the magnitude of service failure negatively effects customer satisfaction with complaint handling. Complaint handling positively affects customer satisfaction with complaint handling. In addition, the results suggest that customer satisfaction with complaint handling positively influences brand credibility and overall satisfaction. Finally, overall satisfaction positively impacts brand credibility. The results revealed that if the complaint handling occurs instantly at the right time, it would have been a positive influence on customer satisfaction and ultimately develop brand credibility. Therefore, banks can adopt customer relationship management systems and processes which enable quick responses to customer complaints. Bank managers could find the results of the present study useful and beneficial in developing complaint handling efforts and expanding appropriate service recovery and brand credibility strategies. stakeholders involved in the service process. Service recovery follows service failure and helps business to recover the damaged service experience. An effective service recovery strategy transforms the existing service processes to avoid the recurrence of a service failure Service sector adopts more technological advancements to tailor customers dynamic expectations However, business service processes are facing numerous challenges to sustaining brand value

PAPER 3

TITLE: Exploring the influence of the human factor on customer satisfaction in call centres

AUTHOR: Dorian Chicua,∗, Maria del Mar Pàmies

The aim of this study is to explore the human or employee-related factors that shape customer satisfaction in the context of call centres. The literature review draws from a range of disperse disciplines including Service Quality, Human Resource Management and Marketing. The empirical study explores the different variables identified to obtain a nuanced analysis of the employee-related paths that lead to customer satisfaction in call centres. The study employs data from 109 call centres and utilises PLS for our exploratory purposes. Call centre managers should note that investing in HR practices will pay off in terms of improving the elusive phenomenon of customer satisfaction within call centres. The call centre industry is a peculiar service industry, in as much as it is almost entirely based on a voice-to-voice encounter between the employee and the customer, on opposite ends of the telephone line. In general, customers are less satisfied with the service they receive from call centres than from the more traditional brick n' mortar, or face to face service encounters In call centres, employees (call centre operators) are the main connection between the organization and the customer. Employees are often required to undertake many different tasks at the same time. They are expected to display ambidextrous behaviour, being able to accomplish managerial requirements such as: maintaining service quality, including attentiveness, perceptiveness, responsiveness and assurance, satisfy customers, solve problems attend a large number of calls in a short time while ensuring first call resolution.

PAPER 4

TITLE: Analysing and Implementing a System For Reporting, Follow Up and Resolving of Complaints

AUTHOR: Angham AL Abbas, Khadeeja Alzayer

In every aspect of life either it is personal or professional we use internet. It makes life easier, and overcomes unsatisfactory and unacceptable services or issues on various fields. We can use online complaint management system which is considered as an essential part of quality services. Complaints and compliments are valuable sources of information that organizations can use it to improve program delivery and service. “A web system for reporting, follow-up and resolving of complaints” is a web application analysed and developed for managing various complaints in any place such as universities, hospitals shopping centres, damaged roads, unwanted load Shedding or sewerage proble…. etc. This work aims to make complaints easier to be reported, coordinated, monitored, tracked and resolved, and to provide governments with effective tool to keep records of complaint data, to use them for identifying problem areas and to improve services. Today's development cycles for web-applications such as Portals and Marketplaces are short, and getting shorter with continuous improvements and enhancements as new requirements and features become apparent. Therefore, developing “Web Services” using the “Service-Oriented Architecture” paradigm is a widely accepted concept. On the other side, most of user’s complaints are apparent when a system has inappropriate communication between the organizations, their employees and customers (Citizens). Poor communication can result in poor services or products being provided by the organization or Government. Whilst concentrating on the topic of complaint handling, organizations can achieve an efficient success factor by increasing their user satisfaction and their loyalty. Therefore each organization needs to develop its internal and external communication towards its staff and customers to achieve success. Although appropriate communication can reduce user dissatisfaction complaint.

PAPER 5

TITLE: Theory and Practice of Customer-related Improvements

AUTHOR: Daniel Gyllenhammar, et al

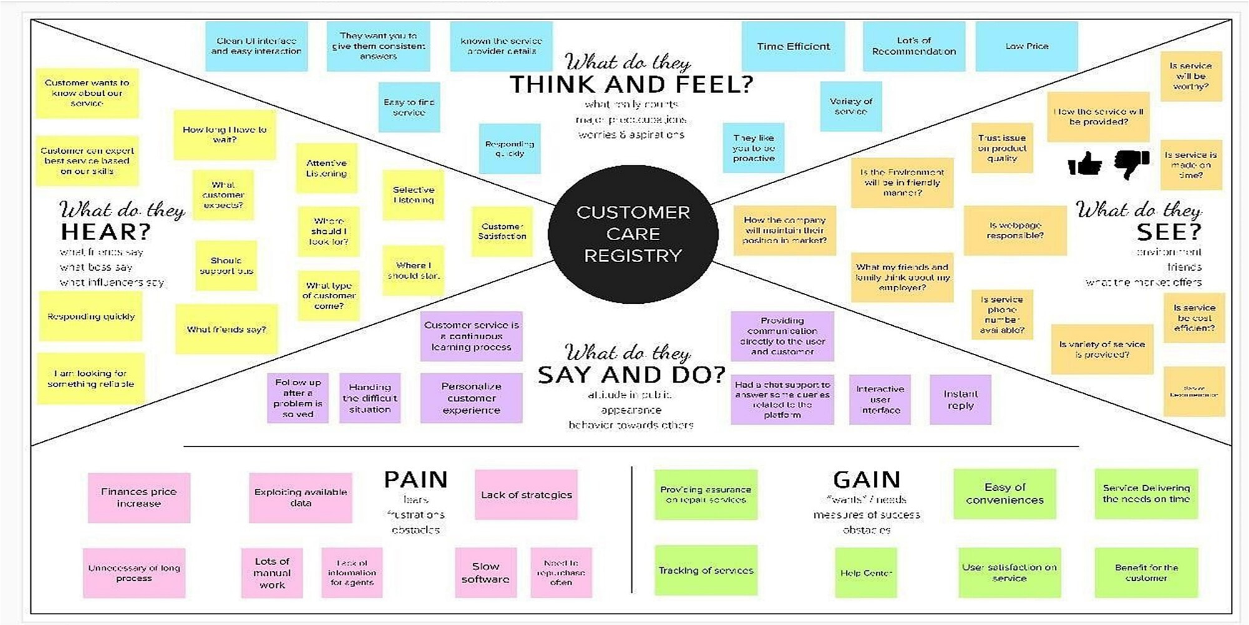
A knowledge-Based helpdesk system is a web-based system that is used to provide technical support to an organization or to management. Then, it acts as a Service Provider to that particular organization. The main objective of this Knowledge based system is to provide technical support to the end users of a particular organization. Using this Knowledge-based Helpdesk system, an organization can improve their end user's performance and make their end users technically well educated. Once the Knowledge based helpdesk system is designed, it is tested on the Information Technology (IT) Center, Engineering Division (BKJ), etc. To have a better support solution for management, the Knowledge-based system is introduced. Usually, the Knowledge-based system consists of questions that are frequently raised by the end users. All the frequent questions are combined into categories and then, it is provided as a solution. The end users can solve their problems manually by themselves just by reading and implementing the solution that is provided. Also, the solutions that are provided by the helpdesk team can be used on future problems too. Hence, it is called a continuity and contingency process.

2.3 PROBLEM STATEMENT DEFINITION

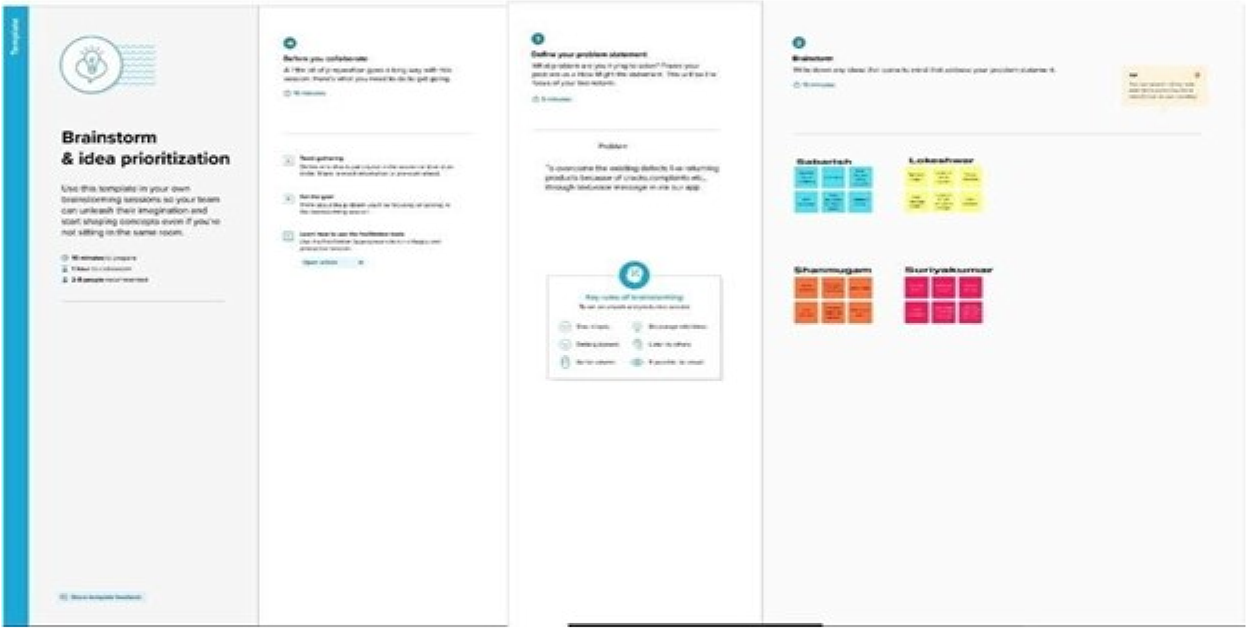
Call Center management is the way in which organizations manage the daily operations of call Center, including forecasting, scheduling, employee training, reporting and all customer interactions. Call Center management can be modernized with workforce optimization (WFO) solutions. A call Center management system refers to a software solution that helps improve customer interactions, service levels, and user experience. Simply put, it's a modern way of managing the day-to-day call Center operations-training, forecasting, reporting, scheduling, and many others. Callers should be able to leave messages in cases where all agents are preoccupied and no one can take the call. A call Center functions through operators, known as agents or sometimes customer representatives, and computerized telephony (CTI).

3.IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

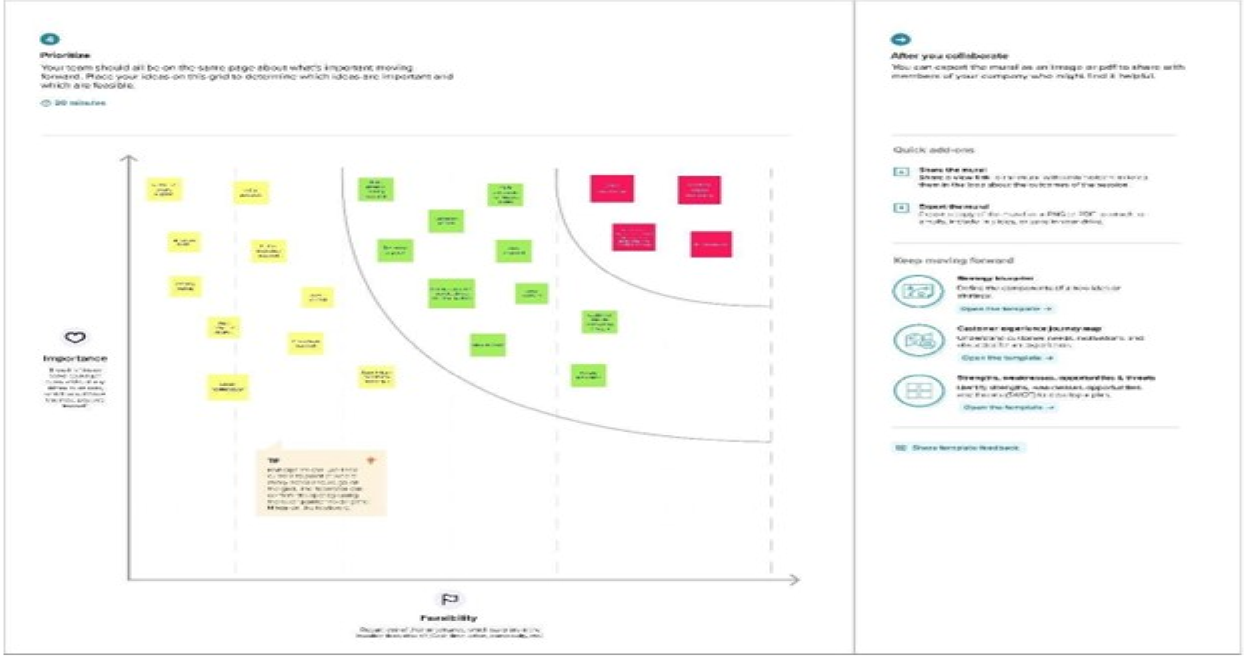

What do they
THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
Clean UI interface and easy interaction
They want you to give them consistent answers
known the service provider details
Time Efficient
Lot's of Recommendation
Low Price
Easy to find service
Variety of service
Responding quickly
They like you to be proactive
What do they
HEAR?
what friends say
what boss say
what influencers say
Customer wants to know about our service
Customer can expect best service based on our skills
How long I have to wait?
What customer expects?
Attentive Listening
Selective Listening
Where should I look for?
Customer Satisfaction
Should support bus
Where I should star.
What type of customer come?
Responding quickly
What friends say?
I am looking for something reliable
CUSTOMER
CARE
REGISTRY
What do they
SEE?
environment
friends
what the market offers
Is service will be worthy?
How the service will be provided?
Trust issue on product quality
Is service is made on time?
Is the Environment will be in friendly manner?
How the company will maintain their position in market?
Is webpage responsible?
What my friends and family think about my employer?
Is service phone number available?
Is service be cost efficient?
Is variety of service is provided?
What do they
SAY AND DO?
attitude in public
appearance
behavior towards others
Customer service is a continuous learning process
Providing communication directly to the user and customer
Follow up after a problem is solved
Handing the difficult situation
Personalize customer experience
Had a chat support to answer some queries related to the platform
Interactive user interface
Instant reply
PAIN
fears
frustrations
obstacles
Finances price increase
Exploiting available data
Lack of strategies
Unnecessary of long process
Lots of manual work
Lack of information for agents
Slow software
Need to repurchase often
GAIN
"wants" / needs
measures of success
obstacles
Providing assurance on repair services
Easy of conveniences
Service Delivering the needs on time
Tracking of services
Help Center
User satisfaction on service
Benefit for the customer

3.2 Ideation & Brainstorming



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.

"To overcome the existing defects like returning products because of cracks,complaints etc., through text/voice message in via our app



Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

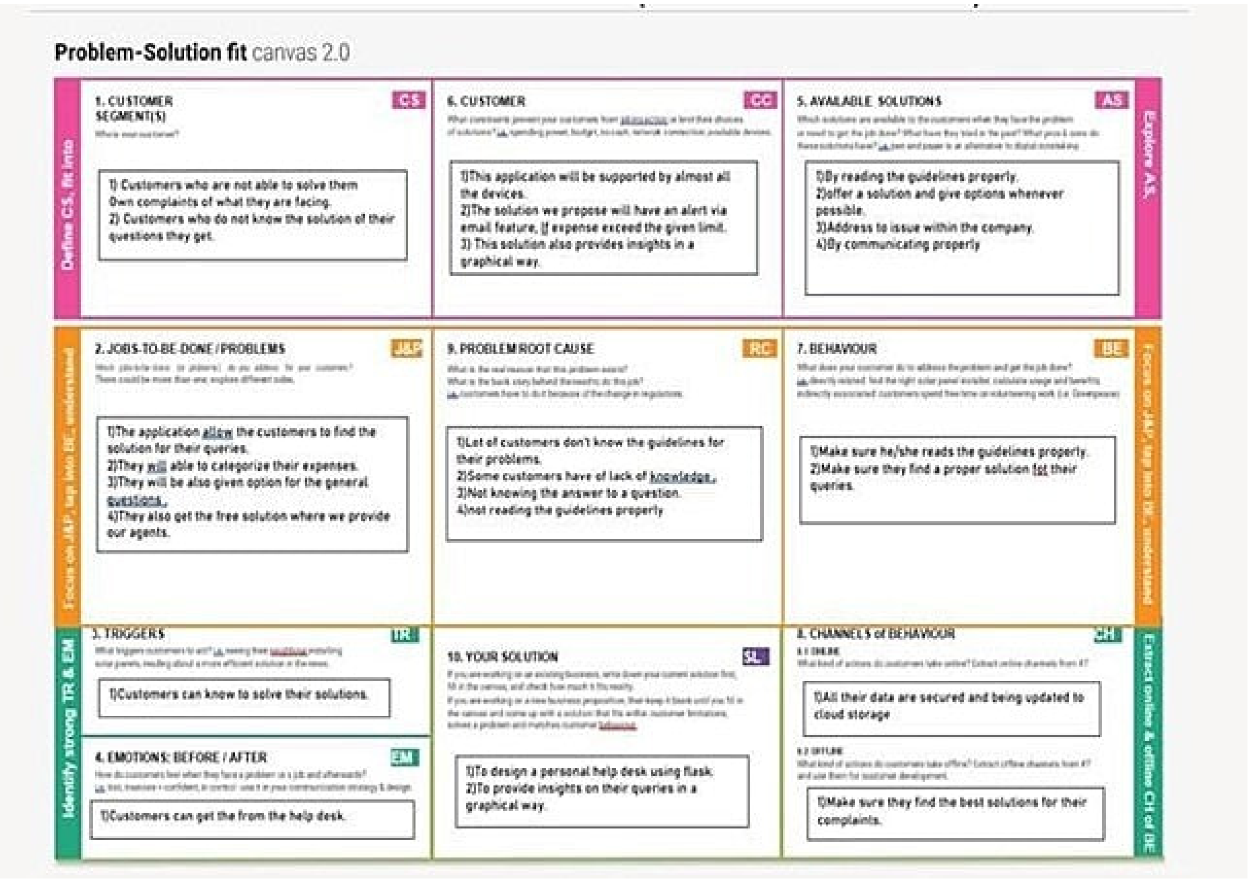
You can export the mural as an image or pdf to share with members of your company who might find it helpful.

3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to besolved)	To solve customer issues using CloudApplication 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)	1. Key Partners - Third-party applications,agents, and customers. 2. Activities - Customer Service,SystemMaintenance. 3. Key Resources -Engineers,Multi-channel. 4. Customer Relationship - 24/7 EmailSupport, Knowledge-based channel. 5. Cost Structure - Cloud Platform, Offices.

6.	Scalability of the Solution	All customers are prioritized based on SLA(Service Level Agreement)-Urgent,Moderate, Low.

3.4 Problem Solution fit



Problem-Solution fit canvas 2.0

1. CUSTOMER SEGMENT(S) — CS

Who is your customer?

1) Customers who are not able to solve them Own complaints of what they are facing.
2) Customers who do not know the solution of their questions they get.

6. CUSTOMER — 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 will be supported by almost all the devices.
2)The solution we propose will have an alert via email feature, If expense exceed the given limit.
3) This solution also provides insights in a graphical way.

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 reading the guidelines properly.
2)offer a solution and give options whenever possible.
3)Address to issue within the company.
4)By communicating properly

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)The application allow the customers to find the solution for their queries.
2)They will able to categorize their expenses.
3)They will be also given option for the general questions.
4)They also get the free solution where we provide our agents.

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)Lot of customers don't know the guidelines for their problems.
2)Some customers have of lack of knowledge.
3)Not knowing the answer to a question.
4)not reading 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)Make sure he/she reads the guidelines properly.
2)Make sure they find a proper solution fot their queries.

3. TRIGGERS — TR

What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

1)Customers can know to solve their solutions.

4. EMOTIONS: BEFORE / AFTER — EM

How do customers feel when they face a problem or a job and afterwards? i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

1)Customers can get the from the help desk.

10. YOUR SOLUTION — SL

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality. If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

1)To design a personal help desk using flask.
2)To provide insights on their queries in a graphical way.

8. CHANNELS of BEHAVIOUR — CH

8.1 ONLINE

What kind of actions do customers take online? Extract online channels from #7

1)All their data are secured and being updated to cloud storage

8.2 OFFLINE

What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

1)Make sure they find the best solutions for their complaints.

Define CS, fit into
Explore AS,
Focus on J&P, tap into BE, understand
Focus on J&P, tap into BE, understand
Identify strong TR & EM
Extract online & offline CH of BE

4.REQUIREMENT ANALYSIS

4.1 Functional requirement

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Admin/Agent Registration	Registration through Gmail.
FR-2	Admin/Agent Confirmation	Confirmation via Email.
FR-3	Customer Query	Access through Email, Chatbot from targeted websites.
FR-4	Customer Confirmation	Confirmation through Ticket ID in Email.
FR-5	Database	Storing the object model.

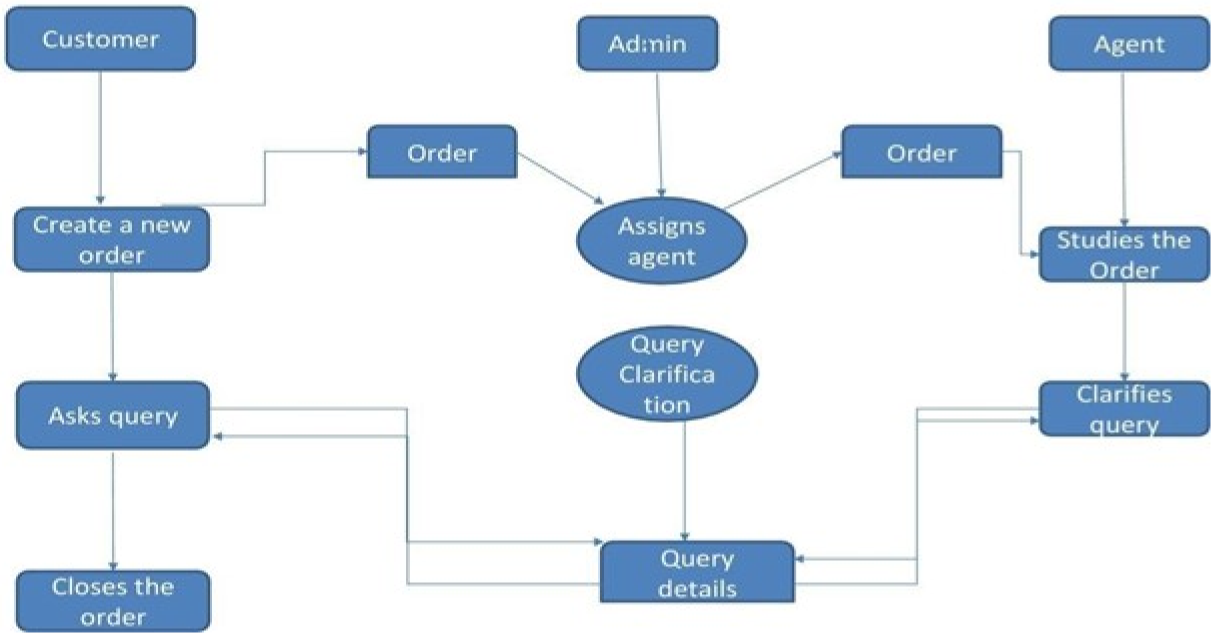
4.2 Non-Functional requirements

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	User Friendly, Easily Accessible.
NFR-2	**Security**	IBM Digital Security Certificate(SSL)for Database.
NFR-3	**Reliability**	Providing Quality Content.
NFR-4	**Performance**	Quick Access, Flexible, and Responsive
NFR-5	**Availability**	24/7 Support
NFR-6	**Scalability**	Good performance for largeCustomers andworkload

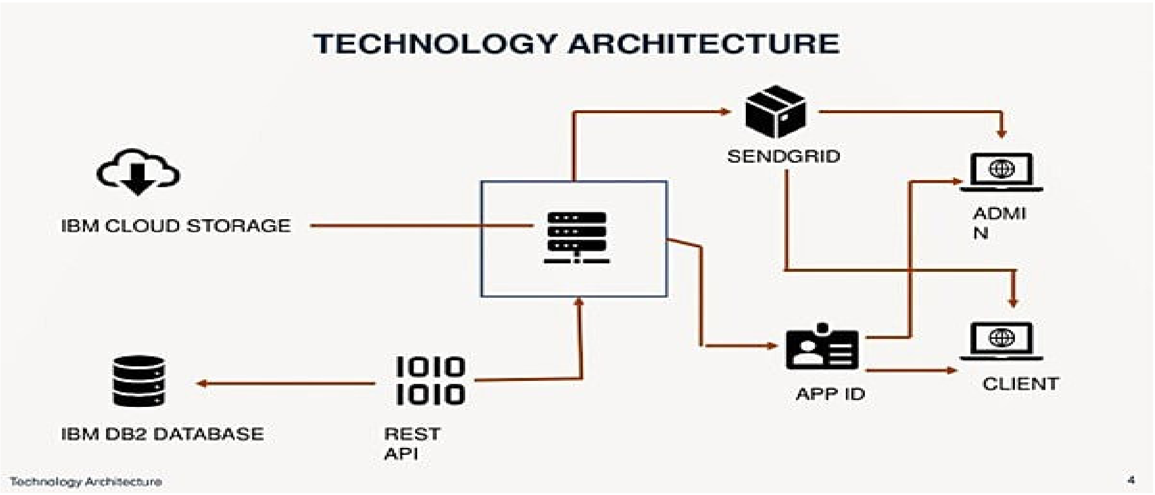
5.Project Design

5.1 Data Flow Diagrams

Data flow diagram for Customer care Registry

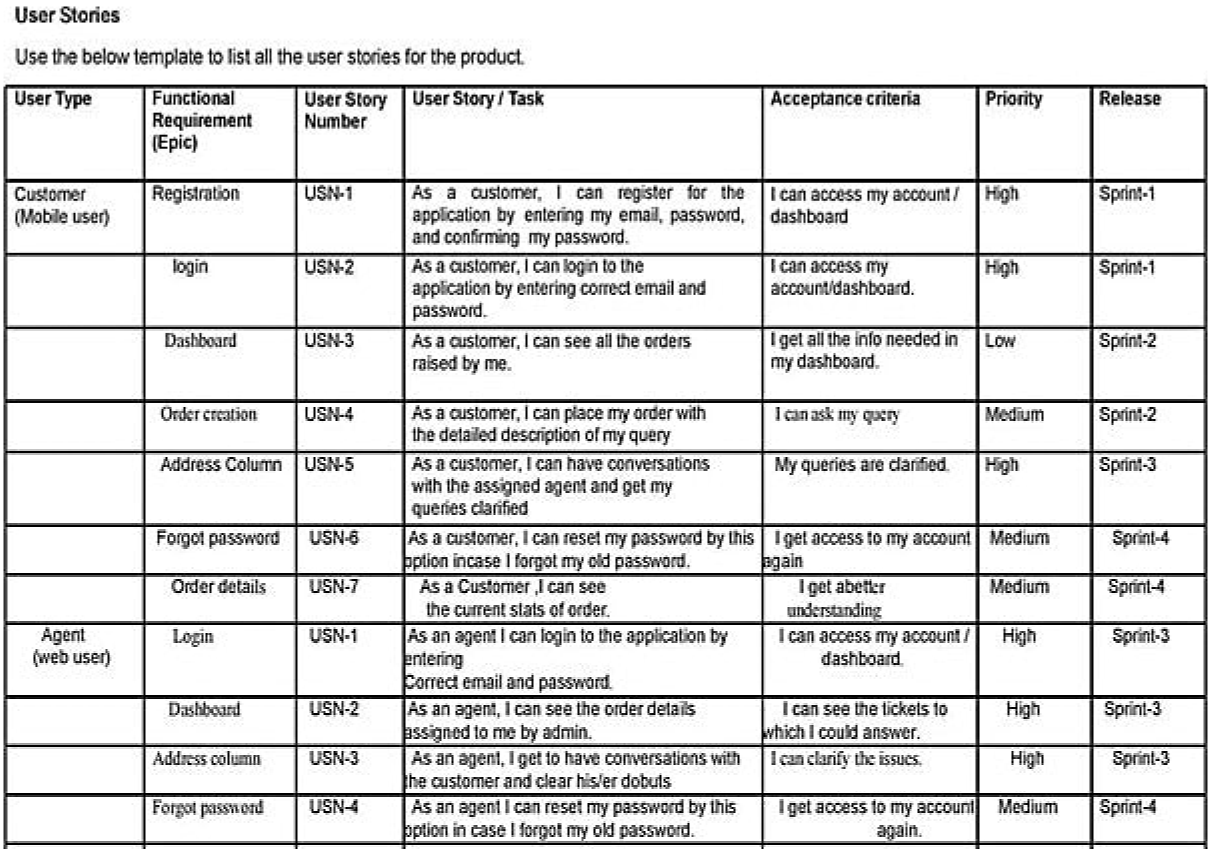


5.2 Solution & Technical Architecture



5.3 User Stories

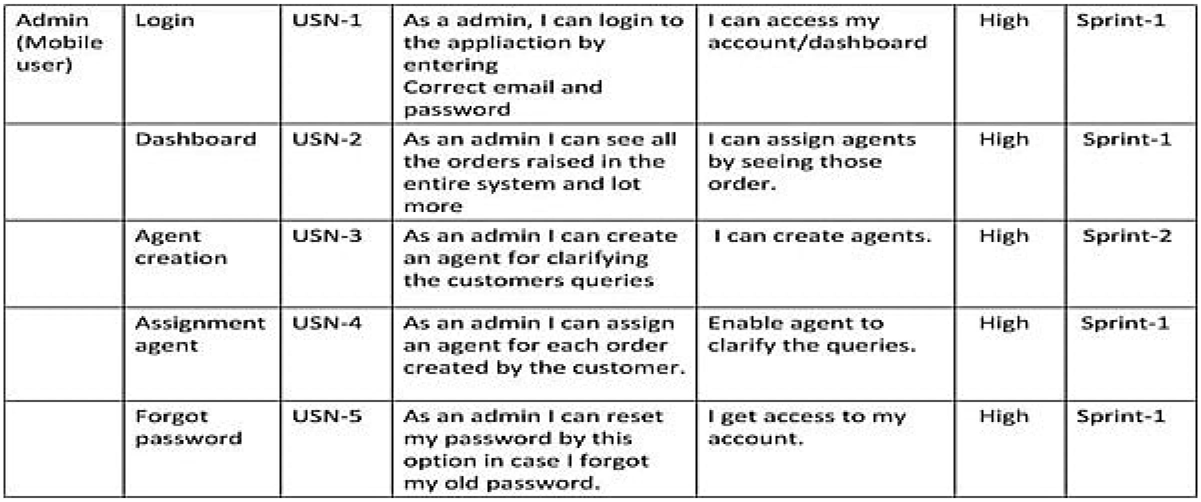
Use the below template to list all the user stories for the product.



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 (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 incase 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 abetter 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/er dobuts	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



Admin (Mobile user)	Login	USN-1	As a admin, I can login to the appliaction 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

TITLE	DESCRIPTION	DATE
LiteratureSurvey &Information Gathering	Literature survey on the selected project & gatheringinformation by referring to technical papers,research publications etc.	09 SEPTEMBER 2022
Prepare EmpathyMap	Prepare Empathy Map Canvasto capture the user Pains & Gains,Prepare list of problemstatements	10 SEPTEMBER 2022
Ideation	List them by organizing the brainstorming session and prioritize the top 3 ideas based on feasibility &importance.	12 SEPTEMBER 2022
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact,scalability of solution, etc.	03 OCTOBER 2022
Problem Solution Fit	Prepare problem - solution fit document.	05 OCTOBER 2022

Solution Architecture	Prepare a solution architecture document.	07 OCTOBER 2022
Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application (entry toexit).	15 OCTOBER 2022
Functional Requirement	Prepare the functional requirement document.	15 OCTOBER 2022
Data Flow Diagrams	Draw the data flow diagrams and submit forreview.	19 OCTOBER 2022
Technology Architecture	Prepare the technology architecture diagram.	14 OCTOBER 2022
Prepare Milestone & ActivityList	Prepare the milestones& activity list of the project.	24 OCTOBER 2022
Project Development - Delivery of Sprint-1, 2, 3& 4	Develop & submit the developed code by testing it.	20 NOVEMBER 2022(PLANNED)

Product Backlog, Sprint Schedule, and Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Customer Panel	USN-1	As a Customer, I can register for the application by entering my email, password, and confirming my password and I will be able to Access my dashboard for creating a Query Order.	2	High	Hariprasath K Haji Ali meeran M Mahabharata V Jidendiran
Sprint-1	Admin Panel	USN-2	As an admin, I can Login to the Application by entering correct login credentials and I will be able to Access My dashboard to create Agents and Assign an Agent to a Query Order.	2	High	Hariprasath K Haji Ali meeran M Bharatkumar V Jidendiran R
Sprint-2	Agent Panel	USN-3	As an agent, I can Login to the Application by entering correct login credentials and I will be able to Access my Dashboard to check the Query Order and I can Clarify the Issues.	2	High	Hariprasath K Haji ali meeran M Bharatkumar V Jidendiran R
Sprint-3	Chat Bot	USN-4	The Customer can directly Interact to the Chatbot regarding the services offered by the Web Portal and get recommendations based on information provided by them.	2	Medium	Hariprasath K Haji ali meeran M Bharatkumar V Jidendiran R
Sprint-4	Final Delivery	USN-5	Container of applications using docker kubernetes and deployment the application. Create the documentation and final	2	High	Hariprasath K Haji ali meeran M Bharatkumar V Jidendiran R

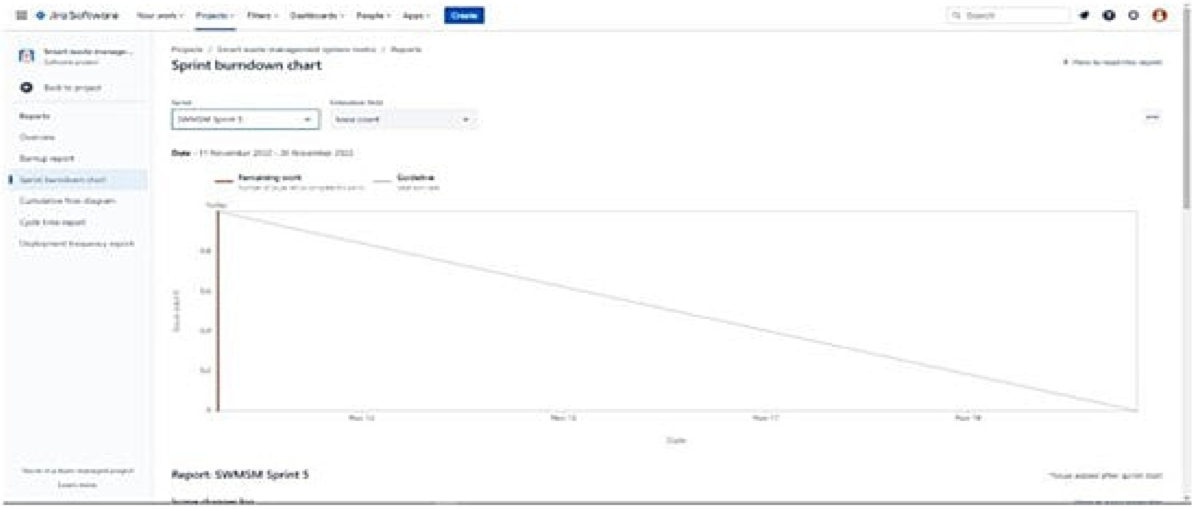
			submit the application			

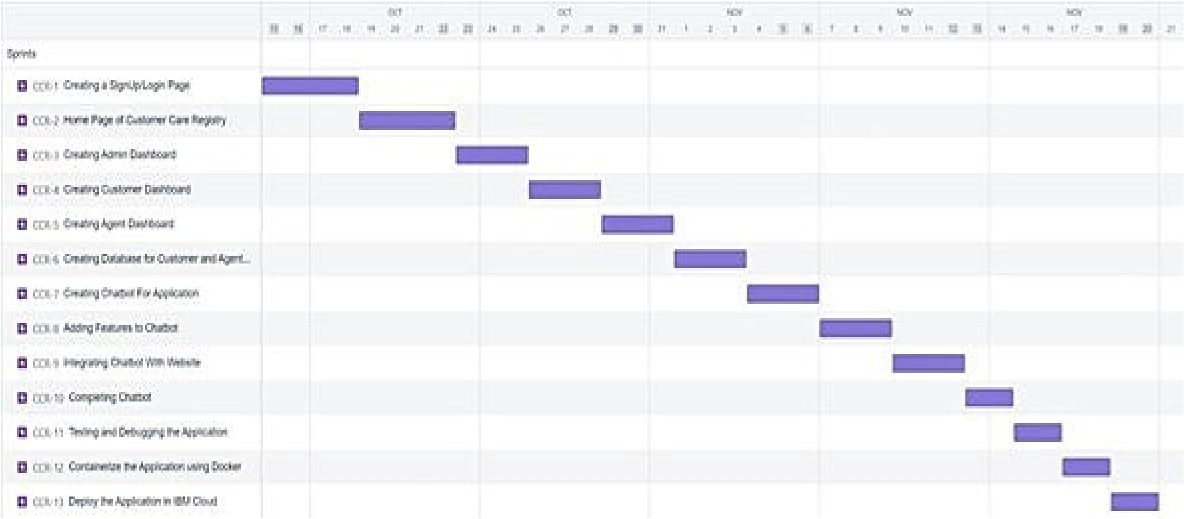
6.2 Sprint Delivery Schedule

Sprint	Total StoryPoints	Duration	Sprint Start Date	Sprint EndDate (Planed)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	7 Days	24 Oct 2022	30 Oct 2022		30 Oct 2022
Sprint-2	20	7 Days	31 Oct 2022	06 Nov 2022		06 Nov 2022
Sprint-3	20	8 Days	07 Nov 2022	14 Nov 2022		14 Nov 2022
Sprint-4	20	7 Days	14 Nov 2022	21 Nov 2022		21 Nov 2022

6.3 Reports from JIRA

BURNDOWN CHART





Sprints

- CCR-1 Creating a SignUp/Login Page
- CCR-2 Home Page of Customer Care Registry
- CCR-3 Creating Admin Dashboard
- CCR-4 Creating Customer Dashboard
- CCR-5 Creating Agent Dashboard
- CCR-6 Creating Database for Customer and Agent...
- CCR-7 Creating Chatbot For Application
- CCR-8 Adding Features to Chatbot
- CCR-9 Integrating Chatbot With Website
- CCR-10 Completing Chatbot
- CCR-11 Testing and Debugging the Application
- CCR-12 Containerize the Application using Docker
- CCR-13 Deploy the Application in IBM Cloud

7. CODING & SOLUTIONING (Explain the features added in the project along with code)

College graduates with prior programming expertise or technical degrees are recruited and transitioned into professional positions with Alabama firms and organisations through the highly competitive Coding Solutions job accelerator and talent refinement programme at no cost to the graduates. We provide a pool of varied, well-trained, techs-savvy individuals that wants to launch and advance their career in Alabama. The mission of veteran- and woman-owned Coding Solutions is to mobilise the next generation of IT talent and provide them the tools and resources they require to make your business successful. Innovative talent is necessary for innovative technologies. We wish to provide Coding Solutions prospects to assist you expand your Alabama team. Our applicants are swiftly hired at the top of the list by growing businesses for lucrative, long-term positions.

7.1 Feature 1

7 Main types of customer needs:

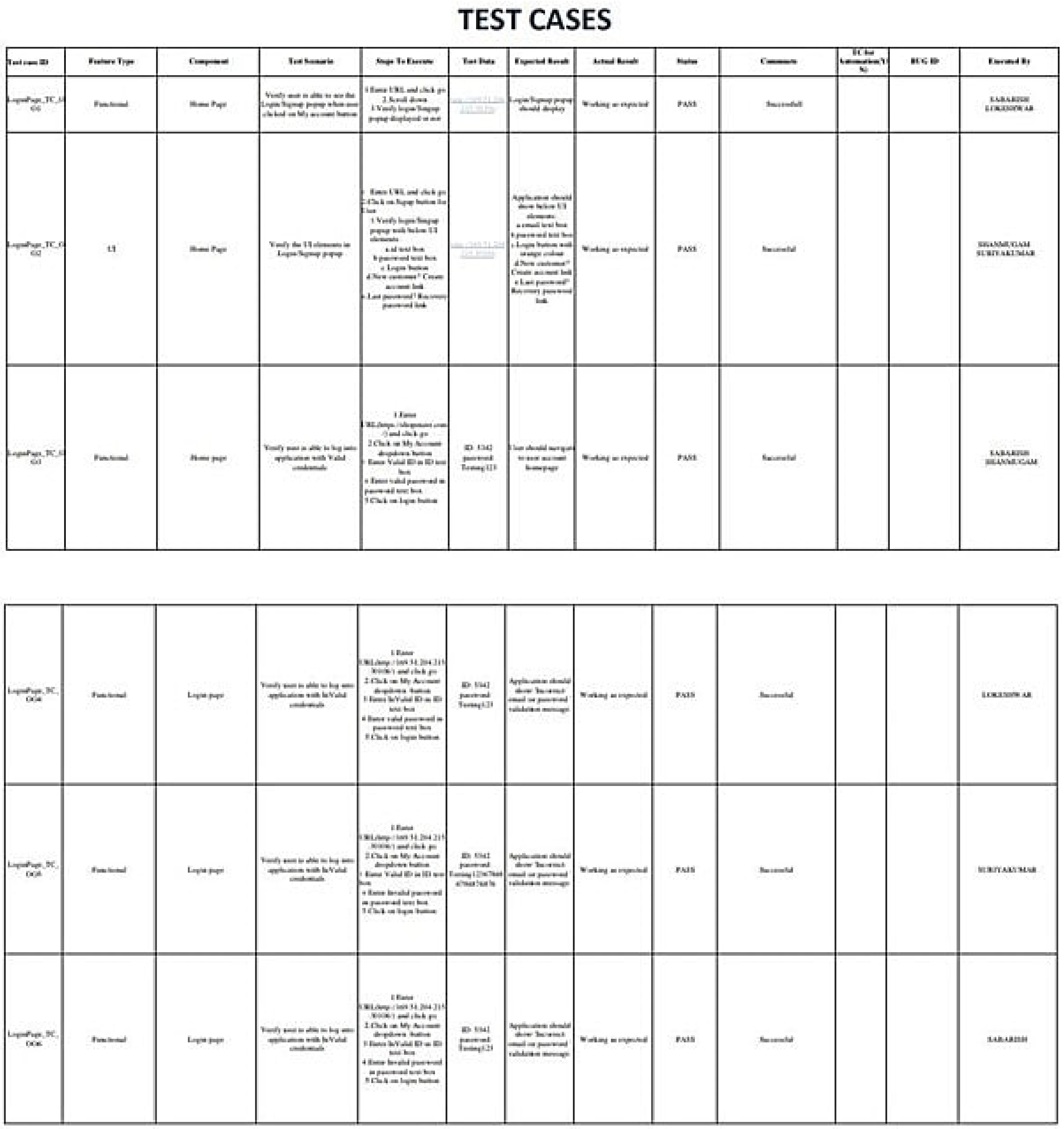
- Friend lines
- Empathy
- Fairness
- Control
- Alternatives
- Information

7.2 Feature

- Complaint Tracking
- Email Alert
- 24/7 Montoring

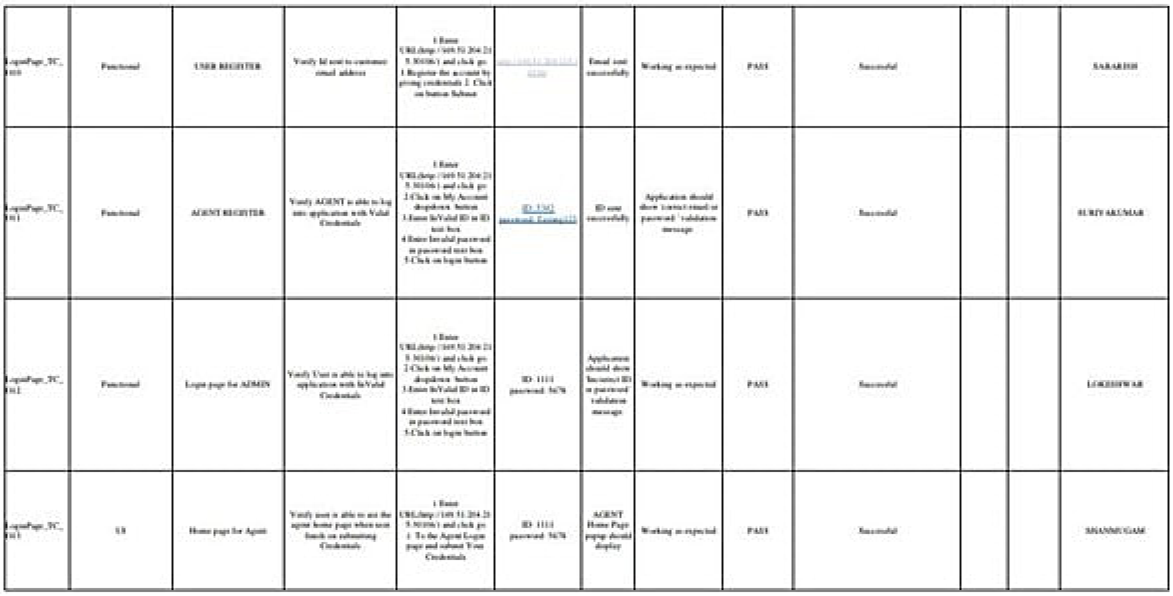
8. TESTING

8.1 Test Cases

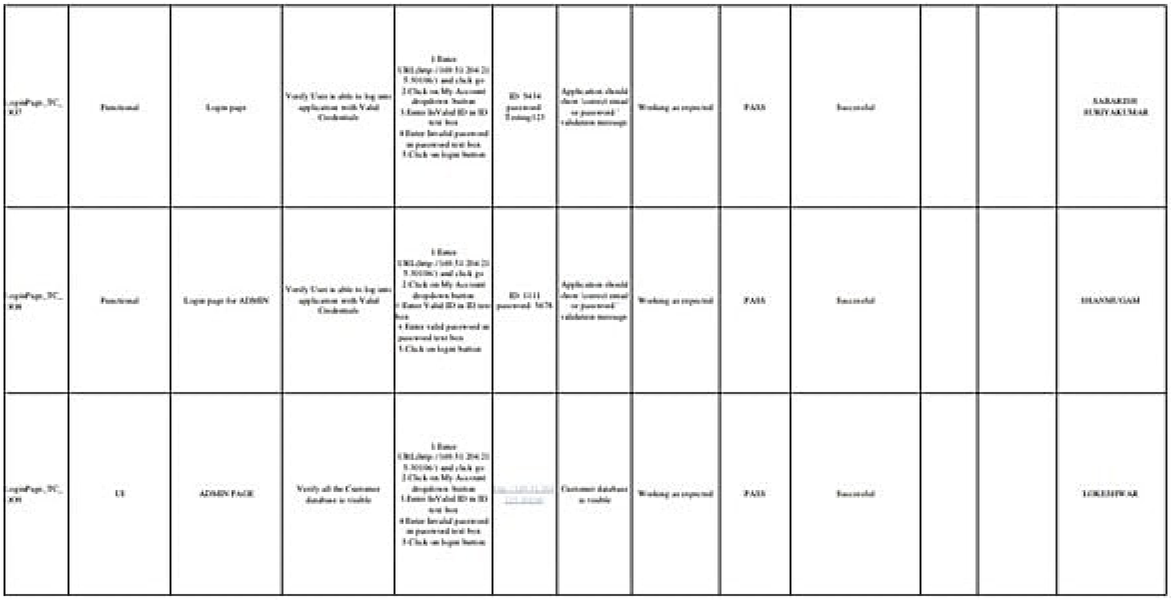


TEST CASES

Test case ID	Feature Type	Component	Test Scenario	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	BUG ID	Executed By
LoginPage_TC_001	Functional	Home Page	Verify user is able to see the Login/Signup popup when user clicked on My account button	1.Enter URL and click go 2.Scroll down 3.Verify login/Signup popup displayed or not	[illegible]	Login/Signup popup should display	Working as expected	PASS	Successful			SABARISH LOKESHWAR
LoginPage_TC_002	UI	Home Page	Verify the UI elements in Login/Signup popup	1. Enter URL and click go 2.Click on Signup button for User 3.Verify login/Signup popup with below UI elements: a.id text box b.password text box c. Login button d.New customer? Create account link e.Lost password? Recovery password link	[illegible]	Application should show below UI elements: a.email text box b.password text box c.Login button with orange colour d.New customer? Create account link e.Lost password? Recovery password link	Working as expected	PASS	Successful			SHANMUGAM SURIYAKUMAR
LoginPage_TC_003	Functional	Home page	Verify user is able to log into application with Valid credentials	1.Enter URL(https://shopmart.com/) and click go 2.Click on My Account dropdown button 3.Enter Valid ID in ID text box 4.Enter valid password in password text box 5.Click on login button	ID: 5342 password: Testing123	User should navigate to user account homepage	Working as expected	PASS	Successful			SABARISH SHANMUGAM
LoginPage_TC_004	Functional	Login page	Verify user is able to log into application with InValid credentials	1.Enter URL(http://169.51.204.215:30036/) and click go 2.Click on My Account dropdown button 3.Enter InValid ID in ID text box 4.Enter valid password in password text box 5.Click on login button	ID: 5342 password: Testing123	Application should show 'Incorrect email or password' validation message	Working as expected	PASS	Successful			LOKESHWAR
LoginPage_TC_005	Functional	Login page	Verify user is able to log into application with InValid credentials	1.Enter URL(http://169.51.204.215:30036/) and click go 2.Click on My Account dropdown button 3.Enter Valid ID in ID text box 4.Enter Invalid password in password text box 5.Click on login button	ID: 5342 password: Testing[illegible]	Application should show 'Incorrect email or password' validation message	Working as expected	PASS	Successful			SURIYAKUMAR
LoginPage_TC_006	Functional	Login page	Verify user is able to log into application with InValid credentials	1.Enter URL(http://169.51.204.215:30036/) and click go 2.Click on My Account dropdown button 3.Enter InValid ID in ID text box 4.Enter Invalid password in password text box 5.Click on login button	ID: 5342 password: Testing123	Application should show 'Incorrect email or password' validation message	Working as expected	PASS	Successful			SABARISH



LoginPage_TC_ 010	Functional	USER REGISTER	Verify Id sent to customer email address	1.Enter URL(http://169.51.204.215:30106/) and click go 1.Register the account by giving credentials 2. Click on button Submit	http://169.51.204.215:30106/	Email sent successfully	Working as expected	PASS	Successful			SARAIESH
LoginPage_TC_ 011	Functional	AGENT REGISTER	Verify AGENT is able to log into application with Valid Credentials	1.Enter URL(http://169.51.204.215:30106/) and click go 2.Click on My Account dropdown button 3.Enter InValid ID in ID text box 4.Enter Invalid password in password text box 5.Click on login button	ID: 5342 password: Testing123	ID sent successfully	Application should show 'correct email or password ' validation message	PASS	Successful			SURIYAKUMAR
LoginPage_TC_ 012	Functional	Login page for ADMIN	Verify User is able to log into application with InValid Credentials	1.Enter URL(http://169.51.204.215:30106/) and click go 2.Click on My Account dropdown button 3.Enter InValid ID in ID text box 4.Enter Invalid password in password text box 5.Click on login button	ID: 1111 password: 5678	Application should show 'Incorrect ID or password' validation message	Working as expected	PASS	Successful			LOKESHWAR
LoginPage_TC_ 013	UI	Home page for Agent	Verify user is able to see the agent home page when user finish on submitting Credentials	1.Enter URL(http://169.51.204.215:30106/) and click go 1. To the Agent Login page and submit Your Credentials	ID: 1111 password: 5678	AGENT Home Page popup should display	Working as expected	PASS	Successful			SHANMUGAM



LoginPage_TC_ 007	Functional	Login page	Verify User is able to log into application with Valid Credentials	1.Enter URL(http://169.51.204.215:30106/) and click go 2.Click on My Account dropdown Button 3.Enter InValid ID in ID text box 4.Enter Invalid password in password text box 5.Click on login button	ID: 5416 password: Testing123	Application should show 'correct email or password ' validation message	Working as expected	PASS	Successful			SARAIESH SURIYAKUMAR
LoginPage_TC_ 008	Functional	Login page for ADMIN	Verify User is able to log into application with Valid Credentials	1.Enter URL(http://169.51.204.215:30106/) and click go 2.Click on My Account dropdown button 3. Enter Valid ID in ID text box 4. Enter valid password in password text box 5.Click on login button	ID: 1111 password: 5678	Application should show 'correct email or password ' validation message	Working as expected	PASS	Successful			SHANMUGAM
LoginPage_TC_ 009	UI	ADMIN PAGE	Verify all the Customer database is visible	1.Enter URL(http://169.51.204.215:30106/) and click go 2.Click on My Account dropdown button 3.Enter InValid ID in ID text box 4.Enter Invalid password in password text box 5.Click on login button	http://169.51.204.215:30106/	Customer database is visible	Working as expected	PASS	Successful			LOKESHWAR

8.2 User Acceptance Testing

Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [CUSTOMER CARE REGISTRY] project at the time of the release to User Acceptance Testing (UAT).

Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved.

Resolution	Severity1	Severity2	Severity3	Severity4	Subtotal
By Design	10	3	1	2	17
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	40
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	13	12	25	78

Test Case Analysis

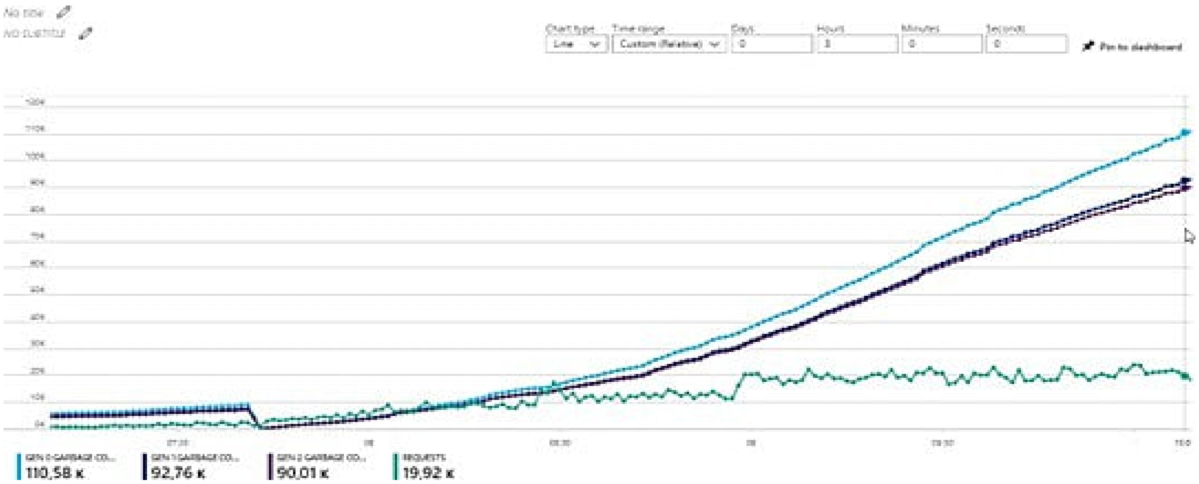
This report shows the number of test cases that have passed, failed, and untested

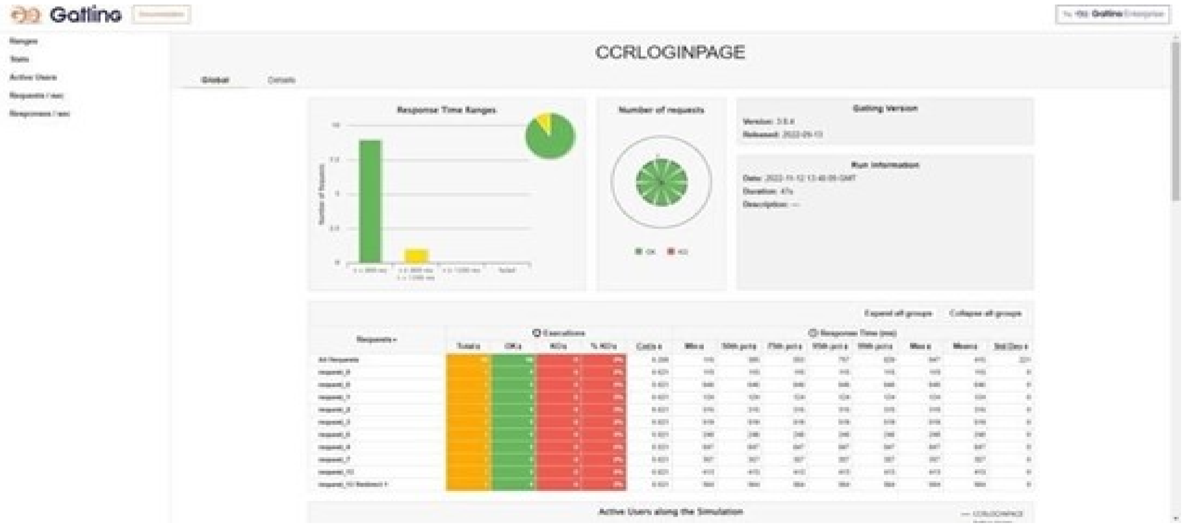
Section	Total Cases	Not Tested	Fail	Pass
Print Engine	10	0	0	10
Client Application	50	0	0	50
Security	1	0	0	1
Outsource Shipping	3	0	0	3
Exception Reporting	8	0	0	8
Final ReportOutput	4	0	0	4

Version Control	2	0	0	2

9. Results

9.1 Performance Metrics





CCRLOGINPAGE

Gatling Version

Version: 3.8.4

Released: 2022-09-13

Run Information

Date: 2022-11-12 13:40:09 GMT

Duration: 47s

Description: —

Expand all groups

Collapse all groups

Requests ▴	Executions					Response Time (ms)							
	Total ▴	OK ▴	KO ▴	% KO ▴	Cnt/s ▴	Min ▴	50th pct ▴	75th pct ▴	95th pct ▴	99th pct ▴	Max ▴	Mean ▴	Std Dev ▴
All Requests	10	10	0	0%	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
request_0	1	1	0	0%	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0
request_6	1	1	0	0%	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0
request_1	1	1	0	0%	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0
request_2	1	1	0	0%	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0
request_3	1	1	0	0%	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0
request_5	1	1	0	0%	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0
request_4	1	1	0	0%	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0
request_7	1	1	0	0%	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0
request_10	1	1	0	0%	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0
request_10 Redirect 1	1	1	0	0%	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0

10. 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.
- It can be particularly difficult for small businesses to cope with these costs

10.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.

11.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. Through this, quarries that will result in a call for assistance can be predicted easily. Implementing ML in customer service trends will give you a significant difference in business growth.

12. Appendix

```
from flask import Flask, render_template, flash, request, session,send_file

from flask import render_template, redirect, url_for, request

import ibm_db

import pandas

import ibm_db_dbi

from sql-alchemy import create_engine



engine = create_engine('sqlite://', echo = False)

dsn_hostname="54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud"

dsn_uid = "wqx88646"

dsn_pwd = "s52Vn19h2F29rIi4"



dsn_driver = "{IBM DB2 ODBC DRIVER}"

dsn_database = "BLUDB"

dsn_port = "32733"

dsn_protocol = "TCPIP"

dsn_security = "SSL"

dsn = (
  "DRIVER={0};"
  "DATABASE={1};"
  "HOSTNAME={2};"
  "PORT={3};"
  "PROTOCOL={4};"
  "UID={5};"
  "PWD={6};"
    "SECURITY={7};").format(dsn_driver, dsn_database, dsn_hostname, dsn_port, dsn_protocol, dsn_uid, dsn_pwd,dsn_security)
try:
  conn = ibm_db.connect(dsn, "", "")
      print ("Connected to database: ", dsn_database, "as user: ", dsn_uid, "on host: ", dsn_hostname)
except:
  print ("Unable to connect: ", ibm_db.conn_errormsg() )


app = Flask(__name__)
app.config['DEBUG']
app.config['SECRET_KEY'] = '7d441f27d441f27567d441f2b6176a'
@app.route("/")
```

```
def homepage():
 return render_template('index.html')
@app.route("/AdminLogin")
def AdminLogin():

    return render_template('AdminLogin.html')
@app.route("/UserLogin")
def UserLogin():
    return render_template('UserLogin.html')
@app.route("/NewUser")
def NewUser():
    return render_template('NewUser.html')
@app.route("/NewComplaint")
def NewComplaint():
    user = session['uname']
    return render_template('NewComplaint.html',uname=user)
@app.route("/NewAgent")
def NewAgent():
    conn = ibm_db.connect(dsn, "", "")
    pd_conn = ibm_db_dbi.Connection(conn)
    selectQuery = "SELECT * FROM agenttb where  "
    dataframe = pandas.read_sql(selectQuery, pd_conn)
    dataframe.to_sql('booktb1', con=engine, if_exists='append')
    data = engine.execute("SELECT * FROM booktb1").fetchall()

    return render_template('NewAgent.html',data=data)
@app.route("/AdminHome")
def AdminHome():
    conn = ibm_db.connect(dsn, "", "")
    pd_conn = ibm_db_dbi.Connection(conn)
   selectQuery = "SELECT * from regtb "
    dataframe = pandas.read_sql(selectQuery, pd_conn)
 dataframe.to_sql('Employee_Data',
                     con=engine,
                     if_exists='append')
# run a sql query
    data = engine.execute("SELECT * FROM Employee_Data").fetchall()
    return render_template('AdminHome.html',data=data)
@app.route("/UserHome")
def UserHome():
    user = session['uname']


    conn = ibm_db.connect(dsn, "", "")

    pd_conn = ibm_db_dbi.Connection(conn)

    selectQuery = "SELECT * FROM regtb where  UserName= '" + user + "' "

    dataframe = pandas.read_sql(selectQuery, pd_conn)

    dataframe.to_sql('booktb1', con=engine, if_exists='append')

    data = engine.execute("SELECT * FROM booktb1").fetchall()

    return render_template('UserHome.html',data=data)

@app.route("/UserComplaint")

def UserComplaint():

    user = session['uname']

    conn = ibm_db.connect(dsn, "", "")

    pd_conn = ibm_db_dbi.Connection(conn)

    selectQuery = "SELECT * FROM booktb where  UserName= '" + user + "' "

    dataframe = pandas.read_sql(selectQuery, pd_conn)

    dataframe.to_sql('booktb1', con=engine, if_exists='append')

    data = engine.execute("SELECT * FROM booktb1").fetchall()

return render_template('UserComplaint.html',data=data)

@app.route("/AdminComplaintInfo")
```

```
def AdminComplaintInfo():
    conn = ibm_db.connect(dsn, "", "")
    pd_conn = ibm_db_dbi.Connection(conn)
    selectQuery = "SELECT * FROM booktb    "
    dataframe = pandas.read_sql(selectQuery, pd_conn)
    dataframe.to_sql('booktb1', con=engine, if_exists='append')
    data = engine.execute("SELECT * FROM booktb1").fetchall()

    return render_template('AdminComplaintInfo.html',data=data)
@app.route("/adminlogin", methods=['GET', 'POST'])
def adminlogin():
    error = None
    if request.method == 'POST':
       if request.form['uname'] == 'admin' or request.form['password'] == 'admin':
         conn = ibm_db.connect(dsn, "", "")
           pd_conn = ibm_db_dbi.Connection(conn)
           selectQuery = "SELECT * from regtb "
           dataframe = pandas.read_sql(selectQuery, pd_conn)
           dataframe.to_sql('Employee_Data',
                            con=engine,
                            if_exists='append')
           # run a sql query
           data = engine.execute("SELECT * FROM Employee_Data").fetchall()
           return render_template('AdminHome.html' , data=data)
       else:
        return render_template('index.html', error=error)
@app.route("/userlogin", methods=['GET', 'POST'])
def userlogin():
    if request.method == 'POST':
        username = request.form['uname']
        password = request.form['password']
        session['uname'] = request.form['uname']
        conn = ibm_db.connect(dsn, "", "")
        pd_conn = ibm_db_dbi.Connection(conn)
         selectQuery = "SELECT * from regtb where UserName='" + username + "' and password='" + password + "'"
        dataframe = pandas.read_sql(selectQuery, pd_conn)
        if dataframe.empty:
            data1 = 'Username or Password is wrong'
            return render_template('goback.html', data=data1)
        else:

            print("Login")

             selectQuery = "SELECT * from regtb where UserName='" + username + "' and password='"
+ password + "'"

            dataframe = pandas.read_sql(selectQuery, pd_conn)

            dataframe.to_sql('Employee_Data',

                             con=engine,

                             if_exists='append')

            # run a sql query

            data = engine.execute("SELECT * FROM Employee_Data").fetchall()


            return render_template('UserHome.html', data=data )

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

def newuser():

    if request.method == 'POST':

        name1 = request.form['name']

        gender1 = request.form['gender']

        Age = request.form['age']

        email = request.form['email']

        pnumber = request.form['phone']
        address = request.form['address']

        uname = request.form['uname']

        password = request.form['psw']

        conn = ibm_db.connect(dsn, "", "")

        insertQuery = "INSERT INTO regtb VALUES ('" + name1 + "','" + gender1 + "','" + Age + "','" + email + "','" + pnumber + "','" + address + "','" + uname + "','" + password + "')"

        insert_table = ibm_db.exec_immediate(conn, insertQuery)

        print(insert_table)

    return render_template('UserLogin.html')

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

def newage():

    if request.method == 'POST':

        name1 = request.form['name']

        gender1 = request.form['gender']

        Age = request.form['age']

        email = request.form['email']

        pnumber = request.form['phone']

        address = request.form['address']

        uname = request.form['uname']

        conn = ibm_db.connect(dsn, "", "")
        pd_conn = ibm_db_dbi.Connection(conn)
        insertQuery = "INSERT INTO agenttb VALUES ('" + name1 + "','" + gender1 + "','" + Age + "','" + email + "','" + pnumber + "','" + address + "','" + uname + "')"
        insert_table = ibm_db.exec_immediate(conn, insertQuery)
        print(insert_table)
        selectQuery = "SELECT * FROM agenttb  "
        dataframe = pandas.read_sql(selectQuery, pd_conn)
        dataframe.to_sql('booktb1', con=engine, if_exists='append')
        data = engine.execute("SELECT * FROM booktb1").fetchall()
    return render_template('NewAgent.html',data=data)
@app.route("/newcom", methods=['GET', 'POST'])
def newcom():
    if request.method == 'POST':

        name = request.form['name']
        com = request.form['com']
        uname = session['uname']
        conn = ibm_db.connect(dsn, "", "")
        pd_conn = ibm_db_dbi.Connection(conn)
        selectQuery = "SELECT  *  FROM  booktb"
        dataframe = pandas.read_sql(selectQuery, pd_conn)

        dataframe.to_sql('booktb', con=engine, if_exists='append')

        data2 = engine.execute("SELECT * FROM booktb").fetchall()

        count = 0

        for item in data2:

            count += 1

        Bookingid = "COMID00" + str(count)

        insertQuery = "INSERT INTO booktb VALUES ('" + Bookingid + "','" + uname + "','" + com + "','','')"

        insert_table = ibm_db.exec_immediate(conn, insertQuery)

        print(insert_table)

        selectQuery = "SELECT * FROM booktb where  UserName= '" + uname + "' "

        dataframe = pandas.read_sql(selectQuery, pd_conn)

        dataframe.to_sql('booktb1', con=engine, if_exists='append')

        data = engine.execute("SELECT * FROM booktb1").fetchall()

        return render_template('UserComplaint.html', data=data)

@app.route("/AgentAssign", methods=['GET'])

def AgentAssign():

    cid = request.args.get('id')

    session['cid'] = cid

    conn = ibm_db.connect(dsn, "", "")
    pd_conn = ibm_db_dbi.Connection(conn)

    selectQuery = "SELECT * FROM agenttb  "

    dataframe = pandas.read_sql(selectQuery, pd_conn)

    dataframe.to_sql('booktb1', con=engine, if_exists='append')

    data = engine.execute("SELECT * FROM booktb1").fetchall()

    return render_template('AgentAssign.html',data=data)

@app.route("/Action", methods=['GET'])

def Action():

    cid = request.args.get('id')

    session['cid'] = cid

    return render_template('Action.html')

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

def ass():

    agid = request.form['agid']

    cid = session['cid']


    uname = session['uname']

    conn = ibm_db.connect(dsn, "", "")

    pd_conn = ibm_db_dbi.Connection(conn)

    selectQuery1 = "SELECT  *  FROM  regtb where  UserName='" + uname + "'"
    dataframe = pandas.read_sql(selectQuery1, pd_conn)

    dataframe.to_sql('regtb', con=engine, if_exists='append')

    data1 = engine.execute("SELECT * FROM regtb").fetchall()

    for item1 in data1:

        Mobile = item1[5]

        Email = item1[4]

        sendmsg(Email,"Assign Agent id"+agid)

    insertQuery = "update booktb set AgentName='"+ agid +"' where ComplaintId='"+ cid +"' "

    insert_table = ibm_db.exec_immediate(conn, insertQuery)

    alert = 'Agent Assign Send Notication'

    return render_template('goback.html', data=alert)

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

def acc():

    com = request.form['com']

    cid = session['cid']

    uname = session['uname']


    conn = ibm_db.connect(dsn, "", "")

    pd_conn = ibm_db_dbi.Connection(conn)

    selectQuery1 = "SELECT  *  FROM  regtb where  UserName='" + uname + "'"
    dataframe = pandas.read_sql(selectQuery1, pd_conn)

    dataframe.to_sql('regtb', con=engine, if_exists='append')

    data1 = engine.execute("SELECT * FROM regtb").fetchall()

    for item1 in data1:

        Mobile = item1[5]

        Email = item1[4]

        sendmsg(Email,"Action Information "+com)

    insertQuery = "update booktb set ACTIONINFO='"+ com +"' where ComplaintId='"+ cid +"' "

    insert_table = ibm_db.exec_immediate(conn, insertQuery)

    alert = 'Action Info Saved Send Notication'

    return render_template('goback.html', data=alert)

def sendmsg(Mailid,message):

    import smtplib

    from email.mime.multipart import MIMEMultipart

    from email.mime.text import MIMEText

    from email.mime.base import MIMEBase

    from email import encoders


    fromaddr = "hajiaslan111apnt@gmail.com"

    toaddr = Mailid
```

```
# instance of MIMEMultipart
msg = MIMEMultipart()
# storing the senders email address
msg['From'] = fromaddr
# storing the receivers email address
msg['To'] = toaddr
# storing the subject
msg['Subject'] = "Alert"
# string to store the body of the mail
body = message
# attach the body with the msg instance
msg.attach(MIMEText(body, 'plain'))
# creates SMTP session
s = smtplib.SMTP('smtp.gmail.com', 587)
# start TLS for security
s.starttls()
# Authentication
s.login(fromaddr, "aohvmkouuitfcdis")


# Converts the Multipart msg into a string
```

```
    text = msg.as_string()

    # sending the mail

    s.sendmail(fromaddr, toaddr, text)

    # terminating the session

    s.quit()

if __name__ == '__main__':

    app.run(host='0.0.0.0', debug='TRUE')
```

GitHub Link: **https://github.com/IBM-EPBL/IBM-Project-18617-1659687770**

Video Demo Link: https://youtu.be/vURy7VwSvGM