

Software Lab 4 Mini Project

Student Specific Chat Bot

OBJECTIVE

We aim to make an application that will cater towards students of colleges, as a helper portal that assists students in day to day activities. with the vision of making online sem feel more like real college attending, We also plan to create a bot that can assist in this application, and can act as a friendly assistant that automates and makes this app more user intuitive. Including the use technologies like NLP's in dialogflow and ML in tensor-flow to build the app and the bot is a future plan We will also use Cloud services to integrate the bot and the app more closely.

Members :

1. UJJWAL SHARMA - BT18CSE021 (TEAM LEADER)
2. KAUSTUBH KATHARE - BT18CSE024
3. ABIN JOSEPH
4. KAUSTUBH WANKHEDE - BT18CSE006
5. ADITYA DHANE
6. ADESH OSWAL

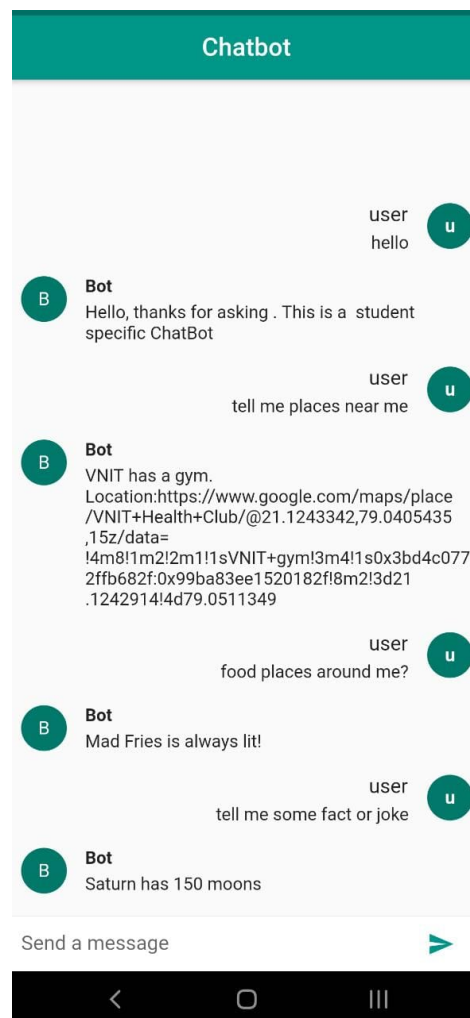
CONTRIBUTIONS

Application Development:

Contributors: BT18CSE006 Kaustubh Wankhede
BT18CSE025 Abin Joseph

The Application has been designed using **Flutter**, the chat application makes a request for every time you press send, and waits for the server to respond.

You can download the apk for this app in the apk branch of the project repo.



Cloud Server Hosting:

Contributors: BT18CSE006 Kaustubh Wankhede
BT18CSE024 Kaustubh Kathare
BT18CSE021 Ujjwal Sharma

The Flask Server combined with the NLP model is hosted on **Heroku**, under the free application resources.

We have the ability to host a free web server for up-to 550 hours this month, and will hope that the evaluation is done before our time limit is exceeded.

The screenshot shows the Heroku dashboard for an application named 'softwarelab4-chatbot'. The top navigation bar includes the Heroku logo, a search bar, and a user profile icon. Below the navigation bar, the application name is displayed along with a 'Personal' dropdown and a 'softwarelab4-project/Project' link. A 'Jump to Favorites, Apps, Pipelines, Spaces...' search bar is also present. The main content area is divided into two columns. The left column shows 'Installed add-ons' with a '\$0.00/month' tag and a 'Configure Add-ons' link. Below this, it states 'There are no add-ons for this app' and provides a 'Learn more' link. The 'Dyno formation' section shows a '\$0.00/month' tag and a 'Configure Dynos' link. It indicates 'This app is using free dynos' and lists a 'web gunicorn app:app' with a status of 'ON'. The right column displays 'Latest activity' with a 'View Activity' link. It lists three events: a deployment by 'softwarelab4project@outlook.com' at 2:10 PM, a successful build by 'softwarelab4project@outlook.com' at 2:08 PM, and a rollback to v14 by 'kaustubhkkathare@gmail.com' at 1:16 PM. Each event includes a timestamp, version number, and a 'Compare diff' link.

Note:

The applications tends to go to sleep over long time periods of no requests made.

When a new request is made after a long time, It will take a lot of time to wake up the application, so please be patient. The first few requests are bound to fail, or be unresponsive.

We don't have the financial resources to keep the application awake all the time, and hence request you to remain persistent and keep trying.

In case the app still fails to response after a good 30 minutes, please contact one of us, and we will restart the application from our side.

This inconvenience is deeply regretted.

Data Gathering

Contributors: UJJWAL SHARMA
KAUSTUBH KATHARE
ABIN JOSEPH
KAUSTUBH WANKHEDE
ADITYA DHANE
ADESH OSWAL

This data collected and modified according to the needs of university students by our team. We processed this data in JSON format and then made classes and associated them with sentences

Statement Processing

Contributors : Aditya Dhane
Adesh Oswal

Statement processing is done to assign specific classes to certain kind of statement , processing the statement, tokenising them, lemmatizing them and saving them in a more machine readable format

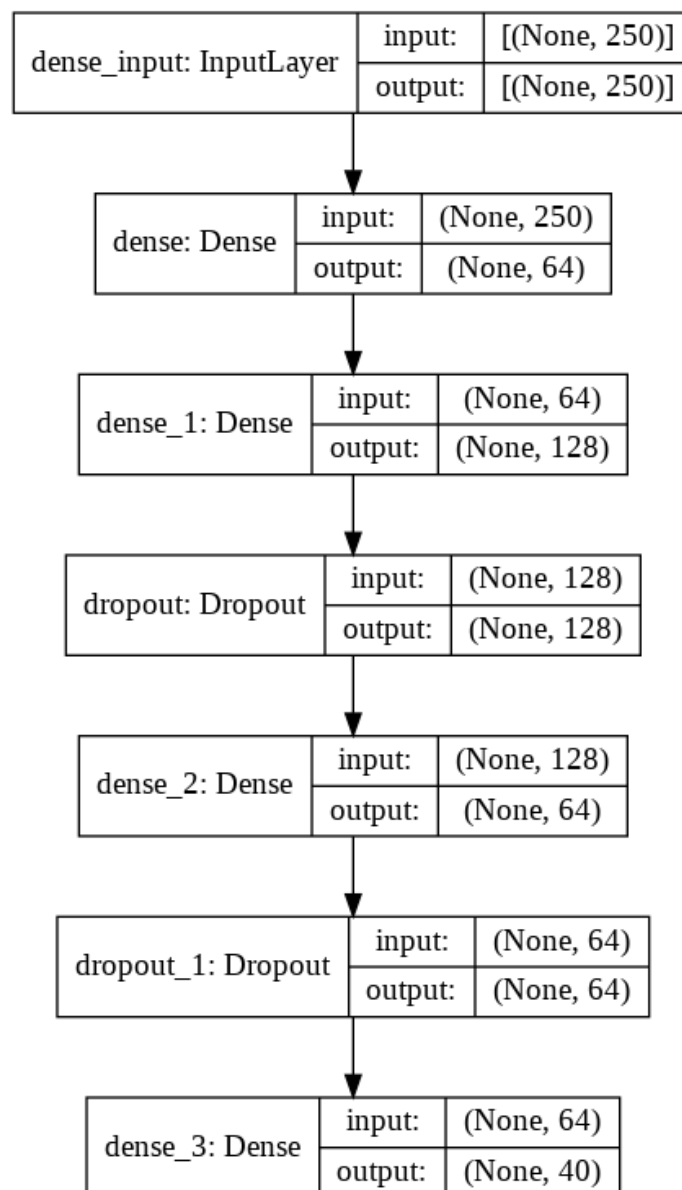
For this purpose we have used python's "nltk" module

Deep Learning (Model Architecture and Training) :

Contributors : Ujjwal Sharma

We have used python's "keras" model to design and train the deep learning model.
The model uses very complex techniques like dropout , Adam Optimiser for training to avoid over-fitting or under-fitting .
The model was trained for 200 epochs and the accuracy is close to 99.6%

THE ARCHITECTURE IS AS FOLLOWS



***** **END** *****
