**TECHATHON**

**TEAM:**

Archana K-23i307

Mounika S S-23i334

Selva Mithra S-23i356

**TEAM NAME:** INNOVATORS

**PROJECT TITLE:** MENTAL HEALTH CHATBOT

**PROJECT DESCRIPTION:**

A simple web-based chatbot designed to provide stress relief and motivational support. This chatbot uses predefined responses, stress-relief exercises, and motivational quotes to help users feel better and calm their minds.

This project is a basic mental health chatbot built using Python, Flask, HTML, CSS, and JavaScript. The chatbot offers predefined responses based on user input, including stress-relief exercises and motivational quotes. The goal is to provide users with a simple tool to reduce stress and anxiety.

**FEATURES:**

**Features of the Stress Relief Chatbot**

* Interactive Chat Interface: The chatbot provides an interactive chat interface that allows users to communicate with the bot in real-time through a web browser.
* Predefined Responses: The chatbot can recognize certain keywords or phrases and respond with predefined messages. This includes greeting users, answering basic questions, and saying goodbye.
* Stress-Related Support: The chatbot offers support to users who mention feeling stressed or anxious. It responds with options for stress-relief techniques, providing a personalized and empathetic experience.
* Guided Breathing Exercise: The chatbot suggests a simple breathing exercise when users express the need to relax or mention breathing. This helps users calm down and manage their stress.
* Progressive Muscle Relaxation Technique: When users show interest in relaxation, the chatbot guides them through a progressive muscle relaxation exercise, promoting physical and mental relaxation.
* Visualization Exercise: The chatbot offers a visualization technique where users are encouraged to imagine a peaceful and calming environment, helping them to reduce stress and anxiety.
* Motivational Quotes: The chatbot provides motivational quotes when requested, offering encouragement and positivity to users who need a morale boost.
* Adaptable Conversations: The chatbot can handle different variations of user input related to stress, breathing, relaxation, and motivation, making the conversation feel more natural and less scripted.
* Fallback Response: If the chatbot doesn't understand the user input, it provides a default response asking for more information about how the user is feeling, keeping the conversation going.
* Web-Based Access: The chatbot is accessible via a web interface, making it easy to use from any device with internet access. Users can simply visit the webpage to interact with the chatbot.
* These features make the chatbot a useful tool for anyone looking to manage stress or seek some quick relaxation and motivation in their day.

**TECHNOLOGY STACK**

Python: The primary programming language used to build the chatbot application.

**Web Framework:**

Flask: A lightweight and flexible web framework for Python used to create the web application and handle HTTP requests. Flask serves as the backbone for the chatbot's web interface.

**Frontend Technologies:**

HTML: Used for creating the structure of the web pages.

CSS: Used for styling the web pages and making them visually appealing.

JavaScript: Used for handling client-side interactions and sending asynchronous requests to the server for chatbot responses.

Jinja2: The templating engine that comes with Flask, used to dynamically render HTML pages and inject chatbot responses into the frontend.

**Data Handling:**

Python Data Structures: The chatbot uses Python dictionaries and lists to store predefined responses, stress relief exercises, and motivational quotes.

Random Module:A Python standard library module used to randomly select a motivational quote from a predefined list.

Web Server:

Flask's Built-in Development Server: During development, Flask's built-in server is used to run the application. For production, a more robust web server like Gunicorn could be used in conjunction with a web server like Nginx.

Local Environment: The chatbot can be run locally using Flask's development server.

**CODE:**

Link:

* <https://drive.google.com/drive/folders/17CkrP4utkaQtIGqtyQ5pVFIJOl-0oW9R?usp=drive_link>