DigiRep – A Chat-Bot For Class Representatives And Teachers

**Abstract**

Our aim was to build a chat bot for CR which act as representative for CR. We all used to ask many questions with CR regarding classes, subjects, lectures, timings etc. Sometimes it is not possible for a CR to respond to all queries, at that time to help CR this digital representative is so helpful. Firstly we created dataset which includes all details about classes. It contains tags, patterns and responses. Tags includes main header line of the sub topic, patterns includes all the different varieties of questions students might ask with CR and finally responses contains reply for all questions of the particular patterns. For code we used python as our base language. We did dataset in Replit. We used numpy, keras, nltk and tensor flow for building this chat bot. We used Twilio and created what’s app sandbox. We trained our bot and connected our Replit account with Twilio and we are able to send and receive messages from our DigiRep bot. Whenever then CR’s want to add information they can edit it in dataset(intents.json) and regarding updating any exams dates he can do it in MongoDB and other features are also included here. Finally, the DigiRep bot was created.

**Introduction**

We all know it’s sometimes tedious for the Class Representatives/Class Guides to answer all messages sent by the students which are most of the time repetitive. Why not build a chatbot which would answer student queries?

This was the prime reason for building our chatbot Digirep. It works as an interface between students and Class Representative making the class guide’s job easy. The basic queries asked by students can be easily answered using our bot. The very hectic job of a CR to answer each and every question which are most of the time repetitive can be easily countered by giving appropriate inputs to our bot which ensures that the student gets the required information.

# Tech Stacks used for the creation of DigiRep:-

DigiRep was accomplished by using the following Tech Stacks -

#### GitHub:

##### GitHub is a web-based platform that uses Git, a version control system, to manage and store code. It allows developers to collaborate on projects by tracking changes, sharing code, and providing a platform for developers to host their code. It also provides a variety of tools for issue tracking, project management, and documentation. Additionally, GitHub is home to a large community of developers who contribute to open-source projects and share their code with the world. It is widely used by individuals and organizations to host and collaborate on code projects.

#### Replit:

##### Replit is an online development environment that provides a platform for users to write, run, and share code. It supports a variety of programming languages, including Python, JavaScript, C++, and Java. Some of the features offered by Replit include a code editor, a file explorer, and a built-in terminal.

#### MongoDB:

##### MongoDB is source available cross platform document oriented database program. MongoDB uses json like documents with option like schemes. This is a document used to build highly available and scalable internet applications.

In this chat bot we use json for dialogue sets i.e: intents.json, These dialogue sets are later used for instant reply to the sender.

We have used Mongodb here to enable the CR to update the existing MSE dates to further dates using a command. We have dialogue set called “exams” in intents.json. A database called ‘chatbot’ with collection “examdates” is created and query is added with object id,name: mse1 and the dates,.whenever a user send a request for exam dates for mse1 the bot sends a response as set in MongoDB which can be modified as required by the CR.

A function called update is written in response.py which redirects the bot to MongoDB which updates the dates and saves it also if there is any error in syntax the bot displays an error. Another function called getdetails gets the updated date from MongoDB and displays the results as response from the bot.

Database access is URL is given as 0.0.0.0 and database URL is pasted in main.py file using app.config which is an XML file that is used as a configuration file for our application

A command given below must be typed to the bot in WhatsApp

* update-examdates-mse1-date-day/month/year to day/month/year -code123

The new dates should be typed in the above command and sent to the bot in with Twilio WhatsApp number and the bot would respond with “Date updated to day/month/year to day/month/year!!”, so the bot would now reply with the updated dates when the request for exam dates would be sent.

#### Python:

##### Python is a very approachable language for AI and Machine Learning as it is easy to read, write and understand while also providing a rich set of built in packages. Some of the popular Python libraries for AI and machine learning include TensorFlow, Keras, PyTorch, and scikit-learn. These libraries provide a wide range of tools for tasks such as data pre-processing, model training, and evaluation.

#### Twilio:

##### Twilio is a cloud platform. It is used to sending and receiving SMS and voice messages making and receiving phone calls. Twilio makes it easy to connect with customers and it's API allows developers to integrate communication into their existing application. We created one Twilio account, at the console navigate to the programmable messaging page. Their created what's app sandbox by clicking create a What's app Sandbox option, entered required information like name of owner, language we are using in our project and accepted API terms and services. Then copied What's Sandbox URL. We got Twilio number and tested Twilio by sending message to the number we have connected. Then in our Replit account, in create project page selected connect to sandbox and entered URL copied earlier. Then with entering all Twilio account credentials we connected to Twilio. When we edit our code in Replit and want to run, click run button in Replit, our code will be sent to Twilio and we will be able to send and receive message.

#### Bot Channel (WhatsApp):

##### WhatsApp was used as the channel for the user’s input as it is used by everybody from the classes. This was done using Twilio.

# Programs Used:-

DigiRep consists of 3 Python files- main.py, response.py, trainbot.py and one. json file i.e:- intents.json to store the dialogue sets

#### main.py:

The main.py file in our project is as the name suggests, a file that deals with the main execution part of the program. We have imported flask, the details from response.py file, Twilio messaging for message responses and also Pymongo from flask Pymongo.

There are 3 functions used in this file.

* 1. get\_db() to define the configuration method to return the database instance, also the unique code which permits to update any details in chat bot is defined here.
  2. Bot() The updates if any to be done in the chat bot is done using MongoDB and the code required for the same is given in this function
  3. Response() The string that has to be sent to the bot in order to make the updates is initialized here along with MongoDB URL. These three functions collectively form the main.py file which mainly has codes related to flask, Twilio messaging and MongoDB updates.

#### response.py:

Response is a powerful object with lots of functions and attributes that assist in normalizing data or creating ideal portions of code. Response object can be used to imply lots of features, methods, and functionalities. responses will search all registered Response objects and return a match. If only one Response is registered, the registry is kept unchanged. Some of responses functions are: response**.json()** returns a JSON object of the result **response.ok** returns True if status\_code is less than 200, otherwise False**response.status\_code** returns a number that indicates the status (200 is OK, 404 is Not Found). To work with responses command we need to have git hub and python installed.

#### trainbot.py:

At first we made use of the TensorFlow in the project for chat bot, which is the main important application that’s mainly used for data automation, model tracking, performance monitoring and model training. The modules used are random, SGD, dense, dropout, sequential, numpy, pickle and WordNetLemmatizer.

To filter the data, we use intents and patterns. WordNetLemmatizer is a large, freely and publicly available lexical database for the English language aiming to establish structured semantic relationships between words.

For initialising a bag of words we declare a list named bag that is used to store information in it. If the entered word is present in the pattern words then the command used is bags.append that appends or adds the word into it. Output row is a list of output that is empty. Then we add on the the training list that is been declared initially as training.append([bag, output\_row]).

Create model - 3 layers. First layer 128 neurons, second layer 64 neurons and 3rd output layer contains number of neurons. This layer has to be created then we have to equal to number of intents to predict output intent with softmax. Later on compile model. Stochastic gradient descent with Nesterov accelerated gradient gives good results for this model

### Sequential Layers/Neural Layers of DigiRep:

The prime idea of sequential application programming interface is to arrange the Keras layers in sequential order.  In sequential order the data flows from one layer to another layer in the given order until the data finally reaches the output layer.

Sequential data is 3-Dimensional:

* Batch dimension (groups of sentences)
* Time dimension (sequence length)
* Input dimension (vector length)

This Sequential model is appropriate for **a plain stack of layers** where each layer has **exactly one input tensor and one output tensor**. The basic functionality of Sequential is to make the provision of inferences and training of the module.

The TensorFlow sequential method helps create a sequential model of TensorFlow as per the specified arguments and attributes we mentioned. The function will execute and create each model layer one by one. The input\_shape is the argument of the sequential function that helps us define the layers that will be visible in the network.

Sequential TensorFlow API is the easiest way using which we can run and create the keras models. Some limitations may include not creating a model that can share the layers, having multiple outputs, having any of the branches, or having more than one input.

The general syntax of Sequential API model is:

TensorFlow. Keras. sequential(layers=none, name=none)

We have used TensorFlow sequential for Keras layers in our application. That is the present input is being influenced by the past input and works sequentially thereafter giving us the required output that the application demands.

**Activation Methods For DigiRep:**

Activation Functions play a vital role in Neural networks. They determine the output for of a given Neuron (Node) for a given input and predict the desired output based on Back Propagation. There are many types of activation methods some of which are Step, Sigmoid, ReLu and Leaky ReLu functions. For the Chat Bot, we have made use of the Rectified Linear Unit (ReLu) method of activation.

**The ReLu Activation Function:**

ReLu (Rectified Linear Unit) is an activation function used in artificial neural networks. It works by setting negative inputs to zero and leaving positive inputs unchanged, thus allowing the model to learn more complex relations. In practice, this activation function often provides superior performance to other standard activation functions. This is because ReLu is able to better approximate nonlinear functions that may exist between inputs and outputs, whereas standard activation functions are limited to linear functions. As a result, ReLu can significantly reduce the number of layers required in a deep neural network, allowing faster training times and improved performance on complex problems.

#### intents.json:

Intents file has all the data we will use to train the model/bot. It contains various tags…like greeting and responding.

The tags which we have implemented are about:

1. What’s going on
2. Greetings
3. Good bye
4. Contact details of
5. Free hours
6. Staff room location
7. Research interests
8. Physics notes
9. Chemistry notes
10. Basic electronic notes
11. See notes
12. Previous year question paper
13. Question bank
14. Exams
15. Moodle link
16. Info of events
17. What is robotech
18. Inquisitive
19. Code relay
20. Time
21. How many classes today
22. Who handle chemistry classes
23. Math classs
24. Biology class
25. Who handle cpp
26. Who handle basic electronics
27. English class
28. Attendance/compulsory
29. Link of chemistry class
30. Link of maths class
31. Link of English class
32. Link of biology class
33. Link of cpp class
34. Link of basic electronics class
35. Contact details of chemistry lecturer
36. Contact details of cpp lecturer
37. Contact details of Basic Electronics madam
38. Contact details of maths
39. Contact details of English
40. Contact details of biology
41. Java script mastery venue
42. Python workshop location
43. AI workshop venue
44. Physics assignment topic,date,portion
45. Chemistry assignment topic,date,portion
46. Basic Electronics assignment topic,date,portion
47. Set remainder for an event
48. Confirmation of whether remainder is set
49. Changes in remainder
50. From when does FLC registrations open
51. From when club location will open
52. When and where is ripple factor even
53. Thank you

# Packager Files:

Poetry.lock file:  Python project relies on external packages, so we have to make sure we’re using the right version of each package After an update, a package might not work as it did before the update. A dependency manager like Python **Poetry** helps you specify, install, and resolve external packages in our projects

Multiple packages like:-

astunparse, bson, cachetools, certifi, charsetnormalizer, click, colorama, cycler, debugpy, dnspython, docstringtomarkdown, etils, flask, flaskpymongo, flatbuffers, fonttools, gast, googleauth, googleauthoauthlib, googlepasta, grpcio, h5py, idna, importlibmetadata, importlibresources, itsdangerous, jax, jedi, jinja2, joblib, keras, keraspreprocessing, kiwisolver, libclang, markdown, markupsafe, matplotlib, nltk, numpy, oauthlib, opteinsum, packaging, pandas, parso, pillow, pluggy, protobuf, pyasn1, pyasn1modules, pyflakes, pyjwt, pymongo, pyparsing, pythondateutil, pythonlspjsonrpc, pythonlspserver, pytoolconfig, pytz, regex, requests, requestsoauthlib, rope, rsa, scipy, six, tensorboard, tensorboarddataserver, tensorboardpluginwit, tensorflow, tensorflowiogcsfilesystem, termcolor, tfestimatornightly, toml, tomli, torch, tqdm, twilio, typingextensions, ujson, urllib3, werkzeug, whatthepatch, wrapt, yapf, zipp have been used in this file.

.**Pyproject.toml:** pyproject.toml is the specified file format of [PEP 518](https://www.python.org/dev/peps/pep-0518/) which contains the build system requirements of Python projects.We have used pyproject.toml in our packager files. By default, poetry creates a python package with an appropriate name. The file pyproject.toml is used by poetry to keep a track of project info, python version, development dependencies and other externally installed packages. It is worth noting that the dependencies of dependencies is not stored in pyproject.toml file.

**Requirements.txt:** In Python requirement.txt file is a type of file that usually stores information about all the libraries, modules, and packages in itself that are used while developing a particular project. It also stores all files and packages on which that project is dependent or requires to run. Typically this file "requirement.”

1. For DigiRep, numpy, keras, nltk, tensorflow, Flask, pymongo[srv] and Flask-PyMongo are used..
2. **Numpy:**

NumPy basically means Numerical Python. It is basically a Python library used for working with arrays. NumPy aims to provide an array object that is up to 50x faster than traditional Python lists. It also has functions for working in domain of linear algebra, Fourier transform, and matrices.

NumPy arrays are stored at one continuous place in memory, so processes can access and manipulate them very efficiently. This behavior is called locality of reference in computer science.

Installation of Numpy can be done by PIP install command, we have installed NumPy using this command and then it is imported to our application.

## Keras:

Keras is a high-level, deep learning application programming interface developed by Google for implementing neural networks. It is written in Python and is used to make the implementation of neural networks easy. It also supports multiple back-end neural network computation.

1. We have installed Keras to our application along with TensorFlow using PIP install and then we have imported it.
2. **Flask Framework:**

Flask is a lightweight Python web framework that provides useful tools and features for creating web applications in the Python Language. It gives developers flexibility and is an accessible framework for new developers because you can build a web application quickly using only a single Python file.

Flask is a web framework, it's a Python module that lets you develop web applications easily. It's has a small and easy-to-extend core: it's a micro framework. it helps the application to create the responses. It does have many cool features like URL routing.

Flask is a web framework. This means flask  provides various tools, libraries and many other libraries are there in flask. So basically python programming has so many libraries but in flask many libraries are stored, This web application can be some web pages, a blog, a wiki or go as big as a web-based calendar application or a commercial website like chatbots and several other applications can be used,

Libraries are required for effective programming, else it will be difficult to perform programming without libraries.

Before setting up Flask for Python, You’ll need Python 2 or 3 installed. You will also need a text editor or IDE and access to the internet. Having some **basic knowledge of Python is useful**. An understanding of programming, data types, and for-loops is also advantageous. Flask is a good starting point to learn how to build web applications. You set

Flask only includes the template engine Jinja and a library called “tool”. But it offers the possibility to integrate third-party functions. It’s free and open source. As a counter-design to Django and other frameworks, Flask Python was quick to inspire a large fan community. you can separately install flask and combined to program or(bot).

## Natural Language Toolkit (NLTK):

The Natural Language Toolkit (nltk) is a library used for natural language processing tasks such as tokenization, lemmatization, and part-of-speech tagging. It is a powerful tool for working with and analyzing text data.

In response.py, nltk is used to lemmatize words in the user input. Lemmatization is the process of reducing a word to its base form, and it is used to preprocess the user input so that it can be passed to the chatbot model for prediction. The lemmatizer from the nltk library is first imported and initialized, and then the clean\_up\_sentence function is defined. This function takes a sentence as input and tokenizes it into words using the word\_tokenize function from nltk. It then applies lemmatization to each word using the lemmatize function from the lemmatizer, and a list of lemmatized words is returned.

In trainbot.py, nltk is used to tokenize the patterns in the intents.json file. The patterns are tokenized into words using the word\_tokenize function from nltk, and these words are then lemmatized. The lemmatized words are added to the words list, which is used to create the bag-of-words representation of the input data. The documents list is also updated to include the lemmatized words and their corresponding tags.

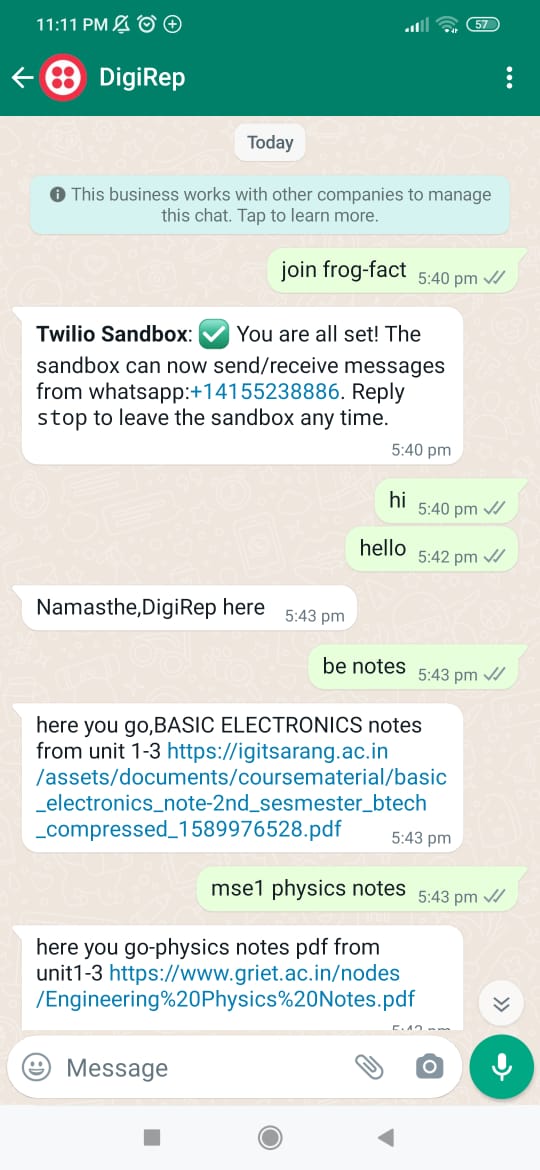
Overall, nltk is used in the model to preprocess the input data by tokenizing and lemmatizing the words. This helps to standardize the input data and make it more amenable to processing and analysis by the chatbot model.

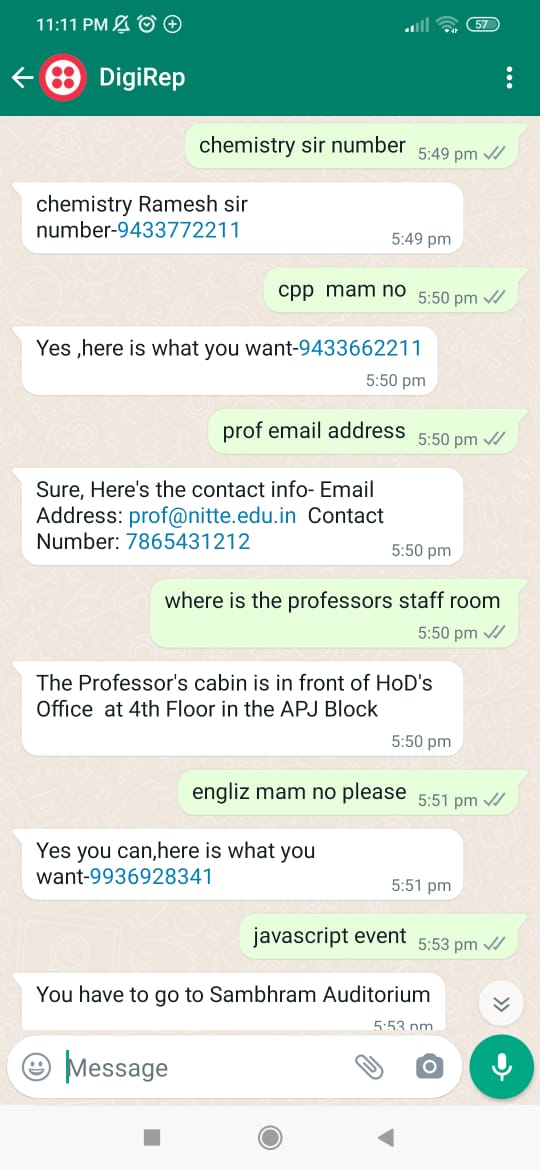
## TensorFlow:

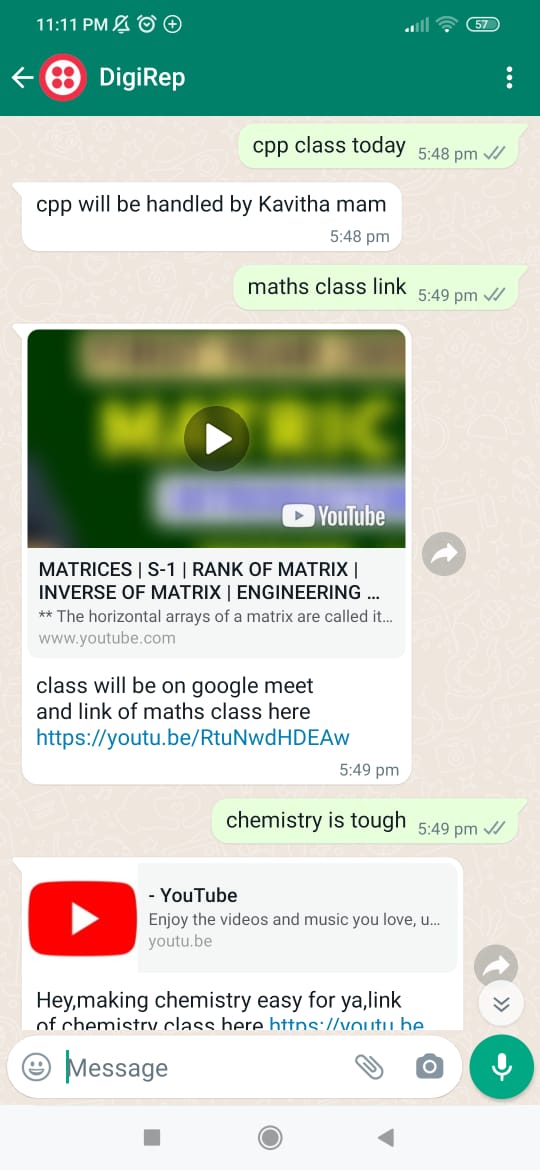
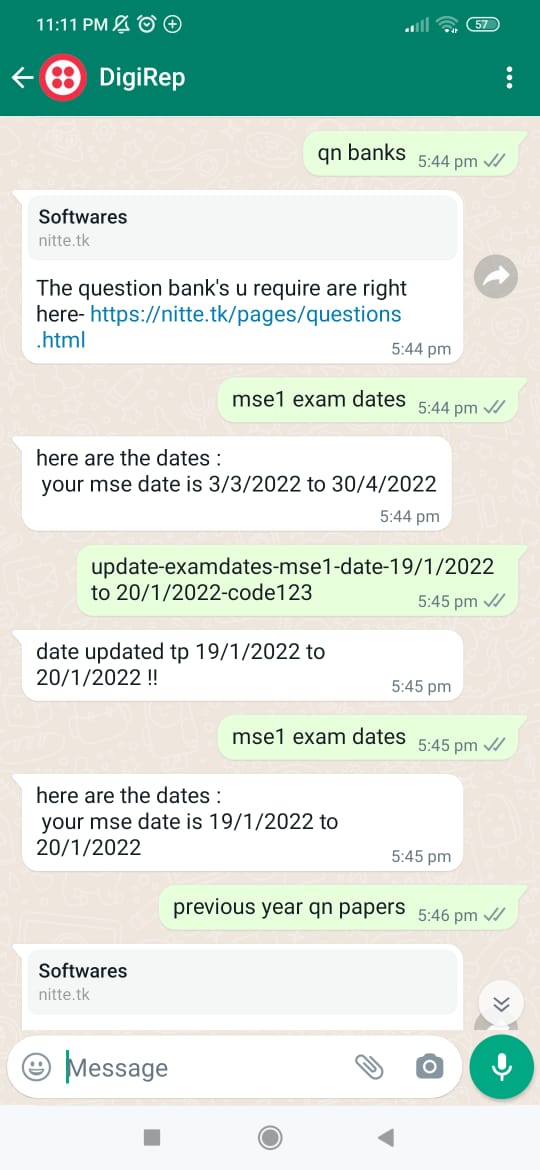
TensorFlow is a popular open-source library for building and training machine learning models. It is known for its ability to define and execute computational graphs. TensorFlow is also known to provide a range of tools for model training, evaluation, and deployment, including a variety of optimization algorithms and a flexible platform for distributed training. In addition, it also provides a range of libraries and tools for tasks such as data loading and preprocessing, model visualization, and deployment. These tools make it easier to build and deploy machine learning models using TensorFlow.

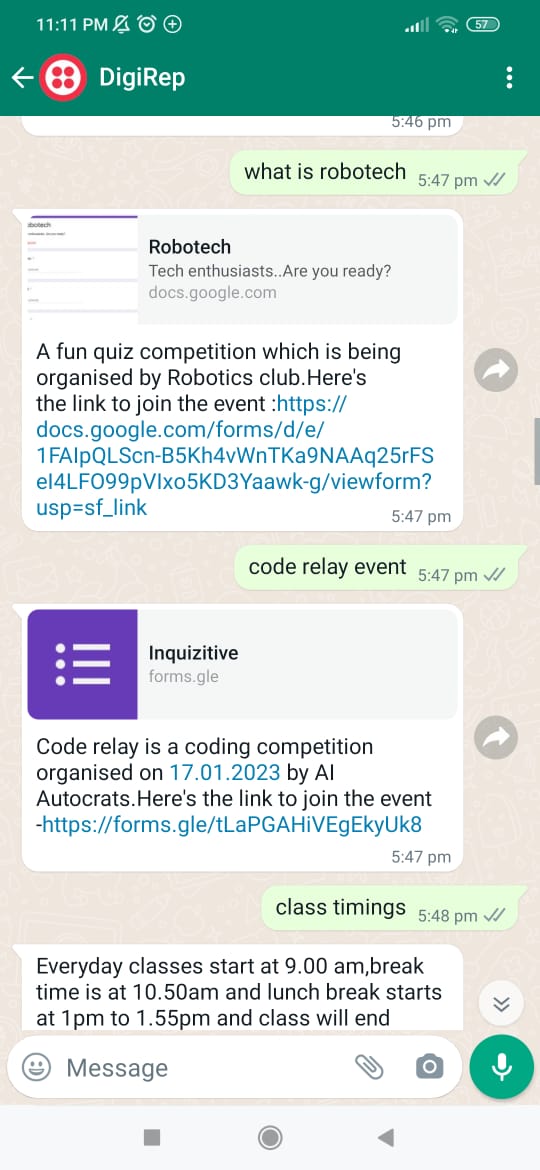
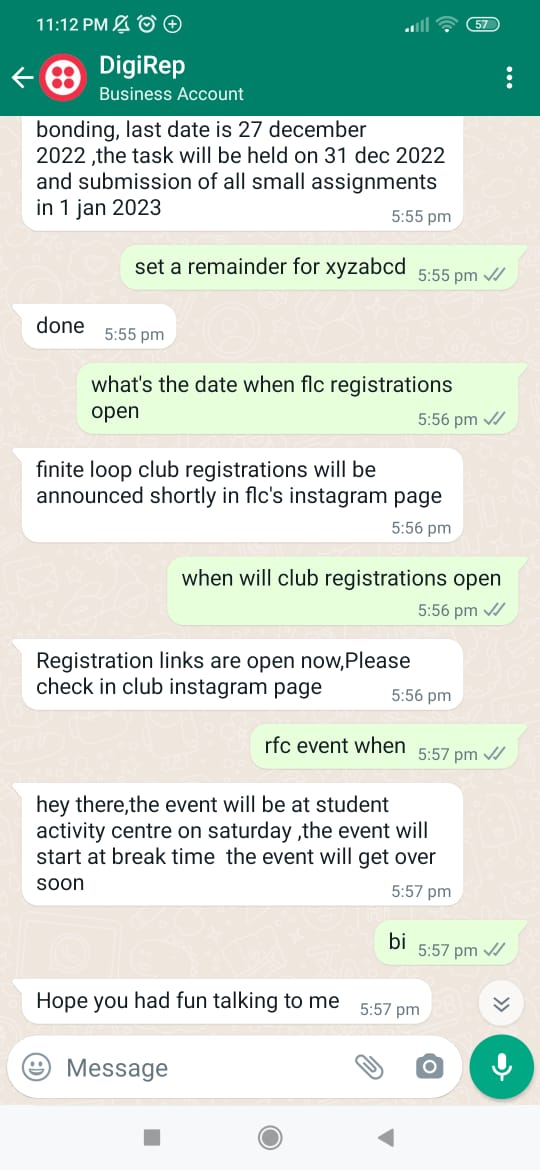
We have used TensorFlow to define (using the Sequential API) and train (using ‘fit’ method) the chatbot model that can classify user input into a set of predefined intents. The model consists of multiple dense layers with the ReLU activation function, and it is trained using the SGD optimization algorithm and the categorical cross-entropy loss function. The trained model is then saved to a file (using ‘save’ function) and can be loaded (using ‘load\_model’ function) and used for prediction at a later time.

**Working of Digirep screenshots:**

****

****





**Conclusion**:

Therefore this is all about our AI chatbot DigiRep which provides results for most of the student queries which are asked to the Class representatives,thus making the task of the CR easier and providing student service 24\*7 with faster response time.