**WELLBOT GLOBAL WELLNESS ASSISTANT CHATBOT**

**PROJECT OVERVIEW:**

AI-powered wellness chatbot for Global users. Wellness chatbot designed to provide personalized health, fitness, mental wellness and lifecycle advice. It integrates with wearables devices and apps for personalized recommendations. Its offering 24/7 conversational guidance to users globally.

**OBJECTIVES:**

* Develop a smart chatbot that provides personalized wellness guidance.
* Leverage AI/ML models for adaptive health suggestions.
* Integrate real-time data from wearable devices.
* Ensure scalability and multilingual accessibility.

**FUNCTIONAL REQUIREMENTS**

* User registration and profile setup.
* Chat-based wellness interaction using NLP.
* Recommendation engine for diet ,fitness,sleep and hydrartion.
* Integration with wearable device APIs.
* Symptom checker and mental wellness support.

**MODULE -1**

**DATA COLLECTION & PREPROCESSING**

(1week)

STEP 1: search for dataset later find it in kaggle site and find the dataset related to the

Fitness for train the dataset model for the chatbot.

I find the dataset with size of 687771rows and 22 rows.

(2days)

STEP 2: programming ,first of all we import the pandas as pd.after import the pandas

We read our dataset it shows our total dataset.

Example:

import pandas as pd

df=pd.read\_csv("C:\infosys project\health\_fitness\_dataset.csv")

df

**COMMANDS**:

* df.head() : it displays the first 5 rows of the dataset.
* df.tail() : it displays the last 5rows of the dataset.
* df.info() : prints summary,including column datatypes,number of non-null

values,and memory usage.

* df.shape : it returns a tuple(rows,columns) indicating the dimensions of the

dataset.

* df.describe() : generates descriptive statistics(mean,min,max,std,quartiles) for

for numerical columns

* df.columns : returns a list of all column names.

(1day)

**DATA CLEANING: I use below commands for data cleaning**

* df.dtypes : returns a series with the data type of each column
* df.isnull(): returns a Boolean dataframe showing where values are missing.
* df.isnull().sum() : provides the count of missing values for each column.
* df,duplicated() : returns a Boolean series indicating which rows are duplicates.
* df.dropna() : removes rows(or columns) that contain missing values.

(3days)

**DATA PREPROCESSING:**

df.pivot\_table(index=’x’,column=’y’,values=’z’,aggfunc=’mean’) : creates a spreadsheet-style pivot table as a dataframe.

(1day)