CUSTOMER CARE REGISTRY

A PROJECT REPORT

Submitted by

GOMATHY S (711619104012)

HEMALATHA P (711619104014)

MOHANA SELVI T (711619104028)

RANJITHA D (711619104036)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCEAND ENGINEERING

KATHIR COLLEGE OF ENGINEERING

“WISDOMTREE”NEELAMBUR,COIMBATORE-641062

ANNA UNIVERSITY: CHENNAI 600 025
JUNE 2022

BONAFIDE CERTIFICATE

"CUSTOMER CARE REGISTRY" is the bonafide workof Certified that this project**S.GOMATHY (711619104012), P.HEMALATHA (711619104014), T.MOHANA SELVI (711619104028), D.RANJITHA (711619104036)** who carried out the project work under mysupervision.

SIGNATURE	**SIGNATURE**
HEAD OF THE DEPARTMENT	**FACULTY MENTOR**
Dr.S.J.K.JAGADEESHKUMAR,ME,Ph.D	MS.K.N.JAYAPRIYA,ME
Department of CSE	Department of CSE
Kathir Collegeof Engineering	Kathir Collegeof Engineering
Coimbatore-641 062	Coimbatore-641 062

External viva voce held on …….

Internal Examiner External Examiner

ACKNOWLEDGEMENT

We express our immense gratitude to **Thiru E.S. KATHIR, Chairman, Kathir Institutions**, Coimbatore for giving us an opportunity to study in their prestigious institution and to take up the project in Partial fulfillment of the Regulation for the B.E Program.

We would like to express our deepest gratitude to **Tmt. LAVANYA KATHIR, Secretary, Kathir Institutions**, Coimbatore for the soul support in our studies.

We would bound to express our gratitude to **Dr.G.DORAISAMY**, CEO and **Dr.R.UDAIYAKUMAR**, M.E., Ph.D., Principal, Kathir College of Engineering, Coimbatore for their permission and constant encouragement throughout our course.

It is great pleasure to express our sincere and whole hearted gratitude to Professor **Dr. S.J.K JAGADEESHKUMAR, M.E., Ph.D., Head of the Computer Science and Engineering Department,** for the constant suggestion and encouragement in the project work.

We also express our heartfelt thanks to **Ms.K.N.JAYAPRIYA,M.E., Assistant Professor and Project Coordinator, Department of Computer Science and Engineering** for being supportive throughout the tenure of our project.

We also express our heartfelt thanks to **Mr.Vasudeva,M.E.,** and **Mr.Haunsh,M.E.,** Industry mentor from IBM, and **Ms.Jayapriya,M.E.,** Assistant Professor and Faulty Mentor,

Department of Computer Science and Engineering and also

Ms.K.R.SHANMUGAPRIYAA,M.E.,Assistant Professor and Project Guide, Department of Computer Science and

We also thank all our Faculty Members and Non-Teaching Staff Members of Department of Computer Science and Engineering and our Lovable Parents and Friends who contribute many suitable ways for achieving final results.

CHAPTER	TITLE NO	PAGNO
1.	**INTRODUCTION**	**7**
	1.1. Project Overview	7
	1.2. Purpose	7
2.	**LITERATURE SURVEY**	**9**
	2.1 Existing problem	9
	2.2 References	11
	2.3 Problem Statement Definition	12
3.	**IDEATION & PROPOSED SOLUTION**	**14**
	3.1 Empathy Map Canvas	15
	3.2 Ideation & Brainstorming	16
	3.3 Proposed Solution	17
	3.4 Problem Solution fit	18
4.	**REQUIREMENT ANALYSIS**	**19**
	4.1 Functional requirement	19
	4.2 Non Functional requirement	20
5.	**PROJECT**	**21**
	5.1 Data Flow Diagrams	21
	5.2Solution & Technical Architecture	22
	5.3 User Stories	23
6.	**PROJECT PLANNING & SCHEDULING**	**25**
	6.1 Sprint Planning & Estimation	25

	6.2Sprint Delivery Schedule	26
	6.3Reports from JIRA	27
7.	**CODING & SOLUTIONING**	**28**
	7.1 Feature 1	28
	7.2 Feature 2	30
	7.3Database Schema	30
8.	**TESTING**	**34**
	8.1Test Cases	34
9.	**RESULTS**	**36**
	9.1Performance Metrics	36
10.	**ADVANTAGES & DISADVANTAGES**	**39**
11.	**CONCLUSION**	**41**
12.	**FUTURE SCOPE**	**42**
13.	**APPENDIX**	**43**
	Source Code GitHub & Project Demo Link	43

CHAPTER 1

INTRODUCTION

1.1 Project Overview

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.

Customer Services also known as Client Service is the provision of service to customers. Its significance varies by product, industry and domain. In many cases customer services is more important if the information relates to a service as opposed to a Customer. This Application has been developed to help the customer in processing their complaints. The Customer can have many problem during their work they can raise their problem as tickets. 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. For each Customer there must be a separate agent .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.

1.2 Purpose

An online comprehensive Customer Care Solution is to manage customer interaction and complaints with the Services Providers over e-mail .The System should have the capability to rectify the customers issue and give the solution through e-mail. The status of the issue will be shown the e-mail so the customer can known about the current status of the issue.

Customer Services also Known as Client Service is the provision of service to customers. In many cases customer services is more important if the information relates to a service as opposed to a customer.

Customer Service may be provided by a Service Representative Customer Service is normally an integral part of a company's customers value proposition

Customer care boosts the overall customer experience by providing answers to common questions through the website, social media, chatbots, or with customer support agents.

Companies benefit from investing in customer care for multiple reasons:

- Customers get the insights they need to make an 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.
- Using AI to optimize customer care can increase the bottom line and provide a positive return on investment.

CHAPTER 2

LITERATURE SURVEY

2.1 Existing Problem

2.1.1 GedeJuanamast a, "The Role Of Customer Service Through Customer Relationship Management (CRM) To Increase Customer Loyalty And Good Image" ,2019

This study aims to determine the role how customer service through Customer Relationship Management (CRM) to improve customer loyalty and good image. This research method using this qualitative study, researchers used a paradigm Non Positivism / Naturalistic / Interpretative. Interpretative paradigm aims to understand the meaning of behavior, symbols, and phenomena by using sampling purposive sampling. Data Collection Techniques using interviews, documentation, observation. The results of this study that the role of customer service through Customer Relationship Management (CRM) to improve customer loyalty and good image .

2.1.2 LucieKanovsk a,"customer services and their role for industrial small and medium companies",2009

The aim of this paper is to present the problems of customer services and their important role for small and medium companies from the theoretical view and also selected results of research held in industrial SME´s. Customer services are kind of services being provided by companies to their products. Customer services can be found in all economic spheres, such as in primary sphere, as well as in secondary and tertiary one

2.1.3 Mohamma d Heydari, HadisehAbaszadeh, HabibollahDanai," The Relationship between Customer Relationship Management and Customer Satisfaction with Services Received.",2015

Today, with the advancement of information technology in the organization of new systems of production that can reduce internal costs, better interaction with the environment and ultimately help to make a profit. Customer satisfaction is one of these tools

2.1.4 A. Michael Knemeyer, Douglas M. Lambert and Sebastián J. GarcíaDastugue," The Customer Service Management Process",2004

The aim of this paper is to present the problems of customer services and their important role for small and medium companies from the theoretical view and also selected results of research held in industrial SME´s. Customer services are kind of services being provided by companies to their products. Customer services can be found in all economic spheres, such as in primary sphere, as well as in secondary and tertiary

2.2 References

- CUSTOMER CARE PROJECT REPORT

https://www.scribd.com/document/452190381/CUSTOMER-CARE-PROJECT-REPORT

- IBM Customer care

https://www.ibm.com/cloud/learn/customer-care

- Customer care service problem and hoe to resolve them

https://www.proprofsdesk.com/blog/customer-service-problems/

- A Model for Customer Complaint Management System

https://www.researchgate.net/publication/314541111_A_Model_for_Customer_Complaint_Management_System_using_SOA

2.3 Problem Statement Definition

Watson Assistant is a breakthrough in optimizing customer care. This AI-powered agent provides customers with fast, consistent, and accurate answers across any messaging platform, application, device, or channel. Watson Assistant's natural learning process analyzes customer conversations, improving its ability to resolve issues the first

An AI technology-first approach to customer care improves containment and customer satisfaction. Watson Assistant takes the basic and repetitive tasks customer support agents perform, freeing them up to do more meaningful tasks like closing the customer feedback loop. Companies can realize a two-fold benefit:

These AI and ML-powered solutions are designed to not only answer customer questions, but to do so in a more human way, creating the emotional connections so important to customer care and customer loyalty.

Fig 2.3 Problem Statement Definition

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	I am a customer in customer care registry	I am expecting instance communication with service department	The communication provided by service department is not understandable	There is a lack of communication skill for that person	If the communication is understandable I solve my problem easily

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	I am a customer in customer care registry	I am expecting instance communication with service department	The communication provided by service department is not understandable	There is a lack of communication skill for that person	If the communication is understandable I solve my problem easily

Fig 2.3.2 Problem Statement-1,11

CHAPTER 3

IDEATION AND PROPOSED SOLUTION

3.1 Empathy Map Canvas

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviors and attitudes.

It is a useful tool to helps teams better understand their users.

Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective

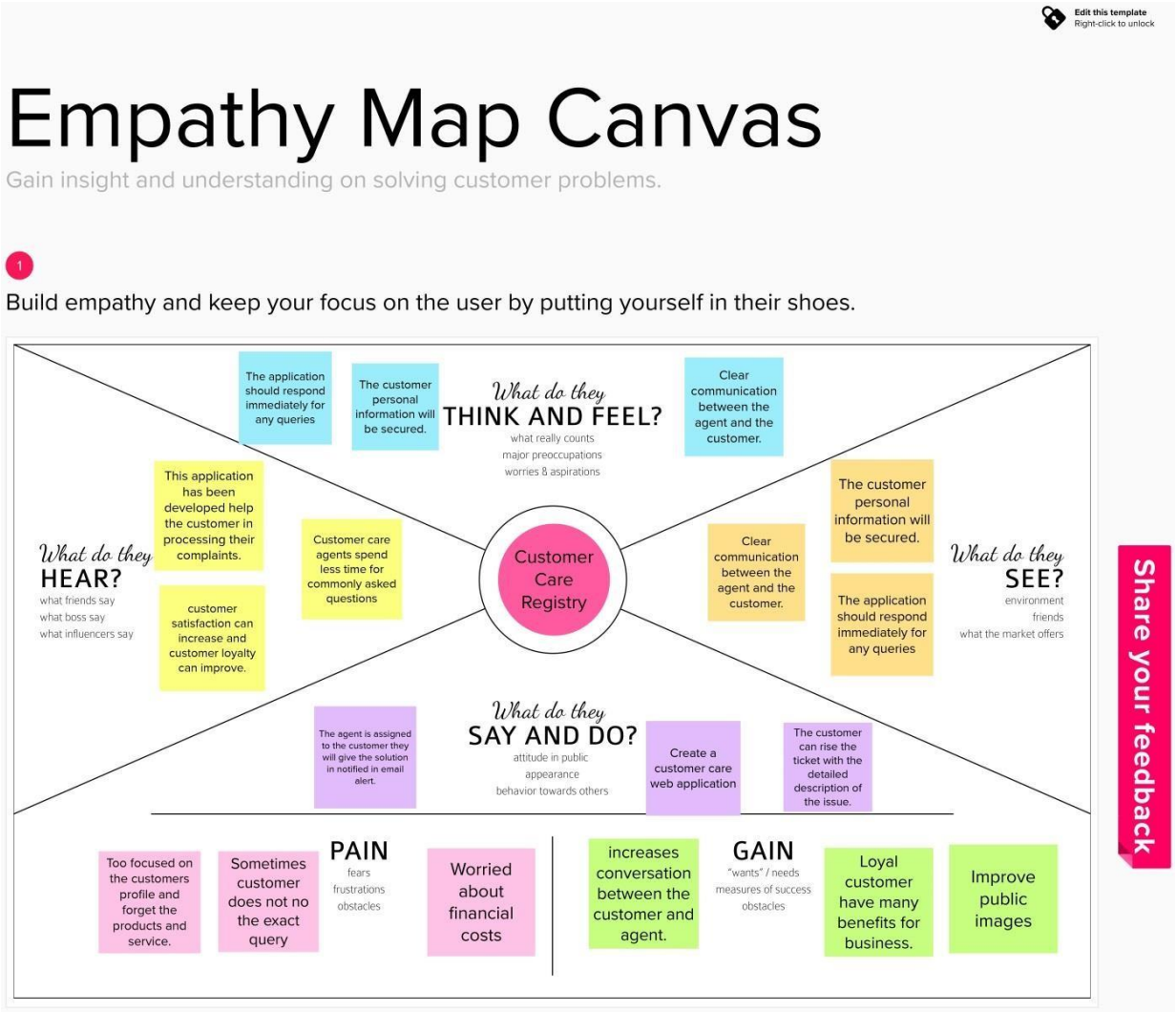


Fig 3.1 Empathy Map

3.2 Ideation and brainstorming

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

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 itself so we can use brainstroming

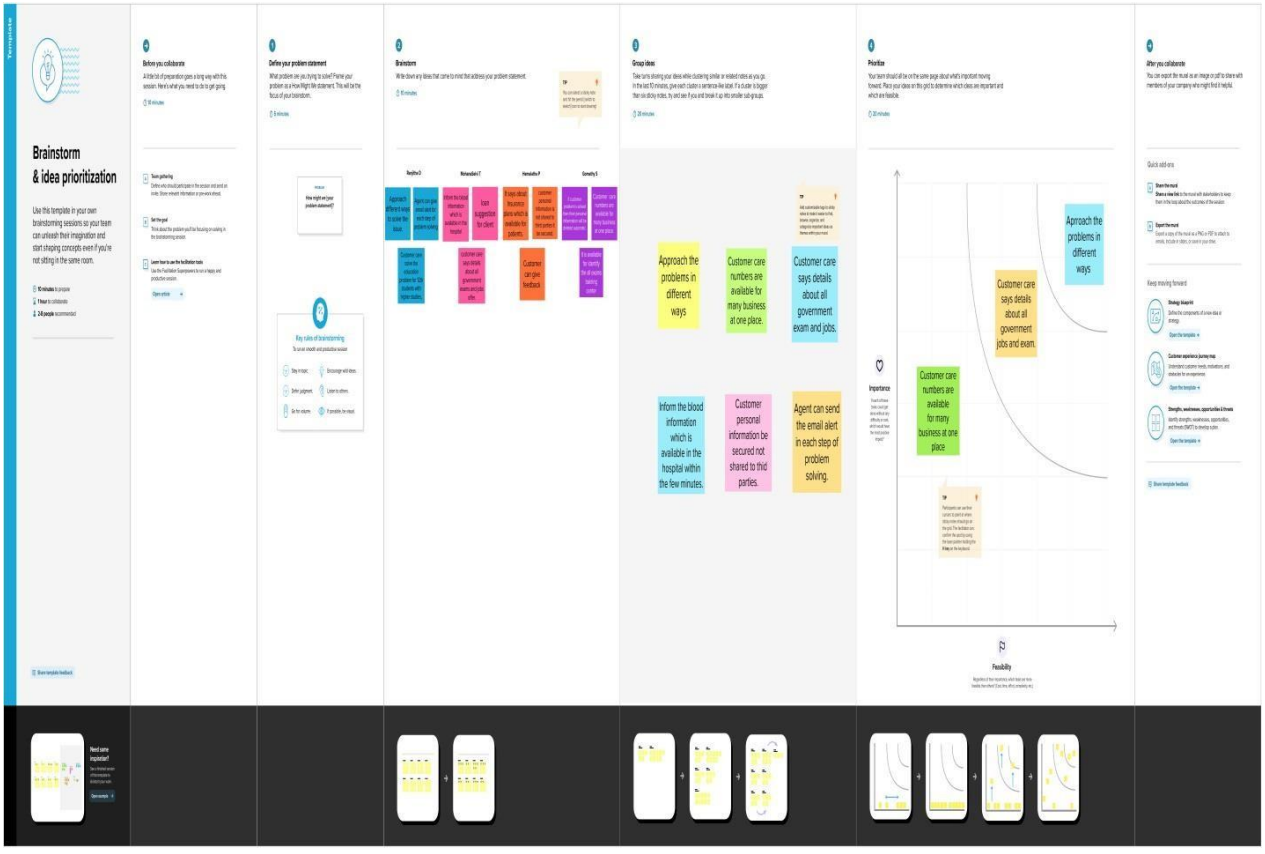


Fig 3.2 Ideation and Brainstorming

3.3 Proposed Solution

Your proposed solution section should offer your solution specifically, with enough detail so that your reader understands exactly what you're proposing. Indicate how your proposed solution will solve the problem and provide tangible benefits. Specifically, explain how it will meet the objectives and abide by the constrains outlined in the problem definition

Proposed Solution Template:

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

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	The information sharing to the agent through mailing feature .Agent resolve the problem 24*7
2.	Idea / Solution description	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.
3.	Novelty / Uniqueness	The Complaint of the user is satisfied at any time using mail. User can view status of the complaint
4.	Social Impact / Customer Satisfaction	Customer satisfaction does have a positive effect on an organisation's profitability. when the customer are happy with the service they receive, they are more likely to trust and be loyal to the company.
5.	Business Model (Revenue Model)	Customer service model includes the policies and strategies .Customer care registry will increase the customer satisfaction and reduce the customer problem
6.	Scalability of the Solution	An environment where they will be able to spend less time on grunt work and more time on actually resolving critical customer issue.

. **Fig 3.3 Proposed solution**

3.4 Problem Fit Solution

The Problem-Solution Fit canvas is based on the principles of Lean Startup, LUM (Lazy User Model) and User Experience design. It helps entrepreneurs, marketers and corporate innovators identify behavioral patterns and recognize what would work and why. It is a template to help identify solutions with higher chances of solution adoption, reduce time spent on testing and get a better overview of the current situation.

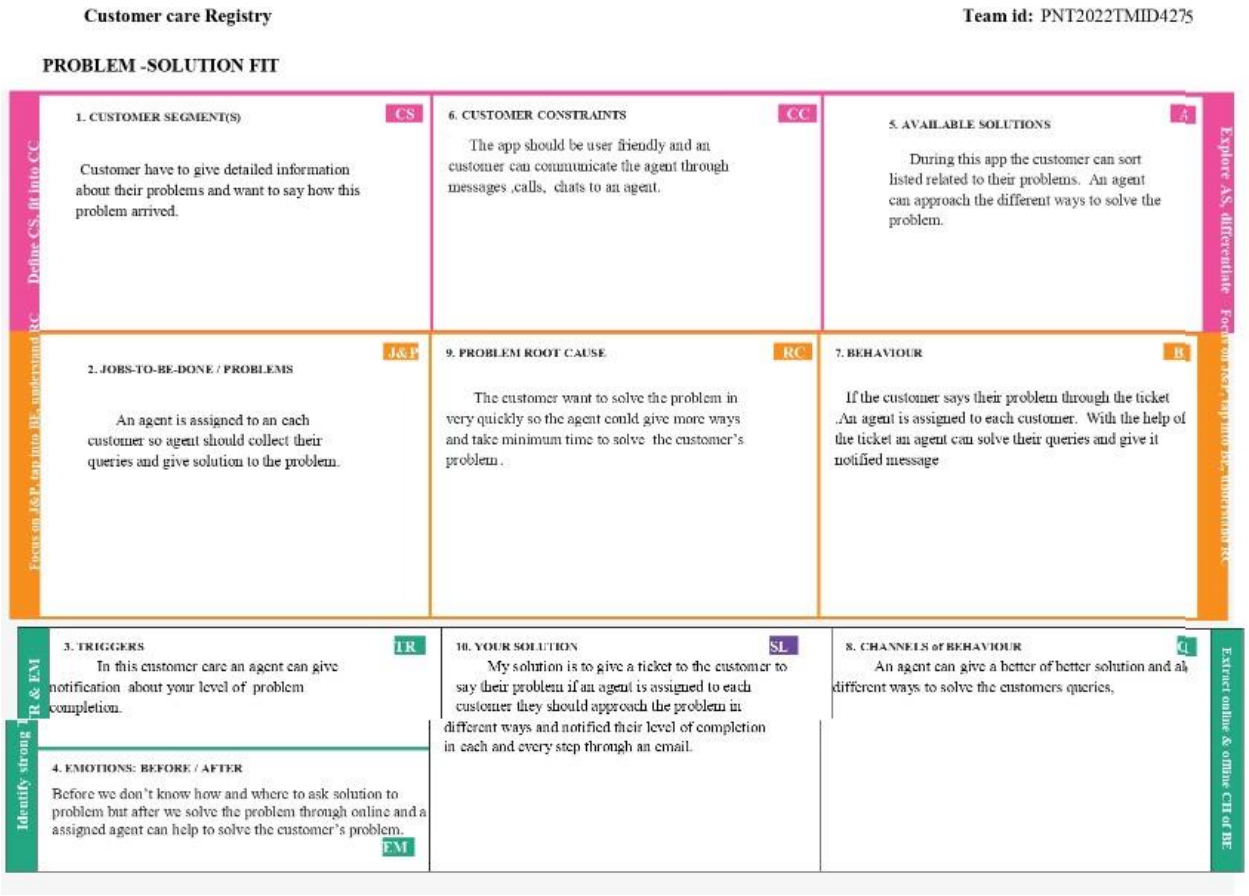


Fig 3.4 Problem Fit Solution

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 Functional Requirement

Functional requirements are product features or functions that developers must implement to enable users to accomplish their tasks. So, it's important to make them clear both for the development team and the stakeholders. Generally, functional requirements describe system behavior under specific conditions.

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User login	Login through Email Login through Password
FR-4	Complaint Registration	User can create the complaint with a description of the problem they are facing.
FR-5	Resolving Problem	Each Customer can be assigned with an agent .Agent can resolve the complaint
FR-6	Customer Satisfaction	The user can view status of the complaint through Email. After resolving the problem notification will be send

Fig 4.1 Functional Requirement

4.2 Non Functional Requirements

These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to other. They are also called non-behavioral requirements. They basically deal with issues like: Portability, Security Maintainability, Reliability ,Scalability.

Non-functional Requirements:

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

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	This Application has been developed to help the customer in processing their complaints
NFR-2	**Security**	This application ensures the security and the integrity of data. This is done by providing a password login system for each authorized user
NFR-3	**Reliability**	The customer information and complaints are stored carefully. There is no risk and loss of data.
NFR-4	**Performance**	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.
NFR-5	**Availability**	The system is available all the time, no time constraint and the user can login it from anywhere.
NFR-6	**Scalability**	Each user will be assigned with an agent the complaints will be rectified easily so it is highly

Fig 4.2 Non Functional Requirements

CHAPTER 5

PROJECT DESIGN

5.1 Data Flow Diagram

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled.

Data flow diagram

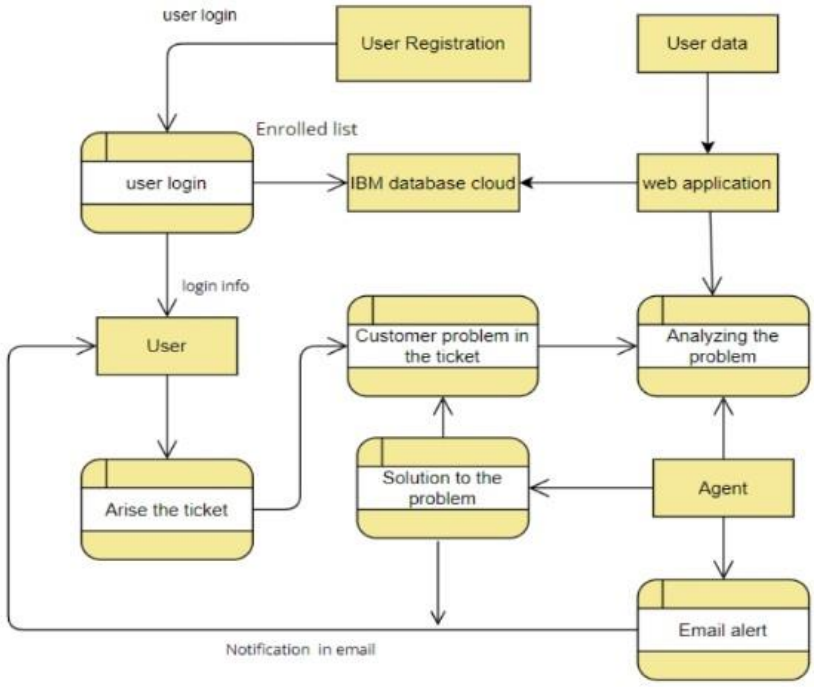


Fig 5.2 Data Flow Diagram

5.2 Solution and Technical Architecture

Technical architecture—which is also often referred to as application architecture, IT architecture, business architecture, etc.—refers to creating a structured software solution that will meet the business needs and expectations while providing a strong technical plan for the growth of the software application through its lifetime. IT architecture is equally important to the business team and the information technology team.Technical architecture includes the major components of the system, their relationships, and the contracts that define the interactions between the components. The goal of technical architects is to achieve all the business needs with an application that is optimized for both performance and security

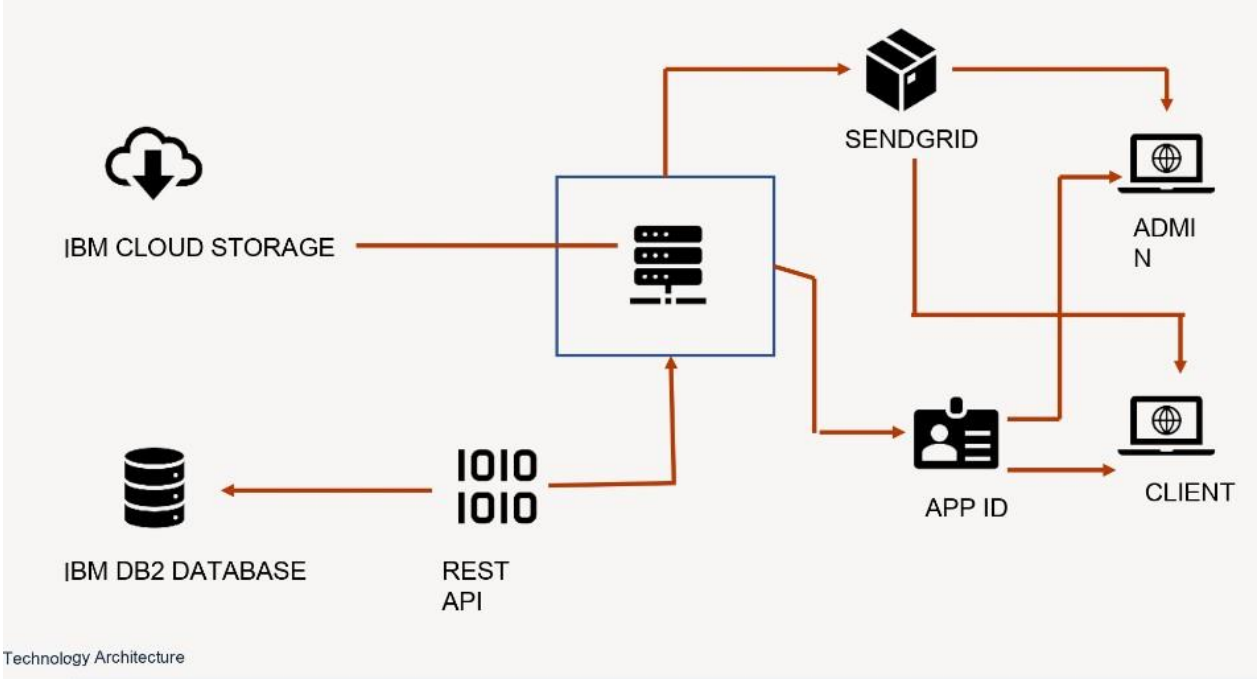


Fig 5.2 Solution and Technical Architecture

5.3 User Stories

A user story is the smallest unit of work in an agile framework. It's an end goal, not a feature, expressed from the software user's perspective. A user story is an informal, general explanation of a software feature written from the perspective of the end user or customer. The purpose of a user story is to articulate how a piece of work will deliver a particular value back to the customer. Note that "customers" don't have to be external end users in the traditional sense, they can also be internal customers or colleagues within your organization who depend on your team.

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 user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can register with Email and access the dashboard	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can register with Email and password	High	Sprint-1
	Dashboard		As a user, I can Arise the ticket to say the problem	I can manually analyzing the problem.	High	Sprint-1
Customer (Web user)	user		As a customer, I can say my problem in the ticket	To provide good solution should analyzing be careful	High	Sprint-1
Customer Care Executive	Technical support		Requested the detailed description of the problem.	The customer will satisfied with the given solution.	High	Sprint-1
Administrator	Creator		An Agent can control the process	Inform about level of solution in mail notification.	Medium	Sprint-1
			Chat with the agent	Optimize the code and say feedback about customer service.	High	Sprint-1

Fig 5.3 Use Stories Diagram

CHAPTER 6

PROJECT PLANNING AND SCHEDULING

6.1 Sprint Planning and Estimation

Sprint planning is an event in scrum that kicks off the sprint. The purpose of sprint planning is to define what can be delivered in the sprint and how that work will be achieved. Sprint planning is done in collaboration with the whole scrum team.In scrum, the sprint is a set period of time where all the work is done. However, before you can leap into action you have to set up the sprint. You need to decide on how long the time box is going to be, the sprint goal, and where you're going to start.

The sprint planning session kicks off the sprint by setting the agenda and focus. If done correctly, it also creates an environment where the team is motivated, challenged, and can be successful. Bad sprint plans can derail the team by setting unrealistic expectations.As described in the Scrum Guide, Sprint Planning initiates the Sprint by laying out the work to be performed for the Sprint. This resulting plan is created by the collaborative work of the entire Scrum Team

Title	Description	Date
Literature Survey and Information Gathering	Gathering Information by referring the technical papers, research publications etc	17 SEPTEMBER 2022
Prepare Empathy Map	To capture user pain and gains Prepare List of Problem Statement	10 SEPTEMBER 2022
Ideation	Prioritise a top 3 ideas based on feasibility and Importance	20 SEPTEMBER 2022
Proposed Solution	Solution include novelty,feasibility,business model,social impact and scalability of solution	24 SEPTEMBER 2022
Problem Solution Fit	Solution fit document	3 October 2022
Solution Architecture	Solution Architecture	11 October 2022
Customer Journey	To Understand User Interactions and experiences with application	14 October 2022
Functional Requirement	Prepare functional Requirement	15 October 2022
Data flow Diagrams	Data flow diagram	15 October 2022
Technology Architecture	Technology Architecture diagram	16 October 2022
Project Development-Delivery of sprint 1,2,3 &4	Develop and submit the developed code by testing it	24 October 2022 – 19 November 2022

Fig 6.1 Sprint Planning and Estimation

6.2 Sprint Delivery Schedule

The main event during agile methodology is the sprint, the stage where ideas turn into innovation and valuable products come to life. On one hand, agile sprints can be highly effective and collaborative. At the same time, they can be chaotic and inefficient if they lack proper planning and guidance. And for this reason, making a sprint schedule is one of the most important things you can do to ensure that your efforts are successful. If you're looking to schedule your next sprint, you've come to the right place. Keep reading to learn everything you need to know about sprint scheduling, including some tips to drive the best results. Since sprints take place over a fixed period of time, it's critical to avoid wasting time during planning and development

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Cloud Application Development- Customer care Registry

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, and password, and confirming my password.	2	High	4
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	4
Sprint-2		USN-4	As a user, I can register for the application through Gmail	2	Medium	4
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	3	High	4
Sprint-1	Dashboard	USN-6	As a user, I can told my problems in a given ticket	3	High	4
Sprint-1		USN-7	As a user, I can wait for my solution.	2	High	4
Sprint-2		USN-8	As a user, I can check if any mail received.	1	High	4
Sprint-3		USN-9	As a user, I can ask any doubts for the given solution.	2	High	4
Sprint-4	Database	USN-11	As a user, I can view and access database information.	2	High	4

Fig 6.2 Sprint Delivery Schedule

6.3 Reports from JIRA

One part of ensuring the success and smooth operations of your projects in JIRA is reporting. It involves gaining the knowledge about the health, progress and overall status of your JIRA projects through Gadgets, report pages or even third party applications. The goal of this guide is to provide an overview of the tools available to JIRA users today and how they can be used to fulfill the different types of reporting needs that users face today.JIRA offers reporting in a number of different formats. Project reports that are available from the home screen of the selected project, Gadgets that can be added and arranged in Dashboards and for each filter, the issue navigator offers various output formats that can be used in third party reporting software. Additionally, we will mention some advanced methods that customers have been using.In JIRA, a project will automatically offer standard reports available to the user without any necessary configuration. These standard reports comprise a wide range of reporting applications such as time tracking, workload and also abstract reports like Pie Charts that can be used in various ways.

The basis of almost any custom reporting is the Issue Navigator that enables you to slice and dice the data in JIRA in almost any way imaginable. The queries in the Issue Navigator can be created by using either a simple search or a JQL statement in the advanced search.

CHAPTER 7

CODING ANG SOLUTIONING

7.1 Feature 1

7.1.1 IBM Cloud

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.

When done well, customer care boosts the overall customer experience by providing answers to common questions through the website, social media, chatbots, or with customer support agents

7.1.2 KUBERNETES CLUSTER

With the widespread adoption of containers among organizations, Kubernetes, the container-centric management software, has become the de facto standard to deploy and operate containerized applications. Google Cloud is the birthplace of Kubernetes—originally developed at Google and released as open source in 2014. Kubernetes builds on 15 years of running Google's containerized workloads and the valuable contributions from the open source community. Inspired by Google’s internal cluster management system, Borg, Kubernetes makes everything associated with deploying and managing your application easier. Providing automated container orchestration, Kubernetes improves your reliability and reduces the time and resources attributed to daily operations.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name:centos-deployment
spec:
   selector:
     matchlabels:
       app:ngnix
   replicas: 2
   template:
    metadata:
      labels:
        app:ngnix
    spec:
    container:
     -name: ngnix
      image :ngnix:1.14.2
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: admin-user
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
```

```
subjects:
- kind: ServiceAccount
  name: admin-user
  namespace: kubernetes-dashboard
kind: ServiceAccount
metadata:
  name: admin-user
  namespace: kubernetes-dashboard
```

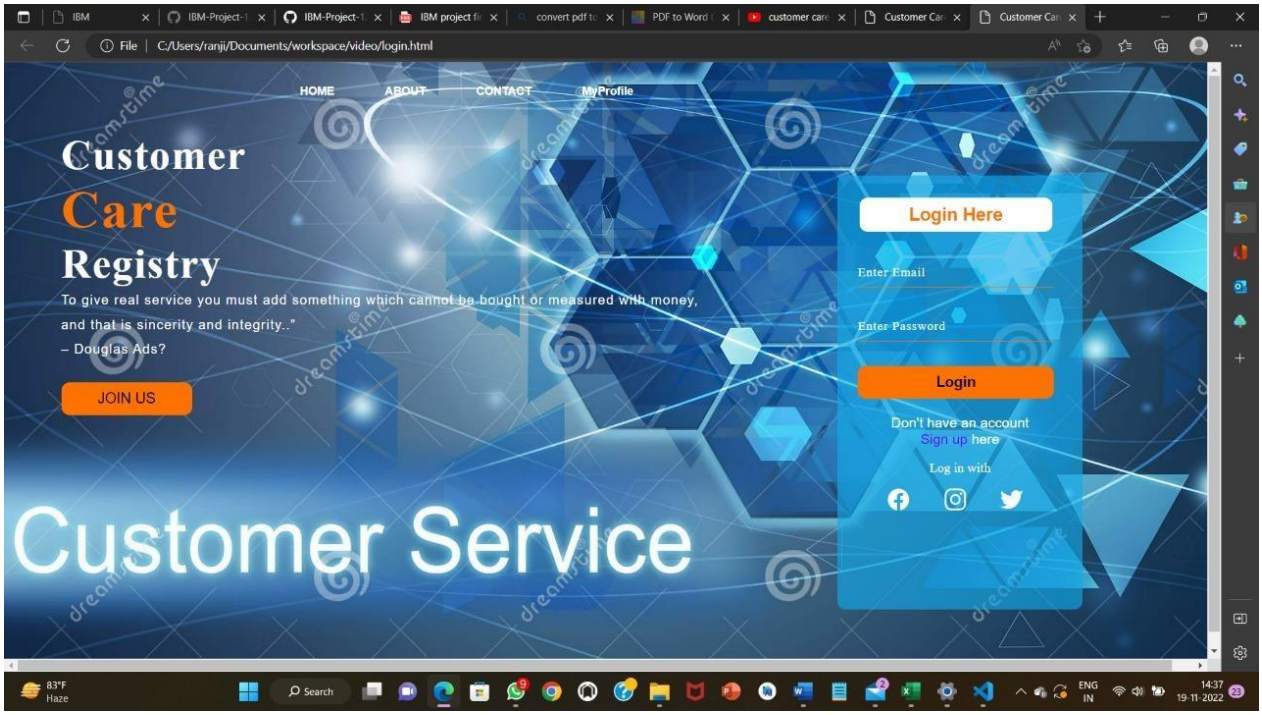
7.2 FEATURE 11

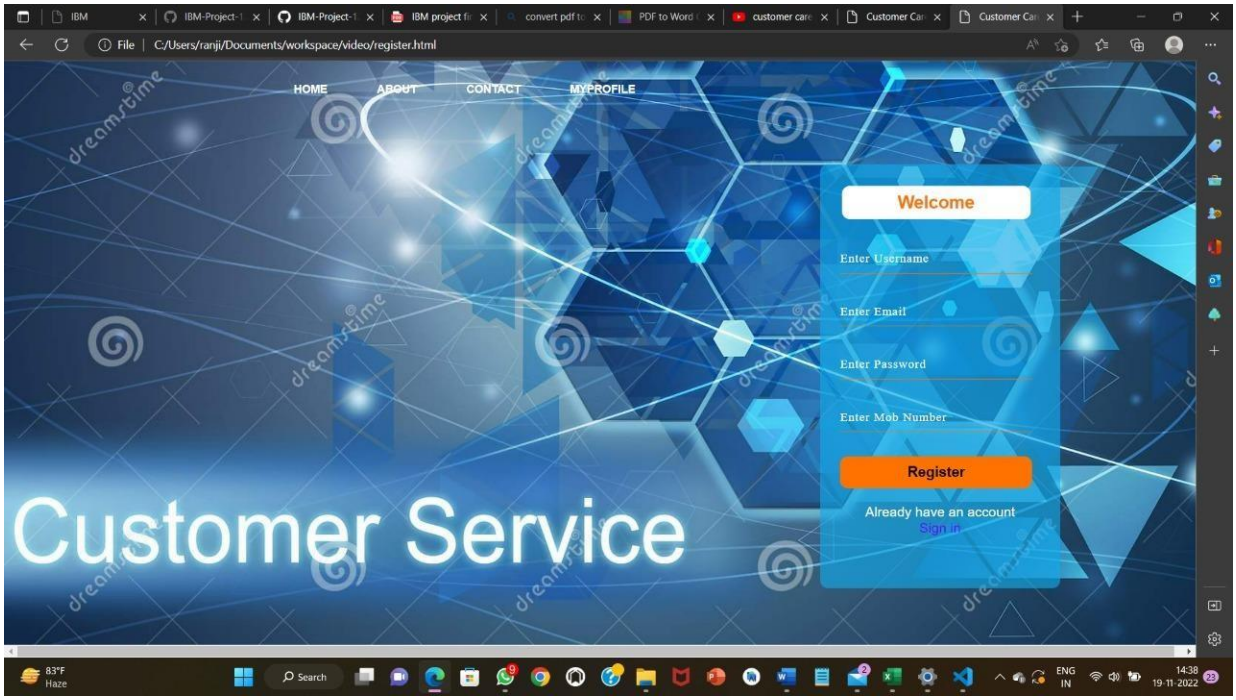
7.2.1 Docker

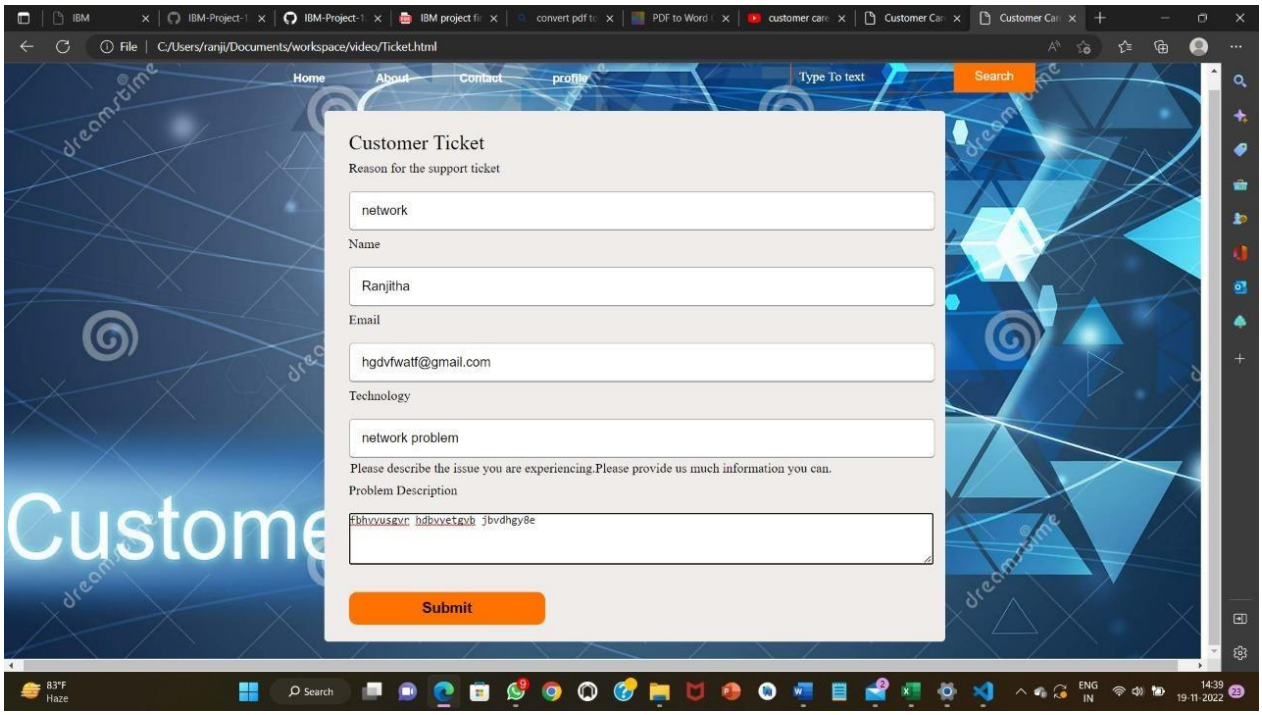
Docker is an open source containerization platform. It enables developers to package applications into containers—standardized executable components combining application source code with the operating system (OS) libraries and dependencies required to run that code in any environment. Containers simplify delivery of distributed applications, and have become increasingly popular as organizations shift to cloud-native development and hybrid multicloud environments.

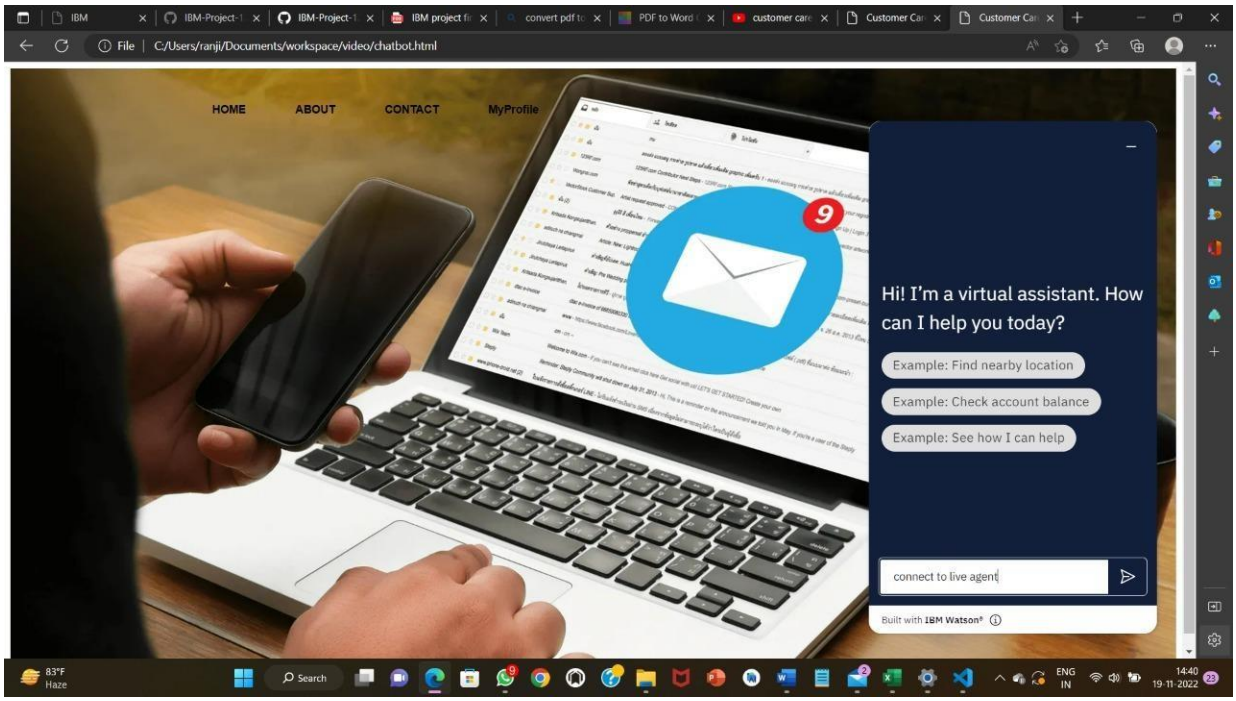
Developers can create containers without Docker, but the platform makes it easier, simpler, and safer to build, deploy and manage containers. Docker is essentially a toolkit that enables developers to build, deploy, run, update, and stop containers using simple commands and work-saving automation through a single API.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name:centos-deployment
spec:
   selector:
   matchlabels:
   app:ngnix
   replicas: 2
   template:
   metadata:
      labels:
        app:ngnix
    spec:
    container:
     -name: ngnix
      image :ngnix:1.14.2
```


File | C:/Users/ranji/Documents/workspace/video/login.html
HOME
ABOUT
CONTACT
MyProfile
Customer
Care
Registry
To give real service you must add something which cannot be bought or measured with money, and that is sincerity and integrity.."
– Douglas Ads?
JOIN US
Customer Service
Login Here
Enter Email
Enter Password
Login
Don't have an account
Sign up here
Log in with


File | C:/Users/ranji/Documents/workspace/video/register.html
HOME
ABOUT
CONTACT
MY PROFILE
Customer Service
Welcome
Enter Username
Enter Email
Enter Password
Enter Mob Number
Register
Already have an account
Sign in





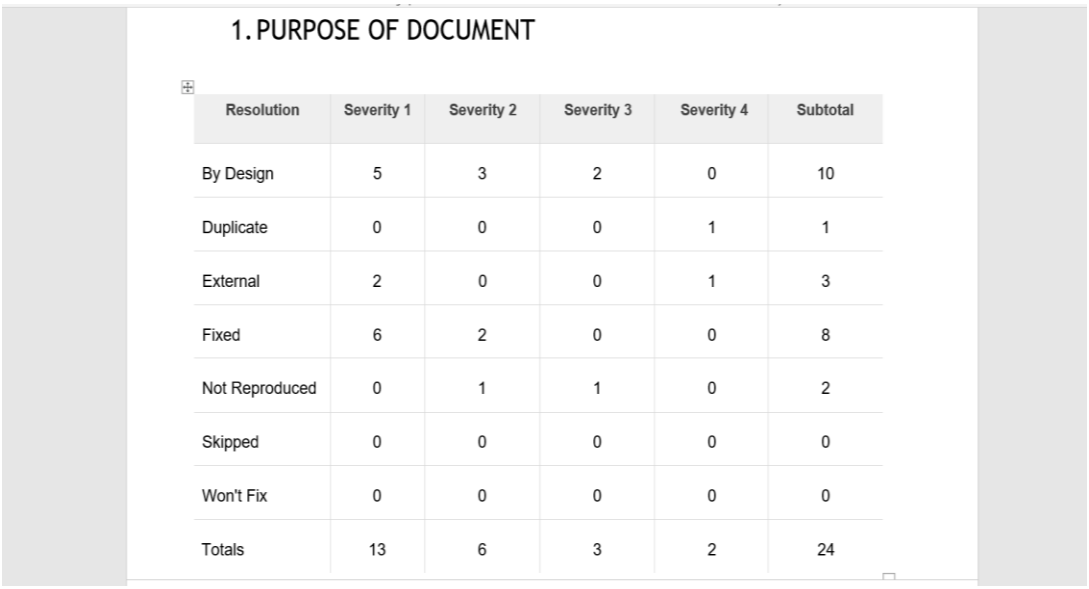
questions. To accomplish this, you define actions for the assistant.

CHAPTER 8

TESTING

8.1.1 FUNCTIONAL TESTING

Functional test can be defined as testing two or more modules together with the intent of finding defects, demonstrating that defects are not present, verifying that the module performs its intended functions as stated in the specification and establishing confidence that a program does what it is supposed to do.



1. PURPOSE OF DOCUMENT

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	5	3	2	0	10
Duplicate	0	0	0	1	1
External	2	0	0	1	3
Fixed	6	2	0	0	8
Not Reproduced	0	1	1	0	2
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	13	6	3	2	24

3. TEST CASE ANALYSIS

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	1	0	0	1
Client Application	2	0	0	2
Security	1	0	0	1
Outsource Shipping	1	0	0	1
Exception Reporting	1	0	0	1
Final Report Output	2	0	0	2
Version Control	2	0	0	2

Testcase

				Date	19 November 2022							
				Team ID	PNT2022TMID42765							
				Project Name	**Project** Cloud Application Development-Customer Care Registry							
				Maximum Marks	4 marks							
Test case ID	**Feature Type**	**Component**	**Test Scenario**	**Pre-Requisite**	**Steps To Execute**	**Test Data**	**Expected Result**	**Actual Result**	**Status**	**TC for Automation (Y/N)**	**BUG ID**	**Executed By**
IBM CLOUD _TC_001	Functional	IBM Cloud Service	Verify the login cloud services	Software IBM Cloud Service	1. Login in cloud.ibm.com 2. Then apply code the and Login 3. Have been login in to the IBM cloud	email: 711619104036@smartinternz.com Password : sJ3LngbI	Successfully created the IBM account	Working as expected	Pass	YES	NIL	Ranjitha D Gomathy S Hemalatha P MohanaSelvi T
IBM Watson IoT Platform_TC_002	Functional	IBM Cloud Service	Verify create a device in the IBM Watson IoT platform and get the device credentials.		1. In IBM Cloud Service go to catalog 2. create and launch the IBM Watson IoT Platform 3.Login the Platform by clicking organization ID 4.Create a device & configure the device type and ID 5.Generate the API Key	Create a device & integrate with code		Working as expected	Pass	YES	NIL	Ranjitha D Gomathy S Hemalatha P MohanaSelvi T

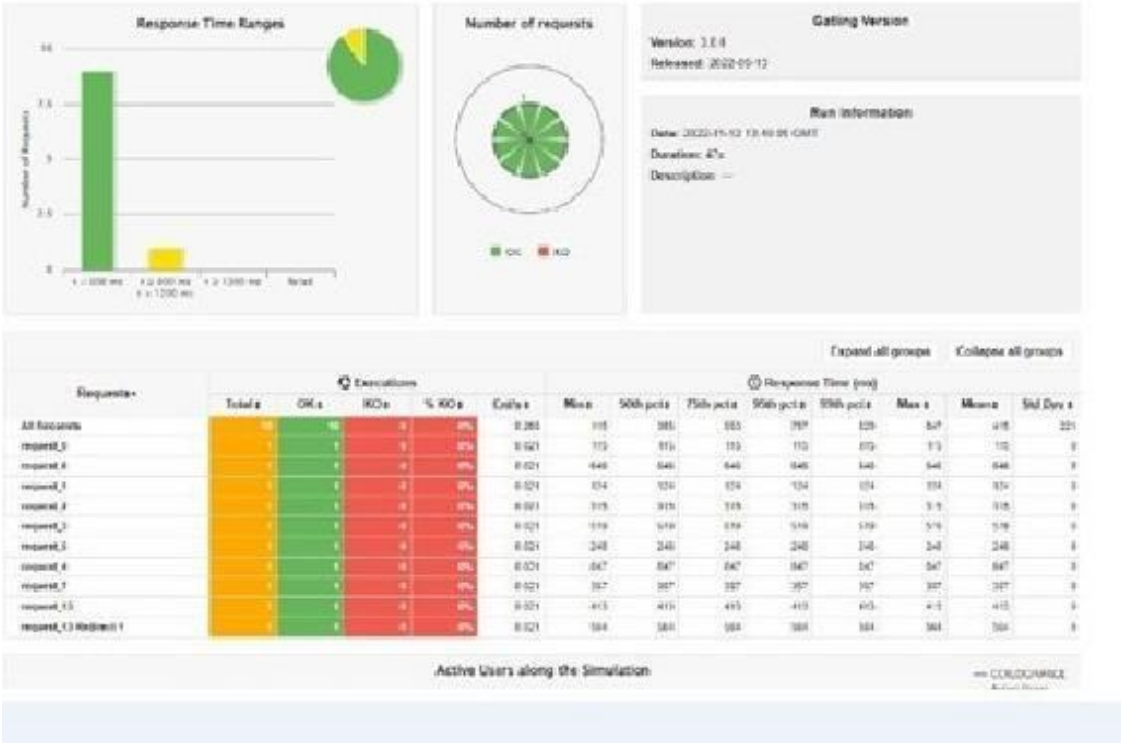
PythonCode_TC_003	Code	Python 3.6.9	Verify whether the python code is without error by running it	Software	1. Download the python version3.6.9 2. Type the program and save it with the extension.py 3. Verify it by run the code and decode the error by completion	import python import flask import mail		Working as expected	Pass	YES	NIL	Ranjitha D Gomathy S Hemalatha P MohanaSelvi T
pythonCode_TC_004	Non-Functional	IBM Cloud Service	Create an IBM Watson Assistant	IBM cloud services	1. In IBM cloud go to catalog 2. To create a IBM Watson Assistant 3. Deploy app 4. Visit the app URL	We create an Watson Assistant.	Successfully created the Assistant	Working as expected	Pass	YES	NIL	Ranjitha D Gomathy S Hemalatha P MohanaSelvi T
Database DB2	Dataset	IBM Cloud Service	Verify the events is stored in the database	IBM Cloud Service	1. Go to IBM Cloud Services 2. In resources list, click on db2.In database, create a table for entering data.	Shows the data in the table	Successfully created the Database	Working as expected	Pass	NO	NIL	Ranjitha D Gomathy S Hemalatha P MohanaSelvi T
IBM Watson Assistant	Functional	IBM Cloud Service	To create a web UI to interact with user	IBM Cloud Service	1. Go to web page 2. In Ticket type your problem	Shows the ticket page.	It stores in the database.	Working as expected	Pass	NO	NIL	Ranjitha D Gomathy S Hemalatha P MohanaSelvi T

CHAPTER 9

RESULT

9.1 Performance Metrices

Login Page



Registration Page

Ticket Page


Response Time Ranges
Number of requests
Gatling Version
Run Information

CHAPTER 10

ADVANTAGE AND DISADVANTAGE

Advantage

A good product or service can only get you so far. If you add great customer service to the equation, the value of your offering immediately increases.

Right from the first electronic mail sent in 1971 to the billions of emails that are sent and received every day, we have come a long way.

Among the different types of customer service available, customers consider email as a more trustworthy and professional channel. No wonder 12% of customers still choose email to register their requests.

The best part about email customer service is that it doesn't cost a fortune. Your agents get more time to respond, and they can use canned responses or email templates for faster replies.

One major challenge with email customer service is that after a point in time it becomes difficult to keep track of every single email. In such a case, you can adopt customer email management software to convert emails into tickets and ensure they can never slip through the cracks.

- Add a professional touch to your customer service using email signatures

- Automated email notifications can be used to update customers about the status of their issue or support ticket
- Easily attach relevant images, videos, docs, or other files
- Record and document customer conversations over a period of time
- Assist customers in real-time when they are shopping and increase sales
- Easily embed a chat window on your website and explore customization options

Disadvantage

- Delayed email responses can make customers feel frustrated
- Keeping track of emails can get challenging when you receive hundreds of them every day
- Typing long replies can be time-consuming
- Lack of real-time human-to-human interaction
- Chat replies can often feel scripted and robotic to customers
- The use of emoticons or slangs by customers can prevent a meaningful conversation
- AI-powered chatbots can be expensive for your business

CHAPTER 11

CONCLUSION

In conclusion, customer care, involves the use of basic ethics and any company whowants to have success and grow, needs to remember, that in order to do so, it must begin withestablishing a code of ethics in regards to how each employee is to handle the dealing withcustomers. Customers are at the heart of the company and its growth or decline. Customer careinvolves, the treatment, care, loyalty, trust the employee should extend to the consumer, as wellin life. This concept can be applied to so much more than just customer care. People need to treatothers with respect and kindness, people should try to take others into consideration whenmaking any decision. If more people were to practice this policy, chances are the world would better, more understanding place for all to exist.Using Customer Care Registry user problems can be rectified and they have the better service with the Customer Service.

CHAPTER 12

FUTURE SCOPE

Talking to customers face to face, *en masse*, every day is unique to customer service. That close contact is an untapped and unappreciated opportunity to build strong customer relationships, increase loyalty, hear customer feedback, improve products, build community, upsell, increase basket size, and so on.

More and more businesses are realizing this opportunity. Forward thinking business leaders are *already* elevating the contact center to strategic status and using it get an advantage over competitors.

The shift from a primarily cost center to primarily growth center worldview.The job descriptionfor a customer service director will focus more on leadership, innovation, and ability to drive company-wide improvement. Customer service will shift to become a strategic partner of marketing, sales, and product development. CS will help with direction, project prioritization, and impact.A need for customer service leaders to take a highly strategic seat at the table. They'll need to argue for investment in talent, technology, and innovation.

CHAPTER 13

APPENDIX

Source Code

```
<!DOCTYPE html>
<html lang="en">
<head>
   <title>Customer Care Registry</title>
   <link rel="stylesheet" href="style.css">
</head>
<body>

   <div class="main">
      <div class="navbar">

      <div class="menu">
            <ul>
               <li><a href="home.html">HOME</a></li>
               <li><a href="About.html">About</a></li>
               <li><a href="contact.html">Contact</a></li>
               <li><a href="Profile.html">Profile</a></li>
               <li><a href="login.html">Login </a></li>
               <li><a href="register.html">Register </a></li>
            </ul>
```

```
</div>


</div>
<div class="content">


<h1>Customer<br><span>Care</span> <br>Registry</h1>

<p class="par">Providing excellent customer care can separate you from the competition. It's more commonplace to provide great customer service, but customer care comes with time, effort and an earnest interest in building a relationship. Customers should be able to see the value your business provides if customer care is a focus.

<br> A successful business will work on both customer care and customer service, making each a focus to form a mutually beneficial relationship with the customer that can satisfy their needs and increase sales and brand awareness for the business</p>

<button class="cn"><a href="#">Learn More</a></but

</div>

</div>

</div>

</div>


<script
src="https://unpkg.com/ionicons@5.4.0/dist/ionicons.js"></script>

</body>
```

Login

```
<!DOCTYPE html>
<html lang="en">
<head>
   <title>Customer Care Registry</title>
   <link rel="stylesheet" href="style.css">
</head>
 <div class="main">
       <div class="navbar">
         <div class="menu">
             <ul>
                <li><a href="home.html">HOME</a></li>
                <li><a href="About.html">ABOUT</a></li>
                 <li><a href="contact.html">CONTACT</a></li>
                <li><a href="profile.html">MyProfile</a></li>
             </ul>
       </div>
                          <!-- <div class="search">
             <input class="srch"  type="search"  name=""  placeholder="Type To text">
             <a href="#"> <button class="btn">Search</button></a>
          </div> -->


       </div>
```

```
<div class="content">
 <h1>Customer<br><span>Care</span> <br>Registry</h1>
   <p class="par">To give real service you must add something which cannot be bought or measured with money,
     <br> and that is sincerity and integrity..”<br> – Douglas Ads?</p
   <button class="cn"><a href="#">JOIN US</a></button
   <div class="form">
     <form action="/login" method="post">
      <h2>Login Here</h2>
        <input type="email" name="email" placeholder="Enter Email ">
       <input type="password" name="pwd" placeholder="Enter Password ">

        <div class="errMsg">
          <p class="text-muted errMsg ">
           {{message}}
          </p>

 <button class="btnn"><a href="Ticket.html">Login</a></button>

        <p class="link">Don't have an account<br>
        <a href="register.html">Sign up </a> here</a></p>
        <p class="liw">Log in with</p>
```

```
                <div class="icons">
                    <a href="#"><ion-icon name="logo-facebook"></ion-icon></a>
                    <a href="#"><ion-icon name="logo-instagram"></ion-icon></a>
                    <a href="#"><ion-icon name="logo-twitter"></ion-icon></a>
                </div>
            </form>
        </div>
    </div>
  </div>
 </div>

  <script src="https://unpkg.com/ionicons@5.4.0/dist/ionicons.js"></script>
</body>
</html>
```

Register

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Customer Care Registry</title>
  <link rel="stylesheet" href="style.css">
</head>
```

```
<body>
    <div class="main">
    <div class="navy”>
        <div class="menu">
          <ul>
            <li><a href="#">HOME</a></li>
            <li><a href="#">ABOUT</a></li>
            <li><a href="#">CONTACT</a></li>
            <li><a href="#">MYPROFILE</a></li>
          </ul>
        </div>
    </div>
    <div class="content">
        <div class="form">
          <form action="#" method="post">   <!--  /receivedata -->
            <h2>Welcome</h2>
            <input type="name" name="name" id="name" placeholder="Enter Username ">
            <input type="email" name="email" id="email" placeholder="Enter Email ">
            <input type="password" name="password" id="password" placeholder="Enter Password ">
            <input type="phoneNumber" name="phonenumber" id="phonenumber" placeholder="Enter Mob Number ">
            <button class="btnn" value="Send message"><a href="Ticket.html">Register</a></button>
```

```
                <p class="link">Already have an account<br>
                <a href="login.html">Sign in</a></p>
                <div class="icons">
                    <a href="#"><ion-icon name="logo-facebook"></ion-icon></a>
                    <a href="#"><ion-icon name="logo-instagram"></ion-icon></a>
                    <a href="#"><ion-icon name="logo-twitter"></ion-icon></a>
                </div>
              </form>
             </div>
         </div>
       </div>
     </div>
   <script src="https://smtpjs.com/v3/smtp.js"></script>
</body>
</html>
```

Ticket

```
<!DOCTYPE html>
<html>
   <head>
      <title>
         Customer Care Registry</title>
         <link rel="stylesheet" href="style.css">
```

```
</head>
<body>
    <div class="main">
      <div class="navbar">
        <div class="menu">
          <ul>
            <li><a href="home.html">Home</a></li>
            <li><a href="About.html">About</a></li>
            <li><a href="Contact.html">Contact</a></li>
            <li><a href="profile.html">profile</a></li>
          </ul>
        </div>
   <div class="search">
            <input class="srch" type="search" name="" placeholder="Type To text">
            <a href="#"> <button class="btn">Search</button></a>
        </div>
      </div>
                        <div class="container">
        <div class="mb-3 successMsg">{{message}}</div>
        <div class="title">Customer Ticket</div>
          <form action="#" method="post">
            <div class="user-details">
              <div class="input-box">
      <span class="details">Reason for the support ticket</span><br>
```

```
            <input type="text" placeholder="Eg:financial" required>
        </div>
        <div class="input-box">
            <span class="details">Name </span> <br>
            <input type="text" name="name" id="name" placeholder="Ram" required>
        </div>
        <div class="input-box">
            <span class="details">Email</span><br>
            <input type="text" name="email" id="email" placeholder="Eg:jkbdksjb@gmail.com" required>
        </div>
        <div class="input-box">
            <span class="details">Technology</span><br>
            <input type="text" name="Technology" id="Technology" placeholder="Eg:banking" required>
        </div>
        <p>Please describe the issue you are experiencing.Please provide us much information you can.</p>
        <div class="input-box">
            <label for="issue" class="details">Problem Description</label><br>
            <textarea class="textarea" name="problem Description" id="problem Description" name="textarea" rows="4" cols="95"></textarea>
        </div>
```

```
                    <button class="btnn"><a href="chatbot.html">Submit</a></button>
                    <!-- <div class="button">
                        <input type="submit"><a href="chatbot.html">Register</a>
                        <button class="btnn"><a href="Ticket.html">Login</a></button>
                    </div> -->
                </div>
            </form>
        </div>
    </div>
</body>
</html>
```

Chatbot

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Customer Care Registry</title>
    <style>
        .main{
            width: 100%;
            background-image: url('https://www.digitawise.com/wp-content/uploads/2017/09/customer-service-email.png');
            background-position: center;
            background-size: cover;
```

```
        height: 100vh;
    }
    .navbar{
        width: 1200px;
        height: 75px;
        margin-left: 150px;
    }
 ul{
 float: left;
 display: flex;
 justify-content: center;
 align-items: center;
}
ul li{
 list-style: none;
 margin-left: 62px;
 margin-top: 27px;
 font-size: 14px;
}
 ul li a{
 text-decoration: none;
 color:black;
 font-family: Arial;
 font-weight: bold;
 transition: 0.4s ease-in-out;
```

```
  }
  ul li a:hover{
   color: #ff7200;
  }
   </style>
</head>
<body>
      <div class="main">
       <div class="navbar">
       <div class="menu">
            <ul>
               <li><a href="home.html">HOME</a></li>
               <li><a href="About.html">ABOUT</a></li>
                <li><a href="contact.html">CONTACT</a></li>
               <li><a href="profile.html">MyProfile</a></li>
            </ul>
       </div>
       </div>
      </div>
    </div>
   </div>
   <script
src="https://unpkg.com/ionicons@5.4.0/dist/ionicons.js"></script>
    <script>
       window.watsonAssistantChatOptions = {
```

```
        integrationID: "8d0666e8-430a-45b7-a995-14976fb338e5", // The ID of this integration.
        region: "au-syd", // The region your integration is hosted in.
        serviceInstanceID: "29f10d11-a4c7-4c3c-b59b-47c08368ad68", // The ID of your service instance.
        onLoad: function(instance) { instance.render(); }
      };
      setTimeout(function(){
        const t=document.createElement('script');
        t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" + (window.watsonAssistantChatOptions.clientVersion || 'latest') + "/WatsonAssistantChatEntry.js";
        document.head.appendChild(t);
      });
    </script>
</body>
</html>
```

Agent

```
<!DOCTYPE html>
<html>
<head>
  <title>Customer Care Registry</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
```

```
 <div class="menu">
   <ul>
      <li><a href="#">Solutions</a></li>
      <li><a href="#">Resources</a></li>
      <li><a href="#">About</a></li>
      <li><a href="#">CONTACT</a></li>
   </ul>
  </div>
  <div class="main">
  <div class="navbar">
  <div class="container">
     <div class="title">Agent Details</div>
          <form action="#" method="post">
      <div class="user-details">
          <div class="input-box">
             <span class="details">Name </span> <br>
             <input type="text" name="name" id="name" placeholder="Enter Agent Name" required>
          </div>
          <div class="input-box">
             <span class="details">Email</span><br>
             <input type="text" name="email" id="email" placeholder="Enter your email" required>
          </div>
          <div class="input-box">
```

```
            <span class="details">Password</span><br>
            <input type="password" name="password" id="password" placeholder="Enter your mail password" required>
        </div>
        <div class="input-box">
         <span class="details">Agent Solving Technology</span><br>
         <input type="text" name="Technology" id="Technology" placeholder="Eg:Network problem" required>
            </div>
        <div class="input-box">
         <span class="details">Number of Years Experience</span><br>
         <input type="number" name="experience" id="experience" placeholder="Enter your Experiene in this field" required>
            </div>
        <div class="button">
            <input type="submit" values="register">
        </div>
      </div>
    </form>
  </div>
</div>
</body>
</html>
```

Notepad

```
<!DOCTYPE html>
```

```
<html>
<head>
	<title>About Us Section</title>
	<meta charset="utf-8">
	<meta name="viewport" content="width=device-width, initial-scale=1.0">
	<link rel="stylesheet" href="https://pro.fontawesome.com/releases/v5.10.0/css/all.css">
   <style>
@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400&display=swap');
*{
	margin:0px;
	padding:0px;
	box-sizing: border-box;
	font-family: 'Poppins', sans-serif;
}
.section{
	width: 100%;
	min-height: 100vh;
   background-image: url('https://thumbs.dreamstime.com/z/team-building-group-people-meeting-big-screen-customer-service-business-background-137059808.jpg');
	/* background-color: #ddd; */
}
.container{
```

```
        width: 80%;
        display: block;
        margin:auto;
        padding-top: 100px;
}
.content-section{
        float: left;
        width: 55%;
}
.image-section{
        float: right;
        width: 40%;
}
.image-section img{
        width: 100%;
        height: auto;
}
.content-section .title{
        text-transform:none;
   color:darkkhaki;
        font-size: 28px;
}
.content-section .content h3{
        margin-top: 20px;
        color:cadetblue;
```

```
        font-size: 21px;
}
.content-section .content p{
        margin-top: 10px;
        font-family:'Times New Roman';
        font-size: 18px;
        line-height: 1.5;
    color:#fff;
}
.content-section .content .button{
        margin-top: 30px;
}
.content-section .content .button a{
        background-color: #3d3d3d;
        padding:12px 40px;
        text-decoration: none;
        color:#fff;
        font-size:        25px;
        letter-spacing: 1.5px;
}
.content-section .content .button a:hover{
        background-color:cadetblue;
        color:#fff;
}
.content-section .social{
```

```
        margin: 40px 40px;
}
.content-section .social i{
        color:#a52a2a;
        font-size: 30px;
        padding:0px 10px;
}
.content-section .social i:hover{
        color:#3d3d3d;
}
@media screen and (max-width: 768px){
        .container{
        width: 80%;
        display: block;
        margin:auto;
        padding-top:50px;
}
.content-section{
        float:none;
        width:100%;
        display: block;
        margin:auto;
}
.image-section{
        float:none;
```

```
        width:100%;
}
.image-section img{
        width: 100%;
        height: auto;
        display: block;
        margin:auto;
}
.content-section .title{
        text-align: center;
   color:antiquewhite;
   font-family: 'Times New Roman';
        font-size: 19px;
}
.content-section .content .button{
        text-align: center;
}
.content-section .content .button a{
        padding:9px 30px;
}
.content-section .social{
        text-align: center;
}


}
```

```
</style>


<body>
        <div class="section">
                <div class="container">
                        <div class="content-section">
                                <div class="title">
                                        <h1>About Us</h1>
                                </div>
                                <div class="content">
                                        <h3>Developer<span>                    &
Designer</span></h3>
                                        <p>A web developer or programmer is
someone who takes a web design - which has been created by either a client
or a design team - and turns it into a website.

                I am a front-end developer,Ican provide clean code and pixel
perfect Design. i also make the website more &more interactive with web

                animations.I can provide clean code and pixel perfect
Design.I also make the website more & more interactive with web
animations.A responsive design

                makes your website accessible to all users,regardless of their
device..</p>
                                        <div class="button">
                                                <a href="">Read More</a>
                                        </div>
                                </div>
                                <div class="social">
```

```
                                    <a href=""><i class="fab fa-facebook-f"></i></a>
                                    <a href=""><i class="fab fa-twitter"></i></a>
                                    <a href=""><i class="fab fa-instagram"></i></a>
                                </div>
                            </div>
                            <!-- <div class="image-section">
                                <img src="https://digitaleyecon.com/wp-content/uploads/2021/03/web-design-company-hyderabad.jpeg">
                            </div> -->
                        </div>
                    </div>
                </body>
                </html>
```

Contact

```
<!DOCTYPE html>
<head>
<title> Contact us</title>
<style>
  *, *:before, *:after {
 box-sizing: border-box;
 -webkit-font-smoothing: antialiased;
 -moz-osx-font-smoothing: grayscale;
}
```

```
body {
   background-image:    url('https://thumbs.dreamstime.com/z/team-building-group-people-meeting-big-screen-customer-service-business-background-137059808.jpg');
   font-size: 12px;
}
.navbar{
   width: 1200px;
   height: 75px;
   margin-left: 300px;
   /* margin: auto; */
}
.icon{
   width: 200px;
   float: left;
   height: 70px;
}
.menu{
   width: 400px;
   float: left;
   height: 70px;
}
ul{
   float: left;
```

```
    display: flex;
    justify-content: center;
    align-items: center;
}
ul li{
    list-style: none;
    margin-left: 62px;
    margin-top: 27px;
    font-size: 14px;
}
ul li a{
    text-decoration: none;
    color: #fff;
    font-family: Arial;
    font-weight: bold;
    transition: 0.4s ease-in-out;
}
ul li a:hover{
    color: #ff7200;
}
  body, button, input {
  font-family: 'Montserrat', sans-serif;
  font-weight: 700;
  letter-spacing: 1.4px;
}
```

```
.background {
 display: flex;
 min-height: 100vh;
 }
.container  {
 flex: 0 1 700px;
 margin: auto;
 padding: 10px;
}
 .screen  {
 position: relative;
 background:#ddd;
 border-radius: 15px;
}
 .screen:after {
 content:  '';
 display: block;
 position: absolute;
 top: 0;
 left: 20px;
 right: 20px;
 bottom: 0;
 border-radius: 15px;
 box-shadow: 0 20px 40px rgba(0, 0, 0, .4);
 z-index: -1;
```

```
}
 .screen-header {
  display: flex;
  align-items: center;
  padding: 10px 20px;
  background:azure;
  border-top-left-radius: 15px;
  border-top-right-radius: 15px;
}
 .screen-header-left {
  margin-right: auto;
}
 .screen-header-button {
  display: inline-block;
  width: 8px;
  height: 8px;
  margin-right: 3px;
  border-radius: 8px;
  background: white;
}
 .screen-header-button.close {
  background: #ed1c6f;
}
 .screen-header-button.maximize {
  background: #e8e925;
```

```
}
 .screen-header-button.minimize {
  background: #74c54f;
}
 .screen-header-right {
  display: flex;
}
 .screen-header-ellipsis {
  width: 3px;
  height: 3px;
  margin-left: 2px;
  border-radius: 8px;
  background: #999;
}
 .screen-body {
  display: flex;
}
 .screen-body-item {
  flex: 1;
  padding: 50px;
}
 .screen-body-item.left {
  display: flex;
  flex-direction: column;
}
```

```
.app-title {
 display: flex;
 flex-direction: column;
 position: relative;
 color: #ea1d6f;
 font-size: 26px;
}
.app-title:after {
 content:  '';
 display: block;
 position: absolute;
 left: 0;
 bottom: -10px;
 width: 25px;
 height: 4px;
 background: #ea1d6f;
}
.app-contact {
 margin-top: auto;
 font-size: 8px;
 color: #888;
}
.app-form-group {
 margin-bottom: 15px;
}
```

```
 .app-form-group.message {
  margin-top: 40px;
}
 .app-form-group.buttons {
  margin-bottom: 0;
  text-align: right;
}
.app-form-control {
  width: 100%;
  padding: 10px 0;
  background: none;
  border: none;
  border-bottom: 1px solid #666;
  color:#b9134f;
  font-size: 14px;
  text-transform: uppercase;
  outline: none;
  transition: border-color .2s;
}
 .app-form-control::placeholder {
  color: #666;
}
 .app-form-control:focus {
  border-bottom-color: #ddd;
}
```

```
.app-form-button {
 background: none;
 border: none;
 color: #ea1d6f;
 font-size: 14px;
 cursor: pointer;
 outline: none;
}
.app-form-button:hover {
 color: #b9134f;
}
.credits {
 display: flex;
 justify-content: center;
 align-items: center;
 margin-top: 20px;
 color: #ffa4bd;
 font-family: 'Roboto Condensed', sans-serif;
 font-size: 16px;
 font-weight: normal;
}
.credits-link {
 display: flex;
 align-items: center;
 color: #fff;
```

```
  font-weight: bold;
  text-decoration: none;
}
.dribbble {
  width: 20px;
  height: 20px;
  margin: 0 5px;
}
@media screen and (max-width: 520px) {
  .screen-body {
    flex-direction: column;
  }
.screen-body-item.left {
    margin-bottom: 30px;
  }
.app-title {
    flex-direction: row;
  }
  .app-title span {
    margin-right: 12px;
  }
  .app-title:after {
    display: none;
  }
}
```

```
 @media screen and (max-width: 600px) {
  .screen-body {
    padding: 40px;
  }
 .screen-body-item {
    padding: 0;
  }
}
</style>
</head>
<body>
      <div class="navbar">
       <div class="menu">
            <ul>
               <li><a href="home.html">HOME</a></li>
               <li><a href="About.html">ABOUT</a></li>
               <!-- <li><a href="service.html">SERVICE</a></li> -->
                <li><a href="contact.html">CONTACT</a></li>
               <li><a href="profile.html">MyProfile</a></li>
            </ul>
       </div>
<div class="background">
   <div class="container">
     <div class="screen">
       <div class="screen-header">
```

```
        <div class="screen-header-left">
          <div class="screen-header-button close"></div>
          <div class="screen-header-button maximize"></div>
          <div class="screen-header-button minimize"></div>
        </div>
        <div class="screen-header-right">
          <div class="screen-header-ellipsis"></div>
          <div class="screen-header-ellipsis"></div>
          <div class="screen-header-ellipsis"></div>
        </div>
      </div>
      <div class="screen-body">
        <div class="screen-body-item left">
          <div class="app-title">
            <span>CONTACT</span>
            <span>US</span>
          </div>
          <div class="app-contact">CONTACT INFO : +92 333 0000</div>
        </div>
        <div class="screen-body-item">
          <div class="app-form">
            <div class="app-form-group">
              <input class="app-form-control" placeholder="NAME"
value="Customer Care">
            </div>
```

```
            <div class="app-form-group">
              <input class="app-form-control" placeholder="EMAIL">
            </div>
            <div class="app-form-group">
              <input class="app-form-control" placeholder="CONTACT NO">
            </div>
            <div class="app-form-group message">
              <input class="app-form-control" placeholder="MESSAGE">
            </div>
            <div class="app-form-group buttons">
              <button class="app-form-button">CANCEL</button>
              <button class="app-form-button">SEND</button>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
 </body>
 </html
```

App

```
from flask import Flask, render_template, request
```

```
from mydb import connect
from flask_mysqldb import MySQL
from flask_mail import Mail,Message
import os
import ibm_db

app = Flask(_name_)

#database connection
def list_all():
   sql= "SELECT * from userregister"
   stmt = ibm_db.exec_immediate(conn, sql)
   dictionary = ibm_db.fetch_both(stmt)
   while dictionary != False:
     print ("The Name is : ",  dictionary["NAME"])
     print ("The email is : ", dictionary["EMAIL"])
     print ("The password is : ", dictionary["PASSWORD"])
     print ("The phone number is : ", dictionary["mobile number"])
     dictionary = ibm_db.fetch_both(stmt)
def insert_values(name,email,password,mobilenumber):
    conn                                                    =
ibm_db.connect("DATABASE=bludb;HOSTNAME=fbd88901-
ebdb-4a4f-a32e-
9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;
```

```
PORT=32731;PROTOCOL=TCPIP;SECURITY=SSL;SSLServer
Certificate=DigiCertGlobalRootCA.crt;UID=zhc67031;PWD=cMr
I3QzWfo581pJ2;", "", "")

   sql = "INSERT INTO USERREGISTER
VALUES('{}','{}','{}','{}')".format(name,email,password,mobilenu
mber)

   stmt = ibm_db.exec_immediate(conn,sql)

   print ("Number of affected rows: ", ibm_db.num_rows(stmt))

#connect or not

try:

  conn                                                    =
ibm_db.connect("DATABASE=bludb;HOSTNAME=fbd88901-
ebdb-4a4f-a32e-
9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;
PORT=32731;PROTOCOL=TCPIP;SECURITY=SSL;SSLServer
Certificate=DigiCertGlobalRootCA.crt;UID=zhc67031;PWD=cMr
I3QzWfo581pJ2;", "", "")

  print("connection succesful")

except:

   print("not connected")

#rendering home page

@app.route("/")

def homepage():

   return render_template("login.html")

#checking values in the login page

@app.route("/login",methods=['POST'])
```

```
def login():
    email=request.form['email']
    pwd=request.form['pwd']
    if not re.match(r'[^@]+@[^@]+\.[^@]+', email):
            msg = 'Invalid email address !'
            return render_template('login.html',signupmsg=msg)
        if pwd != pwd:
            msg = 'Please enter correct confirm password'
            return render_template('login.html',signupmsg=msg)
#entering values are stored
result = ibm_db.exec_immediate(conn,f"SELECT * FROM userregister WHERE email LIKE '{email}'")
existing_user = ibm_db.fetch_row(result)


#get data in the registering page to enter
#store in the database
@app.route("/receivedata",methods=['POST'])
def result():
    name= request.form['name']
    email = request.form['email']
    password = request.form['password']
    phonenumber = request.form['phonenumber']
    connect.insert_values(name,email,password,phonenumber)
```

```
    return("Data received")
#dont have an account sign up the page
@app.route("/sign up",methods=['POST'])
def signup():
    return render_template("Ticket.html")
#rendering ticket page for say their issue
@app.route("/ticket",methods=['POST'])
def result():
        name=request.form['name']
        email = request.form['email']
        Productcategory = request.form['product category']
        ProblemDescription = request.form['Problem Decsription']
        return render_template("Ticket.html",message="Changes saved sucessfully!")
#insert data in the query
query = "SELECT * FROM customerprobem WHERE email=? AND passwrd=?"
          stmt = ibm_db.prepare(conn, query) # type:ignore
          ibm_db.bind_param(stmt,1,mail) # type:ignore
          ibm_db.bind_param(stmt,2,password) # type:ignore
          ibm_db.execute(stmt) # type:ignore
          user = ibm_db.fetch_assoc(stmt) # type:ignore
          print(user,password)
```

```
        if user:
            session["username"] = user['USERNAME']
            session['mail'] = mail
            return
render_template('agent.html',username=session["username"],mail=
session["mail"])
        else:
            msg = 'mail or password is not valid.'
            return render_template('home.html',signinmsg=msg)
        if request.method == "GET":
            return redirect(url_for('home'))
#agents info table
@app.route("/agent",methods=['POST'])
def result():
    name=request.form['name']
    email = request.form['email']
    password = request.form['password']
    technology = request.form['technology']
    experience = request.form['experience']
    return render_template("agent.html",message="Changes saved
sucessfully!")


  if not name == session["username"] or not mail ==
session["mail"]:
```

```
            msg = "please don't change username and mail."
            return render_template('home.html',msg=msg)
    result = ibm_db.exec_immediate(conn,f"INSERT INTO agent
(username,email,Technology,solution)
VALUES('{name}','{mail}','{technology}','{solution}','{'0'}')")

        sendemail(mail,'complaint_creation')
        msg = 'Complaint registerd you check out complaints section.'
        return render_template('home.html',msg=msg)

#for generating email support
app.config['MAIL_SERVER'] = 'smtp.gmail.com'
app.config['MAIL_PORT'] ='465'
app.config['MAIL_USERNAME']= 'abc@gmail.com.com'
app.config['MAIL_PASSWORD']= 'abcdef'
app.config['MAIL_USE_TLS'] = False
app.config['MAIL_USE_SSL'] = True
mail = Mail(app)

#to get and send the mail for confirming my mail
@app.route('/home',methods=['GET','POST'])
@app.route('/',methods=['GET','POST'])
def home():
```

```
    if request.method == 'POST':
        msg = Message("Customer Care", sender='CustomerCareRegistry@gmail.com',recipients=['email'])
        msg.body = "You are Successfully Registered in your account"
        return "Send email."
    return render_template('Agent.html')


if _name=="main_":
    app.run(debug=True)
```

GitHub: IBM-EPBL/IBM-Project-15020-1659593376: Customer Care Registry (github.com)

Demo link: https://drive.google.com/file/d/1pQrQPmSFGKZghoZxztpjiralqG6tlV4L/view?usp=drivesdk