AI HEALTHCARE CHATBOT DOCUMENTATION

NOTE:

Project source code along with X-ray Disease Classification Model is uploaded on Drive.
The link for the drive is:
https://drive.google.com/drive/folders/1nVc0HZwUZ4Eg7BjqiYGpTfGefuz-xE_3?usp=sharing

---

Amazon Lex:

- Importing a Bot

To import our lex chatbot bot

1. Sign in to the AWS Management Console, and open the Amazon Lex console at https://console.aws.amazon.com/lex/.
2. Choose Bots, then choose the bot to import. To import a new bot, skip this step.
3. For Actions, choose Import.
4. For Import Bot, choose the .zip archive that contains the JSON file that contains the bot to import. If you want to see merge conflicts before merging, choose Notify me of merge conflicts. If you turn off conflict checking, the $LATEST version of all of the resources used by the bot are overwritten.
5. Choose Import. If you have chosen to be notified of merge conflicts and there are conflicts, a dialog appears that lists them. To overwrite the $LATEST version of all conflicting resources, choose Overwrite and continue. To stop the import, choose Cancel.

You can now test the bot in your account.
Reference link:
https://docs.aws.amazon.com/lex/latest/dg/import-from-lex.html

- Do the following to publish a version of a bot you created:

1. In the Amazon Lex console, choose one of the bots you created. Verify that the console shows the $LATEST as the bot version next to the bot name.
2. Choose Publish.

3. On the Publish *botname* wizard, specify the alias BETA, and then choose Publish.
4. Verify that the Amazon Lex console shows the new version next to the bot name.

We made use of Boto3 is the Python version of the Amazon Web Services (AWS) Software Development Kit (SDK), which allows us to create software that utilises AWS services.
To utilise as code hooks for Amazon Lex bot, we constructed the AWS Lambda function. In your intent setting, you can specify Lambda functions to do initialization and validation, fulfillment, or both.

Database:

Library required :

- pymysql

For setting up the database we have used AWS RDS which is a SQL-based database. Any other SQL database can be used as per ease of use and access.

To set an AWS RDS follow the following tutorial :

https://www.youtube.com/watch?v=RerDL93sBdY

After setting up the database note the following parameters :

1. host
2. user name
3. password

Enter these parameters in db.py on lines no. 4 , 5, 6 respectively. Also, enter these parameters in app.py online no. 207, 208, 209.

If you have created a database then comment out lines no. 12 and 13. If you haven't then line 12 and 13 will create a database with the name Health_Bot. You can change the name as per your need but remember to put the same name on line no. 212 in app.py and on line no. 12 and 15 in db.py.

After completing the above-mentioned process and making necessary changes in db.py. run this file. This will create the required tables in the database. You can also manually create tables from the format given in db.py.

Main File:

File Name: FINAL_AMAZON_LEX_CHATBOT.ipynb
Description:
All the changes mentioned below are to be made in app.py file

- In line 121 enter the email of the doctor .It is hard coded as well because we did not have a database of doctors.
- In line 122 Enter your email address and password as argument . Do the same in line no. 136 .Do some security changes on the Mail id console for this to work .
- In line 124 put the link of the hosted video call service for which the code is provided in the video_call folder .
- In the line 127 and 128 enter your email address which you have used above .Repeat the same process in line no. 141
- Disease classification model is uploaded on google drive which was mentioned in the Readme file of github. (Add this model in *models* folder)

NLP and Model Block

---

1. Libraries required, Install the following libraries, if not done already. You may use pip/curl for installation:
    a. Pandas
    b. Numpy
    c. Tensorflow
    d. Nltk
    e. Difflib

    f. keras

2. These files are located in the models folder “sym_dis_map_base.json”, “Symptom_severity.csv”, “disease_model.h5” . if you intend to change the path 35, 39, 46.
    a. The sym_dis_map_base.json is a json map file that has the synonyms of the diseases, symptoms, words, split-phrases and terms for correlation and mapping.
    b. The Symptom_severity.csv file contains symptoms and their corresponding severities.
    c. The disease_model.h5 file is a keras based neural network for disease prediction, it can be accessed using keras.load_model function.

3. Place “symp.csv” and “dis.csv” files in the any directory and change the path to these files on lines 386 and 388 .currently they are placed in models folder
    a. The symp.csv file is a dataframe that contains a compressed one-line formatted input template.
    b. The dis.csv file is a dataframe that contains formatted disease variables.

4. Install “nltk.wordnet” if not already installed by opening a terminal/command prompt window.
    a. Type python / python3 or the custom command to open a python cell in the command line.
    b. When the python command line is activated, type “import nltk” and hit enter/return.
    c. Type “nltk.download('wordnet')” and hit return/enter key to start the download.
    d. “True” would be displayed as an output when the download is complete and the terminal can be closed.
5. Run the “finder” function before booting up the chatbot code-block/ line if not run before. The “finder” is a function that maps an input to the nearest term present in the data frames to make the input recognizable and formats the input to model requirements. A synset is made out of the parsed input and direct similarities are calculated to return.

MAP API :

- In line 154 provide **HERE** maps api key which you can get from : https://developer.here.com/
  (Due to unavailability of google maps api key linked with billing account **HERE** api was used)
- In lines 204 and 205 mention you ACCESS_ID and ACCESS_KEY for your AWS account: https://docs.aws.amazon.com/powershell/latest/userguide/pstools-appendix-sign-up.html

---

Multi Class Disease Classification Model:

File Name: MULTI_CLASS_DISEASE_CLASSIFICATION.ipynb
Description:

- Inside this file the model built for Disease Classification is included.
- Model weights which were used before evaluating the model are provided in the ***ModelWeights*** folder with the name FINALMODELDISEASECLASSIFICATION.h5.
- To see the model performance just run all the cells present with appropriate path for the model.
- To train the model just uncomment the cell present below the “TRAINING THE MODEL” markdown.
- Libraries required to run this file on local system:
  - numpy
  - matplotlib
  - pandas
  - tensorflow
  - Keras

The images uploaded should be saved in some folder .For this purpose currently the images will be saved in the Saved_images folder. If you intend to change the folder then please provide the relative or the static path at line no. 351 In the file app.py

Video Consultation:

File Name : index.php and script.js
Description:

- The code for the website is inspired from the open source tutorial provided on scaledrone website available at https://www.scaledrone.com/blog/webrtc-tutorial-simple-video-chat/
- The technologies used are WebRtc and scaledrone as a signalling server.
- The website is deployed on 000webhost.com for now. But you can deploy it as per your requirement .
- After deployment enter the link of deployed video call service at line no. 124 in the file app.py

System Flowchart