Report - SE55 Design and Development of an AI Chatbot-FYP Report 2

Software Project Management (COMSATS University Islamabad)

Design and Development of an AI Chatbot

Final Year Project

Session 2016-2020

A project submitted in partial fulfilment of the
COMSATS University Degree
of
BS in Computer Science / Software Engineering (CUI)

Department of Computer Science

COMSATS University Islamabad, Lahore Campus

04 August 2020

Project Detail

Type (Nature of project)	[ ] **D**evelopment [ ] **R**esearch [ ] **R&D**			
Area of specialization	Chatbot development using NLP			
Project Group Members				
Sr.#	Reg. #	Student Name	Email ID	*Signature
(i)	FA16-BSE-084	Mahnoor	Mahnoor.rauf100@gmail.com	
(ii)	FA16-BSE-168	Ahtasham Sandhu	Ahtashamsandhu6@gmail.com	
(iii)	FA16-BSE-108	Hamza Tahir	Hamzatahir5621@gmail.com	

*The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others

Plagiarism Free Certificate

This is to certify that, I am Ahtasham Sandhu S/D/o Inam Ullah group leader of FYP under registration no CIIT/FA16-BSE-168/LHR at Computer Science Department, COMSATS Institute of Information Technology, Lahore. I declare that my FYP proposal is checked by my supervisor and the similarity index is 0% that is less than 20%, an acceptable limit by HEC. Report is attached herewith as Appendix A.

Date: 04-09-2020 Name of Group Leader: Ahtasham Sandhu Signature: _____

Name of Supervisor: Dr. Rao Adeel Nawab Co-Supervisor (if any):_____

Designation: Assistant Professor Designation: _____

Signature: _____ Signature: _____

HoD: _____ Signature: _____

Abstract

A chatbot is an artificial intelligence software that can carry out a chat with a user. 85% of customer interactions will be managed without a human by 2020(Gartner). Chatbots have a wide range of applications including placing an order, customer support, marketing, sales, and entertainment. There are two main types of chatbots; navigational (work with predefined quick replies and buttons) and artificial intelligence-based bots (work with NLP).

In this final year project, we aim to develop an AI Chatbot and target the travel industry. We'll be taking Travel Agency named PIA travels as a case study. The chatbot will be deployed on Facebook Messenger. Our chatbot will train itself on user inputs to provide appropriate responses. When users interact with our bot using natural language input our problem-oriented trained model will generate an appropriate response against his/her query. If in case, our chatbot can't understand what the user is trying to say it can be directed to chat with a human. Our chatbot will be able to let the user make a query using the provided buttons as well as text-based input. We can use our chatbot for marketing by broadcasting registration dates, success stories, or deals. The chatbot will help in improving services as the chatbot is accessible 24 hours.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1. Chatbot

A Chatbot is a computer program designed to initiate conversations with humans. For this purpose, it utilizes tools of Natural Language Processing. It is used to assist people by conveying a message through a human-like conversational flow or by providing a more convenient interface for humans to perform a certain task. Other tools such as Artificial Intelligence or Digital Image Processing can be used to make a Chatbot act more like a human being. Turing tests are conducted on Chatbots, which evaluate how good a Chatbot is in carrying out a human-like conversation

1.2. Applications of Chatbot

A chatbot can assist in the following ways:

- Reducing the requests load by answering the frequently asked questions and by receiving similar requests.
- Chatbots can effectively collect data from users.
- It can be used to provide recommendations for the users.
- Confidential credentials can be stored in the database using the chatbots such as the ATM pin codes.
- A user can perform certain tasks like seat reservations and online hotel bookings more efficiently by interacting with a human-like conversational interface.
- Chatbots can be used to advertise different products.
- The chatbot will help in improving services as the chatbot is accessible 24 hours.

Chatbots such as Siri, Cortana, Bixby, etc. are frequently used on different platforms as personal assistants to perform important tasks such as reminding and searching in a more efficient and user-friendly way.

1.3. Importance of Chatbot

A chatbot has a good number of benefits. Some of the benefits of a chatbot are:

- Chatbot saves user's and admin's time.
- It provides instant service and improves user experience.
- It saves the cost of hiring a lot of customer service representatives.
- It automatically does most of the work for you (i.e, saves the data provided by the user).
- It automates the chat for you, which is the future of chat.
- It is a developing field and is expected to grow substantially over the coming years.

1.4. Types of Chatbot

In terms of **complexity**, there are three types of Chatbots.

1.4.1. Button-Based Chatbots

These Chatbots present the user with some predefined buttons to choose from. Upon choosing a button, he is again presented with some preset buttons and this goes on until we reach an answer. These Chatbots utilize very basic technology and take up considerable user time.

1.4.2. NLP Based Chatbots

These Chatbots utilize machine learning as well as artificial intelligence to generate feedback. Using tools of Natural Language Processing, They can remember previous conversations and respond differently based on what a user has said before.

1.4.3. Hybrid Chatbots

These Chatbots make use of buttons as well as Natural Language Processing. Parts of conversation where very specific responses are required, buttons are used and for other parts, natural language is allowed to deal with user inputs. Even some of the most advanced NLP based Chatbots have some buttons in specific areas of conversation.

In terms of **purpose**, there are again three types,

1.4.4. Sales Chatbots

These Chatbots are designed to help a user place an order. They present the products to the user, ask for his details, and place the order according to his choice and specifications.

1.4.5. Marketing Chatbots

These Chatbots promote a user product, service, or idea through personalized messages or advertisements. At the most basic level, we broadcast our advertisement and each user gets a message of our promotional content.

1.4.6. Support Chatbots

The prime purpose of this chatbot is to assist a user in some way. It can be as simple as helping a user finds a specific tab or facility on a webpage.

1.5. FYP Focus

In this FYP, we focus on the conversational flow of our chatbot using both buttons and Artificial Intelligence. Our Chatbot should be trained enough to handle user queries smartly while being efficient and easy to use. Buttons will assist t
he user in finding the most frequently asked questions. If the user doesn't find the question he needs to ask in the buttons, he will type it himself, and as a result, our trained AI model will return the relevant response. We are focused on making this process simple enough, so the user knows how he can answer the questions properly and gets the proper response which can help the user in booking a ticket. The responses will be refined and simplified so that they are easy to understand for everyone. They will be formatted in such a way that each piece of information is distinguishable.

1.6. Motivation and Scope of the Study

The chatbot is relatively new and is gaining popularity day by day. Messenger bots are a new method of communication between businesses and customers. By developing this chatbot, we can help users get information or provide services within the palm of their hands no matter where they are. The travel-industry can provide its users with 24 hours available service. Downloading and installing apps is no longer necessary and the use of Chatbots allows for personalization possibilities. We will utilize the machine-learning model to generate responses to the user's queries. Along with that, we will also implement keyword matching for simpler queries. In the future, our chatbot will also implement "webview" which will assist in providing a UI to the user to directly register to the official website.

1.7. Objectives of the Study

The goals and objectives of our FYP project are as per the following:

- To develop skills in designing and implementing an AI-based messenger chatbot.
- Develop an understanding of intents and entities.
- Develop skills to implement natural language processing in Chatbots.
- Implementation of carousels in our bot to enhance the look.
- Design a conversational flow where users will be able to book tours and choose the date and time they want.
- We will give our maximum capacity in learning new methods and techniques using the knowledge gained in our undergraduate degree.
- Develop skills to implement Web Scraping for scraping flights information from the Airline's official source

Chapter 2

Literature Review

2.1. Popular Chatbot Frameworks

2.1.1 ManyChat

ManyChat[1] is a chatbot framework that allows you to create a Facebook Messenger bot. It focuses on bots related to marketing, support, and sales. Manychat is a popular framework. The most popular feature in ManyChat is the Visual flow Builder. It gives a visual look to the bot and makes the conversational flow easy to build. It gives many ways to customize and grow the page engagements. ManyChat has a free and pro version with a difference in features. ManyChat provides different tools like automatic sequences, broadcast, autoresponders which helps turn users into subscribers. No coding is required for the basic functionalities of a bot. Manychat offers the facility to manage your bot on your own or even add more admins, live agents, and viewers. It also offers a Learning curve that lets the bot builder start easily without the need for tutorials. Other features include quick replies, live chat campaigns, etc.

2.1.2. ChatterBot

Chatterbot[2] is a great library that lets you develop bots with all basic functionalities including some machine learning and dialog engine features. It allows businesses to automate their response according to the user's input. It uses different machine learning algorithms to automate conversations. It is a language-independent platform thus making it speak any natural language. It is a python library and can be downloaded using pip command. Accuracy of response increases by testing more inputs.

[1] https://www.maxvancollenburg.com/manychat-tutorial/#what-is-manychat Last Visited: 10/6/2019
[2] https://www.predictiveanalyticstoday.com/chatterbot/ Last Visited: 10/6/2019

2.1.3. ChatterOn

ChatterOn[3] is another great chatbot framework that provides us with different tools to develop a messenger bot. It requires minimum coding. It is India's first development platform for a bot. It provides over twenty prebuilt chatbots including intents and entities and conversational flows for the educational purpose. It allows you to integrate your Business's API with the chatbot for better user interaction. The platform supports a variety of content such as photos, gifs, videos, and carousels, etc. The platform provides free facilities until fifteen thousand messages per month. It aims to provide a flexible and context related good chatbots.

2.1.4. Chatfuel

Chatfuel[4] is one of the most popular bot development frameworks. It started bot development targeting Telegram and then shifted to Messenger bots which is currently its focus. The main reason why Chatfuel is so popular because they have great community support through Facebook groups or email. Issues are resolved as fast as possible and help required is given. It has a user-friendly interface and a great User Experience. It is unique in terms of User experience. Chatfuel is completely free when creating a bot but the bot shows the Chatfuel branding. It can be upgraded to the Pro version. The monthly price of bot depends on the number of reachable users. The core features of Chatfuel include Plugins. Chatfuel offers a lot of different plugins JSON API, Calendar, Google Site Search, Zapier, etc. Chatfuel has different templates that are free and accessible. It is overall flexible and easy to use.

[3] https://therodinhoods.com/post/chatteron-build-ai-chatbots-5-minutes-without-coding/ Last Visited: 10/6/2019
[4] https://docs.chatfuel.com/ Last Visited: 10/6/2019

2.2. Popular Chatbots

2.2.1. BotsifyChatbot

Botsify[5] is an AI-based chatbot where students and teachers interact. The bot simplifies the interactions between teachers and students. Students choose from many educational topics that they want to learn. Botsify allows the students to learn topics in a conversational manner which assists in learning complicated topics step by step at a faster pace. The bot provides images, videos, and text related to the topic. Students learn from these, and after that, they can take quizzes. The marks are submitted to the teachers which makes the teacher's work less hectic.

2.2.2. Hubert

The main purpose of Hubert [6] is to receive feedback. Hubert allows teachers to receive feedback related to courses, their teaching methodology, quizzes, and assignments from students. This helps teachers recognize problems faced by students and therefore, they can work on possible solutions. Hubert chats with students and asks several questions. These questions are mainly focused on how a student is feeling about a course or a teacher. Responses are then compiled and categorized. Finally, a summary of the responses is sent to the respective teachers. Sending the summary instead of the whole text saves a lot of teacher's time. The teacher only needs to know the actual essence of what the student is talking about and Hubert is very efficient in providing that.

[5] https://botsify.com/education-chatbot Last Visited: 03/10/2019
[6] https://hubert.ai/ Last Visited: 03/10/2019

2.2.3. Jill Watson

Jill Watson[7] and online teaching assistant, developed at the Georgia Tech University, which gives a response to student's queries and issues. Jill Watson addresses one of the main issues with online courses. Students learning from an online course cannot ask the teacher about the issues they face and therefore have a hard time learning the subject. Sometimes, there are forums where a student may ask his question and it can be responded by the mentor. But many a time, the number of questions by students exceeds the time a teacher can dedicate to responding to them. Jill Watson categorizes each question asked by the student and returns the answer which is mapped to that category. Jill Watson is deployed in the Georgia Tech University. Jill Watson is trained to identify multiple versions of the same question and therefore it is so efficient that the students of Georgia Tech are unaware that it is not a human.

2.2.4. TinkerBot

Tinkerbot[8] is an AI-based messenger chatbot that helps people learn new and unique concepts by simply talking to them. The usual learning methods include reading boring articles or listening to hectic and dull lectures for hours. Tinkerbot, on the other hand, just makes a conversation with you and in those conversations, he teaches you new concepts and topics. Between these conversations, the tinkerbot will also ask questions to know if you have learned the concept or not. Tinkerbot includes several courses to choose from. The conversational methodology of teaching is particularly useful as it engages the learner in learning the concept. Tinkerbot allows personalized learning as each student receives full-on attention.

[7] https://smartech.gatech.edu/bitstream/handle/1853/59104/goelpolepeddi-harvardvolume-v7.1.pdf Last Visited: 03/10/2019
[8] https://tinkerbot.in/ Last Visited: 03/10/2019

Chapter 3

Work Done so Far

3.1. Restaurant chatbot in the learning process

We'll be targeting the restaurant industry domain when creating our chatbot. Food industry domain, like many other domains, requires considerable staff to answer daily queries related to services, food, order, and rates, etc. which is time-consuming, expensive and leaves a margin of error. If the number of queries each day surpasses the time that can be utilized for answering them, the staff can no longer respond to them and therefore, is not a reasonable solution.

By taking Royal Restaurant as our case study, we shall use our chatbot to show how we can quicken the question-answer process and eliminate all of the issues mentioned above. As a chatbot answers simultaneously to each user's queries, there is no waiting, no delay in each response, and still, no staff required.

3.2. Conversational Flow Design

As the user opens the chat messenger, he receives the welcome message that includes greetings, and a question that he/she wants to order the food(as shown in Figure 1 [1]). As the user continues he receives a request to get the user's location (as shown in Figure 1 [3]). Scrolling further down shows the carousels based menu, in this menu Foods, are separated according to their categories using the carousels and the buttons (as shown in Figure 1 [4]). A carousel has the name of its category on the top and its related questions as the buttons at the bottom of the carousel.[4] shows the category of a carousel.[5] as the user presses the details button the user will get those specific food items shown in different carousels (as shown in Figure 1 [8]). After the user will select one of the food items he/she wants to order (as shown in Figure 1 [9]).After that, the bot will ask in how much quantity do you want that specific item and the user will enter the quantity(as shown in Figure 1 [10]). After that the bot will question the user did you want to order again if user's reply "yes" then the user will be redirected to the carousels based menu again and the procedure will be repeated and if user's reply no then the user is asked that do you want to see the cart or wants to get your invoice if the user presses the cart option the user is redirected to cart and the cart is displayed (as shown in Figure 1 [11]). In Cart, the user is asked that do you want to edit

the cart if the user clicks the edit cart option then the user is redirected to the edit cart (as shown in Figure 1 [12]). After the editing is done user is again redirected to Cart [8]. And if the user doesn't want to edit the cart he/she is redirected to the questions that he wants to see the cart or get the invoice. And if the user selects the invoice option then he/she is directed to the invoice (as shown in Figure 1 [13]). "Live chat" [6]. There is a category called "Live chat" [6]. Live chat is an option where if the user wants to quit interacting with the Chabot, he can talk to an actual person (if one is available or otherwise he gets a message that no operator is available at the moment).and after the live chat order the user is redirected to the invoice (as shown in Figure 1 [13]).

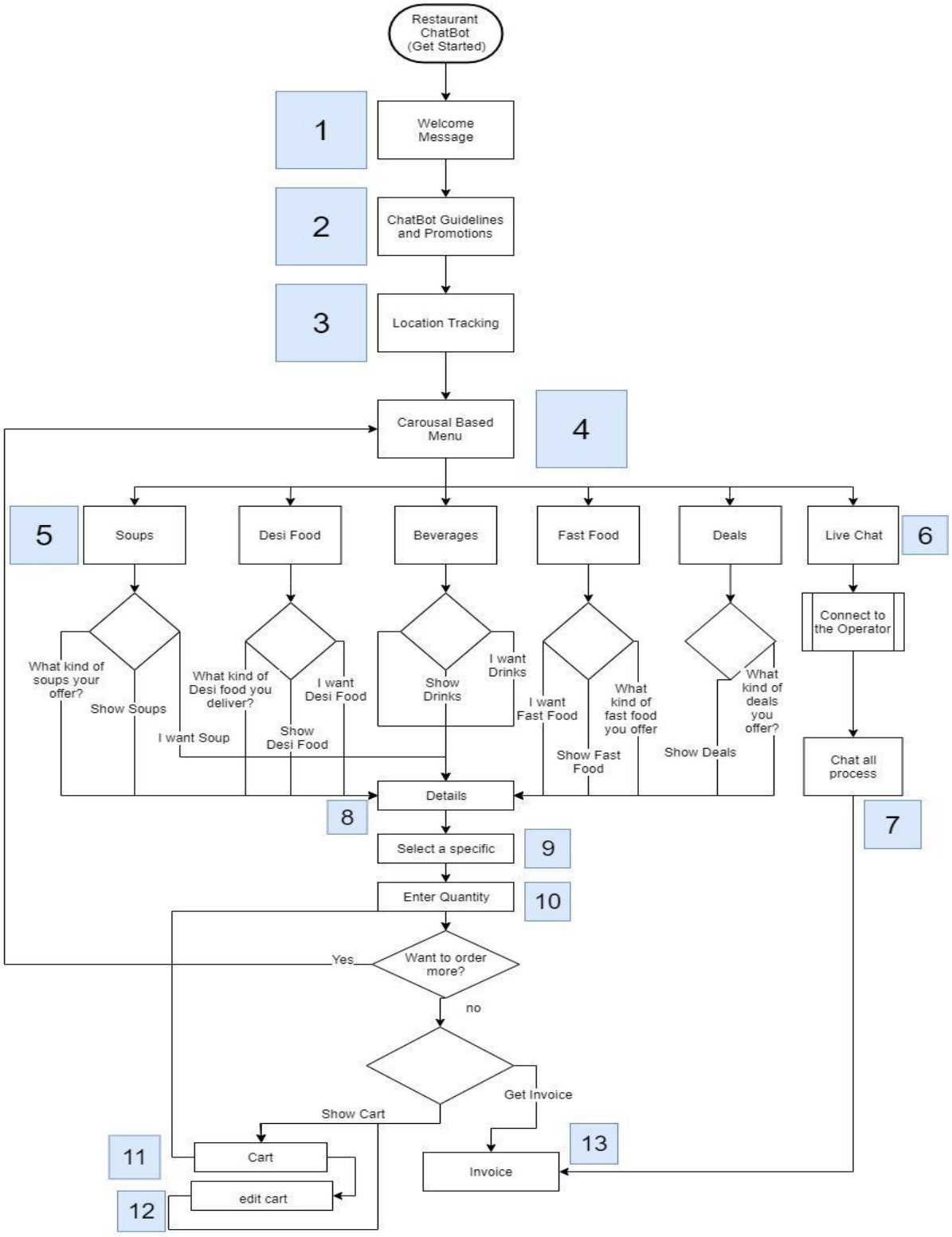
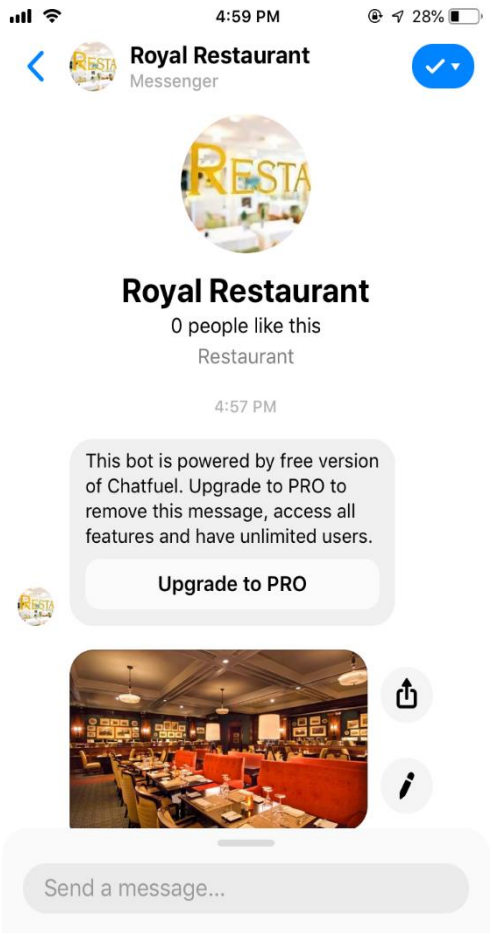
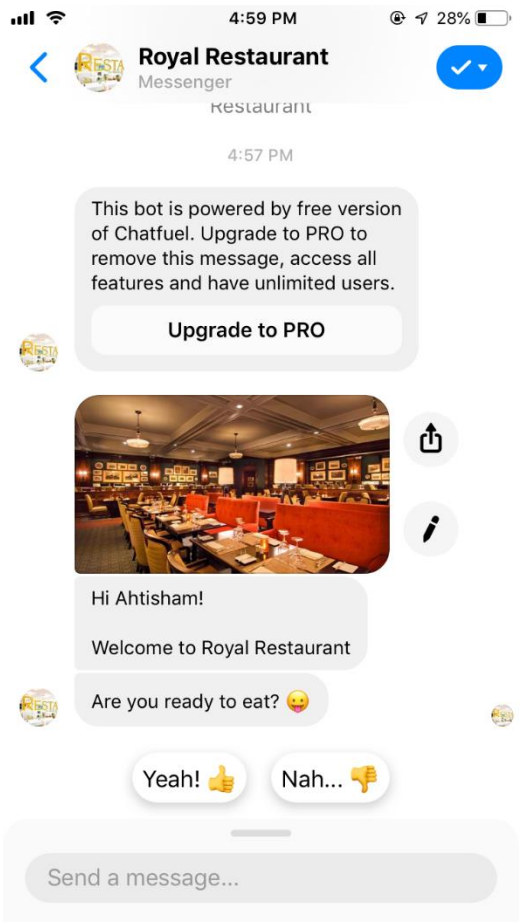


Figure 2: Conversational Flow Diagram for Restaurant Chatbot

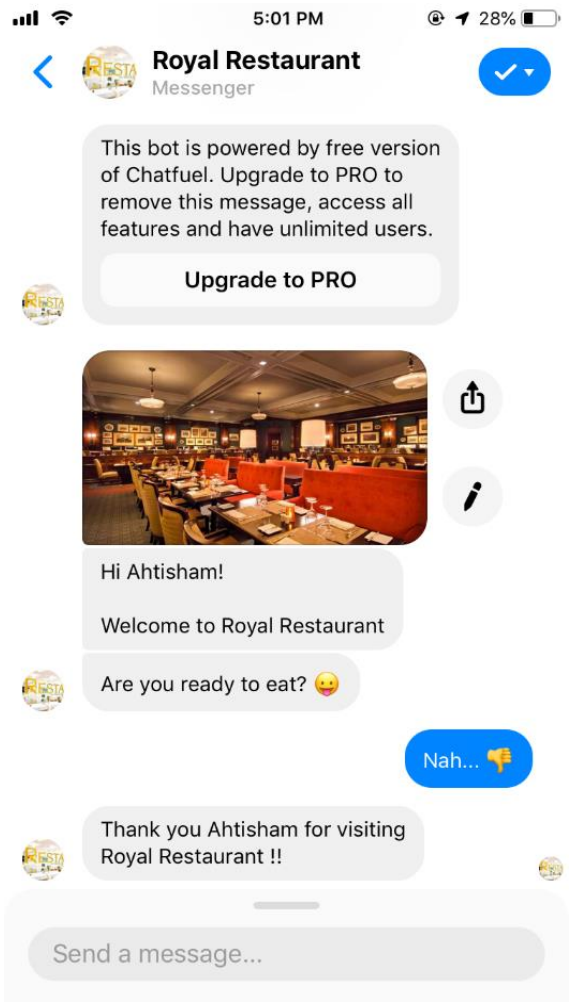
3.3. Implementation of Button-Based Chatbot

Our Chatbot utilizes buttons as well as Natural Language Processing. For our FYP-1, we have implemented the button part. We compiled a list of FAQs and categorized them (as shown in section 3.2.2). To represent each category, we have made use of carousels in our chatbot. Each carousel contains buttons and each button represents some activity related to that category. Carousels can be swiped left or right to reveal other carousels. In this way, our users can navigate through the different categories.

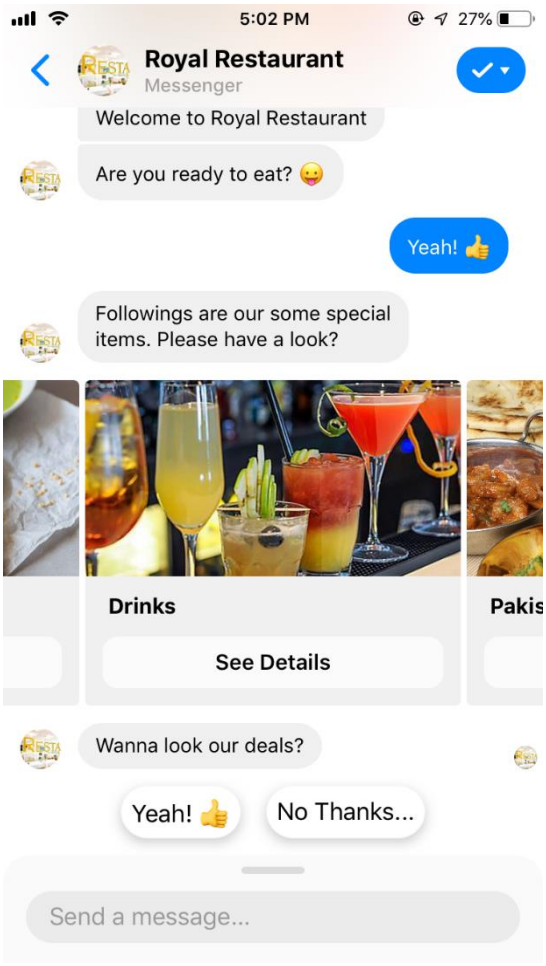




In this figure, we are just showing a welcome message to the user when the user will interact with our bot initially.

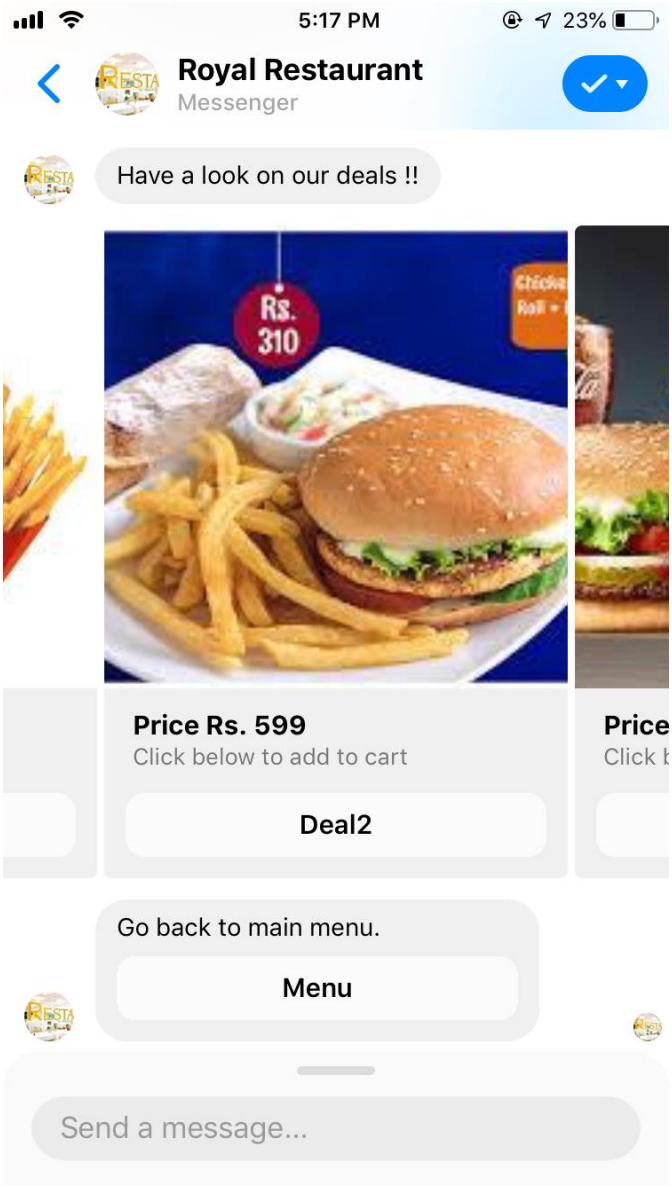


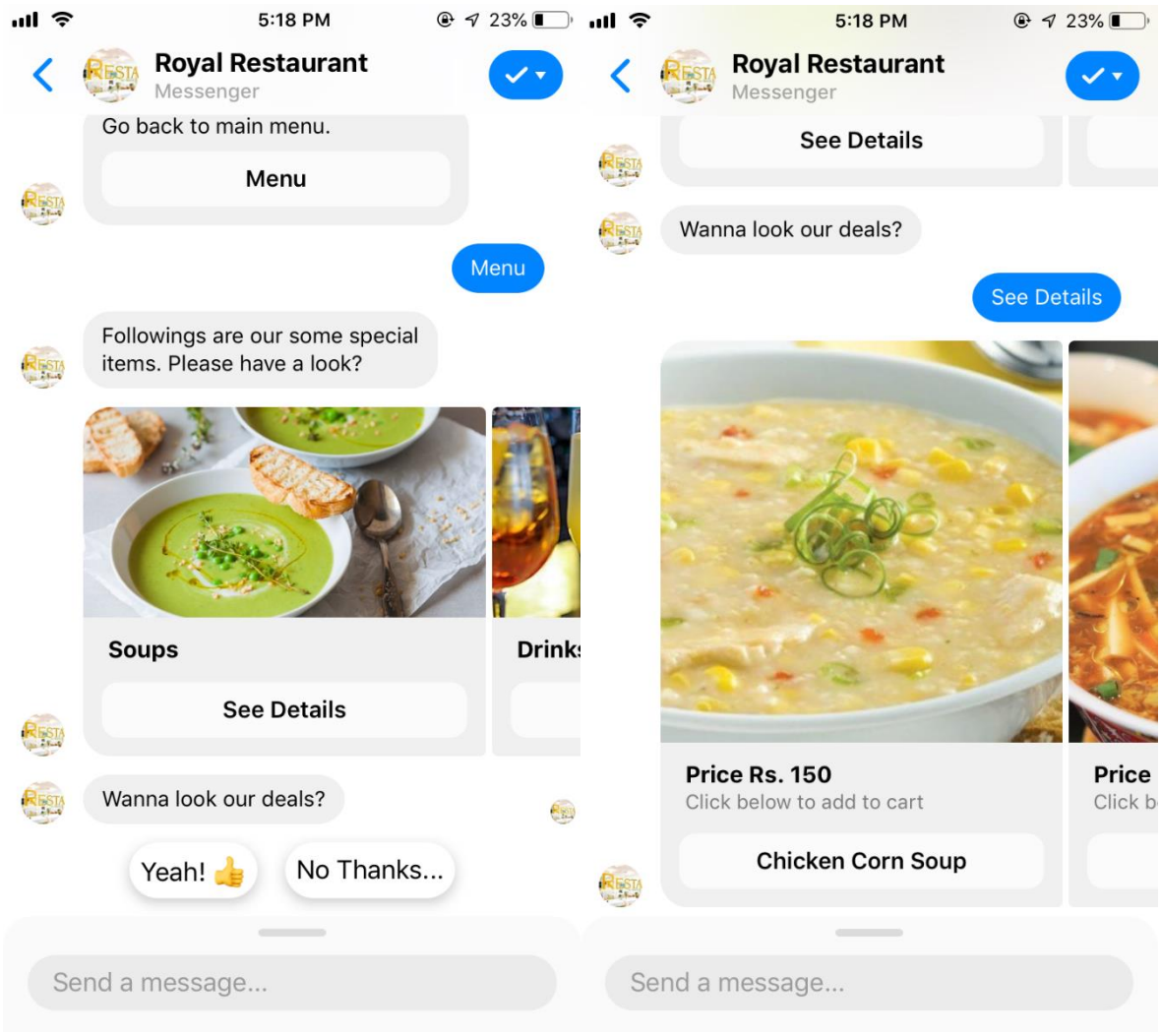
If the user will press the “No” button, then the bot will straight forward display the message that thanks for visiting.



When the user presses the “Yes” button, the food menu will be shown to the user in the form of a carousel and the user can see all the categories by swiping right as shown in the above figure. Moreover, another option will be shown to the user whether he/she wants to see the current deals or not. If the user will click on the “No Thanks” button then again the user will be redirected to the main menu and will be shown different food categories to operate further.

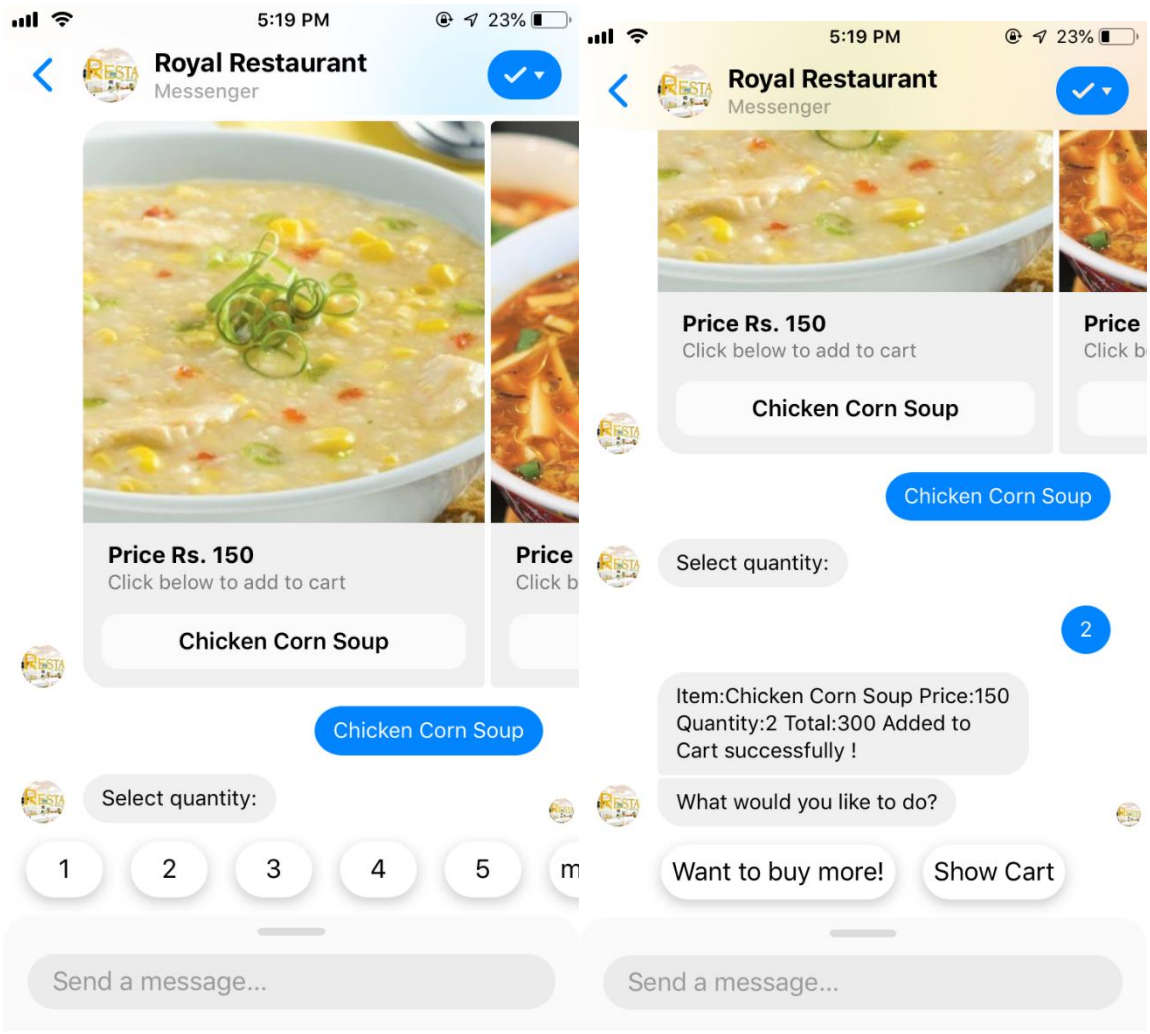
When the user will click on the “Yeah!” button, he/she will be redirected to current deals including different items with the desired price and the user can add them to his/her cart by clicking on the button as shown in the figure given below.





After visiting deals, the user can go back to the main menu by clicking on the “Menu” button as shown in the above figure on the left side.

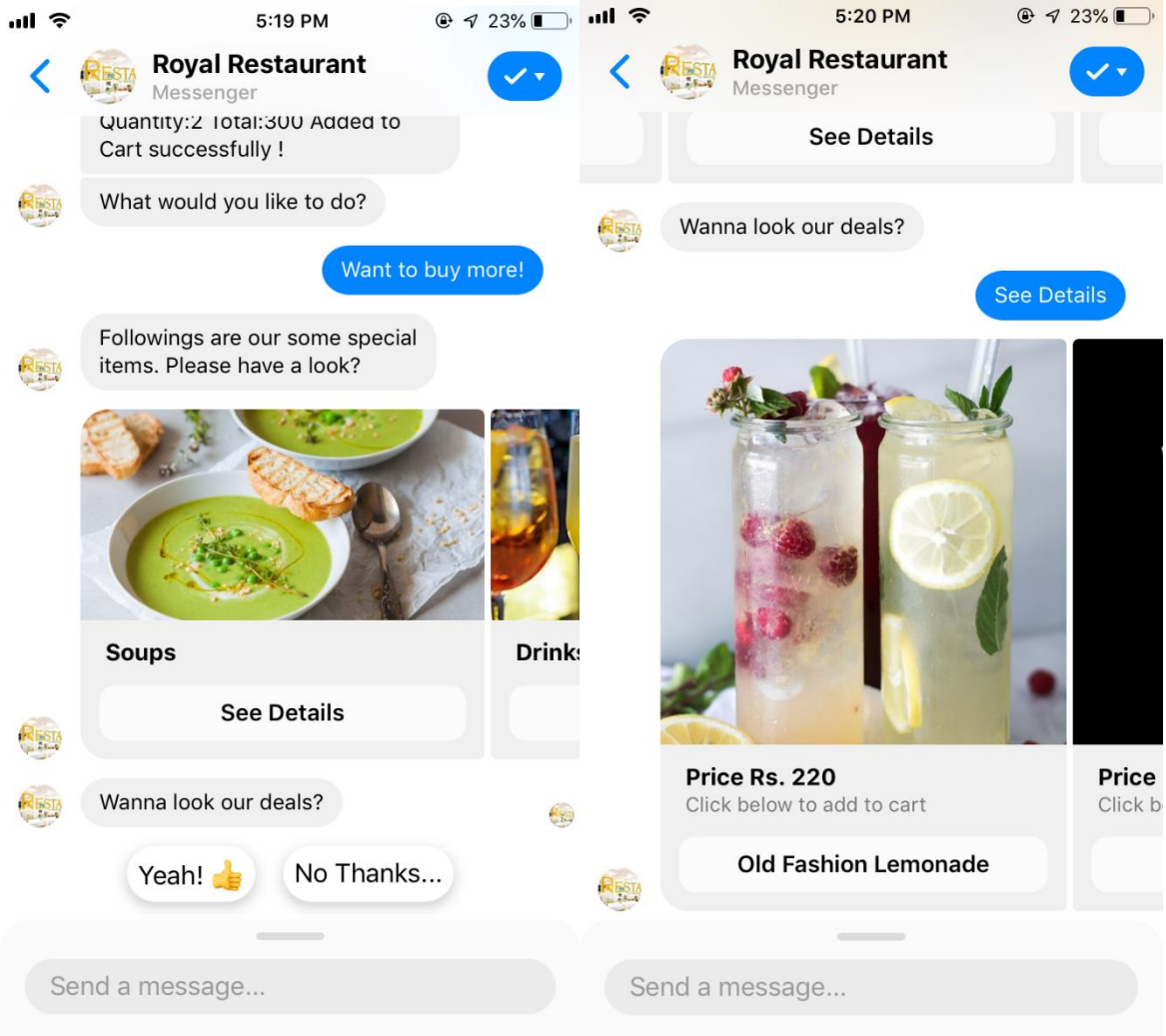
Moreover, details of any category will be shown to the user when the user presses the “See Details” button. Now, here are some items of the soup category as shown in the above figure on the right side, and the user can also have look at more types of soup by swiping right in the carousel.



When the user clicks on “Chicken Corn Soup”, he/she will be asked to select a quantity. In this case, users have selected 2. After this, the chosen item will be added to the cart and a message will be displayed on a screen that says “You have successfully added the item to your cart”.

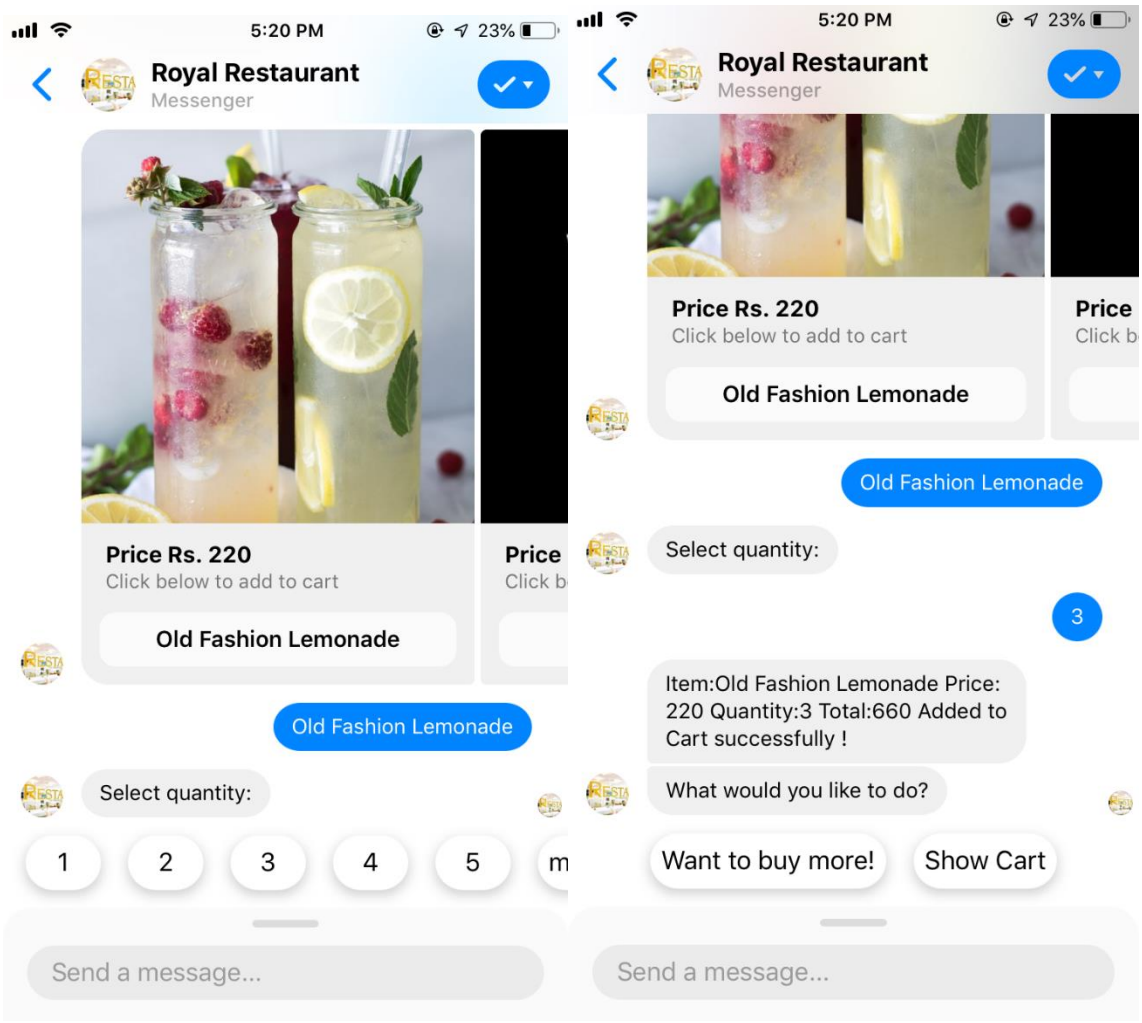
After this user will be provided by two options:

1) Want to buy more?
2) You want to see your cart.



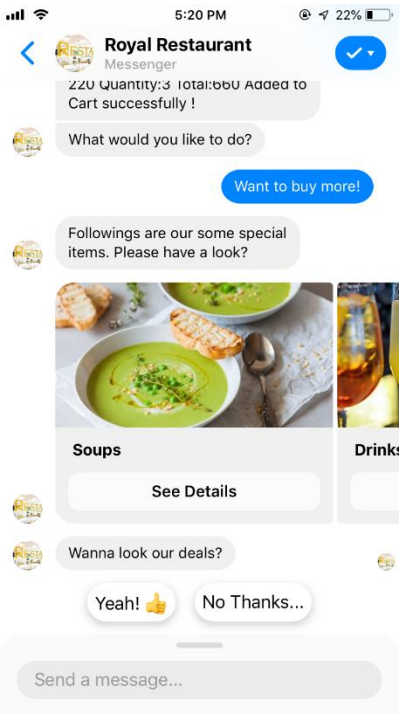
Now if the user will press the “Want to buy more!” button, he/she will be again redirected to the main menu as shown in the above figure.

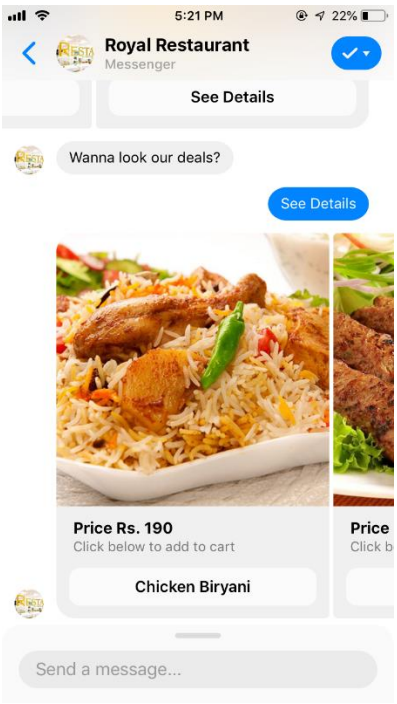
In this case, the user has to press the “See Details” button to see all items related to the drinks category and he/she will be shown other drinks by swiping right.

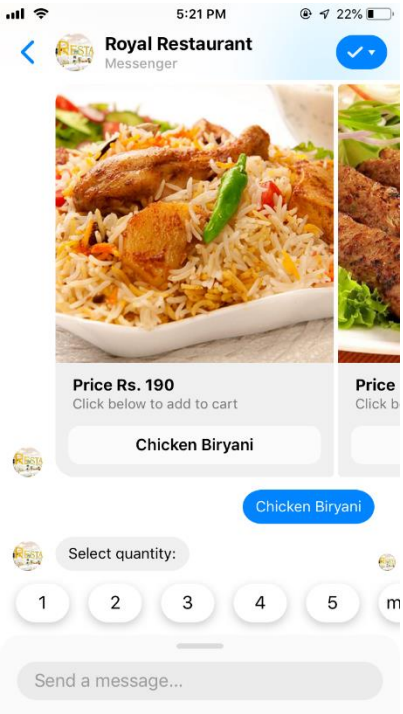


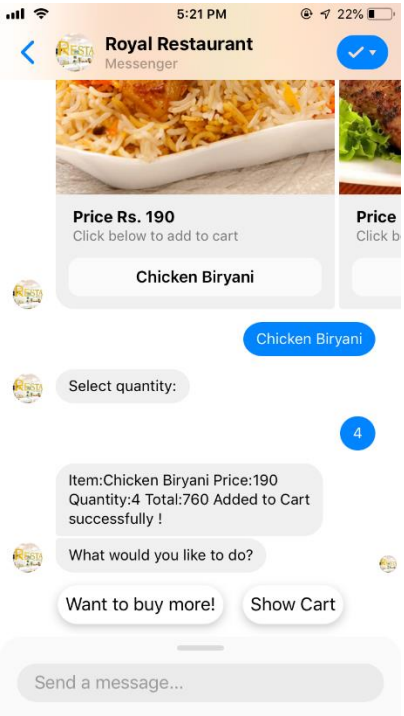
Now, this is the same scenario, as shown above, the user will again press the “Old Fashioned Lemonade” button and then he/she will choose the quantity to buy (as in this case, the user has selected 3). After this, he/she will be informed that his/her chosen item has been successfully added to the cart.

Again, the user will be left with two options: whether he/she “Want to buy some more items” or “he/she wants to have a look on his/her cart.



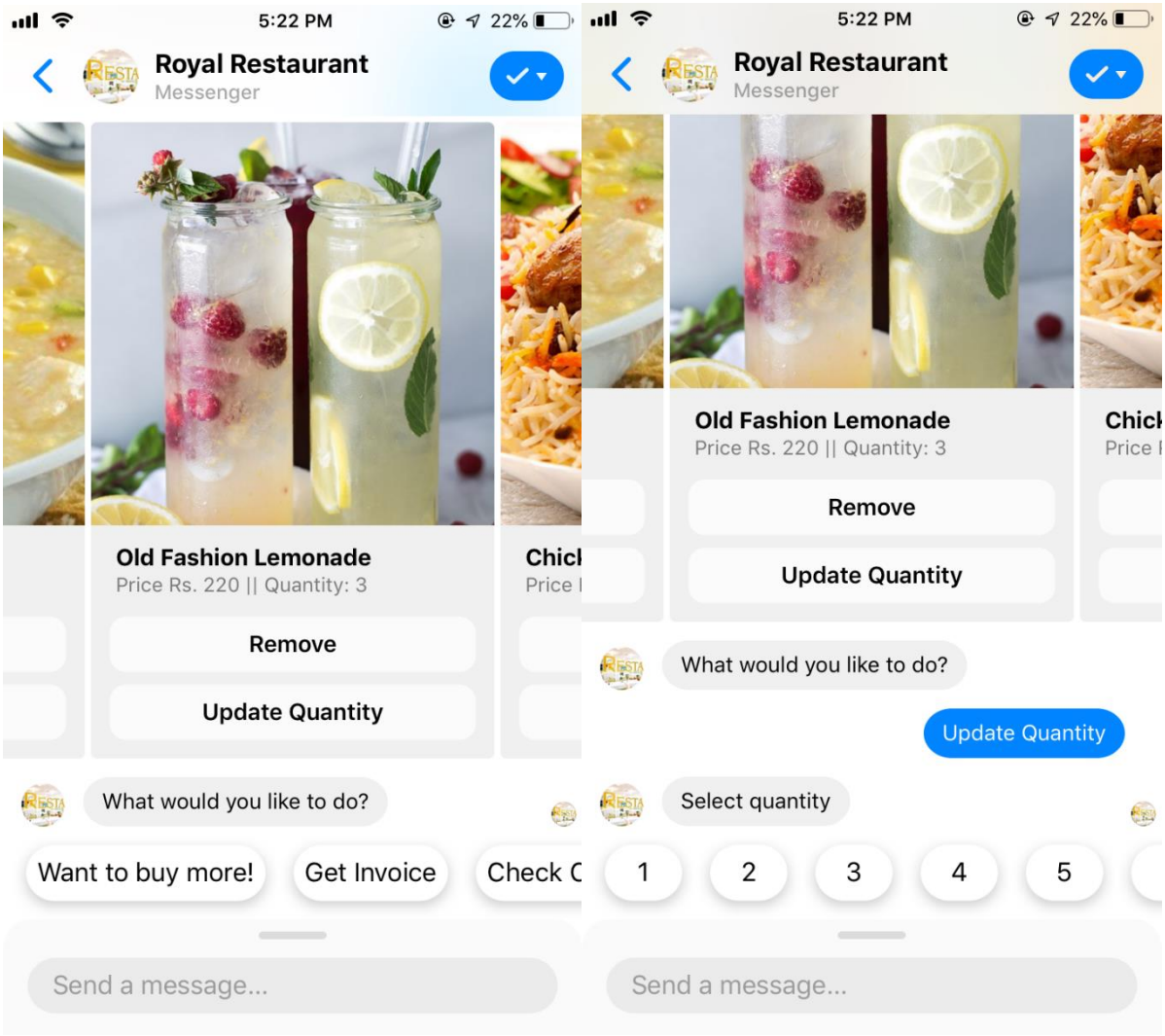






Again, this is the same scenario that user has pressed the “Want to buy more” button for the third time and he/she has been redirected to the main menu from where he chooses “Pakistani Food”

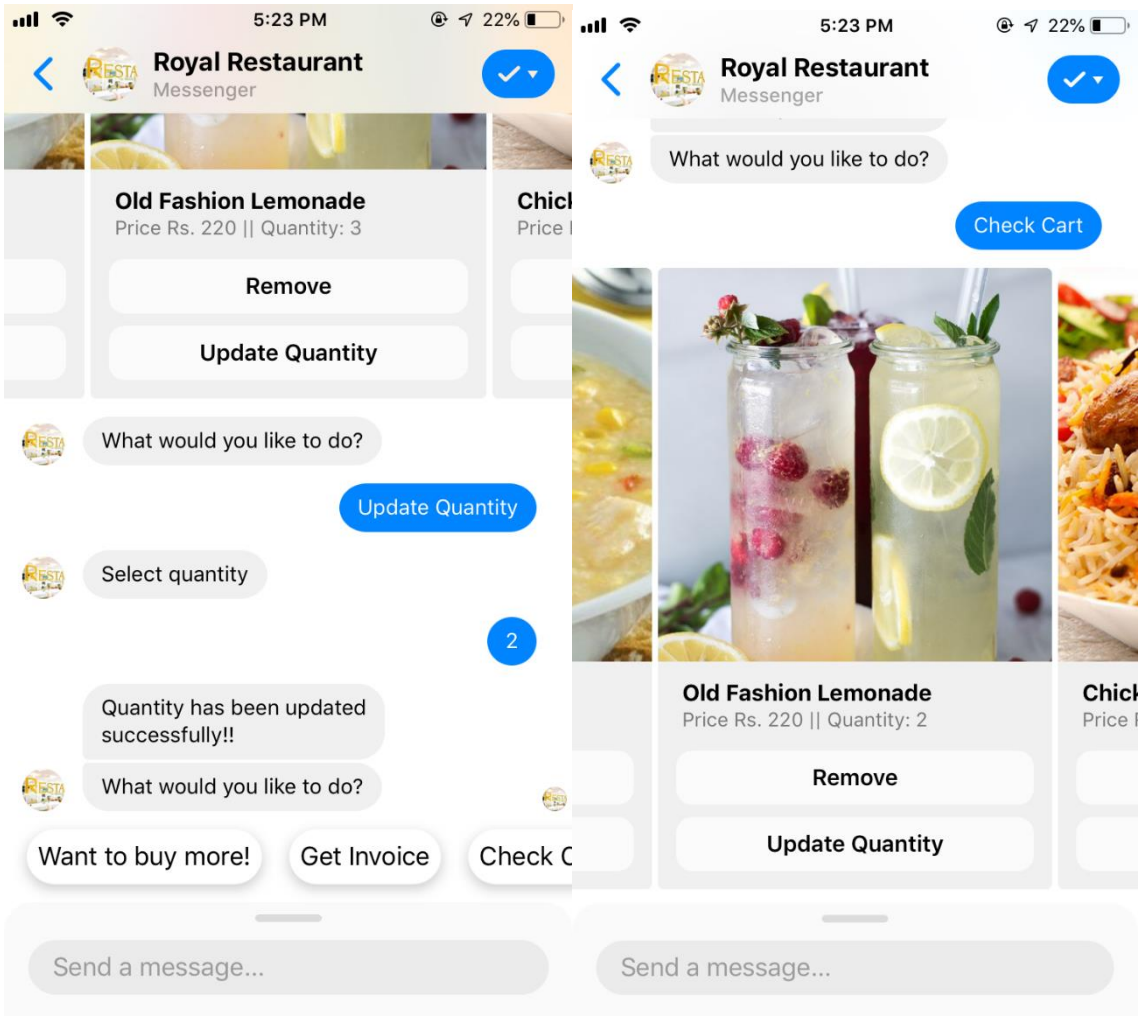
and the selected “Chicken Biryani” with the amount of 4 and then he/she got success message from the bot and again asked him/her with two same options.



Now, if the user will press the “Show Cart” Button. He/she will be redirected to his/her cart where the user can see his/her chosen items with their prices and selected quantity.

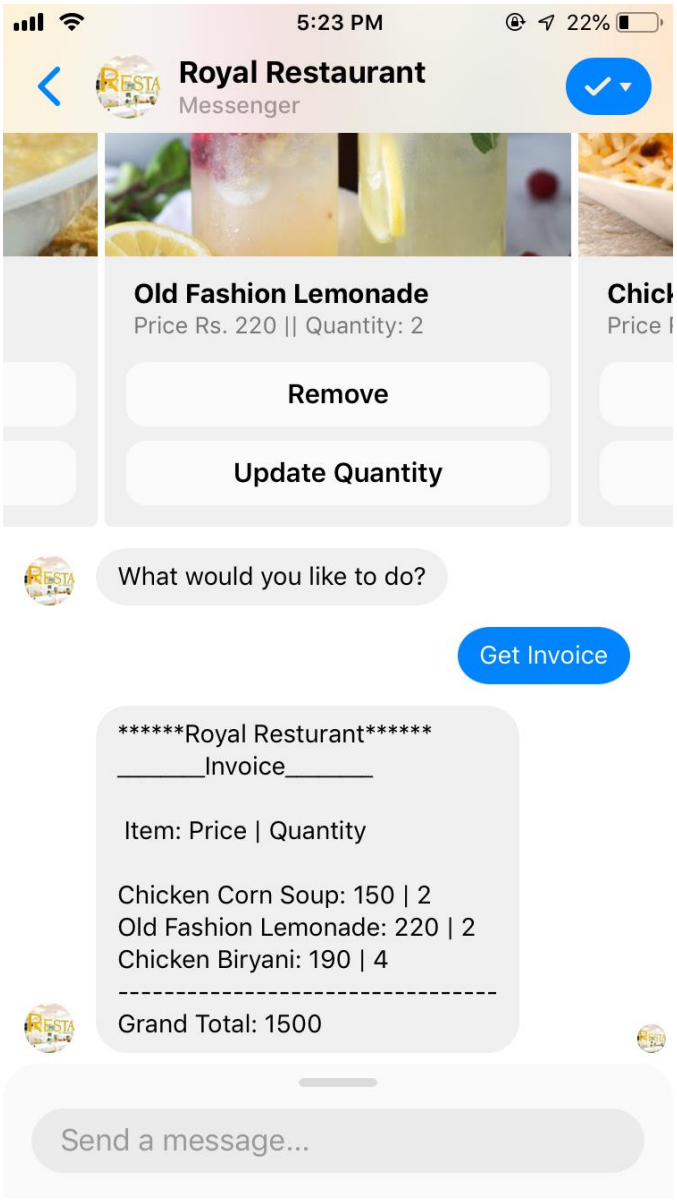
In cart, the user also has the option to edit his/her cart:

1) Users can remove any item from the cart.
2) Users can update the quantity of any item in the cart.
3) Users can also buy more items.



Now, users have requested to update the quantity of “Old Fashion Lemonade” and set it to 2 as shown in the left figure and the bot has displayed the message that “Quantity has been updated successfully!!”.

After updating the quantity, when the user clicks on the “Check Cart” button, the user will be again shown his/her cart with the updated quantity as you can see in the above right figure.



In the end, when the user presses the “Get Invoice” button, the user will get his/her invoice containing all the purchased items, their quantity, total prices of each item, and finally the “Grand Total” of the bill.

After getting the invoice, the user will get a farewell message almost after half an hour of getting the invoice.

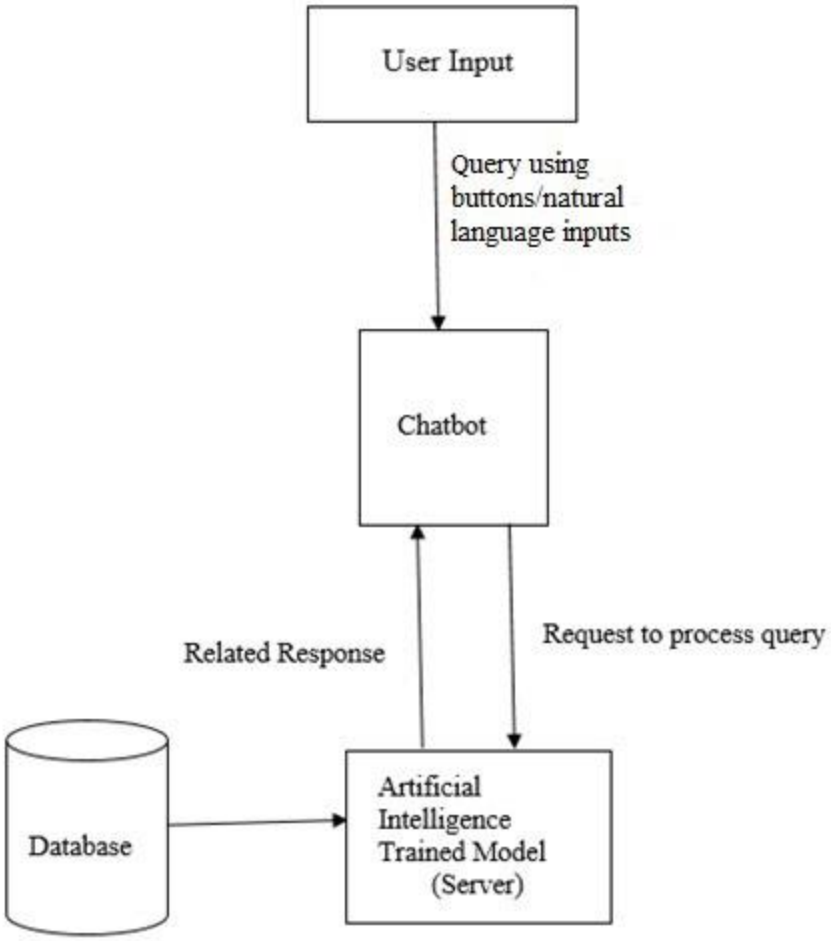
3.4. Proposed Chatbot for Travel Agency

We'll be targeting the travel industry domain when creating our chatbot. Travel agency domain, like many other domains, requires considerable staff to answer daily queries related to flights, tours, and rates, etc. which is time-consuming, expensive and leaves a margin of error. If the number of queries each day surpasses the time that can be utilized for answering them, the staff can no longer respond to them and therefore, is not a reasonable solution.

We shall use our chatbot to show how we can quicken the question-answer process and eliminate all of the issues mentioned above. As a chatbot answers simultaneously to each user's queries, there is no waiting, no delay in each response, and still, no staff required.

3.5. Chatbot Architecture

As we can see in Figure 1 the user will input a query using natural language, which shall be sent to the chatbot. The chatbot will send the request to our AI trained model. The model will match the user intent through entities and send the response back to our chatbot. The chatbot will further send the required response to the user. The AI database contains all the possible intents.



.

Figure 1: Travel Agency Chatbot Architecture Diagram

3.6. Conversational Flow Design

As the user opens the chat messenger, he/she receives the welcome message that includes greetings. As the user continues, he receives a request to pick the departure city. Then the user is asked to pick the destination city. Then the user is asked to select a date in a calendar. Then the user is asked to select the type of the ticket (return or one side) Scrolling further down shows the carousels based menu, in this menu Tickets, are separated according to their categories using the carousels and the buttons. A carousel has the name of its category on the top and its related

questions as the buttons at the bottom of the carousel. How the category of a carousel. As the user presses the details button the user will get that specific ticket type shown in different carousels. After that user will select one of the tickets he/she wants to book.

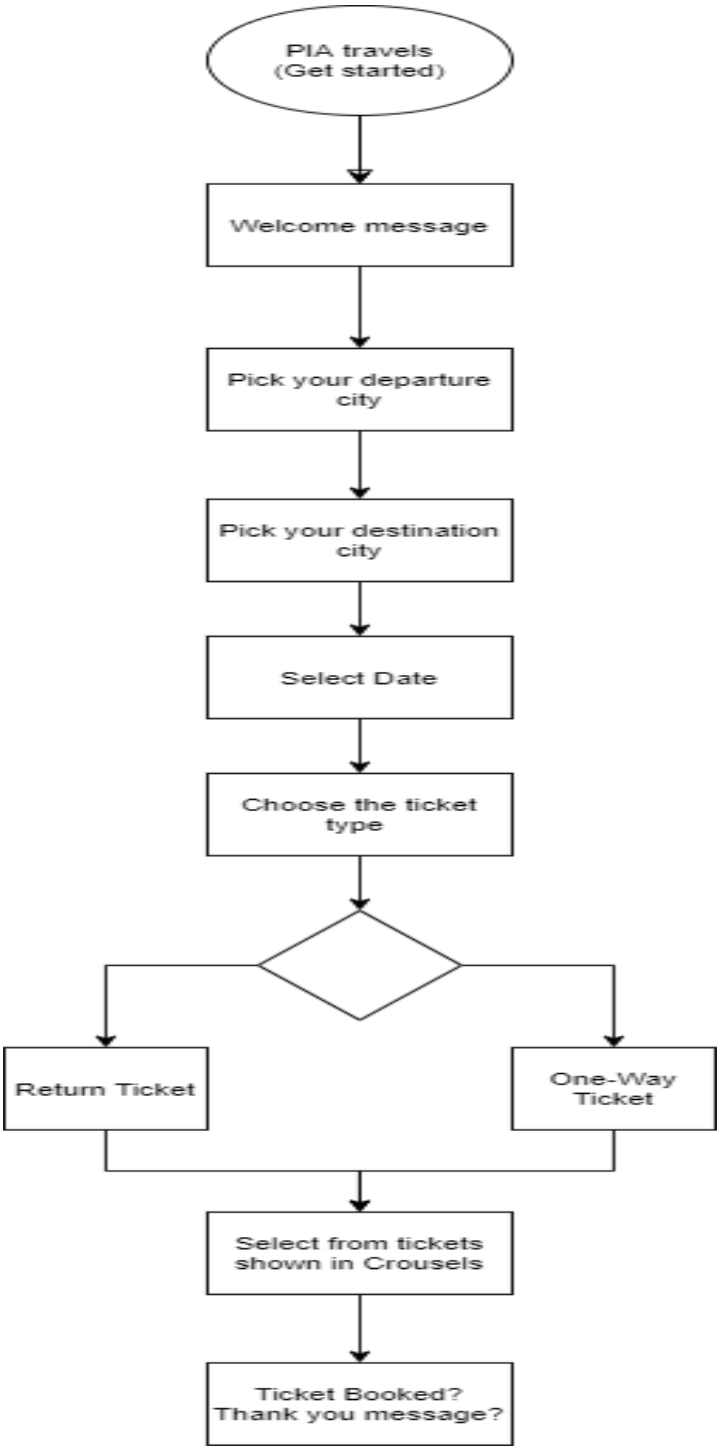


Figure 2: Conversational Flow Diagram

3.7. Development Procedure

Step 1:

Create a Facebook Application

It provides interface & APIs to interact through code.

1. Go to **developers.Facebook.com**
2. Register yourself then create an application.
3. Then choose an application from the messenger.

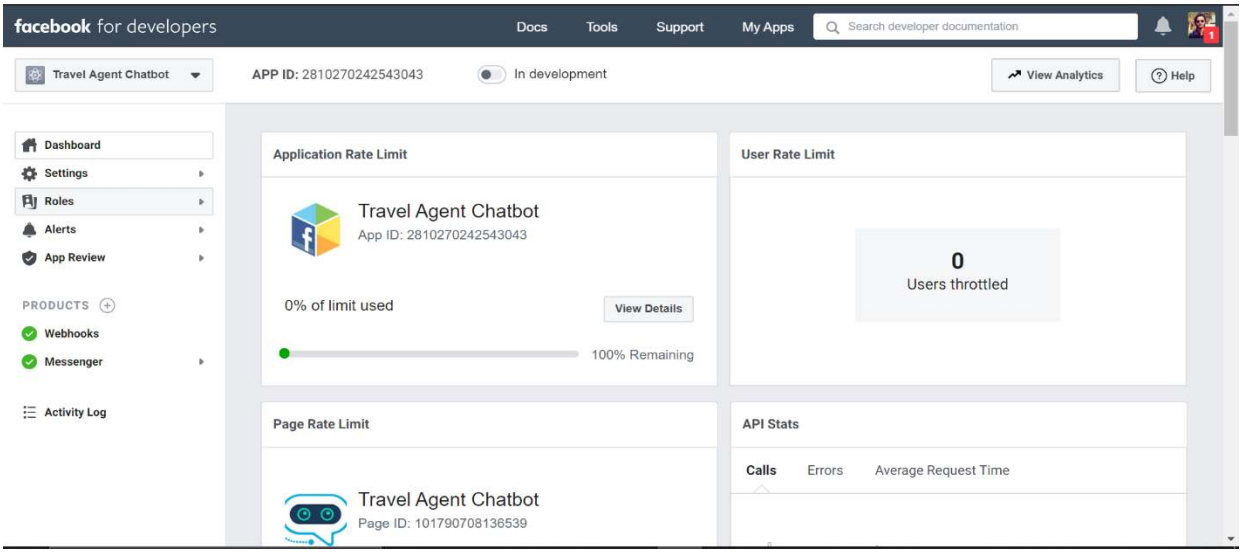


Figure 3. Creating Facebook Application

Step 2:

Create Facebook Page

Create the Facebook page & and give a test application to give authority to interact & control page.

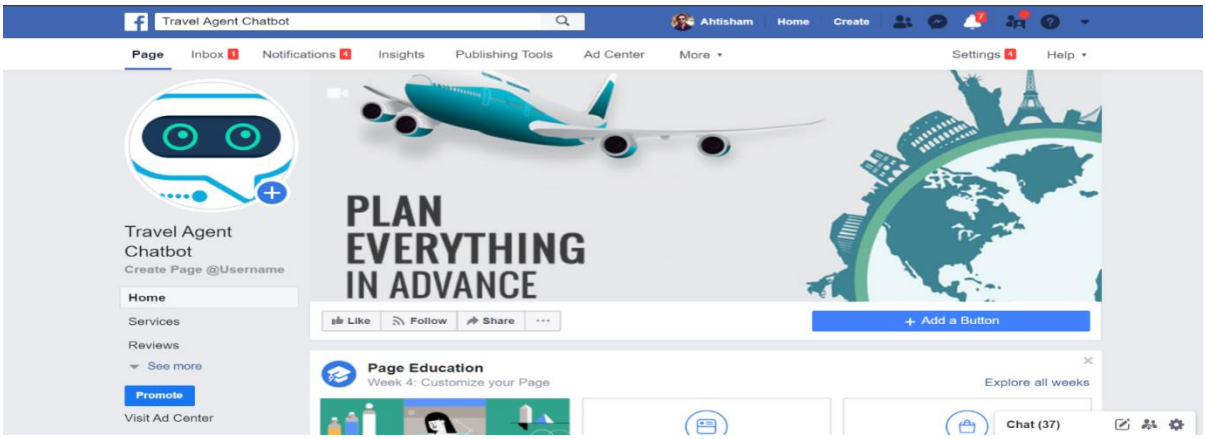


Figure 4. Creating Facebook Page

Step 3:

Setting Up the Project Folder

1. Setup the project folder for the messenger bot
2. Install python 3
3. Install virtualenv
4. Create a new project folder
5. Open the command prompt and open the virtualenv. It contains all the packages we need and to keep the global side packages clean and manageable. A folder is created. Now activate that in the command prompt. **Script\activate**

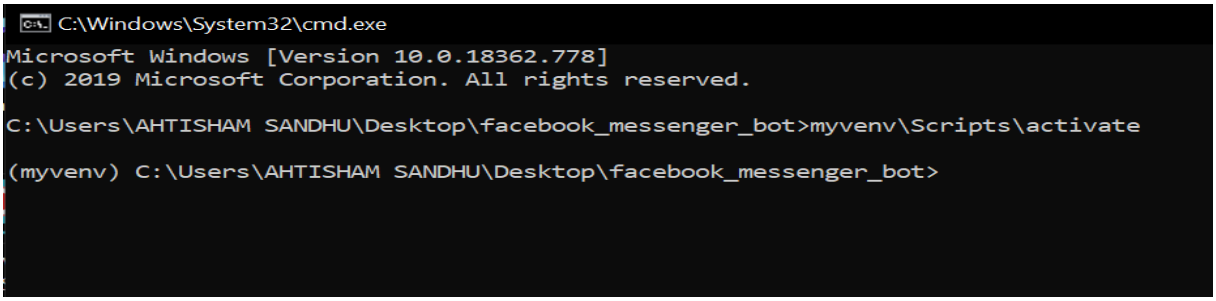


Figure 5. Setting up Project Folder

Step 4:

Install dependencies

1. Run the command Pip install flask requests **pymessenger** on command prompt. All of these will be installed in a few moments.

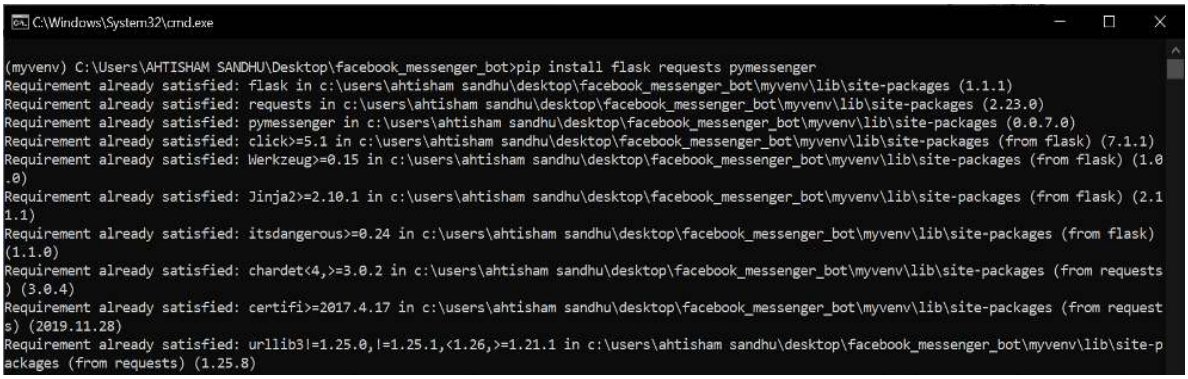


Figure 6. Installing dependencies

Step 5:

Setting a Webhook

It will send all the messages from the Facebook page to the Facebook application using some HTTP requests.

1. Verify the Webhook by using python code.
2. Initialize the flask application.
3. Facebook API will send the HTTP request.
4. Give the verification token as well.

Figure 7. Setting webhook

Step 6:

Installing Ngrok

Ngrok is the tunnel through which local servers will get exposed to the internet.

1. Download the Ngrok & run it on your computer.
2. Give it a port HTTP 80 through command prompt.

After this localhost server is now accessible on the URL on the internet.

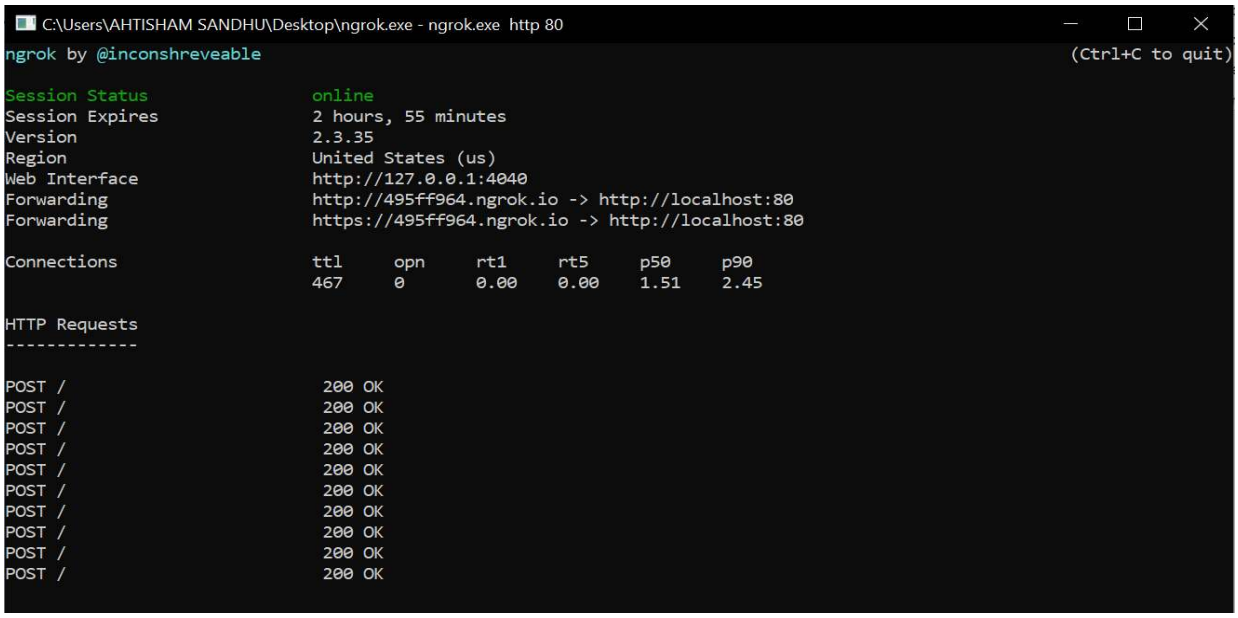


Figure 8. Installing Ngrok

Step 7:

Run the flask application.

1. In your application go to the products then choose messenger and then choose to set up the Webhook.
2. Give the callback URL & Verify Token which was given in the code.
3. Subscribe the Webhook to the page. Select your page & subscribe.

Now Webhook has been subscribed to the Facebook page.

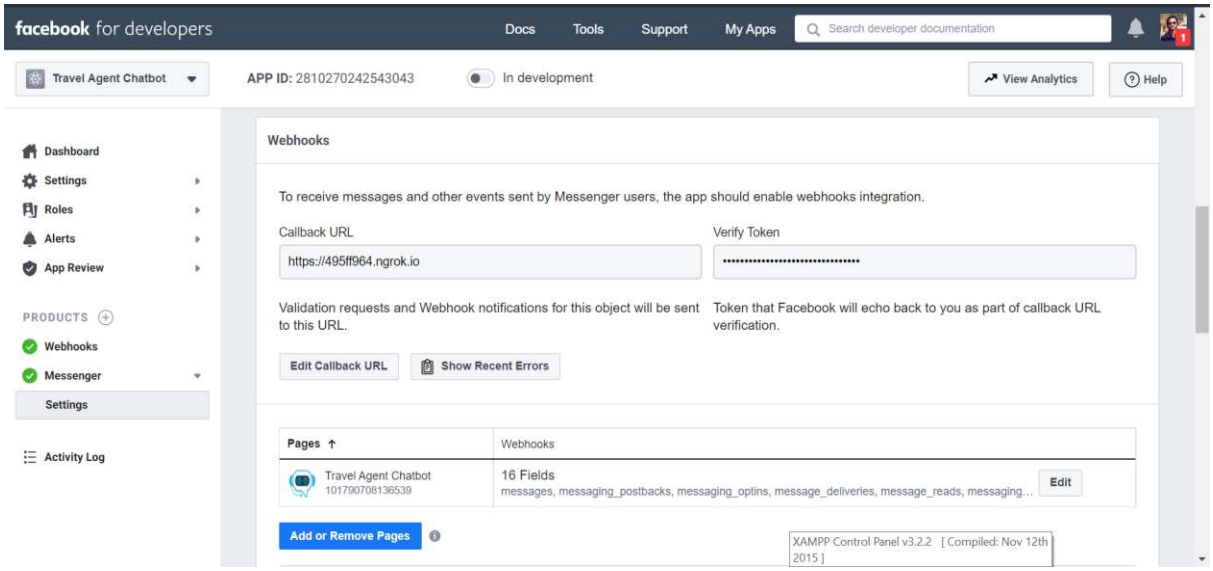


Figure 9. Running the Flask Application

Step 8:

Activating virtual environment

1. Activate virtual environment on command prompt by giving the command activate
2. Now the flask application would work on port 80.

Figure 10. Activating Virtual Environment

Step 9:

Receiving the Messages

Now we want to receive messages from Facebook messenger to the backend.

1. Add the HTTP Post request protocol in the code.
2. Create the login function in code which will print the message received from the messenger
3. In command prompt activate the Virtual environment again and then run the application.
4. Now whatever conversation we will have on messenger that would be displayed on the command prompt. Along with the sender and recipient ID and time stamps.

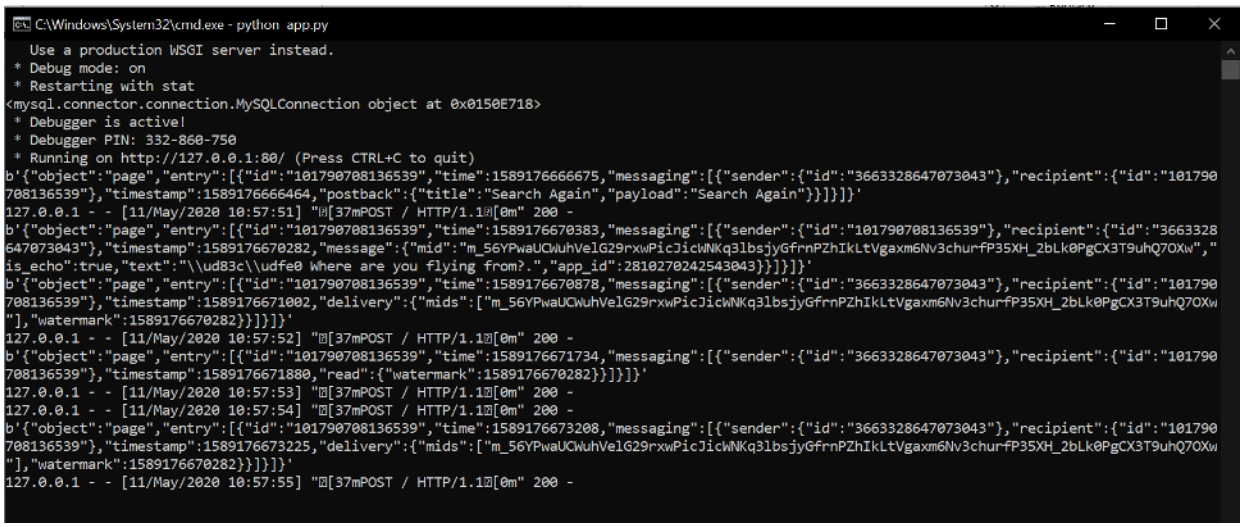


Figure 11. Receiving the Messages

Step 10:

Replying the Messages

1. To fetch the data from the messenger we will add the piece of the code having if and for loops.
2. Then if we open the command prompt where our messages were received we have the option there for replying.
3. To get useful data we can put the restriction in the code. We can put conditions on IDs of the senders and types of text.

4. Then we will create the object of the messenger in the code.
5. Then from Developers.Facebook.Com Copy the URL of the page and paste that in the code.
6. Now if we open the messenger and send a message on our page we will receive the same reply as the message sent because we passed the ‘eco’ command in the code.

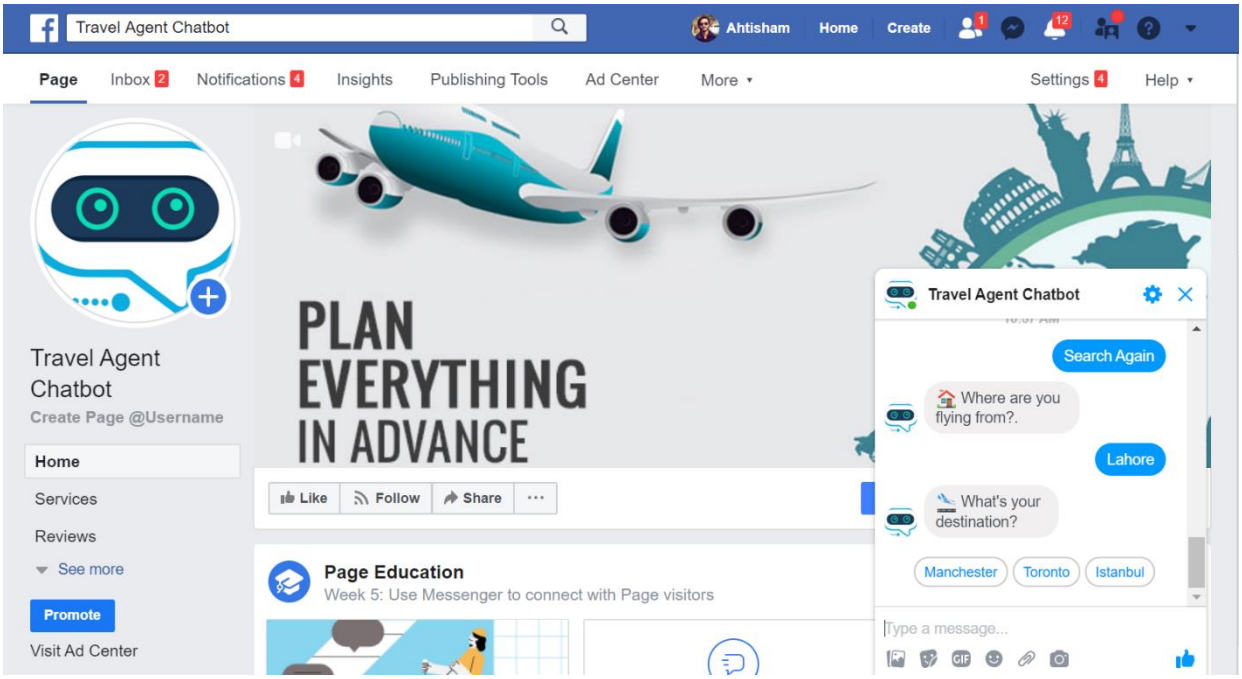


Figure 12. Replying the messages

Step 11:

Installing Heroku

Running Flask Application on the cloud. For that, we will need to set up Heroku.

1. Download Heroku online. It has CLI (Command Line Interface) which will allow us to interact with the website directly from the command prompt.
2. Download git CLI online.
3. To install Python 3 dependencies go the project folder which we created at the start and start installing them.
4. Install all the third-party libraries such as gunicorn that Flask Application would require

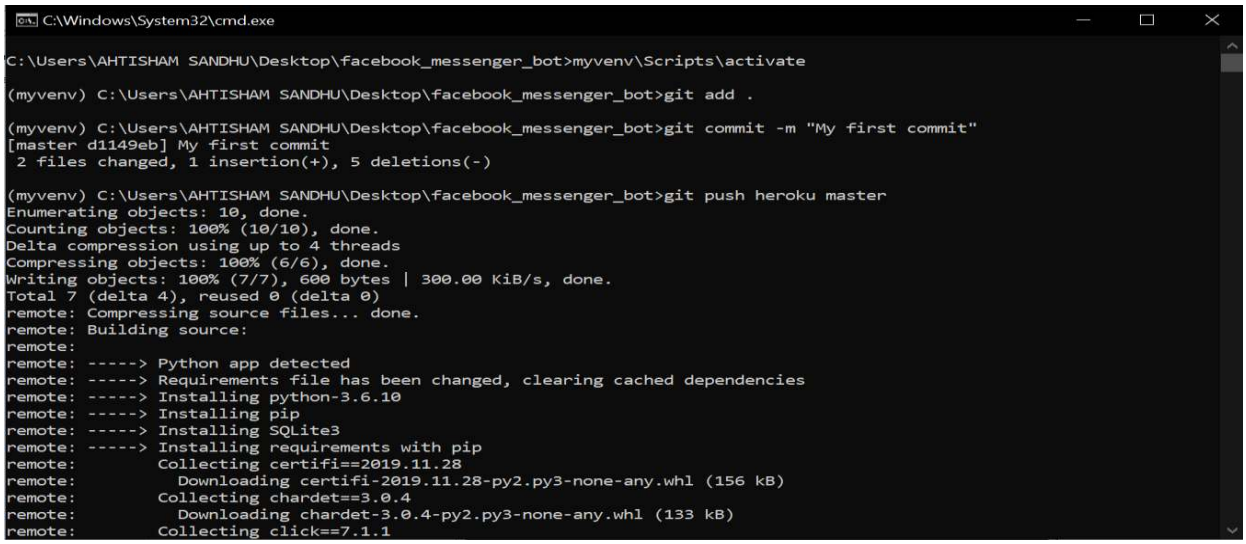


Figure 13. Installing Heroku

Step 12:

Deployment of Flask Application (Heroku Application)

1. Create a Heroku application which will contain all the files of the project. Give the command Create an app in the command prompt.
2. Now to tell git repository what is the URL of the Heroku application. Paste the Heroku application repository.
3. After that, all the requirements would be installed in the Heroku application.
4. And then the application would be deployed. (Figure a)
5. Open the Heroko portal online it will show us how many applications we have on Heroku. (Figure b)
6. Now copy the Heroku application URL and paste that in the Webhook of the Facebook page. Verify and save that. (Figure c)
7. Now if we check that on Facebook Messenger after deployment on Heroku it would still show the correct results. (Figure d)

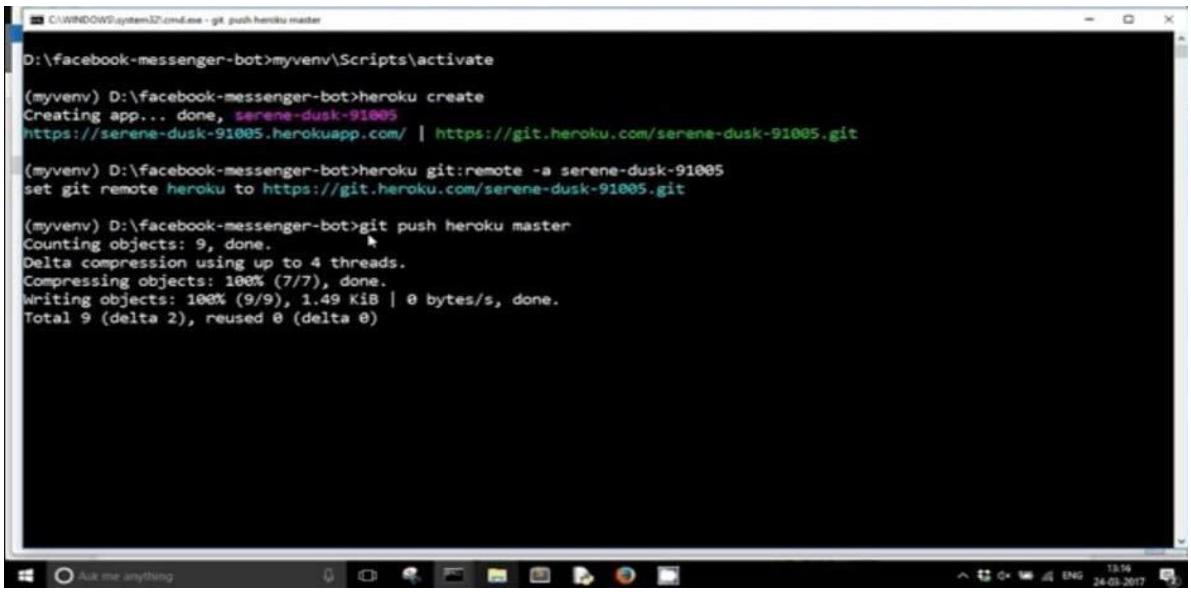


Figure 14.a

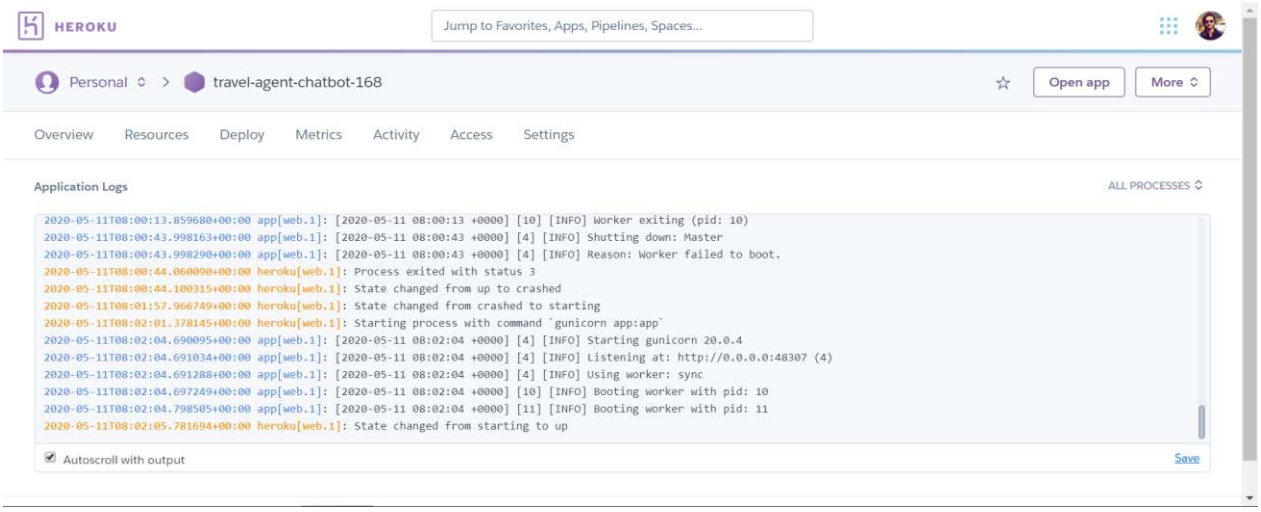


Figure 14.b

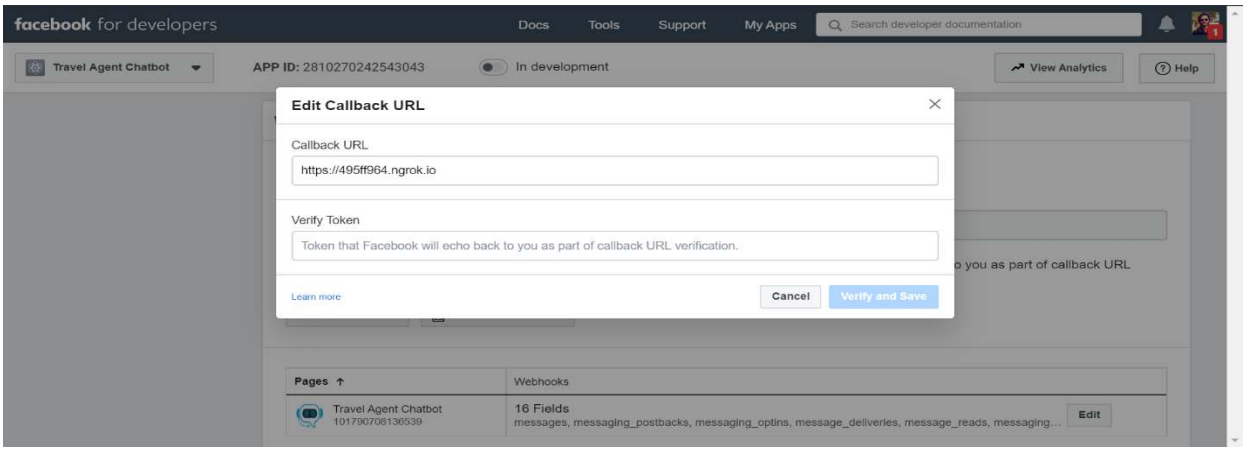


Figure 14.c

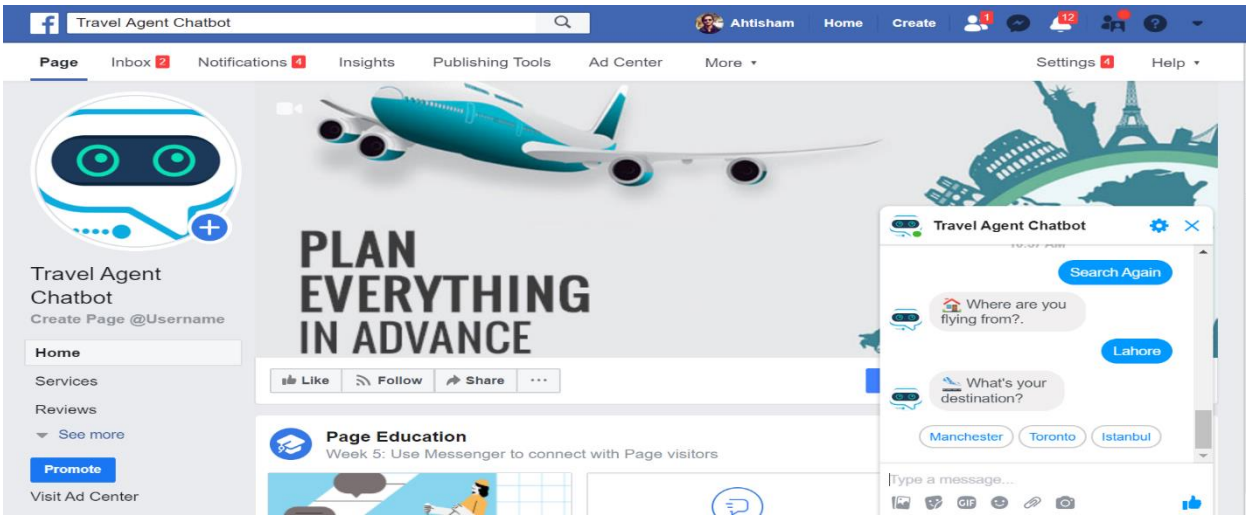
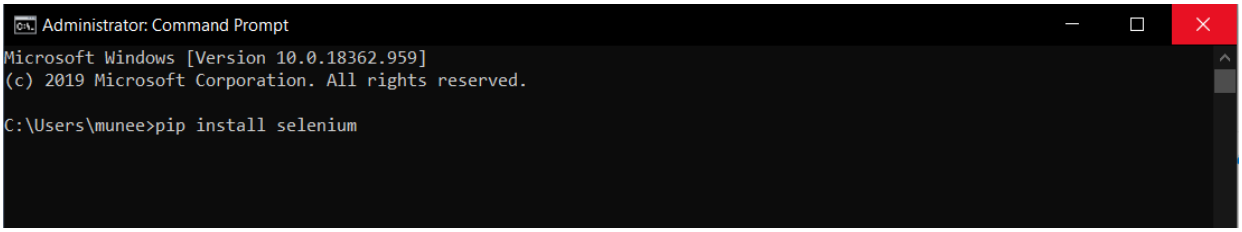


Figure 14.d

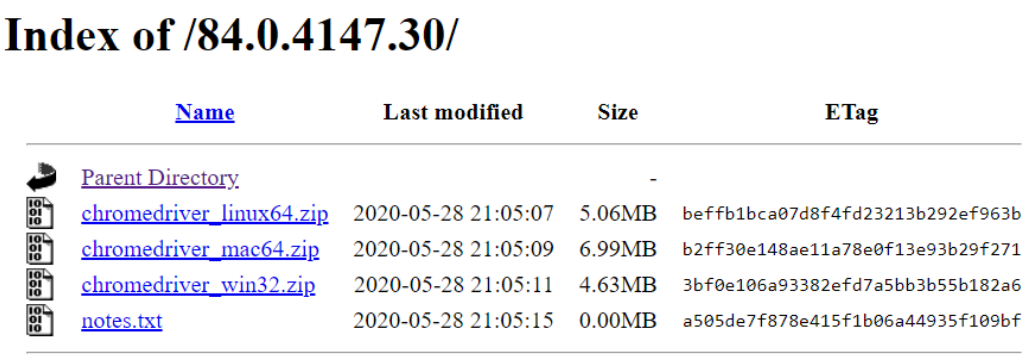
Step 13:

Working of Web Scraper

1. Download Selenium using pip command in command prompt



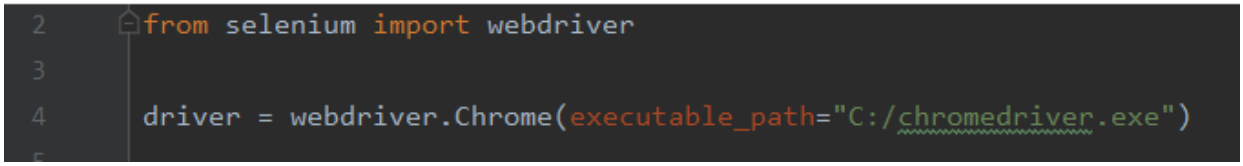
2. Download Chrome driver which is compatible with your Google Chrome from this link https://chromedriver.storage.googleapis.com/index.html?path=84.0.4147.30/



Index of /84.0.4147.30/

Name	Last modified	Size	ETag
Parent Directory		-	
chromedriver_linux64.zip	2020-05-28 21:05:07	5.06MB	beffb1bca07d8f4fd23213b292ef963b
chromedriver_mac64.zip	2020-05-28 21:05:09	6.99MB	b2ff30e148ae11a78e0f13e93b29f271
chromedriver_win32.zip	2020-05-28 21:05:11	4.63MB	3bf0e106a93382efd7a5bb3b55b182a6
notes.txt	2020-05-28 21:05:15	0.00MB	a505de7f878e415f1b06a44935f109bf

3. Create a Python file. Import Selenium and Web Driver and set chrome driver path in Selenium Webdriver



```
from selenium import webdriver

driver = webdriver.Chrome(executable_path="C:/chromedriver.exe")
```

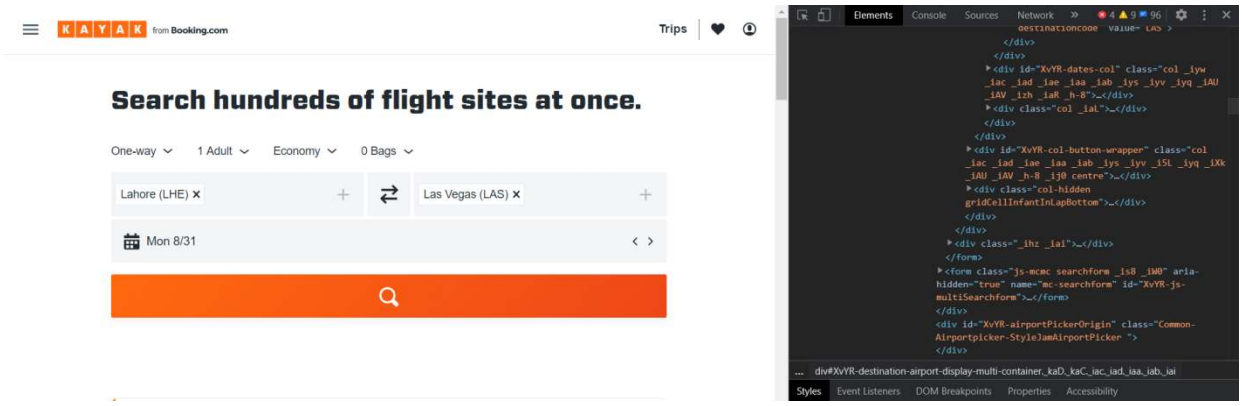
4. We got a list of 229 cities around the world

```
Karachi - Jinnah Intl. (KHI)
Bahawalpur (BHV)
Chitral (CJL)
Dalbandin (DBA)
Dera Ghazi Khan Intl. (DEA)
Dera Ismail Khan (DSK)
Faisalabad Intl. (LYP)
Gilgit (GIL)
Gwadar Intl. (GWD)
Islamabad Intl. (ISB)
Lahore - Allama Iqbal Intl. (LHE)
Moenjodaro (MJD)
Multan Intl. (MUX)
Nawabshah Airport (WNS)
Panjgur (PJG)
Peshawar - Bacha Khan Intl. (PEW)
Quetta Intl. (UET)
Rahim Yar Khan - Shaikh Zayed Intl. (RYK)
Sialkot Intl. (SKT)
Skardu (KDU)
Sukkur (SKZ)
Turbat Intl. (TUK)
Zhob (PZH)
```

5. Get the list of all the 229 cities and store them as Departure and Arrival cities and the status of the ticket

```
file1 = open('airports.txt', 'r')
departure_cities = file1.readlines()
file2 = open('airports.txt', 'r')
arrival_cities = file2.readlines()
status = ['one way', 'return']
```

6. Get all the Xpaths of the form to fill values and search for the flights according to logic



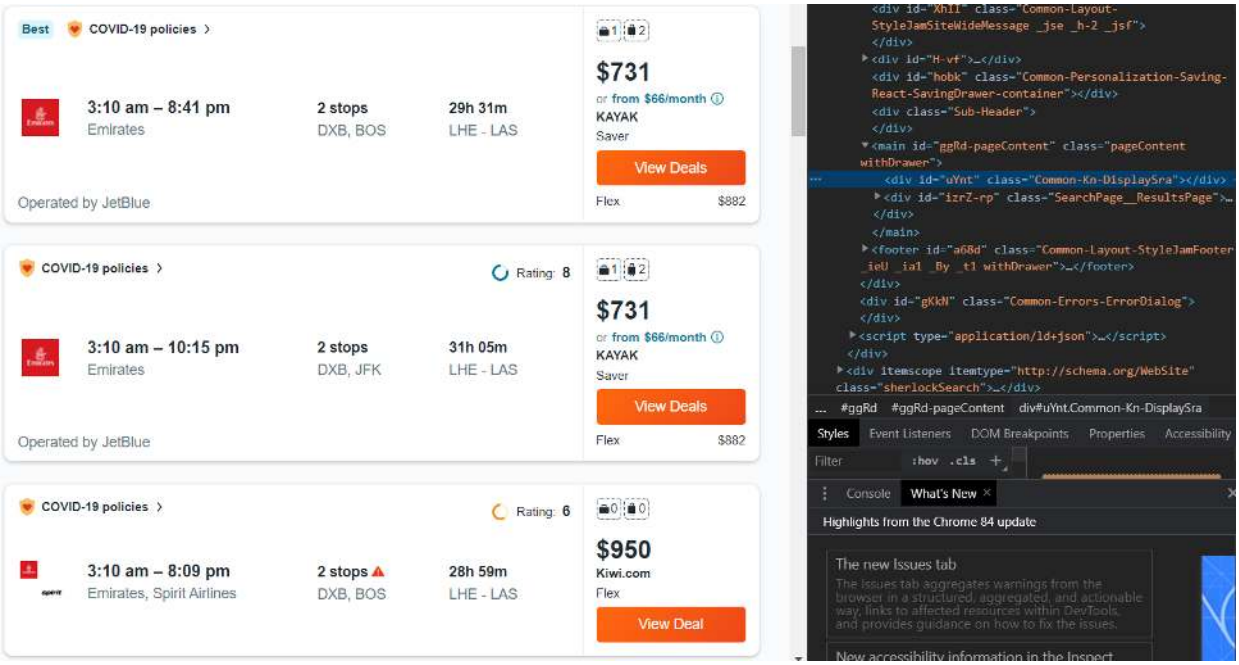
7. Find the Element with help of driver and perform functions on them, like inserting departure city, entering time and click the search button

```
for departure_city in departure_cities:
    for stat in status:
        for arrival_city in arrival_cities:
            for date in dates:
                driver.get('https://www.kayak.com/flights')

                if departure_city != arrival_city:
                    if stat == 'return':
                        driver.find_element_by_xpath('//*[@id="zHrs-switch-display"]').click()
                        driver.find_element_by_xpath('//*[@id="zHrs-switch-option-1"]').click()
                    driver.find_element_by_xpath('//*[@id="tG-U-origin-code"]').send_keys(departure_city)
                    driver.find_element_by_xpath('//*[@id="tG-U-destination-code"]').send_keys(arrival_city)
                    driver.find_element_by_xpath('//*[@id="tG-U-dateRangeInput-display-start-inner"]').click()
                    driver.find_element_by_xpath('//*[@id="tG-U-depart"]').send_keys(date)
                    driver.find_element_by_xpath('//*[@id="tG-U-submit"]').click()

                    time.sleep(5)
```

8. Get the values from the result shown for every combination of the airports by getting Xpaths and extracting through the driver



9. Now after finally getting the value save it in the Database

```
import mysql.connector

mydb = mysql.connector.connect(
  host="localhost",
  user="myusername",
  password="mypassword",
  database="mydatabase"
)

mycursor = mydb.cursor()

sql = 'INSERT INTO `flight_info`(`id`, `departure_city`, `destination_city`, `departure_date`, `departure_day`, `ticket_type`, `ticket_class`, `d
val = ('Islamabad','Manchester','2020-05-21','Thu','Return','Economy','12:45 PM','PK-773',"Travel duration: 9 hour(s) (non-stop)",'189,940.00')
mycursor.execute(sql, val)
mydb.commit()
print(mycursor.rowcount, "record inserted.")
```

10. Sample of some data

0	Islamabad	Manchester	2020-05-21	Thu	Return	Economy	12:45 PM	PK-773	Travel duration: 9 hour(s) (non-stop)
1	Lahore	Manchester	2020-05-18	Mon	One Way	Economy	12:30 PM	PK-709	Travel duration: 9 hour(s) 30 minute(s) (non-stop)
2	Lahore	Istanbul	2020-05-18	Mon	One Way	Economy	02:30 PM	PK-108	Travel duration: 4 hour(s) 30 minute(s) (non-stop)
3	Lahore	Toronto	2020-05-18	Mon	One Way	Economy	04:30 PM	PK-888	Travel duration: 16 hour(s) 30 minute(s) (non-stop...
31	Lahore	Manchester	2020-05-18	Mon	Return	Economy	12:30 PM	PK-709	Travel duration: 9 hour(s) 30 minute(s) (non-stop)
32	Lahore	Istanbul	2020-05-18	Mon	Return	Economy	02:30 PM	PK-108	Travel duration: 4 hour(s) 30 minute(s) (non-stop)
33	Lahore	Toronto	2020-05-18	Mon	Return	Economy	04:30 PM	PK-888	Travel duration: 16 hour(s) 30 minute(s) (non-stop...
34	Lahore	Manchester	2020-05-19	Tue	One Way	Economy	12:45 PM	PK-719	Travel duration: 9 hour(s) (non-stop)
35	Lahore	Istanbul	2020-05-19	Tue	One Way	Economy	05:30 PM	PK-111	Travel duration: 4 hour(s) 30 minute(s) (non-stop)
									Travel duration: 16

Step 14:

Implementation of NLP

Buttons and carousels are the solutions to many of the user's queries, but it is not only the solution to all the user's queries. In many cases, the buttons prove to be inadequate as users may have queries not added to the button. To cope with the queries of all users almost completely, we would need a lot of carousels and, therefore, a huge variety of buttons. But this approach has its consensus because the user cannot find the button for his query and may be disappointed.

That's why we use AI to train our chatbot to understand the purpose of text-based user inputs. Our chatbot needs to be trained enough to understand and sort out users' queries so that the AI can get the best response.

Even after AI is implemented to respond to user queries, the buttons are used with it. Buttons are still a valuable part of most chatbots. So, we have a hybrid chatbot (button-based and AI-based). NLP based chatbot uses an AI technology--Natural Language Processing (NLP) to map user input (only text input) to an intent. After user input language analysis is done to get an answer. The NLP model here analyzes the conversation as a whole as opposed to only the word by word input

from a customer. We are using the wit.ai model to train our chatbot and empower it with some NLP capabilities. **Wit**.**ai** is an open-source chatbot framework with advanced NLP which is also known as natural language processing, capabilities which are owned by Facebook, **Wit**.**ai** is a popular choice for Facebook Messenger bots powered by NLP. **Wit**.**ai** I used to build intelligent chatbots for social channels, mobile apps, websites, and IoT devices.

First of all, we need to make an account on wit.ai and then we can start training our model by creating an app. Once, the app will be created, we will be able to train our chatbot for NLP related queries, by providing a dataset as intent and intentions, on which basis our model will be trained accordingly.

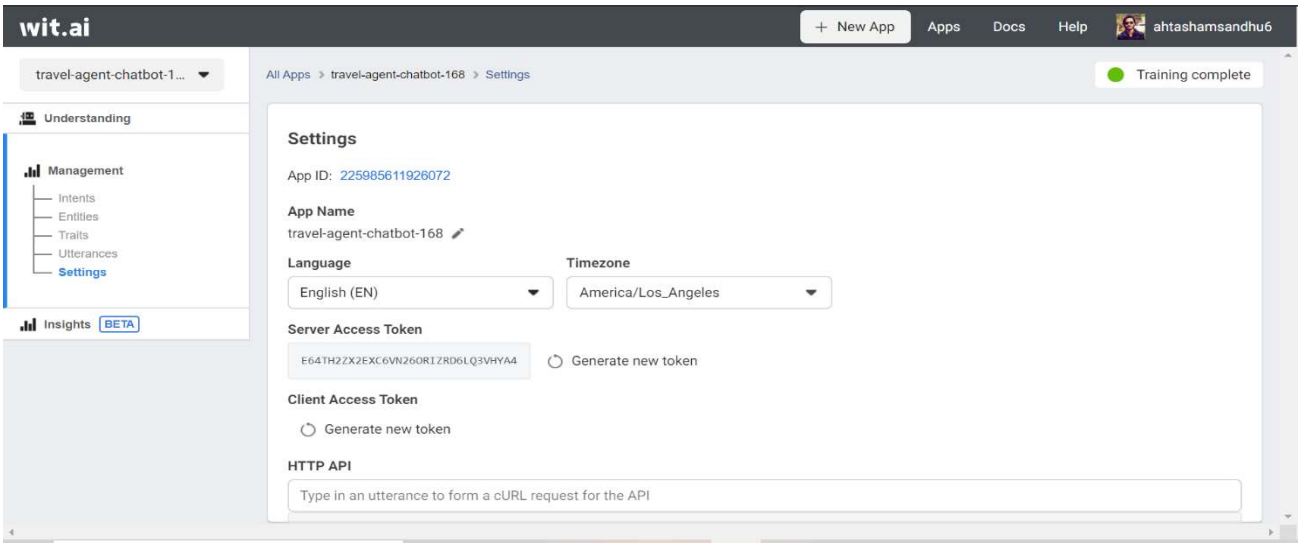


Figure 15.a

Once, our model will be fully trained, our chatbot will be able to process user text-based queries and get intent and entity value from the text to process them and to generate a correct response against that query. We can get that specific information by using URL in which we will pass the text message of the user and the answer will be returned in the form of **JSON** response. We can extract valuable information from that after doing some manipulations

```
{
   "text": "I want to go to turkey",
   "intents": [
      {
         "id": "2669758869933158",
         "name": "arrival-location",
         "confidence": 0.9974
      }
   ],
   "entities": {
      "wit$location:location": [
         {
            "id": "700938947419206",
            "name": "wit$location",
            "role": "location",
            "start": 16,
            "end": 22,
            "body": "turkey",
            "confidence": 0.9335,
            "entities": [

            ],
            "resolved": {
               "values": [
                  {
                     "name": "Turkey",
                     "domain": "country",
                     "coords": {
                        "lat": 39,
                        "long": 35
                     },
                     "timezone": "Europe/Istanbul",
                     "external": {
                        "geonames": "298795",
                        "wikidata": "Q43",
                        "wikipedia": "Turkey"
                     },
                     "attributes": {

                     }
                  }
               ]
            },
```

Here is a python script to interact with the trained wit.ai model and extract particular information that is needed. We pass user text-based queries to our trained model and get JSON response and then extract meaningful information that is the intent of user and entity value, then generate a specific response against that query and sent back to the user.

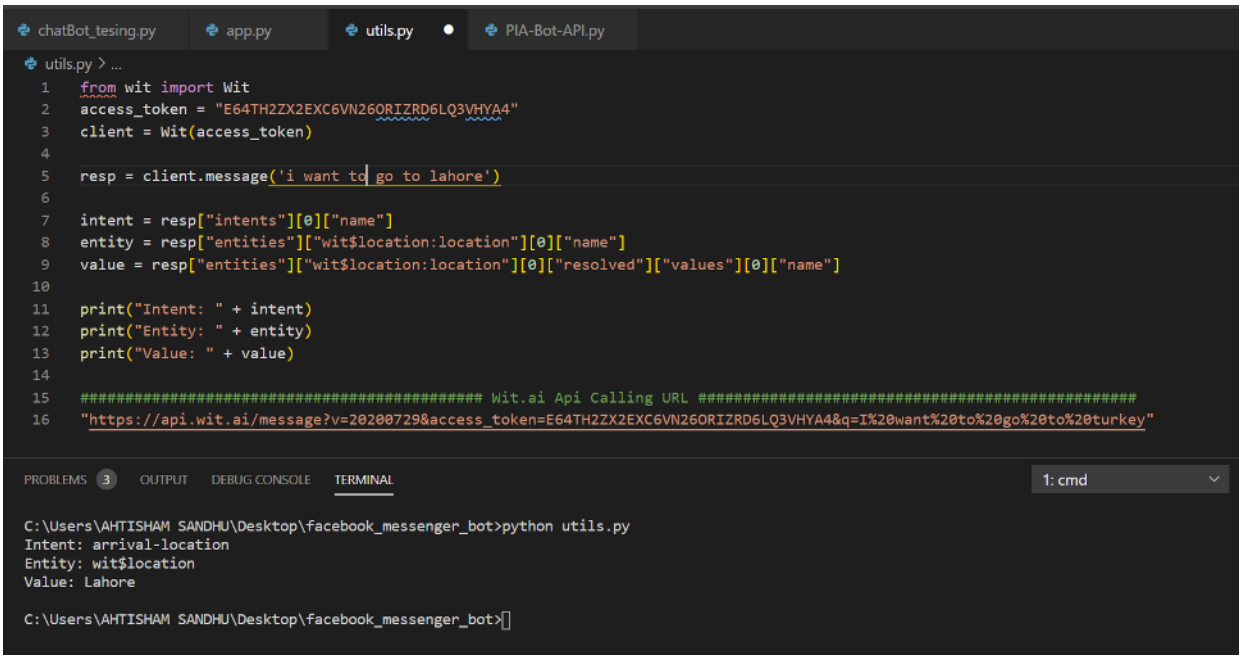
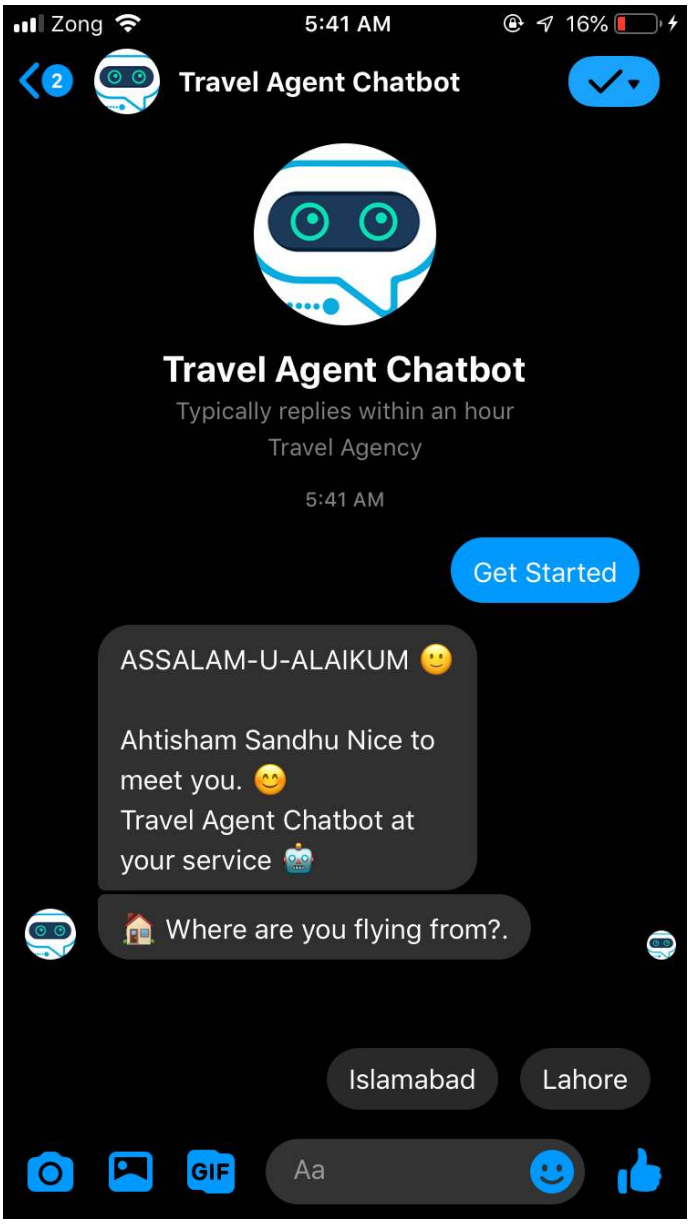
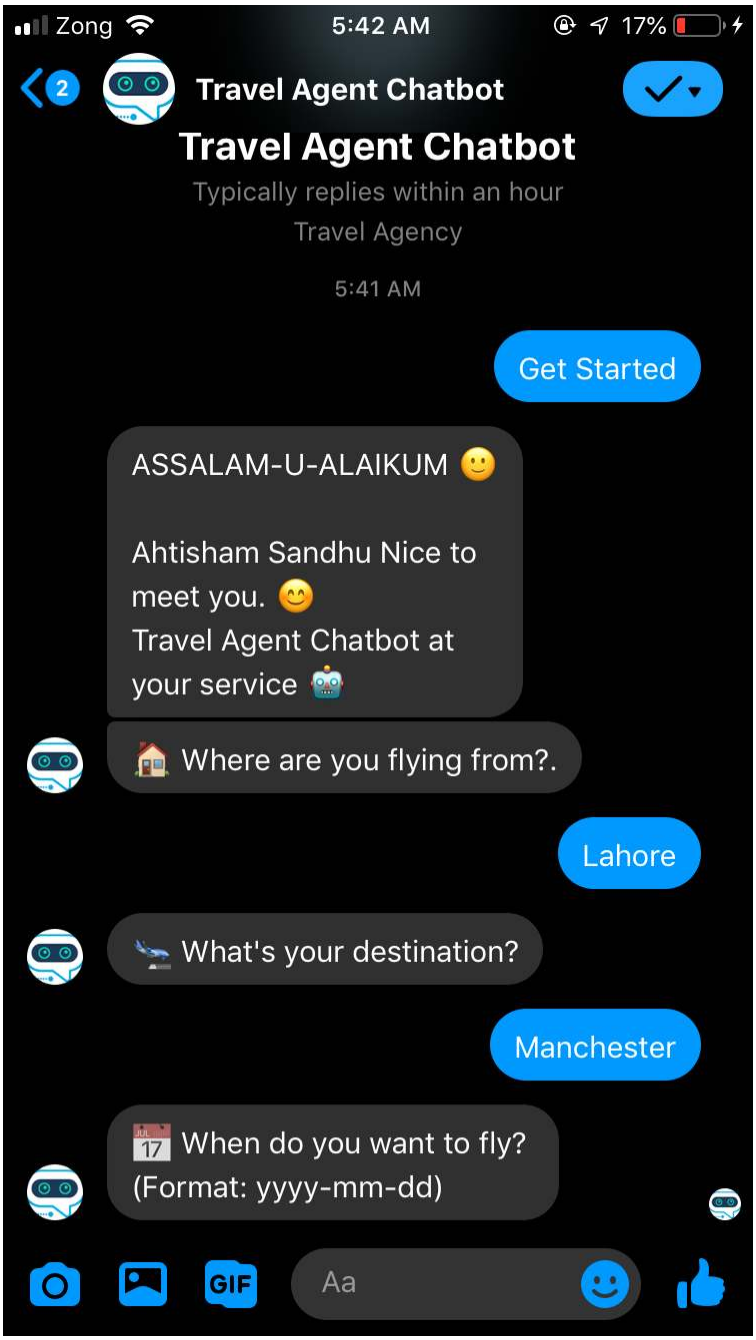


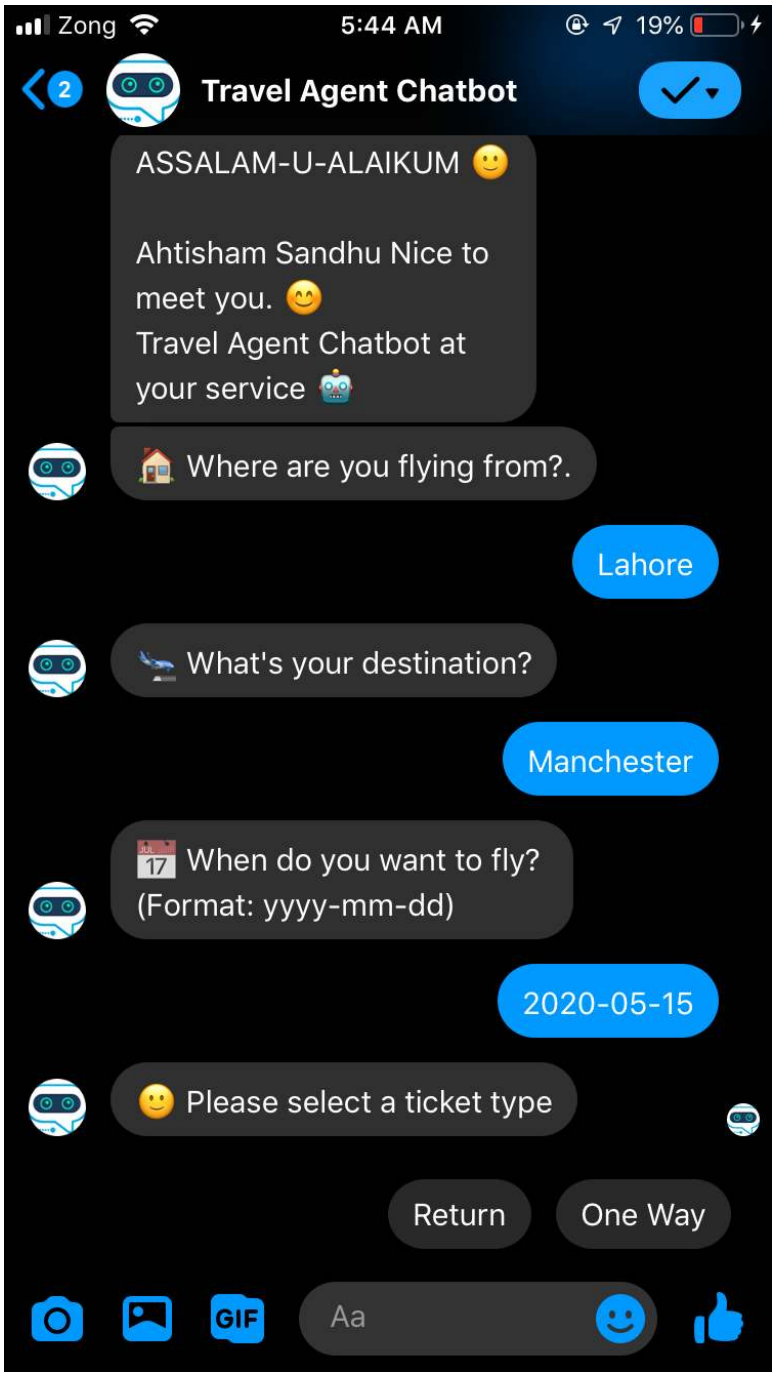
Figure 15.b

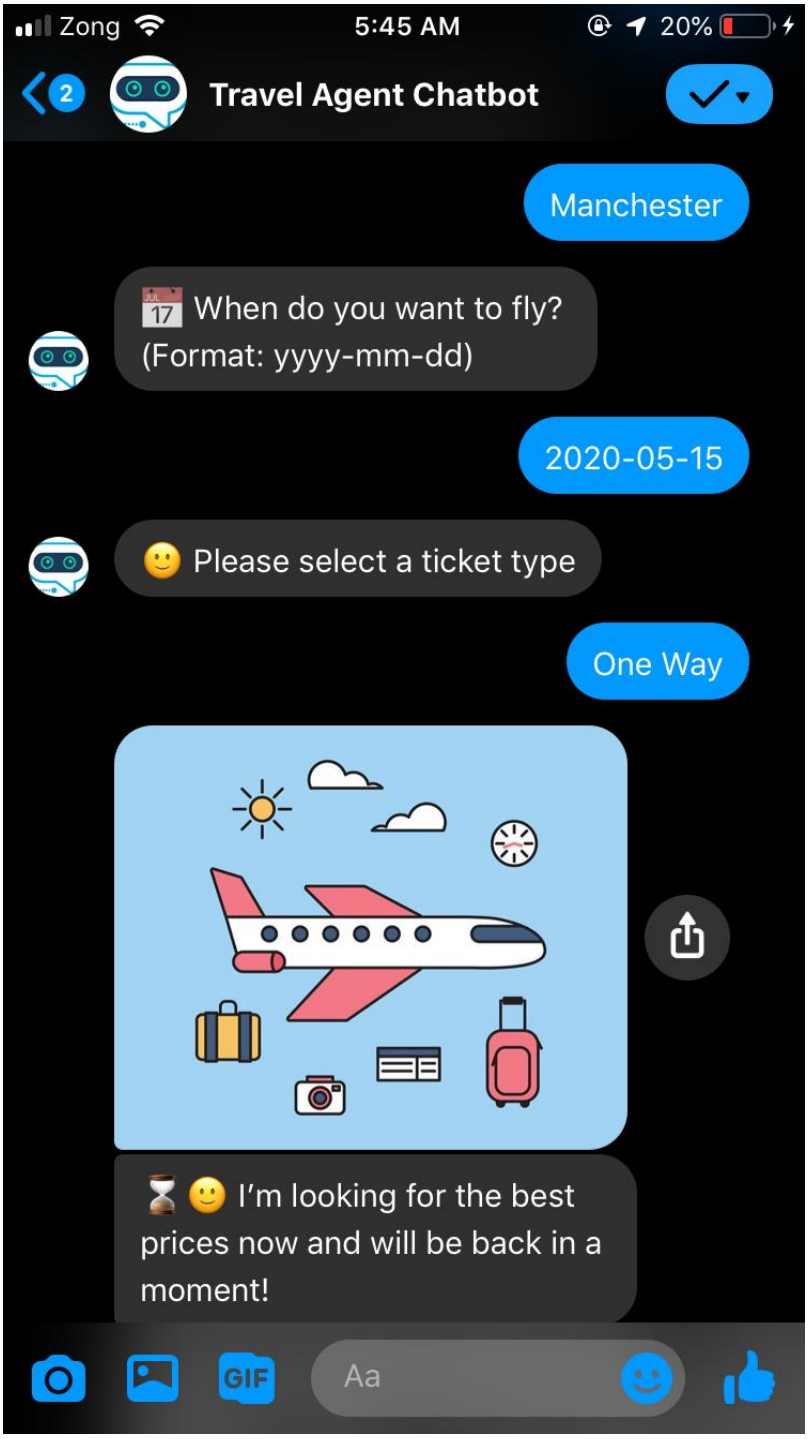
Basically, we use the access token to interact with the specific app that we create in wit.ai and we use the client.message() function of python to sent user query to our trained model and then model in return generate JSON response for us, by which we extract needed info like “What is the intention of a user?” or “What exactly he wants?” and “What’s Entity Value” and then our chatbot generates a particular response for our user based on different evaluations and conditions that we wrote in our script.

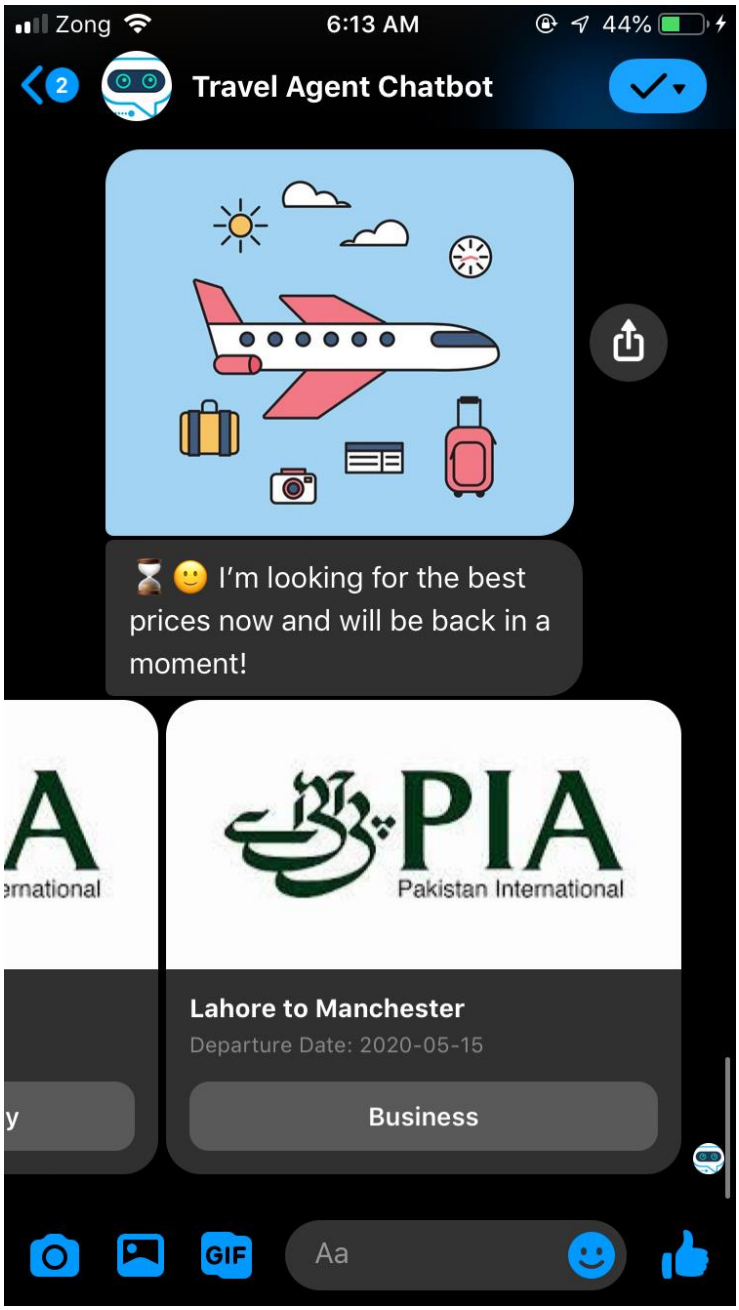
3.8. Working Chatbot of Travel Agency

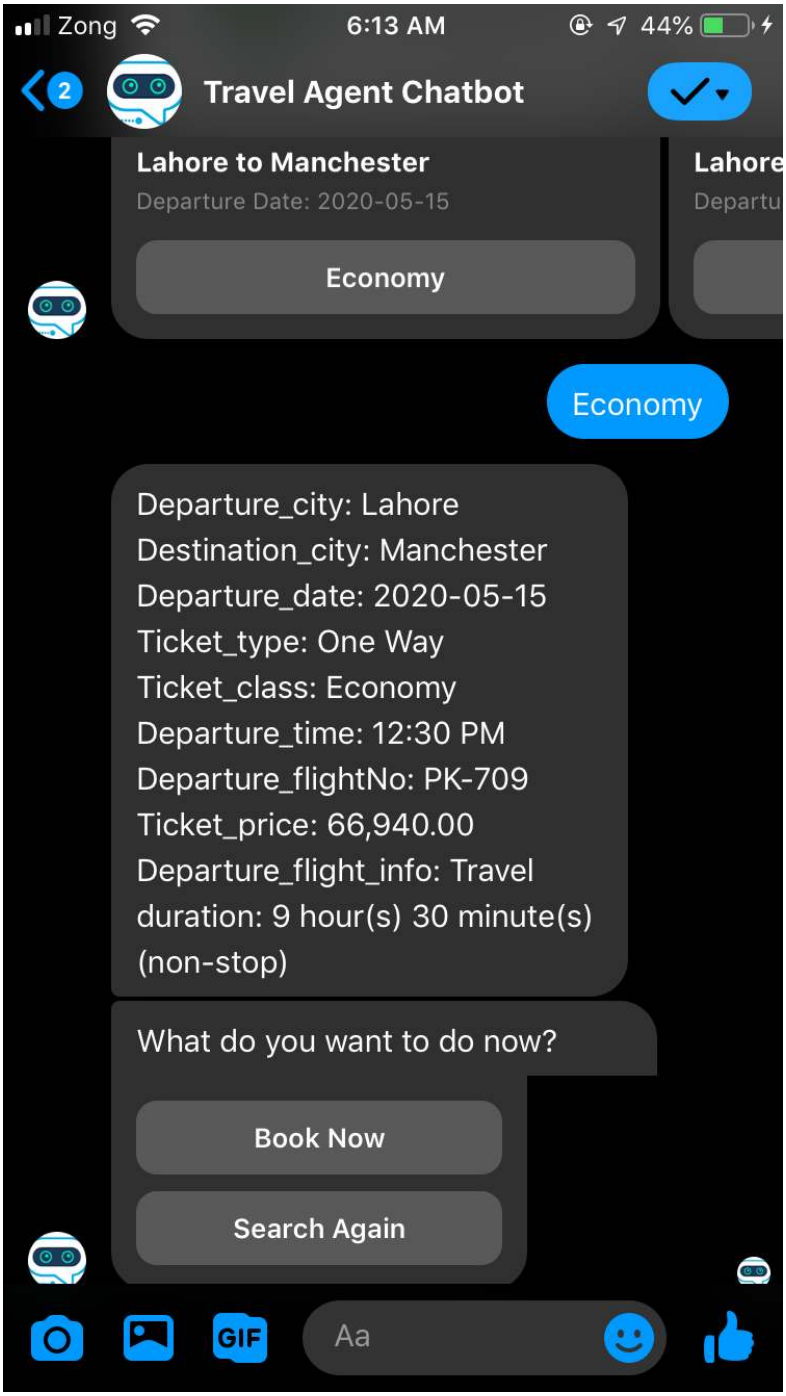












3.9. Marketing

Nowadays, everything is going digital. Messenger is a platform used by more users than any other social media applications. We can use our Chatbot for marketing purposes by broadcasting our sales, special events, or deals. We can use our Chatbot to generate traffic to the travel web page and cover the digital traffic as well for the orders. Our Chatbot can reply to user queries instantly and help in recapturing reach and engage Facebook users.

3.10. Project Summary

We have successfully implemented a button-based and NLP based messenger chatbot, taking a travel agency as our case study. It answers users' queries instantly when it clicks the answer it wants. We have enhanced the look of our bot using carousels and enhanced readability and understandability of our defined message. Our main focus was on conversational flow so we can never leave our users hanging in the middle of the conversation. We have also added made and integrated APIs for performing different functions like showing tickets of different categories, storing user ticket related data in the database, and making invoices by picking up the data from the database and integrated these functionalities using JSON API’s.

3.11. Future Work

We have successfully implemented button-based and NLP based chatbot that respond to user queries. In the future, we are planning to build functionality in our bot which will books the ticket for the user automatically. Users don't have to fill the form by going to a particular website, the bot will do everything itself. Further, we will improve the testing and enhance the accuracy of our bot by training it on a larger dataset. Moreover, after implementing these things successfully, we will build some more unique features in our chatbot like if someone goes to an unfamiliar place, the user will be able to see and find hotels and restaurants of his/her own will and will be able to book them in advance by using our chatbot premier features.

Developing an AI chatbot

ORIGINALITY REPORT

0%	0%	0%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.whoson.com Internet Source	<1%
2	pt.scribd.com Internet Source	<1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		

turnitin

TURNITIN Originality Report

Title	**Developing an AI chatbot**

Author	**Mahnoor, Ahtasham Sandhu & Hamza Tahir**

Submission Date	03-Aug-2020 04:34AM (UTC-0700)
Submitted Class	Reports
Submission ID	1365475012
Word Count	5529
Character Count	27127

Similarity Index	**0%** (Detailed report send to quarter concerned via email)

Remarks	**Report seems OK**

Note: Bibliography and quoted materials are excluded as per HEC rules

Report Generated By	**Nasira Muneer,**Assistant Librarian nmunir@cuilahore.edu.pk
Dated on	**August 3, 2020**

Note: Computer generated report doesn't need signature