

# **CUSTOMER CARE REGISTRY**

## **PROJECT REPORT**

*Submitted by*

**PONNAIAH KARTHIK R M (19EUCS104)**

**PRAVEEN KANTH K (19EUCS109)**

**SAMUEL WYCLIFFE J (19EUCS117)**

**PRASANNA J (19EUCS108)**

*in partial fulfillment of the requirements for the award of the degree*

*of*

**BACHELOR OF ENGINEERING**

*in*

**COMPUTER SCIENCE AND ENGINEERING**

**SRI KRISHNA COLLEGE OF ENGINEERING AND TECHNOLOGY**

**COIMBATORE**

**(An Autonomous Institution)**



**ANNA UNIVERSITY: CHENNAI**

**MAY 2022**

# **SRI KRISHNA COLLEGE OF ENGINEERING AND TECHNOLOGY**

**(An Autonomous Institution)**

**(Approved by AICTE and Affiliated to Anna University, Chennai)**

**ACCREDITED BY NAAC WITH “A” GRADE**

## **BONAFIDE CERTIFICATE**

Certified that this project report titled **Customer Care Registry** is the bonafide work of **PONNAIAH KARTHIK R M (19EUCS104), PRAVEEN KANTH K (19EUCS109), SAMUEL WYCLIFFE (19EUCS117) PRSANNA J (19EUCS108)** who carried out the project work under my supervision.

SIGNATURE

**Dr.K. SASI KALA RANI, M.E., Ph.D.,**

**HEAD OF THE DEPARTMENT**

SIGNATURE

**MS. POORANAM N**

**SUPERVISOR**

Department of Computer Science and Engineering  
Sri Krishna College of Engineering and Technology  
Kuniamuthur,  
Coimbatore

**This project report is submitted for the Autonomous Project Viva-Voce examination held  
on .....**

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

## **ACKNOWLEDGEMENT**

We express our sincere thanks to the management and **Dr.J.JANET, M.E.,Ph.D.,** Principal, Sri Krishna College of Engineering and Technology, Coimbatore for providing us the facilities to carry out this project work.

We are highly indebt to **Dr.K. SASIKALA RANI, M.E.,Ph.D.,** Head of Computer Science and Engineering for her continuous evaluation, valuable suggestions and comments given during the course of the project work.

We are thankful to Project Co-Ordinator, **Dr.P. MOHAN KUMAR, M.E., Ph.D.,** Department of Computer Science and Engineering for her continuous evaluation, valuable suggestions and comments given during the course of the project work.

We express our deep sense of gratitude to our guide, **MS. POORANAM N, M.E.,** Professor in the department of Computer science and Engineering for her valuable advice, guidance and support during the course of our project work.

By this, we express our heartfelt sense of gratitude and thanks to our beloved parents, family and friends who have all helped in collecting the resources and materials needed for this project and for their support during the study and implementation this project.

## **TABLE OF CONTENTS**

<b>CHAPTER NO</b>	<b>TITLE ABSTRACT</b>	<b>PAGE NO</b>
<b>1</b>	<b>INTRODUCTION</b>  1.1 Project Overview 1.2 Purpose	
<b>2</b>	<b>LITERATURE SURVEY</b>  2.1 Existing problems 2.2 References 2.3 Problem Statement Definition	
<b>3</b>	<b>IDEATION &amp; PROPOSED SOLUTION</b>  3.1 Empathy Map Canvas 3.2 Ideation & Brainstorming 3.3 Proposed Solution 3.4 Problem Solution fit	
<b>4</b>	<b>REQUIREMENT ANALYSIS</b>  4.1 Functional requirement 4.2 Non-Functional requirements	
<b>5</b>	<b>PROJECT DESIGN</b>  5.1 Data Flow Diagrams 5.2 Solution & Technical Architecture 5.3 User Stories	
<b>6</b>	<b>PROJECT PLANNING &amp; SCHEDULING</b>  6.1 Sprint Planning & Estimation 6.2 Sprint Delivery Schedule	

### 6.3 Reports from JIRA

## **7 CODING & SOLUTIONING**

7.1 Feature 1 : Register Page

7.2 Feature 2 : Login Page

7.3 Feature 3 : Dashboard

7.4 Feature 4 : Stats Page

## **8 TESTING**

8.2 Test Cases

8.3 User Acceptance Testing

## **9 RESULTS**

9.1 Performance Metrics

## **10 ADVANTAGES & DISADVANTAGES**

## **11 CONCLUSION**

## **12 FUTURE SCOPE**

## **13 APPENDIX**

# **CHAPTER 1**

## **INTRODUCTION**

Managing Customer Queries is a tedious task as the growth of internet and users across all the fields of customer management is one of the important aspects that companies focus on for better management of customer queries. Customer service is reactive. Customer care and customer service together help create a positive customer experience, or the overall impression a person has when interacting with your company. Both are vital, but there are subtle differences in how they are implemented.

### **1.1 Project overview:**

High-quality customer care is proactive. The needs of customers throughout the buyer's journey are anticipated, making customers feel supported. That, in turn, helps create an emotional connection between the customer and the company. Here, the focus is on helping customers solve problems or answer questions before purchase, either in a self-serve fashion or via the customer support team. To achieve the best customer service, your customer service team should address customer needs quickly and with as few customer interactions as possible. customer service is the most impactful driver category for the overall customer experience in B2B companies, eclipsing categories like products and prices. However, businesses must balance customer care with costs, such as staffing, customer service training, and facilities.

### **1.2 Purpose**

Customer care registry is vital. 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 service is the support you offer your customers both before and after they buy and use your products or services that helps them have an easy and enjoyable experience with you. The customer service team is the face of the organization and the frontline when customers require assistance.

## CHAPTER 2

### LITERATURE SURVEY

#### 2.1 Existing Problem

- Managing customer relations is a tedious task.
- Lack of integrated dashboard in the market.
- Tracking of tickets is complex.
- Updates on progress are not available.

#### 2.2References

1. Adamson, I., Chan, K.-M., & Handford, D. (2003). Relationship marketing: Customer commitment and trust as a strategy for the smaller Hong Kong corporate banking. *International Journal of Bank Marketing*, 21(6/7), 347–358.
2. Adiwijaya, K., Wahyuni, S., & Mussry, J. S. (2017). Marketing ambidexterity and marketing performance: Synthesis, a conceptual framework, and research propositions. In *Enhancing Business Stability Through Collaboration* 431–442. CRC Press.
3. Afèche, P., Araghi, M., & Baron, O. (2017). Customer acquisition, retention, and service access quality: Optimal advertising, capacity level, and capacity allocation. *Manufacturing & Service Operations Management*, 19(4), 674–691.
4. Aggarwal, C. C., & Yu, P. S. (2002). Finding localized associations in market basket data. *IEEE Transactions on Knowledge and Data Engineering*, 14, 51–62.
5. Alhakimi, W., & Ghaleb, A. (2019). The impact of CRM components system on customer retention in the telecom industry: A case of Y-Telecom in Yemen. *Middle East Journal of Management*, 6(4), 378–409.
6. Al-Omoush, K. S., Simón-Moya, V., Atwah Al-ma'aitah, M., & Sendra-García, J. (2021). The determinants of social CRM entrepreneurship: An institutional perspective. *Journal of Business Research*, 132, 21–31.
7. Canhoto, A. I., Meadows, M., Ball, K., Daniel, E., Dibb, S., & Spiller, K. (2017). The role of customer management capabilities in public–private partnerships. *Journal of Strategic Marketing*, 25(5–6), 384–404.

8. Chang, C. W., & Zhang, J. Z. (2016). The effects of channel experiences and direct marketing on customer retention in multichannel settings. *Journal of Interactive Marketing, 36*, 77–90.
9. Chen, C., Geng, L., & Zhou, S. (2020). Design and implementation of bank CRM system based on decision tree algorithm. *Neural Computing and Applications, 1–11*.
10. Cricelli, L., Famulari, F. M., Greco, M., & Grimaldi, M. (2020). Searching for the one: Customer relationship management software selection. *Journal of Multi-Criteria Decision Analysis.*
11. Drew, J. H., Mani, D. R., Betz, A. L., & Datta, P. (2001). Targeting customers with statistical and data-mining techniques. *Journal of Service Research, 3*, 205–220.
12. Fidel, P., Schlesinger, W., & Cervera, A. (2015). Collaborating to innovate: Effects on customer knowledge management and performance. *Journal of Business Research, 68*(7), 1426–1428.
13. Foltean, F. S., Trif, S. M., & Tuleu, D. L. (2019). Customer relationship management capabilities and social media technology use: Consequences on firm performance. *Journal of Business Research, 104*, 563–575.
14. Fu, H. P., & Chang, T. S. (2016). An analysis of the factors affecting the adoption of cloud consumer relationship management in the machinery industry in Taiwan. *Information Development, 32*(5), 1741–1756.
15. Ngai, E. W. (2005). Customer relationship management research (1992–2002) An academic literature review and classification. *Marketing Intelligence & Planning, 23*(6), 582–605.

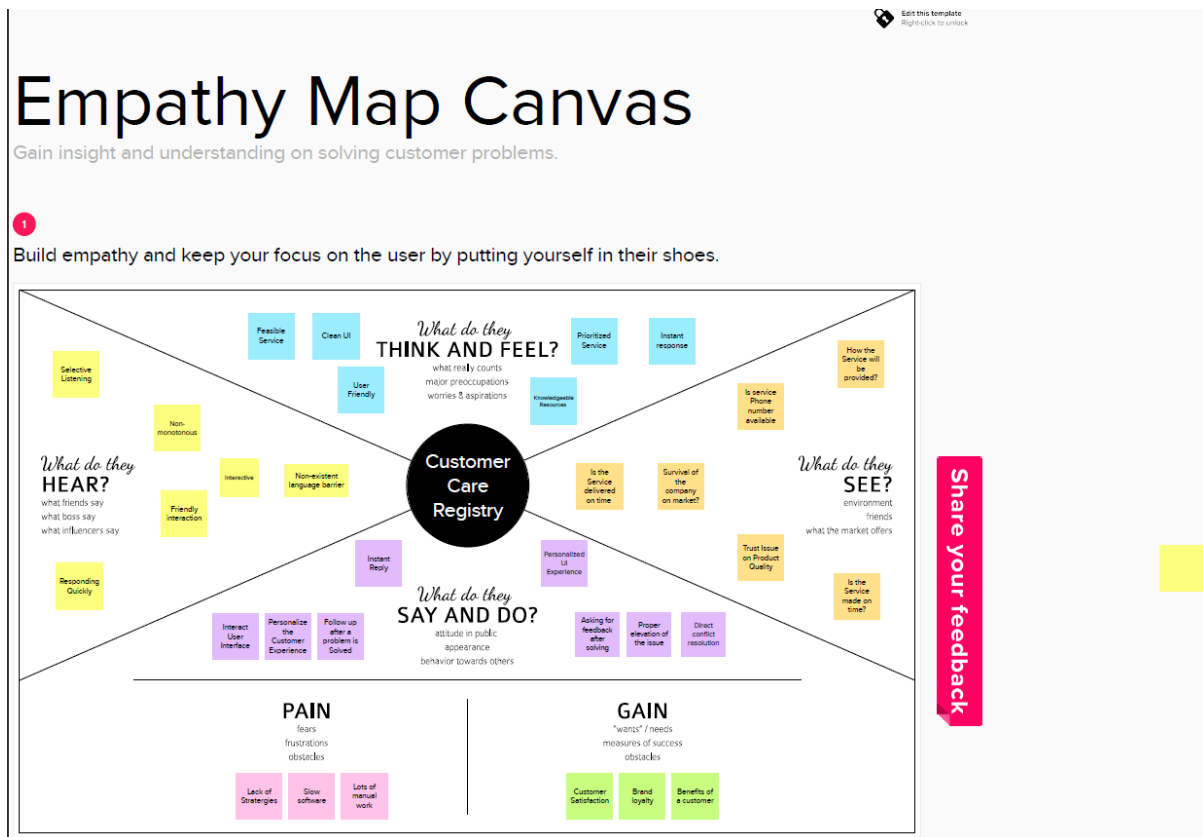
## 2.3 Problem Statement Definition

- The goal is to solve the issues raised by the customer by means of providing domain experts.
- We need to help the customer in processing their complaints. The customers can raise the ticket with a detailed description of the issue. An Agent will be assigned to the Customer to solve the problem. Whenever the agent is assigned to a customer they will be notified with an email alert. Customers can view the status of the ticket till the service is provided.

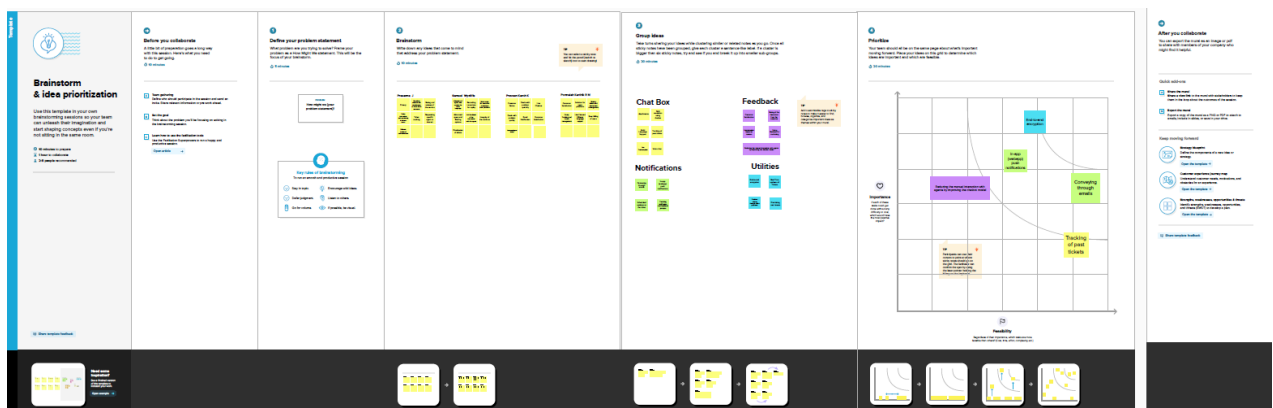


## IDEATION AND PROPOSED SOLUTION

### 3.1 Empathy Map Canvas



### 3.2 Ideation & Brainstorming



### 3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To Create a Platform which helps Resolve Customer issues using Cloud
2.	Idea / Solution description	The solution involves creating a web app that does proper matchmaking with the right experts based on the issue at hand.
3.	Novelty / Uniqueness	Webapp bootstrapped with features such as attachments, voice chats, automated chatbot. and lots of QOL features, all present in a single application.
4.	Social Impact / Customer Satisfaction	Tickets get resolved based on the priority which will be useful in solving critical issues.
5.	Business Model (Revenue Model)	Prioritised tickets and support, matching with the right experts, help to generate stable revenue.
6.	Scalability of the Solution	The Project is implemented using cloud so that it is highly scalable and can be used by multiple users based on the requirements.

### 3.4 Problem Solution fit

Define CS, fit into CC	<b>1. CUSTOMER SEGMENT(S)</b> <span>CS</span> Who is your customer? 1) Is a person in need of assistance. 2) Having a hard time explaining the issue. 3) People who are stuck in the middle of the process.	<b>6. CUSTOMER</b> <span>CC</span> 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) Spending time on explaining the issue. 2) Language Barriers. 3) Probability of getting the issue resolved.	<b>5. AVAILABLE SOLUTIONS</b> <span>AS</span> Which solutions are available to the customers when they face the problem Search for online solutions in forums and 1) Calling Customer care agents. 2) Referring user Manuals. 3) Asking for expert's help.
	<b>2. JOBS-TO-BE-DONE / PROBLEMS</b> <span>J&amp;P</span> 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 helps the customer to find a solution for the issue. 2) The issue will be prioritized based on their severity. 3) Customer can use attachments as well as voice chat for communication	<b>9. PROBLEM ROOT CAUSE</b> <span>RC</span> 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) Customers don't know how to understand their issue. 2) Lack of knowledge in using the project. 3) Problem faced by using the service.	<b>7. BEHAVIOUR</b> <span>BE</span> What does your customer do to address the problem and get the job done? 1) Reads the manual. 2) Take proper time in formulating the issue. 3) Seeking direct help from peers.
<b>3. TRIGGERS</b> <span>TR</span> What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news. 1) Undesired feedback from the service. 2) Deadlines.	<b>10. YOUR SOLUTION</b> <span>SL</span> 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. Resolving client issues using web app based on the pre-defined guidelines designed for specific problems and tracking of all the issues of the client and their status.	<b>8. CHANNELS of BEHAVIOUR</b> <span>CH</span> <b>8.1 ONLINE</b> What kind of actions do customers take online? Extract online channels from #7 Search for online solutions in forums and report. <b>8.2 OFFLINE</b> What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development. Ask for peers help or approach physical stores	Extract online & offline CH of BE
<b>4. EMOTIONS: BEFORE / AFTER</b> <span>EM</span> 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. Frustrated, Irritated > Satisfied, Contended.			



Problem-Solution fit canvas is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 license Created by Datta Nigrahkumar / Anantama.com



## CHAPTER 4

### REQUIREMENT ANALYSIS

#### 4.1 Functional requirement

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 via Google Login with Email id and Password
FR-4	Raise Ticket	Description of the issues Contact information
FR-5	Notifications	Updates on tickets through email and in-app
FR-6	Interaction Room	A chatbox to have conversation 1 to 1 with the matched agent, send attachments, voice chat and live transcriptions
FR-7	Feedback	Customer Feedback Form

#### 4.2 Non-Functional requirements

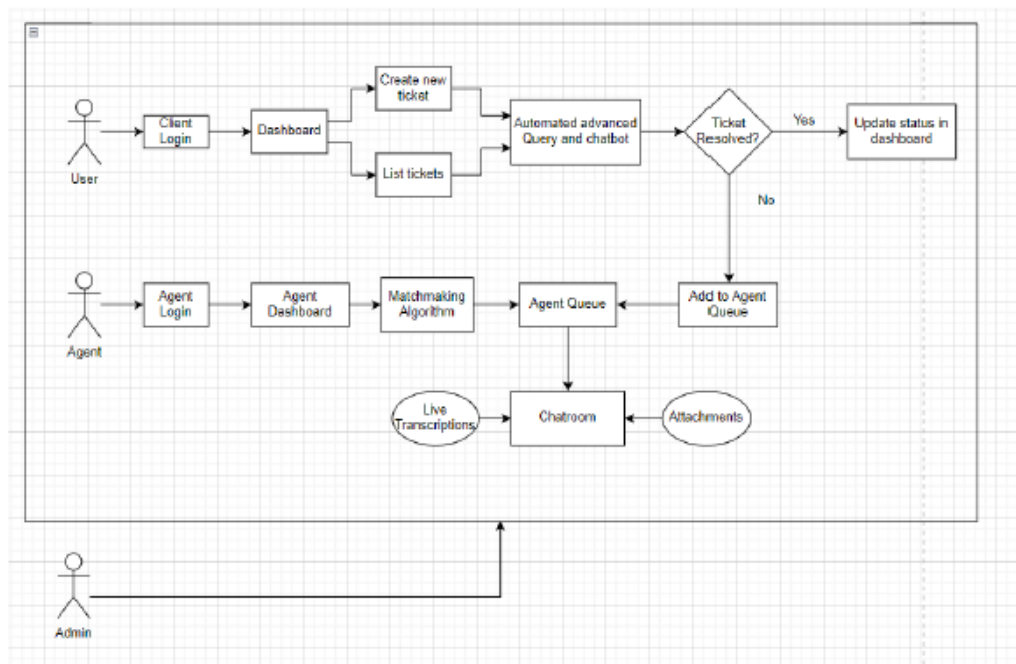
FR No.	Non-Functional Requirement	Description
NFR-1	<b>Usability</b>	A personalized Web App with intuitive UI for ease of access
NFR-2	<b>Security</b>	Authentication of accounts and end to end Encryption
NFR-3	<b>Reliability</b>	Reliable servers hosted on cloud and tested project
NFR-4	<b>Performance</b>	Fast latency times, by load balancing and evenly distributing workloads across clusters
NFR-5	<b>Availability</b>	24/7 service
NFR-6	<b>Scalability</b>	Agents scalability as per the number of customers.

## CHAPTER 5

### PROJECT DESIGN

#### 5.1 Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

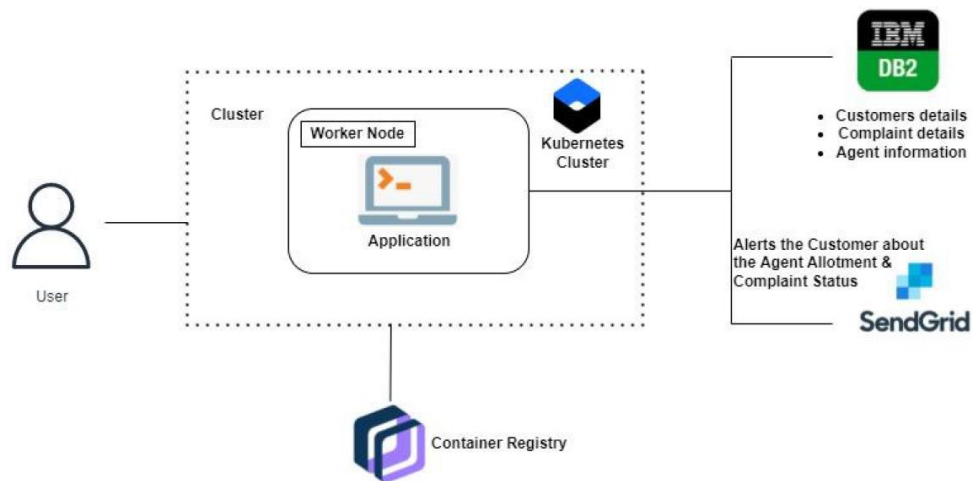


#### 5.2 Solution & Technical Architecture

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.

Provide specifications according to which the solution is defined, managed, and delivered.



## 5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web 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
	Authentication	USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	Medium	Sprint-1
	Login	USN-3	As a user, I can log into the application by entering email & password	I can view the dashboard	High	Sprint-1
	Dashboard	USN-4	As a customer, I can do manipulation of tickets, checking their status	I can add, remove, edit tickets	High	Sprint-2
	Create new Ticket	USN-5	As a user, we can raise new ticket to solve the problem faced	I can use the chatbot	High	Sprint-2
	Chatbot	USN-6	As a user, I can chat with the solutions providing chat solutions based on the description of the issue.	providing list of troubleshooting solutions	Low	Sprint-2
Agent(Web user)	Login	USN-1	As an Agent, I can login with my given credentials from admin	I can access my account / dashboard	High	Sprint-3
	Dashboard	USN-2	As an Agent, I can see the list of tickets assigned to me from the matchmaking algorithm	I can get real-time updates on the status of list of tickets assignment to me	High	Sprint-3
	Ticket Queue	USN-3	As an Agent, I can view the pending tickets and solve them in order of the queue	I can communicate with the client and speak about the tickets and help the client	Medium	Sprint-4
Administrator	Login	USN-1	As an Administrator, I can login using a predefined set of credentials, given to me by the software vendor	I can access my account/dashboard	Medium	Sprint-4
	Manage	USN-2	As an Administrator, I can manage all the tickets raised and the queries solved in the portal with special rights across the functionalities of the web app	I can access all the queries and current process in the dashboard	High	Sprint-4

## CHAPTER 6

### PROJECT PLANNING & SCHEDULING

#### 6.1 Sprint Planning & Estimation

Product Backlog, Sprint Schedule, and Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	User Panel	USN-1	The user will login into the portal, interact and avail the services present	20	High	PRASANNA J PRAVEEN KANTH K Ponnaiah Karthik R M
Sprint-2	Admin panel	USN-2	The admin's role is to monitor and oversee the status of the portal	20	High	SAMUEL WYCLIFFE J Ponnaiah Karthik R M
Sprint-3	Chat Bot	USN-3	The user can directly talk to Chatbot regarding the services. Get the recommendations based on information provided by the user.	20	High	PRAVEEN KANTH K PRASANNA J
Sprint-4	final delivery	USN-4	Container of applications using docker kubernetes and deployment the application.Create the documentation and final submit the application	20	High	Ponnaiah Karthik R M PRAVEEN KANTH K PRASANNA J SAMUEL WYCLIFFE J

#### 6.2 Sprint Delivery Schedule :

Project Tracker, Velocity & Burndown Chart

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	7 Days	21 Oct 2022	28 Oct 2022		29 Oct 2022
Sprint-2	20	8 Days	29 Oct 2022	05 Nov 2022		04 Nov 2022
Sprint-3	20	7 Days	05 Nov 2022	11 Nov 2022		11 Nov 2022
Sprint-4	20	5 Days	13 Nov 2022	17 Nov 2022		18 Nov 2022

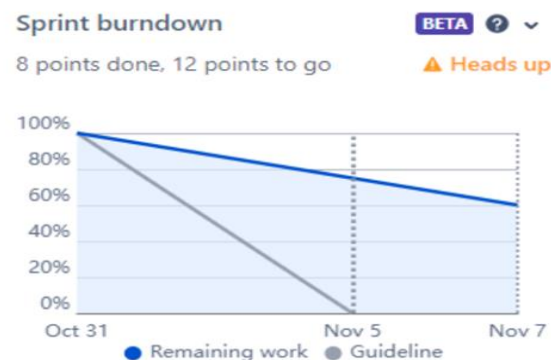
## Velocity:

Velocity is a metric that predicts how much work an Agile software development team can successfully complete within a two-week sprint (or similar time-boxed period). Velocity is a useful planning tool for estimating how fast work can be completed and how long it will take to complete a project

Average velocity = Total story points/ No. of iterations = 80/4 = 20

## Burndown Chart:

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



## 6.3 Reports from JIRA

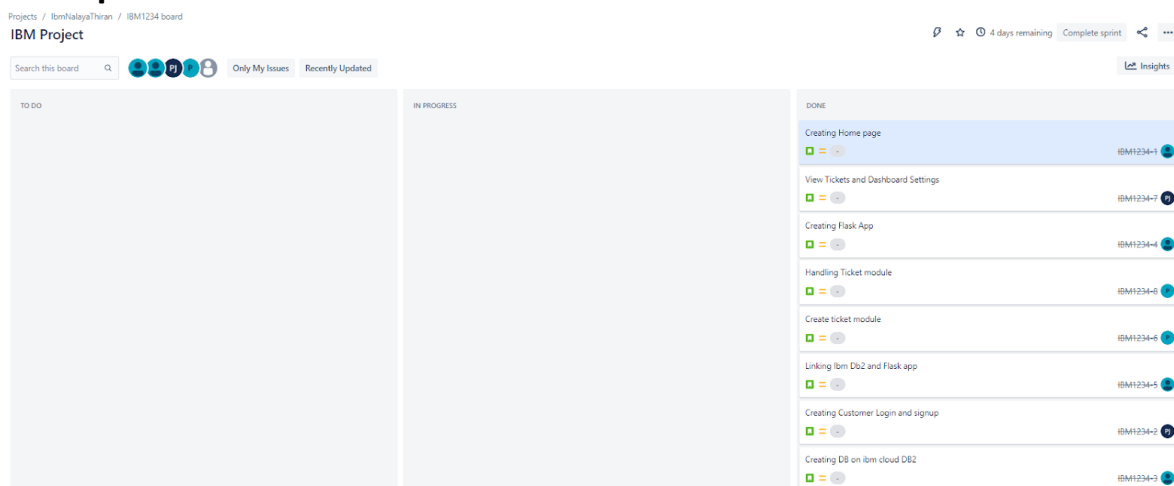
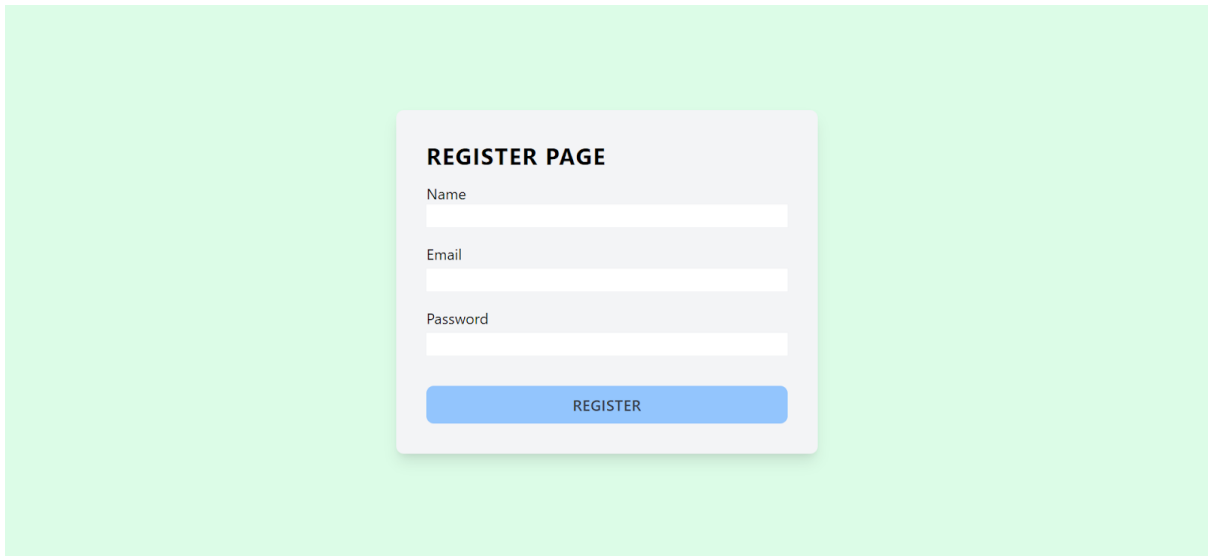


Figure 6.1 Sprint Dashboard

## CHAPTER 7

### CODING & SOLUTIONING

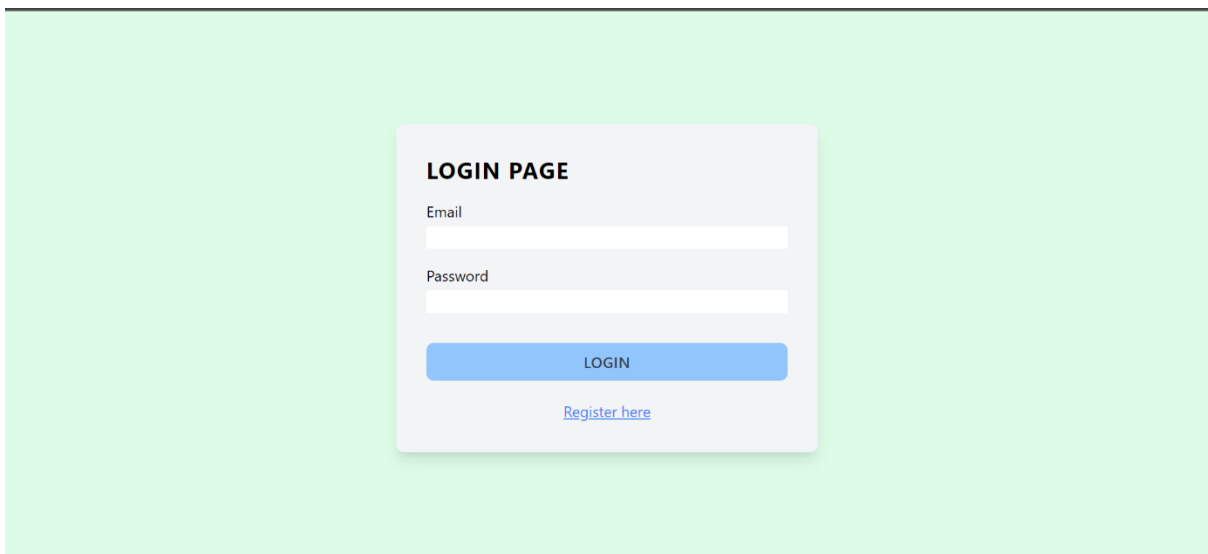
#### 7.1 Feature 1



A screenshot of a customer registration page. The page has a light green background. In the center, there is a white rectangular box with a light gray border and a subtle shadow. Inside this box, the title "REGISTER PAGE" is displayed in bold black text. Below the title, there are three input fields: "Name", "Email", and "Password", each with a light gray label and a white input area. At the bottom of the box is a blue button with the text "REGISTER" in white capital letters.

Figure 7.1 Customer Registration page

#### 7.2 Feature 2



A screenshot of a customer login page. The page has a light green background. In the center, there is a white rectangular box with a light gray border and a subtle shadow. Inside this box, the title "LOGIN PAGE" is displayed in bold black text. Below the title, there are two input fields: "Email" and "Password", each with a light gray label and a white input area. At the bottom of the box is a blue button with the text "LOGIN" in white capital letters. Below the button, there is a blue hyperlink that says "Register here".

Figure 7.2 Customer Login page



## 7.3 Feature 3

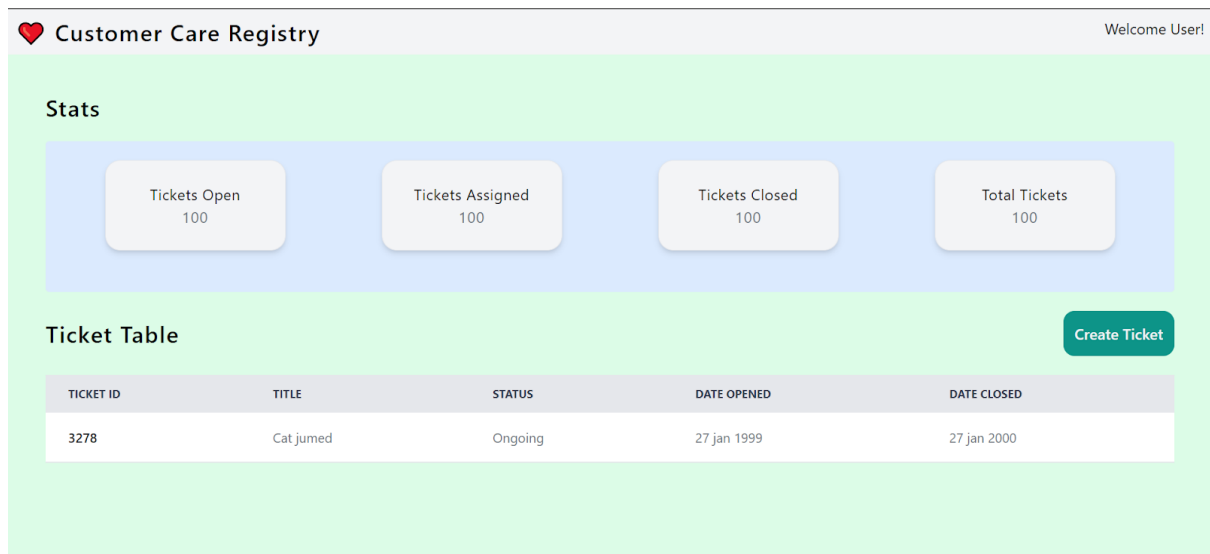


Figure 7.3 Customer Dashboard

## 7.4 Feature 4

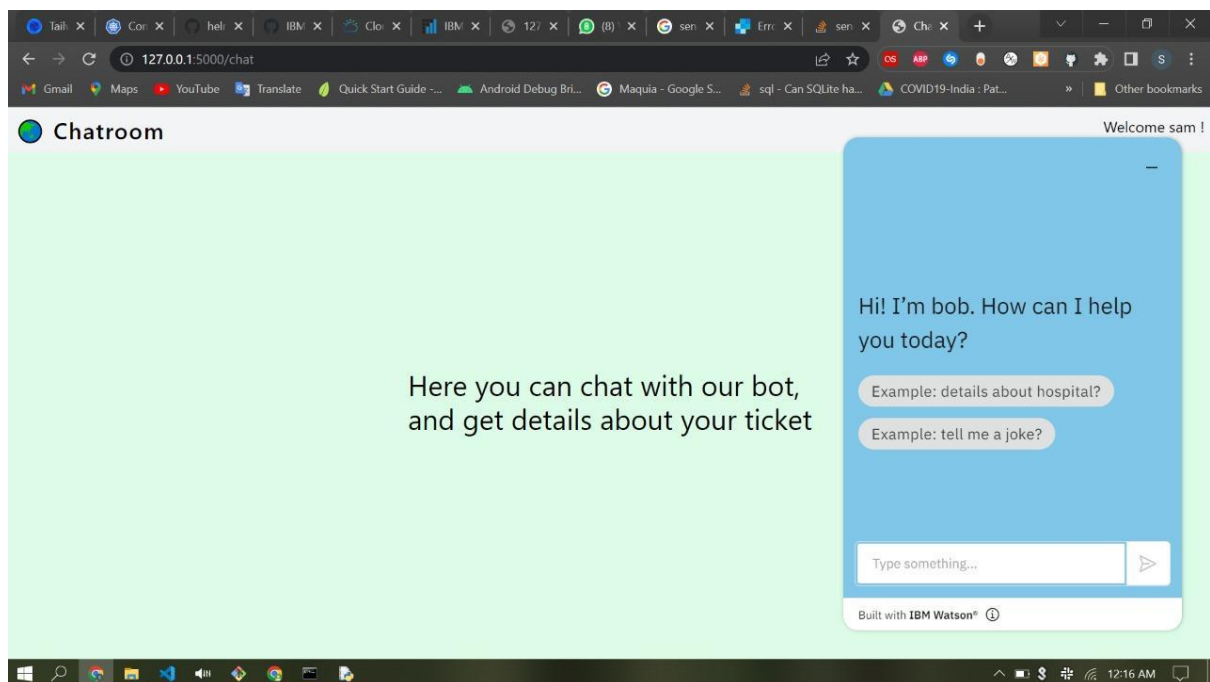


Figure 7.4 ChatBot Page

## 7.5 Feature 5

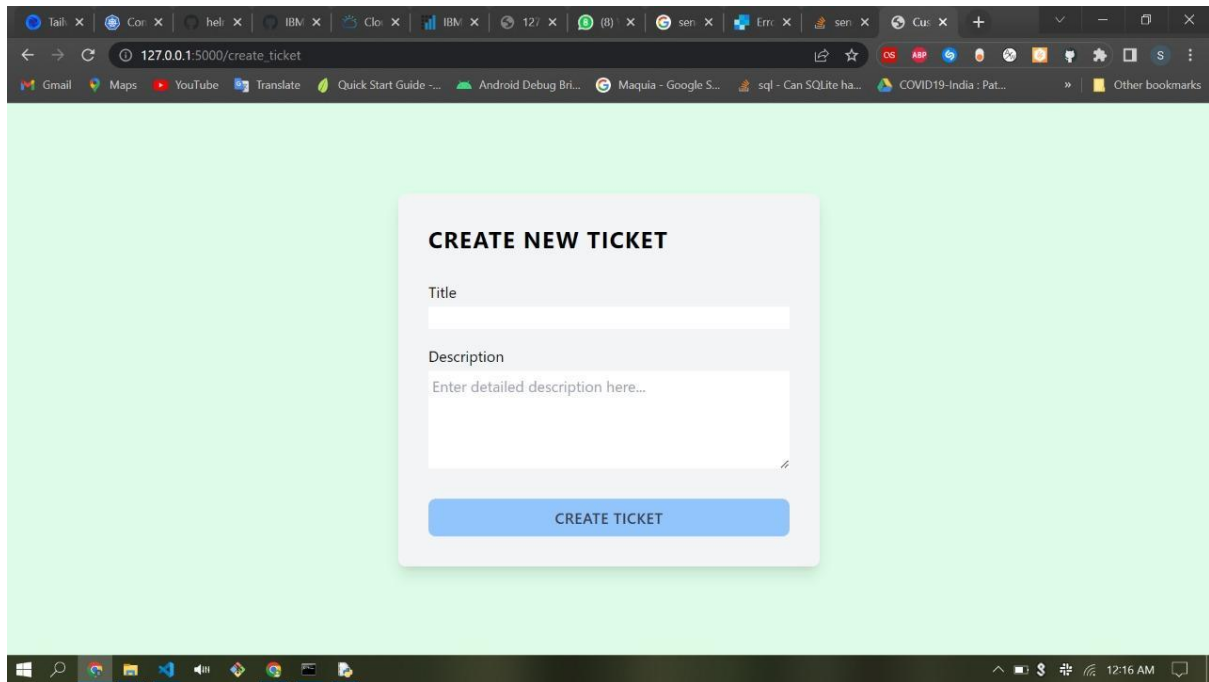


Figure 7.1 New Ticket Page

## CHAPTER 8

### TESTING

#### 8.1 TEST CASES

1. Login button click with wrong credentials entered.
2. Signup with wrong form data entered.
3. Entering home page with logged out session.
4. Creating Ticket without content.

#### 8.2 USER ACCEPTANCE TESTING

s.no	Test Case id	Feature Type	component	Test description	Input test Data	Actual output	Expected output	remarks
1	TC – RG 01	Functional	Register page	register for the application by entering email, password	<a href="mailto:test@gmail.com">test@gmail.com</a> *****	Registration successful	Registration successful	pass
2	TC – SI 01	Functional	Login page	log into the application by entering email & password	<a href="mailto:test@gmail.com">test@gmail.com</a> *****	Login successful	Login successful	pass

3	TC – ST 01	UI	Dashboard page	view my history of tickets of the user		All the tickets are displayed	All the tickets are displayed	pass
4	TC – SI 02	Functiona l	Dashboard	can logout		Go to sign page	Sign in page displayed	pass
5	TC – ST 02	Functiona l	Create Ticket page	create new ticket	Ticket Title  Ticket description	Ticket  created	Ticket  created	pass
6	TC – ST 03	Functiona l	Dashboard page	can delete ticket		Ticket deleted	Ticket deleted	pass
7	TC – ST 04	Functiona l	Home page	can use chatbot		chatBot initiated	chatBot initiated	pass
8	TC – ST 05	Functiona l	Tickets page	updates on tickets is being sent to email		email received successfully	email received successfully	pass

## **CHAPTER 9**

### **RESULTS**

#### **9.1 PERFORMANCE METRICS**

1. Hours worked : 50 hours
2. Stick to Timelines : 100%
3. Consistency of the product : 75%
4. Efficiency of the product : 80%
5. Quality of the product : 85%

## **CHAPTER 10**

### **ADVANTAGES & DISADVANTAGES**

#### **Advantages**

- To get expertise solutions for customer issues and complaints.
- Makes it easy for customers to do business with you.
- Less prone to errors due to expert guidance.
- Effective sharing of information
- Prevention of unnecessary searches to find people and make them understand the issue in hand, for a solution.
- Saves time, by providing expert guidance in a single port
- Automation of ticket status, progress and report generation, makes it easy for customers to get a high-level overview of the given solution.

#### **Disadvantages**

##### **Time and cooperation requirement**

The customer is required to cooperate with the agent after creating the ticket. Any discrepancies or inactiveness from the customer side leads to further delaying the issue. At the worst cases, sometimes, not converging to a proper solution.

## **Data Safety**

Data security is a big challenge that needs to be taken care of during the development phase of the application. Data storage is usually the target of hackers. This endangers the safety of customer data. Organisations are very much concerned about the safety of customers' personal data. For this, all CRM applications must meet the requirement for data security and be HIPAA compliant before they can be deployed for public services.

## **CHAPTER 11 CONCLUSION**

Cloud computing is the delivery of different services through the Internet. It is a popular option for people and businesses for a number of reasons including cost savings, increased productivity, speed and efficiency, performance, and security. Cloud-based storage makes it possible to save files to a remote database and retrieve them on demand. It takes all the heavy lifting involved in crunching and processing data away from the device you carry around or sit and work at. It also moves all of that work to huge computer clusters far away in cyberspace. The Internet becomes the cloud, and voilà—your data, work, and applications are available from any device with which you can connect to the Internet, anywhere in the world.

We've integrated automated mailing systems, and made a highly scalable application.

## **CHAPTER 12 FUTURE SCOPE**

Cloud computing has grown into a major paradigm in the tech world. It enables ubiquitous and simple on-demand access to shared computing resources via configurable Internet services.

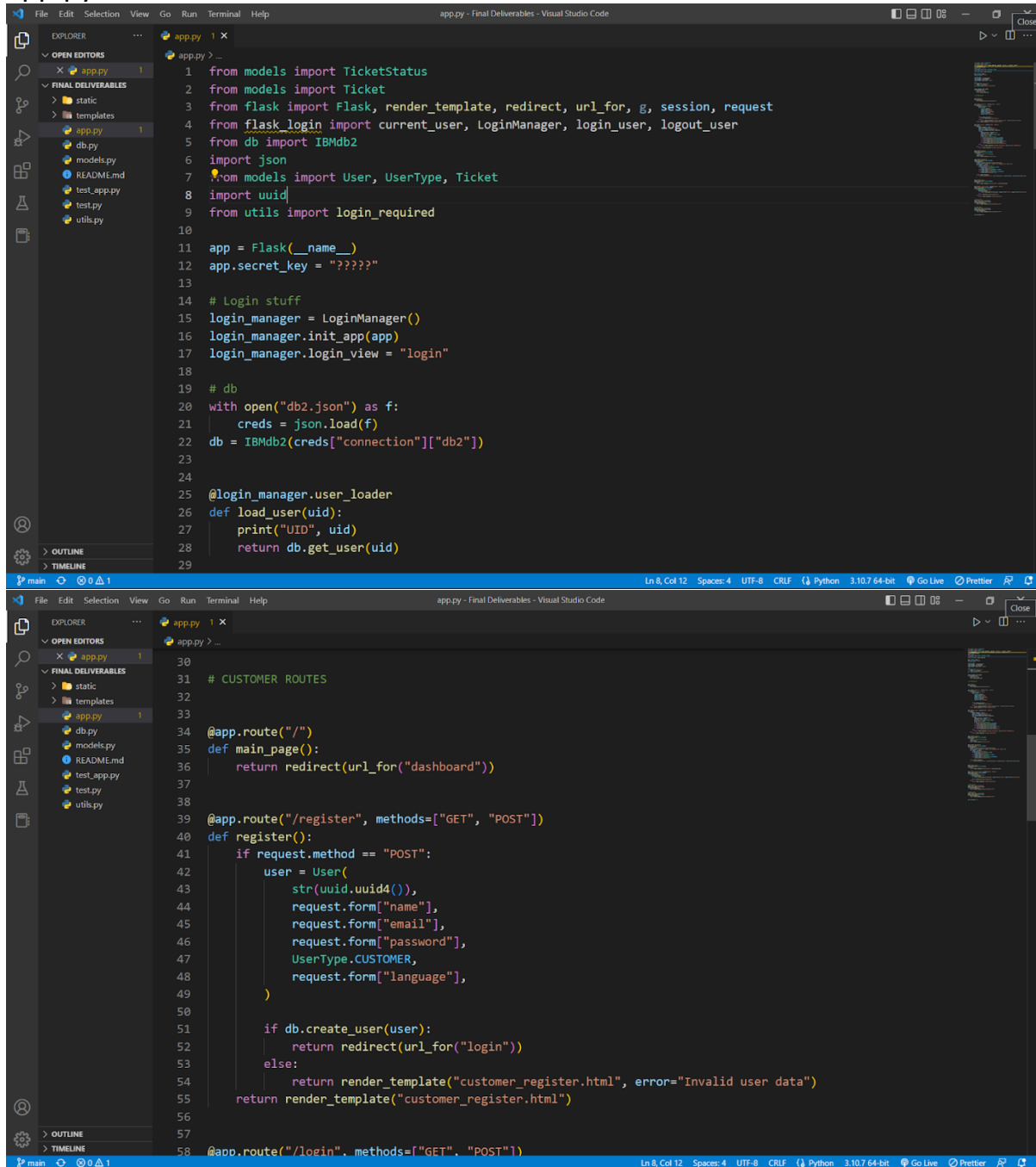
Countries around the world have started to invest more capital in CRM infrastructure, Customer relation enhancement, Better customer support. The market is growing and will continue to expand, given the benefits of business improvements due to good customer support. Customer relations is now considered as one of the important business acumens. It is the most looked after field by many business professionals who are planning to improve the market share. As the technology is evolving there are a lot of new features added on continuous intervals is one of the promising fields of the future.

# CHAPTER13

## APPENDIX

### Source Code

app.py



```
1 from models import TicketStatus
2 from models import Ticket
3 from flask import Flask, render_template, redirect, url_for, g, session, request
4 from flask_login import current_user, LoginManager, login_user, logout_user
5 from db import IBMDb2
6 import json
7 from models import User, UserType, Ticket
8 import uuid
9 from utils import login_required
10
11 app = Flask(__name__)
12 app.secret_key = "?????"
13
14 # Login stuff
15 login_manager = LoginManager()
16 login_manager.init_app(app)
17 login_manager.login_view = "login"
18
19 # db
20 with open("db2.json") as f:
21     creds = json.load(f)
22 db = IBMDb2(creds["connection"]["db2"])
23
24
25 @login_manager.user_loader
26 def load_user(uid):
27     print("UID", uid)
28     return db.get_user(uid)
29
30
31 # CUSTOMER ROUTES
32
33
34 @app.route("/")
35 def main_page():
36     return redirect(url_for("dashboard"))
37
38
39 @app.route("/register", methods=["GET", "POST"])
40 def register():
41     if request.method == "POST":
42         user = User(
43             str(uuid.uuid4()),
44             request.form["name"],
45             request.form["email"],
46             request.form["password"],
47             UserType.CUSTOMER,
48             request.form["language"],
49         )
50
51         if db.create_user(user):
52             return redirect(url_for("login"))
53         else:
54             return render_template("customer_register.html", error="Invalid user data")
55     return render_template("customer_register.html")
56
57
58 @app.route("/login", methods=["GET", "POST"])
```

The image displays two screenshots of a Visual Studio Code editor window, showing Python code for a web application. The top screenshot shows the login and logout routes, and the bottom screenshot shows the dashboard, chat, and create\_ticket routes.

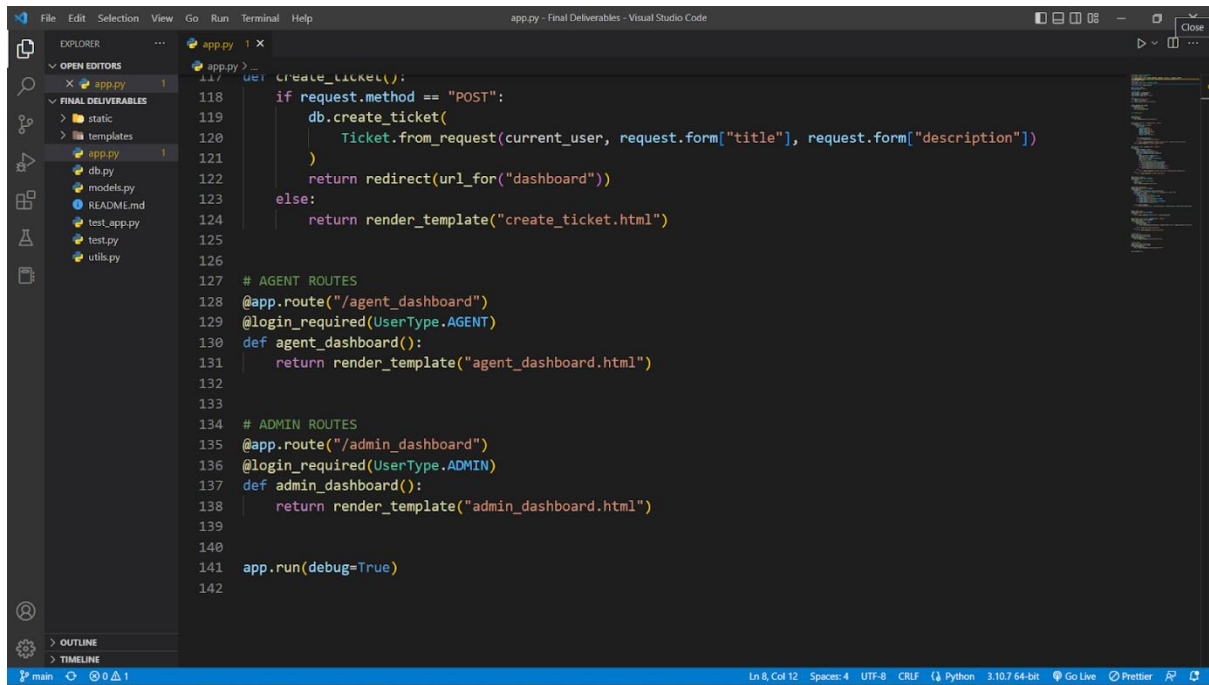
**Top Screenshot (app.py):**

```
57
58 @app.route("/login", methods=["GET", "POST"])
59 def login():
60     if request.method == "POST":
61         email = request.form["email"]
62         password = request.form["password"]
63         user = db.validate_user(email, password)
64         if user:
65             login_user(user, remember=True)
66             session["user"] = user
67             user.role = UserType(int(user.role))
68             print("login", user)
69             # TODO use the next parameter
70             if current_user.role == UserType.CUSTOMER:
71                 return redirect(url_for("dashboard"))
72             elif current_user.role == UserType.AGENT:
73                 return redirect(url_for("agent_dashboard"))
74             elif current_user.role == UserType.ADMIN:
75                 return redirect(url_for("admin_dashboard"))
76         else:
77             return render_template("customer_login.html", error="Invalid credentials")
78     else:
79         return render_template("customer_login.html")
80
81
82 @app.route("/logout")
83 @login_required(UserType.CUSTOMER)
84 def logout():
85     # TODO or redirect to main_page
```

**Bottom Screenshot (app.py):**

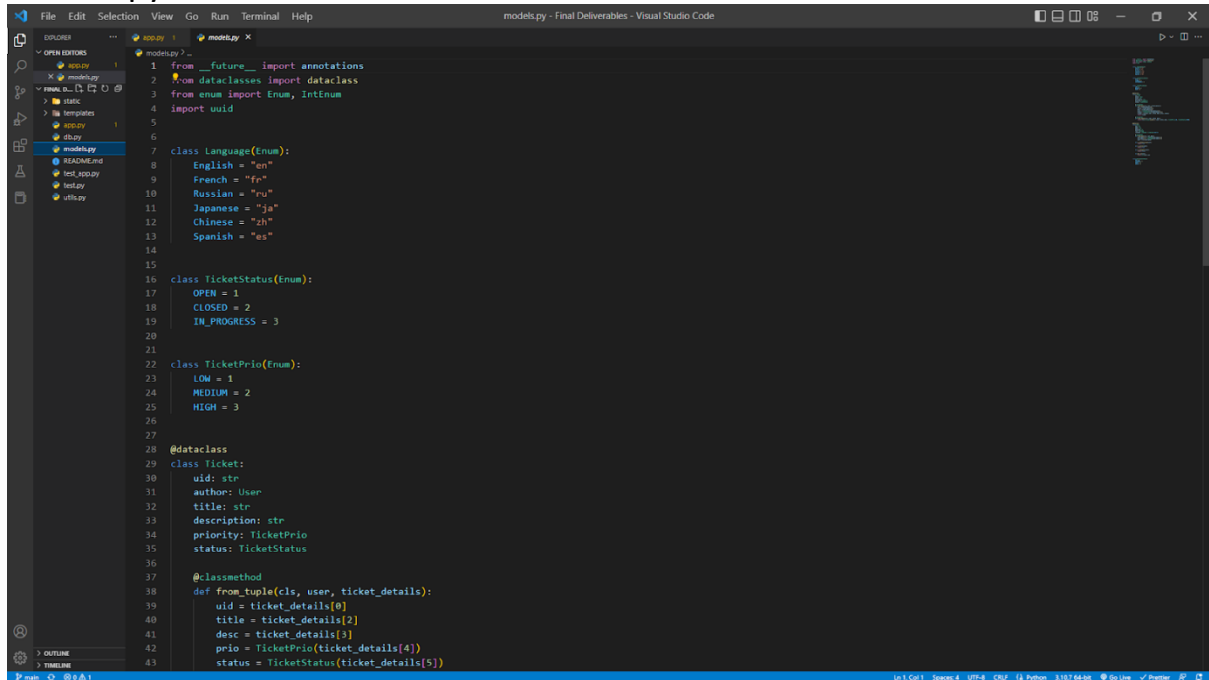
```
90 @app.route("/dashboard")
91 @login_required(UserType.CUSTOMER)
92 def dashboard():
93     tickets = db.get_tickets_for_user(current_user)
94     ticket_stats = {"open": 0, "closed": 0, "assigned": 0, "total": 0}
95     for ticket in tickets:
96         ticket_stats["total"] += 1
97         if ticket.status == TicketStatus.OPEN:
98             ticket_stats["open"] += 1
99         elif ticket.status == TicketStatus.CLOSED:
100             ticket_stats["closed"] += 1
101         elif ticket.status == TicketStatus.IN_PROGRESS:
102             ticket_stats["assigned"] += 1
103
104     return render_template(
105         "customer_dashboard.html", user=current_user, tickets=tickets, ticket_stats=ticket_stats
106     )
107
108
109 @app.route("/chat")
110 @login_required(UserType.CUSTOMER)
111 def chat():
112     return render_template("chatroom.html", user=current_user)
113
114
115 @app.route("/create_ticket", methods=["GET", "POST"])
116 @login_required(UserType.CUSTOMER)
117 def create_ticket():
118     if request.method == "POST":
```





```
117 def create_ticket():
118     if request.method == "POST":
119         db.create_ticket(
120             Ticket.from_request(current_user, request.form["title"], request.form["description"])
121         )
122         return redirect(url_for("dashboard"))
123     else:
124         return render_template("create_ticket.html")
125
126
127 # AGENT ROUTES
128 @app.route("/agent_dashboard")
129 @login_required(UserType.AGENT)
130 def agent_dashboard():
131     return render_template("agent_dashboard.html")
132
133
134 # ADMIN ROUTES
135 @app.route("/admin_dashboard")
136 @login_required(UserType.ADMIN)
137 def admin_dashboard():
138     return render_template("admin_dashboard.html")
139
140
141 app.run(debug=True)
142
```

## models.py



```
1 from __future__ import annotations
2 from dataclasses import dataclass
3 from enum import Enum, IntEnum
4 import uuid
5
6
7 class Language(Enum):
8     English = "en"
9     French = "fr"
10     Russian = "ru"
11     Japanese = "ja"
12     Chinese = "zh"
13     Spanish = "es"
14
15
16 class TicketStatus(Enum):
17     OPEN = 1
18     CLOSED = 2
19     IN_PROGRESS = 3
20
21
22 class TicketPrio(Enum):
23     LOW = 1
24     MEDIUM = 2
25     HIGH = 3
26
27
28 @dataclass
29 class Ticket:
30     uid: str
31     author: User
32     title: str
33     description: str
34     priority: TicketPrio
35     status: TicketStatus
36
37     @classmethod
38     def from_tuple(cls, user, ticket_details):
39         uid = ticket_details[0]
40         title = ticket_details[2]
41         desc = ticket_details[3]
42         prio = TicketPrio(ticket_details[4])
43         status = TicketStatus(ticket_details[5])
44
```

```
File Edit Selection View Go Run Terminal Help
models.py - Final Deliverables - Visual Studio Code

models.py
43
44 status = TicketStatus(ticket_details[5])
45 ticket = cls(uid, user, title, desc, prio, status)
46 return ticket
47
48 @classmethod
49 def from_request(cls, user, title, desc):
50     return cls(str(uuid.uuid4()), user, title, desc, TicketPrio.LOW, TicketStatus.OPEN)
51
52
53 @dataclass
54 class User:
55     uid: str
56     name: str
57     email: str
58     password: str
59     role: UserType
60     language: Language = Language.English
61
62 @classmethod
63 def from_tuple(cls, user_data):
64     user_data[-1] = Language[user_data[-1]]
65     user_data[-2] = UserType[user_data[-2]]
66     user = cls(*user_data)
67
68 def is_authenticated(self):
69     return True
70
71 def is_active(self):
72     return True
73
74 def is_anonymous(self):
75     return False
76
77 def get_id(self):
78     return str(self.uid)
79
80 class UserType(IntEnum):
81     ANY = 0
82     CUSTOMER = 1
83     AGENT = 2
84     ADMIN = 3
85
```

## utils.py

```
File Edit Selection View Go Run Terminal Help
utils.py - Final Deliverables - Visual Studio Code

utils.py
1 from flask import current_app
2 from models import User
3 import re
4 from models import UserType
5 from functools import wraps
6 from flask_login import current_user
7
8
9 def valid_user_data(user: User):
10     if not user.name or not user.email or not user.password:
11         return False
12     if not re.match(r"^[^@]+@[^@]+\.[^@]+$", user.email):
13         return False
14     return True
15
16
17 def login_required(role: UserType = UserType.ANY):
18     def wrapper(fn):
19         @wraps(fn)
20         def decorated_view(*args, **kwargs):
21             print(current_app.name, "AAA")
22             if not current_app:
23                 return current_app.login_manager.unauthorized()
24
25             print("Login request", role, current_user.role)
26             if role != UserType.ANY and current_user.role != role:
27                 return current_app.login_manager.unauthorized()
28
29             return fn(*args, **kwargs)
30
31         return decorated_view
32
33     return wrapper
34
```

## db.py

```
from typing import Optional

import ibm_db
import ibm_db_dbi
from werkzeug.security import check_password_hash, generate_password_hash

from models import Ticket, User, UserType
from utils import valid_user_data


class IBMdb2:
    def __init__(self, db2):
        self.conn = ibm_db.connect(
            f"DATABASE={db2['database']};HOSTNAME={db2['hosts'][0]['hostname']};PORT={db2['hosts'][0]['port']};PROTOCOL=TCPIP;UID={db2['authentication']['username']};PW="
        )
        ibm_db.autocommit(self.conn, ibm_db.SQL_AUTOCOMMIT_ON)

    def create_user(self, user: User):
        if not valid_user_data(user):
            return False

        insert_stmt = "INSERT INTO users VALUES (?, ?, ?, ?, ?, ?)"
        stmt = ibm_db.prepare(self.conn, insert_stmt)
        params = (
            str(uuid.uuid4()),
            user.name,
            user.email,
            generate_password_hash(user.password),
            user.role.value,
            user.language if isinstance(user.language, str) else user.language.value,
        )
        ibm_db.execute(stmt, params)
        return True

    def get_user(self, uid):
        stmt = ibm_db.prepare(self.conn, "SELECT * FROM users WHERE uid = ?")
        ibm_db.bind_param(stmt, 1, uid)
        ibm_db.execute(stmt)
```

```
user_data = ibm_db.fetch_tuple(stmt)
print("get_user data", user_data)
if user_data:
    user = User(*user_data)
    user.role = UserType(int(user.role))
    return user
return None

def get_user_from_email(self, email):
    stmt = ibm_db.prepare(self.conn, "SELECT * FROM users WHERE email = ?")
    ibm_db.bind_param(stmt, 1, email)
    ibm_db.execute(stmt)

    user_data = ibm_db.fetch_tuple(stmt)
    if user_data:
        user = User(*user_data)
        user.role = UserType(int(user.role))
        return user
    return None

def validate_user(self, email, password) -> Optional[User]:
    # Save a db call
    if not valid_user_data(User("", "DUMMY_USER", email, password, UserType.ANY)):
        return None
    user = self.get_user_from_email(email)
    if user and check_password_hash(user.password, password):
        return user
    return None

def create_ticket(self, ticket: Ticket):
    insert_stmt = "INSERT INTO tickets (uid, author, title, description, priority, status) VALUES (?, ?, ?, ?, ?, ?)"
    stmt = ibm_db.prepare(self.conn, insert_stmt)
    params = (
        str(uuid.uuid4()),
        ticket.author.uid,
        ticket.title,
        ticket.description,
        ticket.priority.value,
        ticket.status.value,
    )
    ibm_db.execute(stmt, params)
    return True
```

```
def create_ticket(self, ticket: Ticket):
    insert_stmt = "INSERT INTO tickets (uid, author, title, description, priority, status) VALUES (?, ?, ?, ?, ?, ?)"
    stmt = ibm_db.prepare(self.conn, insert_stmt)
    params = (
        str(uuid.uuid4()),
        ticket.author.uid,
        ticket.title,
        ticket.description,
        ticket.priority.value,
        ticket.status.value,
    )
    ibm_db.execute(stmt, params)
    return True

def get_tickets_for_user(self, user: User):
    stmt = ibm_db.prepare(self.conn, "SELECT * FROM tickets WHERE author = ?")
    ibm_db.bind_param(stmt, 1, user.uid)
    ibm_db.execute(stmt)

    tickets = []
    row = ibm_db.fetch_tuple(stmt)
    while row:
        tickets.append(Ticket.from_tuple(user, row))
        row = ibm_db.fetch_tuple(stmt)
    return tickets
```

## test\_app.py

```
test_app.py > -
1 from flask import Flask, render_template, redirect, url_for
2 from types import SimpleNamespace
3
4 app = Flask(__name__)
5
6 user = SimpleNamespace(name="sus")
7
8
9 @app.route("/")
10 def main_page():
11     return redirect(url_for("dashboard"))
12
13
14 @app.route("/register")
15 def register():
16     return render_template("customer_register.html")
17
18
19 @app.route("/login")
20 def login():
21     return render_template("customer_login.html")
22
23
24 @app.route("/logout")
25 def logout():
26     return render_template("logout.html")
27
28
29 @app.route("/dashboard")
30 def dashboard():
31     return render_template("customer_dashboard.html")
32
33
34 @app.route("/agent_dashboard")
35 def agent_dashboard():
36     return render_template("agent_dashboard.html")
37
38
39 @app.route("/chat")
40 def chat():
41     return render_template("chatroom.html", user=user)
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
```

```
43 <div class="text-bold tracking-wide">Tickets Assigned</div>
44 <p class="text-semibold text-gray-500 tracking-wider">
45   {{ticket_stats.assigned}}
46 </p>
47 </div>
48 <div
49   class="flex items-center justify-center flex-col border shadow-md w-48 h-24 bg-gray-100 rounded-2xl"
50 >
51   <div class="text-bold tracking-wide">Tickets Closed</div>
52   <p class="text-semibold text-gray-500 tracking-wider">
53     {{ticket_stats.closed}}
54   </p>
55 </div>
56 <div
57   class="flex items-center justify-center flex-col border shadow-md w-48 h-24 bg-gray-100 rounded-2xl"
58 >
59   <div class="text-bold tracking-wide">Total Tickets</div>
60   <p class="text-semibold text-gray-500 tracking-wider">
61     {{ticket_stats.total}}
62   </p>
63 </div>
64 </div>
65 </div>
66
67 <div class="flex justify-between grow items-center">
68   <h2 class="tracking-wider font-semibold text-2xl">Ticket Table</h2>
69   <!-- Ticket button -->
70   <button
71     class="bg-teal-600 hover:bg-teal-700 text-gray-100 rounded-xl p-3 font-semibold"
72     onclick="window.location.href = '/create_ticket'"
73   >
74     Create New Ticket
75   </button>
76 </div>
77
78 <!-- Ticket Table -->
79 <div class="overflow-x-auto relative">
80   <table class="w-full text-sm text-left text-gray-500">
81     <thead class="text-xs text-gray-700 uppercase bg-gray-200">
82       <tr>
83         <th scope="col" class="py-3 px-6">Ticket ID</th>
84         <th scope="col" class="py-3 px-6">Title</th>
85         <th scope="col" class="py-3 px-6">Status</th>
86         <th scope="col" class="py-3 px-6">Priority</th>
87       </tr>
88     </thead>
89     <tbody>
90       {% for ticket in tickets %}
91       <tr class="bg-white border-b hover:bg-slate-50 cursor-pointer">
92         <th
93           scope="row"
94           class="py-4 px-6 font-medium text-gray-900 whitespace-nowrap"
95         >
96           {{ticket.uid}}
97         </th>
98         <td class="py-4 px-6">{{ticket.title}}</td>
99         <td class="py-4 px-6">{{ticket.status.name}}</td>
100        <td class="py-4 px-6">{{ticket.priority.name |lower}}</td>
101      </tr>
102      {% endfor %}
103    </tbody>
104  </table>
105 </div>
106 </div>
107 </body>
108 </html>
109
```

GitHub Project Link: [IBM-Project-15570-1659600692](#)

Project Demo Link: