

Plagiarism Scan Report



Characters:5593

Words:824

Sentences:46

Speak Time:
7 Min

Excluded URL

None

Content Checked for Plagiarism

Chapter 3 Methodology 3.1 Technologies used by CHATBANKER When developing ChatBanker, I leveraged a variety of technologies to ensure its functionality and effectiveness. Some of the key technologies utilized include: 3.1.1 NLP(Natural Language Processing) NLP is a branch of artificial intelligence that focuses on the interaction between computers and human language. It enables machines to understand, interpret, and generate human language in a meaningful way. In the context of your chatbot, NLP is used to process and understand user inputs. It helps extract relevant information and intents from the messages sent by users. NLP techniques often involve tokenization, part-of-speech tagging, named entity recognition, and sentiment analysis, among others. These techniques help analyze the structure and meaning of user messages. 3.1.2 Mongo DB MongoDB is a popular NoSQL database that provides a flexible and scalable solution for storing and retrieving data. In your chatbot, MongoDB is used to store and manage various data related to user accounts, such as account balances, transaction history, and other relevant information. You can create a MongoDB database and define collections to store different types of data. Each document in a collection can represent a user account, containing fields like account number, balance, and policies. 3.1.3 TELEGRAM API The Telegram API allows your chatbot to interact with users on the Telegram messaging platform. It provides various methods and functionalities to send and receive messages, process user inputs, and perform other actions within Telegram. To use the Telegram API, you need to create a Telegram bot by registering with the BotFather, which is the official bot for creating and managing bots on Telegram. The BotFather provides you with an API token that you can use to authenticate your bot and make API calls. 3.1.4 Amazon(EC2) Amazon EC2 (Elastic Compute Cloud) is a scalable virtual machine service provided by AWS. It allows you to create and manage virtual servers, known as EC2 instances, in the cloud. These instances serve as the infrastructure to host and run your chatbot's backend server. With EC2, you can choose the specifications for your virtual instances, including the CPU, memory, storage, and operating system. This flexibility enables you to tailor the resources according to the requirements of your chatbot application. To deploy a chatbot on AWS EC2, follow these steps: Create an EC2 instance with the desired specifications. Set up the backend server by installing dependencies and deploying the code. Configure networking and security settings. Connect the chatbot to external services with the necessary credentials. Test the chatbot's functionality and set up monitoring. Scale the EC2 instances as needed and automate deployment for updates. 3.2 Workflow of the BOT User Interaction: Users interact with CHATBANKER through the Telegram messaging platform. They can send

text messages to the chatbot, ask questions, make requests, or initiate actions.

Telegram API Integration: CHATBANKER integrates with the Telegram API to send and receive messages. The Telegram API provides methods for sending messages, receiving updates, and managing various aspects of the chatbot's functionality.

Message Processing: When a user sends a message to CHATBANKER it receives the message through the Telegram API. The backend of CHATBANKER, which is responsible for processing messages, receives the incoming message for further

analysis. **Natural Language Processing (NLP):** The backend utilizes NLP techniques to understand and interpret the user's message. NLP involves various steps, such as tokenization, part-of-speech tagging, named entity recognition, and sentiment analysis.

These techniques help analyze the structure and meaning of the user's message, extract intents (what the user wants) and entities (relevant pieces of information), and determine the appropriate action to take. **Login Process:** If the user initiates the login

process, CHATBANKER prompts for credentials (e.g., username and password). The user provides their credentials, which are then sent to the backend for validation. The backend verifies the provided credentials by comparing them against the stored user

data in MongoDB. **MongoDB Integration:** MongoDB is used to store and manage data related to user accounts, such as account balances, transaction history, and bank policies. The backend interacts with MongoDB to access and manipulate user account

data. When a user logs in, the backend retrieves the relevant account information from MongoDB based on the validated credentials. **Account Information Retrieval:** Upon successful login, the chatbot retrieves the user's account information, such as the

account balance and bank policies, from MongoDB. The chatbot formats and presents this information to the user as a response. **User Actions and Updates:** Users can interact further with the chatbot to perform various actions or inquiries related to their

accounts. For example, they can check the account balance, inquire about specific bank policies, initiate transactions, or update their account information. The chatbot processes these user requests, performs the necessary actions (e.g., updating data in

MongoDB), and generates appropriate responses. **Continued Conversation:** The chatbot maintains the conversation context, allowing users to have back-and-forth interactions. It keeps track of the state and context of the conversation to provide

accurate and relevant responses based on the user's previous messages.

Sources



[Home](#)

[Blog](#)

[Testimonials](#)

[About Us](#)

[Privacy Policy](#)

Copyright © 2022 [Plagiarism Detector](#). All right reserved