# AI BASED DISCOURSE FOR BANKING INDUSTRY

**INTRODUCTION:**

**Overview:**

* Industries are forced to evolve and update their practices due to technological advances and the contemporary market. The banking sector is one of the most developed sectors and is always looking for the latest technological solutions that improve its efficiency.
* Net banking websites are complex and involve navigating through a lot of pages to find the information you need. Bank staff undergoes a lot of stressful situations when communicating with clients directly. Such situations can be avoided gracefully by using chatbots.
* Only 32% of companies in the finance industry currently use AI chatbots, and 37% are planning to start using them within 18 months said a report from Salesforce. This results in a potential growth rate of 118% which indicates the demand in the industry.
* A smart chatbot takes a query from the user in natural language and gives the appropriate response for the same. This paper aims to discuss the relevance of chatbots in the banking sector and explore how chatbots can be implemented using natural language processing techniques that can be used in the banking industry.

**LITERATURE SURVEY:**

**Existing Problem:**

* This paper [1] presents the use of the RASA framework for building smart context-remembering chatbots, it also describes how Rasa NLU works and how its performance is elevated by using intent recognition and entity extraction. It also compares the accuracies of entity extraction using Rasa NLU and a NN, results show Rasa NLU performs better to extract entities when whole sentences are provided as compared to neural networks which require segmented inputs. This paper discusses Rasa by implementing a chatbot related to the finance domain, using which the users can inquire about stock-related information.
* RASA NLU can introduce a vital component in intelligent chatbot systems. We can compose the system to extract the entity after intent recognition. This can be further improved for complicated sentences and more entities.
* This paper [2] briefly discusses advancements in the field of AI and how this has led to major shifts in some organizations about how they operate. It further mentions how the banking industry has moved to use chatbots for providing an interface to customers so that they can have an assistant throughout the day for service. This paper also gauges the ability of current chatbots to provide all the services that a user needs.
* It includes several strategies for managing dialogue in the banking and finance industry based on ontology. Although further use of AI can make the chatbot not only respond to questions but also self-learning to improve itself in more stages, improving user service quality and also reducing human load.

**Proposed solution:**

* The solution to the problem is Artificial intelligence in the banking sector makes banks efficient, trustworthy, helpful, and more understanding. It is strengthening the competitive edge of modern banks in this digital era. The growing impact of AI in banking sector minimizes operational costs improves customer support and process automation.
* Nearly 40% to 50% of financial and banking service providers are using AI in their processes to harness the power of next-generation AI capabilities. The companies believe that AI is the future of banking sector which can perform a range of banking operations in faster, easier, and more secure ways.
* AI banking Chatbots help customers in many ways. AI-based chatbot service for financial industry is one of the significant use cases of AI in banking sector. AI chatbots in banking are modernizing the way how businesses provide services to their customers
* AI chatbots in the banking industry can assist customers 24\*7 and give accurate responses to their queries. These chatbots provide a personalized experience to users.
* AI chatbots in banking is providing a better customer experience.
* Hence, AI chatbots for banking and finance operations let banks attract customer attention, optimize service quality, and expand the brand mark in the market.

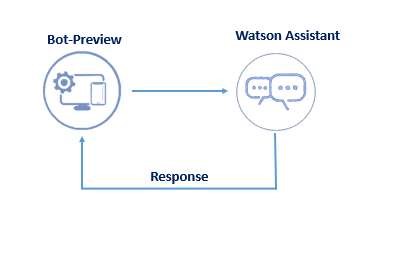
**THEORETICAL ANALYSIS:**

**Services Used:**

• IBM Watson Assistant



**Block diagram:**



**Hardware / Software designing:**

To complete this project, you should have the following software and packages.

**Softwares:**

* Visual studio code
* IBM Watson studio

**Packages:**

* Flask

**FLOWCHART:**

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

* Create IBM Services.
* Creating skills & Assistant for Chatbot.
* Creating Savings account action.
* Creating Current account action.
* Creating Loan account action.
* Creating a general query action.
* Creating a Net banking action.
* Create HTML web page.
* Integrate the Watson Chatbot with web page.

**ADVANTAGES & DISADVANTAGES:**

**Advantages:**

* Round-the-clock service.
* Brand Consistency.
* Increased Productivity.
* Reduced Staffing Needs.
* Consistent Response Rate and Availability.
* Helps with Fraud Prevention.
* Chats can be saved.
* Lower costs.

**Disadvantages:**

* Questions must be programmed beforehand.
* Impersonal
* Must keep information up-to-date.
* Technology issues.
* Needs additional measures to protect identities.

**APPLICATIONS:**

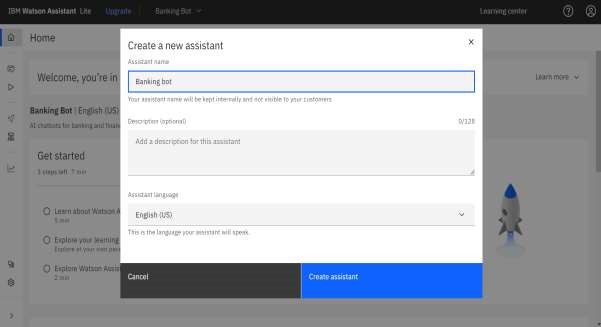
* Banking chatbots have all the data to predict the spending habits of customers and help them keep their finances on track.

**APPENDIX:**

## Create IBM Service

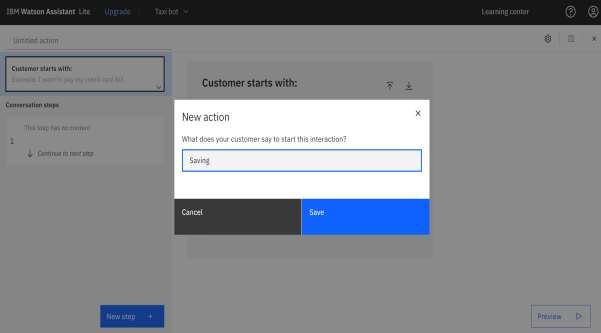
In this activity, you will be creating the Necessary IBM service. The following are the service that you have to create.

• Watson Assistant

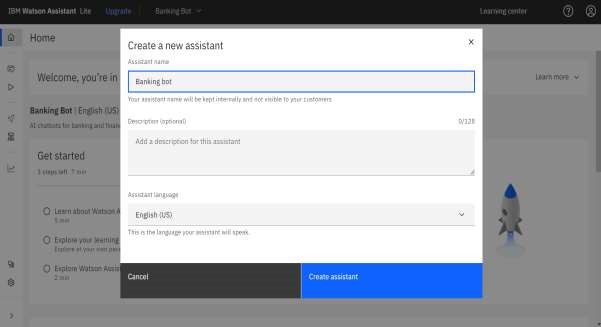


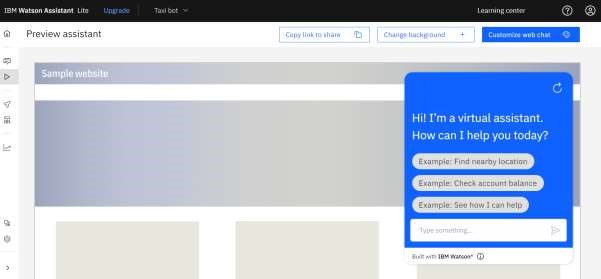
## Creating Skills & Assistant For Chatbot

Skills are nothing but actions and steps. Steps are the subset of actions where conversations are built and Assistant is used to integrate skills.



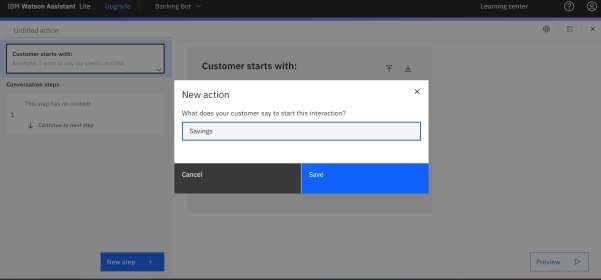
A default template chatbot is created. Need to add actions.



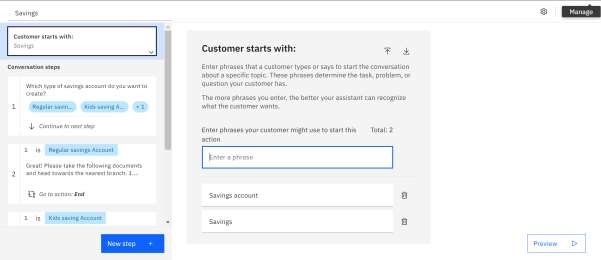


## Creating Saving Account Action

Create a saving account in IBM Watson. Create new **Action** Saving.

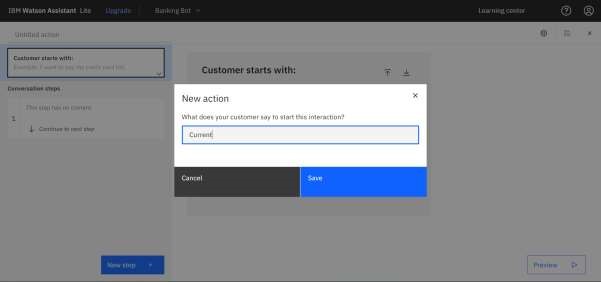


Add steps in savings action.

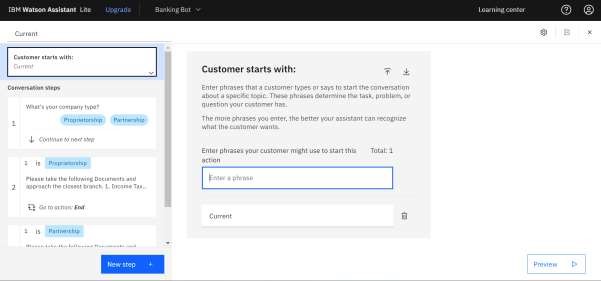


## Creating Current Account Action

Create a new **Action** Current for the current account action.

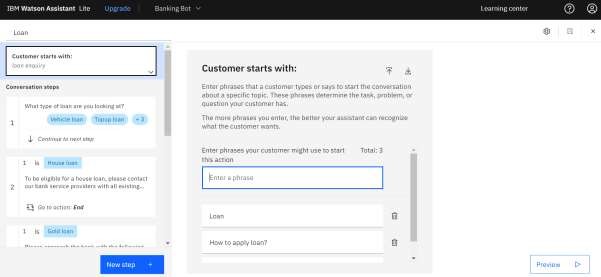


Add steps in current action.



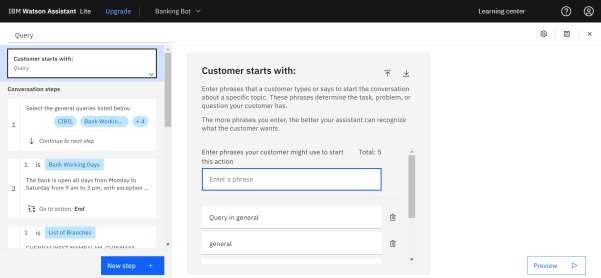
## Creating Loan Account Action

Loan action is created with the necessary steps.



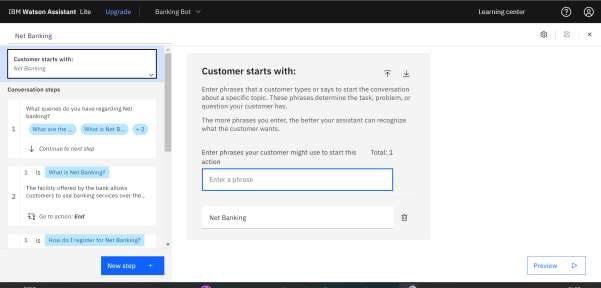
## Creating General Query Action

General query action is created with the necessary steps.

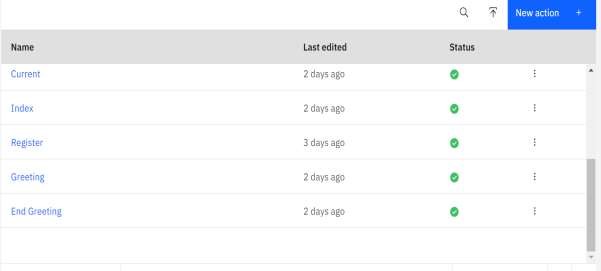


## Creating Net Banking Action

Net banking action is created with the necessary steps.



In addition to this greeting, end greeting ,index and end actions are also created.



## Creating Assistant & Integrate With Flask Web Page

You will be creating a banking bot in this activity that has the following capabilities

1. The Bot should be able to guide a customer to create a bank account.
2. The Bot should be able to answer loan queries.
3. The Bot should be able to answer general banking queries.
4. The Bot should be able to answer queries regarding net banking.
5. With the help of this bot, you can get all the required details related to banking.

Let us build our flask application which will be running in our local browser with a user interface.

In the flask application, users will interact with the chatbot, and based on the user queries they will get the outcomes.

### Build Python Code

#### 1: Importing Libraries

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



Importing the flask module into the project is mandatory. An object of the Flask class is our WSGI application. Flask constructor takes the name of the current module (\_\_name\_\_).

#### 2: Creating our flask application and loading



#### 3: Routing to the Html Page

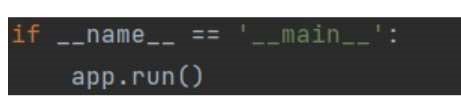
Here, the declared constructor is used to route to the HTML page created earlier.

The ‘/’ route is bound with the bot function. Hence, when the home page of a web server is opened in the browser, the HTML page will be rendered.



#### Main Function

This is used to run the application in localhost.



### 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 app.py” command.
* It will show the local host where your app is running on http://127.0.0.1.5000/
* Copy that localhost URL and open that URL in the browser. It does navigate me to where you can view your web page.

**Source code:**

**<html>**

**<head>**

**<script>**

**window.watsonAssistantChatOptions = {**

**integrationID: "5308436f-fc25-4c3f-8abb-305916aba969", // The ID of this integration.**

**region: "us-south", // The region your integration is hosted in.**

**serviceInstanceID: "0e12d8d0-8932-4290-89ec-89d8761d091d", // 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>**

**<base href="/">**

**<title>Open Financial Cognitive Conversation</title>**

**<meta charset="utf-8">**

**<meta name="viewport" content="width=device-width, initial-scale=1">**

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

**<meta property="og:image" content="conversation.svg" />**

**<meta property="og:title" content="Conversation Chat Simple" />**

**<meta property="og:description"**

**content="Sample application that shows how to use the Conversation API to identify user intents" />**

**<link rel="shortcut icon" href="favicon.ico" type="image/x-icon">**

**<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"**

**integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">**

**<link rel="stylesheet" href="css/app.css">**

**<style type="text/css">**

**body {**

**padding-top: 3.5rem;**

**}**

**</style>**

**</head>**

**<body>**

**<nav class="navbar navbar-expand-md navbar-dark fixed-top bg-dark">**

**<a class="navbar-brand" href="#">Open Financial Bank</a>**

**<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarsExampleDefault"**

**aria-controls="navbarsExampleDefault" aria-expanded="false" aria-label="Toggle navigation">**

**<span class="navbar-toggler-icon"></span>**

**</button>**

**<div class="collapse navbar-collapse" id="navbarsExampleDefault">**

**<ul class="navbar-nav mr-auto">**

**<li class="nav-item active">**

**<a class="nav-link" href="#">Home <span class="sr-only">(current)</span></a>**

**</li>**

**<li class="nav-item">**

**<a class="nav-link" href="#">About Us</a>**

**</li>**

**<li class="nav-item dropdown">**

**<a class="nav-link dropdown-toggle" href="#" id="dropdown01" data-toggle="dropdown"**

**aria-haspopup="true" aria-expanded="false">Services</a>**

**<div class="dropdown-menu" aria-labelledby="dropdown01">**

**<a class="dropdown-item" href="#">Banking</a>**

**<a class="dropdown-item" href="#">Investment</a>**

**<a class="dropdown-item" href="#">Home Loan</a>**

**</div>**

**</li>**

**</ul>**

**<form class="form-inline my-2 my-lg-0">**

**<input class="form-control mr-sm-2" type="text" placeholder="Search" aria-label="Search">**

**<button class="btn btn-outline-success my-2 my-sm-0" type="submit">Search</button>**

**</form>**

**<ul class="navbar-nav navbar-right">**

**<li class="nav-item">**

**<a class="nav-link" href="#">Login</a>**

**</li>**

**<li class="nav-item">**

**<a class="nav-link" href="#">Register</a>**

**</li>**

**</ul>**

**</div>**

**</nav>**

**<main role="main">**

**<!-- Main jumbotron for a primary marketing message or call to action -->**

**<div class="jumbotron">**

**<div class="container">**

**<h1 class="display-3">Welcome to Open Financial Bank</h1>**

**<p>We provide extensive services from retail banking, investment, home loan, auto loan and many more</p>**

**<p><a class="btn btn-primary btn-lg" href="#" role="button">Learn more &raquo;</a></p>**

**</div>**

**</div>**

**<div class="container">**

**<!-- Example row of columns -->**

**<div class="row">**

**<div class="col-md-4">**

**<h2>Banking</h2>**

**<p>Open Financial Bank provide retail and online banking. It offers all kinds of accounts from**

**checking, saving,**

**and bussiness accounts**

**</p>**

**<p><a class="btn btn-secondary" href="#" role="button">View details &raquo;</a></p>**

**</div>**

**<div class="col-md-4">**

**<h2>Investment</h2>**

**<p>Open Financial Bank also provides investment opportunity to clients. It has huge basket of**

**various stocks and bonds. </p>**

**<p><a class="btn btn-secondary" href="#" role="button">View details &raquo;</a></p>**

**</div>**

**<div class="col-md-4">**

**<h2>Loan</h2>**

**<p>Open Financial Bank also helps you find loan for personal, home and auto. We provide best rate**

**and beat the rate thats out in the market.</p>**

**<p><a class="btn btn-secondary" href="loan.html" role="button">View details &raquo;</a></p>**

**</div>**

**</div>**

**<hr>**

**</div> <!-- /container -->**

**</main>**

**<footer class="container">**

**<p>&copy; Open Financial Bank 2019</p>**

**</footer>**

**<div id="contentParent" class="responsive-columns-wrapper">**

**<div id="chat-column-holder" class="responsive-column content-column">**

**<div class="chat-column">**

**<div id="scrollingChat">**

**<h4>Welcome to Open Financial Bank.</h4>**

**</div>**

**<label for="textInput" class="inputOutline">**

**<input id="textInput" class="input responsive-column" placeholder="Type something" type="text"**

**onkeydown="/\*globals CanvasJS \*/**

**ConversationPanel.inputKeyDown(event, this)">**

**</label>**

**</div>**

**</div>**

**</div>**

**<!-- Bootstrap core JavaScript**

**================================================== -->**

**<!-- Placed at the end of the document so the pages load faster -->**

**<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"**

**integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"**

**crossorigin="anonymous"></script>**

**<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"**

**integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"**

**crossorigin="anonymous"></script>**

**<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"**

**integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"**

**crossorigin="anonymous"></script>**

**<script src="js/modal.js"></script>**

**<script src="js/api.js"></script>**

**<script src="js/common.js"></script>**

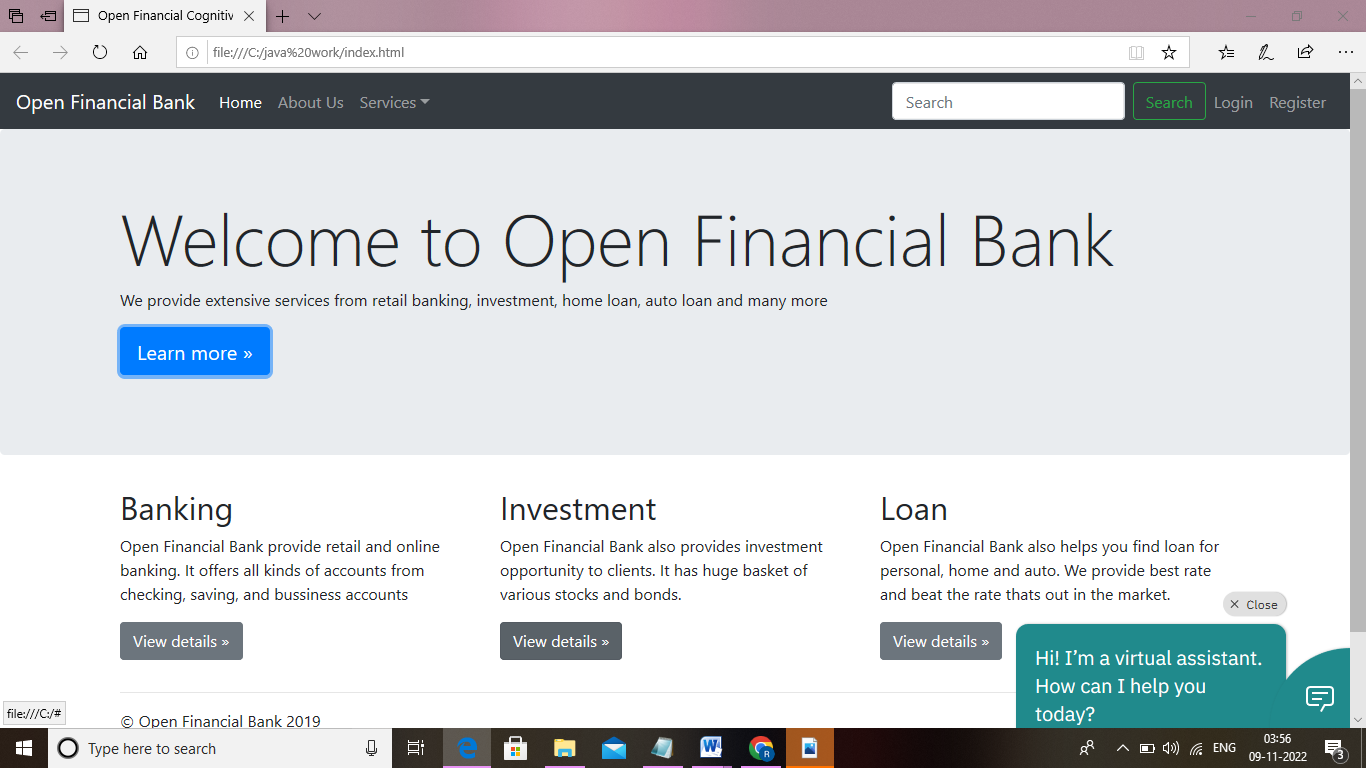
**<script src="js/conversation.js"></script>**

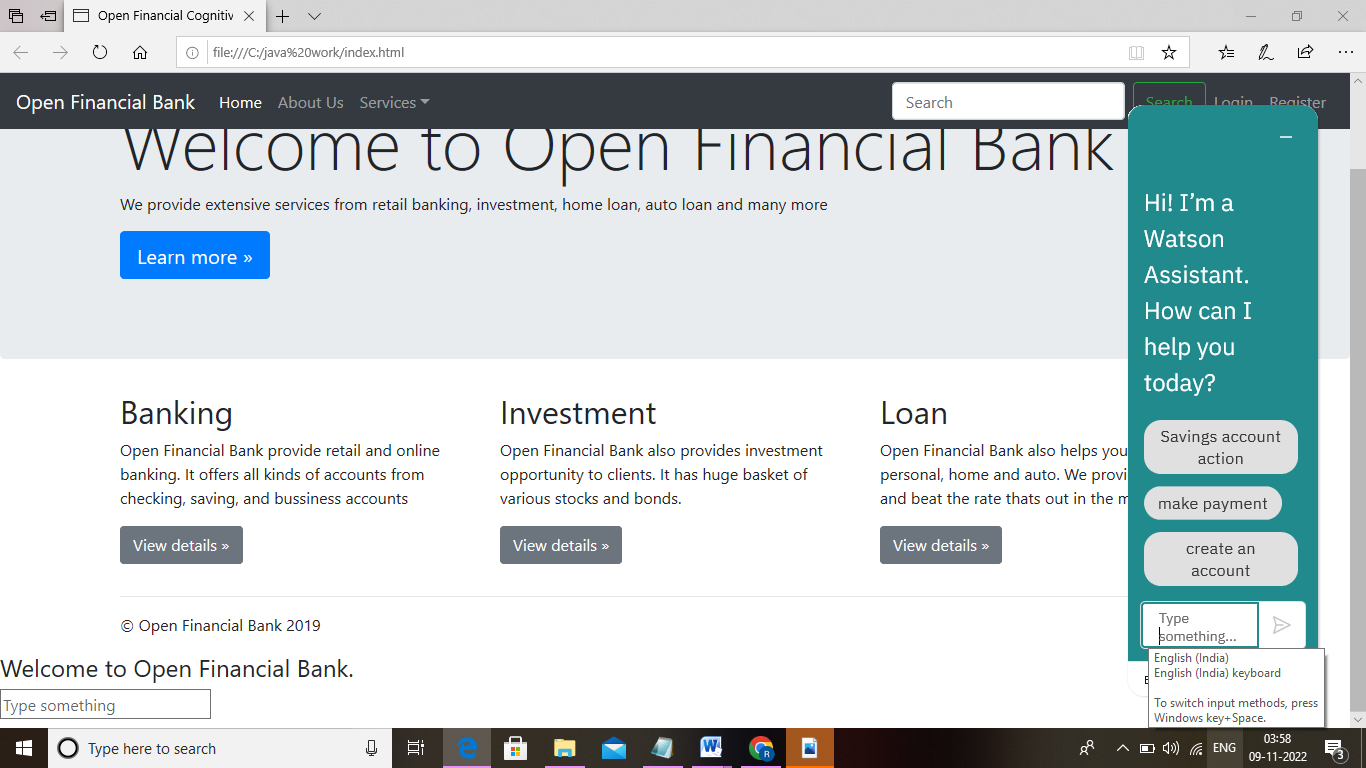
**<script src="js/global.js"></script>**

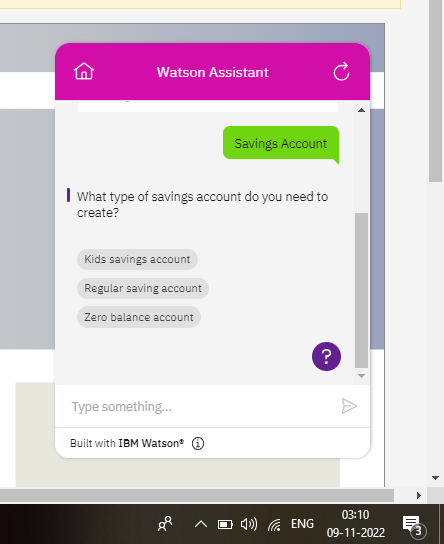
**</body>**

**</html>**

**Output:**







**Banking chatbot**

**Project demo link:** [**file:///C:/Users/ELCOT/Desktop/ChatBot.html**](file:///C:/Users/ELCOT/Desktop/ChatBot.html)

**Preview link :**

[**https://web-chat.global.assistant.watson.appdomain.cloud/preview.html?backgroundImageURL=https%3A%2F%2Fus-south.assistant.watson.cloud.ibm.com%2Fpublic%2Fimages%2Fupx-0e12d8d0-8932-4290-89ec-89d8761d091d%3A%3A48b3dfff-aaa6-4519-926f-a46b028e9729&integrationID=5308436f-fc25-4c3f-8abb-305916aba969&region=us-south&serviceInstanceID=0e12d8d0-8932-4290-89ec-89d8761d091d**](https://web-chat.global.assistant.watson.appdomain.cloud/preview.html?backgroundImageURL=https%3A%2F%2Fus-south.assistant.watson.cloud.ibm.com%2Fpublic%2Fimages%2Fupx-0e12d8d0-8932-4290-89ec-89d8761d091d%3A%3A48b3dfff-aaa6-4519-926f-a46b028e9729&integrationID=5308436f-fc25-4c3f-8abb-305916aba969&region=us-south&serviceInstanceID=0e12d8d0-8932-4290-89ec-89d8761d091d)

**REFERENCE:**

**[1] Jiao, Anran. (2020). An Intelligent Chatbot System Based on Entity**

**Extraction Using RASA NLU and Neural Network. Journal of Physics: Conference**

**Series. 1487. 012014. 10.1088/1742-6596/1487/1/012014.**

**[2] Fathima, Sasha &amp; Student, Suhel &amp; Shukla, Vinod &amp; Vyas, Dr Sonali &amp;**

**Mishra, Ved P. (2020). Conversation to Automation in Banking Through Chatbot**

**Using Artificial Machine Intelligence Language.**

**10.1109/ICRITO48877.2020.9197825.**

**[3] Singh, Netra &amp; Singh, Devender. (2019). Chatbots and Virtual Assistant in**

**Indian Banks. Industrija. 47. 75-101. 10.5937/industrija47-24578.**

**[4] Petr Lorenc, “Joint model for intent and entity recognition” in**

**arXiv:2109.03221v1 [cs.CL] 7 Sep 2021**

**[5] The Rasa documentation. [Online]. Available:**

**https://rasa.com/docs/rasa/2.x [6] Django documentation. [Online].**

**Available:**

**https://docs.djangoproject.com/en/4.0/**