CHATBOT FOR

DATA STRUCTURES

MONISHA B T - 21Z320

SHANMITHA P - 21Z346

SHRINIDHI DINESH - 21Z353

SUPRIYA K - 21Z360

VEDAVARSHINI A - 21Z368

VASANTH C - 22Z462VASANTH C - 22Z462

***PROBLEM STATEMENT -***

Data Structures chatbot for a mobile application that can efficiently manage and retrieve user data from a database while engaging in natural language conversations with users. The chatbot should be able to handle multiple user queries simultaneously and provide accurate and timely responses. It should also be secure, user-friendly, and easily customizable to accommodate future updates and modifications.

***ABSTRACT*** -

Data Structures chatbot for a mobile application that can efficiently manage and retrieve user data from a database while engaging in natural language conversations with users. The chatbot should be able to handle multiple user queries simultaneously and provide accurate and timely responses. It should also be secure, user-friendly, and easily customizable to accommodate future updates and modifications.

***MOTIVATION*** -

* The motivation behind developing a DS chatbot for a mobile application is to enhance user experience by providing a convenient and efficient way to interact with the application.
* With the growing popularity of mobile applications, it has become essential to provide users with an easy-to-use interface that can handle user queries effectively.
* Chatbots have proven to be a useful tool in achieving this goal by providing a conversational interface that can simulate human-like interactions.

***EXISTING WORKS AND PROBLEMS FACED*** -

● There are no existing Data Structure based chatbot application in google play store and apple app store.

***SRS*** *(software requirements specification) -*

1.Introduction: The DBMS chatbot for mobile application aims to provide a conversational interface that can manage and retrieve user data from a database. The chatbot will be designed to handle natural language queries and provide timely and accurate responses. The chatbot will be integrated into an Android 9 (or higher) mobile application and an iOS 12 (or higher) mobile application to provide users with a convenient and user-friendly interface for interacting with the application.

2.Functional Requirements:

a. User Management: The chatbot should be able to manage user data, including user login and authentication, user registration, and user data retrieval and storage.

b. Natural Language Processing: The chatbot should be able to handle natural language queries and provide relevant responses based on the user's input.

c. Database Management: The chatbot should be able to retrieve and store data from a database, including user data, application data, and other relevant data.

d. Security: The chatbot should be designed with security in mind, ensuring that user data is protected against unauthorized access or security breaches.

e. Scalability: The chatbot should be able to handle multiple user queries simultaneously and provide timely and accurate responses.

3. Non-Functional Requirements:

a. Usability: The chatbot should provide a user-friendly interface that enhances the user experience.

b. Performance: The chatbot should be designed to provide quick and accurate responses to user queries.

c. Reliability: The chatbot should be reliable and able to handle user queries without downtime or system crashes.

d. Maintainability: The chatbot should be easy to maintain and update, ensuring that it remains up-to-date and free from bugs or errors.

e. Portability: The chatbot should be easily portable to different mobile platforms and devices.

4. Constraints:

a. The chatbot should be designed to work within the limitations of Android 9 (or higher) and iOS 12 (or higher) mobile devices, including limited processing power and storage capacity.

b. The chatbot should be designed to work with the mobile application's existing infrastructure and architecture.

c. The chatbot should use the Google Text-to-Speech (TTS) API through a Python API to provide audio output.

Assumptions:

a. The chatbot will be designed to work with a specific database management system and mobile application.

b. The chatbot will be designed to handle natural language queries in a specific language.

c. The chatbot will use a Python API to provide audio output through the Google TTS API.

Acceptance Criteria:

a. The chatbot should be able to manage and retrieve user data from the database.

b. The chatbot should be able to handle natural language queries and provide timely and accurate responses.

c. The chatbot should be secure and protect user data against unauthorized access or security breaches. d. The chatbot should be scalable and able to handle multiple user queries simultaneously.

e. The chatbot should be easily customizable to meet the specific needs of the mobile application.

f. The chatbot should provide a user-friendly interface that enhances the user experience.

g. The chatbot should be reliable and able to handle user queries without downtime or system crashes.

h. The chatbot should use the Google TTS API through a Python API to provide audio output.

Database Requirements:

a. The database should store user details, including user login and authentication information.

b. The database should store user query keywords and corresponding answers provided by the chatbot.

In conclusion, this SRS outlines the functional and non-functional requirements, constraints

***HARDWARE SPECIFICATIONS*** -

Processor: At least a quad-core processor, such as Qualcomm Snapdragon or Apple A-series

RAM: Minimum of 2GB of RAM, but 4GB or more is preferable

Storage: Sufficient storage capacity to accommodate the mobile app, database, and other files. At least 32GB is recommended.

Battery: A high-capacity battery to support extended use, preferably over 3000mAh.

Microphone: A built-in microphone that can capture audio input clearly

Speaker: A loud and clear speaker for audio output

Screen Size: A screen size of 5 inches or more is ideal for a better user experience

***SOFTWARE SPECIFICATIONS -***

Operating System: The application can be developed for Android or iOS platform. Android 9.0 (Pie) or later and iOS 12 or later should be supported.

Programming Language: The application can be developed using Java, Kotlin,Dart or Swift programming languages.

Database: A database management system like MySQL,SQLite or realtime Database of Firebase can be used to store and retrieve the data.

Chatbot API: A chatbot API like Dialogflow or IBM Watson can be used to develop the chatbot.

Text-to-Speech/Speech-to-Text API: Google Text-to-Speech API, Amazon Polly or IBM Watson Speech-to-Text API can be used for this purpose.

User Interface: A clean and intuitive user interface with easy navigation and clear instructions should be developed.