

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

1.CONVERSION TO AUTOMATION IN BANKING THROUGH CHATBOT USING ARTIFICIAL MACHINE INTELLIGENCE LANGUAGE:

AUTHOR NAME:

Sasha Fathima Suhel, Vinod Kumar Shukla, Sonali Vyas, Ved Prakash Mishra

DESCRIPTION:

Artificial Machine Intelligence is a very complicated topic. It involves creating machines that are capable of simulating knowledge. Based on basic A.I. (Artificial Intelligence) structuring and working for this, System-Chatbots are made (or chatter bots). However, this paper provides a new concept which addresses machine intelligence and sheds light on the potential of intelligent systems. The rise of chatbots in the finance sector is the latest disruptive force that has changed the way customers interact

2. A STUDY OF APPLICATIONS OF ARTIFICIAL INTELLIGENCE IN BANKING AND FINANCE SECTOR

AUTHOR NAME:

Lakshkaushik Dattatraya Puri

DESCRIPTION:

Artificial Intelligence (AI) is reckless growing as the go-to technology for companies across the world to personalize experience for individuals. Banking sector is becoming one of the first adopters of AI and just like other segments, banks are exploring and implementing the technology in various ways. The rudimentary applications AI include bring smarter chat-bots for customer service, personalizing services for individuals, and even placing an AI robot for self-service at banks. Beyond these basic applications, banks can implement the technology for bringing in more efficiency to their back-office and even reduce fraud and security risks.

3. VIRTUAL BANK ASSISTANCE: AN AI BASED VOICE BOT FOR BETTER BANKING

AUTHOR NAME:

Sarath Chandra Oruganti

DESCRIPTION:

The system is designed for banks to use where users can ask any bank related questions like loan, account, policy, etc. which are bank related queries. The system recognizes the user's query and understands what he wants to convey and simultaneously answers them appropriately. At present, there are chat applications for banks. The questions asked by the users can be in any format. There is no specific format for users to ask questions and provides suitable answers to the user. These voice bots can be built from scratch or they can be deployed on existing chat-bots by enabling them with voice services. It also uses a graphical representation (if necessary)

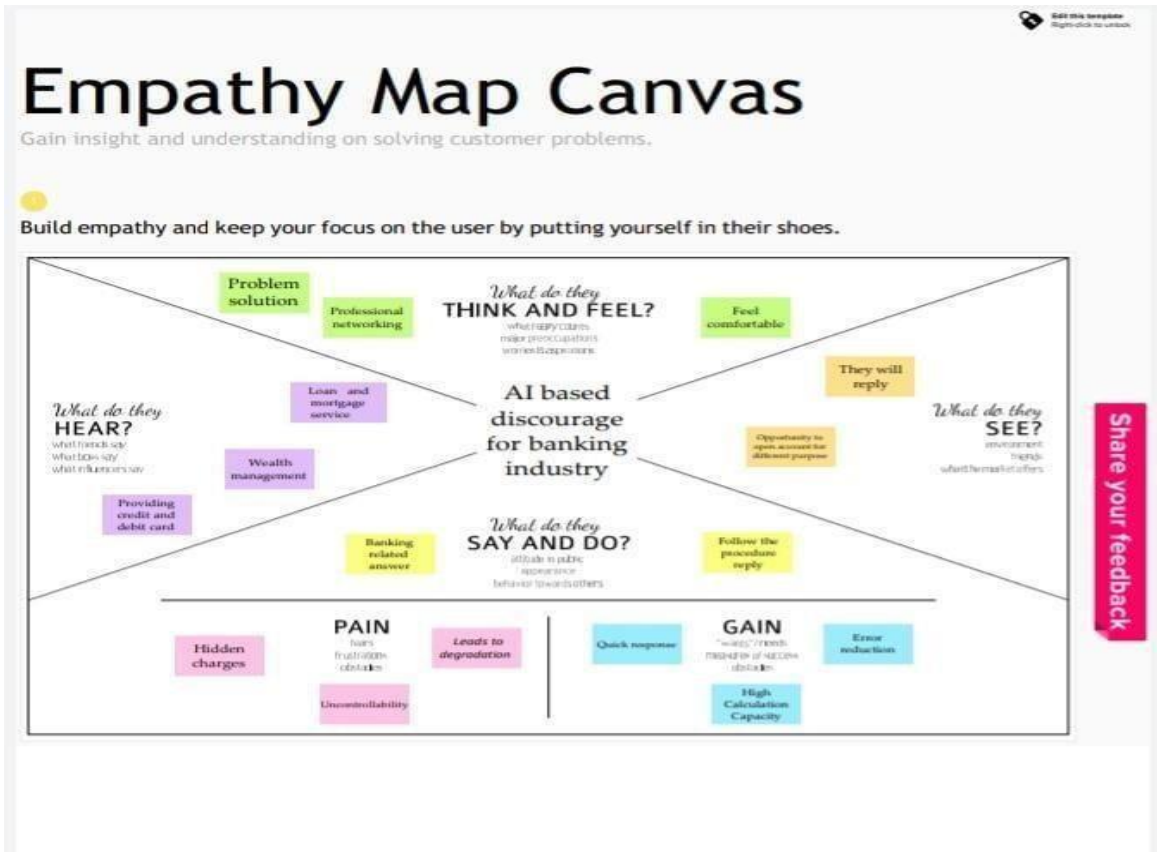
of a person speaking while giving answers as a real person would do as an employee. Bank bot solves the issues a user has and clarifies it with its knowledge

2.2 Problem Statement Definition

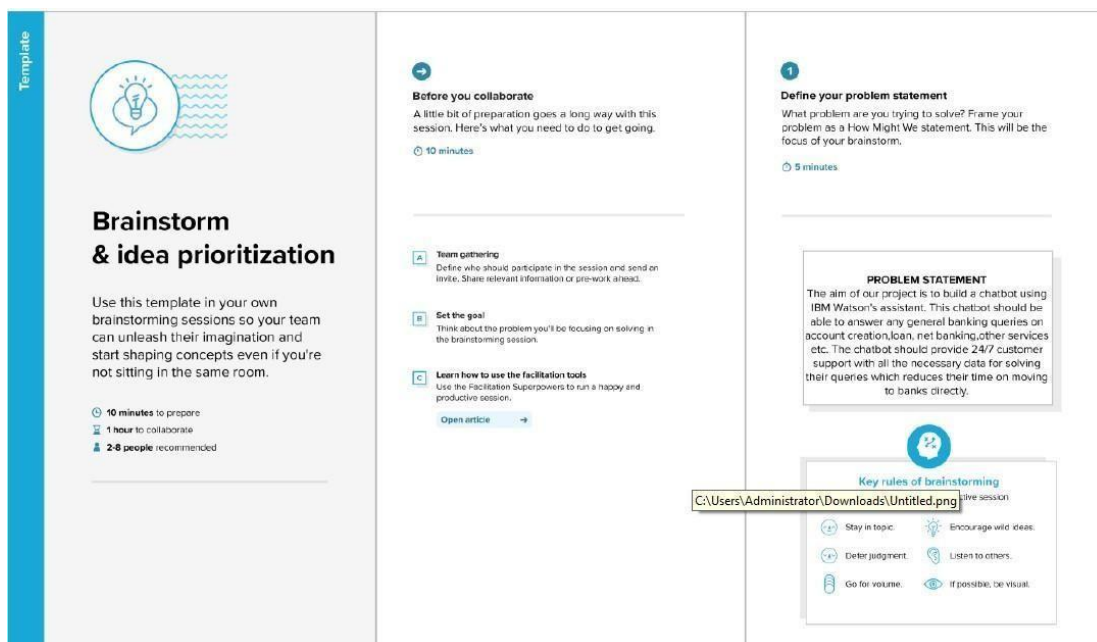
Banking is one the crucial sectors, it deals with financial transactions which can be availed by everyone, but banks are not able to resolve the queries of customers at all times related to the products or services in a satisfactory way in turn hinders the customer satisfaction. In order to guide the 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



2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP
You can select a sticky note and hit the pencil (switch to sketch) icon to start drawing!

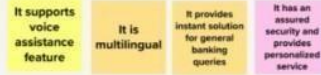
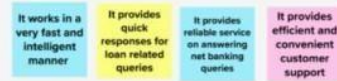
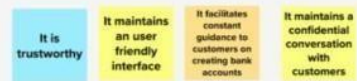
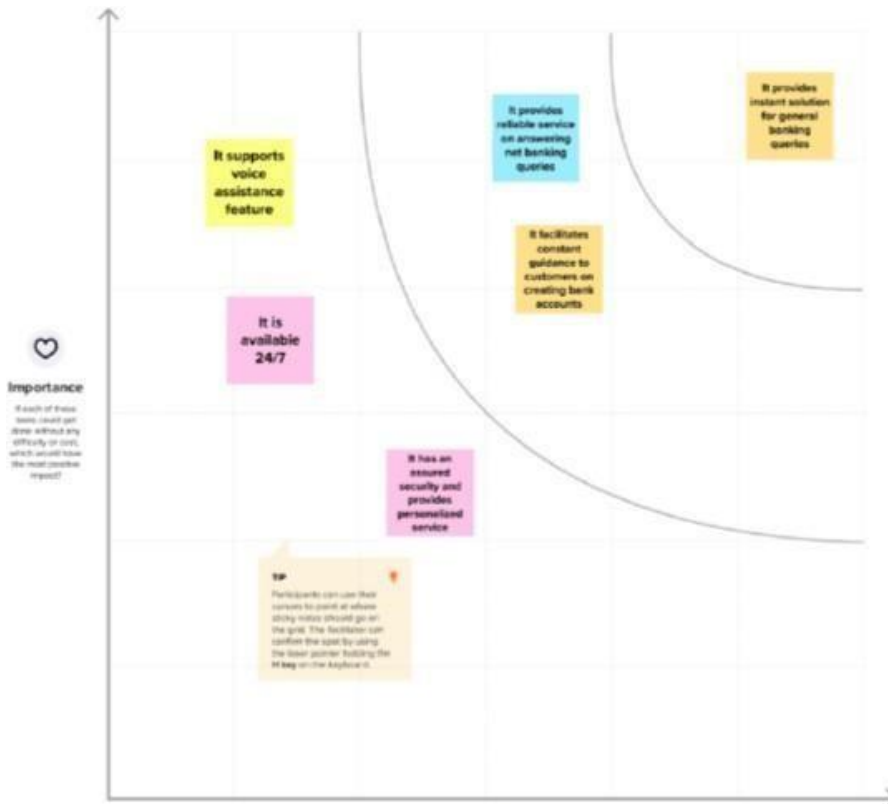
ABHISHEK B**ADITHYA R****PRATHYUSH KRISHNEN J****RAMKUMAR K**

3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you can break it up into smaller sub-groups.

20 minutes

Using Natural Language Processing**Using Neural network****Using Artificial Intelligence****Using Cloud technology**

3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	The main purpose of chatbots in banking is providing a better customer experience. However, they also help the staff and prevent stressful situations that arise from direct communication with clients. For this the chatbot is developed.
2.	Idea / Solution description	In this we will build a chatbot using Watson's assistant. The system is designed for banks to use where users can ask any bank related questions like loan, account, policy, etc. The questions asked by the users can be in any format. The system recognizes the user's query and understands what he wants to convey and simultaneously answers them appropriately.
3.	Novelty / Uniqueness	A chatbot can easily save time and effort. These AI chatbots communicate with humans in a very natural form to resolve their queries and to assist them as required.
4.	Social Impact / Customer Satisfaction	Customers of the bank – Depositors, borrowers, lenders etc.
5.	Business Model (Revenue Model)	Banks will enable much and reliable services which will gain customer loyalty.
6.	Scalability of the Solution	In financial markets, scalability describes an institution's ability to handle increased market demands; in the corporate world, a scalable company is one that can maintain or improve its profit margins while sales volume increases.

3.4 Problem Solution fit

Project Title: Personal Assistance for Seniors Who Are Self-Reliant		Project Design Phase – 1		Team Id: PNT2022TIMID29810	
Problem Solution Fit					
Define CS, fit into	1. CUSTOMER SEGMENT(S) <small>Who is your customer?</small> <div> <p>The persons who need another person (caretaker) to monitor them to take the pills are my customers.</p> <p>Eg: elderly people of 65+ years</p> </div>	6. CUSTOMER <small>What constraints prevent your customers from taking action or keep them choosing solutions? (e.g. spending power, budget, no cash, network connectivity, available devices)</small> <div> <p>The only constraint that prevents the customers from taking pills is when the pills are over.</p> </div>	5. AVAILABLE SOLUTIONS <small>Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pain & costs do these solutions have? (e.g. cost and paper is an alternative to digital monitoring)</small> <div> <p>There are some solutions through an app the person's health will be monitored. People in the past had no monitoring equipment's their caretakers reminds them with a call.</p> </div>	Explore AS.	
	2. JOBS-TO-BE-DONE / PROBLEMS <small>What job(s) do you (or problem) do you address for your customer? There could be more than one, explore different roles.</small> <div> <p>I will address my customers to arrange some caretakers.</p> <p>I will built an application where it reminds with a call or a notification message.</p> </div>	3. PROBLEM ROOT CAUSE <small>Where is the real reason that this problem exists? What is the back story behind the need to do this job? (e.g., customers have to do it because of the change in reg options)</small> <div> <p>The real reason behind this problem is some of the elders take wrong pills instead of the correct one. So this leads to under or over dosing. It will also has some side effects and health issues when we took the wrong one.</p> </div>	7. BEHAVIOUR <small>What does your customer do to address the problem and get the job done? (e.g., already related) Find the right make, point position, resource usage and benefits. Indirectly associated customer spend time for a no volunteering work (e.g. Groupwork)</small> <div> <p>The behaviour of the customers:</p> <p>Directly related: Find a good caretaker for me</p> <p>Indirectly associated: Some of my friends and my neighbours appointed caretaker to monitor their health.</p> </div>	Focus on J&P, tap into BE, understand	
Identify strong TR & EM	3. TRIGGERS <div> <p>To take the pills regularly at the correct time triggers the people to use the automatic medicine reminder system</p> </div>	10. YOUR SOLUTION <div> <p>An app is built for the user (caretaker) which enables him to set the desired time and medicine. These details will be stored in the IBM Cloudant DB. If the medicine time arrives the web application will send the medicine name to the IoT Device through the IBM IoT platform. The device will receive the medicine name and notify the user with voice commands.</p> </div>	8. CHANNELS of BEHAVIOUR <small>8.1 ONLINE</small> <div> <p>The user need to login to the web application to check the pills are over or not.</p> </div> <small>8.2 OFFLINE</small> <div> <p>The user and the volunteers can spread awareness to the people regarding the web application usage using which they can check the availability of pills.</p> </div>	Extract online & offline CH of BE	
	4. EMOTIONS: BEFORE / AFTER <div> <p>Before the people had a fear of forgetting the pills to be taken before the meal or after the meal.</p> <p>After using the reminder system the people's fear about the pills is resolved</p> </div>				

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

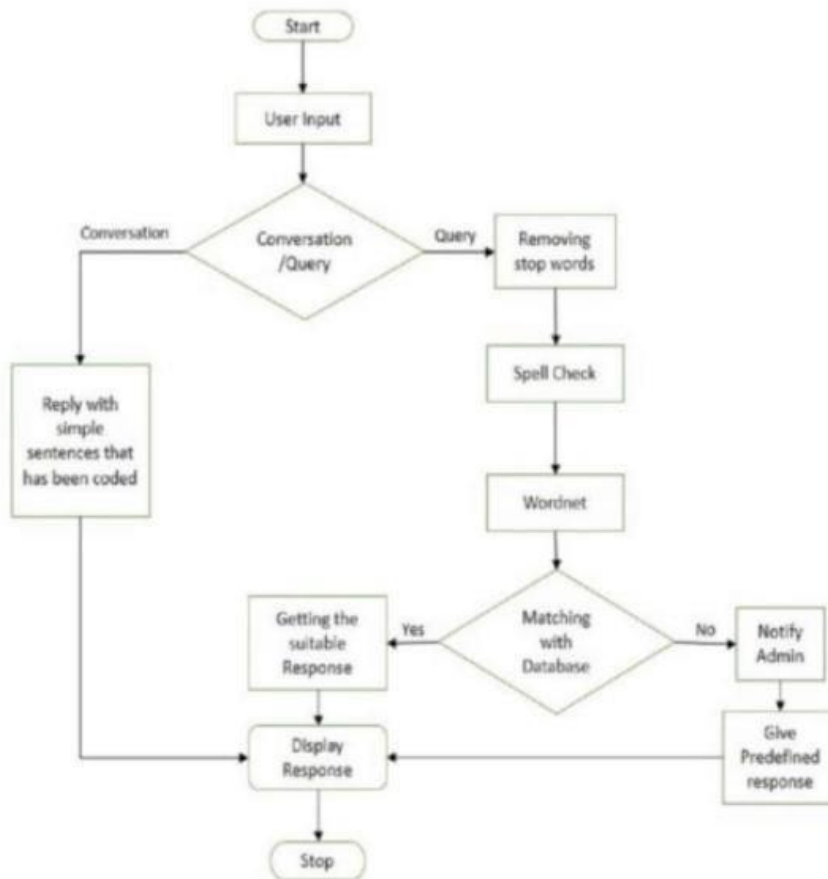
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Requirement	As a user, I can see a Watson assistant, chatbot having banking related skills
FR-2	User confirmation	As a user I can converse with the chatbot regarding saving account related queries and chatbot regarding current account related queries
FR-3	Loan Account Related Actions	As a user I can converse with the chatbot regarding loan account related queries and regarding net banking queries
FR-4	General Queries Related Actions	As a user I can see a flask page for bank and web page integrated with the chatbot
FR-5	Net Banking Related Actions	As a user I can communicate with the chatbot 24*7

4.2 Non-Functional requirements

NFR No.	Non-Functional Requirement	Description
NFR1	Usability	Chatbots developed using AI should be able to answer any general banking queries on account creation, loan, net banking, other services etc. It addresses the queries of customers immediately and effectively in a cost-efficient manner.
NFR2	Security	The AI Chatbot maintains a confidential conversation with customers. Chatbot will provide personal and efficient communication between the user and the bank.
NFR3	Reliability	Chatbots are trained very well using AI to provide solutions for the popular and frequently asked questions, thereby providing the best suited service quickly. Thus, AI Chatbots has a reliable end-user experience.
NFR4	Performance	AI Chatbots is a great way to overcome the limitation of workload of humans. There can be multiple instances of a single chatbot inquiring about different people at the same time. Such chatbots work in real time with no need for the customers to wait. This ensures faster, easier and more efficient face-time with customers.
NFR5	Availability	AI Chatbots provides 24/7 service to clear all customer queries and guide them through all the banking processes. It is available to anyone with access to the internet with basic hardware.
NFR6	Scalability	AI Chatbots is helping the banking industry to scale their customer service and to improve customer service satisfaction at the same time. It can be scaled as per the requirements of the bank to include answers to queries related to any new feature or service introduced by the bank.

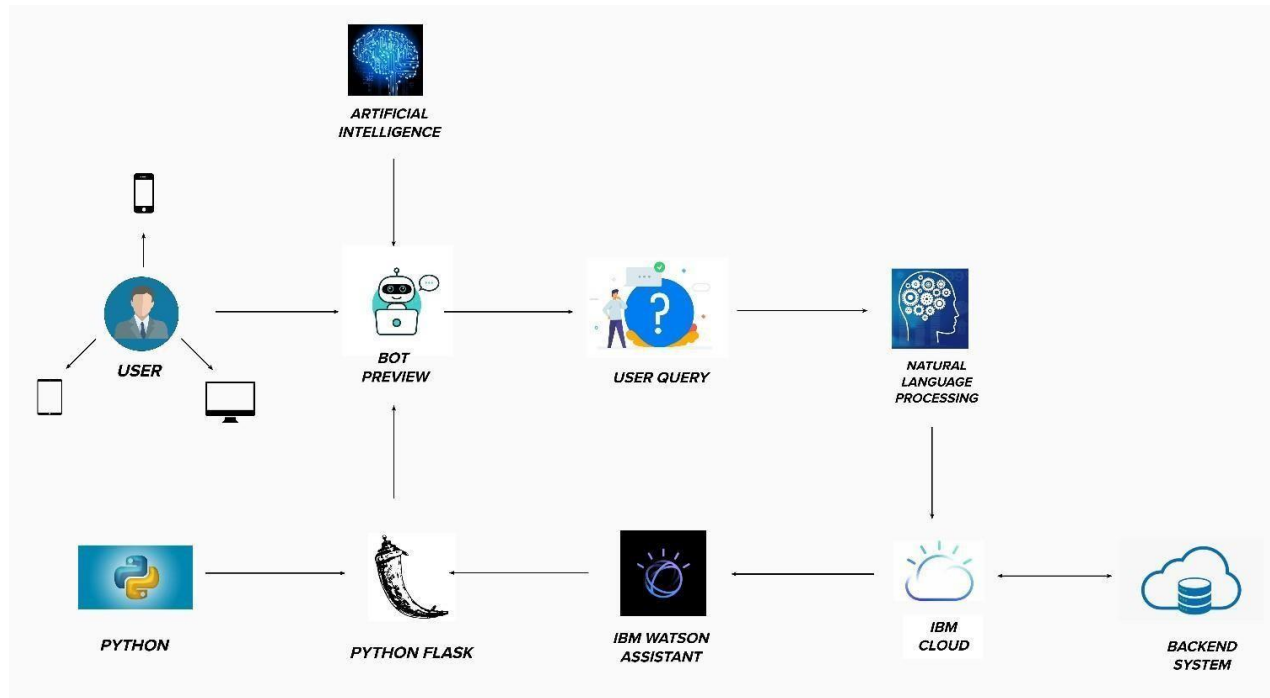
5. PROJECT DESIGN

5.1 Data Flow Diagrams

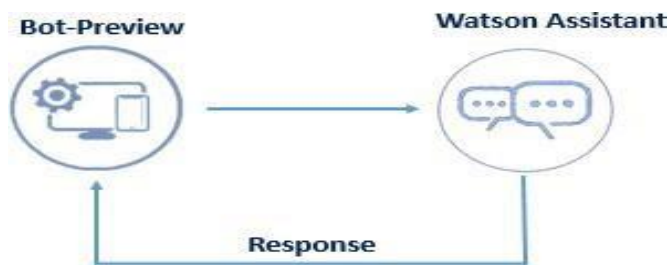


5.2 Solution & Technical Architecture

Solution Architecture



Technical Architecture



5.3 User Stories

User Type	Functional Require men (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile or Web user)	Savings Account Related Actions	USN-1	As a user, in the Savings Account option, I can select Types of Savings Account to get details regarding documents required for creating that savings account.	I can clear my queries regarding types of savings account	High	Sprint-1
		USN-2	As a user, I can check the Interest Rates of Savings Account	I can clear my queries regarding interest rates of savings account	High	Sprint-1
		USN-3	As a user, I can check the Minimum Balance of Savings Account	I can clear my queries regarding minimum balance of savings account	Medium	Sprint-2
	Current Account Related Actions	USN-4	As a user, I can choose the Type of Company to know the information on documents to be submitted for creating current account	I can clear my queries regarding types of companies	High	Sprint-1
		USN-5	As a user, I want to get details on procedure to close my Current Account	I can clear my queries regarding current account closure	High	Sprint-2
	Loan Account Related Actions	USN-6	As a user, I can choose the Type of Loans to know the information on choosing an essential loan scheme	I can clear my queries regarding types of loan account	High	Sprint-1
		USN-7	As a user, I can check the Loan Amounts that can be offered for corresponding Loan Accounts chosen	I can clear my queries regarding loan amounts of loan account	High	Sprint-2
		USN-8	As a user, I can check the Status of Loan for my Loan Accounts	I can clear my queries regarding loan status of loan account	Low	Sprint-2

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
-----------	-------------------------------	-------------------	-------------------	---------------------	----------	---------

		USN-10	As a user, I want to check my CIBIL score for my loan application and to ensure whether my loan application is approved by the bank.	I can clear my queries regarding CIBIL score of loan application	Medium	Sprint 3
		USN-11	As a user, I want to get the procedure details for maintaining Storage Locker facility of my bank account	I can clear my queries regarding storage locker facilities of bank account	High	Sprint-3
	Net Banking Related Actions	USN-12	As a user, I want to get the procedure details for changing the Net Banking password of my bank account	I can clear my queries regarding change of net banking password	Medium	Sprint-2
		USN-13	As a user, I can select types of fund transfers to get details regarding different services available in net banking	I can clear my queries regarding types of fund transfers in net banking	High	Sprint-3
		USN-14	As a user, I want to get the procedure details for adding beneficiaries to my net banking account.	I can clear my queries regarding adding beneficiaries in net banking	Low	Sprint-3
Administrator or		USN-15	As an admin, I can change responses to queries and modify them as and when needed.	I can modify responses of the chatbot	Medium	Sprint-1
		USN-16	As an admin, I can add more options to queries and add new options as new features get added.	I can add more options and queries into the chatbot	Medium	Sprint-1

6. PROJECT PLANNING AND SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Savings Account Related Actions	USN-1	As a user, in the Savings Account option, I can select Types of Savings Account to get details regarding documents required for creating that savings account.	4	High	VINISHAR
Sprint-1		USN-2	As a user, I can check the Interest Rates of Savings Account	4	High	VINISHA R
Sprint-1		USN-3	As a user, I can check the Minimum Balance of Savings Account	3	Medium	VINISHA R
Sprint-1	Current Account Related Actions	USN-4	As a user, I can choose the Type of Company to know the information on documents to be submitted for creating current account	5	High	SANDHIYA P SHAHEERAB ANU C
Sprint-1		USN-5	As a user, I want to get details on procedure to close my Current Account	4	High	SANDHIYA P SHAHEERAB ANU C
Sprint-2	Loan Account Related Actions	USN-6	As a user, I can choose the Type of Loans to know the information on choosing an essential loan scheme	3	High	YOGAMALYA P
Sprint-2		USN-7	As a user, I can check the Loan Amounts that can be offered for corresponding Loan Accounts chosen	3	High	YOGAMALYA P
Sprint-2		USN-8	As a user, I can check the Status of Loan for my Loan Accounts	1	Low	YOGAMALYA P
Sprint-2	General Queries Related Actions	USN-9	As a user, I want to get the procedure details for Currency Conversion facility of my bank account	2	Medium	SANDHIYA P SHAHEERAB ANU C
Sprint-2		USN-10	As a user, I want to check my CIBIL score for my loan application and to ensure whether my loan application is approved by the bank.	1	Low	SANDHIYA P YOGAMALYA P
Sprint-2		USN-11	As a user, I want to get the procedure details for maintaining Storage Locker facility of my bank account	3	High	SANDHIYA P YOGAMALYA P

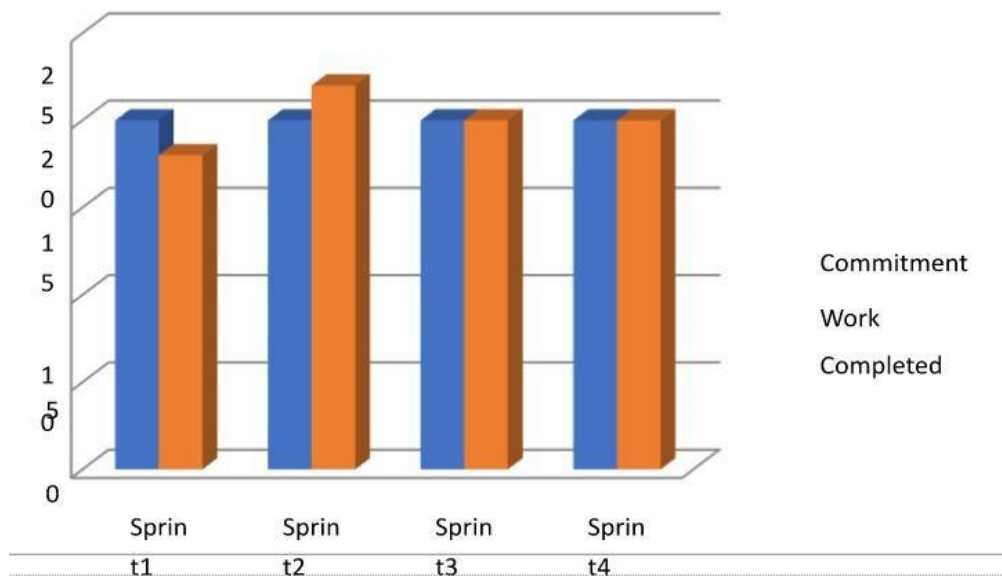
Sprint-2	Net Banking Related Actions	USN-12	As a user, I want to get the procedure details for changing the Net Banking password of my bank account	3	High	SANDHIYA P
Sprint-2		USN-13	As a user, I can select types of fund transfers to get details regarding different services available in net banking	2	Medium	SANDHIYA P
Sprint-2		USN-14	As a user, I want to get the procedure details for adding beneficiaries to my net banking account.	2	Medium	SANDHIYA P
Sprint-3	Web Application	USN-15	As a user, I want to access the chatbot in a web browser that can be accessed from almost all devices.	20	High	SANDHIYA P YOGAM ALYA P SHAHEE RABAN U C
Sprint-4	User Interface and Web Pages	USN-16	As a user, I want to view pages of the banking website and have access to the chatbot easily.	20	High	SANDHIYAP VINISH A R

6.2 Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Average Velocity
Sprint-1	20	Days	20/6 = 3.33
Sprint-2	20	Days	20/6 = 3.33
Sprint-3	20	Days	20/6 = 3.33
Sprint-4	20	Days	20/6 = 3.33
Overall	80	Days	80/24 = 3.33

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	18	02 Nov 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	22	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 chart:



Stand-up Meeting Screenshots

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

7.1 Feature 1

Python Flask

Python Flask is used to develop chatbot applications using python. Flask is mainly used to render and integrate the chatbot application in the browser by providing API. By running the python application, the suitable server domain link is obtained and run in the browser.

HTML

The HTML and CSS is used to design the overall chatbot UI. HTML is used to add UI components and CSS is used to add style to those components. IBM Watson assistant deploys HTML code to train the Chatbot.

Build PYTHON FLASK Code:

APP.PY

APP.PY

```
import email from email import message
from importlib.resources import contents
from tkinter import S from turtle import
title
from flask import Flask, redirect, render_template, request, session, url_for, Flask from
pyexpat import model
from werkzeug.utils import secure_filename
import ibm_db from flask_mail
import Mail, Message from
markupsafe import escape
from flask import Flask, render_template, request import
requests

app = Flask(name) app.secret_key =

b'_5#y2L"F4Q8z\n\Xec]/' mail =

Mail(app)

conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=1bbf73c5-d84a-4bb0-85b9-
ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=32286;SECURITY=SSL;SSLServerC
ertificate=DigiCertGlobalRootCA.crt;UID=hzej88231;PWD=z8f4ZiZ171T0FvR1",",") print(conn)
print("connection successful...")
```

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

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

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

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

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

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

```
@app.route('/register', methods=['GET', 'POST'])
def register():
    if request.method == 'POST': uname =
        request.form['uname'] mail =
        request.form['email'] phone =
        request.form['phone'] password =
        request.form['password']

    sql = "SELECT * FROM customer WHERE email=?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt,1,mail)
    ibm_db.execute(stmt) account =
    ibm_db.fetch_assoc(stmt)

    if account: return render_template('index.html', msg="You are already a member, please login using your
        details....")

    else:
        insert_sql = "INSERT INTO customer VALUES (?, ?, ?, ?)" prep_stmt
        = ibm_db.prepare(conn, insert_sql) ibm_db.bind_param(prepare_stmt,
        1,      uname)      ibm_db.bind_param(prepare_stmt,      2,      mail)
        ibm_db.bind_param(prepare_stmt,      3,      phone)
        ibm_db.bind_param(prepare_stmt,      4,      password)
```

```
ibm_db.execute(prepare_stmt) return render_template('login.html',
msg="Student Data saved successfully..")
```

```
@app.route('/signin', methods=['GET', 'POST'])
def signin(): sec = " if request.method ==
'POST':
```

```
mail = request.form['email']
password = request.form['password']
```

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

```
session["mail"] = escape(mail)
session["password"] = escape(password)
return redirect(url_for('index'))
```

```
else: return render_template('login.html',msg = "Invalid email/ Password or Not
```

```
registered!!?") return "not going to happen dickhead!!??" if name == 'main':
```

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

HOME.HTML

```
<!DOCTYPE html>
<html style="font-size: 16px;" lang="en"><head>
<body>
```

```
<!--Chat Integration:==!>
```

```
<script>
window.watsonAssistantChatOptions =
{ integrationID: "1206a417-c0534bee-
a810-037a99f37b2a", // The ID of this
integration.
region: "us-south", // The region
your integration is hosted in.
serviceInstanceID:
"e81248c69605-415f-843b-
666356f451ec", // 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.c
loud/versions/" +
(window.watsonAssistantChatOptions.cl
ientVersion || 'latest') +
"/WatsonAssistantChatEntry.js";
    document.head.appendChild(t);
    });
</script>

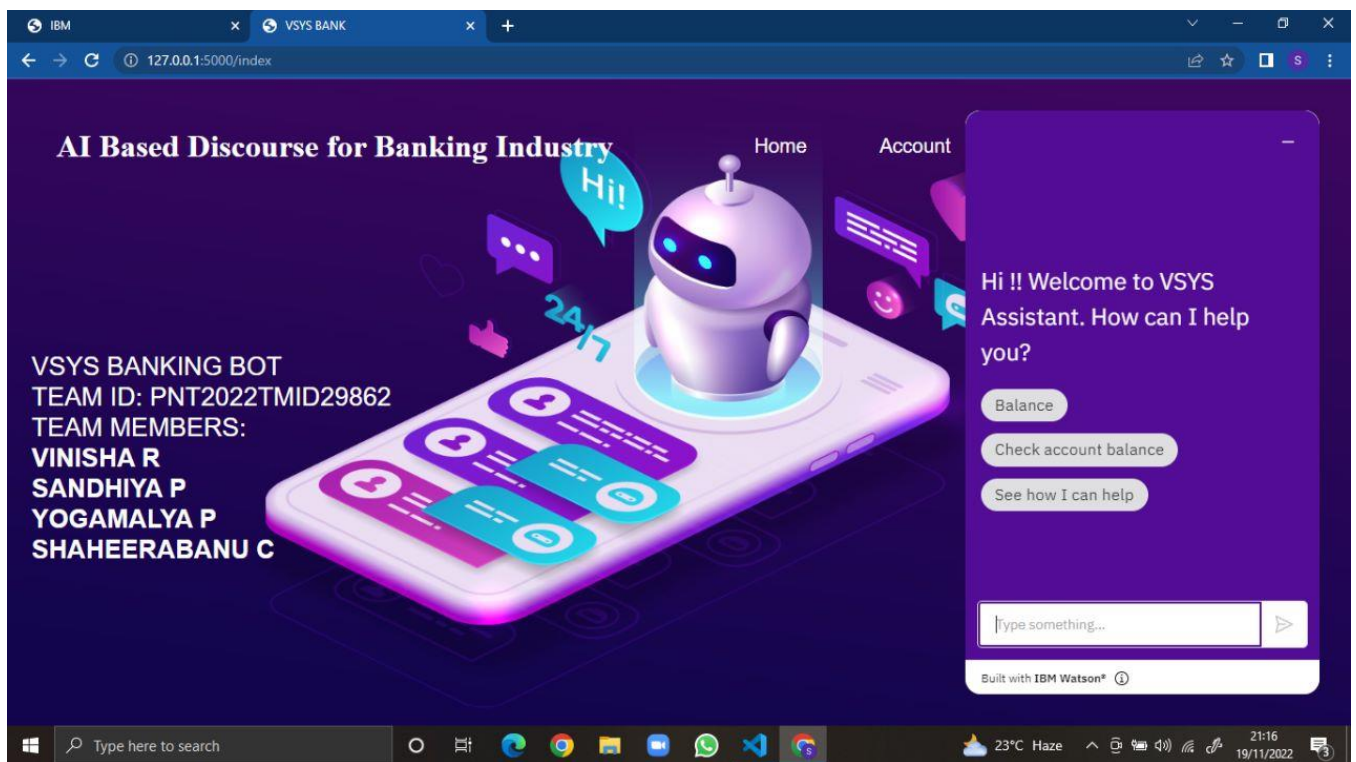
</body>

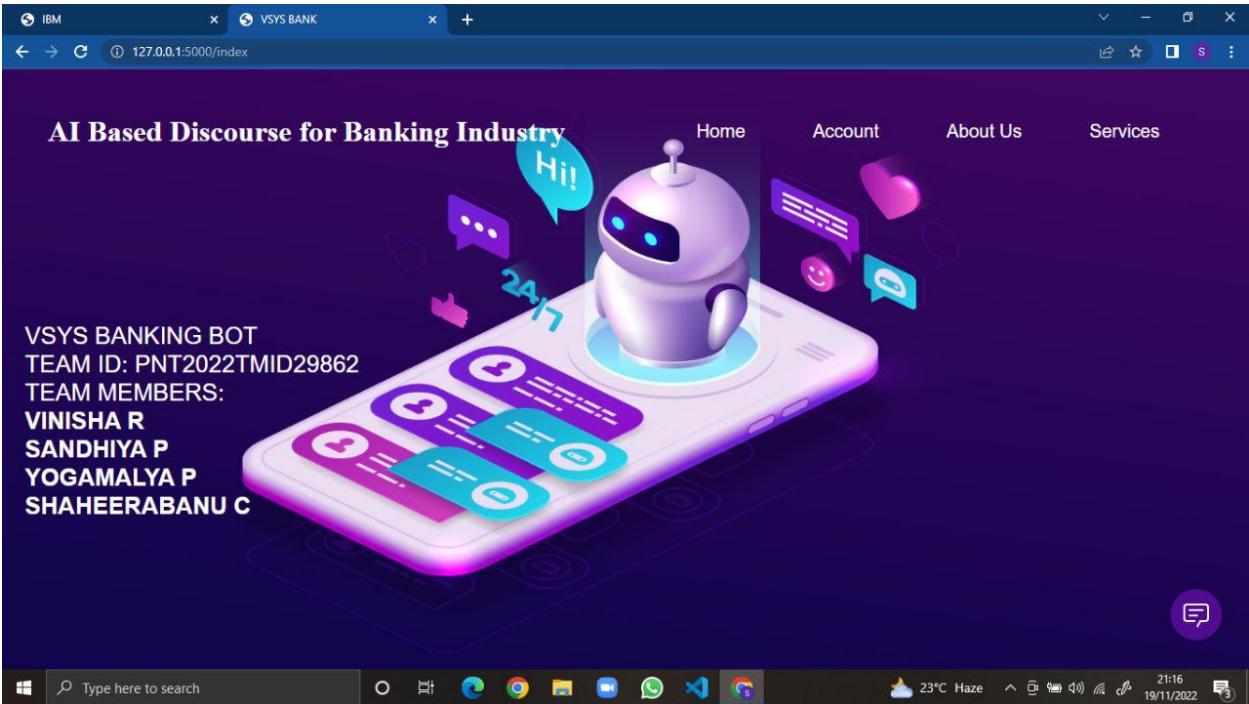
</html>
```

7.2 FEATURE 2

IBMCHATBOT:

- Our chatbot is able to guide a customer to create a bank account. (Both current and savings account)
- Our chatbot is able to answer loan queries.
- Our chatbot is able to answer general banking queries.
- Our chatbot is able to answer queries regarding net banking





8. TESTING

8.1 Test Cases

	Test Scenarios
1	Verify user is able to open and view chatbot UI
2	Verify user is able to interact with chatbot or not
3	Verify chatbot is able to respond to user queries immediately
4	Verify chatbot is able to provide options for user to choose various choices
	Savings Account Related Actions
1	Verify user is able to select type of savings account
2	Verify user is able to know the procedure to create savings account for selected type
3	Verify user is check the minimum balance
4	Verify user is able to find interest rate
	Current Account Related Actions
1	Verify user is able to select type of company
2	Verify user is able to know the procedure to create current account for selected type
3	Verify user is able to know about zero balance current account
4	Verify user is able to know the procedure to close current account
	Loan Account Related Actions
1	Verify user is able to choose options for selecting type of available loan policies
2	Verify user is able to know about available loan amounts
3	Verify user is able to check the loan status
4	Verify user is able to know about joint loan
	General Query Related Actions
1	Verify user is able to know about bank working days

2	Verify user is able to know about list of branches
3	Verify user is able to find the nearest branch
4	Verify user is able to know about storage locker facility
5	Verify user is able to know about currency conversion facility
	Net banking Related Actions
1	Verify user is able to know the procedure to login net banking account
2	Verify user is able to know the procedure to change net banking password
3	Verify user is able to choose options for selecting type of fund transfers
4	Verify user is able to know about daily transaction limit

8.2 User Acceptance Testing

Test case ID	Feature Type	Component	Test Scenario	Pre-Requlite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	BUG ID	Executed By
Chatbot_TC_001	UI	Home Page	Verify user is able to open and view chatbot UI	None	1. Enter URL and click go 2. Click on chatbot icon 3. Verify chatbot UI displayed or not	URL link	Chatbot UI is able to be viewed by user	Working as expected	Pass	-	N	-	Adithya R
Chatbot_TC_002	Functional	Home Page	Verify user is able to interact with chatbot or not	Chatbot is accessible	1. Click on chatbot icon 2. Type the query in the chat window 3. Press enter	User query	User is able to interact with chatbot easily	Working as expected	Pass	-	N	-	Prathiyush Krishnen J
Chatbot_TC_003	Functional	Home page	Verify chatbot is able to respond to user queries immediately	Chatbot is accessible	1. Click on chatbot icon 2. Type the query in the chat window 3. Press enter	User query	Chatbot is able to provide instant replies for user queries	Working as expected	Pass	-	N	-	Ramkumar K
Chatbot_TC_004	Functional	Home page	Verify chatbot is able to provide options for user to choose various choices	Chatbot is accessible	1. Click on chatbot icon 2. Type the query in the chat window 3. Press enter	User query	Chatbot is able to provide options for user to choose various choices	Working as expected	Pass	-	N	-	Abhishek B
Savings_TC_001	UI	Chatbot	Verify user is able to select type of savings account	Chatbot is accessible	1. Click on chatbot icon 2. Type the query in the chat window 3. Press enter 4. Select the desired option	User query	User is able to select type of savings account	Working as expected	Pass	-	N	-	Adithya R
Savings_TC_002	Functional	Chatbot	Verify user is able to know the procedure to create savings account for selected type	Chatbot is accessible	1. Click on chatbot icon 2. Type the query in the chat window 3. Press enter 4. Select the desired option	User query	User is able to know the steps to create savings account for selected type	Working as expected	Pass	-	N	-	Adithya R
Savings_TC_003	Functional	Chatbot	Verify user is check the minimum balance	Chatbot is accessible	1. Click on chatbot icon 2. Type the query in the chat window 3. Press enter 4. Select the desired option	User query	User is able to check the minimum balance	Working as expected	Pass	-	N	-	Adithya R
Savings_TC_004	Functional	Chatbot	Verify user is able to find interest rate	Chatbot is accessible	1. Click on chatbot icon 2. Type the query in the chat window 3. Press enter 4. Select the desired option	User query	User is able to find interest rate	Working as expected	Pass	-	N	-	Adithya R
Current_TC_001	UI	Chatbot	Verify user is able to select type of company	Chatbot is accessible	1. Click on chatbot icon 2. Type the query in the chat window 3. Press enter 4. Select the desired option	User query	User is able to select type of company	Working as expected	Pass	-	N	-	Prathiyush Krishnen J
Current_TC_002	Functional	Chatbot	Verify user is able to know the procedure to create current account for selected type	Chatbot is accessible	1. Click on chatbot icon 2. Type the query in the chat window 3. Press enter 4. Select the desired option	User query	User is able to know the steps to create current account for selected type	Working as expected	Pass	-	N	-	Prathiyush Krishnen J
Current_TC_003	Functional	Chatbot	Verify user is able to know about zero balance current account	Chatbot is accessible	1. Click on chatbot icon 2. Type the query in the chat window 3. Press enter 4. Select the desired option	User query	User is able to know about zero balance current account	Working as expected	Pass	-	N	-	Prathiyush Krishnen J
Current_TC_004	Functional	Chatbot	Verify user is able to know the procedure to close current account	Chatbot is accessible	1. Click on chatbot icon 2. Type the query in the chat window 3. Press enter 4. Select the desired option	User query	User is able to know the procedure to close current account	Working as expected	Pass	-	N	-	Prathiyush Krishnen J
Loan_TC_003	Functional	Chatbot	Verify user is able to check the loan status	Chatbot is accessible	1. Click on chatbot icon 2. Type the query in the chat window 3. Press enter 4. Select the desired option	User query	User is able to check the loan status	Working as expected	Pass	-	N	-	Ramkumar K
Loan_TC_004	Functional	Chatbot	Verify user is able to know about joint loan	Chatbot is accessible	1. Click on chatbot icon 2. Type the query in the chat window 3. Press enter 4. Select the desired option	User query	User is able to know about joint loan	Working as expected	Pass	-	N	-	Ramkumar K
General_TC_001	Functional	Chatbot	Verify user is able to know about bank working days	Chatbot is accessible	1. Click on chatbot icon 2. Type the query in the chat window 3. Press enter 4. Select the desired option	User query	User is able to know about bank working days	Working as expected	Pass	-	N	-	Abhishek B
General_TC_002	Functional	Chatbot	Verify user is able to know about list of branches	Chatbot is accessible	1. Click on chatbot icon 2. Type the query in the chat window 3. Press enter 4. Select the desired option	User query	User is able to know about list of branches	Working as expected	Pass	-	N	-	Abhishek B
General_TC_003	Functional	Chatbot	Verify user is able to find the nearest branch	Chatbot is accessible	1. Click on chatbot icon 2. Type the query in the chat window 3. Press enter 4. Select the desired option	User query	User is able to find the nearest branch	Working as expected	Pass	-	N	-	Abhishek B
General_TC_004	Functional	Chatbot	Verify user is able to know about storage locker facility	Chatbot is accessible	1. Click on chatbot icon 2. Type the query in the chat window 3. Press enter 4. Select the desired option	User query	User is able to know about storage locker facility	Working as expected	Pass	-	N	-	Abhishek B
General_TC_005	Functional	Chatbot	Verify user is able to know about currency conversion facility	Chatbot is accessible	1. Click on chatbot icon 2. Type the query in the chat window 3. Press enter 4. Select the desired option	User query	User is able to know about currency conversion facility	Working as expected	Pass	-	N	-	Abhishek B
Netbank_TC_001	Functional	Chatbot	Verify user is able to know the procedure to login netbanking account	Chatbot is accessible	1. Click on chatbot icon 2. Type the query in the chat window 3. Press enter 4. Select the desired option	User query	User is able to know the steps to login netbanking account	Working as expected	Pass	-	N	-	Adithya R
Netbank_TC_002	Functional	Chatbot	Verify user is able to know the procedure to change netbanking password	Chatbot is accessible	1. Click on chatbot icon 2. Type the query in the chat window 3. Press enter 4. Select the desired option	User query	User is able to know the steps to change netbanking password	Working as expected	Pass	-	N	-	Prathiyush Krishnen J
Netbank_TC_003	UI	Chatbot	Verify user is able to choose options for selecting type of fund transfers	Chatbot is accessible	1. Click on chatbot icon 2. Type the query in the chat window 3. Press enter 4. Select the desired option	User query	User is able to choose options for selecting type of fund transfers	Working as expected	Pass	-	N	-	Ramkumar K
Netbank_TC_004	Functional	Chatbot	Verify user is able to know about daily transaction limit	Chatbot is accessible	1. Click on chatbot icon 2. Type the query in the chat window 3. Press enter 4. Select the desired option	User query	User is able to know about daily transaction limit	Working as expected	Pass	-	N	-	Abhishek B

AI based Discourse for...
Software project

Back to project

Test Cases

Import Tests

PLANNING AND EXECUTION

Cycle Summary

Search Test Executions

Manage Execution Filters

REPORTING

You're in a team-managed project
Learn more

Cycle Summary

User Acceptance Testing

Build : Total Executions : 25 Start Date :
Environment : Cycle Executions : 25 End Date :
Created By : Prathyush Krishnen J Total Executed : 25 Description :
Total Execution Time : 0m Total Logged Time : 0m Executions Not Tracked : 25

Select All Delete Columns Add Tests

ID	Status	Summary	Defect	Compon...	Label	Total	Action
ABDFBI-68	PASS	Verify user is able to know the procedure to create current account for selected type	-	-	-	-	E
ABDFBI-69	PASS	Verify user is able to know	-	-	-	-	E

AI based Discourse for...
Software project

Back to project

Manage Execution Filters

REPORTING

Test Summary

Test Metrics

Traceability Matrix

INTEGRATIONS

API Keys

You're in a team-managed project
Learn more

Executions List

Test Cycle	Test Id	Execution Status	Executed By	Execution Date
User Acceptance Testing	10058	PASS	Prathyush Krishnen J	11/17/2022 5:06:25 PM
User Acceptance Testing	10082	PASS	Prathyush Krishnen J	11/17/2022 5:06:27 PM
User Acceptance Testing	10081	PASS	Prathyush Krishnen J	11/17/2022 5:06:30 PM
User Acceptance Testing	10068	PASS	Prathyush Krishnen J	11/17/2022 5:06:16 PM
User Acceptance Testing	10077	PASS	Prathyush Krishnen J	11/17/2022 5:06:20 PM
User Acceptance Testing	10076	PASS	Prathyush Krishnen J	11/17/2022 5:07:31 PM
User Acceptance Testing	10064	PASS	Prathyush Krishnen J	11/17/2022 5:07:34 PM
User Acceptance Testing	10071	PASS	Prathyush Krishnen J	11/17/2022 5:07:39 PM

Projects / AI based Discourse for Banking / Test Summary

Test Summary

Total Tests
25

Total Unexecuted
0

Total Executed
25

TESTS BY VERSIONS

Search

Name	Tests
Unscheduled	25

Showing 1 - 1 of 1 Prev 1 of 1 Next

TESTS BY COMPONENTS

Search

Name	Tests
No Component	25

Showing 1 - 1 of 1 Prev 1 of 1 Next

TESTS BY LABELS

Search

Name	Tests
No Label	25

Showing 1 - 1 of 1 Prev 1 of 1 Next

Test Report

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the AI based Discourse for Banking Industry project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

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

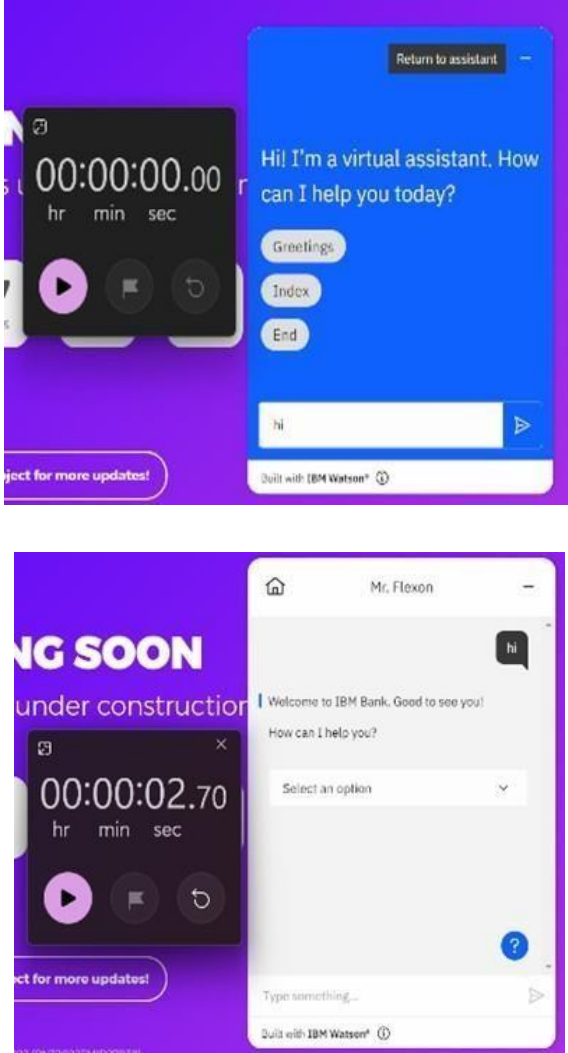
3. Test Case Analysis

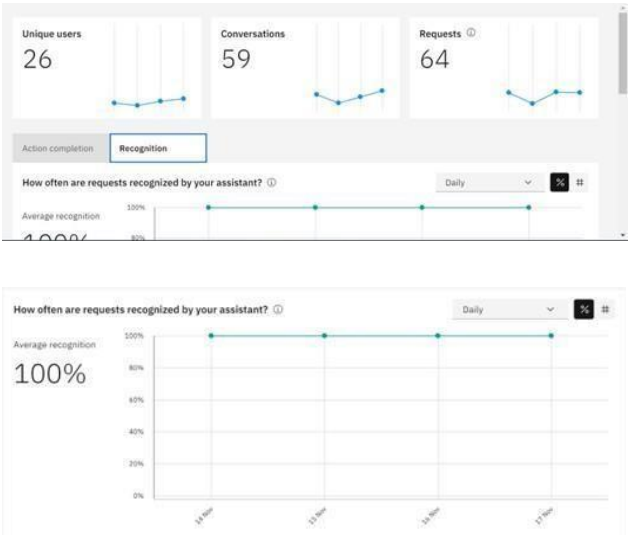
Section	Total Cases	Not Tested	Fail	Pass
Print Engine	0	0	0	0
Client Application	25	0	0	25
Security	0	0	0	0
Outsource Shipping	0	0	0	0
Exception Reporting	0	0	0	0
Final Report Output	25	0	0	25
Version Control	0	0	0	0

9. RESULTS

9.1 Performance Metrics

Model Performance Testing:

S.No.	Parameter	Values	Screenshot
1.	Model Summary	<p>The chatbot is used by the user to interact and select queries that are populated based on frequently asked questions of banking customers. There can be multiple instances of a single chatbot serving different users at the same time. It quickly responds with expected answers to frequently asked customer queries. It can be scaled as per the requirements of the bank to include answers to queries related to any new feature or service introduced by the bank.</p>	

2.	Accuracy	<p>Training Accuracy – 100%</p> <p>Validation Accuracy – 100%</p>	 <p>The screenshot displays a dashboard for an AI assistant's performance. At the top, three summary cards show 'Unique users' (26), 'Conversations' (59), and 'Requests' (64), each with a small line chart. Below these, a section titled 'Action completion' has a 'Recognition' tab selected. It features a line chart titled 'How often are requests recognized by your assistant?' with a 'Daily' filter. The chart shows a constant 100% recognition rate across four dates: 14 Nov, 15 Nov, 16 Nov, and 17 Nov. A large '100%' value is also displayed on the left of the chart area.</p>
----	----------	---	---

10. ADVANTAGES AND DISADVANTAGES

Advantages

1. **Round-the-clock support:** Chatbots provide 24/7 client support, so existing and potential customers can try and solve their banking problems after work hours and on weekends. This ultimately also leads to better customer experience.
2. **Enhanced productivity of bank personnel:** Not all clients' problems require the help of a staff member. Artificial intelligence may successfully deal with minor issues leaving only the most urgent and complicated cases for the human approach.
3. **More convenient mode of communication:** Chatbots may combine various functionalities that would make them convenient for customers of different age groups.

Disadvantages

1. **Internet Issues:** Chatbots will not work without the internet. If there is a network problem, chatbots may stop working and will not respond to client queries.
2. **Requirement of technical knowledge:** Users who make use of chatbots must know how to use chatbots. Otherwise, it is difficult to do interaction with chatbot.
3. **Providing unexpected answers:** Chatbots in some cases may get confused due to prolonged user inputs and may provide irrelevant answers.
4. **Inability to interpret multiple queries:** Chatbots can't understand multiple questions at a time if a user continuously posts queries to it. So, it is necessary for a user to post queries one-by-one.

11. CONCLUSION

Chatbots developed using AI are able to answer any frequently asked banking queries on account creation, loan, net banking, other services etc. It addresses the queries of customers immediately and effectively in a cost-efficient manner. It eliminates the need for a massive customer care workforce and even reduces the workload of the bank employees whose efforts can be used elsewhere. AI Chatbots provides 24/7 service to clear all customer queries and guide them through all the banking processes. It can be scaled as per the requirements of the bank to include answers to queries related to any new feature or service introduced by the bank. 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.

12. FUTURE SCOPE

Chatbots help people obtain the information they need and solve their problems. The technology sector has seen a massive increase in AI development, which has led to the creation of chatbots that can help users easily find the information they need online. AI Chatbots effectively creates a strong brand image. They will continue to evolve and play an important role in customer service for businesses.

1. Bots for Internal Business Communications

Chatbots can be used for various purposes, including addressing common problems, communicating with employees, and finishing HR-related tasks and transactional functions. Chatbots for websites are acting as a guide to new employees through company protocols, recording answers for screen questions, and assisting with the onboarding process for new employees. Chatbots can save time for IT desk agents and help with more complicated issues.

2. Content Marketing

Chatbots can communicate with the target market by speaking with them in complete sentences with a natural and easy-going conversational flow. Some markets use these bots to serve customer service and retail, amongst others. Chatbots can segment consumers for future marketing campaigns. This can be incredibly useful for creating more accurate target marketing. Based on the information from dialogue with chatbots, marketers can use this information to help with personalizing brand content.

3. Social Media

Chatbots have better customer interaction rates on social media. Chatbot interactions increase consumer confidence in a brand or business. Customers are informed with daily or weekly announcements about deals, events, and promotions via social media. With time, it is used for voicing opinions, ordering products and services, offering reviews, and even getting in touch with businesses.

5. Use of AI in contact centers

AI chatbots without human involvement can handle simple requests such as changing a password, requesting a balance, scheduling an appointment, etc.

13. APPENDIX

Source Code

Signup and login pages: Signup.html and:

```
<!DOCTYPE      html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <title>VSYS BANK SIGNUP</title>

<style>
  body {
    background-image: url('static/flight.jpg');
    background-color: #3ca9e2; background-
    repeat:    no-repeat;    background-
    attachment:  fixed;    background-size:
    100% 100%;
    font-family: 'Poppins', sans-serif;

  }

  h2 {
    font-weight: 300;  text-
    align: center;
  }

  p {
    position: relative;
  }

  a, a:link,
  a:visited,
  a:active
  { color:
  #3ca9e2;
  -webkit-
  transition
```

```
: all 0.2s  
ease;  
transition  
: all 0.2s  
ease;  
}
```

```
a:focus, a:hover,  
a:link:focus,  
a:link:hover,  
a:visited:focus,  
a:visited:hover,  
a:active:focus,  
a:active:hover {  
    color: #329dd5; -webkit-  
    transition: all 0.2s ease;  
    transition: all 0.2s ease;  
}
```

```
#login-form-wrap {  
    background-color:  
    #fff; width: 35%;  
    margin-top: 100px;  
    margin-left: 470px;  
    text-align: center;  
    padding: 20px 0 0 0; border-radius: 4px; box-  
    shadow: 0px 30px 50px 0px rgba(0, 0, 0, 0.2);  
    opacity: 0.8;  
}
```

```
#login-form  
{ padding: 0  
60px;  
}
```

```
input {  
    display: block; box-  
    sizing: border-box;  
    width: 100%;
```

```
outline: none;
height: 60px; line-
height: 60px;
border-radius: 4px;
}
```

```
input[type="text"],
input[type="email"]
{width: 100%; padding: 0 0
0 10px; margin: 0; color:
#8a8b8e; border: 1px solid
#c2c0ca; font-style:
normal; font-size: 16px; -
webkit-appearance: none; -
moz-appearance: none;
appearance: none; position:
relative; display: inline-
block; background: none; }
```

```
input[type="text"]:focus,
input[type="email"]:focus
{border-color: #3ca9e2;
}
```

```
input[type="text"]:focus:invalid,
input[type="email"]:focus:invalid
{color: #cc1e2b; border-
color: #cc1e2b; }
```

```
input[type="text"]:valid~.validation,
input[type="email"]:valid~.validation
{display: block; border-
color: #0C0;
}
```

```
input[type="text"]:valid~.validation
input[type="email"]:valid~.validation span
{background: #0C0;
position: absolute;
```

span,

```
border-radius: 6px;
}
```

```
input[type="text"]:valid~.validation
input[type="email"]:valid~.validation span:first-child
{ top: 30px;
  left: 14px;
  width: 20px;
  height: 3px;
  -webkit-transform: rotate(-45deg);
  transform: rotate(-45deg);
}
```

span:first-child,

```
input[type="text"]:valid~.validation
input[type="email"]:valid~.validation span:last-child
{ top: 35px;
  left: 8px;
  width: 11px;
  height: 3px;
  -webkit-transform: rotate(45deg);
  transform: rotate(45deg);
}
```

span:last-child,

```
.validation
{ display: none;
  position: absolute;
  content: " ";
  height: 60px;
  width: 30px;
  right: 15px; top:
  0px;
}
input[type="submit"] { border:
none; display: block;
background-color: #3ca9e2;
color: #fff; font-weight: bold;
text-transform: uppercase;
cursor: pointer; -webkit-
```

```
transition: all 0.2s ease;
transition: all 0.2s ease; font-size:
18px; position: relative; display:
inline-block; cursor: pointer;
text-align: center; }
```

```
input[type="submit"]:hover { background-
color:
#329dd5;
-webkit-transition: all 0.2s ease;
transition: all 0.2s ease; }
```

```
#create-account-wrap {
background-color:
#eedf1; color: #8a8b8e;
font-size: 14px; width:
100%; padding: 10px 0;
border-radius: 0 0 4px 4px;
}
</style>
```

```
</head>
```

```
<body>
```

```
<div id="login-form-wrap">
```

```
<h2>Sign in</h2>
```

```
<form id="login-form" action="{ { url_for('register') } }" method="post">
```

```
<p>
```

```
<input type="text" id="name" name="uname" placeholder="Username" required><i
class="validation"><span></span><span></span></i>
```

```
</p>
```

```
<p>
```

```
<input type="email" id="email" name="email" placeholder="Email Address" required><i
class="validation"><span></span><span></span></i>
```

```
</p>
```

```
<p>
```

```
<input type="text" id="num" name="phone" placeholder="Phone number" required><i
class="validation"><span></span><span></span></i>
```

```

</p>
<p>
  <input type="text" id="pass" name="password" placeholder="Password" required><i
    class="validation"><span></span><span></span></i>
</p>

<p>
  <input type="submit" id="login" value="signin">
</p>
</form>
<div id="create-account-wrap">
  <p>Already a member? <a href="{{ url_for('login') }}">Login now</a> <p>
</div>

</div>

</body>

</html>

```

Login.html

```

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <title>VSYB BANK LOGIN</title>

  <style>
    body {
      background-image: url('static/flight.jpg');
      background-color: #3ca9e2; background-
        repeat: no-repeat; background-
        attachment: fixed; background-size:
        100% 100%;
      font-family: 'Poppins', sans-serif;
    }

```

```
h2 {  
    font-weight: 300;  
    text-align: center;  
}
```

```
p {  
    position: relative;  
}
```

```
a, a:link,  
a:visited,  
a:active  
{  
    color: #3ca9e2;    -webkit-  
transition: all 0.2s ease;  
transition: all 0.2s ease; }
```

```
a:focus,  
a:hover,  
a:link:focus,  
a:link:hover,  
a:visited:focus,  
a:visited:hover,  
a:active:focus,  
a:active:hover  
{ color: #329dd5;  
  -webkit-transition: all 0.2s ease;  
  transition: all 0.2s ease;  
}
```

```
#login-form-wrap {  
    background-color:  
    #fff; width: 35%;  
    margin-top: 170px;  
    margin-left: 470px;  
    text-align: center;
```

```
padding: 20px 0 0 0; border-radius: 4px; box-  
shadow: 0px 30px 50px 0px rgba(0, 0, 0, 0.2);  
opacity: 0.8;  
}
```

```
#login-form  
{ padding: 0  
60px;  
}
```

```
input {  
display: block; box-  
sizing: border-box;  
width: 100%; outline:  
none; height: 60px; line-  
height: 60px; border-  
radius: 4px; }
```

```
input[type="text"],  
input[type="email"]  
{ width: 100%; padding:  
0 0 0 10px; margin: 0;  
color: #8a8b8e; border:  
1px solid #c2c0ca; font-  
style: normal; font-size:  
16px; -webkit-  
appearance: none; -moz-  
appearance: none;  
appearance: none;  
position: relative;  
display: inline-block;  
background: none; }
```

```
input[type="text"]:focus,  
input[type="email"]:focus  
{ border-color: #3ca9e2;  
}
```



```
input[type="text"]:focus:invalid,  
input[type="email"]:focus:invalid  
{ color: #cc1e2b; border-  
color: #cc1e2b; }
```

```
input[type="text"]:valid~.validation,  
input[type="email"]:valid~.validation  
{ display: block;  
border-color: #0C0; }
```

```
input[type="text"]:valid~.validation span,  
input[type="email"]:valid~.validation span  
{ background: #0C0;  
position: absolute;  
border-radius: 6px;  
}
```

```
input[type="text"]:valid~.validation span:first-child,  
input[type="email"]:valid~.validation span:first-child  
{ top: 30px;  
left: 14px;  
width: 20px;  
height: 3px;  
-webkit-transform: rotate(-45deg);  
transform: rotate(-45deg);  
}
```

```
input[type="text"]:valid~.validation span:last-child,  
input[type="email"]:valid~.validation span:last-child  
{ top: 35px;  
left: 8px;  
width: 11px;  
height: 3px;  
-webkit-transform: rotate(45deg);  
transform: rotate(45deg);  
}
```

```
.validation
```

```
{ display: none;
position: absolute;
content: " "; height:
60px; width: 30px;
right: 15px; top:
0px; }
```

```
input[type="submit"]
{border: none; display:
block; background-color:
#3ca9e2; color: #fff; font-
weight: bold; text-
transform: uppercase;
cursor: pointer; -webkit-
transition: all 0.2s ease;
transition: all 0.2s ease;
font-size: 18px; position:
relative; display: inline-
block; cursor: pointer; text-
align: center;
}
```

```
input[type="submit"]:hover {background-
color:
#329dd5;
-webkit-transition: all 0.2s ease;
transition: all 0.2s ease;
}
```

```
#create-account-wrap {
background-color:
#eedf1; color: #8a8b8e;
font-size: 14px; width:
100%; padding: 10px 0;
border-radius: 0 0 4px 4px;
}
```

</style>

</head>

<body>

<div id="login-form-wrap">

<h2>Login</h2>

<form id="login-form" action="{ { url_for('signin') } }" method="post">

<p>

<input type="email" id="mail" name="email" placeholder="E-mail Address" required><i
class="validation"></i>

</p>

<p>

class="validation"></i>

</p>

<p>

<input type="submit" id="login" value="Login">

</p>

</form>

<div id="create-account-wrap">

<p>Not a member? Sign up now <p>

</div>

</div>

</body>

</html>

Login.css

*

{ margin: 0; padding: 0;
box-sizing: border-
box;
}

.full-page
{

```
height: 100%;
width: 100%;

background-position:
center; background-size:
cover; position: absolute;
}
```

```
.navbar
```

```
{ display: flex; align-
items: center;
padding: 20px;
padding-left: 50px;
padding-right:
30px; padding-top:
50px; } nav
```

```
{ flex: 1; text-
align: right; } nav
```

```
ul
```

```
{ display: inline-block;
list-style: none; } nav ul
```

```
li
```

```
{ display: inline-block;
margin-right: 70px; }
```

```
nav ul li a
```

```
{ text-decoration: none;
font-size: 20px; color:
white; font-family: sans-
serif; } nav ul li button
```

```
{ font-size: 20px; color:
white; outline: none;
border: none; background:
transparent; cursor:
pointer; font-family: sans-
serif; } nav ul li
```

```
button:hover
```

```
{ color: aqua; }
```

```
nav ul li
```

```
a:hover
```

```
{ color: aqua; } a { text-
decoration: none; color:
```

```

palevioletred; font-size:
28px;
}
#login-form
{ display: none;
}
.form-box
{ width:380px;
    height:480px;
    position:relative;
    margin:2% auto;
    background:rgba(0,0,0,0.3)
    ; padding:10px;
    overflow: hidden;
}
.button-box
{ width:220px;
    margin:35px
    auto;
    position:relative;
    box-shadow: 0 0 20px 9px #ff61241f; border-radius:
    30px;
}
.toggle-btn
{ padding:10px 30px;
    cursor:pointer;
    background:transpa
    rent;border:0;
    outline: none;
    position: relative;
}
#btn
{ top: 0; left:0; position:
    absolute; width:
    110px; height: 100%;
    background:
    #F3C693; border-

```

```

        radius: 30px;
        transition: .5s;

    }
    .input-group-login
    { top: 150px;
        position: absolute
        ; width: 280px;
        transition: .5s;
    }
    .input-group-register
    { top: 120px;
        position: absolute;
        width: 280px;
        transition: .5s;
    }
    .input-field
    { width: 100%;
        padding: 10px
        0; margin: 5px
        0; border-left: 0;
        border-top: 0;
        border-right: 0;
        border-bottom:
        1px solid #999;
        outline: none;
        background:
        transparent;
    }
    .submit-btn
    { width: 85%; padding: 10px
        30px; cursor: pointer;
        display: block; margin:
        auto; background:
        #F3C693; border: 0;
        outline: none; border-
        radius: 30px;
    }

```

```

.check-box
{ margin: 30px 10px 34px 0;
}
span
{ color:#777; font-
    size:12px;
    bottom:68px;
    position:absolute
    ;
}
#login
{
    left:50px;
}
#login input
{ color:white; font-
    size:15;
}
#register
{ left:450px;
}
#register input
{ color:white; font-
    size: 15;
}
.content{ font-size:
    20px; color:
    white;
}

```

Index.html

```

<!DOCTYPE      html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <title>VSYB BANK</title>

```

```

<link rel="stylesheet" href="../static/css/style.css/login.css">
<style>
  body {
    background-image: url('static/1.png');
    background-repeat: no-repeat; background-
    attachment: fixed; background-size: 100%
    100%;
  }

  h1 {
    color: white;
  }
  .detail{ font-size:
    26px; color: white;
    margin-top: 170px;
    margin-left: 25px;
    font-family:sans-
    serif;
  }
</style>
<!--we had linked our css file----->
</head>

<body>
  <div class="full-page">
    <div class="navbar">

      <p>
        <h1>AI Based Discourse for Banking Industry </h1> </p>

      <nav>
        <ul id='MenuItems'>
          <li><a href=" {{ url_for('login') }}">Home</a></li>
          <li><a href=" {{ url_for('account') }}">Account</a></li>
          <li><a href=" {{ url_for('aboutus') }}">About Us</a></li> <li><a
          href=" {{ url_for('services') }}">Services</a></li>

```


</nav>
</div>

<p class="detail">VSYS BANKING BOT
 TEAM ID: PNT2022TMID29862

TEAM MEMBERS:
 VINISHA R

<p class="detail">VSYS BANKING BOT
 TEAM ID: PNT2022TMID29862

TEAM MEMBERS:
 VINISHA R

SANDHIYA P
 YOGAMALYA P
 SHAHEERABANU C
 </p>

</div> <script> window.watsonAssistantChatOptions = { integrationID: "1206a417-c053-4bee-a810-037a99f37b2a", // The ID of this integration. region: "us-south", // The region
your integration is hosted in.

serviceInstanceID: "e81248c6-9605-415f-843b-666356f451ec", // 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>

Main.css

```
body {  
  background-image: url('static/flight.jpg');  
  background-repeat: no-repeat;  
  background-attachment: fixed;  
  background-size: 100% 100%;  
  font-family: 'Poppins', sans-serif;  
}
```

```
h2 {  
  font-weight: 300; text-align: center;  
}
```

```
p {  
  position: relative;  
}
```

```
a, a:link,  
a:visited,  
a:active  
{  
  color: #3ca9e2;  
  -webkit-transition: all 0.2s ease;  
  transition: all 0.2s ease;  
}
```

```
a:focus, a:hover,  
a:link:focus,  
a:link:hover,  
a:visited:focus,  
a:visited:hover,  
a:active:focus,  
a:active:hover {  
  color: #329dd5; -webkit-transition: all 0.2s ease;  
  transition: all 0.2s ease;  
}
```

```
#login-form-wrap {  
  background-color: #fff; width: 35%;  
  margin-top: 170px; margin-left: 70px;  
  text-align: center;  
  padding: 20px 0 0 0; border-radius: 4px; box-shadow: 0px 30px 50px 0px rgba(0, 0, 0, 0.2);  
  opacity: 0.8;
```

```
}
```

```
#login-form  
  { padding: 0  
    60px;  
  }
```

```
input {  
  display: block; box-  
  sizing: border-box;  
  width: 100%; outline:  
  none;  
  height: 60px; line-  
height: 60px;  
border-radius: 4px;  
}
```

```
input[type="text"],  
input[type="email"]  
{ width: 100%; padding: 0 0  
0 10px; margin: 0; color:  
#8a8b8e; border: 1px solid  
#c2c0ca; font-style:  
normal; font-size: 16px; -  
webkit-appearance: none; -  
moz-appearance: none;  
appearance: none; position:  
relative; display: inline-  
block; background: none; }
```

```
input[type="text"]:focus,  
input[type="email"]:focus  
{ border-color: #3ca9e2;  
}
```

```
input[type="text"]:focus:invalid,  
input[type="email"]:focus:invalid  
{ color: #cc1e2b; border-  
color: #cc1e2b; }
```

```
input[type="text"]:valid~.validation,  
input[type="email"]:valid~.validation  
{display: block; border-  
color: #0C0;  
}
```

```
input[type="text"]:valid~.validation span,  
input[type="email"]:valid~.validation span  
{background: #0C0;  
position: absolute;  
border-radius: 6px;  
}
```

```
input[type="text"]:valid~.validation span:first-child,  
input[type="email"]:valid~.validation span:first-child  
{top: 30px;  
left: 14px;  
width: 20px;  
height: 3px;  
-webkit-transform: rotate(-45deg);  
transform: rotate(-45deg);  
}
```

```
input[type="text"]:valid~.validation span:last-child,  
input[type="email"]:valid~.validation span:last-child  
{top: 35px;  
left: 8px;  
width: 11px;  
height: 3px;  
-webkit-transform: rotate(45deg);  
transform: rotate(45deg);  
}
```

```
.validation  
{ display: none;  
position: absolute;  
content: " ";  
height: 60px;  
width: 30px;
```

```

    right: 15px; top:
    0px;
}
input[type="submit"] {border:
none;      display:      block;
background-color:      #3ca9e2;
color: #fff; font-weight: bold;
text-transform:      uppercase;
cursor:  pointer;  -webkit-
transition:  all  0.2s  ease;
transition: all 0.2s ease; font-
size: 18px; position: relative;
display: inline-block; cursor:
pointer; text-align: center; }

input[type="submit"]:hover {background-
color:
#329dd5;
-webkit-transition: all 0.2s ease;
transition: all 0.2s ease; }

#create-account-wrap {
background-color:
#eedf1;color: #8a8b8e;
font-size: 14px; width:
100%; padding: 10px 0;
border-radius: 0 0 4px
4px;
}

```

Account.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>

```

```
<link rel="stylesheet" href="../static/css/style.css/login.css">
```

```
<style>
```

```
  body {
    background-image:
      url('static/1.png');    background-
    repeat:    no-repeat;    background-
    attachment: fixed; background-size:
      100% 100%;
  }
  h1{
    color:  white; font-
    size: 40px;
  }
  .cont{ color:
    white; font-
    size:20px;
  }
  .home-img
  img{ float:
    right;
    max-width: 100%;
    width: 300px;
    height: auto;
    margin-right:
    120px;

  }
  .deposit-img
  img{ float: left; max-
    width: 100%; width:
    300px; height: auto;
    margin-right: 120px;
    margin-left: 20px;
  } cmax-width: 100%;
  width: 300px;
  height: auto;
  margin-right:
  120px; margin-top:
  100px;
```

```
}  
.head{    font-  
size:      30px;  
margin-top: 10px;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div class="navbar">
```

```
<h1 class="head">AI Based Discourse for Banking Industry </h1>
```

```
<nav>
```

```
<ul id='MenuItems'>
```

```
<li><a href=" {{ url_for('login') }} ">Home</a></li>
```

```
<li><a href=" {{ url_for('account') }} ">Account</a></li>
```

```
<li><a href=" {{ url_for('aboutus') }} ">About Us</a></li>
```

```
<li><a href=" {{ url_for('services') }} ">Services</a></li>
```

```
</ul>
```

```
</nav>
```

```
</div>
```

```
<br><br><br>
```

```
<div class="home-img">
```

```

```

```
<b> <h1>DEPOSIT </h1> </b><br>
```

```
<p class="cont">Bank deposits are a savings product that customers can use to hold an amount  
of money at a bank for a specified length of time. In return, the financial institution will pay the  
customer the relevant amount of interest, based on how much they choose to deposit and for how  
long.</p><br>
```

```
<div class="deposit-img">
```

```

```

```
<b><h1>LOANS</h1></b><br>
```

<p class="cont">A loan is a sum of money that an individual or company borrows from a lender. It can be classified into three main categories, namely, unsecured and secured, conventional, and open-end and closed-end loans.</p>

<div class="loan-img">

<h1>INSURANCE</h1>

<p class="cont">Insurance is a legal agreement between two parties - the insurer and the insured, also known as insurance coverage or insurance policy. The insurer provides financial coverage for the losses of the insured that s/he may bear under certain circumstances</p>

<script> window.watsonAssistantChatOptions = { integrationID: "1206a417-c053-4bee-a810-037a99f37b2a", // The ID of this integration. region: "us-south", // The region your integration is hosted in.

 serviceInstanceID: "e81248c6-9605-415f-843b-666356f451ec", // 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>

Aboutus.html

<!DOCTYPE html>

<html lang="en">

<head>

 <meta charset="UTF-8">

 <meta http-equiv="X-UA-Compatible" content="IE=edge">

 <meta name="viewport" content="width=device-width, initial-scale=1.0">

 <title>Document</title>

 <link rel="stylesheet" href="../static/css/style.css/login.css">

 <style>

 body {


```

        background-image:
        url('static/1.png');        background-
        repeat:    no-repeat;    background-
        attachment: fixed; background-size:
        100% 100%;

    }
p{
    color:    white;
    font-size:30px;
    text-align:
    center;
    margin:150px;
    font-weight: 900;
}
.head{ font-size:
    30px; margin-top:
    10px; color:
    white;
    }
</style>
</head>
<body>

<div class="navbar">

    <h1 class="head">AI Based Discourse for Banking Industry </h1>

    <nav>
        <ul id='MenuItems'>
            <li><a href=" {{ url_for('login') }}">Home</a></li>
            <li><a href=" {{ url_for('account') }}">Account</a></li>
            <li><a href=" {{ url_for('aboutus') }}">About Us</a></li>
            <li><a href=" {{ url_for('services') }}">Services</a></li>

        </ul>
    </nav>

```

```

</div>
<br><br><br>
<p>
    Online banking allows a user to conduct financial transactions via the Internet. Online banking
    is also known as Internet banking or web banking. Online banking offers customers almost every
    service traditionally available through a local branch including deposits, transfers, and online bill
    payments. It's quick, usually free and allows you to do tasks, such as paying bills and
    transferring money, without having to visit or call your bank<p>
</body>
<script> window.watsonAssistantChatOptions = { integrationID: "1206a417-c053-4bee-
    a810-037a99f37b2a", // The ID of this integration. region: "us-south", // The region your
    integration is hosted in.
        serviceInstanceID: "e81248c6-9605-415f-843b-666356f451ec", // 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>

</html>

```

Service.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="../static/css/style.css/login.css">
    <style>
        body {
            background-image:
                url('static/1.png');      background-
            repeat: no-repeat;      background-

```

```

        attachment: fixed; background-size:
        100% 100%;
    }
    h1{
        font-size:30px;
        color: white;
    }
    .list{
        color: white; font-
        size:30px; font-
        weight:bolder;
    }
    .place{

        margin-left: 55px; margin-
        top: 70px;
    }
    .head{ font-size:
        30px; margin-top:
        10px; color:
        white;
        }
</style>
</head>
<body>
    <div class="navbar">

        <h1 class="head">AI Based Discourse for Banking Industry </h1>

    <nav>
        <ul id='MenuItems'>
            <li><a href=" {{ url_for('login') }}">Home</a></li>
            <li><a href=" {{ url_for('account') }}">Account</a></li>
            <li><a href=" {{ url_for('aboutus') }}">About Us</a></li>
            <li><a href=" {{ url_for('services') }}">Services</a></li>

        </ul>

```

```
</nav>
</div>
<br><br><br>
<div class="place">
  <h1>SERVICES</h1>
  <ul class="list">
    <li>Mobile & Online Banking Features.</li>
    <li>Mobile Check Deposit.</li>
    <li>Send & Receive Money</li>
    <li>Transfer Between Accounts.</li>
    <li>Bill Pay.</li>
  </ul>
</div>

<script> window.watsonAssistantChatOptions = { integrationID: "1206a417-c053-4bee-
a810-037a99f37b2a", // The ID of this integration. region: "us-south", // The region your
integration is hosted in.
  serviceInstanceID: "e81248c6-9605-415f-843b-666356f451ec", // 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>
```

