### ChatBot (Architecture)

### 1. Users Table:

* Table Name: users
  + Fields:
    - user\_id (Primary Key): Unique identifier for each user.
    - username: User's chosen username.
    - password\_hash: Hashed and salted password for security.
    - email: User's email address.
    - registration\_date: Timestamp indicating when the user registered.

### 2. Questions Table:

* Table Name: questions
  + Fields:
    - question\_id (Primary Key): Unique identifier for each question.
    - category: Category to which the question belongs.
    - question\_text: The text of the question.

### 3. Answers Table:

* Table Name: answers
  + Fields:
    - answer\_id (Primary Key): Unique identifier for each answer.
    - question\_id (Foreign Key): Links to the corresponding question in the questions table.
    - answer\_text: Text containing the answer to the question.

### 4. User Feedback Table:

* Table Name: user\_feedback
  + Fields:
    - feedback\_id (Primary Key): Unique identifier for each feedback entry.
    - question\_id (Foreign Key): Links to the corresponding question in the questions table.
    - user\_id (Foreign Key): Links to the corresponding user in the users table.
    - satisfaction: Boolean indicating whether the user was satisfied or not.
    - feedback\_date: Timestamp indicating when the feedback was given.

### 5. Support Agents Table:

* Table Name: support\_agents
  + Fields:
    - agent\_id (Primary Key): Unique identifier for each support agent.
    - agent\_name: Name of the support agent.
    - availability: Boolean indicating whether the agent is available for one-to-one chats.

### 6. Chat Messages Table:

* Table Name: chat\_messages
  + Fields:
    - message\_id (Primary Key): Unique identifier for each chat message.
    - user\_id (Foreign Key): Links to the corresponding user in the users table.
    - agent\_id (Foreign Key): Links to the corresponding agent in the support\_agents table.
    - timestamp: Timestamp indicating when the message was sent.
    - message\_text: Text content of the message.

### Collaborations and Flow:

1. User Registration:
   * Users interact with a registration interface to provide their information.
   * The registration data is stored in the users table.
2. Chatbot Interaction:
   * Users interact with the chatbot interface, triggering the retrieval of questions from the questions table.
   * The user selects a question, and the chatbot retrieves the corresponding answer from the answers table.
   * After presenting the answer, the chatbot asks for satisfaction feedback, and the user's response is recorded in the user\_feedback table.
   * If the user is not satisfied, the chatbot checks the availability of support agents in the support\_agents table.
3. One-to-One Chat Initiation:
   * If a support agent is available, a one-to-one chat session is initiated.
   * A chat session entry is created in the chat\_messages table to store messages between the user and the support agent.
4. User Satisfaction Check in Chat:
   * The chatbot or support agent may inquire about user satisfaction during the one-to-one chat.
   * If the user is satisfied, the chat session may end, or the chatbot can inquire if the user has additional questions.
5. Continued Interaction or Resolution:
   * If the user is not satisfied, the chat continues until the issue is resolved or the user is satisfied.
   * Messages exchanged during the one-to-one chat are stored in the chat\_messages table.
   * If resolved, the chat session ends.
6. Repeat Interaction (if necessary):
   * If the user has additional questions or issues, the chatbot can proceed to the next set of questions from the questions table.
   * The user selects a question, and the chatbot retrieves the answer from the answers table.
   * The chatbot asks for satisfaction feedback, and the response is recorded in the user\_feedback table.
7. Data Flow:
   * Data flows between the users, questions, answers, user\_feedback, support\_agents, and chat\_messages tables based on interactions and feedback.
   * Foreign keys establish relationships between tables.



