A sample document on **Calculator Chatbot**  
  
  
1) Creating a chatbot needs following steps:  
  
 Define the purpose and goals of your chatbot-

* The purpose of this calculator chatbot is to provide users with a conversational interface to perform various calculations based on their specific needs. The primary function of this chatbot is to ask the user which type of calculator they want to use, either a currency calculator, a student percentage calculator, or a CGPA to percentage calculator, and provide them with accurate results based on their input.
* Greeting the user
* Ask which kind of calculator user wants.
* Convert currency between different currencies.
* Provide real-time exchange rates.
* Ask for Cgpa of student.
* Give the converted percentage.

2) Choose a opensource platform or framework:

* Rasa: Rasa is a popular open source chatbot development platform that provides tools for building advanced conversational AI assistants.
* It offers a flexible and customizable architecture that allows developers to build chatbots with more complex functionalities such as natural language understanding and context handling.

3) Design the conversation flow:

Here is a sample conversation flow for a calculator chatbot that can handle currency conversion and student CGPA to percentage conversion:

* Greeting the user:

Chatbot: Hello! How can I assist you today?

Ask which kind of calculator the user wants:

* Chatbot: Which calculator do you need? Currency calculator or CGPA calculator?  
  If the user chooses currency calculator:
* Chatbot: Great! Which currencies would you like to convert?  
  User: I want to convert USD to EUR.
* Chatbot: Sure thing! The current exchange rate is X. How much USD do you want to convert?
* User: I want to convert 100 USD.
* Chatbot: Okay, that would be Y EUR. Is there anything else I can help you with?
* If the user chooses CGPA calculator:
* Chatbot: Alright! Can you please tell me your CGPA?
* User: My CGPA is 3.5.
* Chatbot: Got it! Your percentage is X%. Is there anything else I can help you with?
* If the user wants to end the conversation:
* Chatbot: Alright then. Have a great day!
* The conversation flow can be further enhanced based on the specific requirements and user needs. Additionally, the chatbot can be programmed to handle errors and provide helpful messages to guide the user throughout the conversation.

4) Integrating with Java application:

* Choosing a chatbot development platform that provides integration.
* Create and train your chatbot on the chosen platform to handle the intended conversation flows and user inputs.
* Choose a web application development framework, such as Angular, that provides the ability to embed external components into application.
* Use the appropriate chatbot integration method provided by your chosen platform, such as APIs, or software development kits (SDKs), to embed the chatbot into your web application.
* Test and verify the chatbot integration by deploying the web application on a development environment and running a few sample conversations with the chatbot.
* Make necessary adjustments and improvements to the chatbot and web application based on user feedback and usage metrics.

5)Do we need to separate server for running chatbot:  
 Yes, you will typically need a server to run your chatbot.  
A server provides the necessary computing resources for your chatbot to process user inputs, generate responses, and maintain conversations with users.

* The type of server you need may vary depending on your chatbot's complexity and the platform you use to develop it.

6) how that server would be deployed or kept in line with existing spring boot server:  
 If we have an existing Spring Boot server, there are several ways to deploy your chatbot server and integrate it with your Spring Boot application. Here are some possible options:

* Deploy the chatbot server as a separate service: You can deploy your chatbot server as a separate service on a different port or subdomain and configure our Spring Boot application to communicate with it using REST APIs or web sockets. This approach allows you to keep our chatbot server separate from your Spring Boot application and scale them independently.
* Embed the chatbot server in your Spring Boot application: we can embed your chatbot server in your Spring Boot application using libraries such as **Jetty or Tomcat**. This approach allows you to have a single deployment unit for both your Spring Boot application and chatbot server.