AI BASED DISCOURSE FOR BANKING INDUSTRY

Team ID: PNT2022TMID40307

Team Members:

Arunkumar R - 513319104006

Nithish Kumar B - 513319104022

Panneerselvam K - 513319104025

Rajakumaran M - 513319104029

Surya A – 513319104041

Content

1. INTRODUCTION

1.1 Project Overview

This project, titled “AI Based Discourse for Banking Industry”, aims at providing a platform for customers to clarify their queries online regarding banking. This is achieved with the help of a chatbot that is trained with some of the most frequently asked questions that banking customers usually tend to come up with. The chatbot is created using IBM Watson Assistant and is trained by manually adding the queries to it along with the relevant responses. The chatbot will then be available for use on a website which usually is the bank’s website for anyone to access. Customers or anyone in fact can access the chatbot to interact with it and find solutions to their queries.

1.2 Purpose

With banking being an essential service that people require and with it being a slightly complicated and confusing topic for many, a lot of queries naturally tend to arise. For them to be answered as they are predominantly now in a manual aspect either by face to face interactions with a banking employee or through a customer care service, will require a lot of workforce and still end up with long waiting times. Hence, comes the need for an automated solution to the problem which can be easily handled by our chatbot. A chatbot is free, easy to use and is readily accessible at all times and from anywhere. It also provides instant reliable answers to queries and hence eliminating the need for the customers to wait to get their queries cleared. It also ensures that there is no spread of misinformation by providing official and authentic responses to queries straight from the bank sources.

2. LITERATURE SURVEY

2.1 Existing problem

Banks are not able to resolve the queries of customers at all times related to the products or services in a satisfactory way which in turn hinders the customer satisfaction. Customers need to visit banks frequently for simple queries.

2.2 References

Paper 1 Authors: Vinod Kumar Shukla, Sasha Fathima Suhel, Sonali Vyas, Ved Prakash Mishra

Year: 2020

Title: Conversation to Automation in Banking Through Chatbot Using Artificial Machine Intelligence Language

Methodology: This paper examines some of the latest AI patterns and activities. System-Chatbots are made. In the banking industry, the introduction of Artificial Intelligence has driven chatbots and changed the face of the interaction between banks and customers.

Advantage: Artificial Intelligence involves creating machines that are capable of simulating knowledge. It also explores the existing usability of chatbot to assess whether it can fulfill the customers ever-changing needs.

Disadvantage: The dialogue capability can be limited to a very specific set or format of questions that are established by the chatbot development team.

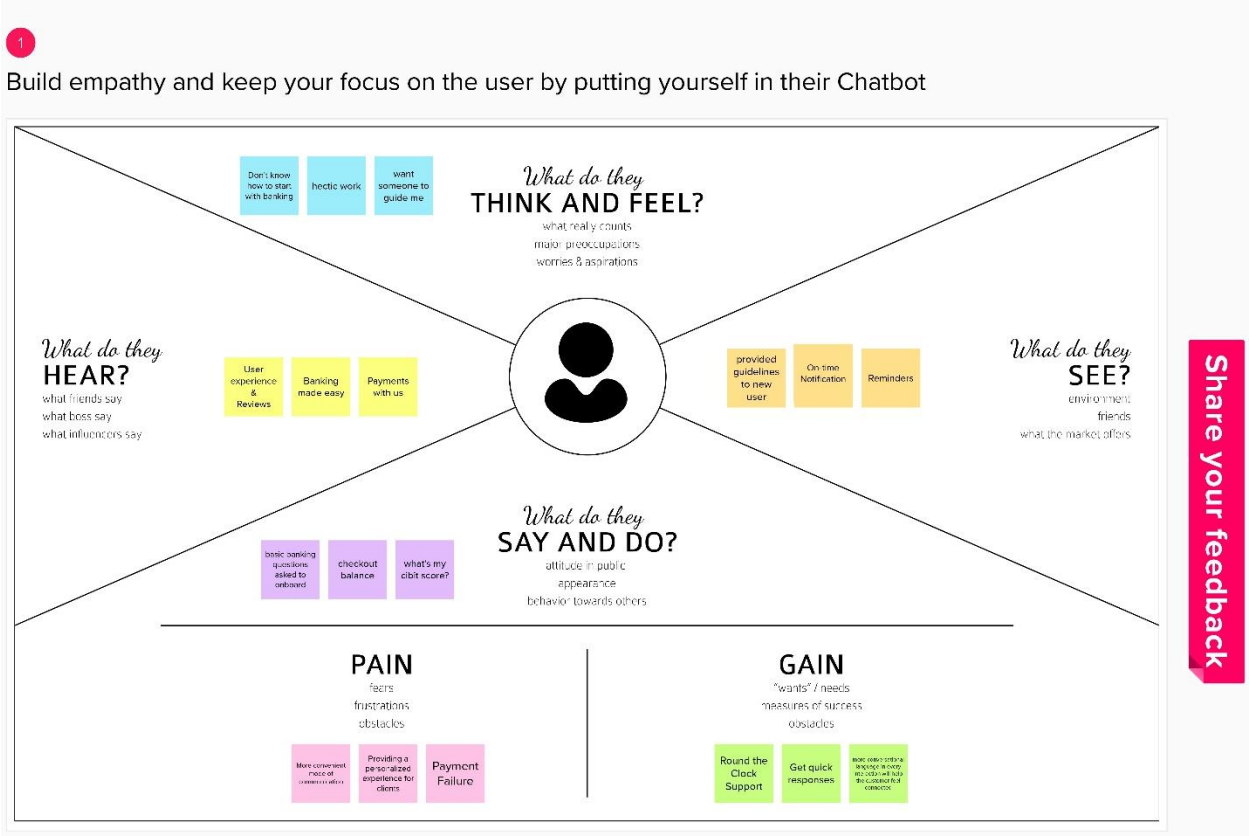
2.3 Problem Statement Definition

Banking is one the crucial sectors, it deals with financial transactions which can be availed by everyone, but banks are not able to resolve the queries of customers at all times related to the products or services in a satisfactory way in turn hinders the customer satisfaction. In order to guide the customers throughout all the financial services provided by the bank, an intelligent system has to be introduced to provide people with the best solution possible. The users are bank customers who need 24/7 service to clear all their queries and guide them through all the banking processes. So, an enhanced and smarter way of interaction with the customers has to be built to ensure efficient delivery of service. In order to overcome the user satisfaction issues associated with banking services, chatbot will provide personal and efficient communication between the user and the bank. It is

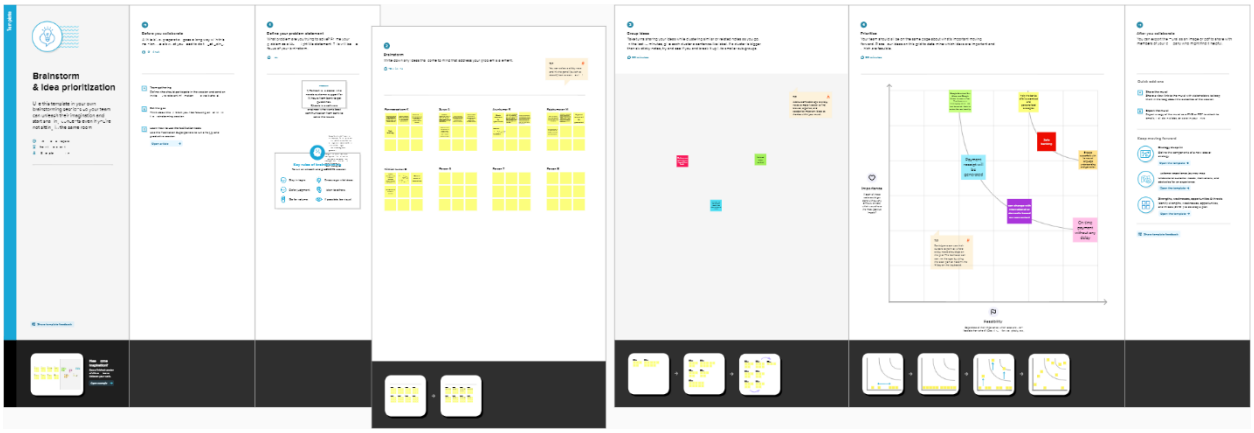
built to be the overall virtual assistant that can facilitate customers to ask banking-related questions without visiting the bank or calling up customer service centers as well as providing them with relevant suggestions.

3. IDEATION AND PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming



3.3 Proposed Solution

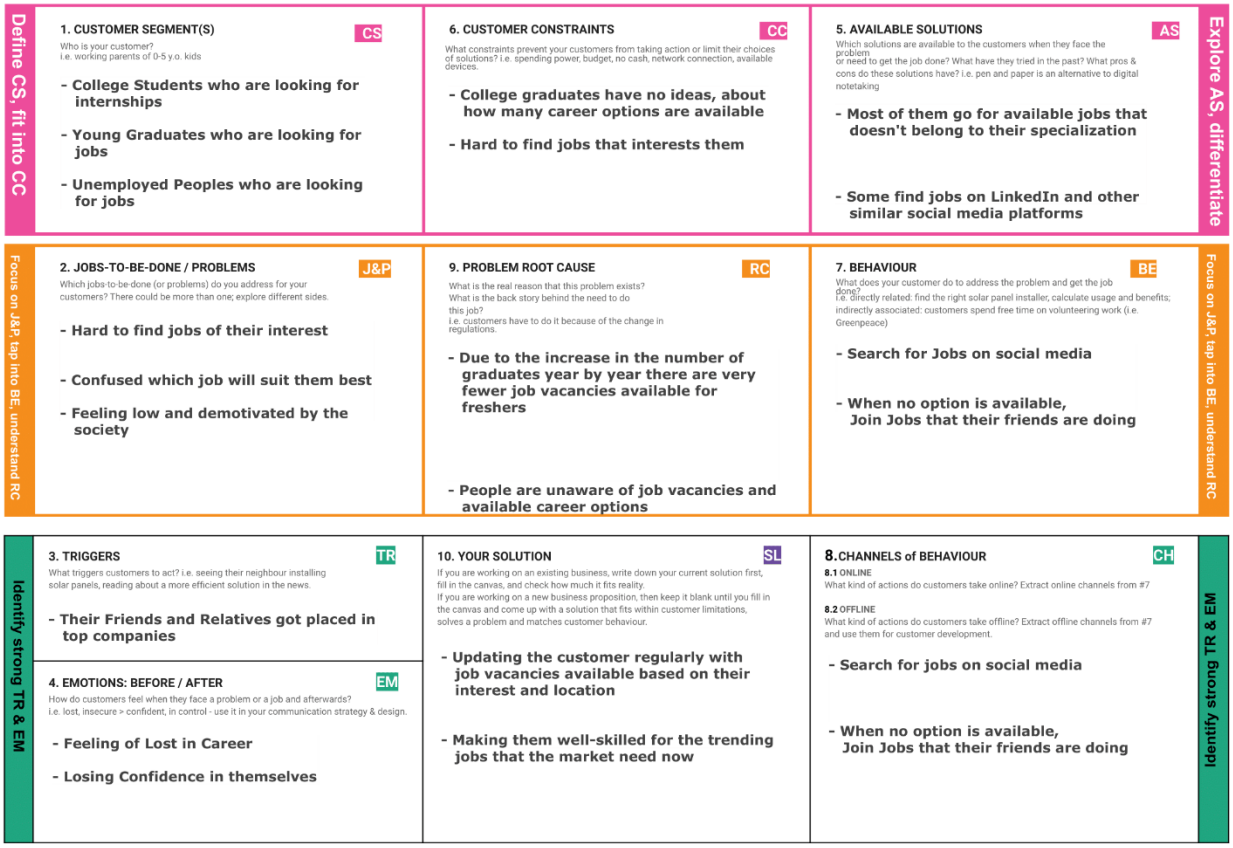
S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Technology spend in the banking space continues to be high, driven by the ever increasing need for more efficient banking services. In today's competitive banking landscape, customer expectations on response/turnaround times are increasing at a pace that large banks are struggling to catch up with, compared to their new generation peers. Also, the online lending institutions following an algorithmic approach enabling quick credit disbursements have resulted in upping the

		expectations of retail customers towards turnaround times. Customer-bank contact channels are also heavily tilting towards the digital platform. Despite this digital shift, customers still have to rely heavily on time-consuming phone and IVR systems for even basic banking enquiry and operations.
2	Idea / Solution description	One such initiative that is gaining popularity in banking circles is the use of chatbots which are essentially computer programs that can comprehend and act on people conversations. This whitepaper seeks to explore the potential for these artificial intelligence-based chatbots in the banking industry including pitfalls and supplier landscape.
3	Novelty / Uniqueness	These are bots powered by artificial intelligence coupled with deep banking-specific domain

		knowledge that facilitate customers to ask banking-related questions without visiting the bank or calling up customer service centres. Banking activities can be done throughconversation-like interactions thereby reducing turnaround time
4	Social Impact / Customer Satisfaction	The AI powered chat bot has become an integral part of the organization. It enables, 24×7 customer service, has lowered processing time and facilitated faster query resolution, leading to increased customer satisfaction. This has also helped the bank to better utilize its customer service agents and reduce operational costs.
5	Business Model (Revenue Model)	Banks will enable much and reliable services which will gain customer loyalty.
6	Scalability of the Solution	Chatbots understands the customer concerns and assists them just right, round the clock. In case, there comes a moment when you need human assistance, the bot

		carries forward your entire conversation to the human executive

3.4 Problem Solution fit



4. REQUIREMENT ANALYSIS

4.1 Functional requirement

Following are the functional requirements of the proposed solution

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Savings Account Related Actions	➤ Type of Savings Account Creation Details ➤ Interest Rate ➤ Minimum Balance ➤ Debit Card ➤ Credit Card
FR-2	Current Account Related Actions	➤ Type of Company ➤ Current Account Closure Steps ➤ Update GSTIN ➤ Zero Balance Current Account
FR-3	Loan Account Related Actions	➤ Type of Loan ➤ How long for approval ➤ Available Loan Amounts
FR-4	General Queries Related Actions	➤ Bank Working Days ➤ List of Branches ➤ Storage Locker Facility ➤ Currency Conversion Facility
FR-5	Net Banking Related Actions	➤ Login Steps ➤ Change Net Banking Password ➤ Daily Limit ➤ Types of Fund Transfer ➤ Add Beneficiary

4.2 Non-Functional requirements

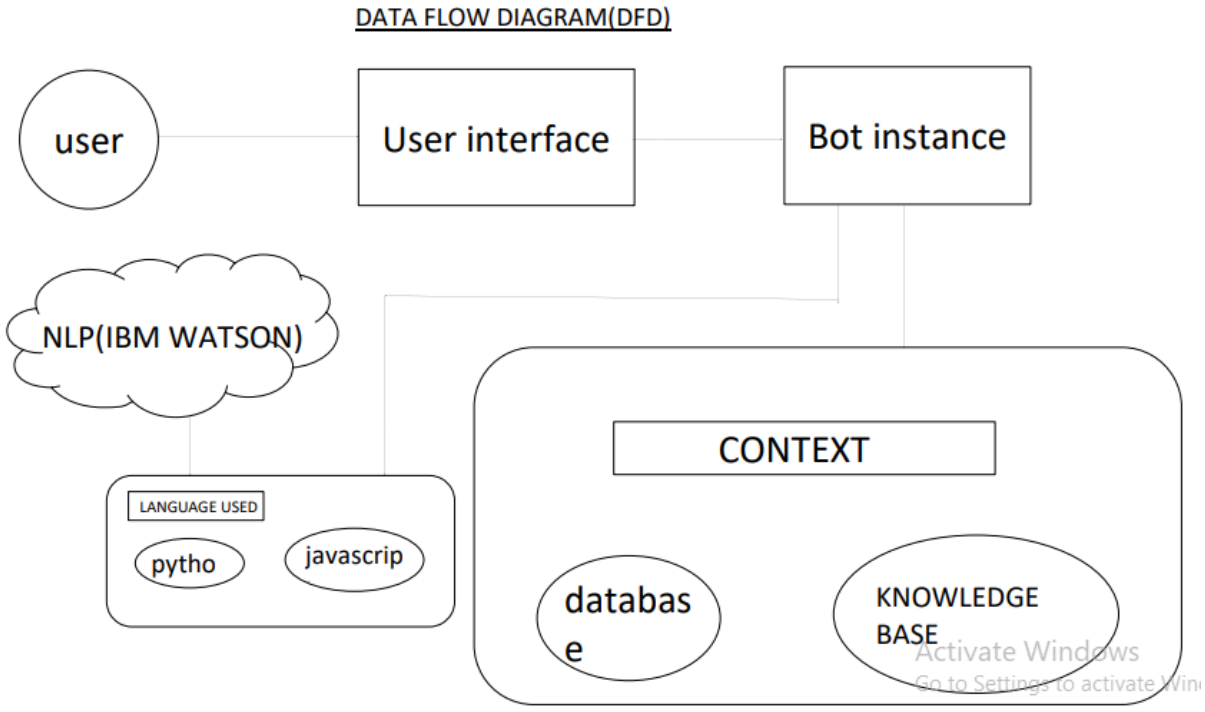
Following are the non-functional requirements of the proposed solution

NFR	Non-Functional requirements	Description
NFR-1	Usability	The chatbot must be efficient with very little lag in response time for instance no longer than 5 seconds to reply to a user message
NFR-2	Security	The chatbot must be secure as sensitive data is being used, Googles 2-Factor Authentication will be implemented as an extra security feature
NFR-3	Reliability	The chatbot must be reliable with next to no faults or bugs
NFR-4	Performance	The use of natural language used to interact with the chatbot promotes human computer interaction.
NFR-5	Availability	Appropriate handling of unexpected input & , and correctly inform the user if it cannot provide assistance
NFR-6	Scalability	The database must be scalable to adopt to a growing number of users

5. PROJECT DESIGN

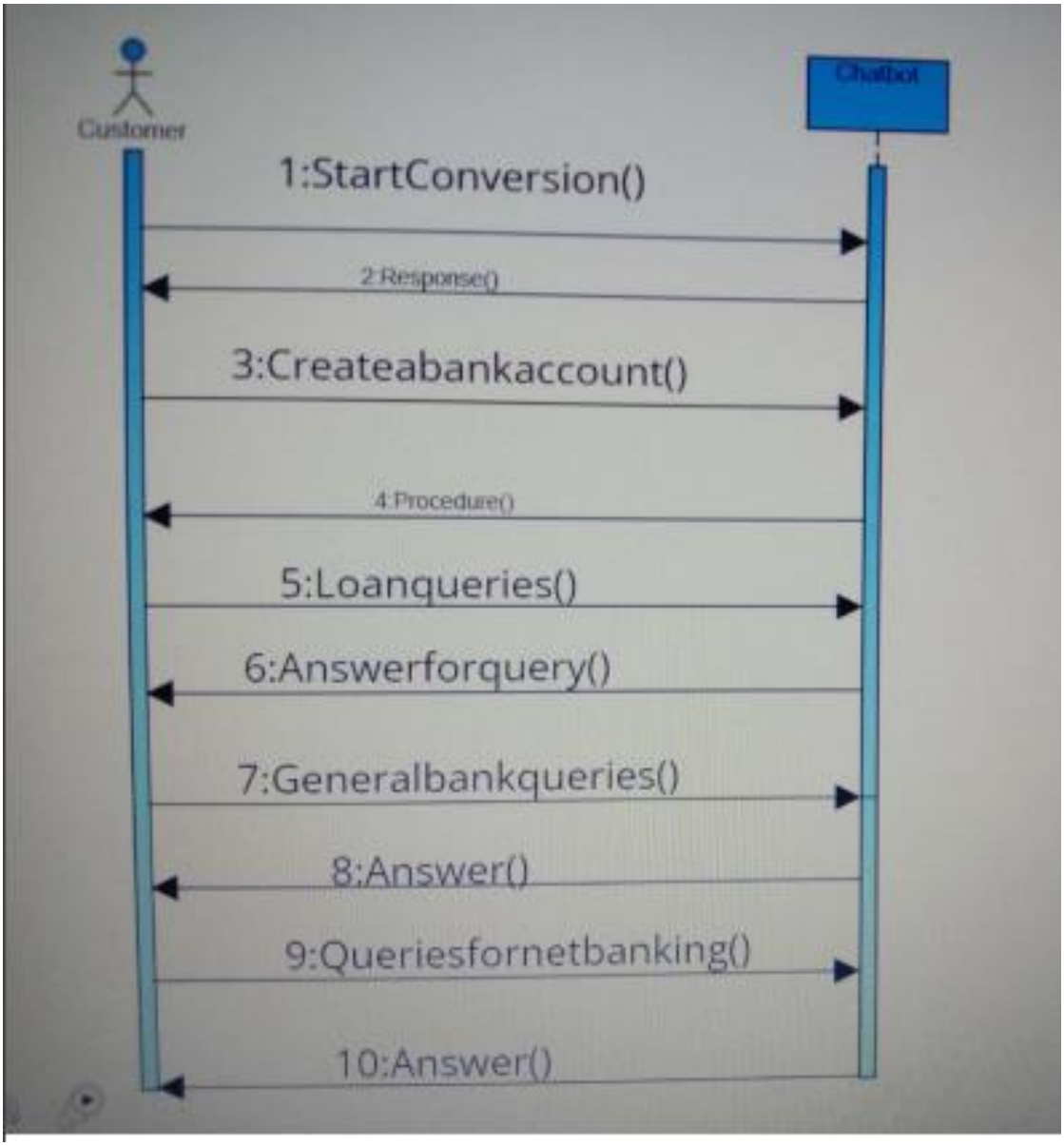
5.1 Data Flow Diagrams

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:



Techinical Architecture

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

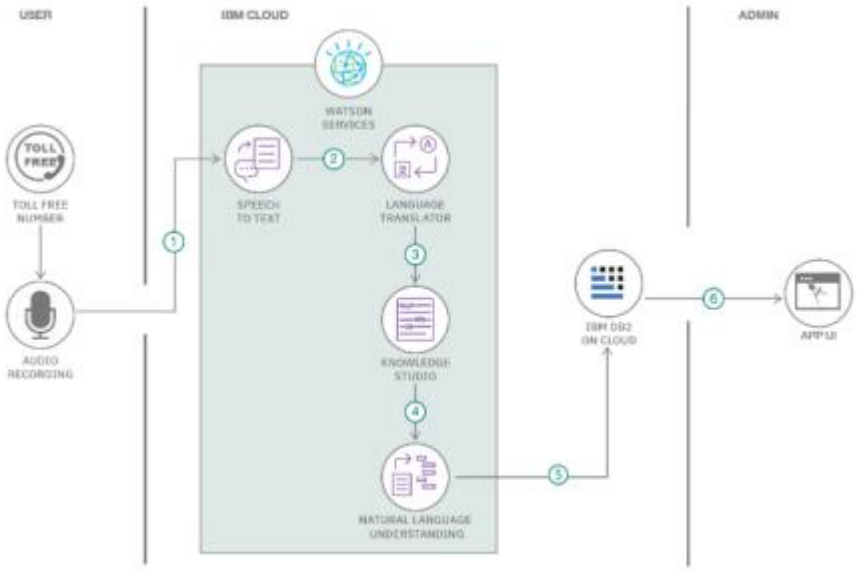


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript / Angular Js / React Js etc.
2.	Application Logic-1	Logic for a process in the application	Java / Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.

7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	Purpose of External API used in the application	IBM Weather API, etc.
9.	External API-2	Purpose of External API used in the application	Aadhar API, etc.
10.	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model, etc.
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration : 	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

S.No	**Characteristics**	**Description**	**Technology**
1.	Open-Source Frameworks	List the open-source frameworks used	Technology of Opensource framework
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	e.g. SHA-256, Encryptions, IAM Controls, OWASP etc.
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	Technology used
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	Technology used
5.	Performance	Design consideration for the performance of the application (number of	Technology used

		requests per sec, use of Cache, use of CDN's) etc.	

5.3 User Stories

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		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1

	Dashboard					
Customer (Web user)						
Customer Care Executive						
Administrator						

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

To accomplish the above task, you must complete the below activities and tasks:

1. Create IBM Services.
2. Creating skills & Assistant for Chatbot.
3. Creating Savings account action.
4. Creating Current account action.
5. Creating Loan account action.
6. Creating a general query action.
7. Creating a Net banking action.
8. Create HTML web page.
9. Integrate the Watson Chatbot with web page.

6.2 Sprint Delivery Schedule

Product Backlog, Sprint Schedule, and Estimation

Sprint	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Story Points**	**Priority**	**Team Members**
Sprint-1	Registration	USN-1	Create IBM Watson	2	High	Arunkumar R Nithish Kumar B Surya A

			Assistant Service			
Sprint-2		USN-2	Chatbot Skills Creation	1	High	Panneerselvam K Rajakumaran M
Sprint-2	Enter the query for saving account	USN-3	Creating Saving Account Action	2	Low	ARUNKUMAR R Nithish Kumar B Surya A Panneerselvam K Rajakumaran M Surya A
Sprint-2	Enter the query for current account	USN-4	Creating Current Account Action	2	Mediu m	Arunkumar R NithishKumar B Panneerselvam K Rajakumaran M Surya A
Sprint-3	Enter the query for loan account	USN-5	Creating Loan Account Action	2		ARUNKUMAR R Nithish Kumar B Surya A Panneerselvam K Rajakumaran M Surya A
Sprint - 3		USN-6	Creating General Query Action	2		ARUNKUMAR R Nithish Kumar B Surya A

						Panneerselvam K Rajakumaran M Surya A
Sprint-3		USN-7	Creating Net Banking Action	2		ARUNKUMAR R Nithish Kumar B Surya A Panneerselvam K Rajakumaran M Surya A
Sprint - 4		USN-8	Build Python Code			ARUNKUMAR R Nithish Kumar B Surya A Panneerselvam K Rajakumaran M Surya A
Sprint-4		USN-9	Build HTML Code	2		ARUNKUMAR R Nithish Kumar B Surya A Panneerselvam K Rajakumaran M Surya A
Sprint-4		USN-10	Run the Application	2		ARUNKUMAR R Nithish Kumar B Surya A Panneerselvam K

						Rajakumaran M Surya A

Sprint	**Total Story Points**	**Duration**	**Sprint Start Date**	**Sprint End Date (Planned)**	**Story Points Completed (as on Planned End Date)**	**Sprint Release Date (Actual)**
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

Velocity:

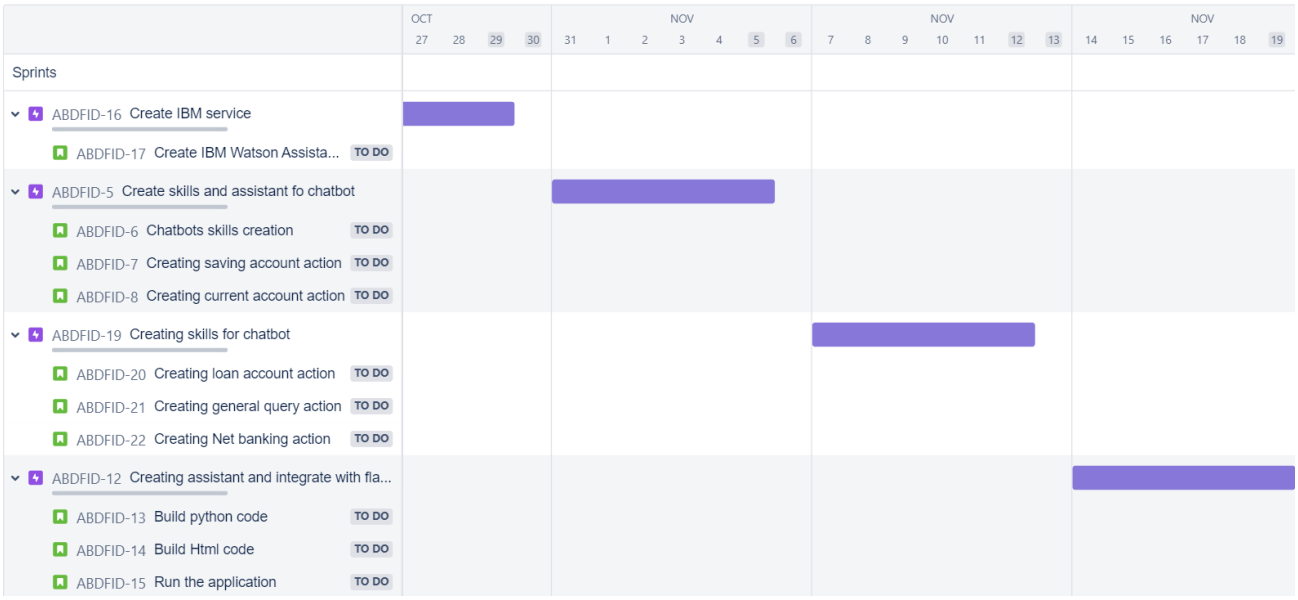
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

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

Burndown Chat:

A burn down 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

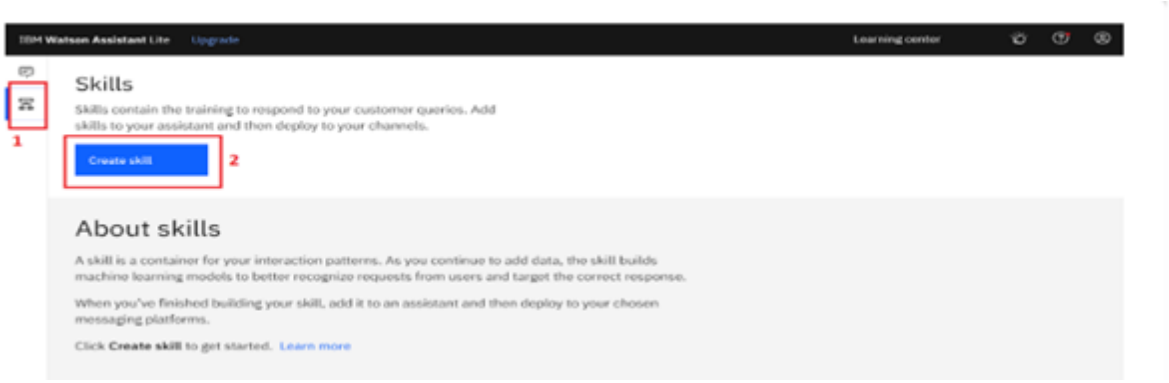


7. CODING & SOLUTIONING

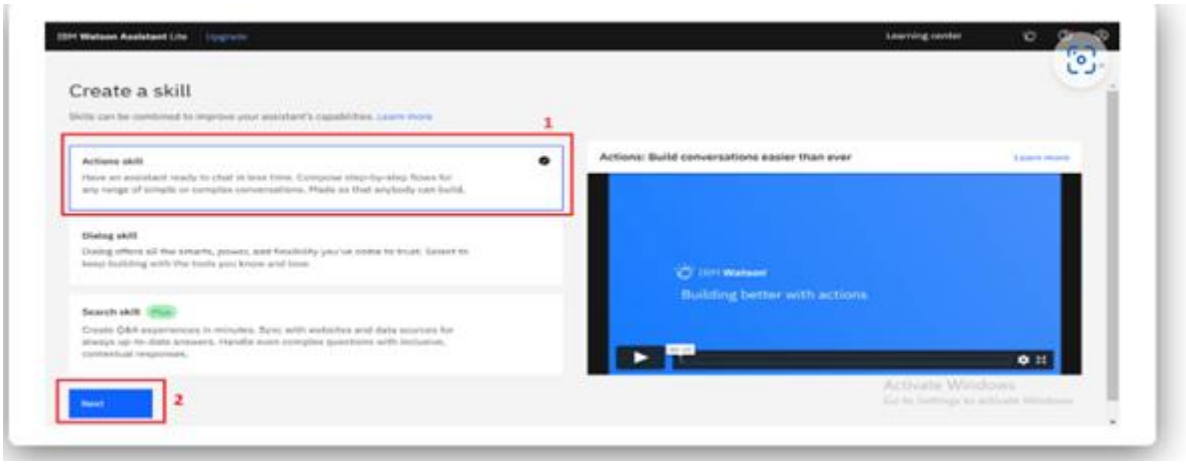
7.1 Feature 1

A.Chatbot Skill Creation

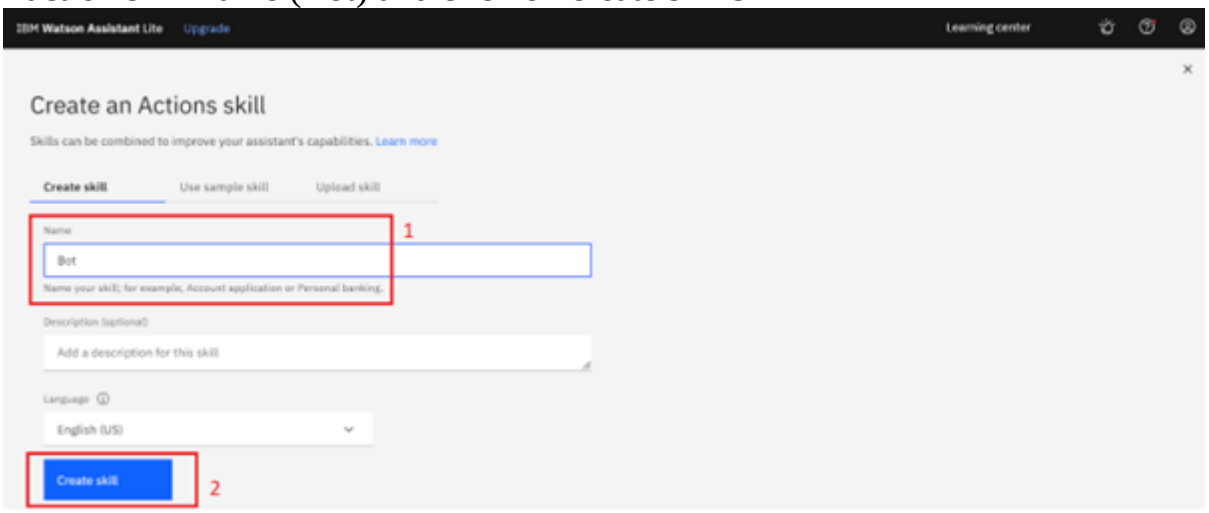
1.Click on the Skills tab and click on create skills



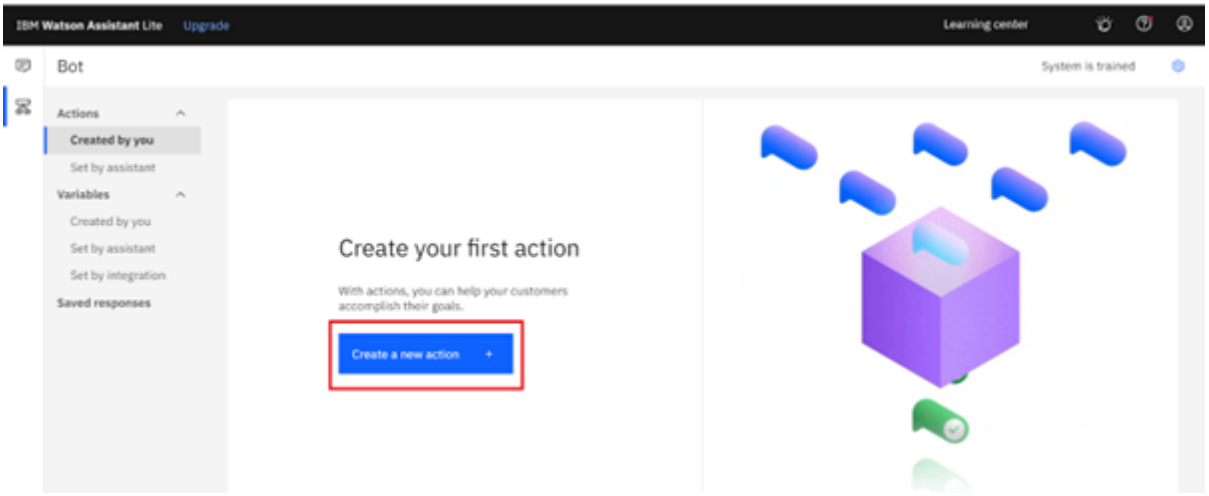
2.Select Action skills and click on next.

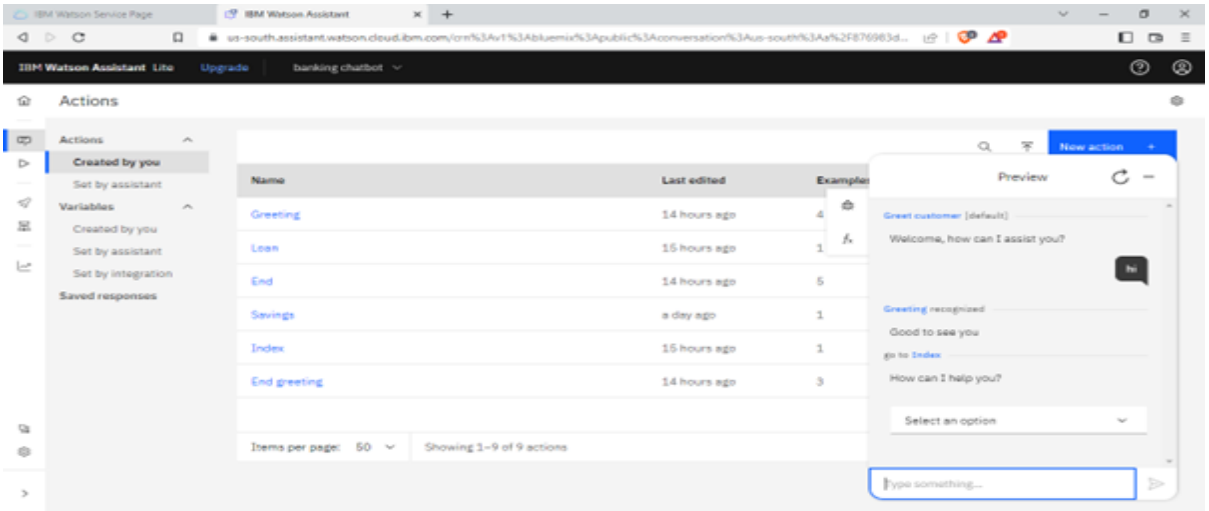


3.Enter action skill name (Bot) and click on create skills

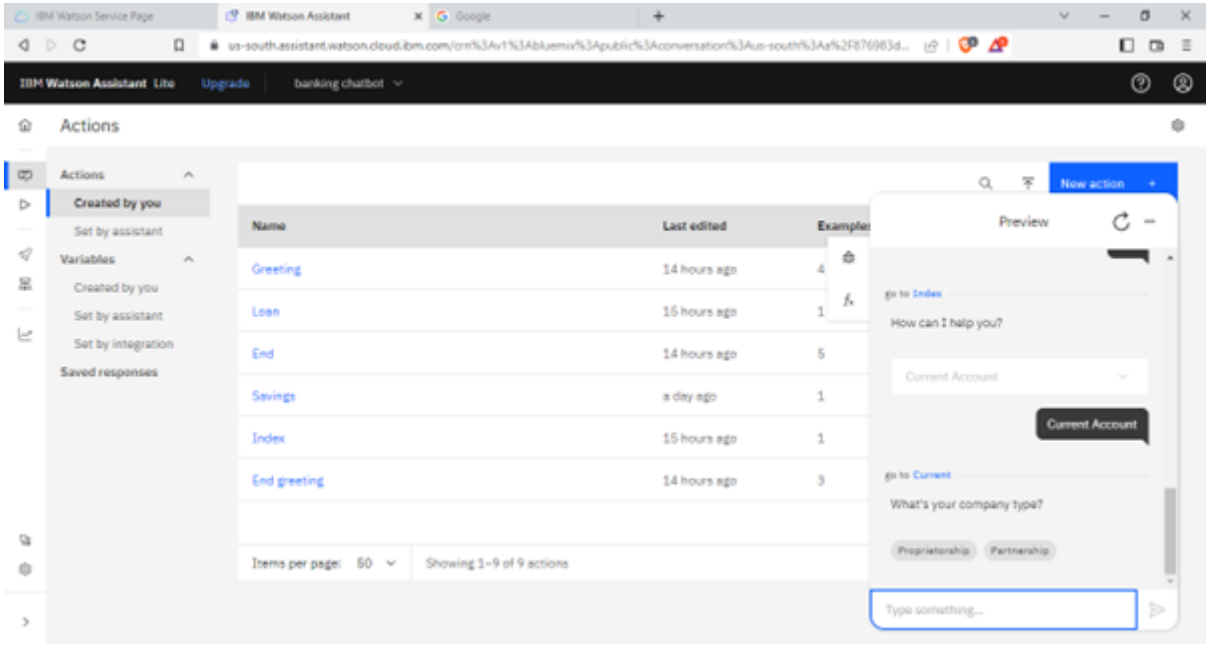


4.To create a new action click on Create a new action button

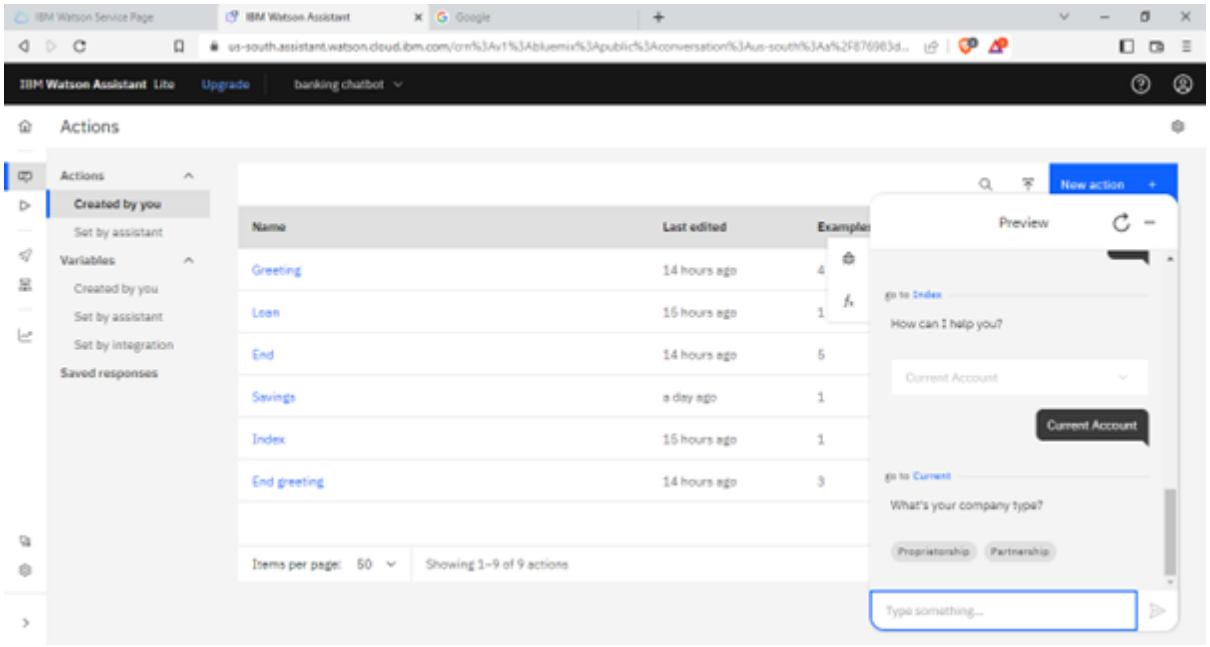




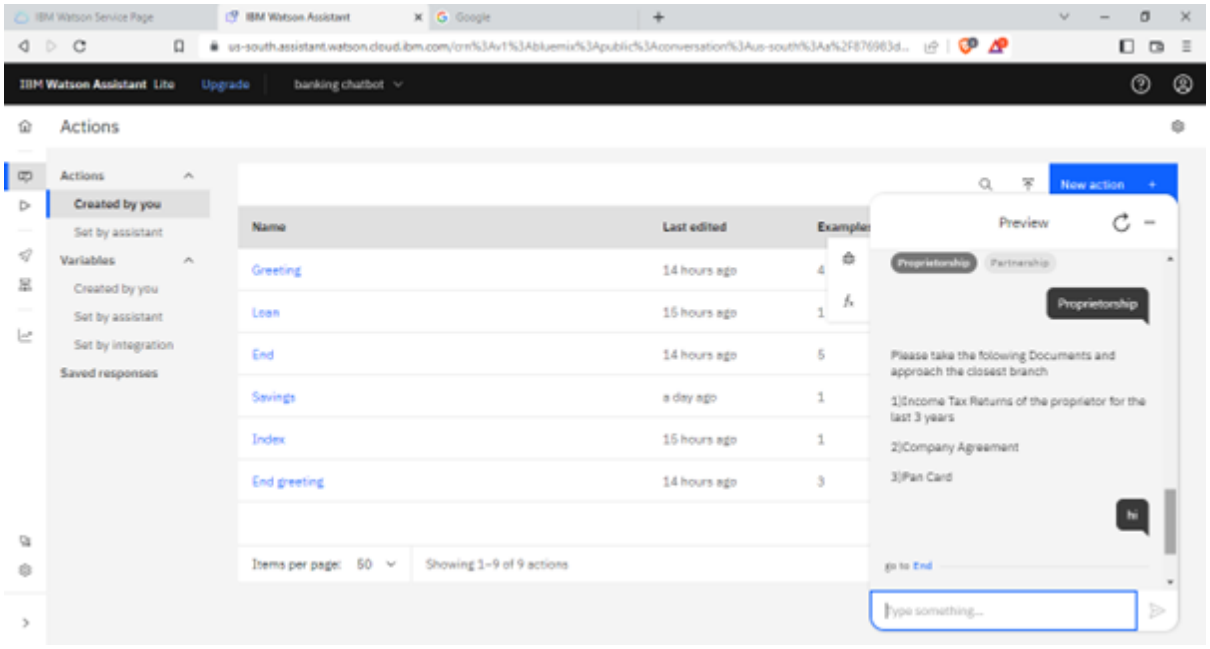
5.Actions are created

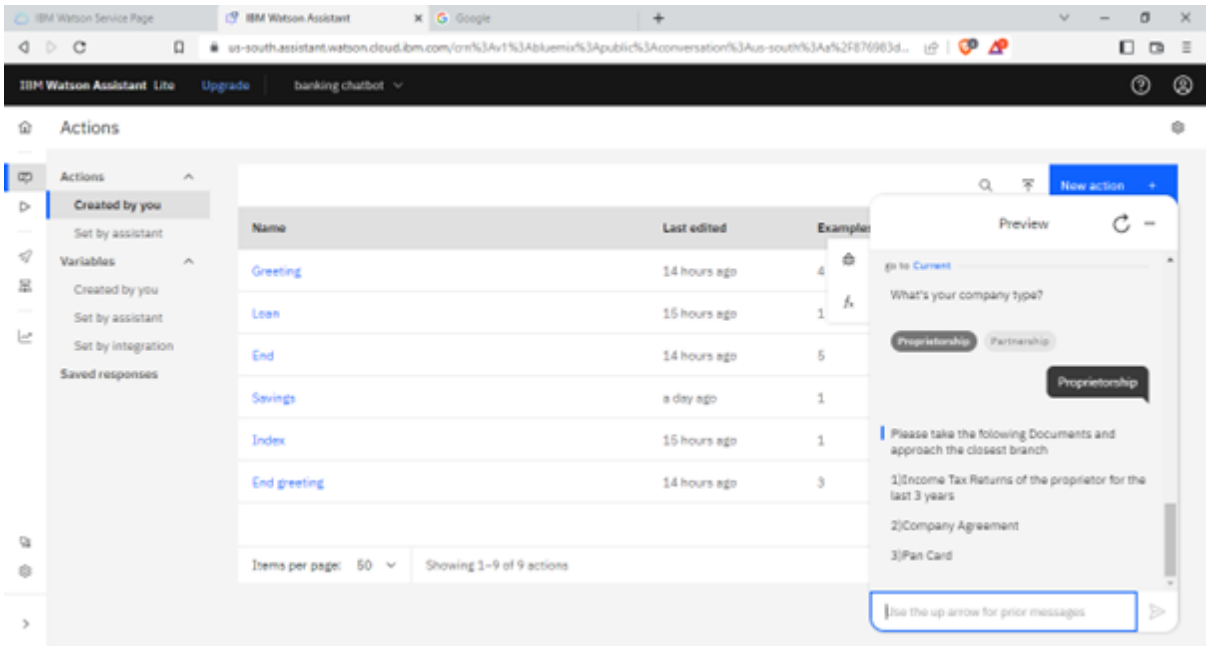


6.When you press current account

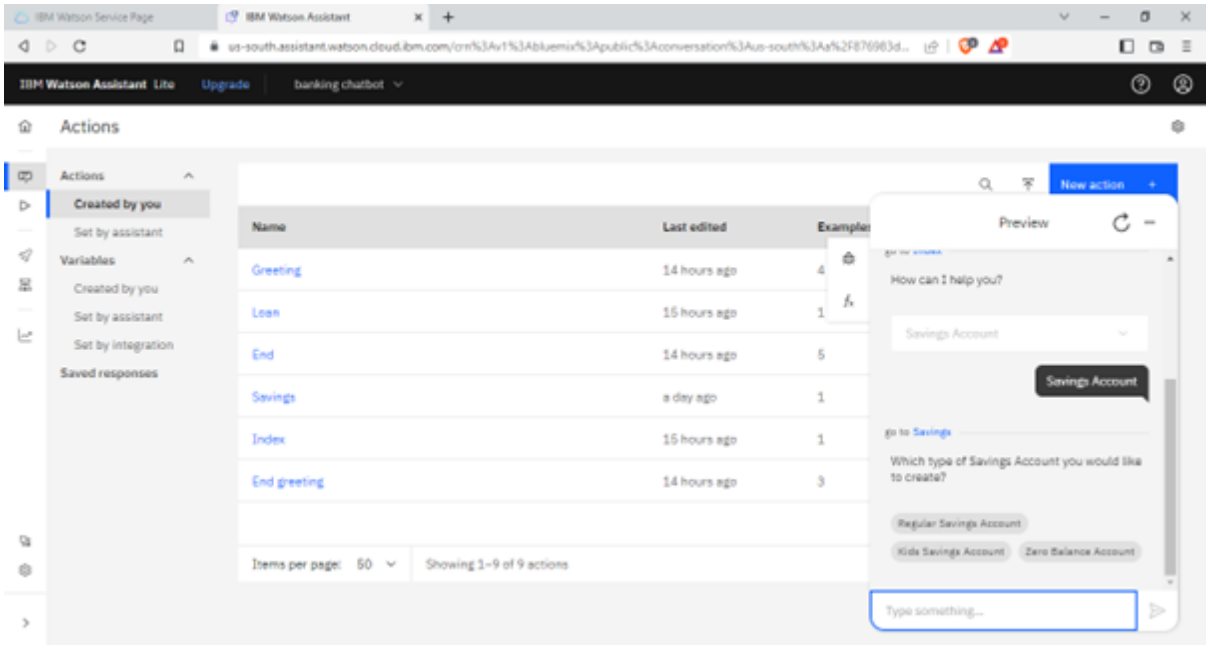


7.Need to select your’s company type

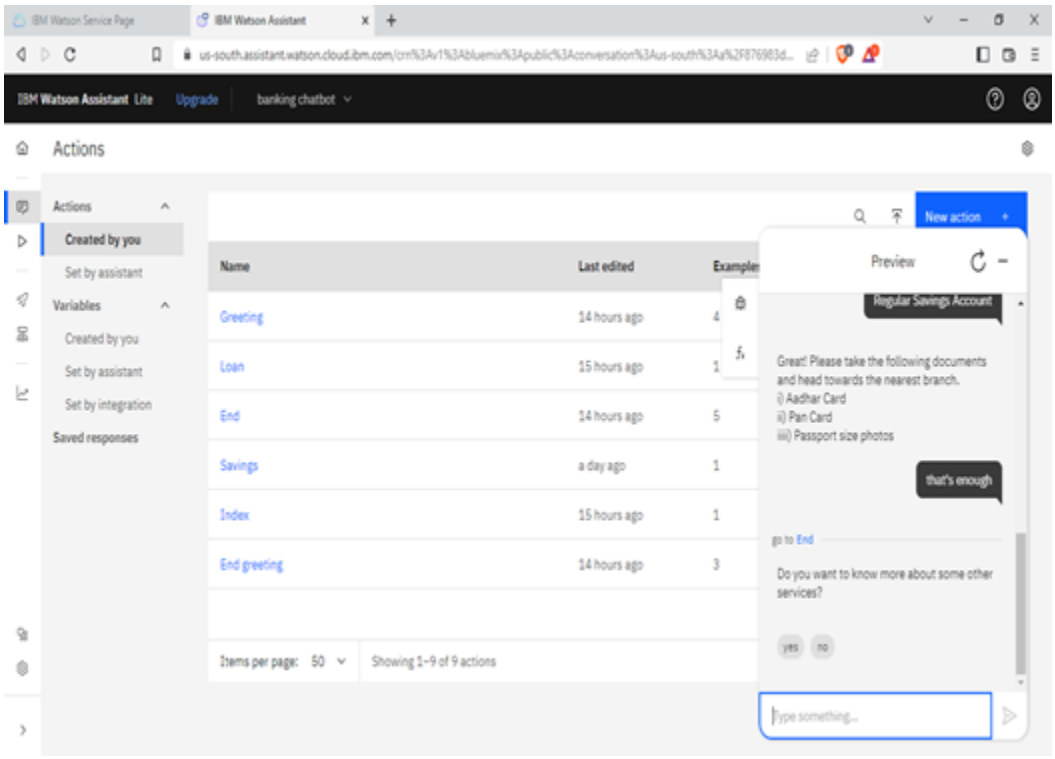




8.Need to select savings account



9.Choose regular savings account



7.2 Feature 2

Build Python Code

1: Importing Libraries

The first step is usually importing the libraries that will be needed in the program.

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

2: Creating our flask application and loading

```
app = Flask(__name__)
```

3: Routing to the Html Page

Here, the declared constructor is used to route to the HTML page created earlier.

The '/' is bound with the bot function. Hence, when the home page of a web server

opened in the browser, the HTML page will be rendered

```
@app.route('/')
def bot():
    return render_template('chatbot.html')
```

Main Function

This is used to run the application in localhost.

```
if __name__ == '__main__':
    app.run()
```

Build HTML Code

- We use HTML to create the front-end part of the web page.
- Here, we have created 1 HTML page-Chatbot.html
- Chatbot.html displays the home page which integrates with Watson Assistant.
- A simple HTML page is created. Auto-generated source code from IBM Watson Assistants is copied and pasted inside the body tag

Run The Application

- Open the anaconda prompt from the start menu.
- Navigate to the folder where your app.py resides.
- Now type the “python chatbot.py” command.
- It will show the local host where your app is running on http://127.0.0.1.500/
- Copy that localhost URL and open that URL in the browser. It does navigate me to where you can view your web page.

8. TESTING

8.1 Test Cases

- Verify user is able to see the chatbot icon when website is launched
- Verify the UI elements in chatbot icon popup
- Verify user is able to see the greeting from chatbot "Hi! I'm a Banking Bot. How can I help you today?
- Verify user is able to type query in text field.
- Verify user is able to get the response from chatbot
- Verify user whether get the response if the user enter the wrong query also

8.2 User Acceptance Test

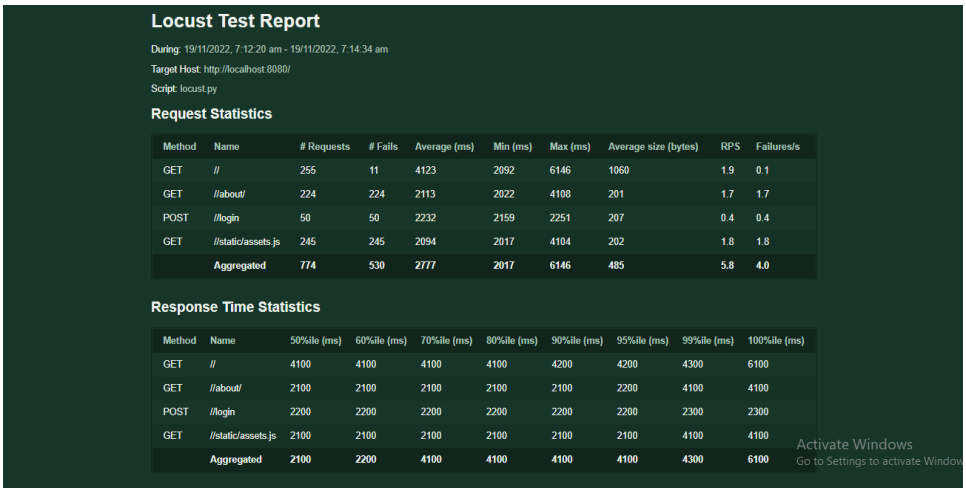
Defect Analysis

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	1	0	0	0	1
Duplicate	3	1	0	1	5
External	1	3	0	1	5
Not Reproduced	0	0	0	1	1
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Total	7	9	3	5	24

9. RESULT

9.1 Performance Metrics



Locust Test Report

During: 19/11/2022, 7:12:20 am - 19/11/2022, 7:14:34 am

Target Host: http://localhost:8080/

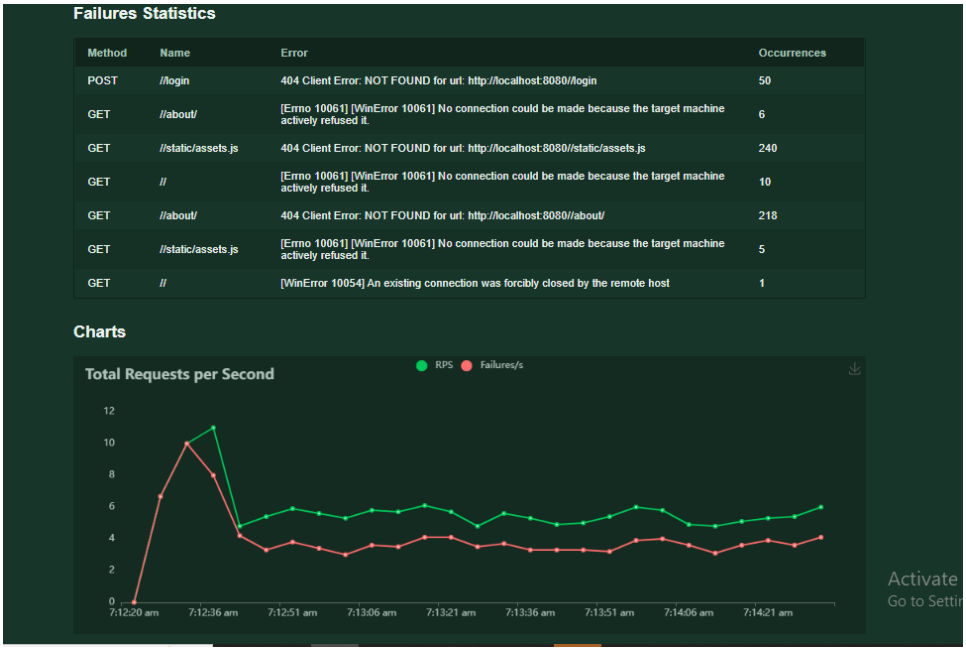
Script: locust.py

Request Statistics

Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
GET	//	255	11	4123	2092	6146	1060	1.9	0.1
GET	//about/	224	224	2113	2022	4108	201	1.7	1.7
POST	//login	50	50	2232	2159	2251	207	0.4	0.4
GET	//static/assets.js	245	245	2094	2017	4104	202	1.8	1.8
	Aggregated	**774**	**530**	**2777**	**2017**	**6146**	**485**	**5.8**	**4.0**

Response Time Statistics

Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
GET	//	4100	4100	4100	4100	4200	4200	4300	6100
GET	//about/	2100	2100	2100	2100	2100	2200	4100	4100
POST	//login	2200	2200	2200	2200	2200	2200	2300	2300
GET	//static/assets.js	2100	2100	2100	2100	2100	2100	4100	4100
	Aggregated	**2100**	**2200**	**4100**	**4100**	**4100**	**4100**	**4300**	**6100**



Failures Statistics

Method	Name	Error	Occurrences
POST	//login	404 Client Error: NOT FOUND for url: http://localhost:8080//login	50
GET	//about/	[Errno 10061] [WinError 10061] No connection could be made because the target machine actively refused it.	6
GET	//static/assets.js	404 Client Error: NOT FOUND for url: http://localhost:8080//static/assets.js	240
GET	//	[Errno 10061] [WinError 10061] No connection could be made because the target machine actively refused it.	10
GET	//about/	404 Client Error: NOT FOUND for url: http://localhost:8080//about/	218
GET	//static/assets.js	[Errno 10061] [WinError 10061] No connection could be made because the target machine actively refused it.	5
GET	//	[WinError 10054] An existing connection was forcibly closed by the remote host	1

Charts

Final ratio

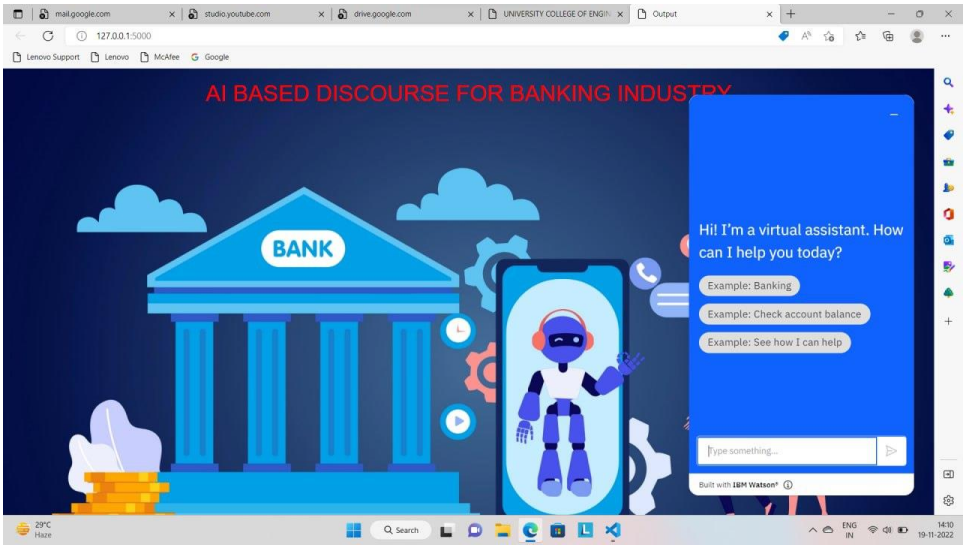
Ratio per User class

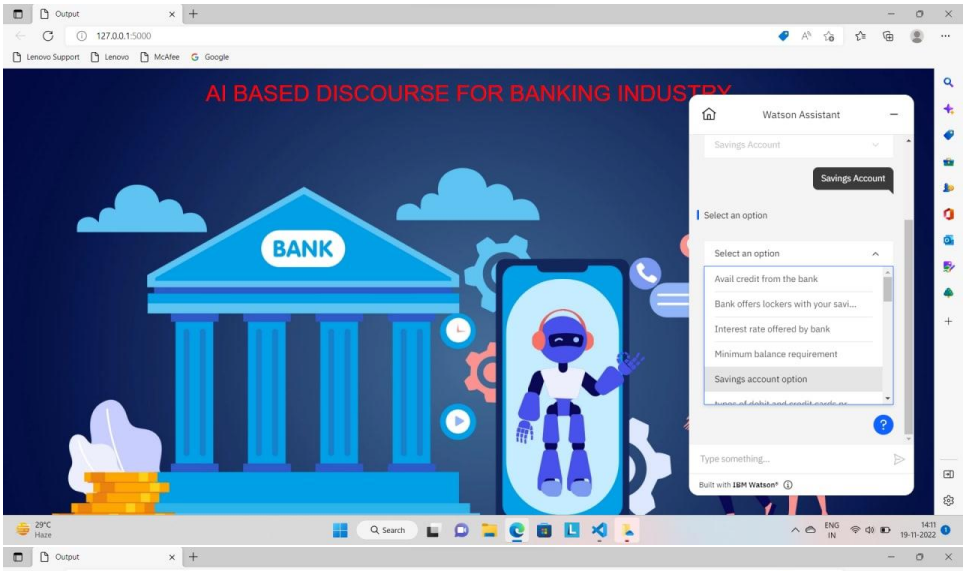
- 100.0% WebsiteUser
  - 50.0% index
  - 50.0% about

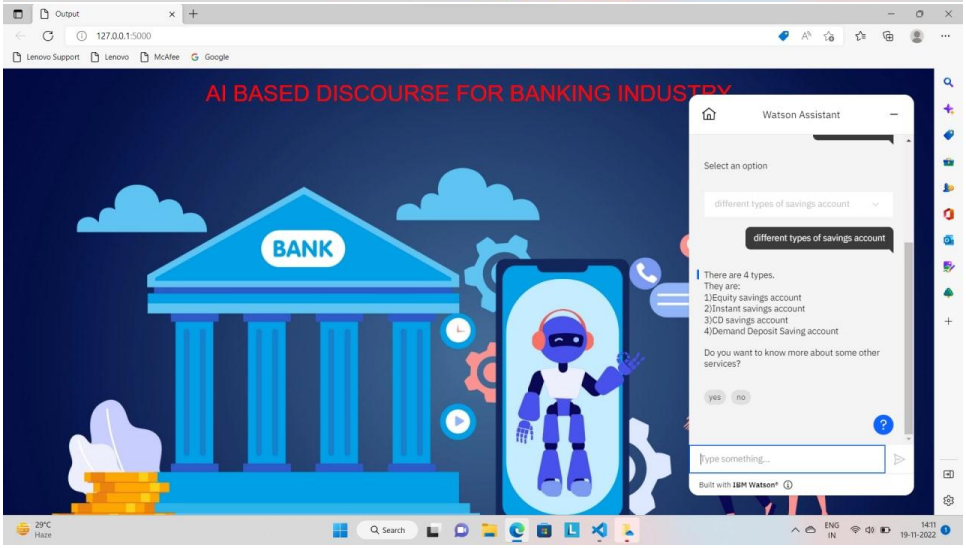
Total ratio

- 100.0% WebsiteUser
  - 50.0% index
  - 50.0% about

9.2 OUTPUT


AI BASED DISCOURSE FOR BANKING INDUSTRY
BANK
Hi! I'm a virtual assistant. How can I help you today?
Example: Banking
Example: Check account balance
Example: See how I can help
Type something...
Built with IBM Watson®


AI BASED DISCOURSE FOR BANKING INDUSTRY
BANK
Watson Assistant
Savings Account
Select an option
Avail credit from the bank
Bank offers lockers with your savi...
Interest rate offered by bank
Minimum balance requirement
Savings account option
Type something...
Built with IBM Watson®


AI BASED DISCOURSE FOR BANKING INDUSTRY
BANK
Watson Assistant
Select an option
different types of savings account
There are 4 types.
They are:
1)Equity savings account
2)Instant savings account
3)CD savings account
4)Demand Deposit Saving account
Do you want to know more about some other services?
yes
no
Type something...
Built with IBM Watson®

10. ADVANTAGES & DISADVANTAGES

Advantages of using AI Chatbots

• Availability

Chatbots are available 24×7 and can respond to your customers instantly. This means that whenever they message you for any reason, they'll be able to get a response immediately. As a result, they'll be satisfied with your brand and you, on the other hand, will be able to move them along your sales funnel.

• More Engagement

Chatbots communicate with your website visitors and social media followers in real-time. This is in stark contrast to other content of your brand that's usually consumed passively. Such engagement can keep people on your website for longer and help in driving sales and improving your SEO.

• Data Collection

Chatbots can be a great way to collect your audience data. They can communicate with your audience and gather information such as their names, email addresses, and more. You can easily access these details by integrating the chatbot with your CRM.

Disadvantages of using AI Chatbots

• Data Security

When you collect your audience data, it's your responsibility to keep it secure. The data needs to be transmitted from the chatbot to your CRM in a secure manner. It must also be stored securely and only relevant data should be collected from your audience.

• Inability to Understand Emotions

Chatbots are codes and hence, they find it difficult to ascertain the emotions of the user. As a result, they may not be able to understand if the user they are chatting with is happy, agitated, or sad. This might lead to the

chatbot coming across as emotionally insensitive and it can harm your brand’s reputation.

To reduce the chances of such a situation, you should consider using chatbots that allow customer support agents to take over the conversation.

Do you want to incorporate chatbots into your marketing strategy? Check out how conversational AI-powered chatbots can help you.

11 CONCLUSION

Hence we had now made this banking chatbot with a condition based system. It is useful to get knowledge about banking sectors, banking credentials, payments, credit cards and loans.

Instead of going to the bank directly we can access it or directly we could check whatever we want:

- Pushes Information
- Answers Queries

Bank executives primarily have 2 jobs:

- Sell plans
- Solve customer queries (and eventually use it as leverage to, guess what, sell more plans)

Does selling plans generate revenue for the bank? Yes. Does solving customer queries generate revenue for the bank? No.

Banking chatbots can take over the mundane, non-revenue-generating tasks and support multiple customers at the same time. Moreover, every time there is a change, chatbots need to be told only once, unlike training executives repeatedly to ensure that they deliver the right message across.

Eg: Swedbank's Nina takes care of service calls while the employees do the sales calls to add more value to the company.

- Saves Time
- Builds Economy
-

As much as we hate to admit it, we have time but it's never enough.

A new visitor must be saved the hassle of going through the bank's entire website to find 1 piece of information. An executive doesn't have to entertain queries when an AI-powered chatbot can do it uninterruptedly without getting bored, seeking motivation, taking breaks, or demanding an appraisal.

Even though banks provide a plethora of services to their customers, not every service can have the correct taker for it. Thus in order to provide the customers with personalized services, banks can accomplish this particular goal by deploying chatbots. As a matter of fact, the delivery of personalized services can improve the overall rates of conversion by 25%.

12 FUTUER SCOPE

FUTURE SCOPE

- We can implement with secure payment sending feature like using a chatbot ID's from one chatbot to another chatbot. With max level encryption for security purposes.
- We can also know the NEFT score which we say with our voice which uses a machine learning algorithm.
- We can add the calc's like SIP, PPF which helps the business men to find the calculation on these sectors.
- Most importantly, we could add a voice recognition feature in this chatbot which could even help those blinds.
- this chatbot we could be infused with NLP, which helps rural and illiterate people to understand and communicate with it.
- We could train models with AI, which may be useful for that to learn itself by gaining experience with user queries.

13 APPENDIX

13.1 SOURCE CODE

chatbot.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Output</title>
<link rel="stylesheet"
      href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
<style>
body
{
background-image: url("https://www.apptunix.com/blog/wp-content/uploads/sites/3/2821/04/show-chatbots-for-banking.jpg");
background-size: cover;
}
</style>
</head>
<body>
  <script>
      window.watsonAssistantChatOptions = {
       integrationID: "ce848dc8-94da-49ca-8b4d-975dda50b7d8", // The ID of this integration.
```

```
        region: "us-south", // The region your integration is hosted in.
        serviceInstanceID: "ab8c28de-50d3-46b9-af60-4b74a64b97bf", // 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>
```

13.2 SOURCE CODE – PYTHON

Chatbot.py

```
from flask import Flask, render_template
app= Flask(__name__)
@app.route('/')
def bot():
   return render_template('chatbot.html')
if __name__=='__main__':
       app.run()
```

Github Link - **https://github.com/IBM-EPBL/IBM-Project-44780-1660726723**

Project Demo Link (Youtube) - https://youtu.be/pGMefUJc9Ro

Project Demo Link (Drive) - https://drive.google.com/file/d/1rT6Gcuk7zEqoearA3XGQQ2gDVbYQEuvX/view?usp=sharing